DEVELOPMENT OF A PLATFORM STRATEGY FRAMEWORK

A good platform is like a box of Lego bricks.
Use and recombine components in any way you want,
they will always fit.

Elise B. Overbeek
2-10-2018
Master Business and IT
University of Twente
Supervisors University of Twente: Klaas Sikkel and Chintan Amrit
Supervisor Company: S. Nijenhuis

Management Summary

The goal of this thesis is to develop a platform strategy framework for the company Tersof (a pseudonym for the protection of the competitive position of the company).
A platform in this thesis is considered an online platform on which a company can present a basic offering to users while also inviting partners to create and offer complementary products on this platform. The platform strategy is developed to give structure and guidance to the efforts that are made to make the platform a success.
The platform strategy framework has been developed based on literature and the context of the company Tersof. The platform strategy framework has then been verified and validated by case studies. Based on the case studies a final revised platform strategy framework has been developed and is presented in the thesis.
From the literature study the most significant result was a list of 6 dimensions of platform strategy which contribute to a selection of critical success factors which are critical for the success of the platform. These six dimensions have been chosen based on how frequently they were mentioned in literature, how compelling the arguments for that dimension in the literature were and their relevance in the specific context of the company Tersof.
The six dimensions of platform strategy are trust, governance, pricing, usability, evolution, and users and partners. These six dimensions were found to be of great importance not only in the studied literature but also in practice as discovered during the case studies. The case studies illustrated the need for an updated version of the platform strategy framework which lead to the final revised version of the framework.
The concept of the platform strategy core has been introduced in the final version and is found at the heart of the framework. On the left side of the framework four dimensions are presented, trust, governance, usability and pricing. In these dimensions choices must be made which will contribute to the platform strategy core. On the right side the concepts of users and partners, and evolution can be found. These concepts are also related to the platform strategy core and to the other dimensions through their relation with the platform strategy core.
The revised framework supports three different viewpoints, or starting off points, to approach platform strategy. The boxes are connected with lines instead of directional arrows to illustrate the fact that the framework can be used and approached from different directions. The platform strategy can be developed with as a starting out point certain choices which are made in the dimensions, a certain strategy core that is deemed most important or by being led by the type of users, partners and evolution that is desirable. When the dimensions are leading they will determine the core of the strategy and as a result the users and partners that are attracted to the platform and how the platform evolves.
If the strategy core is chosen as the starting point the choices made in the dimensions much reflect this strategy core and the strategy core will attract certain users and partners and determine evolution.
If the desired users and partners and/or evolution is chosen as the starting point this determines what the core of the strategy will be and which choices have to be made in the dimensions to attract the desired users and partners or to orchestrate the desired evolutionary process.

The validated, revised and final platform strategy framework can be found on the next page.

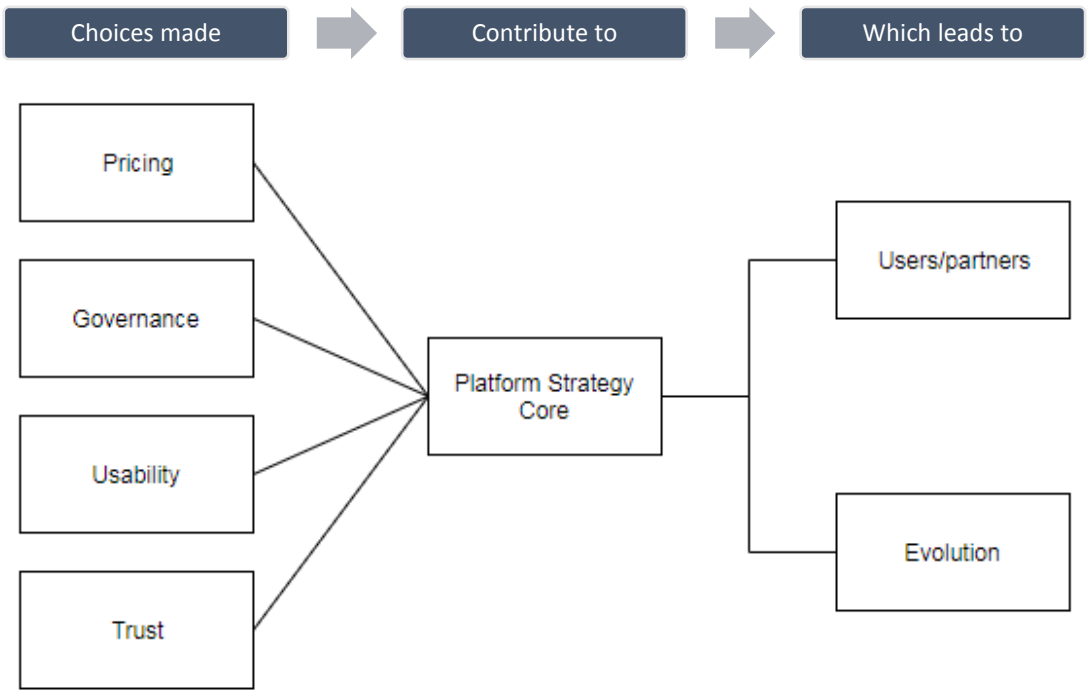


Figure 1: The revised and final platform strategy framework.

Preface

This thesis has been a once in a lifetime experience of six months of hard work and many new learning experiences, found at the intersection of the business world and the University of Twente.

Most of my fellow students have told me that during the first two months of your thesis project you have no idea what you are doing, during the third and fourth month it starts to become clear and during the fifth and sixth month you wish you had known from the start what you should do, because now you know, but you don't have enough time anymore to do everything you want to do. Now, looking back on the project I would have to say they were right. If I had known from the start what the final result would be it would have been much easier. That, however, is not how the world works and it definitely would not have been such a valuable learning experience.
The Tersof software is based on the think-do-learn principle, during my thesis I noticed the same structure happening in my personal process. I thought of a way to do something, I did it, and I learned how I should have done it. This was my first and last master thesis, and as such the learning curve on how to approach such a project was very steep and the process was one of the most challenging things I have come across during my years as a student. It was however also one of the most rewarding processes. I learned a lot about the 'real world' after so many years in school and at the university and I learned a lot about the subject of my thesis, platforms and the strategic world of software businesses, and about myself.
When working on my thesis at the office of Tersof I experienced firsthand how a software company operates from day to day. From daily stand up meetings and monthly reviews to long term strategy planning. After many years of learning in an educational setting it was nice to finally have the opportunity to see all the things I have learned about in practice. After many years of studying only for my own benefit it also feels great to be doing something that contributes to a larger goal and that contributes to the real world.
For me this assignment has been a perfect reflection of my study program and the electives I choose during it. Therefore it has been a very appropriate ending to my years of studying Business and IT at the University of Twente.

Acknowledgements

I would like to thank Klaas Sikkel and Chintan Amrit, my supervisors from the University of Twente for their support and guidance. They have helped me and encouraged me throughout the entire process and I greatly appreciate their support during this project.
I would also like to thank S. Nijenhuis for all his insights and contributions to this thesis and for giving me the opportunity to experience the day to day running's of a software company. It was an incredibly valuable learning experience that I will never forget.
Finally, a special thanks to the guys who played table tennis with me during the lunchbreaks at Tersof, for going easy on me in the beginning when I totally sucked and gracefully accepting defeat in the end.

Definitions and Acronyms

How a PaaS using APIs and a VCE can improve CRM and increase the NPS and contribute to SPM may be a bit of a confusing sentence for those who are not well-versed in the world of IT and Business acronyms. So in the next table the acronyms and core concepts that are used in this thesis are explained.

Concept/ acronym	Definition of concept
Business Model	A conceptual model that describes the product and services a company offers and how revenue streams are related to these product and services [1].
Platform	The shared basis of technologies, rules and agreements on which several players can innovate, and develop new products and services in a coordinated way [2]. In this research project a platform is considered a technological base on which applications can be build and run. Through APIs the platform facilitates interactions between the different applications on the platform.
Ecosystem	The platform together with its users, external complements, partners and external developers
Prince2	A change management methodology that was developed in 1996 by the British Office of Government Commerce.
SaaS	Software as a service
PaaS	Platform as a service
IaaS	Infrastructure as a service
B2B	Business to business
B2C	Business to consumer
C2C	Consumer to consumer
CSF	Critical Success Factor. Critical success factors are properties that the platform or platform business must have to be able to succeed in becoming a successful platform.
Multi-homing	A platform user or developer is active on several platforms at the same time.
API	Application Programming Interface
SDK	Software development kit
ISO	International Organization for Standardization which develops and publishes international standards.
On-premise	The IT and the hosting of the IT are done in-house.
Cloud hosted	Software and related services that are hosted on a remote location and can be accessed through the internet.
One Sided Platform	A platform that has only a single type of users on it.
Two Sided Platform	A platform with two different sides, either two types of users or users and partners.
Multisided Platform	A platform with two or more different sides participating on it.

Table of Contents

1. Introduction

In this thesis the process and results of the development of a platform strategy framework for the company Tersof Solutions B.V (henceforth: Tersof) are documented. Tersof is a pseudonym used to protect the competitive position and interests of the company.

The process of writing this thesis started with getting to know Tersof and the domain in which they operate. Learning more about the company, their strategy, and their plans for the future helped to create a frame of reference. Getting to know the company and the context of this thesis was done by interviewing several employees from different departments, attending review meetings, having meetings with my supervisor at the company, the director S. Nijenhuis, and by reading documentation about the company.

After this exploratory phase a final decision was made on the subject for the thesis that would benefit Tersof the most at this point in time, platform strategy. The goal of this thesis is therefore to design a platform strategy framework for Tersof.

About Tersof

Tersof is a software company located in Enschede and Amersfoort in The Netherlands. The company has about 40 employees and its change enabling software has 80.000 users at companies such as Friesland Campina, Vopak, and many Dutch government bodies. Tersof is going through a big change in its business proposition, the company has existed for 20 years and is gearing up for the future. The traditional product offered by Tersof is based on Prince2 methodologies. However, Agile practices are becoming more popular and therefore a new product, a multi app platform, is added to the Tersof value proposition. The new Tersof product supports working with Scaled Agile.

With the new Tersof product comes a shift in the entire organization. The new product will be available worldwide and not just on the traditional home market. The software is currently personally sold by the sales department, based on leads that have been pursued personally by the marketing department. After the sale has been made a consultant will visit the buyer to personally configure the software for the client. The new Tersof product is supposed to be much more plug and play. Users should be able to buy, download, install and configure the software themselves. This change from personal contact to enabling the users to do everything independently will influence and change many of the current business processes. Considering that Tersof has the goal to grow to 1.000.000 users in the next five years, self-service for users will be a key issue.

Another key issue is how to create software that is diverse enough to appeal to a large user base. This diversity would be extremely costly to create if all possible apps for the multi app platform would have to be build in-house. A solution to this is to become a platform business. In being a platform Tersof would allow other companies or individuals to customize the software and even create entire apps that will run on the Tersof multi app platform and work together with all the other apps that are offered on the multi app platform. To embrace this platform structure means to change the Tersof business model. Changing the business model for an IT company requires the company to look at many different aspects such as the technical realization, APIs, software development kits, ISO standards, business practices, jobs within the company, user experience design, privacy policies, financial structures and many more. As this is too extensive for a single master thesis the focus has been set on developing a platform strategy framework.

Problem Statement, Research Objective and Research Questions

Doing nothing is going backwards, because all your competitors keep moving, a common statement in business which has a certain amount of truth to it. To remain a relevant player in the business software market it is important to regularly change and update your value proposition. Tersof is changing their value proposition by introducing the new multi app platform that supports both agile working and more traditional styles of project management. Tersof is a relatively small company with about 20 developers, with their own development team they can only develop a limited amount of apps to be offered on the multi app platform. This is where the concept of a platform strategy comes into play. By allowing third parties to create apps for the Tersof platform the value proposition of the product is increased. This thesis focuses on the development of a platform strategy framework for Tersof.

Research objective: to develop a platform strategy framework for Tersof

Research Framework

To achieve this objective a research framework has been developed following the research design concept that was created by Verschuren and Doorewaard [3]. The framework is a visual representation of the different steps that will be taken to achieve the determined research objective [3]. It shows how the different phases in the research are connected and in which order they will be done [3]. The research process starts with a literature study on platform strategies and getting to know the context of this thesis, the company Tersof. After studying both the literature and the context on which the literature will be applied a platform strategy framework has been developed. After that two cases studies have been done in which in the case study companies, Atlassian and Salesforce, are compared and measured against the platform strategy framework to see whether the proposed framework contains the proper dimensions for a platform strategy framework. This validation is done by looking at the strategic decisions that the platform companies have made in the different proposed dimensions and if and how these decisions play an important role in supporting the overall platform strategy of the company. In the end this will result in a validated platform strategy framework.

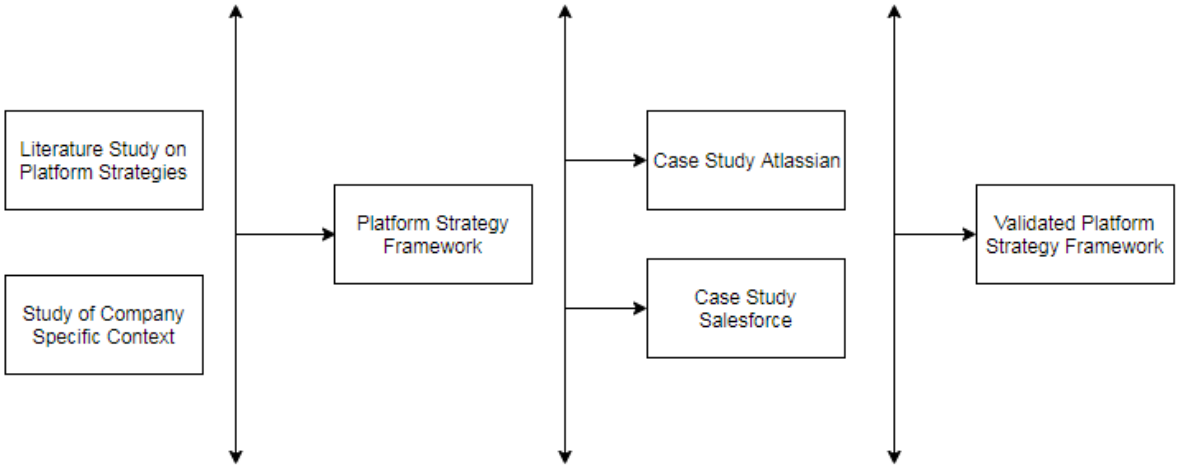


Figure 2: Research Framework

Based on this research framework the research questions have been created. To achieve the research objective the following research questions will be answered to obtain adequate and sufficient knowledge to develop the platform strategy framework.

1. Which critical success factors for platforms can be found in literature on platform strategies?
2. What are the different strategy dimensions in which choices are made to contribute to the achievement of the critical success factors?
    a. For each choice that can be made, what are the expected results of this choice and how does this contribute to the overall platform strategy goal?
3. What context specific requirements should be taken into account when developing a platform strategy framework for Tersof?
4. Based on the literature and the context of this thesis, which critical success factors and dimensions are the most important?
5. What is a suitable platform strategy framework for Tersof based on the answers from the previous questions?
6. For each of the case study companies does the developed platform strategy indeed encompass the most important efforts that are made towards their overall strategy?
7. Based on the case studies, is the platform strategy framework validated?

Structure of the Thesis

The chapter structure of this thesis follows the research framework that was presented earlier in this chapter. In the table below a brief summary of the content of each chapter can be seen.

Chapter	*Content*
Chapter 1: Introduction	Introduction of the thesis, the company and the research method
Chapter 2: Literature Study	A general introduction to platforms based on a literature study
Chapter 3: Platform Strategies from Literature	Overview of the 15 studied platform strategies from the literature
Chapter 4: The Context of the Strategy Development	Study of the company, Tersof, for which the platform strategy framework is developed
Chapter 5: Framework Development	Development of the platform strategy framework
Chapter 6: Case Studies	The case studies of Atlassian and Salesforce
Chapter 7: Revised Platform Strategy Framework	The framework is revised based on the results of the case studies
Chapter 8: Conclusion	The conclusions of the thesis
Appendices	

2. Literature Study on Platforms, Platform Strategies, Platform Strategy Dimensions and Critical Success Factors for Platforms

The goal of this chapter is to report on some existing platform strategies to be able to derive usable strategy dimensions and critical success factors from them. The chapter starts with some background information on IT Business models and a reflection on how platforms are a relatively new phenomenon which are affecting traditional business. Some background information on platforms is given. Then summaries of several platform strategies are given and finally the platform strategy dimensions and platform critical success factors that can be derived from those strategies are presented.

For this part of the thesis the following types of sources have been studied:

Scientific Literature	The scientific literature was found using Scopus and Google Scholar. The goal of this thesis is to develop a high-level platform strategy, so papers that fit with this objective were selected and studied.
Business and management literature	The business and management literature was found using Google, webpages from consultancy agencies and books found through managementboek.nl. Consultancy agencies such as McKinsey publish many articles and reports about business and digital developments. These sources often contain very recent and thoroughly researched data. The books were all written to support companies who want to become a platform or exploit their platform better. The books were recommended by the company supervisor and had good reviews.
Other sources	Because platforms as a business model are so fast changing in nature also some other sources such as blogs, news and other webpages have been used. This has been done to include the latest insights in this thesis which may not yet be covered in scientific or management literature.

IT Business Models

Traditionally a lot of software was sold on a cd or, before that on a floppy disk. As IT and the internet advanced traditional software product sales changed to more Software as a Service (SaaS) sales. This change is often referred to in scientific literature as the change from goods-dominant logic to service-dominant logic, a term first introduced by Stephen Vargo and Robert Lusch [4]–[7]. According to service-dominant logic services, and not goods, are the fundamental basis of economic exchange [6], [7]. In service-dominant logic a service is defined as anything that supports a consumer in their practices, be it resources or processes, that enables them to create value, in exchange for financial gains for the service provider, so a service is support that creates value for another entity's practices [4]. Looking at this concept Tersof is very much an example of service-dominant logic, they offer software to support change processes in other organizations. The Tersof slogan is: "we enable change". Enabling others with your product and offering this product as a service is the core of service dominant logic. Another interesting point from the service-dominant logic school of thought relates to value creation. According to service-dominant logic value is not created by a company [6], [7]. A company can only create value propositions [6], [7]. When a customer decides to use the product or service that is offered, then value can be created, therefore the customer is always a co-creator of value [6], [8]. This is very important for Tersof because the platform only has value when it is used by customers. The concept of value co-creation is also very

interesting because as a platform there is no value unless customers and other third parties are participating on the platform and co-creating value with Tersof.
This change to service-dominant logic in the IT business was made possible because of the creation of the cloud. Companies like Amazon, Google, IBM and Microsoft were among the first to create cloud architectures [9]. They initially used them for their own processes and later started selling their excess capacity [9]. Tersof initially provided an on-premise product, and many clients still use the on-premise version. However, one of the goals for the future of Tersof is to have more clients transfer from on-premise to the cloud and to have all new clients use the cloud version of the Tersof product.
Tersof is a SaaS-provider and looking into how they may also offer some PaaS capabilities. For the sake of clarity and completeness a general introduction to the most common IT business models is presented below.

SaaS – Software as a Service

SaaS stands for software as a service. In SaaS agreements a company supplies applications to a customer. The applications are running on a cloud infrastructure provided by the supplier [10]. The client can access the applications which are running remotely through a thin client interface, for example a web browser, or through an API [10]. Management, control and maintenance of the applications and the system on which they are running is the responsibility of the SaaS provider [10]. The infrastructure, network, servers, operating systems, storage and application capabilities are also all taken care of by the SaaS provider [10]. For the client it is plug and play, and therefore ideal for smaller organizations or any organizations who do not want to invest in their own software and hardware infrastructure. The user may have limited options for application configuration [10].
Tersof currently offers a SaaS solution. Customers can access the Tersof products through a web browser and the software and underlying structures are maintained by Tersof who in turn receives a subscription fee from its users.

PaaS – Platform as a Service

PaaS stands for Platform-as-a-Service. In a PaaS configuration the customer can deploy created or acquired applications onto a cloud infrastructure provided by the PaaS provider. Those applications are build using specific programming languages, libraries, services and tools that have been provided by the PaaS provider [10]. The services provided by the PaaS provider include program deployment, execution support, data storage and management, authentication and user management, background tasking and more [11]. The user has no control over the underlying infrastructure, but can control which applications he deploys and has some configuration options for the application-hosting environment [10]. It is a distributed software system [11]. It exports a wide variety of scalable application services through APIs: application programming interfaces [11].
PaaS is a software architecture that significantly simplifies and automates deployment and management of internet-accessible applications [11]. It does so by providing access to and management of scalable application infrastructure services. These services facilitate application execution (via load balancers, application servers, and background tasking) and provide seamlessly scalable implementations of components that are commonly used by web-facing applications [11], [12].

Polyglot PaaS

A variation on traditional PaaS is Polyglot PaaS. A polyglot PaaS structure supports applications which are written in multiple high-level programming languages [11]. Using several languages can be preferable over supporting only a single language as each programming language has its own strengths. Tersof for example uses several different languages and frameworks to construct their application. When offering a platform for others to build on it can be a selling point to allow potential developers to use different languages as this will increase the likelihood that your platform supports a language the developer is familiar with. However, on the technical side of the platform it may greatly increase the complexity when more different programming languages have to be supported.

IaaS – Infrastructure as a Service

IaaS stands for Infrastructure-as-a-Service. The service provider provides capabilities for processing, storage, networks and other computing resources, the customer can deploy and run any software, including applications and operating systems [10]. The infrastructure is controlled by the supplier, the customers have control over operating systems, storage and deployed applications; they have possibly limited control over the selection of networking components [10].

Ecosystems

An ecosystem is an interconnected set of services and products that fulfils several needs customers may have in a single integrated experience [13]. Ecosystem business models are the result of the accelerated pace of change due the increase in volume of electronic data, the ubiquity of mobile interfaces and the growing power of artificial intelligence [13]. Google and Amazon for example are typical ecosystem players, they offer many different services from a single place thereby redefining traditional industry boundaries [13]. Ecosystems provide additional value compared to traditional business models, through acting as a gateway to many different services [13]. They reduce the friction that customers experience while switching between related services through integration of several user interfaces into a single user interface [13]. An example is Facebook Messenger which enables its users to shop online, check into hotels, talk to their Facebook friends, read news and chat with external parties [13]. A user could do all of this within the single Facebook Messenger interface. There is no need to switch between different service providers, websites or portals and thereby friction is reduced. Ecosystems also harness network effects, a powerful concept, which will be discussed later in this thesis. McKinsey expects that 12 massive ecosystems will emerge by 2025 which will account for 60 trillion dollars, which is 30% of all global revenues [13].

Platforms

With this ecosystem economy comes the rise of the platforms. A platform is a business model in which the provider of the platform offers an infrastructure on which platform players can offer products and services to customers [13]. A platform enables the creation of an ecosystem. Both platforms and ecosystems are strategies that are all about embracing and enabling external entities [14]. Value is created in a joined process with these external entities [14]. With a platform or ecosystem a large complementor community is created that offers products and services that enhance the company's base offering [14]. Digital cloud based platforms are a fundamental feature of the digital revolution and are enabled by recent technological advances [9], [15]. Most major internet businesses are based on platform strategies. Amazon, YouTube, Uber, Airbnb, Takeaway and Facebook for example are platform businesses

and many people use these platforms every day [13]. Many people cannot imagine a world without social media anymore, this dependency on these platforms gives the owners of these platforms tremendous power. A major benefit of platforms is the scalability of the strategy. Because the platform owner does not need to own anything, except for the platform, large numbers of any product or service can be offered without any inventory risk or maintenance costs for the platform owner. Airbnb for example has an inventory of more than a million rooms, without having to build or maintain any of them and has become hugely successful with this strategy [13].

The platform strategy book by Amrit Tiwana gives a nice overview of how platforms differ from traditional business models, shown in the table below [16].

Driver	Traditional Product/Service Markets	Platform Markets
Migration from product competition to platform competition	A good product that offers a valuable value proposition to customers has a fair shot in the market	Rival platforms' ecosystems compete with each other; a good product without a compelling ecosystem has no shot in the market
Organizational boundaries blur	Coordination is achieved through authority and command-and-control structures	Conventional coordination mechanisms cannot scale to large platforms; alternative coordination mechanisms must be created
Architecture matters	Architecture rarely enters strategic thinking beyond centralized and decentralized organizational design choices	Architecture provides the blueprint for coordination across thousands of ecosystem partners where conventional coordination mechanisms fall apart
Evolutionary fit—not just efficiency—determines a platform's fate	Focus on operational efficiency and maximizing predictability	Platforms that evolve faster outlast their rivals
Coevolution of architecture and governance	Architecture of products and services are designed separately from the governance of the organization that produces them	Architecture and governance are the two gears of a platform's evolutionary motor; the two must be mutually reinforcing, interlock, and coevolve

Market Disruption

Of all discussed business models platforms have the most potential for exponential growth and therefore the biggest potential to be disruptive. Establishing a successful platform will cause a disruption in the market [17]. A disruption in the market influences bot the supply and demand side of the market. A disruption comes from a company that has an innovative good or service, combined with an innovative business model and an aggressive marketing strategy which will cause the expectations of customer to change. A disruptive value proposition is one that removes some type of friction. The telephone for example removed the friction of not being reachable. Other gigantic disruptors where the invention of the internet and the smartphone. The everyday lives of people have dramatically changed because of these disruptions. Collaborating becomes essential and new partnerships are created [17].

Phases of Disruption

The arrival of new players on the market has consequences for the old players [17]. In the first phase, a new player with an innovative concept can grow quickly and gain market share [17]. Often a new player uses a low pricing strategy with small margins and very limited profits at this stage of the disruption [17]. In the second phase the old players in the market are starting to feel the pressure of the low prices of their new competitor. Old players may have a tendency to ignore the new threat or underestimate the severity of the new threat [17]. Once it has sunk in that they need to change to remain competitive they will lower their prices to match the new player [17]. However, the old players do not have the business model in place with which they can be profitable at such prices [17]. So when most of the competition has disappeared the new player can increase the prices [17]. The new player with an innovative business model, which enables it to bind customers to the company, can therefor remain successful for a long period of time [17].

Platform Core Concepts

A platform, as the concept is used in this thesis, is a place where an entity (business or consumer) can offer something to another entity (B2B, B2C or C2C), it enables connections and transactions between parties by offering a solid base on which entities can build or which other entities can use. In practice to actually create a business software platform there are many details, concepts and decisions to consider. A platform can also offer a base offering itself, stand-alone value, next to the enabling of others to offer goods or services on the platform. Tersof for example already has several apps to offer, which in the future might be complemented by additional apps from external developers.

Platform Stakeholders

A platform derives its right of existence from the people and organizations who use the platform. In this section the main stakeholders of a platform are discussed including the different terms which are used for these stakeholders.

The Platform Provider

The platform provider, sometimes also called the platform owner or platform supplier, is the party who creates and maintains the platform and represents the legal entity that owns the platform [18], [19]. The goal of this stakeholder is to gain financial benefits from running the platform. The platform owner designs and develops the platform, controls the intellectual property and is in charge of running the platform [20]. The owner also has to manage the users and partners who add value to the platform [20]. In this thesis the platform provider is Tersof and as commissioner of the thesis the platform provider Tersof is the main stakeholder of this thesis.

The Platform Developer

The platform developer is any third party, a company or an individual, who creates a product or service on the platform or offers a product or service through the platform [18], [19]. Platform developers are also known as platform players, platform partners or platform complementors. The goal of this stakeholder is to gain financial benefits by offering applications for sale on the platform. Their applications can contribute to making the software more suited to a specific company situation by offering additional features to the end users. A developer has a partnership with the platform owner. His side of the bargain is to offer complementary products and services that run on the platform [20]. The API plays a central role in this connection.

The Platform User or End User

The platform user is the individual or company who uses the platform and the applications that are available on it [18], [19]. Use in the case of Tersof is the usage of the basis software offering of the platform combined with third parties applications, configurations or sharing of knowledge through the platform. For this stakeholder it is important the platform is easy to use and not too expensive.
The goal of this stakeholder in the case of Tersof is to create value for the organization for which the user is working. The goal of any company using Tersof software is to enable change and they have chosen to support this process using software.

The Platform Buyer

In many cases the person who buys applications on the platform or subscribes to the platform is the same as the end user. However, in large companies the person who decides to buy and install the platform may be a different person than the actual end user. Because Tersof is expecting to get more smaller clients in the future in this thesis the buyer is considered to have the same goals and wishes as the end user of the platform as the buyer is the representative for the end users and wishes to achieve the same organizational goals.

The book by Amrit Tiwana offers an overview of what the different value propositions of a platform are for the different stakeholders groups [16]

Stakeholder group	Value proposition	Explanation
Platform owner	Massively distributed innovation	Innovation on a scale and scope that would not be possible with a traditional business model. Increased likelihood that innovations will be created that will differentiate the platform in the market. The platform owner can focus on core activities. The rate of innovation around the platform is increase by the large number of complementors. This increased innovation rate will raise the bar for rival platforms.
	Risk transfer	The risk of any new innovation around the platform lies with the external party developing it. Only the costs of maintaining the platform are for the platform owner and can be spread across all the different third parties on the platform.
	Capturing the long tail	Both the mass market and the niche markets can be attracted to the platform. Catering to the niche markets is done by complementors of the platform. Catering to them would not be possible by traditional businesses. Mass customization to meet unique needs.
	Competitive sustainability	Competitive sustainability is increased. The more utility the platform offers the more attractive it is. With more users the platform is more attractive to developers. Once the virtuous cycle is established it becomes difficult for rivals to compete.
App developers	Technological foundation	Provides developers the advantages of scale without the burden of ownership. The platform has already invested in the shared assets for all apps. Apps developers need only create their specific functionality that creates distinct value. They do not need to make common elements are provide any infrastructure so there is a lower barrier to entry. Developing for long tail markets becomes economically viable.
	Market access	Developers can reach markets they would otherwise not have been able to. There is a large pool of prospective buyers. The platform will reduce the transaction and distribution costs.
End users	Mix-and-match customization	A platform that offers many complements allows for an immense level of customization. Every person can create their own version of the platform by mixing and matching different apps.
	Faster innovation and network benefits	Platform will continue to evolve after adoption. The competition between platforms will speed up the rate of innovation which will benefit the end user. Innovation on the platform will be an incentive for more users to join the platform creating network effects.
	Competition among rivals	Rival platforms will compete and that will lead to the creation of more value for users. Competition will first be around offering superior functionality and later shift to lowest price competition.
	Lower search and transaction costs	Search costs are brought down by a platform. A user trusts the platform and can therefore trust any party who operates on the platform. Reviews can make choices easier and additional information can be provided. Transactions are standardized on the platform and therefore have lower costs.

Platform Components

A platform consists of three main types of components: physical components, data components and marketplace or community components [18]. Every platform has these components and the components combined make up the platform. The physical component consists of the technology that makes the platform work, the infrastructure, communication protocols and standards for example. The physical component works together with the data component which is responsible for the information exchange between the different parts of the platform and the data analysis. Finally the marketplace and community components bring together the different entities that are active on the platform, so they can exchange goods, services and knowledge.

Platform Components	
Physical component	Digital technology Standards Communication protocol Product or infrastructure
Data component	Information exchange between platform parts Data analysis
Marketplace or community component	Bring together consumers and producers Reputation mechanisms

Modularization

The concept behind the technology of a platform is the use of standardized building blocks which can be modified, reconfigured and extended [2]. This is called modularization, which is breaking a system up into chunks that communicate through standardized interfaces to increase reusability and flexibility because of more possible configurations [21]. By recombining the chunks or modules a large variety of systems can be build. This structure allows for rapid changes and decreases innovation costs [2], [21]. All the modules together are the platform [21]. A platform provides core functionality that can be extended by other modules through standardized interfaces [21]. This enables and encourages platform and ecosystem growth because adding another piece to the modularized structure is easy [14]. The benefit of using modularization is also that it is just as easy to add the first app as it is to add several thousands of apps, because they all use the same interface structure, so the base platform can always remain the same.

API's and SDK's

API's

The modularization of platforms is made possible by the use of standard interfaces. These interfaces are called APIs and they connect the different elements, or modules, of the platform. API stands for application programming interface. The API contains information, specifications and design rules that describe how interactions and information exchanges on the platform work [19]. With an API someone who wants to develop an application for the platform does not need to know everything about how the software works, the API takes care of any communication that the external app needs to have with the platform. So a developer only needs to understand the API which is much less complex than understanding the entire platform. A standardized API allows the applications to work together seamlessly with the hardware and software [2]. A good API is like a black box, stuff goes in and stuff comes out, without there being a need to understand what happens inside the box [18]. The easier the API appears to be from the outside the more user or developer friendly it will be and the more likely therefore that developers will be willing to develop applications for the platform using the API. There are

standard APIs available, such as the REST and SOAP API which are frequently used, or a platform provider could decide to make their own API. To create optimal support it can be an option to combine several API's [18]. Because adding additional functionality to the platform is made easy by APIs they offer great opportunities for scaling up and they can make the platform more agile because it is easy to add and remove components from the platform [18].

Software Development Kits

As a part of the platform a Software Development Kit (SDK) is usually provided [12], [18]. This SDK allows third parties to create software that can run on the platform. Tersof already has a very nice SDK that is used within the company for software development. It could be an option to make this SDK available to third parties. Another option would be to supply a more basic and easier to provide and use SKD to third party developers.

Platform Lifecycle

There are three dimensions in the platform lifecycle. Whether the platform is in the pre- or post-dominant design phase, the adoption among end users which is shown in the next figure and the phase along the s-curve which is explained after that.

There are two main stages in the platform lifecycle described in the book by Amrit Tiwana: the pre-dominant design phase and the post dominant design phase of the platform evolution lifecycle [16].

A new technological development will lead to a mass entry of many firms on the newly created market with competing alternative designs which address roughly the same needs of the market. They will offer different features and test different capabilities and designs on the market. These varying designs will be tested and improved until at some point a certain design will become widely accepted and thereby becomes the winning standard. Some competitors will try to imitate this winning design or make small improvements on it, and most will leave the market. The design will be continuously improved, making it more difficult to introduce a new, radically different design on the market. Complements will be created to work with the dominant design making it even more appealing to users. Finally there will only be one design left, this is called the rule of one [16].

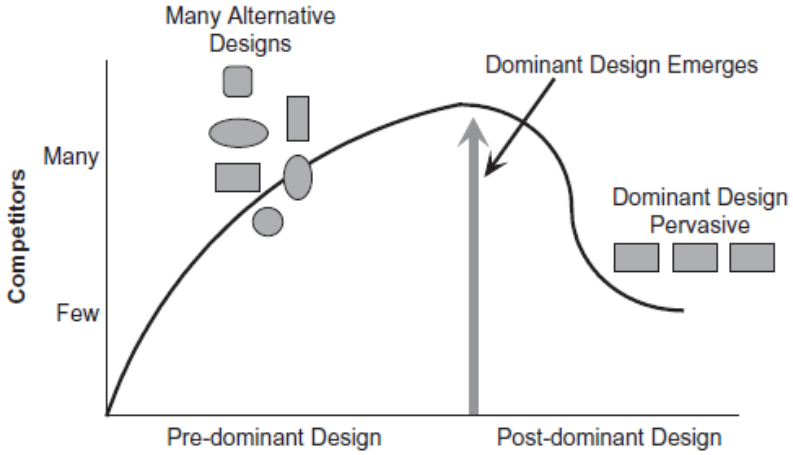


Figure 3: Pre- and Post-dominant Design Phases [16].

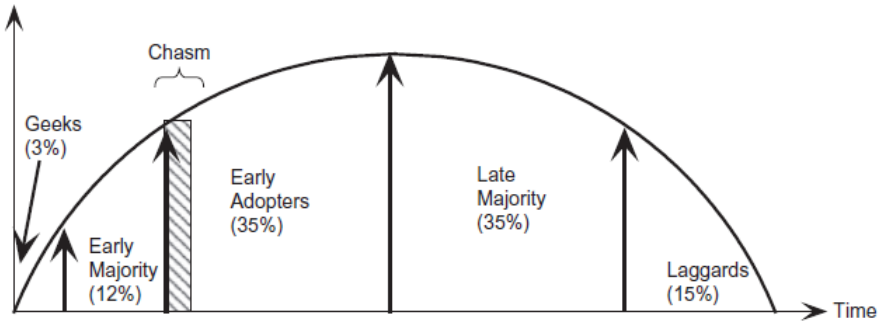


Figure 4: Technology diffusion lifecycle on the end-user side [16].

The second dimension is about the adoption of the platform. Figure 4 shows the average distribution of how technology is adopted. Technology adoption reaches from the very first 'geeks' who adopt to finally the 'laggards' who are the last people to adopt a new technology.
The third dimension is about using succeeding technologies. The s-curve is a construct that is used to describe how technology rises and declines and how company can jump from the top of the curve to the top of the next curve by adopting the succeeding technology to stay competitive.

There are four phases on the s-curve:

0. Research and development stage
1. Introductory stage
2. Ascent stage (break-even is reached and the platform is beginning to gather steam)
3. Maturity stage
4. Decline phase (the platform is not as useful as it used to be)

When the decline phase is reached the company should leapfrog to the next S-curve or accept that usage of the technology will continue to decline.

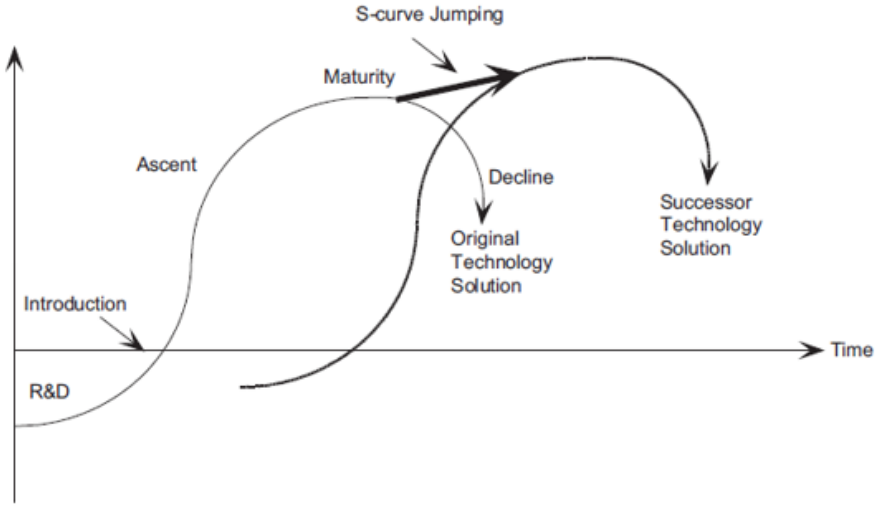


Figure 5: The Technology Lifecycle S-curve [16]

This final figure shows the three dimensions of the technology lifecycle combined.

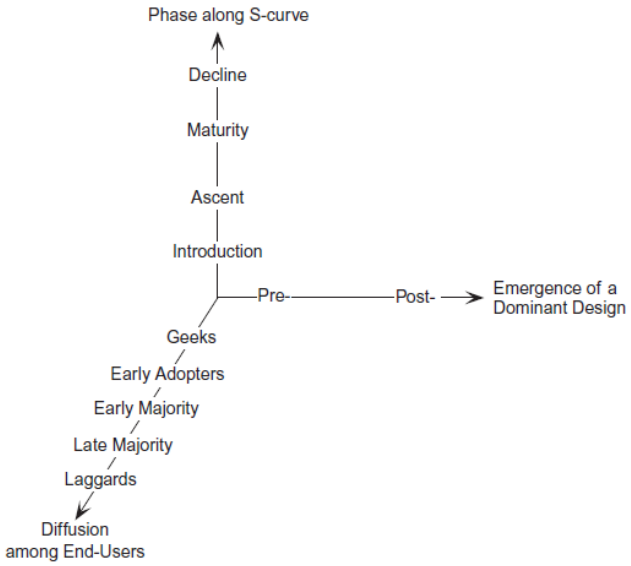


Figure 6: The Three Dimensions of the Technology Lifecycle

Market Leadership

The trust in a market leader is often leading in decision making [1]. Consumers believe that by buying their software from a market leader they are ensured that the software will remain supported for years to come [1]. Market leadership is also seen as a measurement of quality, if most consumers chose a certain product they must have a good reason for choosing that particular product [1].

A platform market goes through three stages [1].
1. The young market. When the market is still young there are often many different providers of a certain type of platform. Consumers are testing the different platforms.
2. The growth phase. During this phase usually two or three dominant vendors emerge.
3. The market is mature, there are two or three winners in the market and the smaller providers have become almost non-existent.

Following these phases a company only has a short amount of time to become a market leader. Therefore if market leadership is an important goal for a company it may be advisable to make market leadership a priority over other goals such as revenue and profitability [1].
The reputation a company may have as a leading party can be an important factor in getting third parties to join the platform [2].
Customers don't switch from platform easily. Because of this once platforms have a position as market leader they can endure for many years and after the initial investments the margins are high [22]. Switching costs are high so once a choice for a platform has been made it tends not to change. Winning platform companies can create 100-fold returns [22].

Platform Stickiness

Stickiness is how actively the platform is used by users and complementors [16]. Users remain stuck to the platform if it keeps evolving and improving. The level of rivalry in the market will influence stickiness. End-user stickiness is the increase or decrease in average usage per user over time.

Ways to Increase Stickiness of End Users [16]:

1. There is a correlation between complementor and user stickiness so increasing complementor stickiness will increase user stickiness. Driving innovation by supporting complementors will create a more valuable platform for end-users.
2. Have a high cost of switching that is noncoercive (offer added value that is not available in a different ecosystem) and value-driven and thereby create a lock-in of users.
3. Be strategically incompatible with rival platforms. A platform can be compatible, one-way compatible or not compatible with rival platforms. This is something that can be varied to throughout the platform lifecycle to best suit the situation at the time. By being incompatible it becomes difficult for users to switch to a different platform. When networks effects are difficult to generate one-way compatibility can be a good choice.

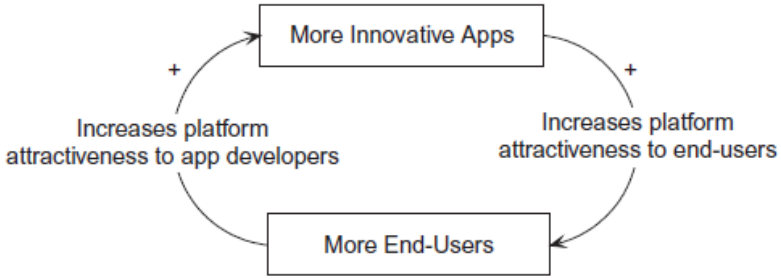


Figure 7: Virtuous Loop Between End-user Stickiness and App-developers Stickiness [16].

Platform Metrics

Metrics can be used to remain informed on how the platform is evolving. If the platform is moving in the wrong direction metrics can help to spot this early on and adjust accordingly. Platform metrics help to steer evolution, separate signals from noise and manage trade-offs [16]. When using platform metrics a short term focus is important for the measure, adjust, and measure again process, but this process should contribute to the long term platform goals [16]. A risk when using metrics is that the tracking and measuring may actually cost more than the amount of value that can be gained from it [16].
The book by Amrit Tiwana suggest nine metrics for platform evolution, focusses on the short, medium and long term; those are summarized in the table below [16].

Nine metrics of evolution	
Short term	
Resilience	How well subsystems continue to function when there is a failure outside of the ecosystem
Scalability	The subsystem must remain economically viable and deliver good functional performance when the ecosystem grows
Composability	How easy it is to change a single component without losing the seamless integration with all other components
Medium term	
Stickiness	How many users continue to use the subsystem
Platform Synergy	How much the app is specifically designed for this platform

Nine metrics of evolution	
Plasticity	How easily a system can be adapted to offer different value than what it's originally intended functionality was.
Long term	
Envelopment	When a subsystem also starts offering the functionality of an adjacent market with overlapping users.
Durability	How much endurance the subsystem has on a competitive market
Mutation	A system that is derived from an existing system and inherent certain properties but is created with a different purpose.

The book also suggests measurable variables to track the performance of each of these metrics as summarized in the table below [16].

Metric	Measurable Proxy
Short Term	
Resilience	• Recovery time of a platform or app after a failure *outside* it
Scalability	• Increase in the subsystem's latency, responsiveness, or error rates per additional 1,000 users • Direction of the shift in the subsystem's financial breakeven point per 1,000 fewer users
Composability	• Integration effort (person-hours) per internal change
Medium Term	
Stickiness	• Change in hours per end-user session over time • Change in averaged end-user sessions per week over time • Change in API calls made by an app on average as the platform ages (platform-level only)
Platform synergy	• Change in the number of functions called by app to APIs unique to platform
Plasticity	• Average count of major features added to a subsystem per release over its lifetime
Long Term	
Envelopment	• Count of successful envelopment moves • Count of envelopment attacks rebuffed • Percent of *new* subsystem adopters using enveloped functionality
Durability	• Change in the percentage of a subsystem's initial adopters who remain active users • Change in a platform's percentage of apps released that are subsequently updated at least once a year
Mutation	• Number of *unrelated* derivative platforms relative to rival platforms • Percentage of carryover users at outset of derivative subsystem • Growth of an app into a platform

The next figure shows for each metric whether they focus on the short, medium or long term, if they are strategic or operational and for some of the metrics how they relate to each other [16].

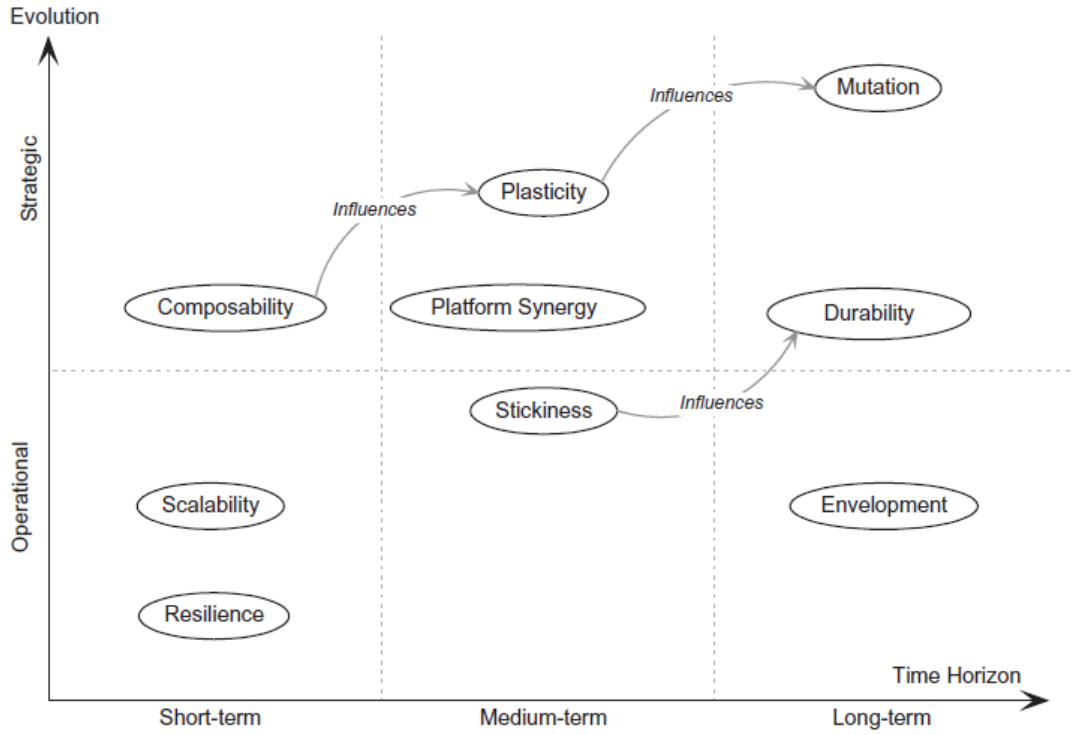


Figure 8: Evolution of Metrics

3. Platform Strategies from Literature

Every book, blog and paper about platforms presents its own version of what the writer thinks the ideal platform strategy is or what the possible strategies are. These different strategies often share some dimensions but differ on other dimensions. A platform strategy dimension in this thesis is considered any variable that can be used for a platform strategy. Being a variable, the platform dimension has at least two different possible values. By changing these values different results can be obtained from the strategy. The combination of the different platform dimensions with the different values and corresponding results of these variable values comprises the platform strategy which will contribute to the critical success factors of the platform. You could compare the strategy to a sound or light desk as you can find in music venues, with a bunch of buttons, faders, levers and dials who each influence different things, by altering the settings of all those variables different results can be obtained. For some strategy dimensions there is a simple yes or no, on or off choice, for other dimensions there is an endless range of possible values like you would have with a fader. If the result is not what you expected or wanted to achieve, you change the settings on the desk, and the new combination of values in the strategy will lead to new results. Just like sound checking or setting stage lights, it is a process of adjusting settings and trying out different settings and combining them in the right way until you get the result you want.

Figure 9: Pearl Tiger light desk, [138].

Limiting the Thesis

Presenting and reflecting on all platform strategies that have ever been published is so time consuming that it falls way beyond the scope of this thesis. Therefore only 15 platform strategies have been studied in detail. The strategies that have been used have been selected based on recommendations by the company project supervisor and a literature search. For each of these strategies a short summary will be given first and in the second part of this section each identified strategy dimension and critical success factors will be discussed along with the possible values and results of each value. Some strategies are very detailed where others have chosen a more high-level view, one paper even looked at more than a hundred platform strategies to find a set of platform strategy dimensions. The chosen strategies are from different types of sources and different years to get a broad cross section of platform strategy dimensions and critical success factors.

When I started working on this section I expected to find variables that you can give different values to in order to create different strategies. However, many strategies found in literature focus on what the result of the strategy should be, the critical success factors, much more than how to achieve that result. I think a strategy should be about how to achieve the goal and not just what the goal is. So when a strategy provided only goals and no usable variables on how to achieve that goal these goals have been considered critical success factors for a platform which can provide guidance for what the result of the strategy should be and how different dimensions could lead to that result.

The following strategies have been studied:

	Strategy	Published by	Publication title	Publisher	Source type
1	Toolbox, Magnet, Matchmaker	[23]	Three elements of a Successful Platform Strategy	Harvard Business Review	Web article
2	2 Question Strategy	[24]	Platform Strategies Explained	MIT Sloan Management Review	Web article
3	10 Basic Rules	[2]	De kracht van platformen (The power of platforms)	Vakmedianet	Management book
4	5p Model	[20]	Five Ways to Win with Digital Platforms	Accenture Strategy	Consultancy report
5	6 Rules for Disruptive Two-way Platforms	[17]	De Kracht van Platformstrategie (The power of platformstrategy)	Boom	Management book
6	Explore, Plan, Deliver and Evolve	[25]	Every Organization needs a Digital Platform Strategy	Gartner	Web article
7	Ambidexterity Perspective	[26]	Unraveling Platform Strategies: A Review from an Organizational Ambidexterity Perspective	Journal of Sustainability	Scientific paper
8	The Nine Guiding Principles	[16]	Platform Ecosystems – Aligning Architecture, Governance and Strategy	Elsevier	Scientific research book
9	8 Ways to Launch a Successful Digital Platform	[27]	8 Ways to Launch a Successful Digital Platform	MIT Sloan Management Review	Web article
10	Platform Evolution	[18]	De plug & play organisatie (the plug and play organization)	Vakmedianet	Management book
11	Characteristics and Success Factors of Digital Platforms	[28]	Characteristics and Success Factors of Digital Platforms	iit Institut für Innovation und Technik	Consultancy report
12	5 Questions	[29]	How to Launch your Digital Platform	Harvard Business Review	Web article
13	Platform Leadership Components	[30]	Success factors of platform leadership in web 2.0 service business	Springer	Scientific article
14	Innovation versus Breaching	[31]	Innovation and Breaching Strategies in Multi-Sided Platform Markets: Insights from a Simulation Study	Thirty Sixth International Conference on Information Systems	Scientific article
15	Bundling or Constellation	[32]	Platform extension design as a strategic choice	Twenty-Fourth European Conference on Information Systems Proceedings	Scientific article

Platform Strategy 1: Toolbox, Magnet and Matchmaker

This strategy was published in the Harvard Business Review [23]. According to the article by Bonchek and Choudary there are three components to a successful platform strategy. All three components need consideration, however, only focusing on one or two of the components can also lead to a successful platform strategy.

1. Connection – The Toolbox

The connection element of a platform strategy refers to how easy or difficult it is for other companies or individuals to plug into your platform and start sharing knowledge and making transactions. The toolbox is what should make plugging in and creating connections easy. The toolbox contains the infrastructure that enables interactions between the various participants on the platform.

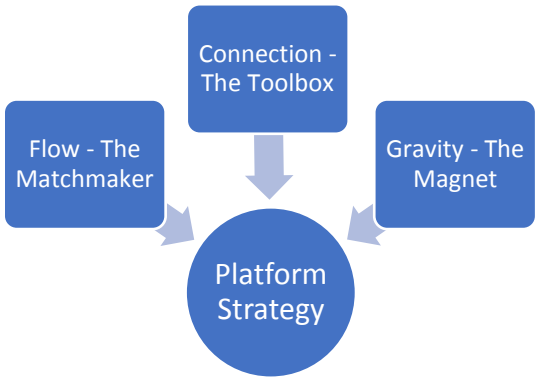


Figure 10: Toolbox, Magnet and Matchmaker

2. Gravity – The Magnet

Gravity, or gravitational force, refers to attracting producers and users to your platform. The magnet is what should create the gravitational forces towards the platform and pull participants towards it. A platform needs a certain critical mass to work and that mass consists of users and producers. To get them to participate media can be used to harness the network effect, the more users there are the more users will join, and create rapid growth in the user base. The design of incentives is also a part of the magnet component, for example pricing models and reputational systems.

3. Flow – The Matchmaker

The platform should be a fostering environment that creates a flow experience for the exchange and co-creation of value. The matchmaker component is what will make the connections between producers and users on the platform. These matches can be made using data analysis and search refinement options.

Reflection on the Strategy

At first glance this strategy article focusses not really on the strategy dimensions but on the desired results that can be achieved by having the proper values for the dimensions. It is about the critical success factors of a platform, not so much about which strategic choices will lead to success.
The toolbox element that is mentioned is all about how easy it is to join the platform. How easy it is to join the platform depends on how open the platform is, or how strict the rules of the platform are, it depends on the costs that are related to joining the platform and maybe other dimensions but those are not mentioned in the article. So although some interesting critical success factors are mentioned, in terms of usable strategy dimensions and what the results of those are the article has little to offer.

Dimensions	Possible values	Positive effects	Negative effects
Ease of joining the platform	Easy to difficult	n/g (not given)	n/g
Pricing	n/g	n/g	n/g
Reputational systems	Yes/no	n/g	n/g

Critical Success factors for platforms	
Gravity	The need to have both users and developers on the platform. A platform needs to achieve critical mass.
Flow	Flowing of value should be fostered by creating connections between platform users.
Matchmaking	The users and developers need to be interested in each other. The developer should develop offerings that the users want. Successful matchmaking is achieved by using data.
Connection	An infrastructure should connect different platform participants.
Network effects	With more users the platform is more valuable to the other users.
Incentives	Provide incentives to join the platform.
Reputational systems	n/g (not given)

Platform Strategy 2: Church – 2 Question Strategy

In this article by Zach Church a platform strategy is considered an approach to enter a market [24]. The value of the platform, and therefore the willingness to pay for it, does not come from the individual product but from how many users and producers are active on the platform, so success is determined more by the strategy than by the platform itself.
According to Church platform strategy is not that complicated. Only two questions must be answered to create a platform strategy:

1. How will the platform attract customers?

2. How will you make your technology the core of an ecosystem?

The first question is related to the chicken-or-egg problem, how to get users to join the platform when there are not many other users yet. A good platform strategy needs a plan on how to overcome this problem. A seeding mindset, focusing on getting participants, is more important than a feature mindset, focusing on what the platform can do (also called product mentality) according to his platform strategy.
How to seed users:

- Recruit a marquee user. This can help kickstart the platform as this user will draw other users to the platform. The risk is that the marquee can usurp control of the platform.
- Low pricing or being free. There is a risk that when customers have gotten used to the low price they expect the platform to remain low priced in the long run.

Another part of this platform strategy is making your platform the core of an ecosystem: where customers and producers are meeting and where transactions are made. A platform needs governance and policies to properly fulfill such a core role.
A platform should be reliable and trusted. Reviews and guidelines can help to build trust among users. Other questions to consider are who can join the platform, what information will be available, will gathered data be openly available, and will others be able to use that data.

Reflection on the Strategy

This strategy also focusses on critical success factors, it does mention a few dimensions, mostly related to how to get users.

Dimensions	Possible values	Positive effects	Negative effects
User attraction strategy	Marquee user	Get users at the start of the platform	The marquee can take control over the platform
Pricing	Low or Free	Get many users fast	Users get used to the price and expect to stay low or free.
Availability of data gathered on the platform	Openly available to external parties	n/g	n/g
	Not available		

Critical Success factors for platforms	
Customers	A platform needs to attract customers and connect different groups. The chicken-or-egg problem needs to be overcome. The worth from a platform comes from the users, not the functionality.
Ecosystem core	It should be a strategic goal to have your platform be the core of an entire ecosystem.
Build trust	Have governance, rules, policies, guidelines and review systems to ensure trust and reliability on the platform.
Data Governance	Decide who can do what with which data.

Platform Strategy 3: Kreijveld's 10 Basic Rules

Maurits Kreijveld, the author of the book "The Power of Platforms"(original Dutch Title: De Kracht van Platformen) concludes his book with 10 basic rules to follow to build an economically successful platform [2].

There are two axes along which platforms can be measured and placed. Dimension 1: collaborative culture/innovation style. Dimension 2: openness towards new users and developers and applications. Platforms will move in the diagram as they evolve through time.

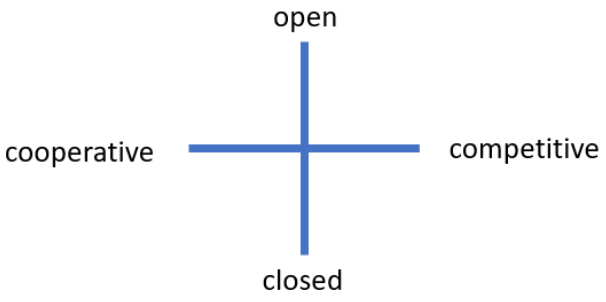


Figure 11: Dimensions of Platforms

Dimension 1: collaborative culture: cooperative or competitive. Platforms with a clear central leader who is in control versus platforms which are controlled by a dispersed community. Platforms have different governance structures ranging from centrally controlled to communities who decide together about governance issues.

Dimension 2: open or closed relates to how easy it is for developers to join the platform. An open platform is accessible to anyone who want to join, while on closed platforms there are strict rules about participation and what can be created on the platform.

The 10 rules

1. Open Your Product or Service

Make the decision to turn your product or service business into a platform business and open up your product to third parties, because innovation can be achieved faster with partners. Make sure you create building blocks that are easy to use and modify, so others can easily plug into your platform and offer their products and services over your platform, to achieve this develop an API to connect all users.

2. Scale Up

Find partners who can add value to your ecosystem. Create either closed partnerships or open up the platform to any third party to stimulate use of the platform. Absorb any new innovations made on the platform back into the platform so the innovations can evolve and be improved.

3. Cherise Developers and Partners

Make use of the newest technologies and software, thereby maximizing your developers capabilities to develop new products and services. Make design and development easy and take into account the costs for developers, such as necessary investments to participate in the platform. Listen to wishes and complaints and regularly create new opportunities. Enter into new markets as a joined effort with your partners to ensure value is created for all parties. Make sure that developers can make a profit by building for your platform. Develop opportunities for platform sponsoring through advertising and other financial models.

4. Prune when Necessary

Make sure that the platform does not get out of control or too many spin-off versions are created. Keep the platform alive by regularly absorbing new functions into to the platform, so developers do not get lazy but remain active with building new features. Standardization should be done in a timely manner.

5. Show Leadership and Trade

Consider an arbitration committee or governance structure to discuss the further development of the platform and make choices together about the specification of hardware, software and new markets. If you, as a platform owner, feel strong you can make these decisions alone. If a collaborative culture is important to you, make sure there is room for discussion. However, do consider that too much discussion and negotiation will decrease the speed with which the platform can be realized.

6. Share

Sharing information and data within the ecosystem is essential for the operational excellence of all those involved with the platform. Each participant should have access to the newest aggregated and combined information from the system. Data is one of the most valuable resources in a digital ecosystem. Combined data has much more value.

7. Cherish Users and Consumers

Have a varied and high-quality offering of hardware, software and services to make the platform interesting to users. Make sure the added value of using the platform is clear to the user. Make it easy to use. Create word-of-mouth, let your fans like, share and invite friends.

8. Stimulate Use

Create scale through sponsoring. Give access to the platform for free or be installed everywhere. Be imbued in processes, products and services.

9. Curate

Have a good reputation management system. Have a strong selection process for new ideas and projects, control quality. Help searching consumers find their way, make recommendations, offer templates and offer a helping hand.

10. Evaluate

Keep testing and determining whether your platform should be more open or closed. Openness can be a good strategy in the starting phase to get the platform started, being closed can offer security to those operating on your platform. Extend the reach of your platform to new markets to profit from network effects in those markets as well.

Reflection on the Strategy

This strategy presents a mixture of critical success factors and platform dimensions. There are many usable factors and dimensions mentioned, although for many dimensions the possible values and corresponding results are not given in the book.

Dimensions	Possible values	Positive effects	Negative effects
Openness	Open (more developers, services and applications, even the competition is allowed to operate on the platform.	Good in the starting phase to get the platform started, easier for user and developers to join the platform, more new apps are developed, more freedom for radical innovation, Innovation is faster with more partners	n/g
	Closed More intensive partnerships but the rest of the world is kept out	Security for those operating on the platform, better opportunities for investors to get their money back, better quality control, less noise, no unprofessional or slow developers and slow consensus processes, clear rules for quick results, more efficiency	Getting stuck because there is no room for alternatives
Scale	Company, several companies, value chain or industry wide	n/g	n/g
Partnerships	Close partnerships with carefully selected partners	n/g	n/g
	Opening up the platform to any partner who want to join	With more partners innovation is faster	
Modularization	Easy building blocks	Easy to develop for the platform, encourages developers to develop for your platform	n/g
Absorbing of platform layers (aka envelopment)	Yes/no	The new innovation can then be further developed and improved, the developers cannot get lazy and will remain active	The functionality that is absorbed belonged to an ecosystem partner who might not be happy that the functionality will now be a part of the base platform

Dimensions	Possible values	Positive effects	Negative effects
Reputation management	Yes/some/no	A good reputation management system will lead to trust	n/g
Sponsoring	Yes	Grow a specific side of the platform, being free can support the goal of being installed everywhere	n/g
Governance Competitive or collaborative	Complete freedom Consensus culture Collaborative Make decisions together with partners	Consensus culture improves collaboration among involved parties. Platform profits are shared equally which will improve long term collaboration	Discussions, meeting and finding consensus can take up a lot of time, a few slow developers can slow the entire platform development down, too much discussion and negotiation will decrease the speed with which the platform can be realized, Risk of dispersion if people can make their own separate versions of the platform as products and services might not work together anymore
	Control, strong leadership Competitive Make decisions alone	Maintain the user experience, Avoid an explosion of offerings, High development speed	Can scare developers away and limit innovation. When there are too many spin-offs the platform gets out of control.
Cost of switching	high	n/g	n/g
Possibility to integrate with other platforms	Yes/no	n/g	n/g
Allow competitors of developers on the platform	No	Incentive for developers to develop for the platform No dispersion	n/g
Horizontal/vertical integration	None, and, or	n/g	n/g
Market leadership or specialization	n/g	n/g	n/g
Ease of use	n/g	n/g	n/g
Social media integration	Yes, allowing liking, inviting and sharing	Create publicity for the platform	n/g

Critical Success factors for platforms	
Get partners	Innovation is faster with partners and they will add value to your ecosystem
Being imbued in processes, products and services	This will stimulate use
Modularization	Easy to modify building blocks will allows other to develop for the platform
Relationship management	n/g
Quality control	Ensure quality through reputation management systems
Network effect	Starts working when the chicken-or-egg problem is overcome
Absorbing of innovations into the platform	This way they can be further improved and developed upon and developers will remain active instead of getting lazy.
Use the newest technology	This maximizes the capability of developers to develop new products and services
Easy design and development on the platform	Keep the costs for developers low
Listen to wishes and complaints	n/g
Regularly create new opportunities	n/g

Critical Success factors for platforms	
Enter new markets as a joined effort with partners	This ensures that value is created for all involved parties
Develop sponsoring opportunities	n/g
Standardization	This should be done in a timely manner
Room for discussion	Make sure there is room for discussion on collaboratively managed platforms
Utilize the value of data	All participants should have access to all data as combined data has more value.
Have a variety of offerings at a high quality	This will make the platform interesting to a variety of users
Have clear added value	The user must have a clear understanding of what the added value of the platform is
Be easy to use	n/g
Create word-of-mouth	Let users like and share and invite their friends to join the platform
Scale	A platform needs to have a certain size, this can be achieved by giving easy access to the platform and thereby getting a large userbase dependent on your platform
Reputation management	n/g
Quality Control	Be strict in the selection of what new ideas and projects are allowed on the platform
Offer guidance	Help consumers to find their way on the platform and get the most out of it.
Evaluate	Keep testing the platform especially in regard to if the platform should be more open or closed
Extend your reach	Enter new markets and profit from more network effects

Platform Strategy 4: Accenture 5 P model

In their research report "Five Ways to Win with Digital Platforms" Accenture describes a platform strategy which has as core the 5 p's: proposition, personalization, price, protection and partners [20].
Being a platform owner is a balancing game. You have to continuously keep an equilibrium between the two sides of your market to generate network effects. Planning should be done for the long term and be focused on acquiring the critical mass and scale that is necessary for your platform to be successful. A strong niche offering can allow smaller players in the market to survive, but a truly successful platform has the ability to access growth capital and scale fast [20]. Issues to consider when building a platform are monopolies, competition, ownerships and intellectual property rights. An app supplier of a platform has access to a new (global) distribution channel and can get additional revenue and reduced transaction costs.

Figure 12: Accenture 5P Model

Proposition – Present a Compelling Solution through Modularity

The value proposition of a platform is not just the products being sold on it, but the users connected to it. The platform owner is the facilitator of user-to-user or business-to-business value creation. To create superior value for users, suppliers and partners in the ecosystem a platform must continuously innovate and update their value proposition and business model. APIs enable the modular approach to platform development and revenue growth. Salesforce, just like Tersof a business software B2B platform, generates 50% of its revenues through APIs.

Personalization – Center on the User Journey

A platform offers the opportunity to create a personalized user experience that focusses on the result for the customer and not just the products. Platforms can use mass personalization by targeting users with tailored experiences. By collecting data about the user and using that data a uniquely tailored experience can be created for each customer. However, data privacy laws may restrict these options.

Price – Engage Participants through Sophisticated, Dynamic Pricing

Having a flexible pricing policy creates opportunities for increased flexibility and rewards. There are many possible pricing strategies which can support different business strategies. By offering a freemium membership users can easily access your platform to experience it before taking a paid account. Pay-as-you-go pricing or fixed subscription fees can be used and even combined. Price can also be used as a tool to regulate supply and demand, by surge pricing, increasing the prices when demand is high, and discounting, decreasing the prices when demand is low. Especially in the initial starting phase of a platform pricing can be an incentive or demotivator for possible users and developers to join the platform.

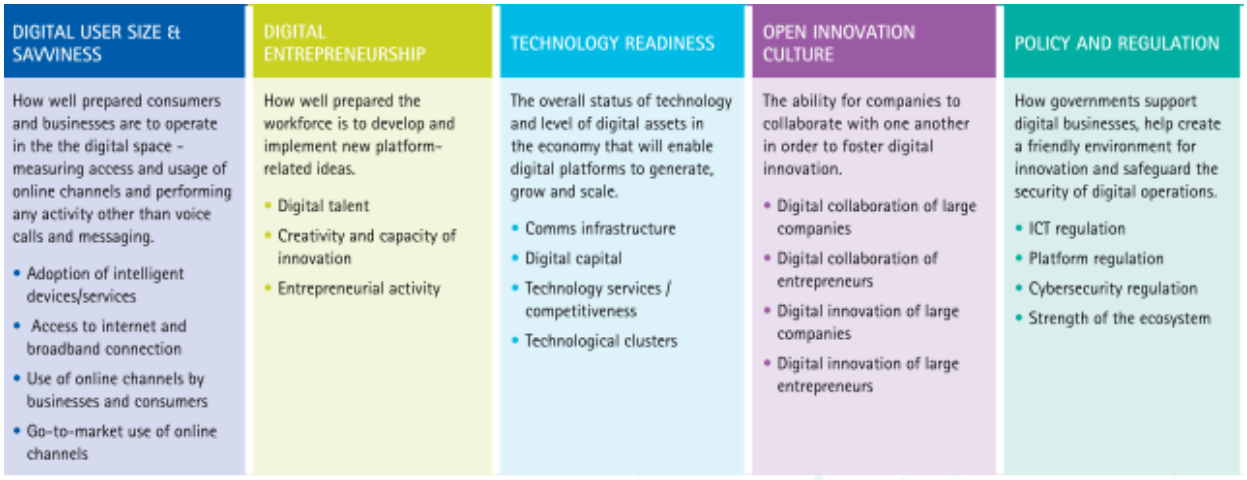
Protection - Embed Trust at the Heart of the Platform

Protection refers to the importance of cyber security and the importance that users of the platform trust that all cyber security issues are being handled properly. An example of a cyber security issue is proper authentication of community members. Commitment to security can be a differentiating factor for a platform.

Partners - Collaborate for Scalable Capability and Agility

A platform cannot operate without partners. Partners can be product or service complementors, payment providers or app developers. Together with the partners the platform fulfills customer needs and with the additions from external partners the platform owner can quickly scale up the platform.
These 5 p's are the factors that generate network effects and thereby the critical mass that is necessary for a successful platform ecosystem. Next five environment enablers are given, that according to this report are necessary for a platform to flourish; they relate to what kind of environment is conducive to platform success.

5 Environmental Enablers for Platforms



Dimensions	Possible values	Positive effects	Negative effects
Proposition	Update continuously	Superior value for users is created	n/g
Personalization	Mass customization	More effective user targeting with personalized experiences	Data privacy laws can restrict data collection and customization based on that data collection
Price	Freemium	Easy and free access to the platform so users can experience it first	n/g
	Decrease price (discount pricing)	Increase demand, incentive to join	n/g
	Increase price (surge pricing)	Keep supply and demand in check, manage peak demand	Discourage joining the platform
	Pay-as-you-go	n/g	n/g
	Fixed subscription fees	n/g	n/g
Protection	High commitment to cyber security	Differentiating factor for the platform	n/g
Partners	Product or service complementors, Payment providers, App developers	n/g	n/g

Critical Success factors for platforms	
Proposition	Have a compelling value proposition by offering a modular platform and a large userbase connected to it. Continuously update the value proposition
Personalization	Target users with personalized experiences
Price	Price is a strong tool to regulate supply and demand
Protection	Being committed to cyber security of the platform can be differentiating factor
Partners	A platform cannot operate without partners to support in fulfilling the customer needs
Potential user base	Platforms are easier to establish in countries which have a large installed digital base and uniform culture, language and regulations.
Digital talent and entrepreneurship (environment)	As a platform business locate yourself somewhere where you have access to talent from science, technology and engineering and people with entrepreneurial and creative skills.
Technology readiness (environment)	Connectivity, digital assets and investment in new technology influence platform creation and size.
Open innovation culture	Being open to collaboration with partners is increasingly important for innovation.
Adaptive policy and regulation	Due to the increasing speed of change policies should be created proactively and with participation of platform partners.

Reflection on the Strategy

With their five P's Accenture has, like most strategies, invented their own names and definitions for what are basically the same concepts as are mentioned in many of the other strategies. The five concepts are well chosen and the reasoning behind them is well thought out. The strategy source goes a bit deeper than some of the other researched strategies while still being quick to read and easy to understand.

Platform Strategy 5: Molenaar Strategy

Cor Molenaar defines six rules and four key points in his book "The Power of Platform Strategy" (original Dutch title: De kracht van platform strategie) for disruptive two-way platforms [17]. He based his six rules on the book The Platform Revolution from Geoffrey Parker.

The 6 Rules

1. There is a centralized market that coordinates relationships between supply and demand. This market aggregator supplies the tools, governance and central infrastructure to support consumer and producer interactions.

2. There are social or economic exchanges between buyers and suppliers of information, goods or services, these are the platform interactions. Payment can be monetary or social (time and effort). An interaction starts with information exchange, after which an agreement about the exchange of goods, services and/or payment is reached. Finally the exchange takes place on the platform when it concerns digital products or services or outside of the platform like with Uber and Airbnb. It is a great source of value for platforms that they enable direct communication with the users of the platform.

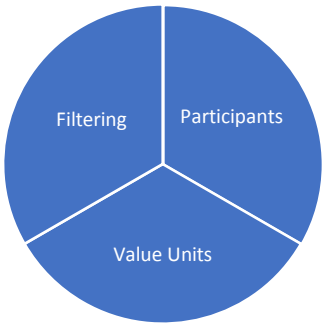


Figure 13: Interaction Components

3. Core interactions of the platform have three important components: the participants, the value unit and the filter. Any interaction on the platform is initiated as communication

about a value unit. A filter allows participants to filter the value units to only those they are interested in. Without filters participants could never find the value unit they are looking for.

4. On a platform value units are offered and can be selected independently. There is no traditional value chain in which a single player can control the production process. Value units are produced by many parties and the platform owner does not own any value units. Producers of value units can easily and quickly reach a large target audience. The platform also offers added value through people reviewing the offered products and services.

5. Network effects entail that the value of the platform is determined by the number of users that are on the platform. There is a positive network effect when an increase on one side of the platform means the platform becomes more valuable for the other side of the platform and the other way around. Competition is about the network size now, no longer about price and quality.

6. Find a balance between openness and quality control for all involved parties. When the platform is open there are no limitations to participation, commercialization and platform usage. Openness will allow the platform to scale up quickly and eliminate the competition. However, a platform should also have some rules and regulation to ensure the quality of the platform.

How Platforms Become Successful

1. Trust. The platform must be engaging and remove friction for both players and customers. There should be incentives to buy or supply. Incentive is usually but not necessarily financial.
People usually do not trust small unknown companies, when buying through a large established platform there is more trust that the product will be good and the payment and other services are properly handled.

2. Visitors through sales support. The vicious cycle of more users leading to more players needs to be stimulating. The platform increases in value and is more interesting when there are more visitors.

3. Areas of interest and domain thinking. Attract users who share a certain area of interest. Offer both general information and product sales.

4. Low prices. The growth rate of a platform is influenced by the price. Advertising can be used to lower the product price for customers, this can lead to more users, which can lead to a higher price to be paid per advertisement.

Requirements for successful platforms

- They remove frictions, the greater the friction that is removed the greater the chances of success are.
- There must be added value for both the supply and demand side
- The platform must be adopted quickly in the market and reaction time of possible competition should be slow
- A partnership network with existing parties should be created

- Break-even time should be short, the shorter the better
- A large impact should be made in a short time

Interaction and communication are important drivers of platforms. The focus should shift from the traditional 4 p's (price, product, place, promotion), to the new 5 c's (customers, connection, communication, collaboration and commitment).

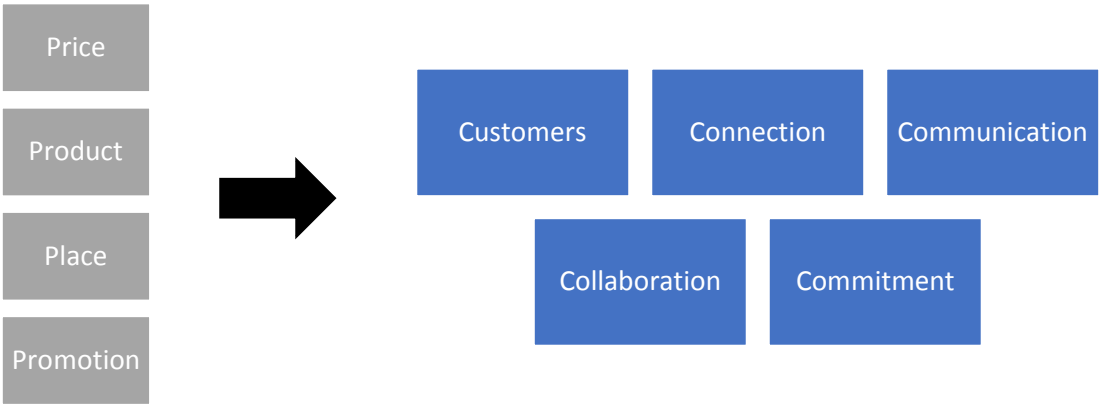


Figure 14: 4P's to 5 C's

In his book he also recommends to fill in the business model canvas to gain insight into the choices of the strategy.

Reflection on the Strategy

The strategy offers many good points, however they tend to be structured in a not entirely logical way. Personally I would not consider the removal of friction as a sub item of trust for example. Removal of friction is about why your platform has added value for users and trust is about convincing users that your platform is reliable and safe to use.

Dimensions	Possible values	Positive effects	Negative effects
Openness	Open	No limits to commercialization, participation and platform usage Scaling up quickly and thereby eliminating the competition	n/g
	Closed	Control over who has access to the platform. Competition and potentially harmful developers can be denied access to the platform. New services can be offered exclusively on the platform. Quality control	n/g
Rules and regulations	Yes	Ensures quality	n/g
Payment	Monetary	n/g	n/g
	Social (investment of time and effort)		
Pricing	Low	Platform growth rate is higher	n/g
What is offered	Information Goods services	Offering both information and goods or services makes the platform more valuable and interesting for users	Takes more effort to offer both high quality information and products

Dimensions	Possible values	Positive effects	Negative effects
Exchange of goods	On the platform (digital goods/services) Outside of the platform (physical goods and services)	n/g	n/g
Reviews	Yes	Added value for users	n/g
Advertising	Yes/no	Yes: lower platform price, more users, so higher price per add, so lower platform price and so on…	n/g

Critical Success factors for platforms	
Centralized market/ connection	Supply and demand are coordinated by a market aggregator
Social and economic exchanges/ enable transactions and interactions	The goal of the platform is to enable these interactions
Filtering	Filters allow platform participants to only find the value items they are interested in
Network effects	The number of users defines the value of the platform. Network size is the main point of competition today instead of price and product quality.
Trust	Users are more likely to trust an established platform than a small unknown supplier
Offer incentives to buy or supply	Usually but not necessarily financial incentives
Remove friction	They remove frictions, the greater the friction that is removed the greater the chances of success are.
Added value	There must be added value for both the supply and demand side
Adoption and competition speed	The platform must be adopted quickly in the market and reaction time of possible competition should be slow
Partnership network	A partnership network with existing parties should be created
Quick impact	A large impact should be made in a short time
Break-even time	Break-even time should be short, the shorter the better
Connection, communication, collaboration and commitment	No explanation is given why these factors are so important.

Platform Strategy 6: Every Organization needs a Digital Platform Strategy

Explore, Plan, Deliver and Evolve your digital platform strategy, this article was published by Gartner [25]. You do not necessarily have to strive for a leadership role in a business ecosystem. Every organization needs a digital platform strategy and the ideal strategy is different for every company. A platform strategy could be to build your own platform or to join an existing platform. The strategy should fit with the long-term business goals.

"The strategy must integrate business and IT needs and establish a collective leadership vision."

Figure 15: Strategy Development according to Gartner

Explore

In this phase the CIO needs to explore the possibilities for a platform and how a platform can be properly integrated with the overall corporate strategy. The primary question in this phase is whether to create your own platform or integrate with an existing platform from another organization and co-create a new product. The platform strategy will influence strategic and operational investments.

Plan

Business and technology planning needs to be integrated in the platform strategy. Match business objectives with the ecosystem or platform strategy. Take into account how likely scenarios are and what the implications, constraints, interdependencies and risks are of different platform strategy scenarios.

Deliver

The support for the platform strategy will be significantly influenced by the business and technology opportunities and risks. A platform has to create value for three stakeholder groups: the platform owner, the customers and the partners. It is important that the platform attracts customers. Both the creation and maintaining of a platform will be challenging. Platform requirements should focus on integration, security, compliance and reputation risk. A platform needs governance to ensure that all users of the platform behave as intended. A team should be created in this phase whose task it is to create momentum for the strategy and who should identify monetization potential and risk impact. This team should define the business goals and decide based on risk and time constraints whether to build the platform in-house or buy it.

Evolve

The architecture of the platform will become increasingly complex as the platform evolves and the business ecosystem expands. The architecture should be designed to support likely scenarios and future plans. Will the platform be supported internally or externally? What will be the roles of the platform core provider, consumer and owner? Focus on technology growth and evolving business architecture to support governance, security, economics and trust.

Reflection on the Strategy

The strategy is mostly a requirement list. Little information is given on different possible platform dimensions and possible values and corresponding results. The strategy reminds me of the “think, do, learn” concept that is integrated at Tersof as a company and in their software.

Dimensions	Possible values	Positive effects	Negative effects
Create or integrate	Build your own platform	n/g	n/g
	integrate with an existing platform		
Platform support	Internal	n/g	n/g
	external		

Critical Success factors for platforms	
Integration	Integrate business and technology planning into the platform strategy. Integration with overall corporate strategy.
Matching	Match business objectives with the ecosystem or platform strategy
Create scenarios	Take into account how likely scenarios are and what the implications, constraints, interdependencies and risks are of different platform strategy scenarios.
Create value	A platform has to create value for three stakeholder groups: platform owner, customers and partners.
Attract users	The most important critical success factor of all, getting users for the platform
Requirements	Requirements should focus on integration, security, compliance and reputation risk
Governance	A platform needs governance to ensure that all users of the platform behave as intended.
Dedicated platform team	A team should be created in this phase whose task it is to create momentum for the strategy and who should identify monetization potential and risk impact. This team should define the business goals and decided based on risk and time constraints whether to build the platform in-house or buy it.

Critical Success factors for platforms	
Future proof architecture	The architecture should be designed to support likely scenarios and future plans. Focus on technology growth and evolving business architecture to support governance, security, economics and trust.

Platform Strategy 7: The Ambidexterity Perspective

This strategy by Javier Cenamor, Geoffrey Parker, Marshall Van Alstyne, and Xing Wan places platform strategy in an ambidexterity perspective [26]. The ambidexterity principle is about keeping your current business activities running smoothly whilst at the same time expanding the business with new business activities, so simultaneously creating new business value and maintaining current business value. Ambidexterity can be achieved by temporal, domain or organizational separation and any activity can be classified as an exploration or an exploitation activity.

The writers of this paper examined 109 different papers about platform strategies, narrowed down from a sample of 1812 papers, making it one of the best researched platform strategy sources I have come across during my research. They concluded that there are 5 main dimensions for platform strategies. Those dimensions are openness, pricing, differentiation, integration and envelopment. For each of these dimension strategic choices can be made whether to separate current and new processes based on domain, organizational or temporal separation. Domain separation is the separation of different efforts within the company based on the different domains in which the company operates. Organizational separation is separating efforts by dividing them over different business units. Temporal separation is creating a separation with time, doing one thing first and at a later point in time doing the next thing.

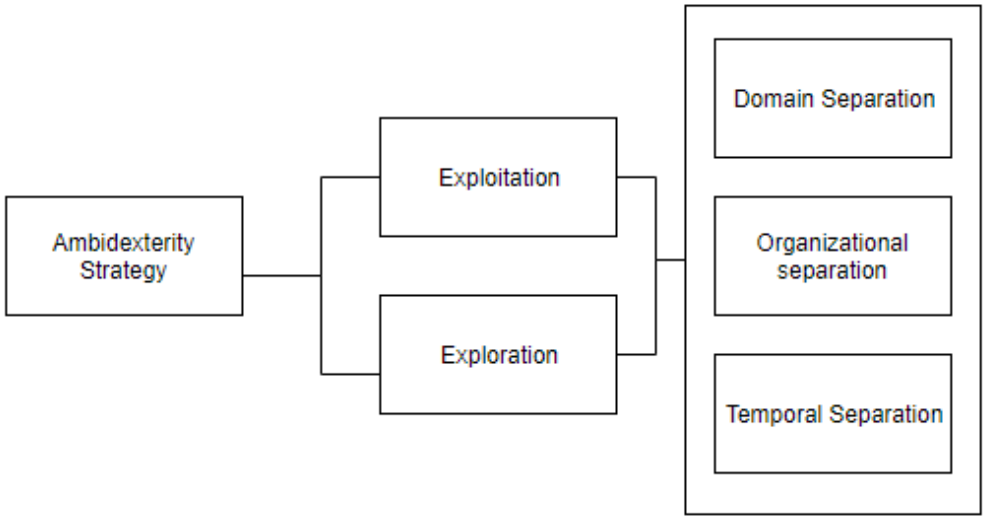


Figure 16: any activity is either exploration or exploitation. Exploration and exploitation activities can be separated by domain, organizational unit or time.

Openness

The strategy dimension openness is about how freely others can join the platform. By varying how open or closed a platform is, both the number of participants and the quality of the platform will be influenced. The openness of a platform is determined by the participation rules of the platform. Platform openness can also be defined as to what degree developers need permission to have access to the platform, build on the platform and offer what they have created to the users of the platform [26]. So an open platform

has soft requirements to access the resources of the platform owner. A closed platform will lead to a limited number of third-party developers because there is a strict list of requirements that have to be met to be allowed on the platform. A closed strategy can help to ensure the quality of the ecosystem, as low-quality complements can be rejected, and it will also diminish the risk of crowding effects that will lead to negative experiences.

Participation on the platform can be facilitated or limited by inter-platform compatibility decisions (multihoming). Compatibility and multihoming however do also entail that the platform owner will have to share his customer base. The researchers of this paper mention that there is empirical evidence that the rate of innovation around a platform is increased by opening it up to third parties. Innovations by third parties can be used to improve the platform. New innovations can be explored and innovation by third parties can be exploited to benefit the entire ecosystem. It is strategically wise to allow compatibility and multihoming on the side of the platform with small network effects and opt for a more closed strategy on the platform side with larger network effects.

Ambidexterity in Openness Strategy

Ambidexterity in openness can be achieved through organizational and domain separation. Domain separation is often used to keep the core part of the platform closed to protect exploitation and to be simultaneously open in other domains such as complementary components to explore new resources and capabilities. Organizational separation can be used by creating a separate business unit that is responsible for extending the size and power of the entire market which in turn will lead more third-party capabilities which can be explored. Intel for example has such a unit that is committed to growing the entire market and not just the Intel market share.

Pricing

Pricing is a dimension of great importance and complexity. It is made complex by interdependencies among the participants on the platform and the lifecycle of the platform ecosystem. Pricing has a strong influence on the size of the customer base. The right pricing strategy can lead to a virtuous cycle of an ever-increasing number of customers. Subsidizing and taking advantage of asymmetric cross-network effects can be used to increase the customer base. The price is kept artificially low, which would not be possible without network effects. Subsidizing is not necessarily done only by offering an artificially low price: free information, privileges access or technical support can be offered to entice users to join the platform. Choosing the right side to subsidize is critical to platform success. Make this choice by looking at own-price and cross-price elasticity of demand, quality sensitivity, multihoming and substitutable or complementary relationships between applications. Existing customers should be taken into account when designing platform pricing. Prices can be fixed or variable and a skimming or penetration strategy can be chosen. Penetration strategy entails charging a low price and skimming is charging a high price. Pricing can be driven by the competition in the market. If you have a monopoly you have more freedom in pricing choices compared to a highly competitive market. Pricing strategy effectiveness is determined by price sensitivity of customers over time. When price sensitive decreases over time a penetration strategy can be a good choice to start with to then later increase the prices.

Ambidexterity in Pricing Strategy

Ambidexterity can be used in pricing strategy by domain and temporal separation. Domain separation can entail exploring one market and simultaneously exploiting another. Domain separation can also take place on the user versus complementor sides of the platform, usually one side is more valued by the

other. The side that is dependent on the other side can be exploited while the more valuable side is explored. Temporal separation can be achieved by changing the prices of the platform throughout the different stages of the platform lifecycle. To achieve critical mass a platform can first exploit its existing user base and then later exploit new ways to gain value from that user base.

Differentiation

Differentiation strategy is the alternative to pricing strategy. With differentiation the goal is to provide a unique ecosystem that users will adopt, whereas pricing focusses on getting the users in a market with many alternatives for your platform. So there is a choice for a platform owner to attract users with the best functionality or to attract users by offering the lowest price. Differentiating on the quality of your offering can help to overcome latecomer disadvantages and challenges the dominant position of an incumbent. A platform can distinguish itself on the platform level or on the ecosystem level. By enhancing the value that is offered to users and complementors and by investing to improve the intrinsic characteristics of the platform a higher quality product can be offered. Differentiating on the ecosystem level can be done by attracting marquee users and having exclusive contracts with complementors and high-quality suppliers. Differentiation on the ecosystem level can also be achieved when the market is so heterogeneous that different versions of the same platform can be launched to meet the needs of the different customers.

Ambidexterity in Differentiation

Ambidexterity of differentiation can be achieved through domain separation. A company can differentiate itself in the platform domain or in the extended ecosystem domain. New market domains can be explored by offering trial versions to potential adopters, existing customers can be exploited by offering premium versions at higher prices later on.

Integration

Vertical integration, i.e. entering the space of a complementor, is not the same business model as being a multi-sided platform. Moving into the space of complementors will demotivate them to contribute to your platform as you are taking revenue away from them. When a platform has not yet established its user base, complementary products or services are usually offered by the platform. Once enough users have adopted the platform it will become interesting for third party complementors to create products or services for the platform. Once the platform has many third party complementors revenue can be generated by imitating successful complementary products. Amazon for example does this by offering often sold products themselves, following the third parties who sold those products first. When there are enough external complementors quality can be improved by removing poor-performing complementary parties from the platform.

Ambidexterity in Integration

Domain separation is simultaneously exploiting the platform resources you have, while exploring complementary markets by moving into them. A platform owner has access to customer data and knowledge and is therefore in a better position to enter a complementary market compared to a third-party supplier. Exploiting platform resources can help to explore new complementary markets.
Organizational separation can also be used by using a dual structure to realize both exploration and exploitation concurrently. A platform owner should establish a unit committed to serving the whole

ecosystem so new platform capabilities for the entire ecosystem can be developed. Develop a separate unit for each new market that is being explored.

Envelopment

Use envelopment strategy to achieve growth: a platform can move into another platform market by bundling their functionality with the functionality that is desired in the target market Competitive advantage is gained by operating in multiple platform-based markets simultaneously. This way the platform can cross traditional industry boundaries and increase the scope of the platform. By using bundling advantage can be taken from an established platform but this can also be done in other ways. A new platform can utilize resources of an existing platform with the right interfaces and thereby create synergy between both platforms.
A company can piggyback on an existing platform. A new product or service is offered to an existing user base. Existing platform features and users can also be used to create a new successful platform. Piggybacking: exploit the existing customer base of an established platform first and then later explore the opportunities for becoming independent when a reputation has been established.
Airbnb for example started by piggybacking on Craigslist. PayPal started by piggybacking on eBay.

Ambidexterity in Envelopment

Ambidexterity can be achieved through temporal and domain separation.
Domain separation entails that there are two domains: existing and new markets. Existing market: here the platform owner has an established position, the platform owner already has resources and capabilities such as a customer base, components, knowledge and governance capabilities. When moving into the adjacent market the platform owner must search, learn and develop new resources and capabilities.
Exploitation of resources in the established market can be combined with exploration of new resources in the new market. How to combine those two depends on the relationship between the established and new market. They can either be complements, weak substitutes or functionally unrelated. The benefit of linking the established and the new market lies in scope economies and price discrimination benefits. Temporal separation: first the platform must be established in one market to be able to later move into another market. A platform owner must first explore the market and then exploit it by envelopment.

Ambidexterity Matrix	Pricing	Openness	Differen-tiation	Integration	Envelop-ment
Domain separation	x	x	x	x	x
Temporal separation	x				x
Organizational separation		x		x	

Reflection on the Strategy

They claim there are 5 dimensions, but I think they actually use more. Sponsoring is considered a sub item of pricing, but I think it can be viewed as a separate dimension. Multi-homing is considered a sub item of openness, however, I would also consider this a separate dimension as an open platform may not necessarily allow multi-homing and a closed platform may allow it. The paper tried to simplify platform strategy by combining and grouping elements until there were only 5 left, and although there is an argument to be made for simplicity, oversimplification may degrade the value of the strategy.

Dimensions	Possible values	Positive effects	Negative effects
Openness (how strict the participation rules are, to what degree need developer permission to access the platform) **A range from completely open to completely closed**	Open (soft requirements)	More participants, increased rate of innovation, new market opportunities. Third party innovations can be used to improve the platform or the entire ecosystem	Less quality control, crowding (too many offerings) effects
	Closed (Strong requirements)	Protects exploitation of internal resources Higher quality (reject low quality complements) No crowding effects	There will be only a limited number of partners
Fixed or variable pricing	Fixed	n/g	n/g
	variable	n/g	n/g
Skimming or penetration pricing	Penetration strategy entails charging a low price and	penetration: low price: incentive to join the platform, low price to attract customers (large customer base with low price)	Penetration: prices are raised over time, customers might not be willing to pay the new higher price
	skimming is charging a high price.	n/g	n/g
Sponsoring	Sponsoring yes/no	Attract users to the platform	n/g
Type of sponsoring	Sponsoring by offering: low price, free information, privileged access or technical support.	Attract users to the platform	n/g
Differentiation based on price or quality	Price	Attract users with a low price	n/g
	Quality	overcome latecomer disadvantages and challenges the dominant position of an incumbent.	n/g
Differentiation scope	Platform	n/g	n/g
	Ecosystem	n/g	n/g
Piggy backing	Yes	The customer base and features of an existing platform can be used to create a new platform	n/g
Envelopment	Yes	Achieve growth. Gain competitive advantage. Increase platform scope by crossing traditional industry boundaries. Take advantage of an existing platform by bundling. Create synergy between two platforms. The benefit of the linking the established and new market lies in scope economies, price discrimination benefit and attractive tying prices.	n/g
	No	n/g	
Integration	Vertical integration	When there are enough external complementors quality can be improved by removing poor-performing complementary parties from the platform.	Moving into the space of complementors will demotivate them to contribute to your platform as you are taking revenue away from them.
Multihoming	yes	Facilitate inter-platform compatibility	You have to share your customer base
	No	Protects your customer base	High cost of switching for potential customers

Dimensions	Possible values	Positive effects	Negative effects
User attraction strategy	Pricing	Attract users with a low price	Costly to keep the price low
	Differentiation	Attract users with superior product quality	costly to develop superior functionality

Critical Success factors for platforms	
Extract revenue from the existing customer base	If there is an existing customer base the pricing strategy should be designed in such a way that revenue can also be extracted from this existing customer base
Achieve critical mass	To create a sustainable platform critical mass has to be achieved early on both platform sides
Be fair	A platform owner must a have credible commitment to remain fair to participants once they have entered the platform
Ambidexterity	keeping your current business activities running smoothly whilst at the same time expanding the business with new business activities, so simultaneously creating new business value and maintaining current business value
Be open and closed	Keep the core closed and open complementary components for exploration and new development
Grow the entire market	Organizational separation can be used by creating a separate business unit that is responsible for extending the size and power of the entire market which in turn will lead more third-party capabilities which can be explored. Intel for example has such a unit that is committed to growing the entire market and not just the Intel market share. Develop a separate unit for each new market that is being explored.
Pricing strategy to create a virtuous cycle	Pricing has a strong influence on the size of the customer base. The right pricing strategy can lead to a virtuous cycle of an ever-increasing number of customers.
Harness network effects to increase the customer base	Subsidizing and taking advantage of asymmetric cross-network effects can be used to increase the customer base. The price is kept artificially low, which would not be possible without network effects.
Use non-financial subsidizing	Subsidizing is not necessarily done only by offering an artificially low price, free information, privileges access or technical support can be offered to entice users to join the platform.
Take price sensitivity into account	Pricing strategy effectiveness is determined by price sensitivity of customers over time. When price sensitive decreases over time a penetration strategy can be a good choice to start with to then later increase the prices.
Let price be driven by the level of competition in the market	Pricing can be driven by the competition in the market. If you have a monopoly you have more freedom in pricing choices compared to a highly competitive market.
Prepare for moves into adjacent markets	When moving into the adjacent market the platform owner must search, learn and develop new resources and capabilities.
Choose the right side to subsidize	Choosing the right side to subsidize is critical to platform success. Choosing the right side to subsidize is critical to platform success. Make this choice by looking at own-price and cross-price elasticity of demand, quality sensitivity, multihoming and substitutable or complementary relationships between applications
Allow compatibility and multihoming on the proper side	It is strategically wise to allow compatibility and multihoming on the side of the platform with small network effects and opt for a more closed strategy on the platform side with larger network effects.

Platform Strategy 8: The Nine Guiding Principles

This book by Amrit Tiwana is very extensive. He has done a lot of research into platform strategies [16]. He presents an overview of many possible choices: he does not a recommend a single strategy but presents nine guiding principles for platforms. The book is about 300 pages long, so a selection has been made of the most relevant information in regard of the type of platform that could be constructed at Tersof.

According to this book to be considered a platform the platform must at least be two-sided. He considers a single-sided platform to not be a platform. A platform cannot succeed on its own, it is dependent on partners from the ecosystem. There are two gears in the evolutionary motor of a platform: architecture and governance. These gears must interlock and be aligned and coevolve. Platform evolution is driven by the race between rival ecosystems. Developing creative orchestration, which is reflected in your platform architecture, will set you ahead of your rivals. Because the platform is dependent on partners for the integration of external innovation the focus of the platform owner should be on orchestration instead of

on management. The value of a platform is determined by the number of users. The architecture refers to architecture on platform level and architecture on app level. Governance in this book is considered to consist of the partitioning of decision rights, pricing and the control portfolio configuration.

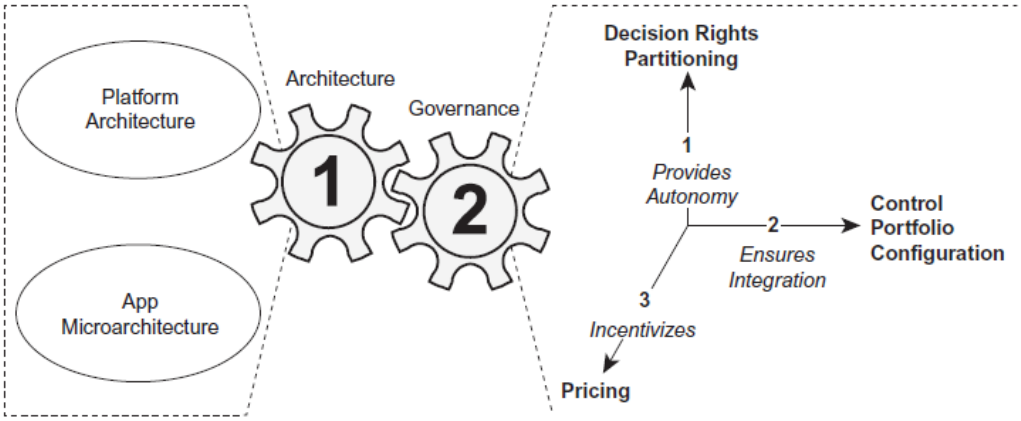


Figure 17: The two gears, architecture and governance which should be aligned and interlocked [16].

Ecosystem Architecture - Modularization

A platform ecosystem is composed of many interacting subsystems. The interactions and division of functionality are guided by the architecture of the platform. The architecture should divide the platform into a highly reusable stable core and complementary apps which can vary. This modularization makes the platform extensible. Architecture exists on two levels, platform level and app level (microarchitecture). The platform level architecture contains the platform core and the interfaces. The architecture should communicate to apps what the platform does and how to use it. Platform architecture should be simple, resilient, maintainable and evolvable. Platform interfaces should be precise, frozen and versatile. If they are pleasant and easy to use app developers will stick to them. There are two extremes in platform architecture, modular architecture and monolithic architecture and all platforms are somewhere in between these two extremes. Modular architecture is the most suitable choice for future evolvability while monolithic architecture will create immediate performance. Modular platform architecture has both advantages and downsides for external complementors, these are summarized in the next tables.

Benefits and downsides of modularization	
Benefits for the platform owner	Downsides for the platform owner
Distributed innovation at a large scale	Costly to implement full modularity
Larger variety of apps that is offered on the platform	Decreases technical performance
More incremental innovations	Limits future options for architectural innovation
Platform is controlled by the structure of the architecture and not by ownership	Higher risk of being imitated by rivals

Benefits and downsides of modularization for app developers	
Benefits for app developers	Downsides for app developers
No need to reinvent the wheel so there is room for specialization	Additional costs due to the modular structure
No need to know everything about the platform	Decreases app performance
Easier to evolve apps	Limits the options for experimentation
Option to multi-home on different platforms	High risk of getting locked in the platform

The Platform Core

Another aspect of platform architecture is what to include in the platform core and what to keep out side of the core. In the book 5 magic rules for what should be included in the platform core are given:

1. Functionality with highly reusable functionality. Things that will be shared by many apps. They can then be improved more rapidly as only the interfaces need to stay the same.
2. Generic functionality. Generic functionality will be useful for many app implementations.
3. Interfaces. These should never change to guarantee compatibility. New interfaces can be added and any processing that takes place behind the interface can be updated as long as the interface remains unchanged.
4. Stable functionality. Functionality that is subject to change should be kept out.
5. Functionality that is relatively certain. Keep uncertain functionality out.

Platform Management versus Traditional Business Management

Blackberry used to have about 50% of the smartphone market, but by 2012 they had difficulty getting a 1% market share. When Apple and Android entered the market Blackberry responded by lowering their prices, investing in innovation and increasing marketing efforts. The product was not the problem, but the fact that Apple and Android had entire ecosystems behind them. Blackberry had 8000 external innovators, apple 200.000. Blackberry did not fail because their product was inferior, but because their ecosystem was inferior. This example illustrates that platform management requires a different mindset compared to traditional business management, not only your product matters but your entire ecosystem, especially in a highly competitive market.

There are several differences between platform and traditional product or service business models. Because of these differences a platform requires orchestration instead of management and a careful balancing of autonomy versus control. Platforms mostly differ from traditional product and service businesses in terms of market potential, structure and management. Platforms are multisided and therefore have a much larger market potential. There is an opportunity for far greater economies of scale compared to traditional businesses. Both mass markets and their long tails can be tapped into through customization by end users. Customers are (non-coercively) locked-in and the service-based business model creates a continuous revenue stream. Lock-in combined with network effects can allow a platform to charge high prices even in a highly competitive market.

A platform is structurally different because the ecosystems of which they are the center are diverse, large and fluid. A platform is more difficult to control as it is only partially under control of the owner. Ownership and intellectual property are highly fragmented. Innovation is driven by outside partners and they also bear the costs and risks of those innovations. Because there is little control over the partners the management style on a platform should also be different. The diversity among complementors and users is both the biggest strength and the biggest challenge of platforms. The platform owner must

balance individual needs with overall integration of all complements. A platform owner must control things he has no ownership over. Therefore the term “orchestration” is introduced instead of “management”. Another big difference is how innovation occurs, because there is no authority, command or control over external developers, innovation is always emergent. For the platform owner there lies a challenge in respecting the autonomy of external partners, while at the same time making sure that the seamless integration of all elements of the platform is not compromised. Conventional control mechanisms will not work, the platform owner should construct a control portfolio. Architecture should be the main coordination and control mechanism. Competition is no longer between products but between entire ecosystems and therefore the complementors in your ecosystem will determine your success and the faster you evolve the more likely you are to survive, so the platform must be designed for evolvability.

The second gear of the platform evolutionary motor according to this book is governance. Governance should be aligned with the platform architecture and it should always be cheap and simple. The book categorizes three sub dimensions of governance: division of decision rights (autonomy), the control portfolio (integration), and the pricing policy (incentives); these dimensions must be orchestrated, rather than managed, by the platform owner because they involve partners over whom the platform owner has no control.

Decision Rights Partitioning

The first dimension of platform governance is decision rights partitioning. Decisions can be made by the platform owner or by app developers. When the app developers have many decision rights the governance is decentralized. When the owner has all decision rights the platform is completely centralized. Decision rights are a spectrum dimension from centralized to decentralized. Decision rights refer to decisions about the platform or about apps. About strategic things or implementation (what to accomplish and how to accomplish it).

Control Portfolio Design

The platform control portfolio is part of the platform governance. The goal of control is to align the work of the app developers with the best interests of the platform. Such control can be implemented through the use of control mechanisms which can be either formal or informal. The combination of all control mechanisms is the control portfolio. The formal control mechanisms are gatekeeping, metrics and process control. The informal control mechanism is relational. Control mechanisms should always be legitimate, fair and reasonable. The overall control portfolio that is constructed should be simple, transparent, realistic, reflect shared values, and be fair. The control portfolio should be transparent for app developers. Ambiguity makes compliance difficult, it is necessary to be explicit about expectations and how they will be measured and to make the evaluation process clear to developers. Control should be realistic guidelines should be created and not a rule book. Shared values means that controls must be based on a shared philosophy of all ecosystem participants. Controls should reinforce and not contradict the philosophy. It should be fair: control should be consistently applied to all app developers and there should be no contradictions. Controls are costly to implement as every control should be designed, implemented and enforced. So the benefit of any control mechanism should outweigh the costs. There will also be compliance costs for complementors as it will require time, effort and financial burdens from them to comply with the control mechanisms. So if these compliance costs are too high, the frequent release of apps will be limited. The platform should only have those controls which are absolutely

necessary to keep the costs low and to make sure that controls can be realized in practice and can be enforced.

Control Mechanism	Definition	Prerequisites
Gatekeeping	The degree to which the platform owner uses predefined criteria for what apps are allowed into the platform's ecosystem	• Platform owner must be competent to judge • Platform owner must be fair and speedy • App developers must be willing to accept such gatekeeping
Process	The degree to which a platform owner rewards or penalizes app developers based on the degree to which they follow prescribed development methods and procedures that it believes will lead to desirable outcomes	• Platform owner must have the knowledge to mandate methods to app developers • Platform owner should be able to monitor app developers' behaviors or verify compliance
Metrics	The degree to which the platform owner rewards or penalizes app developers based on the degree to which the outcomes of their work achieve performance targets predefined by the platform owner	• Metrics must be set by the platform owner, predefined, and objectively measurable
Relational	The degree to which the platform owner relies on norms and values that it shares with app developers to shape their behaviors	• Existence of shared norms and values between app developers and platform owner • Low app developer churn

Pricing

The third dimension of governance is the pricing policy. The pricing policy should motivate developers to develop for the platform. Pricing must therefore be aligned with the business model, the stage in the platform lifecycle and the platform architecture.

In the book there are five pricing choices described:

1. Whether pricing should be symmetric or asymmetric for the two sides of the platform.
2. If asymmetric pricing is chosen, which platform side to subsidize and for how long?
3. Should pricing be based on access versus usage?
4. How app revenues are divided, pie-splitting using a fixed scale or a sliding scale?
5. How are the apps licensed?

The first and second pricing decisions refer to the symmetry of pricing. Asymmetric pricing is making money from one side of the platform and subsidizing the other. Symmetric pricing is making money from both sides of the platform. The profit from the priced side should outweigh the subsidizing costs of the other side. Pricing decision should be made for the long-term profitability. Users are likely to leave the platform when subsidizing is abruptly ended. Subsidizing can be used to quickly attract a lot of users but the retention of those when the subsidizing ends is questionable.

The third decision is about how to charge your users: access fees are fees that have to be paid to gain access to the platform, whereas usage fees are fees that only have to be paid for actual usage of the platform. Usually at least one of these is zero depending on the stage in the platform lifecycle, pre or post dominant design phase, accumulation of critical mass of both the user and developer side.

Pie splitting on a fixed scale or a moving scale refers to how revenue from app sales will be shared among the developer and platform owner. Fixed scale means that the owner keeps a predetermined percentage of sales. Moving scale means that the percentage changes when more apps are sold (goes up or down). Fixed scale is most commonly used. The choice influences the motivation of app developers. Pie splitting can be changed depending on the phase in the platform lifecycle.

App licensing decisions, 3 structures:
1. Single perpetual license: one-time payment for the right to use the app indefinitely. License is per Individual, per machine, or floating (a limited number of licenses is shared among a larger number of users over time).
2. Subscription-based license: a specific user has access to the app for a predetermined amount of time. If the user wants to continue using the app after this time he has to pay a fee again.
3. Usage-based license: the user pays dependent on how much the app is used. Usage must be measured, for example in minutes a day. This is also called the utility-model. The difficulty of this model is, that there must be a way to precisely and cost-effectively measure usage.

Pricing Decision	Business Model	Lifecycle	Architecture
Pricing symmetry?	Asymmetric if two-sided from outset and dependent on cross-side network effects	Asymmetric if first mover advantage can be secured through network effects or switching costs	–
Subsidized-side?	• If one side values the other more • "Superstar" apps	• Dominant design emerged? • Diffusion among end-users?	–
Access fees?	• Generally avoid • Token access fee from app developers to signal credible commitment	Negative or zero access in early but nonzero in later lifecycle stages	Usage fees for native services-intensive app microarchitectures
Moving pie-splitting scale?	Rising scale if app developers multihome rival platforms	Rising scale if intense cross-platform rivalry	Sliding scale if low platform scalability
App pricing model?	–	–	Viable app licensing models are constrained by platform architecture

The Nine Guiding Principles

The book presents nine guiding principles in platform markets. These principles describe rules as well as problems related to platforms. The first four principles refer to the initial development of platform ecosystems.

1. The Red Queen Effect

The Red Queen effect refers to a concept in Lewis Carroll's "Alice in Wonderland": if you and someone else both run very fast, but at the same speed, in a way you stay in the same place. The pressure to adapt faster just to survive is ever increasing and driven by the pace of technological innovation. A platform must evolve at the same speed as other platforms are evolving, if they wish to survive in the market. In trying to get ahead of a rival platform the bar for all platforms is raised and your competition will rise to the new required level and even higher, leaving the platform that was trying to get ahead behind.

2. The Chicken-or-Egg Problem

If a platform has two sides than either side will only want to join after the other side has joined the platform. So it is a problem of who comes first. Developers do not want to develop for a platform with

very few users and users do not want to join a platform with very few complementary offerings. Not addressing this issue will guarantee that the platform fails.
Starting out as a standalone product, service or one-sided platform has been the road to success for many platforms, because this is an excellent way of overcoming the chicken-or-egg problem. There is already value offered to one side of the platform long before the other side is added to the platform.

3. The Penguin Problem

Users have a tendency not to adopt a platform when they are unsure whether others will do so as well. Users wait for the network effect to occur before they want to join. This greatly stalls the achievement of critical mass. The connection of this behavior to penguins lies in the fact that penguins wait to dive into water when they are hungry because they are afraid there might be predators. There is doubt which is stronger, the desire for food or the risk of getting eaten. There is an adoption inertia barrier which must be overcome. Subsidizing a particular platform side can help to overcome this problem.

4. Emergence

Partners in the ecosystem pursue their own interests based on their expertise and on feedback from what others in the ecosystem are doing. This will lead to innovations that arise spontaneously and not in way that was organized by the platform owner. The platform owner cannot control or plan emergence, but he must enable and facilitate it. Innovation is usually a result of the personal desires of complementors.

The next five principles refer to how to orchestrate an ecosystem to evolve.

5. The Seesaw Problem

The biggest strength as well as the biggest challenge of platforms is the diversity. App developers should be given autonomy to encourage them to create innovations around the platform, but their apps should still seamlessly integrate with the platform. This is a delicate balance that must be managed by the platform owner. Aligning governance and architecture at the app level will help to manage the seesaw problem.

6. The Humpty Dumpty Problem

When an app is embedded in a platform extracting it from the platform, updating it and then putting it back on the platform again can be difficult. With a large amount of subsystems and interdependencies complexity increases. A small change in the platform can break many apps that were previously functioning. Software architecture should manage this problem and prevent it from happening. Modularization is a Lego brick idea: as long as the connectors work these subsystems can be any size, shape, amount or color and put together in any way and they will fit. The platform and components can be separated and put together again. Apps can be updated, revised or refined without the need for coordination and interaction with the platform owner or other app developers. The app can simply be put back on the platform and then it will again seamlessly integrate with all other platform elements. This modular structure will also allow the platform owner to make changes to the platform without having to coordinate or interact with the app developers as long as the interfaces remain unchanged.

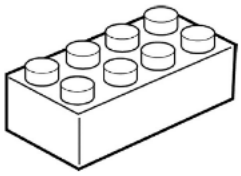


Figure 18: Lego

7. The Mirroring Principle

The organizational structure of a platform ecosystem should be a mirror image of the platform architecture down to the app level and then they will be mutually reinforcing. Apps which only have a loose connection to the platform can suffice with loose governance. Apps with a strong integration into the platform need to be governed more strictly, they will have a greater influence on the platform and therefore present a bigger risk of negatively influencing the platform.

8. Coevolution

Coevolution is the concept of evolving governance and architecture together. Architecture and governance should be designed together and evolve together. If they are designed together and evolve together they will mutually reinforce each other.

9. The Goldilocks Rule

When given three options people have a tendency to choose the middle one. This option is perceived to be 'just right' just like the middle size bed was in the Goldilocks fairytale. By offering a too big and a too small choice people are nudged towards the middle choice. This middle option is generally engineered to be the most profitable and is by far the most sold option. Offering more than three options can cause confusion and choice overload.

Reflection on the Strategy

This platform strategy book is by far the best and most extensive source I have come across. Although it only considers two and multi-sided platforms to be platforms it is still the best book to read if you would only want to read a single book. Because it is such an extensive book it took quite a long time to read and process and summarize the most important part of the book.

Dimensions	Possible values	Positive effects	Negative effects
Number of platform sides	Single-sided platform	n/g	n/g
	Multi-sided platform		
Platform evolution driven by	Architecture	n/g	n/g
	Governance		
Governance dimensions	Partitioning of decision rights	Provides autonomy	n/g
	Pricing	incentivizes	
	Control portfolio	Ensures integration	
Architecture **(for the platform owner)** **(range from completely modular to completely monolithic)**	Modular	Distributed innovation Increase variety of apps. More incremental innovations. Control through architecture. Increases future evolvability.	Modularity is not free. Negative influence on technical performance. Foreclosures architectural innovation. Increases risk of imitation by competitors.
	Monolithic	Creates immediate performance	n/g
Architecture **(for the app developer)** **(range from completely modular to completely monolithic)**	Modular	More specialization, no reinventing the wheel Valuable ignorance Better evolvability for apps Multihoming on rival platforms	Additional costs imposed by the modularity Negative influence on app performances Limits experimentation Risk of being locked-in the platform
	Monolithic	n/g	n/g
Platform core	Elements that are included	n/g	n/g

Dimensions	Possible values	Positive effects	Negative effects
	Elements that are excluded		
Autonomy/control	Range from complete autonomy to complete control	A balance is optimal.	With too much autonomy there is no control but too much control will scare developers away
Source of innovation	Outside partners	The outside partner bears the costs and risks of the innovation	For the platform owner there lies a challenge in respecting the autonomy of external partners while at the same time making sure that the seamless integration of all elements of the platform is not compromised.
Control Mechanisms	Gatekeeping (formal mechanism)	Ensures platform quality. Cost effective compliance measurement method.	App developers might not accept it. High requirements on judgement of platform owner.
	Process (formal mechanism)	Ensures platform quality Process compliance is verified	Need for monitoring and penalizing
	Metrics (formal mechanism)	Ensures platform quality Fairness in judgement (objective measurement)	Costly to implement
	Relational (informal mechanism)	Ensures platform quality Low developer churn Cheaper than formal controls Fills gaps in formal controls	Only works if norms are shared
Pricing symmetry	Asymmetric pricing is subsidizing one side of the platform and making money from the other	Subsidizing can be used to quickly attract a lot of users	Users are likely to leave the platform when subsidizing is abruptly ended.
	Symmetric pricing is making money from both platform sides	n/g	n/g
Pricing based on access versus usage	Access fees are fees that have to be paid to gain access to the platform.	n/g	n/g
	Usage fees are fees that only have to be paid for actual usage of the platform.	Lower risk for the user to join the platform	
Division of revenue with pie-splitting	Pie-splitting on a fixed scale: the owner keeps a predetermined percentage of sales.	n/g	n/g
	Moving scale: the percentage changes when more apps are sold (goes up or down).	More profit can be made from big selling applications	
App licensing structures	single perpetual license: one-time payment for the right to use the app indefinitely. Individual license, machine license, floating license.	n/g	High risk for the user
	Subscription-based license: a specific user has access to the app for a predetermined amount of time. If the user wants to continue using the app after this time he has to pay a fee again.	Low risk for the user. Continuous revenue stream for the platform owner.	n/g

Dimensions	Possible values	Positive effects	Negative effects
	usage-based license: the user pays dependent on how much the app is used.	n/g	The difficulty of this model is that there must be a way to precisely and cost-effectively measure usage
Governance: decisions rights partitioning	Decisions made by platform owner (centralized governance)	Great control over the platform	Demotivates developers
	Decisions made by app developer (decentralized governance)	Autonomy will motive developers	Risk for seamless integration of all platform elements.

Critical Success factors for platforms	
Partners	A platform is dependent on partners in its ecosystem to succeed
Multi-sided	To be considered a platform there should be at least two different sides
Architecture and governance alignment	Architecture and governance drive platform evolution, they should be aligned, interlock and evolve together.
Coevolution / Development of orchestration	Coevolution is the concept of evolving governance and architecture together. Architecture and governance should be designed together and evolve together. If they are designed together and evolve together they will mutually reinforce each other.
Modularization in architecture	The architecture should divide the platform into a highly reusable stable core and complementary apps which can vary. This modularization makes the platform extensible.
Architecture requirements	simple, resilient, maintainable, and evolvable
Interface requirements	Platform interfaces should be precise, frozen and versatile.
Develop creative orchestration	Developing creative orchestration which is reflected in your platform architecture will set you ahead of your rivals. Focus on orchestration rather than management. Because there is little control over the partners the management style on a platform should also be different.
Be cautious in subsidizing	The profit from the priced side should outweigh the subsidizing costs of the other side. Pricing decision should be made for the long-term profitability. Users are likely to leave the platform when subsidizing is abruptly ended. Subsidizing can be used to quickly attract a lot of users but the retention of those when the subsidizing ends is questionable.
Red Queen / development rate	Develop at least as fast as your competitors do.
Growth of number of users and apps chicken-or-egg	Address the chicken-or-egg problem by starting as a standalone product and once one sides has joined open up the platform to the other side.
Getting the first users (penguin problem) Subsidize	It is Important to attract the first users, who will test and inspire other users to join the platform. There is an adoption inertia barrier which must be overcome. Subsidizing a particular platform side can help to overcome this problem.
Innovation is emergent	The platform owner cannot control or plan emergence, but he must enable and facilitate it. Innovation is usually a result of the personal desires of complementors.
Balance / seesaw	Manage the delicate balance between autonomy and seamless integration. Align this with governance and architecture.
Modular architecture	The platform and components can be separated and put together again. Apps can be updated, revised or refined without the need for coordination and interaction with the platform owner or other app developers. The app can simply be put back on the platform and then it will again seamlessly integrate with all other platform elements. This modular structure will also allow the platform owner to make changes to the platform without having to coordinate or interact with the app developers as long as the interfaces remain unchanged. Modularization is a Lego brick idea: as long as the connectors work they subsystems can be any size, shape, amount or color and put together in any way and they will fit.
The Mirroring Principle	The organizational structure of a platform ecosystem should be a mirror image of the platform architecture down to the app level and then they will be mutually reinforcing.
Nudge towards a 'just right' option	If there are too many choices, users get confused; if there are too little choices they go to the competitor. Engineer a 'just right' option and nudge people towards it.
Be aware of the differences in management compared to traditional business	Not the product is the main competitive item but the ecosystem around it. Because of the different orchestration instead of managerial control A platform owner must control things he has no ownership over. Balancing of autonomy versus control.
Lock-in and network effects	Lock-in combined with network effects can allow a platform to charge high prices even in a highly competitive market.

Critical Success factors for platforms	
Tap into mass markets and long tails	Both mass markets and their long tails can be tapped into through customization by end users.
Align pricing with business model and life cycle stage	The pricing policy should motivate developers to develop for the platform. Pricing must therefore be aligned with the business model, the stage in the platform lifecycle and the platform architecture
Governance	Governance should be aligned with the platform architecture and it should always be cheap and simple. Orchestrate governance.
Platform core	what should be included in the platform core: 1. Functionality with highly reusable functionality. Things that will be shared by many apps. They can then be improved more rapidly as only the interfaces need to stay the same. 2. Generic functionality. Generic functionality will be useful for many app implementations. 3. Interfaces, these should never change to guarantee compatibility. New interfaces can be added and any processing that takes place behind the interface can be updated as long as the interface remains unchanged. 4. Stable functionality (functionality that is subject to change should be kept out) 5. Functionality that is relatively certain (keep uncertain functionality out)
Control mechanisms and portfolio	Control mechanisms should always be legitimate, fair and reasonable. The overall control portfolio that is constructed should be simple, transparent, realistic, reflect shared values and be fair.
Control mechanism: gatekeeping	Requires competent judgement, fairness and speed from the platform owner. App developers must be willing to accept the judgement.
Control mechanism: process	Platform owner must be knowledgeable about the prescribed methods and monitor developer compliance.
Control mechanism: metrics	Metrics must be predefined and measurable
Control mechanism: relational	There must be shared norms between the platform owner and the app developers.

Platform Strategy 9: 8 Ways to Launch a Successful Digital Platform

This strategy, or actually these 8 strategies, are all about getting users to join the platform shortly after the launch of the platform. The difficulty in launching a platform is getting a user base on both sides when users are only interested in joining if there already is another side of users. This is often referred to as a chicken-or-egg dilemma. These strategies are based on the book “Platform Revolution: How Networked Markets Are Transforming the Economy – And How to Make Them Work for You” by Geoffrey Parker, Marshall W. Van Alstyne and Sangeet Paul Choudary [27].

Main Points

- Bring together producers and users in efficient exchanges of value
- Leverage network effects: the more participants, the greater the value of the platform
- Pull and not push strategy: not advertising but a platform with incentives for participation to attract users; Push strategies can be useful, but they are much less effective because there is so much being pushed at people these days, the message is likely to get lost.

All these strategies have in common that the main goal is to get users. They are also all about motivating users to join the platform with non-financial incentives. So where many strategies focus on pricing these strategies show that there are many other ways to attract users to the platform. If they are motivated by one of these reasons and the platform is priced competitively, then users will be likely to join the platform.

Figure 19: 8 user attraction strategies

1. The Follow-the-rabbit Strategy

Do not start out as a platform but first show you are successful with a non-platform project and attract users and producers first and then convert to a platform. Amazon for example used a follow-the-rabbit strategy, they first created a successful pipeline business and later converted to a platform with external producers.

2. The Piggy-back Strategy

Do not start your own platform but connect with an existing platform that already has a user base. PayPal for example piggy-backed on eBay before becoming an independent platform.

3. The Seeding Strategy

Attract a specific set of users by creating value for them on the platform and then other users who want to interact with that specific user set will follow.

4. The Marquee Strategy

Decide who your key users are, their participation will make or break the platform, and then provide incentives to attract those users. If you have the marquee users other users will also join the platform.

5. The Single-side Strategy

Create a product or service that benefits certain users, once you have those users convert to a platform and invite users from the other side of your platform. OpenTable for example first created table booking software for restaurants and got a lot of restaurants to start using the software before converting to a platform to connect people to restaurants with open tables.

6. Producer Evangelism Strategy

Get producers who already have a customer base to join your platform, then they can bring their customers to your platform. If the teacher uses Blackboard the students will also use it.

7. The Big-Bang Adoption Strategy

Attract a high volume of interest and traffic using traditional push strategies. The goal is to get a large group of users to join the platform at the same time. Twitter used big screens at a festival to display tweets live, by the end of the festival twitter had tripled its number of users.

8. The Micro Market Strategy

Launch the platform first within an already existing community. Facebook for example was first launched for Harvard students only which allowed Facebook to improve the platform before launching it for the rest of the world.

Reflection on the Strategy

This strategy is very interesting because it offers so many options on how to achieve the critical mass user base for a platform. Most other strategies only state that it is important to get users without giving any indication on how to actually get users to join the platform. So despite the fact that no information is given about any other platform dimensions it still offers valuable insights.

Dimensions	Possible values	Positive effects	Negative effects
User attraction strategy	Follow-the-rabbit	Users already have faith in the company because they already know its non-platform product	n/g
	Piggy-back	Easier to start because there already is an existing user base	n/g
	Seeding	The users will attract a second user type	n/g
	Marquee	This user will convince others to also join the platform	n/g
	Single-side	The product has stand-alone value for users who are likely to stay once the platform capabilities are added	n/g
	Producer Evangelism	Developers bring their already existing user base to the platform	n/g
	Big-bang Adoption	A large group of users joins at the same time so critical mass can be achieved instantly	n/g
	Micro-market	The already existing community is already interested in interaction with each other	n/g

Critical Success factors for platforms	
Get users	Users are difficult to get when there are not many users on the platform yet, this is the chicken-or-egg dilemma which can be and has to be overcome.
Leverage network effects	the more participants, the greater the value produced
Pull instead of push strategy	Advertising is less effective compared to offering incentives for participation as a method for getting users to join the platform
Facilitate efficient exchanges of value	bring together producers and users in efficient exchanges of value
Use non-financial incentives to join	Motivate users to join the platform with non-financial incentives. They are very likely to join is the price of the platform is not too high.
Do not start out as a platform	Do not start out as a platform but first show you are successful with a non-platform project and attract users and producers first and then convert to a platform. Or by piggybacking on an existing platform. Create a product or service that benefits certain users, once you have those users convert to a platform and invite users from the other side of your platform.
Use specific user types to get users	Decide who your key users are, their participation will make or break the platform, and then provide incentives to attract those users. If you have the marquee users other users will also join the platform. Get producers, who already have a customer base to join your platform, then they can bring their customers to your platform. If the teacher uses blackboard the students will also use it.
Use push strategy for a kick-off	Attract a high volume of interest and traffic using traditional push strategies. The goal is to get a large group of users to join the platform at the same time.

Platform Strategy 10: Plug and Play Organization

In the book by Maurits Kreijveld a basic introduction to the platform business model is given [18]. He describes a platform as a combination of economic, technical and social elements. A plug and play organization according to him consists of the cloud, the crowd, a platform and an ecosystem. An organization does not have to have all competencies and resources to deliver a product or service themselves. They only need to have access to enough capabilities and capacity to orchestrate an entire user experience or contribute to a complete user experience. Plug into other organizations or let them plug into you. Everything is becoming digital which is enabling the rise of networked platform structures and economies of scale as well as the option to capture long tails in the market. As a platform owner one can benefit from innovations that are created around the platform, as a developer a platform can offer a basic infrastructure which allows the developer to remain agile as he does not need to maintain the infrastructure. New components and new collaborations are easily created on top of the platform. The key to a successful organization is being plug and play: it should be easy for the platform owner, developers and users to add and use new components on the platform. Standardization allows for easy communication and modularization, which is needed for a plug and play structure. The focus in the market is switching from product-focused to service- and relationship-focused. Relationships with customers are growing from a single point-of-sales interaction to continuous interaction. Products can be made on-demand and personalized. By collaborating organizations share knowledge and competencies as well as costs and risks. These new collaborations can improve customer satisfaction, develop new markets and increase the speed of innovation, all at a lower cost. Collaboration in flexible and agile structures is key for organizations in the future. Competition is between ecosystems, not between products. The value of the ecosystem comes from the number of participants in the ecosystem and can be measured by how much people or organizations are willing to pay to join the ecosystem. Ecosystems should be scalable, adaptive and innovative. Platforms can leverage network effects which can lead to the winner-takes-it-all effect. So to get the stage of network effects the chicken-or-egg problem first needs to be overcome. A platform should improve on the user experience of traditional products and enable transactions.

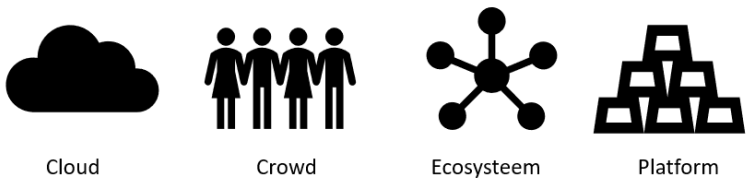


Figure 20: Building Blocks of a Plug and Play Organization

The main strategy focusses on two dimensions, openness and governance (central or distributed). A platform can be anything from completely open to completely closed and governance can range from completely centralized to completely distributed. Although some platforms may remain in the same place on the grid, many platforms make a journey through time and their place in the grid changes as the platform matures. This journey in illustrated by the green arrow in figure 21. The chosen platform structure in

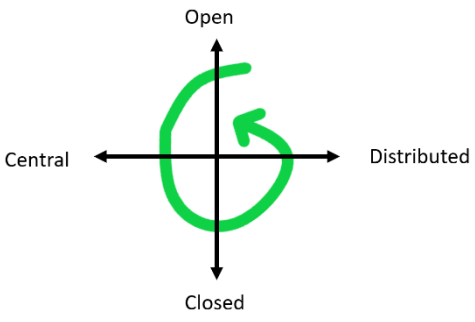


Figure 21: Openess and Governance

terms of openness level and governance will determine the type of collaboration that takes place around the platform.
There are four types of business models described in the book and the different shapes of these business models can be seen in the figure below.

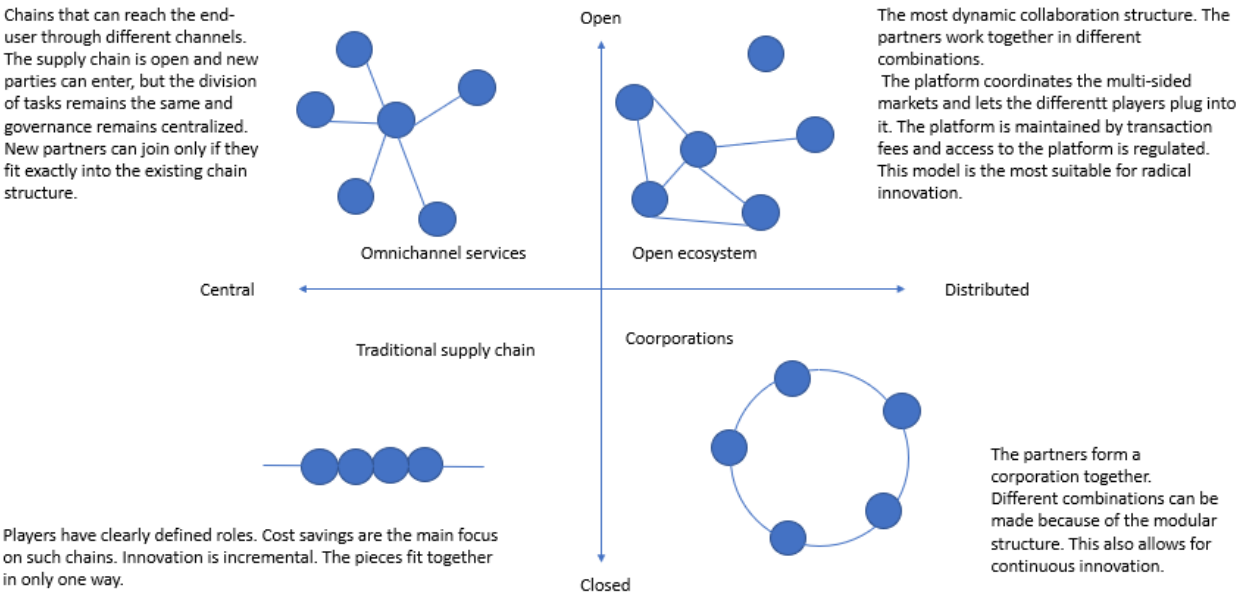


Figure 22: Business Models

Open Versus Closed

The main concept of this strategy is shown in the next figure.
Openness is the amount of control the platform owner exerts over the platform architecture, both physical components and data-infrastructure.

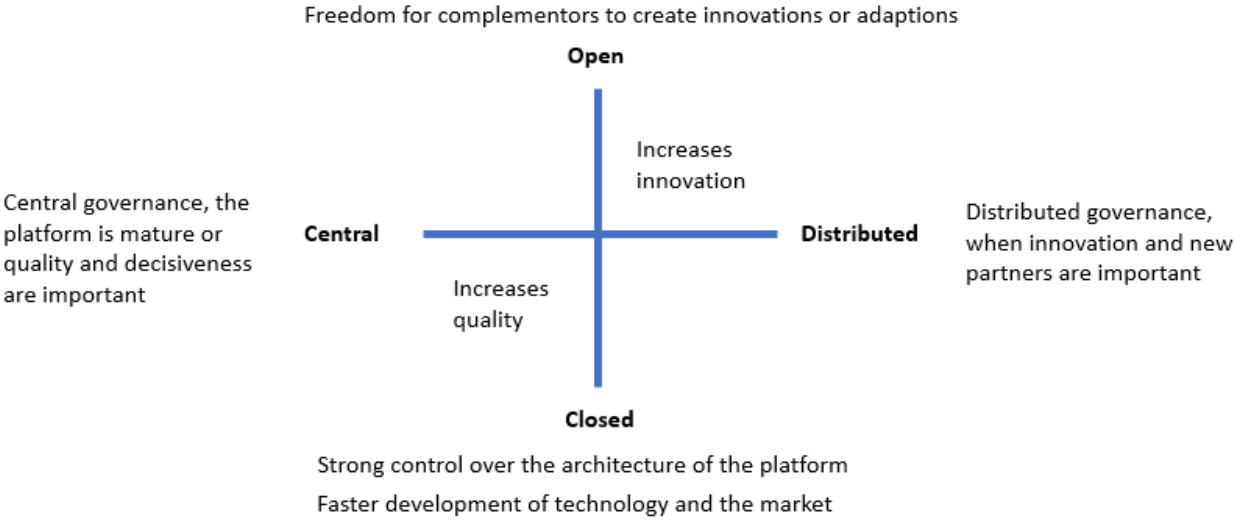


Figure 23: Two Dimensions: Openness and Governance Distribution

In this table the properties of open and closed platform structures are given.

Open	Closed
More freedom for complementors to make changes or additions to the components of the platform	Control on hardware creates a seamless user experience and compatibility of products
Openness will increase the innovative ability of the platform. This is important in a young market where there is no winner yet. It offers many options to partners to help develop the platform.	Platform can be established fast because of limitations and standardization agreements
Risk: growth that is out of control, platform components are no longer compatible with all other platform components. There is so much on offer the user is overwhelmed by it.	More control over quality, revenues, positioning and focus
Room for new innovation because of diversity in hardware and software, branches are allowed.	Risk: innovation is limited, it can only take place within se, predefines limits. Slows down innovation, especially in developing markets
Loss of control. Risk of out of control growth and branches that are no longer compatible, lessened user experience, dispersion on the user side	Closing the platform will allow it to optimize revenue streams
Loss of value	Easier to attract sponsors, especially when the platform is offered for free to users
Quick user growth	
Large offering of products and services	
Lower development costs	

Central versus Distributed Governance

This dimension refers to the type of governance that is used on the platform. The platform can be controlled by a central organization or be more collaborative between several organizations who organize the platform together in a distributed manner.

Central	Distributed
Strong control and governance allow successful platforms to scale up quickly, leading to a winner-takes-it-all all effect	Combining knowledge and best practices to quickly improve the quality of the product
Control over the complete user experience, seamless integration of all components and applications	Freedom can lead to higher acceptance and a faster spread of the platform and accompanying standards
Dividedness: different branches are created, and companies must choose which to join or to join a competing platform.	Greater community support for the platform because social initiatives are given a chance, the sharing economy
	Risk: the creation of islands and random growth of standards, confusion for the consumers what is offered and at what quality

Platform Evolution

In the book the 5 phases of platform evolution are described because a third dimension that is key to a strategy is time. These phases are summarized in the next figure and the following table.

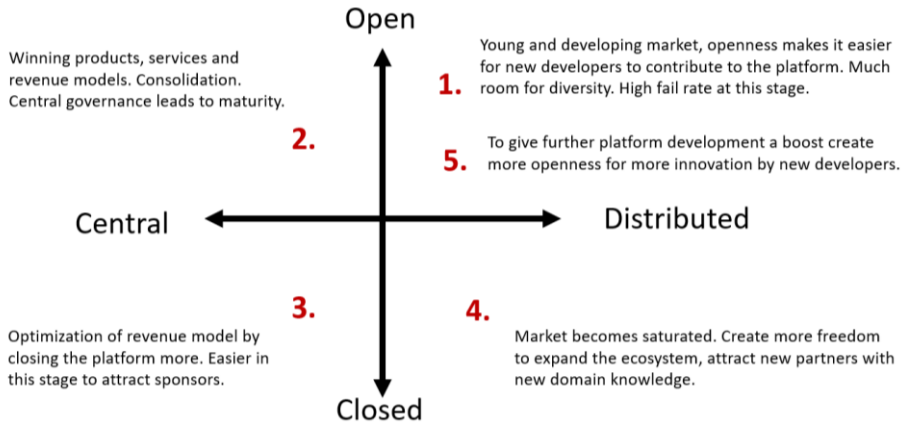


Figure 24: Platform Evolution: From an Open to a more Controlled Ecosystem

Phase	*Place in grid*	*Description*
Phase 1	Open and distributed	Young and developing market: more openness is appropriate, or even complete openness trough opensource hardware and software. This will make it easy for new developers to join. Especially when governance is distributed there much room for a diversity of applications which can be tested on the market. In this stage many platforms and products fail, the markets still need to mature.
Phase 2	Open and central	When it becomes more clear which products, services and revenue models are winning then it is time for the platform to start consolidating. By creating more central governance the platform can continue to grow and reach maturity. More central governance can be used to limit the number of branched and stifle random and exponential growth, the user experience can be improved, and speed of the platform can be controlled. When there are too many users with outdated hardware and software this will have a negative influence on the growth and innovative ability of the platform in the long run.
Phase 3	Closed and central	This phase the platform is closed more so the revenue model can be optimized. Because the platform is closed it is easier to attract sponsors to the platform.
Phase 4	Closed and distributed	As the market becomes saturated the governance structure should become more distributed so there is more freedom to expand the ecosystem. New partners with new domain knowledge can then be attracted.
Phase 5	Open and distributed	The market is saturated and the platform is mature. By opening up the platform again a boost can be given to innovation and new developers can join the platform.

Platform Strategy Questions

In the book many questions are presented that a platform owner could ask himself to establish a successful platform. A selection of these questions, those which are likely most relevant for Tersof, is given below. By answering these questions the company can gain a better understanding of how platform thinking could affect their business.

1. Who are the players in your own supply chain or network? Which parts of your work could these players take over for you when given the freedom to do so? Which capabilities do they need to do so? Which resources do they need to have access to?
2. Which players from adjacent markets supply products or services which you could build upon or which could be plugged into your products or services? What is needed to be able to collaborate with these players? How could you expand your product or service in their direction?
3. Which user communities are active in your market or around your product? Do you already have a connection to these user communities? What could your customers contribute to new product development?

4. What does the user need to 'do it themselves' and how can you deliver this with as little overhead as possible and as few links in your supply chain?
5. What is your added value in the new supply chain?
6. How could you open up your product or service to third parties, so they can develop products or services on top of them? Which hardware, software, infrastructure of process could be a basis for this?
7. Which part of your organization could you share with other organizations? In which area could you join forces by bundling components?
8. What component could be an industry wide standard?
9. Who are the developers or complementors and who are the customers on your platform?
10. Which interactions between consumers and suppliers does your platform facilitate? Why is there currently friction in these interactions? In which way will you remove this friction?
11. How will your partners profit form platform growth? What value can the platform have for the platform supplier, complementor and user?
12. If you were to offer your product for free from now on, from which activities would you then be able to gain revenues? Who could be a sponsor for your platform?

Reflection on the Strategy

A strong point of this strategy is the simplicity, it focusses on only two dimensions (or three if you count the time dimension). This makes the strategy and the book it comes from a simple introduction to platform strategy thinking. However, this could be considered the downside of the strategy as focusing only those factors may be too much of a simplification of reality to still be useful.

Dimensions	Possible values	Positive effects	Negative effects/ risks
Open/Closed **(Spectrum between completely open and completely closed)**	Open	More freedom for complementors to make changes or additions to the components of the platform Openness will increase the innovative ability of the platform. This is important in a young market where there is no winner yet. It offers many options to partners to help develop the platform. Room for new innovation because of diversity in hardware and software, branches are allowed. Quick user growth Large offering of products and services Lower development costs	Risk: growth that is out of control, platform components are no longer compatible with all other platform components. There is so much on offer the user is overwhelmed by it. Risk of out of control growth and branches that are no longer compatible, lessened user experience, dispersion on the user side. Loss of value. Loss of control.
	Closed	Easier to attract sponsors, especially when the platform is offered for free to user. Closing the platform will allow it to optimize revenue streams. More control over quality, revenues, positioning and focus. Platform can be established fast because of limitations and standardization agreements. Control on hardware creates a seamless user experience and compatibility of products	Risk: innovation is limited, it can only take place within se, predefines limits. Slows down innovation, especially in developing markets
Central/Distributed	Central	More maturity. Control over the complete user experience, seamless integration of all components and applications. Scale up quickly leading to a winner-takes-it-all effect.	Dividedness: different branches are created, and companies must choose which to join or to join a competing platform.

Dimensions	Possible values	Positive effects	Negative effects/ risks
	Distributed	More freedom for developers. Greater community support for the platform because social initiatives are given a chance, the sharing economy. Freedom can lead to higher acceptance and a faster spread of the platform and accompanying standards. Combining knowledge and best practices to quickly improve the quality of the product.	Risk: the creation of islands and random growth of standards, confusion for the consumers what is offered and at what quality
Time / evolution phases	Vary strategy through different maturity levels of the platform and the market	By varying the strategy through time the optimal choices for each phase in the lifecycle of the platform can be made. In the end this will lead to more users and complementors, higher quality, more innovation and a good revenue model.	Some developers may not appreciate the changes in openness and governance
	Constant strategy through different maturity levels of the platform and the market	No need to constantly reinvent the business model and structure	The strategy is no longer the perfect fit for the maturity level of the platform and the market

Critical Success factors for platforms	
Quality	A platform must have sufficient quality, this is controlled by the governance of the platform.
Be plug and play/ collaborate	An organization does not have to have all competencies and resources to deliver a product or service themselves. They only need to have access to enough capabilities and capacity to orchestrate an entire user experience or contribute to a complete user experience. Plug into other organizations or let them plug into you. By collaborating organizations share knowledge and competencies as well as costs and risks. These new collaborations can improve customer satisfaction, develop new markets and increase the speed of innovation, all at a lower cost. Collaboration in flexible and agile structures is key for organizations in the future.
Utilize economies of scale and capture the long tails of the market	Everything is becoming digital which is enabling the rise of networked platform structures and economies of scale as well as the option to capture long tails in the market
Standardization and modularization	Standardization allows for easy communication and modularization, which is needed for a plug and play structure in which components can easily be added and removed.
Be customer and service oriented	The focus in the market is switching from product-focused to service- and relationship-focused. Relationships with customers are growing from a single point-of-sales interaction to continuous interaction. Product are on-demand and personalized.
Focus on the entire ecosystem	Competitions is between ecosystems, not products. The value of the ecosystem comes from the number of participants in the ecosystem and can be measured by how much people or organizations are willing to pay to join the ecosystem. Ecosystems should be scalable, adaptive and innovative.
Focus on improved user experience	A platform should improve on the user experience of traditional products and
Enable transactions	A platform should enable transactions that would otherwise not be possible or would be very difficult.
Network effects/ winner-takes-all/ chicken-or-egg problem	Platforms can leverage network effect which can lead to the winner-takes-it-all effect. So to get the stage of network effects the chicken-or-egg problem first needs to be overcome

Platform Strategy 11: Characteristics and Success Factors of Digital Platforms

In this report 6 key factors for establishing a successful digital platform are presented [28]. The platform market has different dynamics than traditional business. Intermediaries acting as the central link who connect different user groups through a platform are becoming increasingly important. The platform creates an opportunity for efficient interaction between the user groups. The user groups benefit from the size of the other user groups on the platform. A platform provides a network and transparency.

The data in this report is based on interviews with managers from 14 different German platform companies who are engaged in B2B commerce. The study provides an overview of features of the platform economy and key factors for establishing a successful digital platform. The report makes a distinction between transaction centered and data-centered platforms. A transaction centered platform functions as a facilitator, it brings supply and demand sides together and facilitates transactions between them. A data centered digital platform has a focus on data networking, there is a data-based integrated system offered on the platform in which complementary products are linked and together they form an integrated system, which is also known as a digital ecosystem. In such a digital ecosystem hardware, software and data or services can be integrated. For each of these two platforms types the report proposes an ideal design.

Advantages for Established Companies who enter the Platform Market

It can be a huge advantage for a company to already be established as a non-platform business before entering a platform market for the following reasons:

- They already have made a name for themselves
- They can offer complementary products themselves
- Currently existing business ties and contacts can be used to attract the relevant players to the ecosystem

Characteristics of Digital Multi-sided Markets

A digital multi-sided market has the following characteristics, which are explained below

- High degree of scalability and range as well as data analysis
- Low transaction costs due to technological and digital innovations
- Strong network effects
- Special market dynamics

High Degree of Scalability and Range as well as Data Analysis

Acquiring additional computing capacity is no longer a large obstacle scaling up is easy for digital platforms. Because of this platforms are very adaptable and they have no geographic boundaries, although there can be cultural, linguistic and legal issues for international platforms. Data is a core component of digital platforms. Data analysis can be used to discover and create new products and services for the platform. When data is exchanged on the platform compatibility and interoperability of the data is important.

Low Transaction Costs

Technological developments have made low cost transactions possible and lower transaction costs will lead to more transactions taking place. If a transaction takes place depends on how difficult it is to find someone to make a transaction with (information cost), negotiation and contract costs, amendment costs, monitoring and enforcement costs. Transaction costs influence how much economic activity takes place, so lower costs will lead to more activity.

Strong Network Effects

Different user groups are connected by the platform. The number of transaction partners that the platform offers determines its value. Positive indirect network effects: every group benefits from the network size of the other groups, so there is the chicken-or-egg problem for newcomers on the platform

market. Critical mass needed is needed first and the network effect will keep increasing all sides of the platform.

Special Market Dynamics

Compared to traditional markets digital markets are highly dynamic, the pace of change is very high. There are constantly new competitors with new innovative business models. Because of the network effect a 'winner-takes-it-all' scenario is likely to occur. Timing is critical, you have to be the first to reach the critical mass and need to enter the market early and proactively to build up a high number of users. The network effect causes market dynamics which are likely to create monopolies, however, differentiation and multihoming can counteract the monopoly effect. Multihoming is the parallel use of more than one platform by individual users.

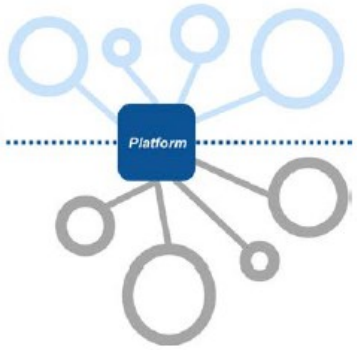


Figure 25: The platform as the central player in a two-sided market with positive indirect network effects. The more grey players use the platform, the more the platform appeals to blue players – and vice versa.

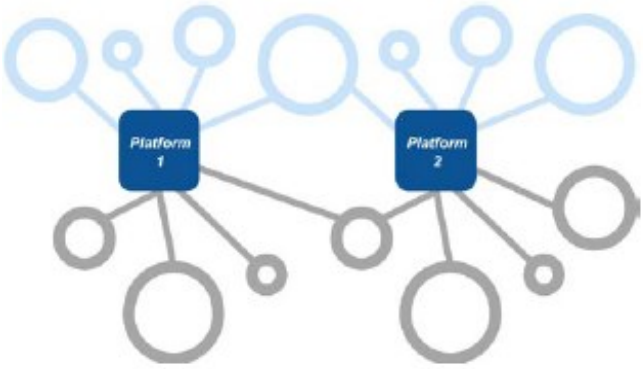


Figure 26: Two-side market with multihoming.

Key Factors of Successful Platforms

- Function of the platform
- Sales and revenue concept
- Openness of the digital platform
- Platform independence
- Contact to (potential) platform users
- Dynamic strategy

Function of the Platform

The function of a platform is determined by what is offered on the platform, by whom and to whom. The function of the platform can be transaction-centric or data-centric. The focus is on functioning as facilitator in a transaction-centric platform.

Methods to reduce transaction costs for a facilitator platform:

- A good search function will lower the information cost
- Standardized transaction handling and contracts reduce the negotiation and contract costs
- Reputation mechanisms

In a data-centric platform the focus lies on data-based networking, there is an integrated system based on data. Complementary hardware and software can be linked to the platform that then comprises the ecosystem. There is a reduced transaction cost by compatibility and interoperability between the different system components. The platform is responsible for usability and satisfaction management.

Sales and Revenue Concept

What will be sold on the platform and how will it be priced? Asymmetric pricing is used on most platforms. One group usually subsidizes the other. There are two rules for optimum pricing:

1. The group that is more interested in the other groups pays a higher price, for example advertisers wanting to reach Facebook users.
2. The more price sensitive group pays less than the not so price sensitive group.

It can make sense to enter the market at a low price to quickly gain users. Price can be kept artificially low to create room for growth for a digital ecosystem with long-term profitability.

Openness of the Digital Platform

Openness is determined by access control, multihoming and compatibility. Conditions for participation on the platform can limit access to the platform as a quality control mechanism. Multihoming: allowing users of the platform to also use other platforms or restrict this? Restrictions can be difficult to enforce and are generally perceived as negative by users.

Platform Independence

A platform can be an independent third party or be integrated with on the platform sides. For example if the platform owner also offers products or services on the platform.

Contact with (Potential) Platform Users

How is marketing organized and how will platform user groups be contacted? The platform can easily be found online, but choices can be made how to present the platform and trough which channels potential users can be reached. Traditional and online marketing channels can be used.

Dynamic Strategy

Digital platform markets are dynamic, and this should be reflected in the strategy. The chicken-or-egg problem is a key dynamic challenge. How to get users at the start when the users they are interested in have not yet joined the platform? Once critical mass is achieved more users will join the platform but achieving critical mass is difficult. Platforms should be agile, flexible and easily adapted.

Three basis approaches to the chicken-or-egg problem, they can be combined.

1. The platform is integrated into one side of the market and can offer complementary products or services as soon as it is launched
2. One group of users is already attracted before the platform is launched so there is already a critical mass on at least one side of the platform.
3. A special effort is made to get a large group of users immediately after the launch of the platform, for example through low pricing or push marketing.

Figure 27: Ideal type of transaction-centric digital platform

Figure 28: Ideal type of data-centric digital platforms

Two Types of Platform

The two different platform types have their own requirements and benefits, but there are also some characteristics shared by both platform types. Both types of B2B platforms have to consider the trade-off between quality and reliability versus growth, speed and agility. Usually it makes sense to enter the market with a simple and reliable service and gradually expand and grow in response to customer feedback.

The Ideal Digital Transaction-centric Platform

According to the research in this report the ideal digital transaction-centric platform has the following characteristics:

- It is a meeting place for transactions, it brings supply and demand together
- It offers a suitable information and search function
- It offers rating and reputation mechanisms
- Access to the platform is restricted and suppliers of products or services undergo screening to ensure quality on the platform
- The platform is a neutral marketplace and independent of other market players
- It charges a fee for access and use
- The pricing structure that is used is group-specific and therefore asymmetric
- To generate critical mass strategic partnerships are created or the platform should already have users before going to market
- Once the platform has entered the market it should have a pro-active approach to quickly gain a high number of users
- It must be credibly perceived as a neutral marketplace or a source of reliable and professional services
- It has strategic partnerships who do not have an adverse effect on perceived independence of the platform
- The pricing strategy is optimal to achieve critical mass
- It uses surveys and respond quick to customer feedback

The Ideal Data-centric Digital Platform

According to the research in this report the ideal digital data-centric platform has the following characteristics:

- It is an integrated data-centric system of complementary products
- It offers data analysis for users of the platform
- It offers coordination of usability and customer-satisfaction in the related eco-system
- There is an accreditation system for component suppliers
- There are technical or minimum requirements for access to secure data interpretability and interoperability of integrated components
- A fee is charged for access and use
- Additional fees are charged for certain data analyses
- price structure is asymmetric and group-specific
- There is already a sufficient quantity of system components and suppliers on the platform and integrated into the ecosystem before the market is entered so a critical mass is quickly achieved
- There are strategic partnerships with suppliers of complementary products
- The system is seen as open and providing easy access

Reflection on the Strategy

The fact that this strategy is based on 14 B2B platform businesses makes it very interesting. With some sources it is difficult to find out what they are based on and how accurate or relevant that base is. In this case the study has been done on companies that are similar to Tersof and therefore the results are likely to be transferable to Tersof. Many point mentioned in the strategy are also mentioned in other strategies and they are supported by arguments and good reasoning.

Dimensions	Possible values	Positive effects	Negative effects
Asymmetric pricing/sponsoring	yes	Grow faster	n/g
Transaction costs (negotiation, contract, amendment and enforcement costs)	Low (less difficult to find someone to make a transaction with)	More transactions take place, more economic activity takes place	n/g
	High	Less economic activity takes place	
Monopoly effect counteraction strategy	Differentiation	The monopoly, or winner-takes-it-all, effect is counteracted	n/g
	Multihoming		
Openness (Access control)	Conditions for participation	Access to the platform is limited and thereby the quality can be controlled	n/g
	No conditions	n/g	Risk of bad quality which cannot be kept out of the platform
Compatibility	Yes/no	n/g	n/g
Openness (access control, multihoming and compatibility)	Range from open to closed	Open: Conditions for participation on the platform can limit access to the platform as a quality control mechanism.	n/g
Multi-homing	Yes	Monopoly can be counteracted	n/g
	no	n/g	Restrictions can be difficult to enforce and are generally perceived as negative by users
Platform independence	Independent third party maintains the platform	Can increase trust	n/g
	Platform owner is integrated with one of the platform sides (when the platform owner offers products and services on the platform)	Standalone value before partners have joined the platform	
Contact to (Potential) Platform Users	Online marketing channels	Easy to find the platform	n/g
	Traditional marketing channels	n/g	
What is sold on the platform, by whom and to whom	n/g	n/g	n/g
Platform function	Transaction centric: focus on function as facilitator.	Reduced transaction costs for a facilitator platform: A good search function will lower the information cost Standardized transaction handling and contracts to reduce the negotiation and contract costs Reputation mechanisms	n/g
	Data-centric: focus on data-based networking, there is an integrated system based on data. Complementary hardware and software can be linked to the platform, that then comprises the ecosystem.	There is a reduced transaction cost by compatibility and interoperability between the different system components. The platform is responsible for usability and satisfaction management.	
Chicken-or-egg problem solving strategy **three basis approaches to the chicken-or-egg problem, they can be combined.**	The platform is integrated into one side of the market and can offer complementary products or services as soon as it is launched	n/g	n/g
	One group of users is already attracted before the platform is launched so there is already a critical mass on at least one side of the platform.		

Dimensions	Possible values	Positive effects	Negative effects
	A special effort is made to get a large group of users immediately after the launch of the platform, for example through low pricing or push marketing.		

Critical Success factors for platforms	
Low information costs/ search function	A search function will lower the information cost.
Reputation mechanisms	Offer rating and reputation mechanism to ensure quality and trust and be viewed as reliable.
Usability	Use surveys to get customer feedback and adapt. Consider usability.
Quality control through access restriction	Access to the platform is restricted and suppliers of products or services undergo screening to ensure quality on the platform. This accreditation for system component suppliers by the platform ensures quality.
Neutrality	The platform is a neutral marketplace and independent of other market players. Neutrality will give the platform credibility. Strategic partnerships can make the platform be perceived as dependent. By offering product or services yourself on the platform you can remain independent.
Charge for access and use	Charge a fee for access and use. Charge additional fees for certain data analyses.
Group-specific pricing (sponsoring)	There are two rules for optimum pricing: 1. The group that is more interested in the other groups pays a higher price. For example advertisers wanting to reach Facebook users. 2. The more price sensitive group pays less than the not so price sensitive group. It can make sense to enter the market at a low price to quickly gain users. Price can be kept artificially low to create room for growth for a digital ecosystem with long-term profitability.
Create partnerships	Enter into strategic partnerships with suppliers of complementary products, this will support the generation of critical mass.
Ensure quality and function with minimum requirements	Technical or minimum requirements for access to secure data interpretability and interoperability of integrated components
Increase economic activity with low transaction costs	Technological developments have made low cost transactions possible and lower transaction costs will lead to more transactions taking place. If a transaction takes place depends on how difficult it is to find someone to make a transaction with (information cost), negotiation and contract costs, amendment costs, monitoring and enforcement costs. A good search function will lower the information cost. Standardized transaction handling and contracts to reduce the negotiation and contract costs. There is a reduced transaction cost by compatibility and interoperability between the different system components. Transaction costs influence how much economic activity takes place, so lower costs will lead to more activity.
Monopoly counteraction	How to counteract the monopoly effect: differentiation and multihoming.
Achieving critical mass to get network effects **Grow fast, achieve critical mass** **Overcome chicken-or-egg problem**	Be the first to achieve critical mass because of the winner-takes-it-all effect. To achieve critical mass already have a sufficient quantity of system components and partners on the platform and integrated into the ecosystem before entering the market. Quickly get a high number of users to achieve critical mass and create network effects which will then likely lead to a monopoly position. The value of the platform is determined by the number of users on it. Overcome the chicken-or-egg problem. Pricing is essential to achieve critical mass.
Be open and easily accessible	Create a system that is seen as open and provide easy access
Internationality	there can be cultural, linguistic and legal issues for international platforms, be conscious of those
Data compatibility	When data is exchanged on the platform compatibility and interoperability of the data is important. Technical or minimum requirements for access to secure data interpretability and interoperability of integrated components.
Openness of the digital platform	Access control, multihoming and compatibility. Conditions for participation on the platform can limit access to the platform as a quality control mechanism. Multihoming: allowing users of the platform to also use other platforms or restrict this? Restrictions can be difficult to enforce and are generally perceived as negative by users.
Pricing	Charge a fee for access and use. Charge additional fees for certain data analyses.
Asymmetric pricing	Use asymmetric group-specific pricing. The group that is more interested in the other groups pays a higher price. For example advertisers wanting to reach Facebook users. The more price sensitive group pays less than the not so price sensitive group.
Low price entry strategy	It can make sense to enter the market at a low price to quickly gain users. Price can be kept artificially low to create room for growth for a digital ecosystem with long-term profitability.
Gradually expand the platform	Both types of b2b platforms have to consider the trade-off between quality and reliability and on the other hand growth, speed and agility. Usually it makes sense to enter the market with a simple and reliable service and gradually expand in response to customer feedback.
Connection	brings supply and demand together

Critical Success factors for platforms	
Be established before entering the platform market	It can a huge advantage for a company to already be established as a non-platform business before entering a platform market for the following reasons: they already have made a name for themselves, they can offer complementary products themselves, currently existing business ties and contacts can be used to attract the relevant players to the ecosystem
Be agile/ dynamic strategy	Platforms should be agile, flexible and easily adapted.

Platform Strategy 12: 5 Questions

Title: How to Launch Your Digital Platform, Benjamin Edelman, 2015 April issue Harvard Business Review, [29].

Customer mobilization is a challenge, you need early users to join the platform before it has reached scale. The most successful platforms connect different types of users to each other. You need enough of the different user types.

Successful platforms:

- By enabling communication and commerce that otherwise might not occur significant value can be created
- Have modest operating costs because they don't have tangible goods or inventory
- Have a protected position once established through network, users rarely leave a vibrant platform

The Strategy: Asking Five Basic Questions

1. Can I attract a large group of users at once?
2. Can I offer stand-alone value?
3. How will I build credibility with customers?
4. How should I charge users?
5. Can I make my platform compatible with legacy systems?

Can I Attract a Large Group of Users?

It is important to have a large number of users on the platform shortly after the launch of the platform. This critical mass will cause the network effect to occur. To attract a large number of users there are two conditions, at least one of which needs to be true:

- The company already has the users it needs on another platform
- User data is publicly available and is a source of value when provided through the platform

Can I Offer Stand-Alone Value?

It is important to have value to offer to users even when there is not a large group of users on the platform, for example by creating apps and functionality. Two strategies can be considered for offering stand-alone value:

1. Start with an industry niche
2. Find or build small social groups

How Will I Build Credibility with Customers?

Convince people to join the platform instead of the competitor by showing credibility. This can be done by attracting a key marquee platform contributor, preferably exclusive to the platform to drive user growth. This marquee contributor has already established credibility.

How Should I Charge Users?

There are strategic choices to be made about which users to charge and how to charge them.

Two important pricing levers:

1. Pay-as-you-go pricing reduces the risk for users.
2. User subsidies can be used to get a lot of users on the platform from the start.

Can I Make My Platform Compatible with Legacy Systems?

Users who join a platform often have used other platforms before joining yours. There must be a balance between being compatible with legacy systems and not limiting oneself to much by the support of legacy systems. The cost of switching should be low.

Figure 29: The Platform Builder's Checklist

Dimensions	Possible values	Positive effects	Negative effects
Stand-alone value	Yes	offer value before developers have joined the platform adding functionality to software or an app is easy Additional content will make the platform more appealing and therefore attract more users to it	Can be difficult, extra features might require costly hardware
	Small social groups	Users invite their friends and relatives, so the platform grows without the need for expensive marketing	n/g
	Start with and industry niche	Word-of-mouth will facilitate growth	
	no	n/g	
Pricing	Pay-as-you-go	Reduces risk, lowers the barrier to join the platform, users can try the platform before committing to it	n/g
Subsidizing	Yes	convince early joiners, being able to charge higher fees to other types of users such as advertisers	n/g
	no	n/g	
Compatibility with legacy systems	A little,	Just enough to attract new users	Legacy systems might be resistant to the platform, marrying yourself to outdated technology
Inhouse development	yes	Cheaper compared to attracting external users to offer features	Might be seen as competition by prospective users
Trust	Marquee user	This user already has credibility and by joining the platform he extends this credibility to the platform	High costs to attract marquee users
	Exclusive marquee user	Even greater effect than a non-exclusive marquee user	

Critical Success factors for platforms	
Large user base	The platform must have a large user base to be successful and therefore the chicken-or-egg problem has to be overcome
Standalone value	The platform should offer value in itself, even before others have joined the platform
Marque users	A marquee user can help create trust in the platform for potential users
Low user risk	Subsidize early joiners and offer pay-as-you-go to reduce the financial risks of joining the platform for users
Compatibility with legacy systems	Make the cost of switching for potential users low through interoperability, data conversions, and information synchronization but do not limit yourself by legacy systems, key to a successful launch

Reflection on the Strategy

This strategy source is very strong in its simplicity. The defined concepts are explained clearly but short and the concepts are well thought-out. The suggested dimensions overlap with many of those mentioned by other strategies and the proposed five basic questions are a good starting point for any company that is thinking about, or in the first phase of, becoming a platform.

Platform Strategy 13: Platform Leadership

This scientific study by Lee at al. identifies 5 interdependent dimensions for platform leadership: innovation ability, connectivity, complementarities, efficiency, and network effects. The 5 components are related to each other and influence each other [30]. This paper states that understanding web 2.0 characteristics is vitally important for platforms, because one of the most important things on the platform is the interaction among users.

Prerequisites for a Platform

- Have an essential role in improving a service or solving a technological issue
- Provide easy connectivity so the use of the platform can be expanded
- Allow for new and unintended end-users
- Reduce user risk through defined standards and practices
- Foster specialization among participants to increase the value and functionality when more users join the platform

Coring and Tipping

A possible strategy for platform leadership is coring. This entails developing a platform of a type that did not previously exist, this is also called a blue ocean strategy. Tipping is another platform strategy that entails using existing competitive advantage to gain platform leadership with a new core technology or service.

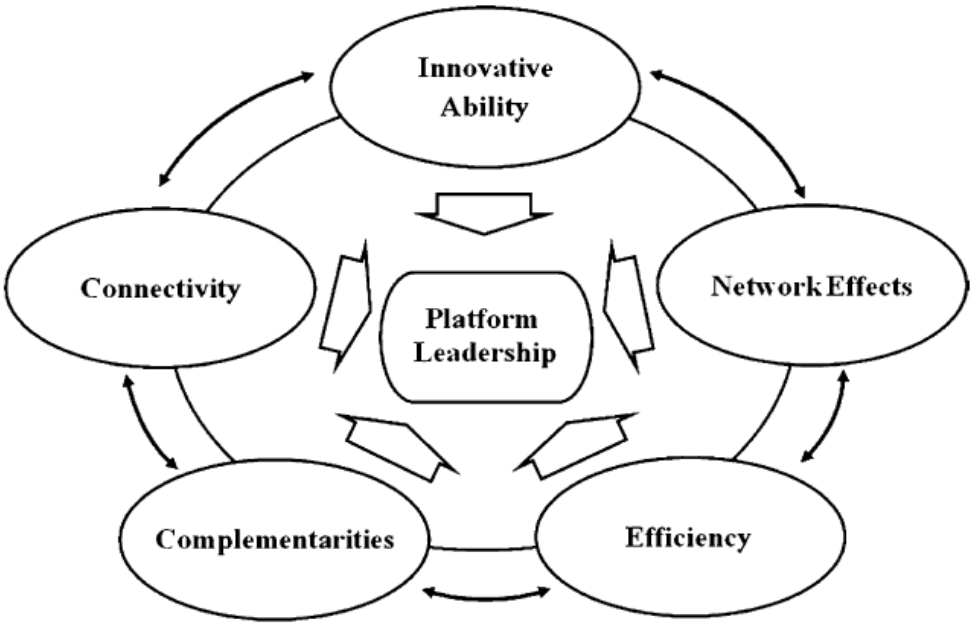


Figure 30: The 5 components of platform leadership strategy

Component 1: Innovation Ability

This component is about the potential to create new value. Openness will expedite innovation and lead to more value. A platform should solve an essential problem. New value can be created by solving current industry problems.

Component 2: Connectivity

Web 2.0 allows people to participate in content creation, so platforms should provide access to users to create complements for the platform. Easy connectivity will lead to more collective value creation and a long-tail effect with large margins.

Component 3: Complementarities

A complementor creates a product or service that makes your product or service worth more. A complementor is a value creator. The platform features openness, connection, conversation and collective intelligence, which accelerates the development of complementarities.

Component 4: Efficiency

Through platforms transaction costs can be reduced. A platform can also support many more transactions compared to traditional markets. More transactions at a lower cost per transaction will lead to increased revenue for the platform owner. On the platform the profit is shared with the developers on the platform, which will help increase the number of platform developers.

Component 5: Network Effects

Users determine the value of the product or service for other users, this is called the network effect. There is no sense in owning a telephone if no one else has one. There are 4 types of network effects: direct network effects, indirect network effects, two-sided network effects, and social network effects. Direct network effects: with more users the value of the good or service increases. Indirect network effects: new products and services are created to complement to original product or service as more

users join the platform. Two-sided network effects: an increase in one user group will increase the value of the product or service for the other group. Social network effects: groups of participants are joined on the network.

Reflection on the Strategy

This research paper very strongly focused on web 2.0, this feels somewhat outdated in 2018 with the rise of web 3.0 technologies, but the five components that are presented do still make sense today. There are some dimensions mentioned, however very little details about the possible values of the dimensions and results of those values are given.

Dimensions	Possible values	Positive effects	Negative effects
Openness	Open	Expedite innovation and lead to more value	n/g
	Closed	n/g	
Conversation	n/g	n/g	n/g
Connectivity	Easy	More value creation and more long tail effects	n/g
	Difficult	n/g	
Connection	n/g	n/g	n/g
Collective intelligence	n/g	n/g	n/g

Critical Success factors for platforms	
Innovation ability	New value must be created by solving current problems
Easy connectivity	Easy connectivity will lead to more collective value creation and a long-tail effect with large margins
Complementarities	This factor is influenced by the following dimensions: openness, connection, conversation, and collective intelligence.
Efficiency	Use the platform capabilities to reduce transaction costs
Network effects (direct, indirect, two-sided and social)	Users determine the value of the platform, so the platform should have a large userbase

Platform Strategy 14: Innovation versus Breaching

In this scientific paper three different strategies are studied [31]. A complementary based innovation strategy, a breaching strategy and a combination of these two strategies. There is much turbulence and unpredictability in the interdependent multi-sided platform market, consumer preferences and needs change frequently, and platforms are complex adaptive systems, so the focus of a strategy should be achieving and maintaining users in the long run and not only the chosen strategy but also how it is pursued will be important. There is fundamental uncertainty due to the nonlinear interactions among the diverse users operating on the platform. In a turbulent environment superior exploration and adaptability are key for platform organizations. What consumers claim to desire now may not be what they adopt a few months from now. The combination of diversity, adaption, connectedness and interdependence among agents make platform ecosystems complex adaptive systems. Platforms are different from traditional business because of the strength of network effects, the high switching costs, governance and pricing structures and product design. The interdependency on the market leads to increased complexity. Because of this increased complexity traditional organization learning, keep doing things that go well and stop doing things do not give good results, should be transformed into a new learning style of searching and experimentation. Direct and indirect network effects should be taken into account and a platform owner should attempt to increase the costs of switching in his favor.

With more knowledge about consumers preferences platforms can get a larger market share. Consumer knowledge is competitive intelligence that complements the innovation process. The platform can then offer products which are aligned with the needs and preferences of consumers.
As a platform owner you can focus on getting customers away from the competition by being innovative, which may be difficult because of network and lock-in effects, or you can use a breaching strategy to gain access to customers without having to migrate them.

"These feedback loops and increased connectedness and interdependencies of nonlinearly interacting diverse constituents of an ecosystem lead to increasing complexity, thereby increasing the turbulence and unpredictability in platform ecosystems." (Ozer & Anderson Jr., 2015)

Switching Costs

Cost of switching determines how easy it is for users to change platforms. If there are no switching costs, users are likely to multi-home. When there are many relationships between the different platform offerings and these relationships are platform specific, the switching costs increase, and this will create a lock-in effect.
There are three types of switching costs:

- Procedural (economic risk, evaluation, learning, setup costs)
- Financial (benefit loss, monetary loss costs)
- Relational (personal relationship, brand relationship cost)

Network Effect

When the network size of a platform increases the potential usefulness of the platform for consumers also increases. Direct network externalities: when more users adopt the platform the value of the platform for the users increases. Indirect network externalities: the value of the platform for users increases when there are more complementary offerings on the platform. Network externalities together with a high cost of switching contribute to the lock-in effect. The lock-in effect in turn often leads to the winner-takes-it-all effect.

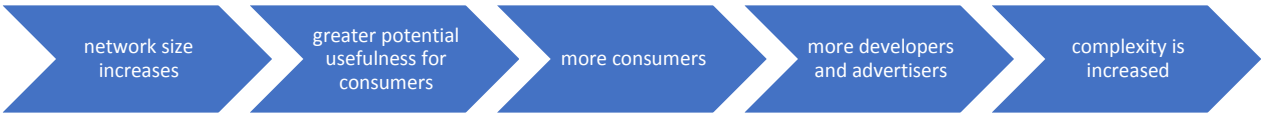


Figure 31: Network Effect

Diversity in Consumer Preferences

There is a large variety in consumer needs and preferences and to support these needs and preferences internal variety is needed. Internal variety is the creation of a large repertoire of offerings that reflect the variety in consumer needs and preferences and by exploratory offerings with can be but don't have to be bundled with the current offerings. The internal variety of the platform should also match the variety that is imposed by the environment. The more innovative complementary offerings are offered on the platform the better it will match consumer needs and preferences.

Breaching Strategy

The Breaching Method:

1. Establish the demand of consumers.
2. Determine what applications are demanded by consumers that are not offered by the rival platform with the most users.
3. Offer the most wanted application to the rival platform with the highest number of users.

A breaching strategy entails expanding into rival territories by offering a product or service on a rival platform. By doing so access can be gained to the customers of this rival platform without requiring the customers to switch platforms so there is no hindrance from lock-in effects. Another benefit of this strategy is that intelligence on consumer and complementor needs and preference can be collected, there is an opportunity to interact with the customers of your competitor and do exploratory learning and thereby have better adaptability to consumer needs and preferences and later on more easily attract them to your platform. Feedback and reviews on a rival platform can also be a source of new knowledge and an opportunity for experimentation and learning by doing is created, which ensures robustness and adaptability. Access to the developer community can be gained. Learnings about consumer preferences on rival platforms can be used for new product development. This strategy will increase the fitness of the company as additional sales and revenue streams can be created. All these interactions lead to increased exploratory learning and thereby increased platform fitness. The breaching strategy complements and positively reinforces the effects of the complementarity-based innovation strategy. This method supports taking long jumps based on emergent learning from the platform of a competitor. Doing both long jumps and local research makes it more difficult for competitors to imitate you and therefore platform fitness is increased. Complexity from switching costs is managed better because breaching platforms are better aware of changes in the rival platforms. They can better create a lock-in effect by constructing the right barriers.

Innovation Strategy

Complementary-based innovation method:

1. Establish the demand of consumers.
2. Determine what applications are demanded that are not yet available.
3. Offer the most wanted application.

Because of the competitive nature of platforms innovation must be complementary to current offerings to increase the fitness of the platform. Innovation means introducing new applications which were not available yet. Complementary innovative applications are adopted by consumers, so they can benefit from economies of scope. Innovations can be product-level or complementary-based. When innovation only occurs at the product level this might not be sufficient to create network externalities and switching costs, with complementary innovation a high level of platform fitness can be achieved.

Combined Strategy

Using a breaching strategy in combination with innovation will lead to a higher preforming platform than a platform that only engages in innovation. Breaching strategy will give positive reinforcement and complements a complementary-based innovation strategy. Innovation strategy without breaching will

allow a platform organization to develop insights about consumer and complementor needs and preferences on their own platform, but no insight can be gained into the needs and preferences of consumers and complementors on competitive platforms. This strategy will lead to local and incremental innovation. A combined innovation and breaching strategy can be used to achieve and sustain higher fitness levels. Complementary-based innovation strategy will have a positive impact on platform success and fitness and this effect will increase when the complementary-based strategy is combined with a breaching strategy. The effects will be enriched and enhanced by the learning benefits of the breaching strategy. The combined strategy will make the platform more adaptable, but not necessarily more adoptable.

Reflection on the Strategy

This strategy focusses on different methods for innovation for the platform, most other strategies had different focusses making it an interesting addition.

Dimensions	Possible values	Positive effects	Negative effects
Getting customers strategy	Use innovation strategy (product level or complementary- based) Innovation: introducing new applications which were not available yet	Complementary innovative applications are adopted by consumers, so they can benefit from economies of scope. With complementary innovation a high level of platform fitness can be achieved.	Users have to migrate to your platform, difficulty because of lock-in effect and network effects. When innovation only occurs at the product level this might not be sufficient to create network externalities and switching costs.
	Use breaching strategy A breaching strategy entails expanding into rival territories by offering a product or service on a rival platform.	Gain access to rival platform's customers and developers, learn from (exploratory learning) them, interact with them, better understand needs and preferences for and attract them to your platform. New knowledge from feedback. Robustness and adaptability through constant experimentation. and reviews, learn by doing. No lock-in effect because there is no cost of switching to your platform, higher cost of switching away from your platform, easier to construct the right barriers for lock-in. Increased organizational fitness from additional app sale revenues. Interdependencies and the network effect can be managed more effectively. Complexity from switching costs is managed better because breaching platforms are better aware of changes in the rival platforms. More difficult for competitors to imitate you because of long jumps and therefore platform fitness is increased.	n/g
	Joint strategy Combined breaching and innovation.	Complementary based-innovation is positively reinforced and complemented. Develop insights into consumer needs and preferences on your own platform. Local and	The joint strategy will make the platform more adaptable but not necessarily more adoptable. No new insights into consumer needs and preferences of consumers on rival platforms.

Dimensions	Possible values	Positive effects	Negative effects
		incremental innovation. Achieve and sustain higher platform fitness levels. The benefits from breaching and innovation strategy are combined.	
Governance	n/g	n/g	n/g
Cost of switching from your platform **(lock-in)**	High. Many relationships between the different platform offerings and these relationships are platform specific, switching costs increase creating a lock-in effect.	Difficult for users to go to a different platform.	n/g
	Low	n/g	Users will multi-home.
Pricing structure	n/g	n/g	n/g
Product design	n/g	n/g	n/g
Internal variety (offerings)	Large repertoire of offerings	The more innovative complementary offerings are offered on the platform the better it will match consumer needs and preferences.	n/g
	Small repertoire of offerings	n/g	
Internal variety (bundling)	bundled or not bundled with current offerings	n/g	n/g
Research method for breaching strategy	Long jumps	Difficult for competitors to keep up with you. Gain new knowledge.	n/g
	Local research	Gain new knowledge.	
	both	Doing both long jumps and local research makes it more difficult for competitors to imitate you and therefore platform fitness is increased.	
Switching costs types	Procedural (economic risk, evaluation, learning, setup costs)	Switching costs can keep users on your platform.	Switching costs can prevent new users from joining your platform.
	Financial (benefit loss, monetary loss costs)		
	Relational (personal relationship, brand relationship cost)		

Critical Success factors for platforms	
Getting and keeping users	The focus of a strategy should be on getting users to join the platform and then convincing them to stay on the platform for a long time.
Learning style of searching and experimentation	Platform organizations should not rely on traditional learning methods but actively pursue new learning styles
Direct and indirect network effects	They will contribute to a higher cost of switching and attract users to your platform and make them more loyal to it. Direct network externalities: when more consumers adopt the platform the value of the platform for the consumers increases. Indirect network externalities: the value of the platform for users increases when there are more complementary offerings on the platform.
High cost of switching/ lock-in	Keep the costs of switching high in your favor, lock-in effect and a strong network will support this. Lock-in effects can lead to a winner-takes-it-all situation.
Consumer knowledge	Collect consumer knowledge to create a larger market share. Consumer knowledge is competitive intelligence to offer products aligned with consumer needs and preferences.
Network size	Potential utility of the platform for users increases as network size increases
Create a lock-in effect	Network externalities together with a high cost of switching contribute to the lock-in effect. The lock-in effect in turn often leads to the winner-takes-it-all effect.
Match environmental variety with internal variety	The internal variety of the platform should also match the variety that is imposed by the environment.
Focus on long-term user maintenance	the focus of strategy should be achieving and maintaining users in the long run
Superior exploration and adaptability	In a turbulent environment superior exploration and adaptability are key for platform organizations

Learning by doing	Because of this increased complexity traditional organization learning, keep doing things that go well and stop doing things do not give good results, should be transformed into a new learning style of searching and experimentation.

Platform Strategy 15: Bundling or Constellation

This scientific paper by Kalina Staykova and Jan Damsgaard presents a large literature review on platform strategies and it presents their own developed platform strategy framework [32].
Multi-sided platforms are very powerful and valuable business models compared to more traditional business models. However, even platforms that achieve critical mass can fail if they do not continuously evolve. New services need to be offered to the existing user base to lock them in, this is called expansion. Platform expansion can be done through the bundling of features, the launch of an envelopment attack or by attracting complementors. This is considered equifinality in the paper: the same end goal can be reached in different ways and from different starting points. There is flexibility in strategic choices and there is not a single best way to reach a goal.

The paper presents a list of platform design decision areas:

- Number of features and which features are included in the platform
- Number of and which affiliated distinct groups of participants to include
- Order of attracting groups of participants
- Structure of the platform
- Design of entry (one-sided, two-sided, multi-sided)
- Design of expansion
- Governance

In the paper a platform strategy framework is constructed and then applied to two case studies. The framework is created based on strategy dimensions they found in literature and for the dimensions the criterion is used that it is a strategy element that can impact the ability of the platform to operate and expand. They came up with five key elements which according to them make up a platform strategy.

The 5 Platform Strategy Elements

1. Platform design (features, number of sides, order in which sides are added to the platform, platform architecture)
2. Platform adoption (platform size/critical mass, network effects, switching costs, multi-homing costs)
3. Platform governance (pricing: subsidy side, revenue side; non-pricing: openness, decision rights, control, regulation of access, regulation of interaction)
4. Platform innovation (provided by platform, provided by complementors, evolvability, incorporation costs)
5. Platform ecosystem (number of complementors, orchestration: cooperation versus competition)

Expansion Design Approaches

The choice of expansion design will impact the platform strategy.
1. Platform bundling: adding features and additional services to the original value proposition.
2. Platform constellations: add a variety of services in separate platforms which share the same log-in credentials, so users can easily multi-home.

The choice between one of these or using a combination of them will have an impact on platform adoption, innovation, governance, business models and orchestration of the platform ecosystem.

Platform strategy	Platform Bundling	Platform Constellation
Adoption	Limited audience-making abilities High search costs for complementors	Risk of partitioning the user base. Need for mechanism allowing users to multi-home across platforms. The adoption of one platform drive the adoption of others Reduces search costs.
Innovation	Difficulty to build upon a feature Dependency on partners Competition with partners platforms	Easy to build upon a framework. Lower innovation risk. Coordination costs between platforms.
Governance	Difficulty monetizing cross-side network effects	Various business models Various subsidy and revenues sides (on each platform)
	High degree of openness. High degree of monitoring and censorship.	Less degree of openness. Selective openness on some platforms. High level of initial control and less monitoring.
Ecosystem	Multiple external complementors and partners. Conflict among complementors and between complementors and platform owner	Fewer external complementors. External partners organized around separate platforms.

Platform Bundling

Properties of Platform bundling:

- There is a strong need for monitoring and control as a result of the openness that is needed for the external complementors. Control must be an ongoing activity.
- This type of platform relies on external complementors. These complementors provide additional products or services. Because of the different complementors there needs to be governance and access rules for these complementors.
- Innovation is dependent on partners so there is little need to invest in new developments.
- It is difficult to experiment with different business models.
- Risk: new offerings from external parties need to be incorporated into the current value proposition which can be challenging.
- The options for future evolvability are limited because the bundling structure only allows separate features to be added.
- Risk: there is greater chance of conflict. Because there is entire ecosystem of partners to be managed the risk of conflicts between the partners or between partners and the platform owner is higher.

Platform Constellations

Properties of platform constellations:

- There is a closed structure, so the health of the constellation can be maintained.
- Some parts of the platform can be opened up for external in innovation, while at the same time other parts of the platform can be kept closed to maintain control over them.
- Tight control over the platform is required to ensure the health of the entire constellation.
- In the design phase a high level of control is required. By having strong rules in this phase less monitoring will be required once the platform is up and running.
- The platform owner is responsible for orchestration and coordination between the different platforms.

- There must be access rules and interaction regulations for user and complementors of the platform.
- Potential risk: too many subsidy sides which can jeopardize the revenue of the constellation.
- Selective about which partners to allow on the platform or all innovation are developed inhouse, so there is more control over the user experience.
- It is easier to experiment with different business models.
- A new service can be offered on a new standalone platform.
- New features can be added to build upon the user experience.
- A constellation can be more selective about which partners to allow on the platform.
- Different partners can be organized on separate platforms.

Reflection on the Strategy

The writers draw the same conclusion as I have done during my research: most papers only focus on a single strategy element and do not offer a complete strategy.
As stated in the paper there is no ideal approach; both bundling and constellations offer different risks and benefits and these are also discussed in the paper. This makes this source more detailed compared to most other strategies in which little information is given about the consequences of different strategic choices.

Dimensions	Possible values	Positive effects	Negative effects
Expansion design approach	Bundling	Very limited investment effort required for innovation, this is taken care of by complementors	The openness results in a strong need for monitoring and control and control is an ongoing activity. Reliance on external complementor, governance and access rules required. Difficult to experiment with different business models Risk: new offering from external parties need to be incorporated into the current value proposition which can be challenging. Risk: there is greater chance of conflict. Because there is entire ecosystem of partners to be managed the risk of conflicts between the partners or between partners and the platform owner is higher. The options for future evolvability are limited because the bundling structure only allows separate features to be added.
	Constellation	Ability to be selective about which partners are allowed on the platform. New features build upon the existing user experience and health of the entire constellation. More control over the user experience. Easy experimentation with different business models. Control only needed in the initial phase.	too many subsidy sides which can jeopardize the revenue of the constellation
Innovation	Bundling	n/g	Difficulty to build upon a feature Dependency on partners

Dimensions	Possible values	Positive effects	Negative effects
			Competition with partners platforms
	Constellation	Easy to build upon a framework Lower innovation risk	Coordination costs between platforms
Governance	Bundling	High degree of openness	Difficulty monetizing cross-side network effects High degree of monitoring and censorship
	Constellation	Various business models. Various subsidy and revenues sides (on each platform). Selective openness on some platforms. High level of initial control and less monitoring.	Less degree of openness
Adoption	Bundling	n/g	Limited audience-making abilities High search costs for complementors
	Constellation	The adoption of one platform drive the adoption of others Reduces search costs	Risk of partitioning the user base Need for mechanism allowing users to multi-home across platforms
Ecosystem orchestration (The choice between one of these or using a combination of them will have an impact on platform adoption, innovation, governance, business models and orchestration of the platform ecosystem)	Bundling	Multiple external complementors and partners	Conflicts among complementors and between complementors and platform owner
	Constellation	Fewer external complementors External partners organized around separate platforms	n/g
Number of features	n/g	n/g	n/g
Which features are offered	n/g	n/g	n/g
Platform sides	Number of and which affiliated distinct groups of participants to include	n/g	n/g
Order of attracting groups of participants	Users first, complementors first, specific user group first	n/g	n/g
Structure of the platform/ platform architecture	n/g	n/g	n/g
Design of entry	(one-sided, two-sided, multi-sided)	n/g	n/g
Platform expansion methods	the bundling of features	n/g	n/g
	the launch of an envelopment attack		
	attracting complementors		
Switching costs	High/low	n/g	n/g
Multi-homing costs	High/ low	n/g	n/g
Platform size	Small/ big	n/g	n/g
Subsidizing	Which side	n/g	n/g
openness	Proprietary(open)	Large number of complementors	Need for control and monitoring
	closed	Easier the maintain the health of a constellation	n/g

Dimensions	Possible values	Positive effects	Negative effects
governance	decision rights, control, regulation of access, regulation of interaction	n/g	n/g
evolvability	Low/ high	n/g	n/g
Incorporation costs	Low/ high	n/g	n/g
innovation	By platform owner	n/g	n/g
	By complementors		
Ecosystem design	Number of complementors	n/g	n/g
Ecosystem orchestration	Cooperation Or Competition	n/g	n/g

Critical Success factors for platforms	
Platform expansion strategy	The choice of expansion design influences platform adoption, governance, business model, innovation and orchestration of the platform ecosystem.
Continuously evolve	even platforms that achieve critical mass can fail if they do not continuously evolve New services need to be offered to the existing user base to lock them in, this is called expansion
Bundling: control	There is a strong need for monitoring and control as a result of the openness that is needed for the external complementors. Control must be an ongoing activity Tight control over the platform is required to ensure the health of the entire constellation
Governance for complementors	These complementors provide additional products or services. Because of the different complementors there needs to be governance and access rules for these complementors.
Constellation control	Tight control over the platform is required to ensure the health of the entire constellation In the design phase a high level of control is required. By having strong rules in this phase less monitoring will be required once the platform is up and running. There is a closed structure, so the health of the constellation can be maintained.
Access and interaction rules and regulations	There must be access rules and interaction regulations for users and complementors of the platform.
Critical mass	Achieve critical mass.
Networks effects	Leverage network effects
Monitoring and control (in the case of bundling)	There is a strong need for monitoring and control as a result of the openness that is needed for the external complementors. Control must be an ongoing activity
Different rules for different partners (in the case of bundling)	Because of the different complementors there needs to be governance and access rules for these complementors.
Control during design phase **(in the case of Constellation)**	In the design phase a high level of control is required. By having strong rules in this phase less monitoring will be required once the platform is up and running.
Access rules and regulations (in the case of Constellation)	There must be access rules and interaction regulations for user and complementors of the platform. Be selective about which partners to allow on the platform or all innovations are developed inhouse, so there is more control over the user experience.

Cumulative Tables

Below there are two tables which show in which strategy each dimension or critical success factor was mentioned and how often each dimension and critical success factor is mentioned. These tables are meant to support the process of choosing the most important factors to construct the framework with. They are however not leading. Relevance for the case studies and relevance for Tersof is the most important selection criterion, however, network effects for example are mentioned by 11 out of 15 strategies which does give an indication of the importance of network effects compared to some other aspects which are only mentioned by a single strategy.

Critical Success Factors Frequency Found in Literature Matrix

Critical Success Factors/ strategy in which they are mentioned	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	#
Agility in strategy/ dynamic strategy				x							x			x		3
Alignment and coevolution of architecture and governance								x								1
Attract users with specific users									x			x				2
Create partnerships/ attract developers	x		x	x	x			x			x		x			7
Attracting users/ user retention strategy	x	x				x		x	x			x		x		7
Autonomy/control (guaranteed integration) balance								x								1
Ambidexterity							x									1
Awareness of international differences											x					1
Awareness of the differences between platforms and traditional businesses								x								1
Customer and service oriented										x						1
Established before becoming a platform/ stand-alone value								x	x		x	x				4
Clear added value/ proposition/ be imbued in processes			x	x	x	x										4
Closed platform core, open complementary components							x	x								2
Compatibility/ multi-homing/ cost of switching on the right side							x					x		x		3
Connections between users and/or developers/ enable transactions/interactions	x				x				x	x	x		x			6
Counteract monopolies											x					1
Governance: control, rules and regulations															x	1
Create new opportunities			x													1
Create scenario's/ dedicated platform team						x										1
Create word-of-mouth			x													1
Creative orchestration of governance								x								1
Gain customer knowledge														x		1
Data governance/ utilize big data		x	x													2
Development rate at least equal to competitors								x								1
Easy development on platform/ easy to join/ usability			x								x					2
Ecosystem core		x														1
Enter new markets (together with partners)			x													1
Environment in which the platform is established				x												1
Platform expansion design											x				x	2
Fairness							x									1

Critical Success Factors/ strategy in which they are mentioned	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	#
Filtering/ low information cost/ search function					x						x					2
Foster and adopt or absorb external innovation/ emergent innovation			x	x				x								3
Future proof/ evolvability of architecture/ continuously evolve						x									x	2
Governance: alignment with platform architecture/ cheap and simple								x								1
Grow the entire market/ecosystem							x			x						2
High adoption speed/ quick impact					x											1
High quality offering			x													1
Incentives to join/use	x				x											2
Innovation ability													x			1
Integration of platform strategy with overall business strategy/ matching						x										1
Kick-off the platform with push strategy (quickly gain large number of users)									x							1
Learning style of searching and experimentation														x		1
Listen to users/ gather feedback/ foster discussion/ evaluate/ improve user experience/ learn by doing			x							x				x		3
lower transaction costs/reduce friction/efficiency					x						x		x			3
Supply to mass markets (economies of scale) and long tails (personalization)			x	x				x		x				x		5
Minimum requirements to ensure quality/ compatibility between components											x					1
Mirroring between architecture and governance								x								1
Modularization/ standardization			x					x		x						3
Multi-sidedness in the platform								x								
Network effects/ reach critical mass/ chicken-or-egg/ lock-in	x		x		x		x	x	x	x	x		x	x	x	11
Neutral marketplace											x					1
Non-financial subsidizing (incentives)							x		x							2
Nudge towards a just right option								x								1
Offer guidance			x													1
Openness											x					1
Potential user base				x												1
Precise, frozen and versatile interfaces								x								1
Pricing to attract users/lower risks/adjust to lifecycle phase/ market driven				x			x	x			x	x				5
Pull instead of push marketing								x								1
Relationship management			x													1
Reputational systems/ trust/ quality control/ security	x	x	x	x	x	x		x			x					8
Revenue streams from existing customer base							x									1
Scalability			x													1
Short break-even time					x											1
Simple, resilient, maintainable an evolvable architecture								x								1
Slow competitor response in market					x											1
Sponsoring/asymmetric pricing/ subsidizing			x				x	x			x					4
Use new technology			x													1

Strategy Dimensions Frequency Found in literature Matrix

Dimensions/ strategy in which they are mentioned	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	#
Price or quality differentiation							x									1
Advertising on the platform					x											1
Autonomy/control								x								1
Collective intelligence													x			1
Connectivity/connection/conversation													x			1
Contact to potential users											x					1
Control mechanisms								x								1
Create new or integrate with existing platform						x										1
Data availability		x														1
Decision rights partitioning								x								1
Driver of platform evolution								x								1
Ease of joining/ease of use	x		x													2
Ecosystem orchestration/ design															x	1
Envelopment			x				x									2
Evolvability															x	1
Expansion design approach															x	1
Governance			x		x					x				x	x	5
Horizontal/vertical integration			x													1
Independence											x					1
Inhouse development												x				1
Innovation source/style (internal/external)								x							x	2
Integration/multi-homing/ compatibility/switching costs/ lock-in			x				x				x	x		x	x	6
Internal/external platform support						x										1
Licensing structures								x								1
Modularization/ monolithic architecture			x					x								2
Monopoly effect counteraction strategy											x					1
Number of platform sides								x							x	2
Openness			x		x		x			x	x		x		x	7
Partnerships			x	x												2
Piggybacking							x		x							2
Platform function (transaction/data-centric)											x					1
Platform core								x								1
Point of exchange (physical/digital)					x											1
Pricing	x	x		x	x		x	x			x			x		8
Sponsoring/subsidizing (asymmetric pricing)			x				x	x			x	x			x	6
Proposition (offered value)				x	x						x			x	x	5
Reputational Systems/ security/ reviews/trust	x		x	x	x							x				5
Research method for breaching strategy														x		1
Revenue division								x								1
Scale			x													1
Social media integration			x													1
Specialization/ personalization/ variation in offerings			x	x										x		3

Dimensions/ strategy in which they are mentioned	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	#
Stand-alone value												x				1
Time (lifecycle, evolutionary phases)										x						1
Transaction costs											x					1
User attracting strategy/ chicken-or-egg		x					x		x		x			x	x	6

4. The Context of the Strategy Development: Tersof

Tersof is a software company located in Enschede and Amersfoort in the Netherlands and in Adelaide in Australia. With about forty employees they make a software product to enable change processes in organizations, from small organizations to multi-nationals.

During my time at Tersof one of the things I learned is that Tersof is not only enabling change but also changing themselves. Not only their product is evolving but also as a company they are changing, they are adapting to changes in the market and strengthening their competitive position. New people are joining the company, new customers are signing up and a new agile value proposition has been developed.

The product has over 80.000 users and the company strives to increase that number to 1.000.000 users in the coming years. Currently many steps are already being taken in the company to facilitate this growth. From the development of new apps which are in line with currently popular management styles such as agile and waterfall methodologies and development techniques, to the creation of a free trial environment and the translation of the software into other languages.

Another line of thought that has arisen with the future in mind is to optimize the platform-like nature of the Tersof software and to incorporate more platform thinking into future strategic decisions, in order to be able to achieve their goal of getting a 1.000.000 users. A platform is all about bringing people or organizations together, so they can benefit from each other. In the case of Tersof added value can be created for customers by connecting them to each other, and in the future maybe even to external developers of applications which will integrate seamlessly with the Tersof software.

When I started writing this thesis the question was very much about how can Tersof become a platform. But shortly after starting the thesis the realization came that Tersof is in fact already a platform business. So the question is not so much how to become a platform but more about exploring how they can exploit the platform they have already created better. These concepts of exploring and exploiting fit in perfectly with the ambidexterity perspective strategy [26] that was mentioned in Platform Strategy 7.

The Added Value of the Software

The goal of the software from Tersof is to enable change. It can assist companies in implementing change at higher speeds and with lasting results. The software is based around the think-do-learn concept and for each of these steps there are several apps available. The applications are divided into four groups: “Strategic Control“, Scaled Agile, Project Portfolio Management and “Insights“.

The Strategic Control applications are “Strategy“, and “Invest“. The goals of these applications are to create and maintain strategic directions and objectives and to remain in control of the strategic organizational goals.

The Scaled Agile apps are “Agile Portfolio”, Agile Program“, and “Agile Team“. The goals of these applications are to support the use of agile methodologies not just in a single team but in the entire organization.

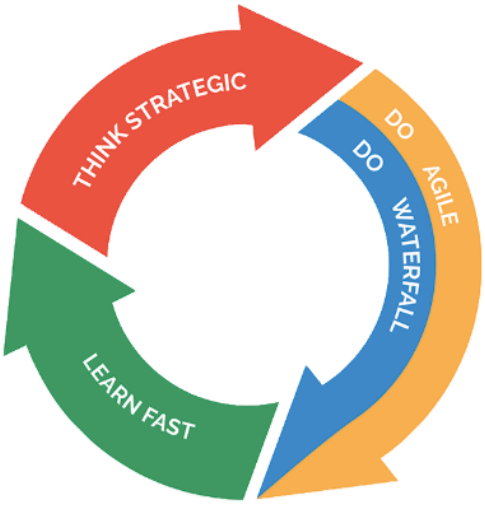


Figure 32: Enabling Change

The Project Portfolio Management apps are “Portfolio” and “Program and Project” These tools support portfolio design using best practices and the waterfall model.
The Insights app is appropriately called Insights and the goal of this application is to learn from what has been done through monitoring the effect of strategic decisions and by providing costs and benefits information.
Finally there is also a selection of support apps offered: “Hours“, “Resources“, “BI Connectivity“, “Import Export“, and the “Admin Center“. So there are already quite a few change enabling applications offered by Tersof, in the future this proposition could be extended with applications which are made by external partners and which seamlessly integrate with the applications made by Tersof. Among the customers of Tersof there are several organizations with particular needs, ranging from municipalities to multi-nationals. Because there is such a variety in customers it could be of great added value to those customers to learn from each other and to connect to each other and in the future to even buy specialized apps especially for their business type and fitting with their specific needs which have been created by external partners of Tersof.

Core Values of the Tersof Value Proposition

Tersof has defined several core values for their software which characterize their offerings and provide guidance for future offerings. These core values are very important to Tersof and a big selling point for their customers. It is therefore important that when new platform abilities are introduced they will take these core values into account and live up to them.

1. Easy to use. Each application has a clear role to fulfil in the larger process and meets the needs of the customer. The applications are intuitively designed and will work on any device.
2. Quick to implement. The tool is quick and easy to implement and easy to maintain.
3. Adaptability and personalization. The current processes of an organization can be integrated into the Tersof software without the need to write any extra code. The software is easily adapted to fit the customer context in which it will be used.
4. Makes using best practices easy. The software supports different portfolio and project management styles such as SAFe, Prince2, Stage-Gate and AgilePM. The enablement and support of these popular management styles are a huge selling point of the software.
5. Easy integration with common tools. The software can be integrated with software that is already in use in organizations, such as Jira, SAP and Microsoft products. This compatibility with current or legacy systems lowers the barrier for potential customers to start using the software and with the seamless integration added value for the customer is created.

The Users of the Tersof Software

Tersof support several different types of customers: ranging from municipalities to large multinationals, government agencies, financial institutions and healthcare institutions. These different types of customers have their own requirements and might need applications which are specific to their context. In such cases the platform structure is ideal. Customers can pick those apps which fit their business and skip those for which they have no need. Developing apps for all different niche markets is very costly and time consuming and therefore outsourcing this to external development partners could be an ideal solution. This way Tersof can offer more applications on their platform, better meet the needs of more niche markets and share the risks and benefits with partners. Offering applications from external partners

of course does require that Tersof must ensure the applications meet their quality standards and create a seamless user experience for their customers.

Multi-channeling Strategy

Tersof uses a multi-channeling strategy to reach their (potential) customers. The company has a website with a lot of information and factsheets. They regularly organize seminars to promote their software and meet their (potential) customers. They visit expositions and network events to get in touch with potential customers. Finally they organize training courses for their users to teach them more about the software, so they can get the most out of it. All these different points of interaction contribute to the retention of their current customers and the attraction of new customers. A platform where customers can connect to each other and to the company could be another addition to this multi-channel strategy.

Experiment – Embed – Extend Strategy

Tersof believes in the experiment, embed, extent strategy. Users should first experiment with the software in a single program or business unit, then embed the success in the organization and finally extend by spreading the software and processes over the entire organization. The same kind of staged approach is desirable for the platform strategy: starting small, seeing if and how it works and gradually extending the platform to offer more and different types of value to a more diverse set of customers.

Legacy

It can be argued that it is easier for a new company to start a disrupting platform because they have no legacy systems or processes that might make starting a platform more difficult [17]. A legacy are the existing systems, structures and organizational processes that are already in place and will influence any future endeavors [17]. Historical context has to be overcome to implement a new strategy [14]. However, it can also be argued that for Tersof the customer base they already have, is very valuable as a platform needs users to be attractive to other companies to create offerings for the platform. Several of the studied platform strategies even suggested that is the best option when starting a platform to first offer stand-alone value and collect a large user-base based on your stand-alone value before evolving into a platform [16], [26]–[28].

Legacy at Tersof

During my time at Tersof we had a party because Tersof existed 20 years in April 2018. There were balloons, streamers, pie, and also pizza and beer for lunch - it is a software company after all. Because the company has already existed for 20 years there is a legacy to take into account. Part of this legacy is a technical legacy: the old computers and applications will need to work together with any new platform or applications that are developed. This is something that Tersof is currently working on. However, in this thesis the focus is not on the detailed technical workings of the Tersof platform, but more on the high-level strategy. Another, very significant, part of the legacy of twenty years of Tersof is the existing customer base. Tersof already has a customer base consisting of 80.000 users, all of which they want to retain when the new platform capabilities are introduced. This existing user base is a great asset, for many starting platforms getting users on the platform is the most difficult thing to achieve, many of the studied strategies talk about overcoming the chicken-or-egg problem of how to attract both partners and users when either one only wants to join once the other side has joined [2], [16], [18], [24], [27]–[29], and Tersof has already accomplished this. However, the risk is that people don't like change and might

be resistant towards the new value proposition. People also prefer fairness [16], [26]. The existing customers have existing contracts and pricing agreements. If for the new platform an open pricing policy is adopted and maybe also a low pricing policy to attract more new customers, the existing customers will dislike paying more for the same software. So plans should be made to make sure that the pricing feels fair to the existing customers [26]. A way to mitigate the resistance against change could be to make the changes slowly and incrementally as compared to a big bang approach where everything is changed at once.

Current Integrations

Currently there are some integrations already between Tersof software and external software. Integration with other applications is managed through REST API or file-based integration. Some integrations already have been proven such as with ERP, document management and BI systems such as SAP, Oracle, Exact, Infor, MS SharePoint, IBM Cognos, Jira and Microsoft TFS. If the transition to a platform which offers applications made by external partners is made, then all these applications should seamlessly integrate with the Tersof applications and also with each other and currently existing integrations.

Desires and Wishes/ Requirements

Several desires and wishes have already been construed by the strategic management of the company. For example the desire to gradually expand the current offering into a more platform-style offering. Some of the studied platform strategies also suggest gradually expanding and considering the market in which Tersof operates this seems a suitable approach. Many of the studied strategies also advised to not start out as a platform but to become an established producer first. Tersof already exists for 20 years and the 80.000 users are clear evidence that the company has excellent stand-alone value to offer.

The Most Important Guidelines for Tersof

Grow Gradually

The platform concept should grow gradually. The first step is to allow customers to get into contact with each other and share knowledge and best practices about using the software. The second step is the sharing of configurations of the software. The third step is to consider offering applications made by third parties.

The Five Core Values

Tersof has five core values at the heart of their value proposition. These should be upheld in the platform and external applications.

1.000.000 Users Goal

Tersof has set a goal to grow to 1.000.000 users, so any platform activities should take this desired number into account.

Fairness to Current Customers

Tersof wishes to maintain their current customers, so they should be offered equal value for a fair price when compared to the new customers or also benefit from newly developed pricing policies.

Intuitive Design with High Usability

On July 9th there was an "increment kick-off meeting", during which many employees mentioned the importance of usability of the software. To attract new users who will make a purchasing decision based on a free trial, the software should be immediately easy to use. The consultancy and support branch of Tersof has a goal to go to a 100% self-service in the future. To support this the software platform should have an intuitive design and high usability.

Low Threshold for New Customers

To grow to 1.000.000 users the software should be easily accessible to potential users. Potential users should also be able to start working with the software without the need for a consultant or extensive training.

Compliance with ISO 27001

The software currently is compliant with ISO 27001 and it is important that if in the future applications made by third parties are offered on the platform, that these applications will uphold this standard to maintain security and availability for Tersof software users.

5. The Framework

The platform strategy framework has been developed based on the literature study and the context of Tersof. The framework is presented in this chapter and will be validated with the use of case studies in the next chapter.

The Meta Model of the Framework

After studying the platform strategies the stage of formalizing the framework began. Initially the expectation was to derive strategy dimensions from the literature. However, it was soon discovered that many strategies where more focused on critical success factors and why those factors are so critical and mentioned very little on how to achieve compliance with these critical success factors. Because of this the platform strategy framework incorporates both strategic choice dimensions and critical success factors for platforms and platform strategies.

From the literature it was discovered that in the end the ultimate goal of any commercial platform is always to have users, and most of the time also partners, join the platform, as in the end users and partners are the only way to generate revenue from the platform. Without users and partners there is no platform and no viable business. In the end no matter from what angle platform strategy was approached, the final goal of every strategy is to convince users and partner to join the platform and thereby facilitate the growth of the ecosystem (platform + users + partners).

So ecosystem growth is the end goal of a platform. To achieve this certain conditions need to be met, these are the critical success factors. The critical success factors can in turn be achieved by making the right strategic choices in the different strategy dimensions.

So, a choice is made, which has a certain expected result which in turn will contribute to the goal. As shown in the diagram below: a specific choice for a certain dimension will contribute to a certain critical success factor which must be present in order for the ecosystem to grow.

Figure 33: Choice, Result, Goal

The Framework

So looking at this meta model it starts with dimensions in which the strategic choices are made. In the studying of the existing platform literature it became clear that there are many dimensions to choose from. However, some are more relevant than others, and many are actually the same dimensions but by different names. From all the dimensions present in the literature it was narrowed down to six core dimensions. These six dimensions encompass the larger part of the most mentioned dimensions from the literature and will contribute to most of the critical success factors. The framework model containing these six dimensions is shown on the next page.

The six dimensions are: user attraction, evolution, pricing, trust, usability and governance. By making the right choices in each of these dimensions users and partners will join the platform. The type of users and partners you want to join the platform can in turn determine which decisions you should make. An open source looking to attract many partners should make different choices for these dimensions than a company that wishes to attract only a few partners. A platform may want to offer the highest possible quality to attract users that are looking for quality or offer a lower price for a lower quality product just to attract as many users as possible.
The model can be used in both directions. A company can first determine which partners and users they would like to attract and make strategic choices based on that or a company may make strategic choices first which will then lead to certain users and partners being attracted to the platform. Each of these dimensions will be discussed in further detail below.

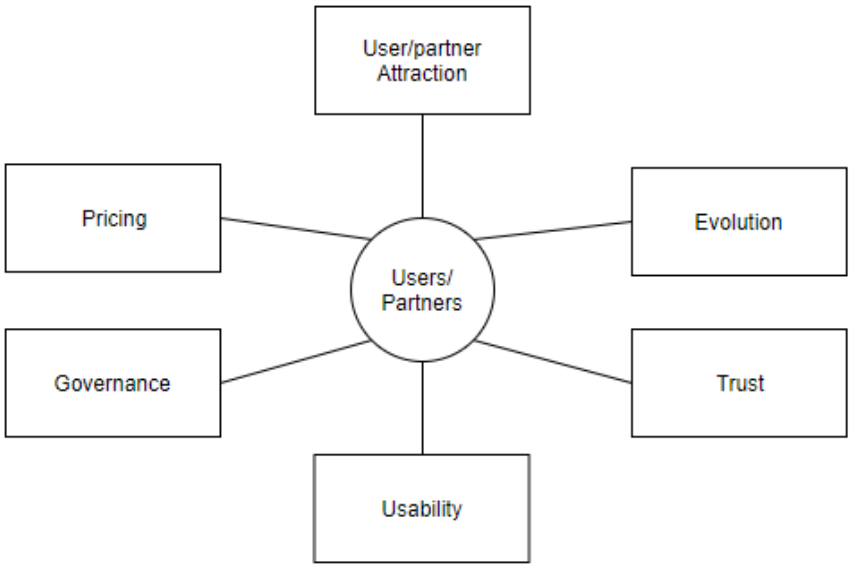


Figure 34: The platform strategy dimensions in the framework.

This framework will be used on the case studies to validate the framework. The results of the validation and the final revised version of the framework can be found in the chapter “Revised Platform Strategy Framework” after the case studies.

Dimension: Trust

The concepts related to this dimension are mentioned by 9 of the 15 studied platform strategies [2], [16], [17], [20], [23]–[25], [28]. Users and potential users need to trust that the platform is of high quality and secure and safe to use [24]. You have to build credibility with your customers [29]. Tersof for example complies to ISO 27001 which means they guarantee that they are committed to security and that user data is safe. Below the conceptual model of the trust dimension is shown. There needs to be trust from users that the platform ensures quality control, which can be done through reputational systems and reviews and the users must trust that the platform has adequate cyber security measures in place. Guidelines, governance, rules, policies and review systems can ensure the reliability of the platform [16], [24].

Security

A platform can offer security to potential users, especially in the case of applications made by third parties. The platform layer in between the buyer and suppliers offers additional security and trust [17]. For example, personally I would not buy things online from small unknown companies, however when that same company sells their product through an established platform such as bol.com. I have no doubts in buying from them through bol.com. I know bol.com is a legitimate organization that will handle it properly should anything go wrong. Compliance to ISO standards will also show (potential) users that the company is committed to cyber security. Trust must be embedded at the heart of the platform and cybersecurity is a large part of that [20]. A commitment to cyber security can be a way for a platform to differentiate [20].

Quality Control

Quality control is about protecting the user experience. The software should live up to the quality standards of the users. This can be ensured through quality control mechanisms. Reputational systems and reviews are examples of quality control mechanisms [2]. Having strict guidelines for app developers will also help [24]. Governance is also a quality control mechanism which will be discussed in the subchapter about that dimension. Minimum requirements that partners have to adhere to can also ensure quality complements on the platform [28].

Reputational Systems and Reviews

A way to enhance trust is to allow users of the platform to rate the applications that are available on the platform [2], [18], [23], [24]. By having users write reviews of the available applications potential users of the application have more trust that the application will fit their needs or search for another app which suits them better. Rating systems are also a way to automate quality control [18]. Bad reviews or constructive reviews can be a trigger for the supplier or developer of an app to improve that app based on the received feedback [18]. These systems are a way to use the wisdom of crowds [18].
Reviews and voting systems can also give potential users information about the quality of the platform. Users need to trust that they are getting quality and security before they will make a purchasing decision and populate the software with their company data.
One study claimed that platform are viewed as more reliable when they are neutral [28], meaning that the platform is independent and not integrated with one of the platform sides.

Credibility through Partnerships

By entering into a partnership with a well-known company or by inviting marquee users the platform can also gain credibility [29]. The effect is even larger when these marquee users are exclusive to the platform [29]. Tersof is an official Microsoft partner and certain government agencies use Tersof's software: these facts will convince potential customers that Tersof is a trustworthy, legitimate business.

Choice	CSF and goal contribution
Offer reputational systems such as reviews and ratings	Ensures trust and makes the platform seem more reliable. This will decrease doubt that potential customers might have when considering signing up for the platform or buying applications from partners on the platform. Bad reviews will trigger improvements. The platform will remain of high quality for current customers and give potential customers a quality indication which they can use in their decision-making process.
Ensure compliance to ISO	This offers security to potential customers and reflects a certain level of security and reliability of the platform. It is a strong selling point.

Choice	CSF and goal contribution
Set minimum requirements for developers	Minimum requirements that partners have to adhere to can also ensure quality complements on the platform and guarantees a high-quality user experience.
Neutrality	Neutrality can give the platform credibility.
Create strategic partnerships/ Attract marquee users	Be entering into a partnership with a well-known company or marquee users the platform can gain credibility.
Guidelines, rules, policies for external app developers	Having these in place will increase trust in the platform among (potential) users.
Control mechanism: governance	Choices in governance structure can ensure quality control.
Commitment to cyber security	A commitment to cyber security can be a way for a platform to differentiate

Dimension: Usability

This dimension was chosen because of its importance in the specific context of Tersof. Only two of the studied platform strategies mention usability and they do not really go into detail about it.
Strategy 11 Characteristics and Success Factors of Digital Platforms mentions that the platform owner is responsible for the usability of the platform [28]. Customer surveys can be used to monitor the usability and improve it [28]. Strategy 3 Kreijveld's 10 Basic Rules mentions the importance of making design and development easy for developers and listening to wishes and complaints [2]. The platform should also be made easy to use for users[2]. Easy design and development will lead to low costs for developers [2].
Apart from this very little is said about usability in the studied platform strategies. However, because it is so vitally important for the value proposition of Tersof it has been chosen as a dimension.
Self-service is part of the strategy of Tersof to achieve their goal of 1.000.000 users. The software and the process of buying and configuring it should be so easy that potential customers can start working with it without the need for help from a consultant or from customer support employees. During conversations with several employees in the company it was often repeated that they saw usability both as one of the biggest challenges and opportunities for Tersof. It is definitely an important critical success factor that the software should have high usability to support the growth goals of the company.
If the software has high usability it is easier to use for current and potential (or free trial) customers, new customers will be more likely to sign up for a paid plan and current users are more likely to continue to use the software. If the software is difficult or if it takes long to learn how to use the software potential customers will decide not to become a customer. The costs of offering personal consultancy and intense customer support will be extremely costly for 1.000.000 users and therefore the high usability which will lead to less need for consultancy service and customer support services contributes to the long terms goals of the company.
With low usability there is the risk that the software will only attract users who can and are willing to spend time to learn how to work with anything, or with a more colloquial term, nerds. Low usability therefore greatly limits the size of the potential user base. Low usability can be mitigated with training materials. A guided tour can make it easy to learn how to use the software with little effort required from the user. Other training documentation or a forum where users can help each other can also be used as mitigation for low usability.
Unfortunately both in the 15 studied strategies and many other sources about platforms which I studied very little guidance was given on what exactly platform usability entails, how this can be achieved, monitored and tested. What 'usability' in the context of 'platforms' exactly constitutes could be an area for further study.

Choice	CSF and goal contribution
High usability	It is easy for users and partners to join the platform and use the platform. Potential customers are more likely to become to customers and current customers are more likely to remain a customer.
Low usability	No contribution to the goal

Dimension: User Attraction Strategy

This dimension encompasses several concepts which are mentioned very frequently in the different platform strategies. It is a bit of an odd one compared to the other dimensions, as the other dimensions all have as goal to attract users as well, it is however worth considering it separately because the network effects are so important for platforms.

The user/partner attraction and retention process goes through several phases as can be seen in the model below. These steps will be discussed further below.

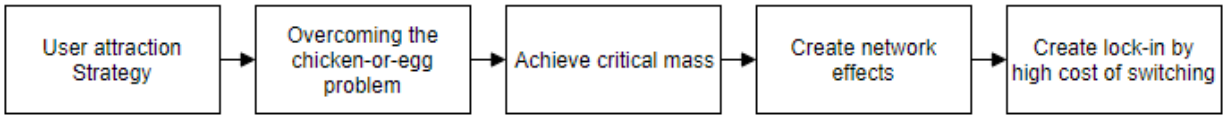


Figure 35: User attraction Strategy

User Attraction Strategy

Users can be attracted by choices made for the other dimensions, because of specific user attraction strategies or because of network effects. Low prices can attract users and openness can attract developers. Strategy number 9: 8 Ways to Launch a Successful Digital Platform discusses 8 ways to attract users to your platform with non-financial incentives [27]. For example entering into a strategic alliance with a marquee user will attract certain other users or user groups to the platform [27], [33]. The risk can be however that such a user will usurp the governance over the platform [24]. Network effects finally form the virtuous cycle in which as more users or partners join the platform more users and partners will want to join the platform.

Chicken-or-Egg Problem

The chicken-or-egg problem mentioned in many of the strategies is about how to get users and partners to join the platform when there are not many users and partners on the platform yet [33]. For example, if there are only 10 people on Facebook it is not very interesting for other users to join the platform and for advertising partners the platform also does not have value yet. A way to overcome this is to create an alliance with one side of the platform to have a starting user base to stimulate network effects [33]. Only when and if the chicken-or-egg barrier has been overcome can the platform start to achieve critical mass.

Achieve Critical Mass

Critical mass is the amount of users and partners a platform needs before network effects will start to take effect [16]. A platform can only achieve critical mass if there is a viable market for the platform [16].

Create Network Effects

A topic covered in almost every blog, article or book about platforms is the so called ‘network effect’. The network effect basically says that once you start growing you will grow exponentially. When a platform

has more users it becomes more interesting for developers to develop applications for the platform and the number of developers on the platform will grow [2]. When more developers are developing applications, the platform becomes more interesting for the users, so then the number of users will grow again [2]. When there are more applications available on the platform, the platform will have more potential value for potential users [2]. This is the start of a virtuous cycle in which growth of users stimulates growth of developers and the other way around continuously [2]. Users and developers stimulate the demand for each other [2]. The final result of this virtuous cycle can create a 'winner-takes-it-all', effect [2], [13], [22]. There is a winner and there may be a few small niche players. The network effect makes platforms powerful [23]. The network effect makes it difficult to compete with a platform once it has captured most of the market [17]. Sometimes there are two or a few winners such as in the smartphone market. Both Apple and Android are winners in the smartphone platform market, but other players, even those who were once very big, such as Windows Phone and Blackberry have almost completely disappeared, not because their product was inferior, but they had a much smaller network [13].

The Value of a Platform

The value of a platform is determined by the size of its user base and how active those users are [2]. Because of the network effect, use attracts more users. The community of users and developers determines the value of a platform [2].

Direct and Indirect Network effects

Direct and indirect network effects generate platform power and results in a winner-takes-it-all effect which confers enormous power to the platform owner [9]. Direct network effects or same side network effects: the platform has greater value when more users join on the same side of the platform [16], [33]. Indirect network effect also called cross-group or cross-side network effects: having more users on one side will increase the value of the platform for the other side [14], [16], [33].

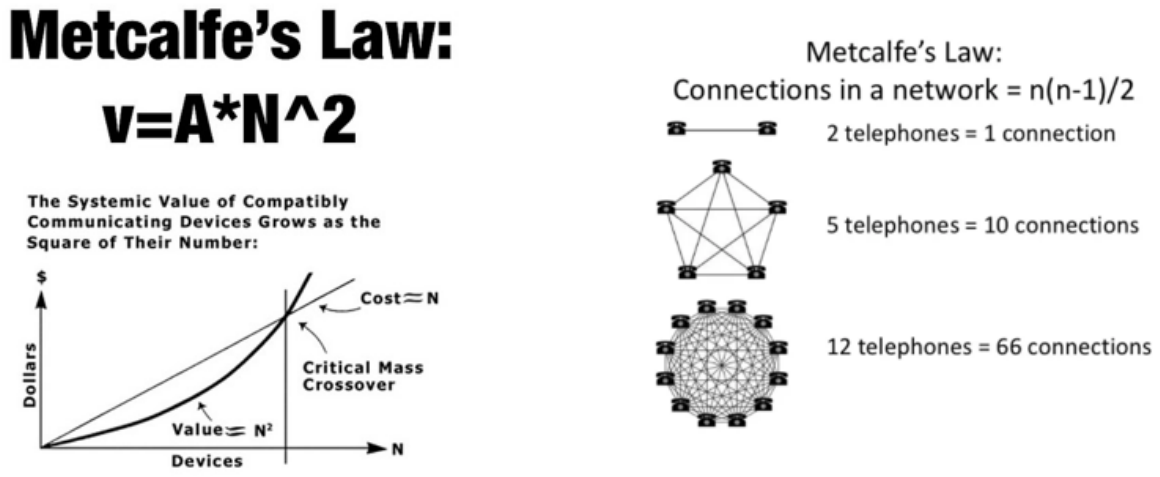


Figure 36: Metcalfe's Law for the Network Effect

Market leadership and Network Effects

The network effect in platforms is another good reason to strive to become a market leader early on. Only the platform with the largest network will survive and every additional user or partners makes the platform more valuable, this is Metcalfe's law [13], [16]

. The value increase is not linear but exponential [16]. A platform is interesting for developers when it has a large user base and therefore for the developer a large potential customer base [13]. With more active developers the platform has more to offer to potential users [13].
A platform only has value by virtue of the players on and users of that platform. A platform with many users is more attractive for players, they will have a bigger market to which they can sell their products. A platform with many partners who offer many products to run on that platform make the platform more interesting for users because it offers more possible applications. [16]

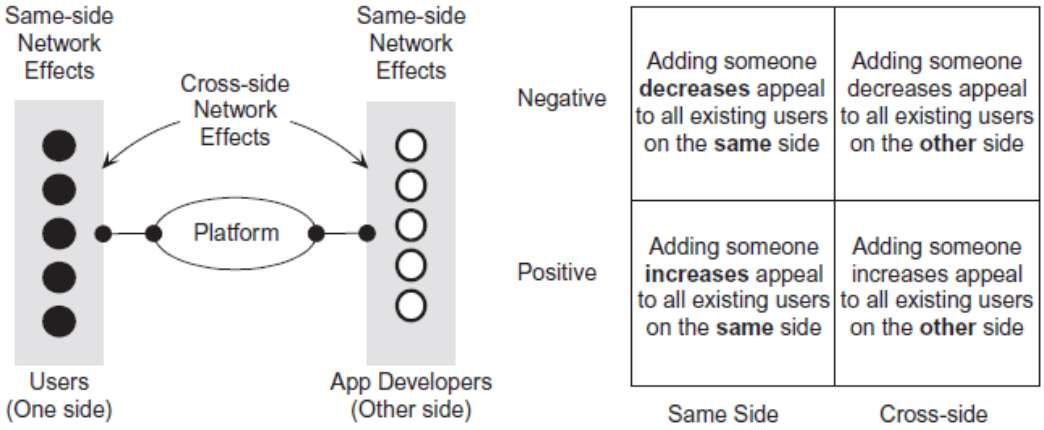


Figure 37: Same side and cross side network effects

Lock-in Effect

Bundling products and services related to a platform creates a dependency on that platform for the user and thereby a lock-in effect [2]. Transferring to another platform is no longer an interesting option or made is nearly impossible [2]. A platform provider should make the cost of switching high and make sure it remains high [2].
Coercive or value-driven lock-in [16]. A coercive lock-in can be broken out of through middleware, adapters or protocol translators [16].

Cost of Switching

The lock-in effect is the result of a high cost of switching [16]. The cost of switching is how much it costs for the user or partner to switch to another platform. An alternative lock-in strategy which is much harder to achieve is to be increasingly more valuable to users and partners than your competition is [16].
Some argue that a platform does not become the winner by offering the best or having the best idea but by being one of the first. Once companies rely on a certain platform for their needs they are not likely to switch again soon. The longer a system has been used the higher the cost and effort required to switch [1]. Switching to new software will also mean that all users need to learn how to use this new system, so there is an education cost to switching [1]. However, being a market leader and having a large network does not mean that you have no more competition. Lotus once dominated the spreadsheet market with their “1-2-3”, Netscape was a leader in the browser market and WordStar was the leading solution for text editing. Then came Microsoft. By bundling their spreadsheet, text editor and browser with Windows they were able to disrupt the market and become market leader [1].
Switching costs is going from Windows Live mail to Gmail and thereby losing all your previous emails and contacts.

Lock-in by Single-homing

A platform can make a choice whether or not to allow users and partners to multi-home or how easy it is to multi-home. Multi-homing costs are not the same as switching costs, but high multi-homing costs can cause customers to stick to your platform [16]. For example multi-homing means subscribing to Netflix and to Videoland, this means paying twice. Multi-homing is when a user or partner simultaneously commits to multiple rival platforms and therefore participates in several ecosystems at the same time [16]. This also means that the platform owner has to share his customer base with other platforms [26]. Multi-homing is usually allowed on the side with small network effects [26]. Consider Netflix, it makes sense to want to offer exclusive content but not restrict users from using other streaming service providers. Single homing can be forced by incompatibility and exclusivity contracts [16]. Developers are very likely to multi-home when there is no clear winner yet in the market so they do not get stranded on a losing platform [16]. Multi-homing is deterred by the cost of maintaining several platforms affiliations, with higher costs users and developers are more likely to single-home [16]. Modularization on the other hand makes multi-homing easier for developers [16]. It is more likely that they can reuse components which is an incentive to join [16].

Winner-takes-it-all or Pie Splitting

This is about whether there will be a single winning platform in the market or whether the market will be divided over many different platforms [16]. In a pie-splitting market there will be less network effects and users and partners are more likely to multi-home.

Winner-takes-it-all	Pie splitting
High costs for multi-homing for users	Easy and cheap to multi-home for both users and developers or only for developers
Positive and strong cross-side network effects	Weak network effects, low number of users and/or complementarities
Low demand for a hard-to-copy product	High demand for differentiating features
	In this type of market it is better to specialize in a specific market segment

Choice	CSF and goal contribution
High cost of switching for customers	With a high cost of switching users and partners are more likely to stay on the platform once they have joined the platform
Low cost of switching for potential users	Makes it more likely that potential users will join the platform as it is easier/cheaper to switch to the platform because of the low switching costs
Allowing/supporting multi-homing, compatibility with legacy/ competing platforms	You might have to share your customer base with your competitors, but you can also share in the customer base of your competitors. So this choice is most advisable when you wish to connect to customers which are currently only involved on a competing platform
User getting method to overcome chicken-or-egg problem	There are several methods to overcome the chicken-or-egg problem, differentiating on quality or price or by using the other mentioned user attraction strategies such as marquee users, producer-evangelism strategy, innovations or offering stand-alone value first.
Aiming for same or cross side network effects	Network effects will cause the number of users to grow exponentially such how to create these effects should a be at the core of any platform strategy in order to 'win' in the market
Sponsoring/subsidizing to get network effects started	To get enough critical mass to create network effects potential users of the platform can be sponsored as an incentive to join the platform. This can be financial or non-financial

Dimension: Pricing

One of the most important dimensions, described in 10 out of 15 platform strategies, is pricing [2], [16], [17], [20], [23], [24], [26], [28], [29], [31]. Both the pricing strategy for platform developers and the pricing strategy for platform users will hugely influence how many developers and users will join the platform.

Pricing is a simple variable in terms of possible values, any monetary value is possible. The variable is much less simple in terms of what different price settings might achieve. The platform could be offered for free to attract a large number of users, but that also creates high costs to supply the platform to all those users and you could let developers offer their applications for free and take no cut of their profit, but then how will you finance the development and maintenance costs of your platform? On the other hand, if you make the prices too high there might not be anyone who wants to join your platform. Pricing is also often driven by the amount of competition there is in the industry [26]. Is there a monopoly or many competitors? Pricing strategy can be split into pricing policies for users of the platform and pricing policies for developers of the platform.

The Goal of Pricing Strategy

Simply put, the goal of the pricing strategy is extract revenue from the platform in order to make the platform viable. Depending on the chosen pricing strategy different types and different amounts of users and developers may be attracted to the platform.

There are three major pricing decisions to be made:

Market entry pricing strategy	Skimming or penetration	With a penetration strategy a low price is charged to join the platform. With a skimming strategy a high price is charged. With a penetration strategy there is often a low price charged in the beginning to attract a large number of users and prices are then raised over time. A skimming strategy is usually used for prestige products who at first only are sold to the segment of the market that can afford it, once the top segment has bought the product for a high price the prices are lowered so lower market segments can also afford the product.
Pricing model	Variable or fixed fees	Usage based pricing will lower the risk for potential customers as they will only have to pay for actual usage. For the company it creates a continuous stream of revenue. A fixed fee can offer more certainty to the company and the customer as there is only one monetary transaction, no matter how much or how long the product is used.
Sponsoring	Stimulate users and/or developers	Offering the platform for free to users can greatly increase the number of users on the platform, but it does mean that the costs have to be covered by partners (sponsors) of the platform or the platform will run at a loss.

Achieving Critical Mass

Price has a strong influence on the number of users of the platform. If the prices are very high few people or organizations will want to use the platform, if the prices are low this can attract new customers. Critical mass is the number of users and developers that the platform needs to be viable. It is important critical mass is achieved soon after the launch of the platform to survive [26]. So as a platform first consider the user base you already have and how that user base can be a source of potential value for the platform [26].

This is why an already existing user base is a great asset for any new platform. This existing user base should therefore be maintained and not scared away. One way to maintain the existing user base is making sure that the pricing of the platform feels fair to them. If the prices of the new platform are much lower than what they are currently paying this will create negative feelings among the existing customers. If there is a large price difference, then existing customers who might have long term contracts should be offered the same services at the same price as new platform users. If there is an existing customer base the pricing strategy should be designed in such a way that revenue can also be extracted from this existing customer base [26].

Exploration and Exploitation

Achieving ambidexterity in pricing can be done by temporal and domain separation [26]. Following the ambidexterity theory it can be advisable to first explore the existing user base and then exploit the potential value of that user base [26]. Ambidexterity in pricing is the easiest to achieve through temporal separation. Throughout the platform lifecycle the prices can be changed to support the current phase of the platform. Ambidexterity can also be achieved through domain separation [26]. One side of the platform can be exploited while the other is explored [26].
Explore the side that is more valued by another side [26]. Exploit the side that is more dependent on the other side [26].
For Tersof this could mean to exploit the user side, they already have these users. Explore the partner side, the third-party app developers and such, those are new and unsure, so explore them. As for temporal separation a starting discount for example could be used.

Sponsoring of the Platform

Pricing in platforms is made complex by the interdependencies among participants and the lifecycle of platform ecosystems. The price can determine the size of the customer base, one of the most critical resources for a platform. Pricing can be used to stimulate a virtuous cycle which can lead to a large market share. Pricing is also the reason many platforms have failed to create a large user base, as with a too high price users will not want to join. Pricing decisions can also lead to exceptional market share through virtuous cycles. Keeping the price artificially low on the side of the market can stimulate getting more participants from that side of the market, taking advantage of asymmetric cross-network effects. This is subsidizing [26], [33]. Subsidizing of a specific platform side can also be done by giving information away for free, granting privileged access to certain aspects of the platform or by giving technical support.
Studying the cross-network effect and determining which side of the market needs to be subsidized is critical for the success of the platform. This can be determined by own-price and cross-price elasticity of demand, quality sensitivity, multi-homing, and substitutable or complementary relationships between applications.
Determinants which side to sponsor: quality sensitivity, multi homing, substitutable or complementary relationships between applications, own-price and cross-price elasticity of demand [26].

Pricing for Platform Partners

There are several pricing choices to be made for the partners of the platform. First choice is whether you wish to extract revenue from your partners or only want to use them as a way to make the platform more attractive for your (potential) users. If you wish to extract revenue from partners on your platform there are many possible pricing strategies to consider. A fee can be charged to partner for offering their application on your platform regardless of how many times the application is sold. A percentage of each app sale can go to the platform owner or a fixed amount for each app sold. When charging the partner for each app sale a scale can be used, meaning that of the first app sold a different amount goes to the platform owner than for example app sale 2000. A maximum amount a partner may charge for their app can be determined or a minimum amount if it is not desirable for the platform that apps are available for free.

Differentiation: An Alternative to user attraction through Pricing

Differentiation is an alternative strategy to pricing [26]. The aim of differentiation strategy is to provide a unique ecosystem with high quality that will attract users to adopt it. Latecomer disadvantage may be overcome by a quality differentiation strategy as that will challenge an incumbent's dominant position [26]. Differentiation can be achieved on the platform itself or on the ecosystem side [26]. With differentiation the value offered to agents is enhanced by improving the intrinsic characteristics of the platform. It is about utilizing investments to improve the main features of the platform and offer a high-quality product. A marquee user can also differentiate the platform. If charging a low price to differentiate the platform from competitors is not desirable, then differentiation on the other dimensions is the best strategy.

Choice	CSF and Goal Contribution
Penetration (low price) strategy	By choosing to set a low price the platform will be within reach for more different organizations thereby increasing the potential user base.
Skimming (high price) strategy	Charging a higher price will lead to more profit per user and gives the users a sense of exclusivity.
Fixed fees	For both the platform owner and user there is more certainty. There is only a one-time transaction, how much the platform costs for the or how much the platform owner earns from the users is a fixed number independent of how much or how long the platform is used.
Variable fees	This option offers lower risk for potential users as they only have to pay for their actual usage therefore the barrier to join the platform is lower. The risk for the platform owner is that they may make less money if the customers leave the platform again shortly after joining compared to a fixed price. However, there is also a major benefit that a continuous stream of revenue is created and if users stay on the platform for a long time than the total revenue earned from that user is likely to be higher compared to a fixed price.
Sponsoring/Advertising	This asymmetric strategy is most suitable when one party is much more interested in the other.
Fair pricing	Pricing fair for existing customers so you do not drive them away
Pricing evolution	Change prices as the platform changes in maturity level

Dimension: Governance

The large number of participants and transactions between participants make governance of a platform business different from governance in traditional businesses [20]. Governance encompasses the dimensions of openness and distribution of decision rights and is mentioned in 9 out of 15 studied platform strategies [2], [16]–[18], [26], [28], [30]–[32]. Governance issues such as intellectual property, data ownership, management of the open platform ecosystem and licensing models are relevant for platform businesses [20]. Another key governance element is who has decision rights over the platform [16]. Governance is therefore the balancing game between giving freedom and autonomy to partners of the platform and remaining in control enough to ensure seamless integration between all platform components [16]. The rules in a platform dictate the competitive nature of the environment and the innovation dynamics [2]. A platform can be viewed as a regulatory structure that sets rules and parameters for social and economic activity [9]. A platform is a regulatory structure and thus a governance system with rules, regulations and policies [9].

Orchestration

Tiwana presents a nice metaphor that a platform ecosystem is like a symphony where the platform owner is the conductor and the developers are the musicians who each contribute a part to the music [16]. The conductor leads and synchronizes the musicians and helps them to create a collective performance. The conductor is an orchestrator. Musicians chose to follow the conductor or can choose to leave for another orchestra. The director cannot play each instrument as good as the individual musicians and they need

each other to achieve the desired result. Each musician has an individual contribution which is presented as a whole to the consumers. A platform should be run by orchestration and not by control [16].

Governance: Openness

The first sub dimension of governance is openness. Openness is a dimension that ranges from completely open to completely closed [34]. Open and closed here refer to how easy it is for partners to join the platform and once they have joined the platform how much freedom they have in creating applications for the platform. The openness of a platform is therefore determined by the participation rules of the platform [26].

By varying how open or closed a platform is both the number of participants and the quality of the platform will be influenced [26]. Platform openness can also be defined as to what degree developers need permission to have access to the platform, build on the platform and offer what they have created to the users of the platform [26]. Even a very open platform can be closed at its core. There can be very strict rules about the API, standardization and other rules and regulations while being open in other ways [2]. Modularization in architecture can be way to open up only those components which partners need while keeping other components such as the platform core closed [35].

Closed Platform

With a closed platform the provider has more control over the platform [2]. The owner of the platform can charge higher prices to join the platform, set more strict rules to ensure quality and keep away any complementarities or partners that might interfere with their product or business [2]. For example by not allowing partners onto the platform that are in direct competition with the existing partners on the platform makes joining the platform much more lucrative for potential partners [2].

Open Platform

An open platform is easier to join and gives more freedom to partners. This can stimulate the growth of the ecosystem but does come with the risk of losing control over the platform and getting too many and/or low-quality applications on the platform. The model below shows how an open platform can lead to a larger user base [2].

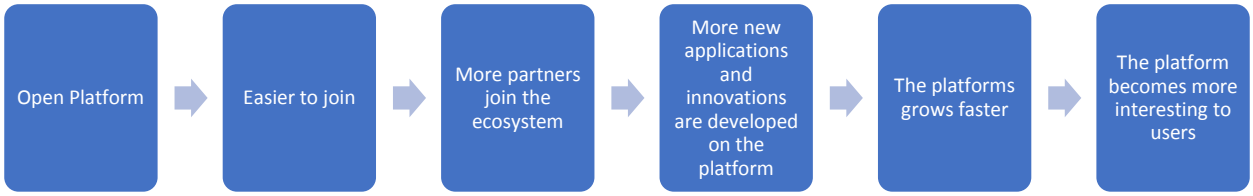


Figure 38: Open platform leading to larger user base

Open versus Closed

In the next table the main differences, risks and benefits of being open or closed are presented.

Open Platform	Closed Platform
Few rules for participation	Many rules for participation
Easy to get permission or no permission needed/ soft requirements to access resources and create applications. Easy to join.	Difficult to get permission/hard requirements to access resources and create applications. Difficult to join.
Multi-homing is allowed	Multi-homing may not be allowed
Difficult to control quality of platform offering	High control and therefore high-quality offerings

Open Platform	Closed Platform
Risk: low quality apps due to limited control over the platform	Risk: not enough apps Risk: too strong rules scare away potential partners
Risk: crowding effects which can lead to negative user experiences	Benefit: Competition and potentially harmful developers can be denied access to the platform.
No exclusive content deals	Exclusive content deals
No limits to commercialization, participation and platform usage	More security control on the platform
Faster innovation with more partners/ more radical innovation around the platform	Benefit: limits the amount of services offered and prevents the platform from getting out of control; customers will not be overwhelmed by too many offerings/less noise and pollution
Scaling up quickly and thereby eliminating the competition	Being closed can give investors and platform sponsors a better chance to get back their investment
Maintain control by not opening up all components	Decision making not slowed down by partners, more efficiency, higher speed [2][2][2][2][2][2][2][2][2][2][2][2][2][2][2][2][2]

Absorptive capability

Absorptive capability of a company is the company's ability to recognize and assimilate external information and innovation and apply it to commercial ends within their own business [34]. If a company has high absorptive capacity more can be gained from opening up the platform and integrating external functionalities into the platform [34]. In general the higher the level of absorptive capability the platform provider has, the higher the degree of openness the platform generally has [34].

Clan Control

Clan control is an informal mode of control for a platform in which common norms and interests and mutual monitoring align efforts and guide development actions [34]. The more clan control is encouraged by the provider, the higher the degree of openness of the platform generally is [34].

Examples Closed Platform

The Nintendo game console is a typical example of a closed platform. There is a very strict and rigorous process of permissions for game developers [26]. Nintendo requires this to be able to ensure high quality games to their user base [26]. The console strategy is completely closed but game development by licensed developers is encouraged [26]. In most platforms the core module of the ecosystem is closed [26]. The most-used example of closed platforms for most literature is Apple [17]. Apple uses the power it gets from having a closed platform to limit and restrict who can have access to their platform [17]. Apple has very strict rules for any applications in its App store which has allowed them to keep out viruses and competitors to their own products [17].

Example of Middle of the Spectrum Platform

Microsoft makes part of their source code available to developers who can then produce and distribute Windows applications without going through an approval process for the application [26]. However, only developers that have gone through an approval process and are now licensed developers can create applications that do not need to be approved [26].

Example Open Platform

The Android platform for smartphones is a typical open platform. Anyone can develop apps for the platform and there are very few checks regarding the quality of the apps that are offered in the App store.

Ambidexterity in Openness Strategy

Keeping the platform closed can be a strategy when starting a new platform [2]. However, in the long term this may not be the best strategy [2]. Openness is required for innovation and will contribute to a platform that remains competitive for years to come [2].
Openness can also be considered from the ambidexterity perspective [26]. Domain separation: closed and open strategy. Often: closed-oriented strategy in one domain, this is to protect exploitation, simultaneously an open strategy in another domain to explore new resources and capabilities.
Ambidexterity of openness strategies by organizational and domain separation.

User Experience in an Open Platform

The user experience should not be lessened by opening up the platform. The greatest risk of an open platform is that the quality of the user experience will decrease. Low quality apps or plugins that are offered on the platform should be avoided as they will lead to a negative user experience on the platform. Crowding effects should also be avoided. When too many apps are offered users become overwhelmed by the large amount of choices offered and cannot even find what they were looking for because there is so much on offer. For example Salesforce offers several thousand apps to choose from, this is not an optimal amount of choice for people as will be explained in the next section on the psychology of choice.

The Psychology of Choice

Humans are addicted to choice, they want to have options, even if only in theory [36]. Humans rather have choice from a list of 5 then form a list of two. The idea behind wanting as much choice as possible it that with more choices the chances of finding the perfect match for their needs are the highest [36]. This has its limits in practice: humans are less likely to make a decision when they have more options [36].
A study of e-commerce websites has shown that customers had a much harder time making decisions when given a large number of options [36]. When people use the filtering options making a decision get easier [36]. When there are too many options to choose from no decision is made at all, or if a decision is made the decider generally feels frustrated [36]. When people have more options the fear of making the wrong choice increases [36]. Humans can also cognitively not compare more than 5 items at once [36]. The information excess makes it impossible for the human cognitive system to process the information efficiently and without becoming overwhelmed [36]. With too much choice the decision making process is disrupted [36].

Figure 39: Effects of the number of choices on decision making

The Jam Experiment [36], [37].

This experiment was conducted by Sheena Iyengar, a professor of business a Columbia University in 1995. A too large amount of choice can be debilitating [37].
I think I quite literally experienced this last year on holiday on France. On a village market there was a stand with at least 30 different flavors of jam many of which sounded tasty, we could not decide and ended up buying nothing.

The Jam Experiment			
6 flavors of jam	Conversion rate of 30%	40% interest generated	More sales, less generated interest
24 flavors of jam	Conversion rate of 3%	60% interest generated	More interest generated, less sales

Benjamin Scheibehenne, a research scientist at the University of Basel in Switzerland, thinks it is not just the number of choices but what information we are given while making these choices, the type of expertise we have to rely on and how important the choice is [37]. He says the distinction between choice overload and information overload is important to make [37]. If we have more choice, we are less satisfied once we have made a choice [37]. There is always the feeling that maybe there was a better choice [37].

APPLYING THE PSYCHOLOGY OF CHOICE TO BUSINESS

If there are too many applications offered on the platform this may actually decrease the value proposition as customers will get overwhelmed, confused and maybe even frustrated by the choice they are trying to make, and they might end up buying no applications at all. Filtering and a good search function are mitigations for this problem.

Limit the number of options that are presented to the customer without limiting the number of available options too much [36]. By presenting choices in well-thought-out manner the selection process becomes easier [36]. For example, never show more than 5 items in a row [36]. Filtering is very important in support the selection process, this efficiently and effectively limits the number of options without limiting what is available [36]. Proving a default of suggested options also is a proven method as the effort to select the best option has already been done for the customer [36]. Social comparison is another tool to support decision making [36]. “People who looked at this also bought…” for example. Nudging is another technique which can be used to gently steer people towards a certain choice. It makes it easier to choose and people feel better about their choice. This nudging option if often labelled as “recommended” or “most popular”. This option is often engineered to be the most profitable for the company.

Governance: Distribution of Decision Rights

The second sub dimension of governance is how decision rights are distributed. Decisions about the platform can be made by the platform owner and by partners of the platform. This is often referred to as central versus distributed decision rights in platform strategies [16], [18]. When the app developers have many decision rights the governance is decentralized. When the owner has all decision rights the platform is completely centralized. The platform can be controlled by a central organization or be more collaborative between several organizations, who organize the platform together in a distributed manner.

This is also sometimes described as a range from cooperative to competitive, this then describes how the different partners of the platform view each other [2]. They are working together to achieve a common goal or offering rival complements on the same platform.

Decision rights refer to decisions about the platform or about apps, about strategic things or implementation (what to accomplish and how to accomplish it) [16].

Central	Distributed
Strong control and governance allow successful platforms to scale up quickly, leading to a winner- takes-it-all effect	Combining knowledge and best practices to quickly improve the quality of the product
Control over the complete user experience, seamless integration of all components and applications	Freedom can lead to higher acceptance and a faster spread of the platform and accompanying standards
Dividedness: different branches are created, and companies must choose which platform to join.	More freedom for developers. Greater community support for the platform because social initiatives are given a chance, the sharing economy. Freedom can lead to higher acceptance and a faster spread of the platform and accompanying standards. Combining knowledge and best practices to quickly improve the quality of the product. Risk: the creation of islands and random growth of standards, confusion for the consumers what is offered and at what quality
Autonomy will motive developers Risk for seamless integration of all platform elements.	
Dividedness: different branches are created, and companies must choose which to join or to join a competing platform.	Greater community support for the platform because social initiatives are given a chance, the sharing economy
	Risk: the creation of islands and random growth of standards, confusion for the consumers what is offered and at what quality

Choice	CSF and Goal Contribution
Open platform	This will attract more partners and therefore there will be more applications available on the platform which will make it a better value proposition for (potential) customers.
Closed platform	This will ensure the quality of everything offered on the platform which will increase the likelihood that users of the platform have a good user experience and will therefore stay on the platform for longer.
Limiting choice	Keeping control over the amount of applications on the platform is advised to prevent low quality complements and choice overload which will cause dissatisfaction among customers.
Variety in openness	Opening and closing the platform throughout various stages is advisable. Keeping the core closed while opening other components is also advised.
Central governance	There is more control over the platform, so the user experience can be controlled
Distributed governance	This will attract more partners, however it does also mean releasing some control over the platform which does not seem like the best choice for Tersof as ensuring the user experience is vitally important.

Dimension: Evolution

The concept of how a platform can evolve through time was mentioned in several of the studied strategies [2], [16], [18], [25], [32]. A platform can evolve in many ways. It can start as a stand-alone product and later add complements from external partners, it can evolve by enveloping functionality from adjacent markets by growing the entire ecosystem around the platform [16]. Evolving is important for platforms to stay ahead of the competition. The Red Queen effect for example describes how evolving at the same speed as your competition is in fact standing still [16]. With platform businesses competition is between the entire ecosystem and not just the platform itself.

Evolution through Time

This dimension focusses on how a platform can evolve through time to stay competitive. Choices made in the other dimensions will likely have to be reconsidered regularly through time as the platform grows, evolves and matures. For example although a low-price strategy can be a good way to launch the platform and gain enough users for critical mass, the low price will not be enough to maintain the platform in the long run so prices should be changed as the platform grows. The same goes for non-financial user attraction strategies, getting the first users is much more difficult and requires a different approach than a platform which already has grown to a substantial user base. A platform may not desire many partners at first but if later there is a desire to attract more partners than the governance needs to evolve to a

more distributed and open structure. So for every dimension the right decision will change with time as the platform evolves.

Evolve-ability through Modular Architecture

The architecture of the platform will become increasingly complex as the platform evolves and the business ecosystem expands [25]. The architecture should be designed to support likely scenarios and future plans [25]. Architecture is the combination of components of a complex system, what they do and how they interact [16]. It is a high level description of the building blocks of the platform [16]. Architecture is a technical decision with strategic consequences [16].The decisions are both technical and strategic and require involvement of both technical and strategic persons -who are not familiar with each other's domain. Architectural choices are often irreversible and therefore they impose constraints which often only become apparent long after the decision has been made [16].
By having a modular architecture the platform is plug and play. Architecture is a continuum between completely modular and completely monolithic [16]. Architecture reflects how evolution and innovation will occur as it defines how innovation work is partitioned among complementors and the platform owner [16]. A modular Lego-like architecture is the most evolvable [16].
Architecture allows complementors to contribute to the platform while being ignorant to most offerings on the platform, without this negatively impacting the entire platform ecosystem and the integration and cohesion [16].

Emergent Innovation

Partners in the ecosystem pursue their own interests based on their expertise and on feedback from what others in the ecosystem are doing. This will lead to innovations that arise spontaneously and not in a way organized by the platform owner [16]. The platform owner cannot control or plan emergence, but he must enable and facilitate it [16]. Innovation is usually a result of the selfish desires of complementors.

Evolving through Growing the Entire Ecosystem

Being committed to growing and extending the size of the entire ecosystem and not just the platform itself is important. By growing the entire market there will be more innovation which can be absorbed into the platform. The more innovation is stimulated around the platform the more interesting the platform will remain for users.

Evolution by Envelopment

A platform can evolve through envelopment [2], [26], [32]. According to the ambidexterity perspective framework this concept refers to the option to gain competitive advantage by operating in multiple platform-based markets simultaneously [26]. By bundling functionality of their platform owners can move into different markets [26]. Microsoft for example bundled Windows Media Player with the Windows operating system. [26] The platform owner already has an established position in the original market and can gain market share by moving into a new market [26]. In the book by Tiwana platform envelopment is defined slightly different. Platform envelopment is including functionality that was previously offered on a different platform in an adjacent market in your own platform, as can be seen in the figure below. This same definition can be found in more sources, adding new functionality to an existing platform to increase the value that the platform offers to its users [33]. If you, as a platform owner intend to launch an envelopment attack, you can be a complementor to the target, a weak

substitute for the target or have functionality that is unrelated to the target [26], [33]. When a platform offers the functionality of another platform in an adjacent market, then it has swallowed, or enveloped, that platform [16]. There must be overlap between the two platform markets in user base for this strategy to be viable [16]. The new bundled value proposition should be more valuable to users of the platform that is being enveloped [16].

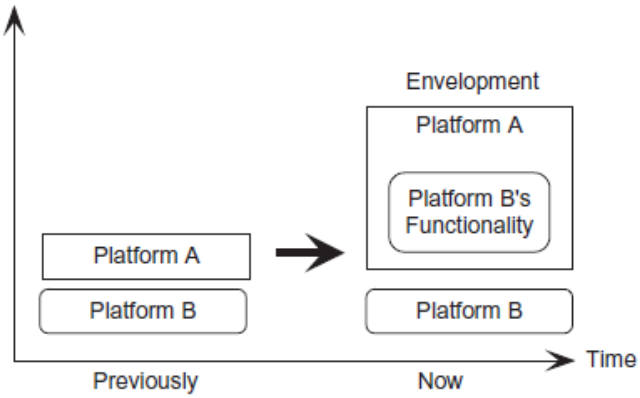


Figure 40: Evolution by Envelopment[16].

Evolving through Vertical Integration

Vertical integration is moving into the space of a complementor. A platform is vertically integrated when it offers a product or service themselves that used to be only offered on the platform by an external partner [26]. This is a risky strategy as partners will generally not appreciate this move as it will cost them income and they might be less willing to create complements for the platform in the future. This is also called the follower strategy, Amazon for example does this by offering the most popular products from external partners themselves [26].

Evolving through Bundling or Constellations

Platform expansion can be done through the bundling of features, the launch of an envelopment attack or by attracting complementors. Bundling is the adding of additional services to the platform. Constellation is creating new platforms to offer new services but still remaining some cohesion between the platforms for example by shared log-in credentials so multi-homing between the different platforms is encouraged. The third option is to attract external developers to expand the platform.

Complementary Innovation or Breaching

Another way to look at evolving is to make a choice between innovation and breaching [31]. With an innovation strategy the goal is to lure users to your platform by being innovative [31]. With breaching strategy to goal is also to reach new customers but then to do so on the platform of someone else [31]. This would be to offer for example the Tersof made applications on a rival platform.

Choice	CSF and Goal Contribution
Monolithic architecture	This can help to get the platform up and running quickly but limits future evolvability and is therefore not advisable.
Modularized architecture	This greatly increases the options for future evolvability of the platform as many options and directions are left open and components can easily be added and removed from the platform.
Absorb new innovations back into the platform	By absorbing external innovations the platform can grow with limited effort required from the platform owner and increase the value proposition for customers.

Choice	CSF and Goal Contribution
Standalone first	This a good way to gain a large amount of users before transitioning to a platform
Grown entire market	This is way to stimulate innovation around the platform with as goal to increase the value of the ecosystem for customers.
Enable emergent innovation	Partners cannot be controlled but they can be guided in a direction which suits the platform to extract maximum value from them.
Vertical integration	This is a way of absorbing external innovation into the platform, this will make it easier for users of the platform to access this functionality.
bundling	The additional services make the platform have more value for (potential) customers.
Constellation	Users can join only those platforms they have a need for without being required to pay for the entire constellation, so there is less financial risk in joining.
envelopment	New functionality is added to the platform which makes it more usable for current and potential users.
Innovation	By being innovative the platform remains relevant for users and being more innovative than the competition can attract users to your platform.
Breaching	New markets and new potential customers can be reached by offering your value on a rival platform.

6. Case Studies

For this thesis two case studies have been done using the companies Atlassian and Salesforce. Atlassian and Salesforce have been selected because they produce B2B software, although they are not direct competitors of Tersof, as they have a slightly different value position, they do operate in a similar market. The companies are also roughly the same age (Atlassian founded in 2002, Salesforce 1999, Tersof 1997) and both Atlassian and Salesforce have already established themselves as platforms in the market [38], [39]. These characteristics make them both interesting cases to learn from and not to dissimilar to Tersof to not be able to extrapolate the lessons learned to Tersof.

Thesis Framework

For these case studies the framework of this thesis has been used. The framework is used to show which strategic choices these companies have made and what the results of these choices are in a systematic way. For each case study a general introduction about the company is given. The case studies are used to validate the framework and the six dimensions (trust, usability, user/partner attraction strategy, pricing, governance and evolution), are studied to see how they relate to the platform strategy and if they support the framework or suggest a needed change in the framework.

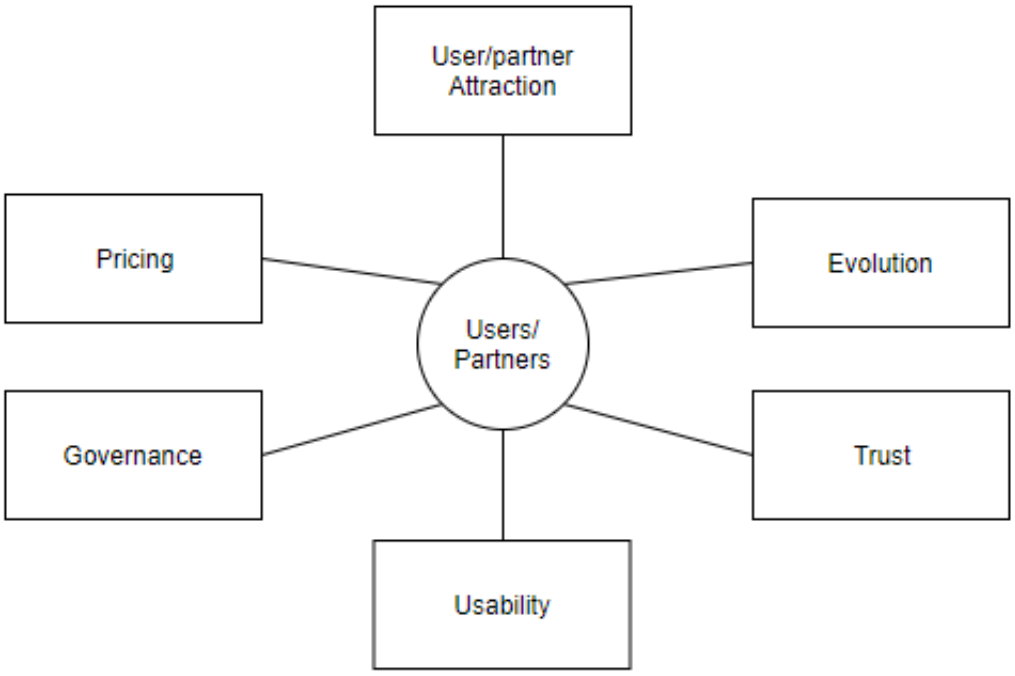


Figure 41: The first version of the platform strategy framework for validation by the case studies.

Atlassian Case Study

About Atlassian

Atlassian is a software company that was founded in Australia in 2002 and has since grown to a company with locations in Sydney, Amsterdam, Poland and San Francisco [39]. The company started out as two people and now employs over 2,500 people who together make software products which are used by over 112,000 different companies [40]. Atlassian is an independent software vendor that produces several different products. Atlassian offers products for several business processes. There are tools to support planning, tracking and control, tools for collaboration and chatting and tools for coding, building and shipping software [40].

The products are Jira Software, Jira Service Desk, Jira Core, Statuspage, Confluence, Trello, Stride, Hipchat, BitBucket, Sourcetree, Bamboo, Fisheye, Crucible and Crowd [41]. In 2017 Capterra, a large and independent software review site, named Jira Service Desk the #1 Most Affordable ITSM solution and the #1 Most Popular ITSM software [42]. For a complete overview of the Atlassian products see appendix 1.

Atlassian Mission

The Atlassian mission is to develop affordable, lightweight software that helps enterprises collaborate better [43]. The goal of Atlassian is to help teams build better software faster [44]. Atlassian has defined several core values for their company. A core Atlassian principle has been from day one that all products are customizable and can be extended. From early on Atlassian has encouraged its users to modify and enhance the Atlassian applications using the Atlassian Plugins framework [44].

Atlassian Strategy Core

The core of the Atlassian platform strategy is not #@!%ing their customers. This is at the heart of everything they do and the choices they make.

Atlassian Core Values

Just like Tersof Atlassian has defined core values guiding the development of their products and which are communicated to their customers to show them what Atlassian stands for [45]. The Atlassian culture highly values customer trust [46].

The five Atlassian core values:

1. Open company, no bullshit
2. Build with heart and balance
3. Don't #@!% the customer
4. play, as a team
5. Be the change you seek

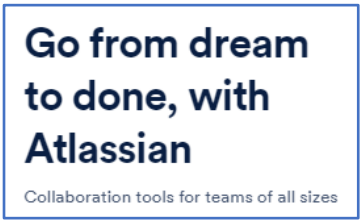


Figure 42: Motto Atlassian

Open Company, No Bullshit

The company highly values openness. Internally all information is openly available and shared as a default setting [45]. To customers there is a lot of openness in terms of what the company is doing and why and how extensible the software is.

Build with Heart and Balance

Without limiting passion and considering urgency the company still strives to make well-thought-out decisions [45].

Don't #@!% the Customer

Atlassian is very strong in their statements about the importance of their customers, they are always the most important factor to consider when building a product or making company decisions [45].

Play, as a Team

Customers are hugely important but so are the employees of Atlassian. They believe in creating a good, pleasant and healthy work environment [45].

Be the change you seek

Atlassian stands for customization in their software and they encourage their employees to bring their own ideas and create change within the company [45].

Atlassian Platform

A key feature of Atlassian products is that they are customizable and extensible [47], [48]. With the plug-in framework developers can build anything to run within Atlassian products [48]. Any feature that is not in the core product can either be bought from the marketplace or you can build it yourself or get an Atlassian expert to build it for you [47]. Add-ons are written in Java, JavaScript, CSS and HTML and use REST APIs [47], [48]. Atlassian uses their own plugin framework to build new features for their products.

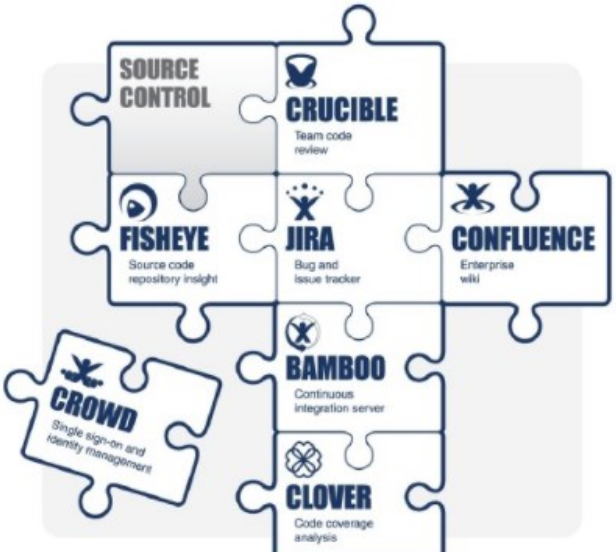


Figure 43: Atlassian Platform[84]

Add-ons/ Apps

An add-on is web application that can display content within or interact with Atlassian products, it is a small piece of functionality that can be added to an Atlassian product [44]. The add-ons can be made in any language, using any web framework and any libraries [44], [49],for example popular frameworks such as Node, Play, Rails or Django can be used [44].

"Our products will always have something lacking for a set of users. And as agile as we are, we'll still never be able to build everything ourselves in the time users want them. In the end, the only way to scale our development is by adding more developers. And since there are always resource constraints, getting our users to help us is an excellent solution to that problem. So, because of our plugin framework, users (developers, hackers, customers, etc.) are empowered to enhance the products themselves. If you're a developer in the Atlassian ecosystem, "Thank you!" [48]

Key Atlassian Platform features

The Atlassian platform for developers has four key features [50]:

1. Platform: the platform is continuously evolving to become easier to use and make it easier to build and deliver Atlassian applications to customers.
2. Developer Experience: both new and existing developers should have a pleasant experience when creating products for Atlassian.
3. App Marketplace: this should make it easy for potential customers to find the applications.
4. Customer Success: allow developers to create products that truly have added value for their users.

Atlassian SDK

Atlassian has chosen to supply an SDK especially for plugin developers [51]. With this toolkit Atlassian products such as JIRA, Confluence, Bitbucket Server and Bamboo can be extended [51]. They offer several tools to support the building and testing of plugins [51]. Atlassian currently supports both on premise and cloud based versions of their software [51]. The guidelines for creating a plugin are different for server versus cloud. This issue might apply to Tersof as they also provide both cloud and on-premise instances of their software. The Atlassian SDK can be used in common integrated development environments (IDEs) such as Eclipse, IntelliJ IDEA and NetBeans [52]. The technical structure of the plugin development platform can be seen in figure 44.

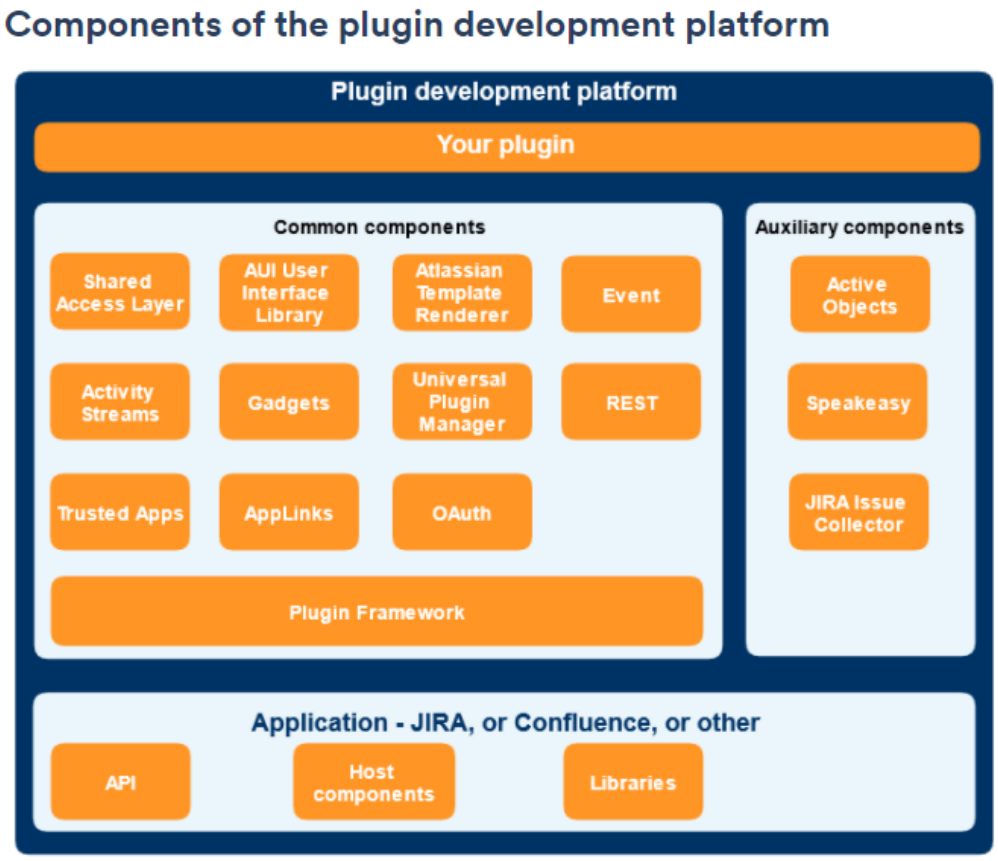


Figure 44: Components of the plugin development platform [53]

The Atlassian Platform Marketplace

Atlassian marketplace is a platform that customers can use to discover, try and buy apps, made by Atlassian or third parties, for Atlassian products [44], [49], [54], [55].

Examples of Extendible Atlassian Products

Jira

Jira is software for issue tracking and project management. It is one of the most used Atlassian products. Jira is easily extensible [47]. In Jira for example there is an inventory plugin for asset management [56]. The extensibility layer makes it easy to add new features and extend existing features [47]. The users can develop custom functions themselves or an Atlassian expert can build a custom tailored experience [47]. Data is standard format, Atlassian will help to migrate the data for free [56].

Jira Service Desk

Jira Service Desk integrates with other Atlassian apps such as Confluence, Confluence Team Calendars, Jira, Hipchat, Slack and Statuspage [42]. Jira Service Desk can also be extended with add-ons [42]. These add-ons can be used for customization of the application [42]. For example the portal can be customized to reflect the company brand, dynamic forms can be created and dashboards can be made [42].

Confluence

Confluence is software for collaboration, it is an advanced enterprise wiki [43]. It is extendible through third-party plugins and the Platform is open to the developer community [57]. Anyone can build on top of the basic platform and extend it [57]. Wide range of free and paid plugins [57].

Atlassian in the Framework

In this next section for each dimension of the framework a description of the choices Atlassian has made will be given. It is difficult to figure out exactly what the results of each specific choice throughout the years has been but based on their gigantic user base they seem to have made some very good business decisions on the different aspects of the framework.

Dimension: Trust

Security

Atlassian writes on their website about how information is at the heart of all business and lives and that they therefore consider customer trust to be at the center of their operation and that security is a top priority for them [58]. In relation to trust they again highlight their dedication to transparency stating that they are transparent about their security so users can feel both informed and safe [58]. After this introduction the webpage continuous with another 4931 words on all the security measures they have in place and the choices they have made [58]. The page links to many other pages about security at Atlassian. Looking at all the different information on security at Atlassian that can be found it is clearly something they are very committed to and very open about, in line with the core values of being good to their customers and being an open company. They strive for continuous improvement on security [58]. Security is embedded in the architecture [58].

Quality Control

Atlassian has acquired all the accreditations below:

Figure 45: Accreditation logos

[58] Name	Organization/ owning body	Controls/requirements
SO27001	International Organization for Standardisation	26 requirements
ISO27002	International Organization for Standardisation	114 requirements
PCI-DSS	Payment Card Industries	247 requirements
CSA CCM	Cloud Security Alliance	133 controls
SOC2	Service Organisation Controls	116 requirements
SOX 404 (IT)	US Federal Law	22 requirements
GAPP	AICPA	106 requirements

As stated before Atlassian is committed to security and open about their security measures that to summarize everything they do would require over 40 pages, therefore a few examples are given, for a complete overview the source materials are available on the Atlassian website [58].

Balance

Atlassian strives to find the right balance between effective controls to ensure confidentiality, integrity and availability of customers data without limiting the flexibility of the company and their products too much [58]. Their security policy is designed to maintain this balance [58].

Access Control

Access control is done through strict firewall rules, multi-factor authentication and encrypted connections [58]. There are intrusion detection and prevention systems in place to identify potential security issues [58].

Threat Modeling

Threat modeling is used at Atlassian to understand the specific risks of products and features [58]. Threat modeling is a brainstorm session between engineers, security engineers, architects, and product managers that are related to the product or features. The goal of this threat modelling brain storm session is to identify and prioritize threats and to use the outcomes of the brainstorm sessions during the design process and testing of the product or feature [58]. For this they use the Microsoft Threat Modeling tool and the STRIDE Threat Model framework. STRIDE stands for Spoofing, Tampering, Reputation, Information Disclosure, Denial of Service, and Elevation of Privilege. They claim to use threat modeling often and early to ensure security and have mitigation in place for identified threats in every product or feature that is developed [58].

Resiliency and Disaster Recovery Program

Atlassian had resilience practices based on SOC2, ISO 27002 and ISO 22301, this to minimize downtime [58]. There is also a Disaster Recovery Program which is based on continuous improvement, assurance through testing and dedicated disaster recovery teams [58]. The business continuity and disaster recovery plans have been tested [58].

Credibility through Partnerships / Certified Partners

Partners are also certified. For example the hosting partners for Bitbucket are SOC-2 and ISO 27001 certified to ensure security and availability [58]. Data centers are regularly audited to make sure they comply to for example SOC1 and ISO/IEC 27001 standards [59].

Back-up Regimes

There are daily and weekly back-up regimes in place. The daily back-ups are stored for 30 days to ensure data availability to customers [58]. Potential employees and partners are required to sign confidentiality agreements and during the onboarding process security awareness training is mandatory for new employees [58].

Security Testing

There are internal and external security testing programs [58], [60]. Atlassian also partnered with Bugcrowd to encourage users to find and report bugs, in exchange for which they then receive financial compensation trough the bug crowd platform [60]. They even have an official policy on their website which states how quickly different types of bugs will be fixed [61].

Breach Detection and Monitoring

The systems are monitored so any security breaches can be detected [60].

Privacy Policy

To ensure customers that their personal and company data is kept safe there is a privacy policy [62].

Dimension: Usability

Usability is perhaps best measured by simply asking a lot of users what they think about the Atlassian software. Unfortunately this would be too time consuming for the scope of this thesis. Therefore the assessment of usability is based on independent software rating websites, their commitment to it, and the personal experience of a user to give some practical examples.

Independent Software Rating Platforms

Gartner Peer Insights

Gartner Peer Insights is an independent platform on which users can rate software vendors such as Atlassian. The products Confluence, Jira and Hipchat are reviewed [63]. The first review that is shown for Jira, based on the review criterion 'most helpful' has as title "Unsurpassed Usability" [63]. The user experience is rated at 4.2 out of 5 based on 509 reviews [64].

Capterra

Capterra is a platform to help users find business software. On this platform users can rate and review business software [65]. Jira has 4 out 5 stars based on 2888 reviews for the overall product and for the category 'Ease of Use'[65].

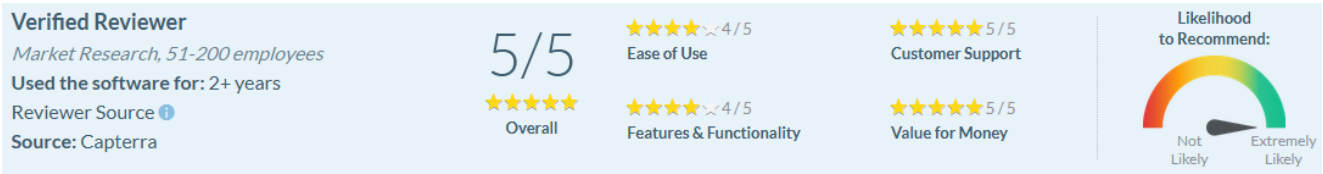


Figure 46: one of the 2888 reviews, without the detailed review text, [65].

Commitment to Usability

The Atlassian Research Group

On their website Atlassian invites users to join their research group [66]. Users and non-users can participate in interviews, usability tests and surveys to help Atlassian to improve their products [66]. Some of the research studies are even paid for by Atlassian and through video conferring anyone anywhere in the world can join to give their opinion [66].

Partner of the World Usability Day

Atlassian was an official partner of the World Usability Day 2017 [67]. Although this does not automatically mean that Atlassian products have high usability, it does illustrate that Atlassian is committed to improving usability and that this is something they want to show the world. They want to communicate to their users that they consider usability to be important.

Now Hiring: Usability Researcher

As usability is something that can always be improved they are looking to hire new usability researchers [68]. They value a consistent user experience throughout all their different products and are therefore willing to invest in hiring new employees to safeguard this consistent users experience [69].

Personal Experience of a user

To get more information about the usability of Atlassian products I interviewed a regular user of Atlassian products. This was only a single interview and therefore not necessarily the 'truth' about usability, nor scientifically completely sound, however he gave some interesting examples about his experiences with the product. I interviewed Stefan Kampman, he has been a software engineer for 12 years at different companies such as ASML, BDR Thermea and Demcon. He has mostly experience with Jira. He found the overall usability of the product rather good, scoring it an 8 out of 10. However, as a main usability concern he stated that the software is so flexible that it is easy to implement a far from optimal workflow using Jira. So the usability of the product is highly dependent on how a company chooses to use it and how workflows are implemented. An example of a bad process flow he gave was the option to go from any state of a Jira ticket to open, closed or reject. Therefore tickets are never fully solved as they can always be reopened again. According to him this is one the instances in which the tooling offers the possibility to, very easily, be used in the wrong way. Overall, he claimed the program is easy to use, however, there is a steep learning curve at the beginning. It takes a while to go from additional overhead to added value of the product. However, he still said that he would definitely recommend using it to other people. He found it was easy to use, the search function really adds value, although switching between different projects can be difficult. Another risk of using the software is that it only works properly when everyone uses it and when everyone uses it properly.

Dimension: User/partner Attraction Strategy

User Attraction Strategy

Attraction is based on advertising the ease of use, ease and low price of getting started, positive company values and mentioning Fortune 100 companies using their software and organizing annual conferences.
Atlassian advertises with being easy to use and easy to get started with. Starting is cheap: a preconfigured and hosted environment which can be up and running within minutes is offered for 10 dollars for 10 users [70], [71]. This will allow any company to try out the software without any significant financial risk. This low starting price lowers the threshold for potential customers.
The advertised company values and their pledge to donate 1% of employee time, equity, profit and software licenses to the community show a positive image of the company to potential customers [45].
On the website there are also several success stories from well-known companies that use Atlassian's software, such as Capgemini, Domino's pizza and Nasa, as a way of convincing potential users that their products are worth buying [72]. They claim that more than 75% of Fortune 100 companies are Atlassian customers [73].

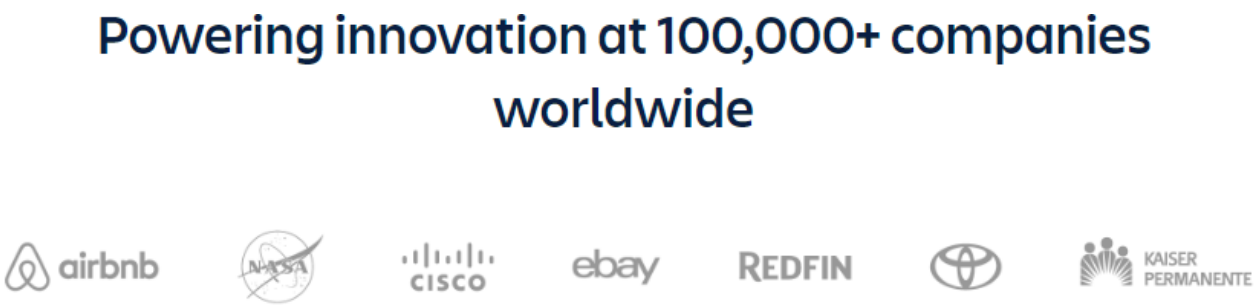


Figure 47: Some of Atlassian's customers [40].

Partner Attraction Strategy

For potential partners Atlassian offers a very large potential user base, 125,000+ customers, and they only take 75% of revenue from app sales and there is no cost to list and app in the marketplace [39], [55], [74]. Atlassian provides the licensing support and handles the checkout process [55]. There are some strict rules with might deter some potential partners but overall the gigantic user base and fair revenue division has led to 3,500 different apps being available in the Atlassian market place [39]. There are several awards and recognition opportunities for partners. For example the plugin of the month can be presented on the Atlassian blog to give it more visibility with potential customers [75].

Inspiration and Events

To promote their app third party vendors can write a guest blog for Atlassian [76]. Atlassian does require that the vendor who writes the blog has practical, tangible, hands-on advice as the topic of their blog [76]. There is a large partner and user community where connections can be made, and help can be found. Ideas and inspiration are shared.

Atlassian Summits are annual user conferences, every year there is one in Europe and one in the U.S. [77]. Visitors of the conference get a first look at the latest Atlassian innovations, can get training and meet other Atlassian users." It is all about learning and being inspired by industry professionals doing amazing things [78]." The Atlassian Summit in 2016 offered 75 different talks in 7 different tracks [78].

Dimension: Pricing

The pricing dimension is actually two dimensions, pricing for users and pricing for partners.

Pricing for Users

Atlassian has a clear pricing policy. They believe in transparency so all prices are stated on the website [70]. For example by looking at the monthly pricing the strategy can be seen. They want to attract new users by offering the software cheap for small groups of users. After that there is an increase in price to separate the trial and small business pricing from small/medium enterprise pricing. After this the prices decrease again to entice companies to sign up more users within their company. The price for a single product is very low, for example Jira is only a dollar per user per month for a company with more than 251 users. However, if a company uses all Atlassian products the price does add up. [70]. Atlassian also offers the option to pay annually, there is a discount for the user to entice them to choose this option and the company has more certainty that customers will not unsubscribe after a month and they have immediate cash inflow instead of smaller amounts throughout the year. However, they do make more money from the monthly pricing when customers do not unsubscribe because these prices are higher.

For the monthly pricing the price remains the same for 251 to 2000 users, for the annual pricing there are more tiers in the pricing. The annual pricing can be seen in Appendix 2.

The low and transparent pricing makes it easier for potential customers to make their decision to purchase the software. The discounted pricing for larger user groups can entice a company to roll out the software throughout the entire company and not just a few small teams.

Monthly Pricing

Product	Up to 10 users (users 1-10)*	For more than 10 users (11-100 users)	For the next 150 users (101-250 users)	For each additional user (251-2,000 users)
JIRA Software	$10.00 flat	$7.00/user	$4.00/user	$1.00/user
JIRA Core	$10.00 flat	$5.00/user	$3.00/user	$1.00/user
Confluence	$10.00 flat	$5.00/user	$3.00/user	$1.00/user
Portfolio for JIRA	$10.00 flat	$3.50/user	$2.00/user	$0.50/user
Capture for JIRA	$10.00 flat	$2.30/user	$1.30/user	$0.30/user
Questions for Confluence	$10.00 flat	$2.50/user	$1.50/user	$0.50/user
Team Calendars for Confluence	$10.00 flat	$2.50/user	$1.50/user	$0.50/user

Figure 48: Monthly pricing [70]

Pricing for Partners

External partners that develop apps for Atlassian products can choose from three different marketplace models. The app can be offered for free, as an open source application or it can be sold [55]. Atlassian sets a minimum required price for apps and maintenance renewals are always 50% of the commercial license price [74]. The revenue from app sales is always 25% for Atlassian and 75% for the party that has made the app [74]. There is no cost to list an app in Atlassian's marketplace [55].

Dimension: Governance

Open versus Closed

On the scale from open to closed I would rate Atlassian as very open. They claim themselves to be an open company, it is in their core values [40]. About many aspects related to Atlassian a large amount of information can be found on their website; ranging from how exactly revenues are paid to guidelines for making apps and detailed pricing lists. On their blog and website a lot of information can be found, I would estimate there are several hundred pages explaining every little detail of any Atlassian related aspect. Extensibility is key concept for development at Atlassian [79]. The Atlassian products are built with both users and developers in mind [79]. Anyone is allowed to extent features and therefore a certain level of openness is required from the company to facilitate this [79].

Public Roadmap

Fitting with the Atlassian core value of being open is the fact that their roadmap is openly available to the public.

The roadmap shows what has been shipped, what they are currently working on, what they are planning to work on and initiatives under consideration [50]. The roadmap is hosted on a Trello board, so it is made up of individual cards. Users can vote on these cards which changes or innovations they find most important [50]. The product teams monitor the votes, questions and comments [50].

Distribution of Decision Rights

The distribution of decision rights is mostly central. To ensure the quality of the software there are quite a few rules developers have to follow and there are strict guidelines which must be adhered to [80], [81]. It can take 5 business days for an app to be approved and quite often apps have to be submitted for approval several times before being available for sale in the app store [80]. This is great for customers as the quality is ensured through this strict control process, however, it may discourage partners if their apps are rejected several times. The long list of rules an application must follow will also limit the options that developers have to create the desired apps.

Dimension: Evolution

Evolution can be described based on many different things, but the goal is to show how the company has grown since it got started and how it is currently growing.

Atlassian Apps Growth

The following tables show the number of apps that are available on the Atlassian marketplace on 22-3-2018 and 06-7-2018 to show how fast the number of apps in the market place grows. In the tables it can be seen that the number of apps has grown with over a thousand new apps in just a few months.

Atlassian Products [82]	Number of apps available in app store 22-3-2018	Number of apps available in app store 06-7-2018
Jira	1,651	1,752
Jira Service Desk	659	760
Confluence	869	923
Bitbucket	299	303
Hipchat	205	202
Fisheye/Crucible	41	39
Bamboo	225	231
Crowd	19	18
Stride	17	36
Total apps available	**3,985**	**4,264**

Atlassian Hosting [82]	Number of apps available in app store 22-3-2018	Number of apps available in app store 06-7-2018
Cloud	1,037	1,129
Server	2,656	2,748
Data Center	1,522	1,623
Total apps available	**4,178**	**5,500**

Atlassian started out as a company run by Mike Cannon-Brookes and Scott Farquhar, after four years they had 40 employees and after 16 years they had 2,500 employees [39]. The company started with a single product and through the years they have grown by developing new applications and by buying other companies and adding their applications to their own offerings [83]. This is a key part of the Atlassian growth strategy, to not only develop themselves but to also attract new talents and capabilities by buying companies with promising products and by connecting with partners.

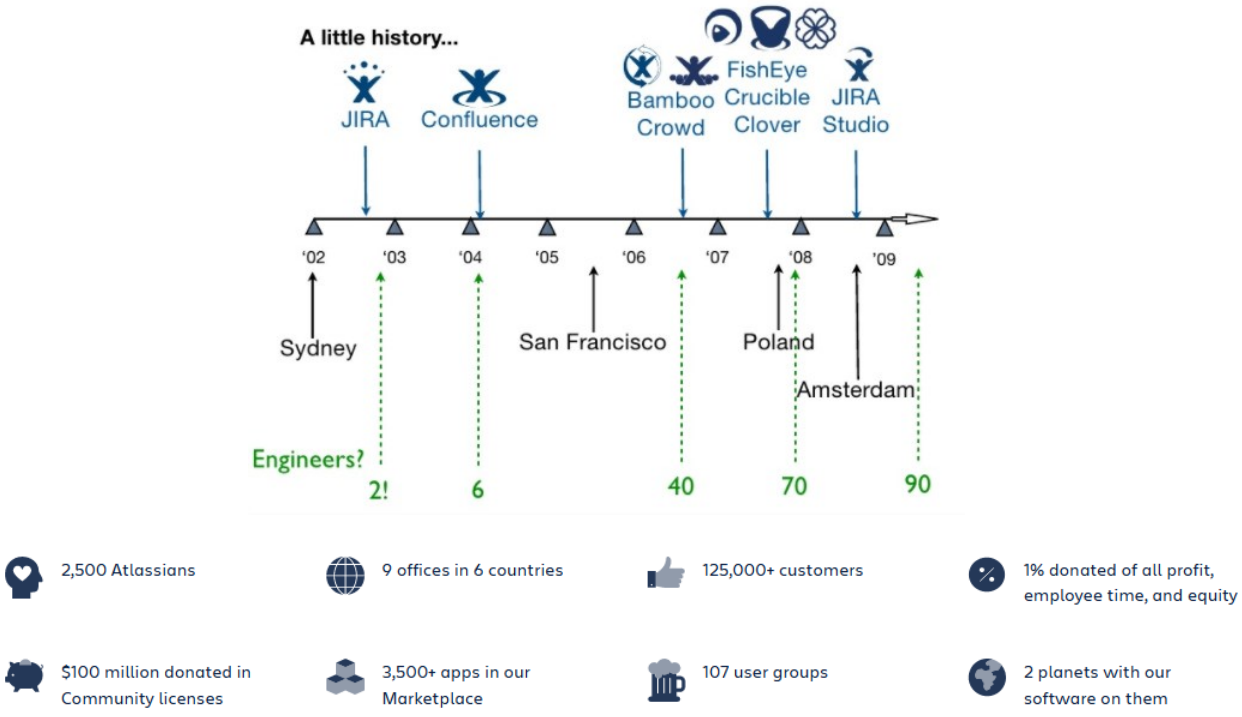


Figure 49: Atlassian history [84]

To show the evolutionary process of Atlassian I have constructed a time line with a selection of highlights out of the sixteen-year history of Atlassian, see appendix 3.

Atlassian Case Study Analysis and Conclusion

In this section the correctness and suitability of the platform strategy framework is analyzed.

Accurateness of the Dimensions

Based on the amount of information that could be found and the type of information it seems that the chosen dimensions do indeed play a very significant role in platform strategy. For each dimension much information could be found about choices which have been consciously made by the company Atlassian

to support a higher-level overall platform strategy. The fact that the choices that are made in the dimensions relate to a higher-level platform strategy and are not stand-alone choices suggests that there is indeed a relation between these strategic choice dimensions and the overall platform strategy.

Relationships between the Dimensions and the Core of the Strategy

It became apparent during the execution of the case study that not only the individual dimensions have a strong presence in strategy but also that they have a very strong relation in the sense that they all seemed to contribute to a greater higher level, the core of the platform strategy. Therefore perhaps in a revised version of the framework this strategy core should be at the heart of the framework and the users and partners dimension. Based on the case study it seems that the users and partners are in fact a result of this strategy core and the corresponding choices.

Suggestions for Improvements in the Framework

Based on the case study a different placement of the dimensions in relation to each other seems in order. The four dimensions trust, usability, governance and pricing from a coherent group and the users and partners and evolution form a second coherent group. Because all the choices Atlassian has made so strongly relate to a single strategy core a platform strategy core element should be added to the framework.

Salesforce Case Study

Salesforce is a cloud customer relationship management solution that was launched in 1999 by Marc Benioff, a former Oracle executive [30], [85]. They have been chosen as a case study because they have many similarities with Tersof. Salesforce is also a business to business software vendor. Most of the literature on platforms is written about business to consumer platforms making this case study a valuable addition. Salesforce offers customer relationship management software for small, medium and large enterprises. Their software allows companies to develop their own apps to engage their customers and extend sales and service capabilities [86]. They claim that apps can be launched 59% faster using the Salesforce platform [86]. Gartner has named Salesforce as leader in the "Magic Quadrant for Enterprise high-productivity application Platform as a Service (hpaPaaS)" [86]. Their disruptive business model has led to significant market penetration [85]. Salesforce has also succeeded in collecting a massive amount of complementarities [30]. For an overview of the Salesforce products see appendix 4.

Figure 50: Logo Salesforce

Salesforce World Tour

The claims in this section are based on web sources, information received during the Salesforce World Tour conference in Amsterdam on April 12th 2018 and personal experience with the software [87]. The conference was attended by over 5000 Salesforce users and sold out.

Figure 51: Logo Salesforce World Tour

Salesforce Official Value Proposition

Salesforce presents a five point value proposition [30]:

1. Innovation Ability. Salesforce has changed through the years in line with the needs of the market and is now an industry leader in CRM software with a web-based application model.
2. Connectivity. The AppExchange allows Salesforce to be a platform market leader. Developers and users of Salesforce are connected in an ever-expanding network.
3. Complementarities. The platform structure makes Salesforce highly customizable. The web-based cloud model creates great flexibility.
4. Efficiency. There is no high complexity or high initial costs to buy a new custom application. With the platform applications can be created quicker and deployed quicker than with traditional software.
5. Network Effect. There is a huge network of users and developers on the Salesforce platform. The AppExchange and online environment of Salesforce create positive network effects.

Salesforce Core Strategy

Based on everything I have learned about Salesforce the core of their strategy is ecosystem growth. A large part of the official Salesforce communications is based on what they call their 'Ohana' which is at the heart of their strategy. The end goal of Salesforce is to grow their ecosystem, which they consider to be part of their Ohana. Ohana is a Hawaiian concept which means that families—blood-related, adopted,

or intentional—are bound together, and that family members are responsible for one another and should take care of each other [88]. The CEO Marc Benioff was inspired by this concept and made it a core Salesforce value [88]. Ohana is an important aspect of the internal company culture and Salesforce also wants to extend this Ohana concept to the entire Salesforce community and the entire Salesforce ecosystem including all customers and partners [88]. They even have a special learning module on Trailhead, the fun way to learn Salesforce, about Ohana [89]. This is core to their strategy because their final goal is to grow their Ohana (ecosystem) and to take care of this family/ecosystem, make sure everyone is happy and taking care of and receive care from the ecosystem in return ensuring that Salesforce will live on.

Salesforce Ecosystem and Platform

Salesforce uses a multitenant cloud prescription model. During the conference it was said that 89% of customers have bought at least 1 app, 27% have bought 6 apps or more. Through the platform companies can reach a new potential set of customers in a new way. Just like Atlassian Salesforce also shares their roadmap with the general public so developers know what functionality will soon be coming to the platform and they can use this information to decide what they want to build on the platform. The interesting thing of a revenue sharing model with third party app development is that the owner of the platform needs his external developers to be successful to get a profit himself, and therefore he will be motivated to stimulate and create a positive environment for app development.
For an external developer a benefit of developing for a platform such as Salesforce is that the platform is already established. There already is a user base the developer can access and because the developer is selling his or her app through the platform, which has already established itself as trustworthy, therefore apps are automatically seen as trustworthy by potential buyers. The basis a platform offers in for example the SDK and API and cloud behind it also means that in general applications can be developed faster, this is also something Salesforce prides itself on.
Users of Salesforce have many options to customize the software and buy apps from third parties to run on top of the Salesforce platform in a way that is integrated with the standard functionality of Salesforce. Salesforce.com offers a service platform that other companies can use to build apps on that are based on and therefore work with the Salesforce software, companies can then sell these apps to other Salesforce users [34]. This part of the Salesforce platform is called the Lightning Platform [90]. As a part of my research for this thesis I even attempted to build an application using the Salesforce platform to gain some firsthand experience on how the platform can be used. Apps can be built with clicks or with code and can include many different functions, even artificial intelligence (AI) [86]. The platform according to Salesforce is built around the proposition of being fast, easy and fun [91]. My experience is described below in the chapter about the Dimension Usability.

Fast

With the Lightning app builder applications can be created and because of the ecosystem that Salesforce offers with prebuilt themes and components the application can be deployed fast [90]. Complex processes can be translated into apps using the Process Builder with point-and-click workflow and process tools [90]. The platform offers Lightning components, these components are reusable blocks of code for apps that can be used in many different ways to build applications without having to create the entire code from scratch [90].

Easy

It is easy to configure the software to specific needs as entire apps or components can be bought from app exchange and are immediately ready to go [90]. These apps and components are prebuilt to perfectly fit the Salesforce platform making installing and using them easy.

Fun

Trailhead is the learning community of Salesforce. Salesforce claims that the fun part comes from learning how to build apps and by learning this together with other community members [90]. Salesforce has made an effort to make Trailhead fun to use with good quality tutorials and by using interesting examples for the tutorials [92]. The fun is also intended to come from the gamification elements that are included in Trailhead[92]. By learning, completing tutorials and taking quizzes users can get points and badges to motivate users to keep learning using Trailhead [92].

MyEinstein

Salesforce has recently added Einstein to their group of mascots and to their products in the form of AI. There are now many options to integrate AI into Salesforce applications. With just a few click predictive fields can be created. Creating a predictive customer field is as easy as point -> click -> predict. During the live demo of T-Mobile during the Salesforce conference it was shown that is truly is a matter of a few clicks to create a new custom predictive field or widget. So even adding artificial intelligence is made easy.

Customization

A key selling point for Salesforce is customization. With the Lightning platform they have made customization even easier. Users can now create apps on Salesforce that are completely in their own branding or style. Again with simple clicking, no HTML or CSS needed, any app or webpage can get a makeover to fit with any brand. T-Mobile demonstrated this by making their application from Salesforce blue to T-Mobile magenta in just a few clicks.

Salesforce Community

Behind, or around, Salesforce there is a very large activity community. While I was writing this thesis there was a Salesforce conference in the Netherlands which I attended. It was quickly sold out and attended by several thousand Salesforce users. During this conference they also highlighted several Salesforce community aspects. For example the Salesforce Saturdays where anyone who is interested can come to a local cafe or pub and learn how to use Salesforce from other community members.
Salesforce Saturdays are a Dutch initiative that some extra attention was given to during the opening keynote of the conference. Every other Saturday a group of people, meet up to practice working with Salesforce together, anyone can join [93]. If a company created platform causes the creation of such initiatives without influence or incentives from the company, there is no longer just a platform but an entire ecosystem around it. It is an example of how Salesforce is creating a family, Ohana, all over the world of people you join to together and choose to help each other.

Salesforce in the Platform Strategy Framework

Dimension: Trust

On their website Salesforce has dedicated several pages to all their efforts related to the trust dimension. Seeing Ohana as the core of the Salesforce strategy, they claim nothing is more important than trust in the relationship they have with their Ohana [94]. On other pages they also claim trust is a core company value [95], [96].

Figure 52: Security logo Salesforce

Security Standards

Salesforce complies to the following security standards [97]:
ISO 27001, ISO 27017, ISO 27018. SOC 1, SOC 2, SOC 3. PCI DSS, FredRAMP Moderate, DoD IL2, DoD IL4, NEN 7510, NIST SP 800-171, HIPPA, HITRUST, Financial Services Compliance USA, Privacy Shield, TRUSTe Certified Privacy Seal, PrivacyMark, ASP/SaaS, TüV Rheinland CCS, iRAP, UK Cyber Essentials.
Judging by the number of official security standards that they comply to security is indeed a core value and a very important aspect of the Salesforce business strategy.

Security Tips and Advice

On the website there are links to security best practices, security training materials, advisories on security from customers, security information for administrators, security information for developers and security tips for all users [98]–[102]. Suggestions are given for app developers on how to develop secure apps and several badges can be earned on security such as basic knowledge identity, basic knowledge security, user authentication and security specialist [101]. On Salesforce events there is a security booth where administrators can get more information about security [103]

Security Announcements

Security announcements are posted regularly to keep customers up to date security issues, scams, security warnings and security incidents [100].

Security Health Check and Code Security Scanner

Salesforce offers a security health check [103]. This assists organizations in managing the most important security settings [103], [104]. Vulnerabilities in the security settings can be identified and fixed by administrators [104]. For developers there is a code security scanner [99]. Common security issues can be found using this scanner [99]. There is even a YouTube video of over 30 minutes entitles: “Insider’s Guide to AppExchange Security Review”, a security review that requires a 30 minute instructional video must be quite extensive and something the company takes seriously [105].

Security Best Practice Recommendations

Several security best practices are recommended to use. For example two-factor authentication, login-in IP ranges, using the protocol TLS 1.1 or higher, educating users about phishing risks, having password policies, and automatic session timeouts [98].

Privacy Statements

Salesforce has not one but an entire list of privacy statements to protect their customers and partners [106]. There is a Full Privacy Statement, EU General Data Protection Regulation Resources, Salesforce’s

Processor Binding Corporate Rules, applicable only to Sales Cloud, Service Cloud, Chatter, Communities and Force.com ("BCR Services"), Salesforce's Notice of Privacy Shield Certification, Salesforce Data Processing Addendum FAQ,
Salesforce Candidate Privacy Statement, and the Salesforce DMP Privacy Policy [106].

Figure 53: Security Trailblazer logo Salesforce [107]

Dimension: Usability

The usability dimension is presented following the same structure of the Atlassian case study based on independent software rating websites, their commitment to it, and the experience of a user, in this case my personal experience.

Independent Software Rating Platforms

Gartner Peer insights

On the independent software rating platform Gartner Peer Insights Salesforce gets a 4 out 5 stars overall rating [108]. The review rated as second most helpful does state that there is a high level of complexity in getting started with the software [109].

Capterra

On the Capterra business software rating platform Salesforce receives an overall 4 out 5 stars based on 7499 reviews [110]. In the category 'easy to use' Salesforce has 4 out of 5 stars [110]. The review rated as second most helpful states: "Salesforce is not always simple but once it is understood it can be manipulated to fit any business's needs", this is in line with my personal experience with Salesforce, many customization options make the software complex [110].

Commitment to Usability

When searching for the usability efforts of Atlassian it was not difficult to find some sources to show their commitment to usability. For Salesforce this was more difficult and although they claim to make every easy it is difficult to find some concrete examples that show a commitment to usability.

Trailhead – The Fun Way to Learn Salesforce

Salesforce has created an online learning center which they call Trailhead. If offers a large amount of training materials and tutorials. They claim Trailhead makes learning fun and they have certainly made an effort in support of this claim. Everywhere on Trailhead the cute Salesforce mascots can be found. For examples they have chosen funny concepts such as in the tutorial to build a galactic battle station app, which teaches general skills on how to build and customize in Salesforce using the galactic battle station as an example [111].
Every tutorial ends with a quiz which will earn you points and by completing tutorials you can gain badges. If you obtain enough badges you go up in rank, these gamification elements are intended to motivate the

user to take the time and make the effort to learn how to use the software. I did some of the tutorials and they were very doable, however they are also time consuming and you need to take quite a lot of them to really understand the software well enough to do your own customizations and build your own apps.

Developer Experience

Salesforce also puts effort into the DX, the developer experience. This entails making it easier and more pleasant for developers to create applications for the platform. User experience has become a more popular term and focus area in the last years and with the rise of platforms we can probably expect to be hearing more about not only the end user experience but also the platform developer experience.

Personal Experience with Salesforce

During my research for this thesis I also signed up for a free trial of Salesforce to gain a better understanding of the software and to get some firsthand experience. Overall, I found the software is mostly characterized by the ability to customize it. This however is also the great difficulty when getting started. This is definitely not software you can just start using in one afternoon. I spend several hours on following tutorials and in the end, I was able to make some custom fields and parts on an application, but it would have taken me several days to truly learn how to build applications with the software. The usability of a premade application is much better, although the pages do not always look very attractive, they are mostly just rows of data and not very visually appealing. They claim to be easy and fun but it takes quite an effort to learn enough to be able to get started.

Dimension: User/Partner Attraction Strategy

The image that Salesforce tries to project to the world is one of being warm and fuzzy. This is featured in many of their official communications. They are participating in the “1% pledge” (see below) and they have created cute looking characters featuring everywhere on their website. The community serves to make Salesforce appear like a welcoming, helpful and friendly community. This concept of being easy and friendly, inviting you to be part of their Ohana, is the core of their marketing strategy.

Doing Well and Doing Good as a Marketing Strategy

During the conference there was also attention paid to the 1% pledge: the 1-1-1 principle. Salesforce gives 1% of revenue, 1% of their time and 1% of their products to organizations who are trying to make the world a better place. Atlassian has also adopted this initiative. Such an initiative, especially when adopted by a large company such as Salesforce can make a tremendous difference in the world for the better, and it is also good marketing. In all Salesforce communication a friendly and happy experience is created. Even the Salesforce mascots are very cute and cuddly creatures. Showing the world how noble you are may trigger people to choose the Salesforce platform over another platform. During a presentation about the Salesforce partnership with Warchild a video from Warchild was shown about children in war zones. After the video the crowd of over 2000 people stayed dead silent for a while as it was quite a heartbreaking sight to see children living under such conditions, until the keynote speaker finally broke the silence. People care about all the bad things that are still happening in our world and want to help those in need. Therefore showing as a company how you are making a difference in the world, for example by supporting Warchild can be a powerful selling point and a powerful marketing trick playing on human emotions and the innate human desire to contribute to making the world a better

place. Is another Salesforce initiative in line with their positive image campaign is the Buddyforce project in which they help refugees integrate into society and find jobs [112].
In both their noble efforts to make the world a better place and their branding and communication it is clear that Salesforce is working hard create very positive and friendly associations with the company. I am usually a typically Dutch extremely level-headed person, but even I felt a little soft, warm and fuzzy inside hearing about all their positive efforts during the Salesforce conference. Warm and fuzzy feelings for a company are a great competitive advantage and driver of sales.

Positive Company Image to Attract Users and Partners

Another way in which Salesforce creates positive associations with their company is by giving attention to their equality policies. They claim to have invested 9 million last year to create equal pay for equal work for all their employees and gave attention to the VHTO (a Dutch association that stimulates young girls to consider a career in technology or science) and girls who code initiatives during their conference. Another typical aspect of the Salesforce branding is that a user is not called a user, but a trailblazer, fitting with the cuddly creatures by the campfire that are often showed on the website [113]. In this figure we see the Salesforce mascots Codey the bear, Einstein, a goat and Astro enjoying some marshmallows by the campfire. See figure 54 of cuddly creatures by the campfire: [113]

Figure 54: Equality logo Salesforce

Using Marquee Customers to Attract New Customers

Another key aspect of the Salesforce marketing strategy is to attract users with users. This was often referred to as a 'marquee user' in the platform strategy literature. In blogs and on the Salesforce website many references can be found to companies which use Salesforce and their success stories, as a way of convincing potential users that the product has value. If NASA uses it, it must be good, is an example of this type of thinking they try to stimulate. On the product pages success stories are prominently featured and there is an entire database with several hundred entries of customer success stories [114]–[116]. Some of the Salesforce customers mentioned on the Salesforce website are: Michelin, VMWare, MAERSK, Adidas, Nestlé, T-Mobile, Canon, Philips, Toyota, Vodafone, L'Oréal, and Spotify [116].

Dimension: Pricing

Pricing for Users

Salesforce is very open about their pricing, all different pricing plans including details of which functionality is included or excluded in each different plan can be found on their website. Salesforce uses tiered pricing giving four different options [117]. The options range from a basic version of the software with extremely limited functionality to the unlimited option at twelve times the price of the basic version but with a great deal more functionality [117]. Salesforce uses nudging, a concept which was explained earlier in this thesis in the paragraph about Ambidexterity in Openness Strategy, in their pricing strategy. By giving a 'recommended' or 'most popular' option the cognitive effort required to make a choice is reduced and users are therefore more likely to choose this option.

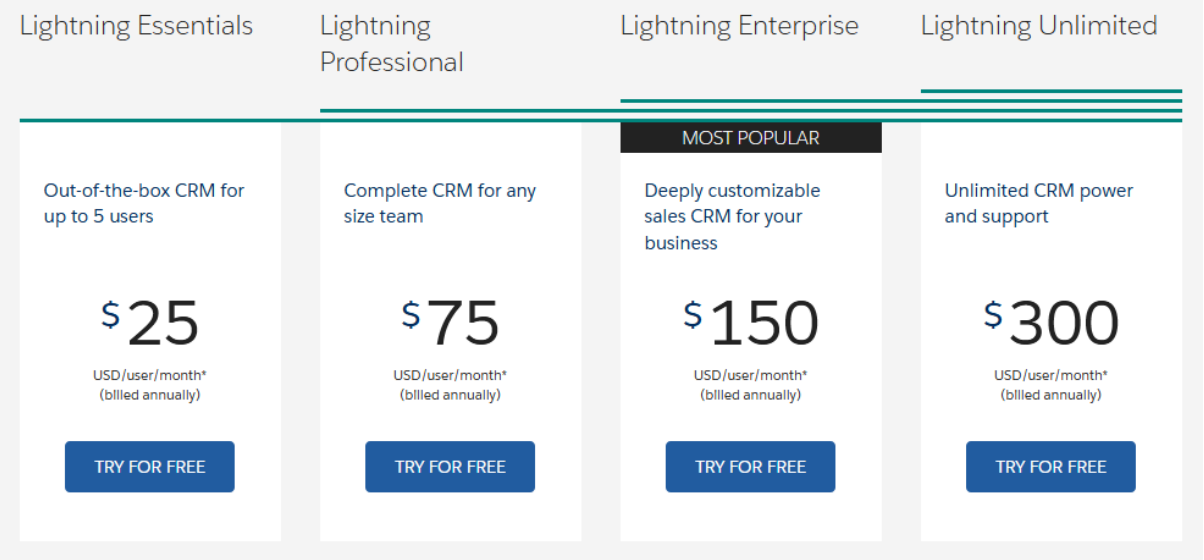


Figure 55: Prices Lightning for users

By offering a relatively cheap option users can get to know the software and experiment with it without too much financial risk. At the same time users will become dependent on the software and realize that with the basic option the functionality is very limited and to grow their business further the intention is that users will than move up to a more expensive plan resulting in more profit for Salesforce. However, compared to Atlassian the prices are much higher, suggesting that Salesforce focusses on a more limited market of those companies who can afford to pay up to 300 dollars a month for a single user. The low prices and free trial may attract customers, but to get all the functionality the price of 300 dollars a month per user is likely to scare those users off. Even the recommended plan of 150 dollars per users per month still really adds up if the software is used for several years by several users, which is the intended use of the software. So where with Atlassian the financial risk was low, with Salesforce companies really should consider if the use of Salesforce will increase their profit enough to justify such high monthly subscription fees.

Free Trial

The price for up to 5 users is low enough to pose little financial risk, but Salesforce also offers a free trial for individual users. The free trial lasts 30 days and requires the user to submit personal and company information to gain access.

Figure 56: Try for free button

Pricing for Developers

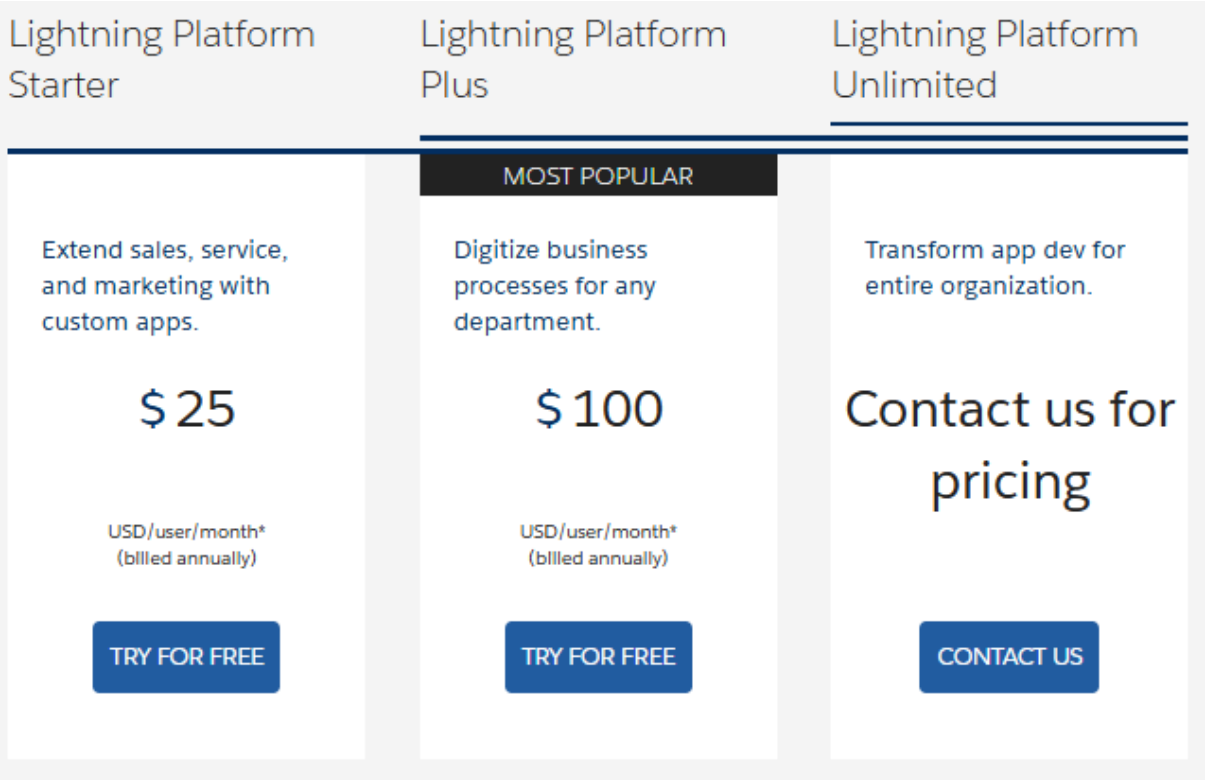


Figure 57: Prices Lightning for developers

Salesforce offers separate pricing for the Lightning development platform [118]. The cheap version offers low risk but has little functionality. To really make interesting applications with a large amount of functionality the more expensive versions have to be bought. The low price is an introduction to the platform which will make most developers quickly realize that to really achieve their goals they must switch to a more expensive version.

Salesforce has clearly chosen for free trials and low introduction prices to invite users to try out the software, however, to benefit from the full functionality a significantly higher price must be payed, so for long term and average use Salesforce clearly does not have a low-price strategy.

Dimension: Governance

Openness

Where Atlassian is very open about governance issues for partners of the platform it proved much more difficult to find similar information for partners of the Salesforce platform. After quite a long time searching I was able to find some sources on governance issues however it felt all felt a big unstructured and random. I could find some information about issues, but if I were a Salesforce partner wanting to sell apps in their platform it would take considerable effort to get all the information needed to start doing that where with Atlassian this was doable in a few hours including reading all the information, I feel for Salesforce it would likely take several days and end with asking the community or Salesforce itself because it was very difficult to find out things about rules, regulation, guidelines, policies and other such governance documentation. So in terms of openness Salesforce is quite closed, information and resources are difficult to find.

Salesforce claims to be transparent to their Ohana [119]. They say they communicate openly and honestly [119]. With the Trailblazer community anyone can join the conversation and share their ideas [119]. Although the community does indeed offer the option to share ideas and talk to other users, this is not the same as transparency about company decisions and having the rights to influence company decisions. Based on my experience I would say they claim more transparency than they actually give based on how difficult it was to find information about their governance, so I would score Salesforce as being on the closed side of the closed versus open spectrum. I would also conclude that although they offer the option to give your opinion, this is a purely the submitting of ideas; partners and customers have no vote in the final decision making, so the governance is centralized.

Distribution of Decision Rights

Salesforce markets themselves with their Ohana, however the company is still very central in their decision rights distribution. They do want to extend their ecosystem, but I could not find evidence of decision rights being given to the ecosystem.

When digging deeper finally governance documents such as a requirements checklist with a very long list of requirements, terms and conditions of use, privacy guidelines and other such documents can be found[120]–[122]. More practical information as how to offer apps, how to get your app approved, how sales and payments will take place and such I was not able to find after several hours of searching. I suspect they do have rather strict governance, however, unlike Atlassian they are not very open about this and for an external person it is practically impossible to get a decent understanding of how open or closed the platform is and how decision rights are distributed. It seems that much of the relevant governance information can only be obtained by registered partners and that the information is not publicly available.

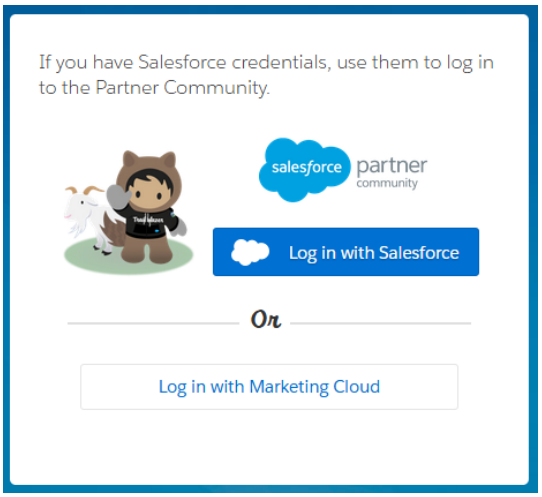


Figure 58: Salesforce User Community

Dimension: Evolution

In twenty years Salesforce has grown from a company of two people to a major company with over 15,000 companies using their software. Salesforce gained platform leadership in the CRM market rather quickly because of their efficient platform which reduced the required investment for their customers [30]. After an initial growth spurt Salesforce was in decline in 2006 due to too much hierarchical bureaucracy, they then changed their traditional management model to a more radical and innovative business model which has contributed to their continued growth and success [85].

Innovation and Growth through Acquisition

The evolution of Salesforce is characterized by the many other companies they have bought [85]. Innovation at Salesforce often has come from buying disruptive companies and adding their innovative solutions to the Salesforce value proposition [85]. Their strong innovation strategy has earned them several awards over the years [38].

Figure 59: Awards Salesforce

Future Ambition

Salesforce has a serious ambition, they have set the goal that for every 1 $ Salesforce makes the ecosystem will make $ 5.18 by 2022. This goal supports the assumption that the core strategy of Salesforce is to grow their Ohana, to not just take care of themselves but create a profitable ecosystem around them.

Achievements

- In 2016 there were 3,820,549 apps installed by Salesforce customers [123]
- Global strategic partners include big consultancy companies such as Deloitte, Accenture, Capgemini and PWC [123]
- Over 30 companies have been acquired by Salesforce [123]
- Growth from 2 employees to over 16,000
    - 2005: 767 employees,
    - 2010: 3,969 employees,
    - 2015: 16,000 employees [123]

Key Points of the Growth Strategy

The Salesforce growth strategy is in large part based around the concept of making it easy with no risks for potential customers to try out the software [124]. Salesforce offers free trials which work as a web app so no software installation is required [124]. Potential buyers can request the free trial without having to submit credit card information, making the threshold of trying out the platform low [124]. This low threshold has supported Salesforce in achieving their rapid growth.

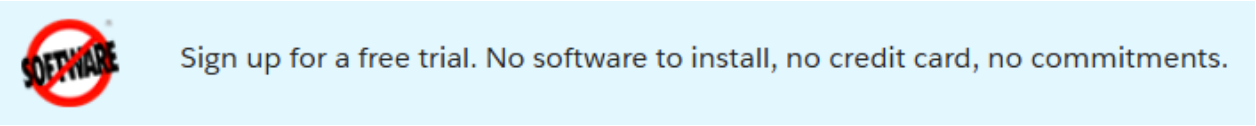


Figure 60: Trial without installation

Steady Revenue Increase

Based on the steady increase of yearly revenue since 2010, as seen in the table below, it could be concluded that the strategic choices that have been made by Salesforce have led to the desired result.

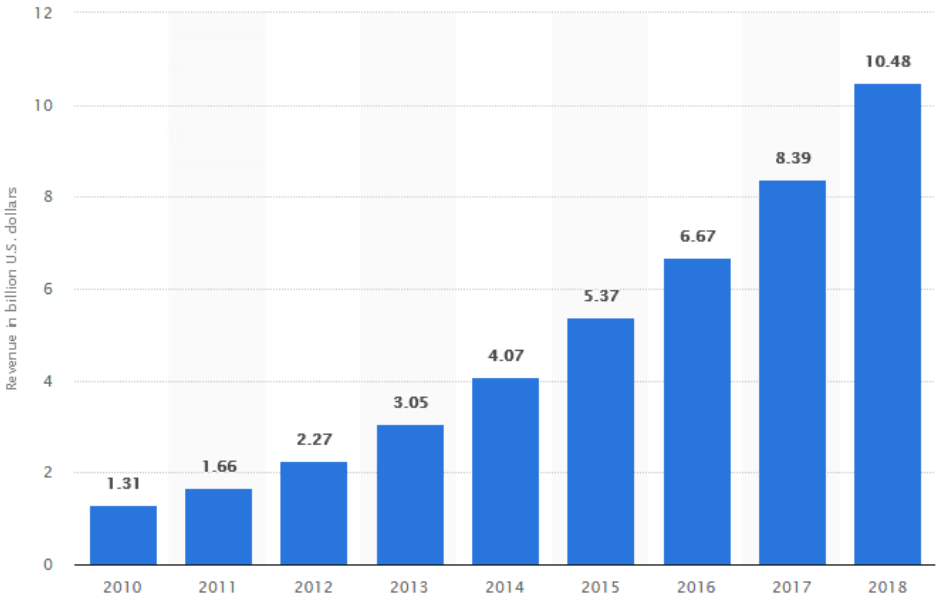


Figure 61:Revenue Increase [125].

Salesforce Case Study Analysis and Conclusion

In the section the correctness and suitability of the platform strategy framework is analyzed.

Accurateness of the Dimensions

For Atlassian it was easier to find information and thereby validate the dimensions. Because of a particular Salesforce dimensional choice, being rather closed in their governance, it proved more difficult to get enough information to validate some of the dimensions. In the end for each dimension the strategic choices that have been made could be found. During the execution of the case study no new dimensions arose which should have been included in the framework but are currently not present in the framework.

Relationships between the Dimensions and the Core of the Strategy

Salesforce's strategic choices are made in the different dimensions based on a higher-level strategy. This high-level strategy has in turn a strong relation with the type of users and partners Salesforce attracts. The higher-level strategy and choices in the dimensions have also shown to be a determinant for how the platform has evolved through the years.

The separation into two sets (set1: trust, governance, usability and pricing, set 2: users and partners, and evolution) of dimensions with a platform strategy core between them as discovered in the case study Atlassian holds up in light of the Salesforce case study.

Suggestions for Improvements in the Framework

The case study seems to suggest that the proposed changes in the analysis of the Atlassian case study are also indeed suitable based on this case study. Based on the discoveries made during the case studies it seems in order to create a revised version of the platform strategy framework. This new version of the framework will reflect the lessons learned from the case studies and the result will be a more suitable platform strategy framework for Tersof which also offers a greater chance of possibly being usability for other companies in similar situations as well. The revised version of the platform strategy framework is presented in the next chapter.

7. Revised Platform Strategy Framework

Based on lessons learned from the case studies and the discoveries made during the execution of the case studies the platform strategy framework has been revised and a final version has been made.

Original Platform Strategy Framework

The figure below shows the first version of the framework.

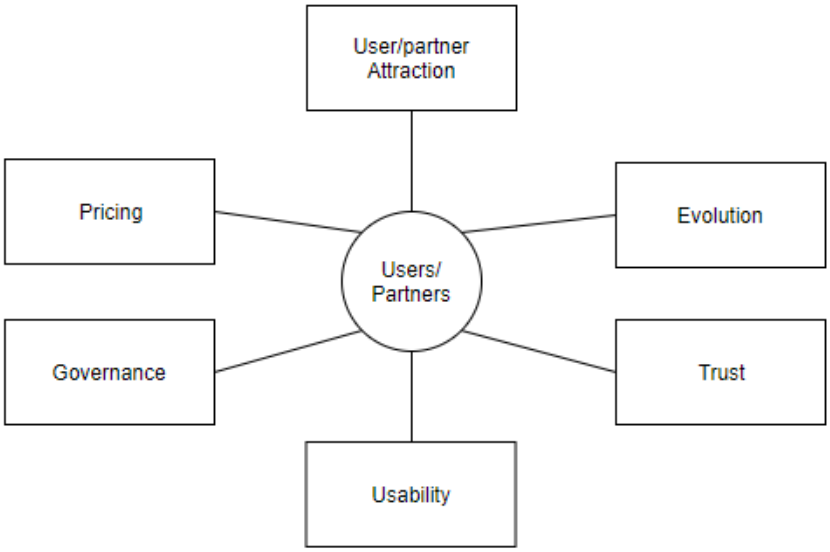


Figure 62: The first version of the platform strategy framework.

Revised Version

The revised version still adheres to the same choice -> result -> goal structure, however the dimensions are presented differently to highlight the differences in the dimensions and the relations between the different components. The revised version of the framework can be seen on the next page.

The revised framework supports three different viewpoints or starting off points to approach platform strategy. The boxes are connected with lines instead of directional arrows to illustrate the fact that the framework can be used and approached from different directions. The platform strategy can be developed with as a starting out point certain choices which are made in the dimensions (approach 1), a certain strategy core that is deemed most important (approach 2) or by being led by the type users, partners and evolution that is desirable (approach 3).

Approach 1: Dimensions are Leading

When this platform strategy approach is chosen the dimensions are leading and will determine in turn what the core of the strategy will be, and which users and partners will be attracted to the platform and how the platform will evolve. For example a company may choose to go for a low-price strategy with loose governance, limited efforts on usability and only minimal trust requirements. Based on the choices the core of the strategy would probably be to grow quickly and to get as many users and partners to join the platform soon after starting out without much concern for longevity. This will then in turn lead to attracting many users and partners quickly. However, in terms of evolution most users might quickly

leave the platform again because of the limited usability and the uncontrolled growth of low-quality complements.

Approach 2: The Platform Strategy Core is Leading

When this approach is used the company first decides on a strategy core which is then used to guide the choices to be made for the different dimensions. The core of the platform strategy will lead to a certain type of users and partners being attracted to the platform and it will determine how the platform will evolve. For example when looking at Atlassian such an approach becomes apparent. The core of their strategy is to not #@!% their customers. In order to achieve this they have made their choices in the dimensions such as the high commitment to trust, openness and usability. These choices have remained the same throughout the evolution of the platform and will continue to guide the evolution of the platform as future choices that are made will always be made with the strategy of not #@!%ing the customers in mind. Because of their strategy core they have attracted a loyal user base and many partners.

Approach 3: The Users, Partners and Evolution are Leading

Finally a company can also start platform strategies deliberations by looking at which users and partners they want to attract and how they wish to grow and evolve in the future. Tersof for example has set a very clear goal or reaching 1.000.000 users and growing to a more international position. So they could start by choosing which markets of users and partners they wish to reach, how they will facilitate this evolution, for example by an online self-service webstore, high usability and free trials. So then the third column is leading and determines the strategy core and which choices should be made in the dimensions to reach those desired users and partners and to evolve in the desired manner.

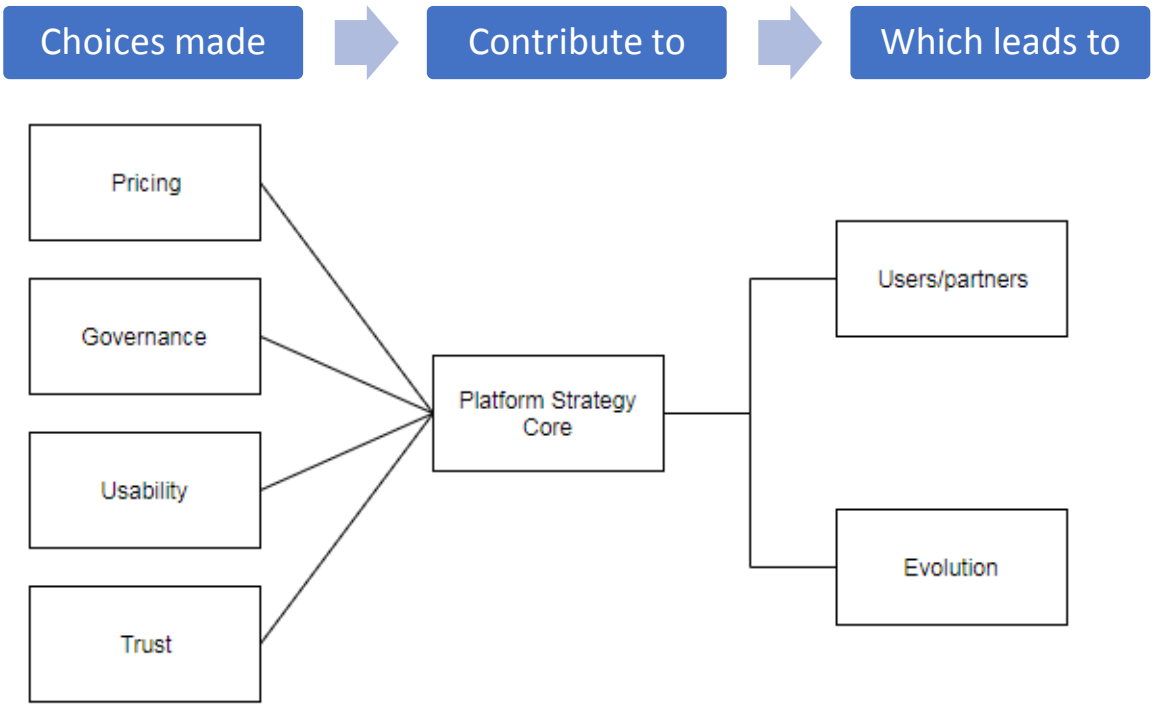


Figure 63: The revised platform strategy framework

8. Conclusion

Goal of the Thesis

The goal of this thesis was to develop a framework for platform strategy. The framework has been developed by looking at platform strategies which were found in literature, ranging from scientific literature to business and management literature. The found and studied literature on platform strategies offered many possible critical success factors and many possible strategy dimensions which contribute to achieving these critical success factors and thereby to the overall goal of the platform strategy, the platform strategy core. The platform strategy framework has been developed with Tersof in mind. The combination of the Tersof context and the platform literature has led to the framework. To validate the framework two case studies have been done. For the case studies platform business have been used which are both very successful and which have built up large customer bases throughout their (almost) 20 years of existence, strongly suggesting that they have made good strategic choices during their lifespan and that they take their platform strategy seriously. The suggested platform strategy model is applied to the case studies to see if these case study companies do indeed place great emphasis on these dimensions as a way to achieve their final goal. Based on the results of the case studies the platform strategy framework has been revised and restructured and is considered validated.

Framework Validation through Case Studies

The case studies have illustrated that the 6 selected dimensions are prominently at the heart of platform strategies of successful platform strategy companies. Both companies have given a fair amount of attention to these dimensions and have made specific choices in regard to contribution to their respective goals and overall company strategies and the platform strategy core. In Atlassian for example it was very clear that all their efforts in the different strategy dimensions seem to lead to the goal of not F#*@ing the customer, being fair, open and doing the best possible work for their customers. Salesforce on the other hand is not quite as open but they do not boast this as a company goal, their goal is to be so customizable almost any company could use the software in some way and they have accepted the complexity that comes with such customization by focusing on their online community and tutorials. The "Ohana" or family feeling is the end goal of the strategy. There are friendly looking mascots everywhere, inviting tutorials and a lot of marketing attention for the giant community, or as they like to see it, family, you get when you join Salesforce. The website does not focus so much on being open about technical issues such as Atlassian, but it focusses more on the image they want to project, which is again reflected in the decisions they have made for the different platform strategy dimensions. So it can be concluded that the strategic choices made in the dimensions of the framework are strongly related to the platform strategy core which in turn has a very strong relation to the partners and users which are attracted to the platform and the way the platform evolves. The final revised version of the platform strategy framework reflects these relationships.

Conclusions for Tersof

Everything in this thesis has been written with Tersof in mind. The research goal is intended to contribute to future developments in the company. During the development of a platform strategy Tersof can use the framework as a guideline and leading structure at the heart of their platform strategy development. By making choices in the different dimensions of the framework Tersof will be able to contribute to the core of their platform strategy and this core strategy will lead to certain users and partners joining the platform. The core of the strategy will also have a large role in how the platform develops and evolves over time. Tersof has a strong goal to grow to 1,000,000 users. I would say for Tersof this is the core of their strategy. Part of that strategy core is that they want to achieve this growth by being extremely easy to use, having an intuitive design, and by making it easy to try the software through quick set ups and low priced or free trials just to name of few examples of choices that could be made in the different dimensions by the company to contribute to the core of their platform strategy and thereby to the larger company wide long term strategic goals.

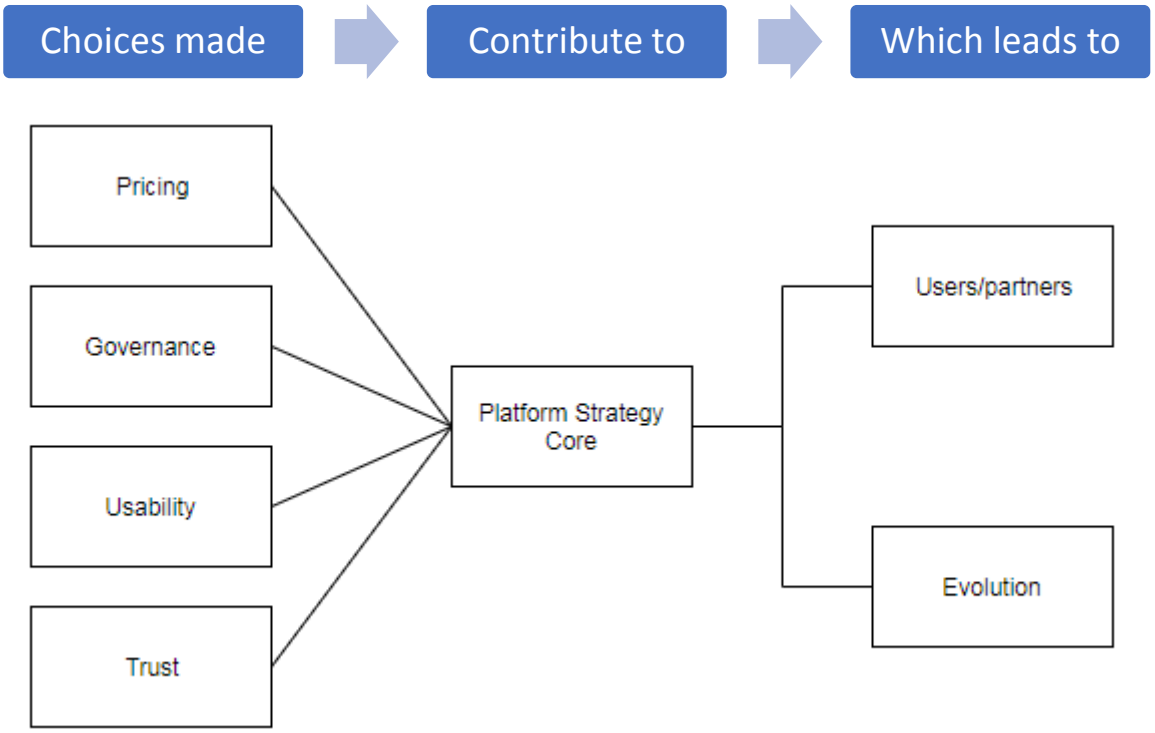


Figure 64 Revised final version of the platform strategy framework

References

[1] S. A. Ficker and H.-B. Kittlaus, *Software Product Management: The ISPMA-Compliant Study Guide and Handbook*. 2017.

[2] M. Kreijveld, “De Kracht van Platformen,” 2014.

[3] H. Doorewaard and P. Verschuren, *Designing a Research Project*, Second. The Hague: Eleven International Publishing, 2010.

[4] C. Gronroos, “Value co-creation in service logic : A critical analysis,” *Mark. Theory*, vol. 11, no. 3, pp. 279–301, 2011.

[5] D. Zwick, S. K. Bonsu, and A. Darmody, “Putting Consumers to Work,” *J. Consum. Cult.*, vol. 8, no. 2, pp. 163–196, 2008.

[6] M. Archpru, P. P. Maglio, and S. L. Vargo, “On value and value co-creation : A service systems and service logic perspective,” pp. 145–152, 2008.

[7] B. Edvardsson, B. Tronvoll, and T. Gruber, “Expanding understanding of service exchange and value co-creation : a social construction approach,” pp. 327–339, 2011.

[8] P. Frow, A. F. Payne, and K. Storbacka, “Managing the Co-creation of Value,” *J. Acadamy Mark. Sci.*, pp. 83–96, 2008.

[9] M. Kenney and J. Zysman, “The next phase in the digital revolution,” *Commun. ACM*, vol. 61, no. 2, pp. 54–63, 2018.

[10] S. Dillon, F. Schönthaler, and G. Vossen, *The Web at Graduation and Beyond*. 2017.

[11] C. Krintz, “Platform-as-a-Service (PaaS),” in *Encyclopedia of Database Systems*, New York, NY: Springer New York, 2017, pp. 1–2.

[12] R. Azevedo, T. Batista, A. Monteiro, J. S. Pinto, and C. Teixeira, “The building blocks of a PaaS,” *J. Netw. Syst. Manag.*, vol. 22, no. 1, pp. 75–99, 2014.

[13] T. Catlin, J.-T. Lorenz, J. Nandan, S. Shirish, and A. Waschto, “Insurance beyond digital: The rise of ecosystems and platforms,” *Insur. Pract.*, no. January, 2018.

[14] E. J. Altman and M. L. Tushman, “Platforms, Open/User Innovation, and Ecosystems: A Strategic Leadership Perspective,” 2017.

[15] K. Täuscher and S. M. Laudien, “Understanding platform business models: A mixed methods study of marketplaces,” *Eur. Manag. J.*, 2016.

[16] A. Tiwana, *Platform Ecosystems*. Elsevier Inc., 2014.

[17] C. Molenaar, *De Kracht van de Platform Strategie - Het is Buigen of Barsten*. Boom, 2017.

[18] M. Kreijveld, *De Plug & Play Organisatie - Winnen in een Digitale Wereld*. Vakmedianet, 2016.

[19] M. Schreieck and M. Wiesche, “The Platform Owner’s Challenge to Capture Value – Insights from a Business-to-Business IT Platform,” in *Thirty Eighth International Conference on Information Systems*, 2017, pp. 1–19.

[20] F. Hintermann, L. Morvan, and M. Vazirani, “Five Ways to Win with Digital Platforms,” 2016.

[21] A. Heckmann, C.S., Maedche, "Towards Platform based Enterprise systems - Conceptualization and research directions," in *Lecture Notes in Business Information Processing*, 2011, pp. 207–211.

[22] E. Lamarre and B. May, "Making sense of Internet of Things platforms," *McKinsey&Company*, no. May, pp. 1–10, 2017.

[23] M. Bonchek and S. P. Choudary, "Three Elements of a Successful Platform Strategy," *Harv. Bus. Rev.*, 2013.

[24] Z. Church, "Platform Strategy, Explained," 2017.

[25] K. Panetta, "Every Organization Needs a Digital Platform Strategy," *Smarter with Gartner*, 2016. [Online]. Available: https://www.gartner.com/smarterwithgartner/every-organization-needs-a-digital-platform-strategy/. [Accessed: 03-May-2018].

[26] J. Cenamor, G. Parker, M. Van Alstyne, and X. Wan, "Unraveling platform strategies: A review from an organizational ambidexterity perspective," *Sustain.*, vol. 9, no. 5, pp. 1–18, 2017.

[27] "8 Ways to Launch a Succesful Digital Platform," *Digital Business*, 2016. [Online]. Available: https://executive.mit.edu/blog/8-ways-to-launch-a-successful-digital-platform#.WusAvYhubIU. [Accessed: 03-May-2018].

[28] S. von Engelhardt, L. Wangler, and S. Wischmann, "Characteristics and Success Factors of Digital Platforms," 2017.

[29] B. Edelman, "How to Launch your Digital Platform," *Harv. Bus. Rev.*, no. April, pp. 90–97, 2015.

[30] S. M. Lee, T. Kim, Y. Noh, and B. Lee, "Success Factors of Platform Leadership in Web 2.0 Service Business," *Serv. Bus.*, vol. 4, no. 2, pp. 89–103, 2010.

[31] G. T. Ozer and E. G. Anderson Jr., "Innovation and Breaching Strategies in Multi-Sided Platform Markets: Insights from a Simulation Study," *Thirty Sixth Int. Conf. Inf. Syst.*, pp. 1–21, 2015.

[32] K. S. Staykova and J. Damsgaard, "Platform Expansion Design as Strategic Choice: the Case of Wechat and Kakaotalk," *Twenty-Fourth Eur. Conf. Inf. Syst.*, 2016.

[33] A. S. Santoso and S. Wahyuni, "Maximizing Strategic Alliances in the Multi-sided Platform Firms," *Int. J. Bus.*, vol. 23, no. 1, pp. 26–52, 2018.

[34] M. Berger, T. Hess, and A. Stefi, "What influences platform provider's degree of openness? - Measuring and analyzing the degree of platform openness," *Lect. Notes Bus. Inf. Process.*, pp. 258–272, 2014.

[35] K. Boudreau, "Open Platform Strategies and Innovation: Granting Access vs. Devolving Control," *Manage. Sci.*, vol. 56, no. 10, pp. 1849–1872, 2010.

[36] L. Margalit, "The Psychology of Choice," 2014. [Online]. Available: https://www.psychologytoday.com/us/blog/behind-online-behavior/201410/the-psychology-choice. [Accessed: 11-Apr-2018].

[37] A. Tugend, "Too Many Choices: A Problem That Can Paralyze," *The New York Times*, 2010. [Online]. Available: https://www.nytimes.com/2010/02/27/your-money/27shortcuts.html. [Accessed: 11-Apr-2018].

[38] "Salesforce - About Us." [Online]. Available: https://www.salesforce.com/company/about-us/. [Accessed: 01-Aug-2018].

[39] "Atlassian - Company." [Online]. Available: https://www.atlassian.com/company. [Accessed: 02-Jul-2018].

[40] "Atlassian Home Page." [Online]. Available: https://www.atlassian.com/?_ga=2.256044107.158553986.1521466437-274383367.1521466437. [Accessed: 20-Mar-2018].

[41] "Products for Teams, from Startup to Enterprise." [Online]. Available: https://www.atlassian.com/software. [Accessed: 20-Mar-2018].

[42] C. Clark, "20 Great ITSM Add-ons for Jira Service Desk," 2017. [Online]. Available: https://www.atlassian.com/blog/jira-service-desk/19-great-itsm-add-ons-jira-service-desk?_ga=2.45213924.158553986.1521466437-274383367.1521466437. [Accessed: 22-Mar-2018].

[43] J. Leyser, "World, Meet Jira Studio. Jira Studio, the World.," 2008. [Online]. Available: https://www.atlassian.com/blog/archives/world_meet_jira?_ga=2.251193414.158553986.1521466437-274383367.1521466437. [Accessed: 22-Mar-2018].

[44] J. Nolen, "Greetings Developer, Welcome to the Future of Atlassian Add-ons," 2013. [Online]. Available: https://www.atlassian.com/blog/archives/25461?_ga=2.41117282.158553986.1521466437-274383367.1521466437. [Accessed: 20-Mar-2018].

[45] "Company Values." [Online]. Available: https://www.atlassian.com/company/values. [Accessed: 01-Aug-2018].

[46] D. Olinsky, "Joining the Atlassian Family," 2016. [Online]. Available: https://www.atlassian.com/blog/statuspage/joining-the-atlassian-family?_ga=2.154217976.158553986.1521466437-274383367.1521466437. [Accessed: 19-Mar-2018].

[47] D. Radigan, "Getting the most out of JQL with Atlassian Marketplace add-ons," 2017. [Online]. Available: https://www.atlassian.com/blog/jira-software/jql-the-most-flexible-way-to-search-jira-44?_ga=2.211439573.158553986.1521466437-274383367.1521466437. [Accessed: 19-Mar-2018].

[48] R. Manalang, "Building an Ecosystem for Hackers," 2011. [Online]. Available: https://www.atlassian.com/blog/archives/building-an-ecosystem-for-hackers?_ga=2.41029602.158553986.1521466437-274383367.1521466437. [Accessed: 28-Mar-2018].

[49] "Marketplace FAQ." [Online]. Available: https://www.atlassian.com/licensing/marketplace. [Accessed: 22-Mar-2018].

[50] R. Whitbeck, "Announcing the Atlassian platform for developers roadmap," 2017. [Online]. Available: https://developer.atlassian.com/blog/2017/12/announcing-atlassian-platform-developers-roadmap/. [Accessed: 28-Mar-2018].

[51] "Getting Started," 2017. [Online]. Available: https://developer.atlassian.com/server/framework/atlassian-sdk/. [Accessed: 04-Apr-2018].

[52] "Working in an IDE," 2017. [Online]. Available: https://developer.atlassian.com/server/framework/atlassian-sdk/working-in-an-ide/. [Accessed: 10-Apr-2018].

[53] “Atlassian Platform Common Components,” 2017. [Online]. Available: https://developer.atlassian.com/server/framework/atlassian-sdk/atlassian-platform-common-components/. [Accessed: 04-Apr-2018].

[54] M. Mancini, “Atlassian Marketplace Surpasses $250M in Sales, Introduces new Partner Add-ons,” 2017. [Online]. Available: /blog/add-ons/atlassian-marketplace-surpasses-250m-in-sales?_ga=2.10150901.158553986.1521466437-274383367.1521466437. [Accessed: 22-Mar-2018].

[55] “About the Atlassian Marketplace,” 2018. [Online]. Available: https://developer.atlassian.com/platform/marketplace/. [Accessed: 19-Mar-2018].

[56] D. Meyer, “Managing Inventory in Jira: a Plugin Approach,” 2014. [Online]. Available: https://www.atlassian.com/blog/archives/managing-inventory-jira-plugin-approach?_ga=2.8974197.158553986.1521466437-274383367.1521466437. [Accessed: 20-Mar-2018].

[57] D. Cook, “You Can Do That,” 2009. [Online]. Available: https://www.atlassian.com/blog/archives/you_can_do_that_part_2_of_3?_ga=2.248245063.158553986.1521466437-274383367.1521466437. [Accessed: 20-Mar-2018].

[58] “Security Practices.” [Online]. Available: https://www.atlassian.com/trust/security/security-practices. [Accessed: 27-Jul-2018].

[59] “Atlassian Compliance.” [Online]. Available: https://www.atlassian.com/trust/compliance. [Accessed: 01-Aug-2018].

[60] “Security at Atlassian.” [Online]. Available: https://www.atlassian.com/trust/security. [Accessed: 31-Aug-2018].

[61] “Security Bug Fix Policy.” [Online]. Available: https://www.atlassian.com/trust/security/bug-fix-policy. [Accessed: 01-Aug-2018].

[62] “Privacy Policy.” [Online]. Available: ttps://www.atlassian.com/legal/privacy-policy. [Accessed: 01-Aug-2018].

[63] “Gartner - Vendor Atlassian.” [Online]. Available: https://www.gartner.com/reviews/market/workplace-social-software/vendor/atlassian. [Accessed: 31-Jul-2018].

[64] “Gartner - Atlassian - Ratings.” [Online]. Available: https://www.gartner.com/reviews/market/workplace-social-software/vendor/atlassian/ratings. [Accessed: 31-Jul-2018].

[65] “Capterra - Jira.” [Online]. Available: https://www.capterra.com/p/19319/JIRA/. [Accessed: 31-Jul-2018].

[66] “Atlassian Research Group - We want to learn from you.” [Online]. Available: https://www.atlassian.com/research-group. [Accessed: 30-Jul-2018].

[67] “Inclusion through UX.” [Online]. Available: http://go.optimalworkshop.com/world-usability-day-2017. [Accessed: 31-Jul-2018].

[68] “Usability Researcher.” [Online]. Available: https://jobs.lever.co/atlassian/17b2bde6-7ff1-4cb3-ad82-325763656282. [Accessed: 31-Jul-2018].

[69] A. Horgan, “Goodbye, Atlassian OnDemand. Hello, Atlassian Cloud!,” 2014. [Online]. Available: https://www.atlassian.com/blog/archives/goodbye-atlassian-ondemand-hello-atlassian-cloud. [Accessed: 22-Mar-2018].

[70] “Future Pricing.” [Online]. Available: https://www.atlassian.com/licensing/cloud/future-pricing. [Accessed: 20-Jun-2018].

[71] M. Knighten, “Introducing Atlassian... OnDemand,” 2011. [Online]. Available: https://www.atlassian.com/blog/archives/introducing-atlassian-ondemand?_ga=2.41029602.158553986.1521466437-274383367.1521466437. [Accessed: 22-Mar-2018].

[72] “Atlassian empowers all industries and teams - Explore customer stories.” [Online]. Available: https://www.atlassian.com/customers?page=1&sortParam=date_created desc. [Accessed: 01-Aug-2018].

[73] “CSA - Atlassian.” [Online]. Available: https://cloudsecurityalliance.org/registry/atlassian/%0A%0A. [Accessed: 01-Aug-2018].

[74] “Pricing, Paymnent and Billing,” 2018. [Online]. Available: https://developer.atlassian.com/platform/marketplace/pricing-payment-and-billing/. [Accessed: 10-Apr-2018].

[75] M. Friberg, “Confluence Plugin Mashup,” 2008. [Online]. Available: https://www.atlassian.com/blog/archives/confluence_plugin_mashup?_ga=2.146337780.158553986.1521466437-274383367.1521466437. [Accessed: 20-Mar-2018].

[76] “Contributing to the Atlassian Blog,” 2017. [Online]. Available: https://developer.atlassian.com/platform/marketplace/contributing-to-the-atlassian-blog/. [Accessed: 10-Apr-2018].

[77] “Atlassian Summit.” [Online]. Available: https://www.atlassian.com/company/events/summit. [Accessed: 22-Mar-2018].

[78] S. Peters, “5 Atlassian Summit presentations you just can’t miss,” 2016. [Online]. Available: https://www.atlassian.com/blog/archives/5-atlassian-summit-presentations-you-just-cant-miss?_ga=2.249294663.158553986.1521466437-274383367.1521466437. [Accessed: 22-Mar-2018].

[79] N. Wade, “The Evolution of Plugins for the Cloud,” 2013. [Online]. Available: https://www.atlassian.com/blog/archives/the-evolution-of-plugins-for-the-cloud-2?_ga=2.41117282.158553986.1521466437-274383367.1521466437. [Accessed: 20-Mar-2018].

[80] “App Approval Guidelines,” 2018. [Online]. Available: https://developer.atlassian.com/platform/marketplace/app-approval-guidelines/. [Accessed: 19-Mar-2018].

[81] “Atlassian Brand Guidelines for Marketplace Vendors,” 2018. [Online]. Available: https://developer.atlassian.com/platform/marketplace/atlassian-brand-guidelines-for-marketplace-vendors/. [Accessed: 10-Apr-2018].

[82] “Explore Apps for Atlassian Products.” [Online]. Available: https://marketplace.atlassian.com/?_ga=2.214041815.158553986.1521466437-274383367.1521466437. [Accessed: 22-Mar-2018].

[83] S. Farquhar, "Atlassian Acquires #1 Status Communciation Platform StatusPage," 2016. [Online]. Available: https://www.atlassian.com/blog/statuspage/atlassian-acquires-statuspage. [Accessed: 22-Mar-2018].

[84] M. Cannon-Brookes, "Atlassian - A Different Kind of Software Company," 2009. [Online]. Available: https://www.slideshare.net/mcannonbrookes/atlassian-a-different-kind-of-software-company. [Accessed: 06-Apr-2018].

[85] W. J. Müller, "An Analysis of Salesforce.com, A Cloud Based Solutions Provider, Best Known for its Customer Relationship Management (CRM) Products," *StrategyMedic*, pp. 1–11, 2012.

[86] "Salesforce Platform Overview." [Online]. Available: https://www.salesforce.com/products/platform/overview/. [Accessed: 17-Aug-2018].

[87] "Salesforce World Tour Amsterdam." [Online]. Available: https://www.salesforce.com/nl/event/amsterdam/. [Accessed: 11-Apr-2018].

[88] J. Kohner, "What is Ohana Anyway?," *Salesforce blog*, 2017. [Online]. Available: https://www.salesforce.com/blog/2017/02/what-is-salesforce-ohana.html. [Accessed: 17-Aug-2018].

[89] "Salesforce Ohana Culture." [Online]. Available: https://trailhead.salesforce.com/en/modules/manage_the_sfdc_way_ohana. [Accessed: 17-Aug-2018].

[90] "Build apps on the Lightning Platform with low-code tools and take your CRM to the next level." [Online]. Available: https://www.salesforce.com/products/platform/lightning/. [Accessed: 17-Aug-2018].

[91] "Empower everyone to build apps fast on the Lightning Platform." [Online]. Available: https://www.salesforce.com/products/platform/app-dev/lightning-platform/. [Accessed: 17-Aug-2018].

[92] "Introducing myTrailhead: Reinvent learning and enablement." [Online]. Available: https://www.salesforce.com/products/platform/products/mytrailhead/. [Accessed: 17-Aug-2018].

[93] "Amsterdam Salesforce Cake & Trailhead Saturday Morning," 2018. [Online]. Available: https://www.meetup.com/nl-NL/Amsterdam-Salesforce-Trailhead-Saturday-Morning/?scroll=true. [Accessed: 13-Apr-2018].

[94] "Blaze Your Trail." [Online]. Available: https://www.salesforce.com/company/careers/culture/. [Accessed: 17-Aug-2018].

[95] "Salesforce - Trust is a core value." [Online]. Available: https://www.salesforce.com/company/careers/culture/trust/. [Accessed: 12-Aug-2018].

[96] "Trust is a core value." [Online]. Available: https://www.salesforce.com/company/careers/culture/trust/. [Accessed: 17-Aug-2018].

[97] "Salesforce - Compliance engineered for the cloud." [Online]. Available: https://trust.salesforce.com/en/compliance/. [Accessed: 12-Aug-2018].

[98] "Stay Current on Security." [Online]. Available: https://trust.salesforce.com/en/security/stay-current-security/. [Accessed: 17-Aug-2018].

[99] "Resources for Developers." [Online]. Available: https://trust.salesforce.com/en/security/for-developers/. [Accessed: 17-Aug-2018].

[100] "Beveiligingsadvies." [Online]. Available: https://trust.salesforce.com/nl/security/security-advisories/. [Accessed: 17-Aug-2018].

[101] "Word een Security Trailblazer." [Online]. Available: https://trust.salesforce.com/nl/security/security-resources/. [Accessed: 17-Aug-2018].

[102] "Blijf op de hoogte over de veiligheid." [Online]. Available: https://trust.salesforce.com/nl/security/stay-current-security/. [Accessed: 17-Aug-2018].

[103] "Resources for Administrators." [Online]. Available: https://trust.salesforce.com/en/security/for-administrators/. [Accessed: 17-Aug-2018].

[104] "Security Health Check." [Online]. Available: https://help.salesforce.com/articleView?id=security_health_check.htm&type=5. [Accessed: 17-Aug-2018].

[105] "Security Review." [Online]. Available: https://partners.salesforce.com/s/education/appinnovators/Security_Review. [Accessed: 17-Aug-2018].

[106] "Privacy Statements." [Online]. Available: https://www.salesforce.com/company/privacy/. [Accessed: 17-Aug-2018].

[107] "Salesforce - Security." [Online]. Available: https://trust.salesforce.com/en/security/. [Accessed: 22-Aug-2018].

[108] "Gartner Peer Insights - Salesforce." [Online]. Available: https://www.gartner.com/reviews/market/sales-force-automation/vendor/salesforce. [Accessed: 13-Aug-2018].

[109] "'Initial implementation was difficult due to process complexity and 3rd party integrator.'" [Online]. Available: https://www.gartner.com/reviews/review/view/336171. [Accessed: 13-Aug-2018].

[110] "Capterra - Salesforce." [Online]. Available: https://www.capterra.com/p/61368/Salesforce/. [Accessed: 13-Aug-2018].

[111] "Build a Battle Station App." [Online]. Available: https://trailhead.salesforce.com/projects/workshop-battle-station. [Accessed: 13-Aug-2018].

[112] B. Bulkeley, "'I get by with a little help from my friends'. BuddyForce Helps Refugees Integrate into German Society," *salesforce.org*, 2017. [Online]. Available: http://www.salesforce.org/get-little-help-friends-buddyforce-helps-refugees-integrate-german-society/. [Accessed: 13-Apr-2018].

[113] K. Honjo, "Meet the Trailhead Characters: Astro, Codey, and Friends," 2017. [Online]. Available: https://www.salesforce.com/blog/2017/07/meet-trailhead-characters.html. [Accessed: 13-Apr-2018].

[114] "Strong teamwork helped G Adventures grow fast in a crowded market." [Online]. Available: https://www.salesforce.com/products/chatter/customer-stories/. [Accessed: 17-Aug-2018].

[115] "Dunkin' Donuts goes digital to help guests get back to what's important faster." [Online]. Available: https://www.salesforce.com/products/marketing-cloud/customer-stories/#!page=1.

[Accessed: 17-Aug-2018].

[116] "Explore all the Trailblazers who are succeeding with Salesforce." [Online]. Available: https://www.salesforce.com/customer-success-stories/#!page=1. [Accessed: 17-Aug-2018].

[117] "Sales Cloud Pricing." [Online]. Available: https://www.salesforce.com/products/sales-cloud/pricing/. [Accessed: 13-Apr-2018].

[118] "Lightning Platform Pricing." [Online]. Available: https://www.salesforce.com/products/platform/pricing/. [Accessed: 17-Aug-2018].

[119] "Explore the Salesfore Ohana Culture and Values." [Online]. Available: https://trailhead.salesforce.com/en/modules/manage_the_sfdc_way_ohana/units/msfw_ohana_explor_vals. [Accessed: 17-Aug-2018].

[120] "Terms of Use & Agreements." [Online]. Available: https://appexchange.salesforce.com/appxUserAgreement. [Accessed: 17-Aug-2018].

[121] "Salesforce - Requirements checklist," 2018. [Online]. Available: https://developer.salesforce.com/page/Requirements_Checklist. [Accessed: 13-Aug-2018].

[122] "Zorg met Salesforce Platform voor vertrouwen en transparantie in je gegevensverwerking." [Online]. Available: https://www.salesforce.com/nl/gdpr/platform/. [Accessed: 17-Aug-2018].

[123] "The Salesforce Economy." [Online]. Available: https://cloudtweaks.com/wp-content/uploads/2017/08/SalesforceEconomy-Simplus-2.0.1.png. [Accessed: 18-Aug-2018].

[124] "Products Overview." [Online]. Available: https://www.salesforce.com/products/ . [Accessed: 12-Aug-2018].

[125] "Salesforce.com's revenue from 2010 to 2018 (in billion U.S. dollars)." [Online]. Available: https://www.statista.com/statistics/211587/total-revenue-of-salesforce-since-2009/. [Accessed: 12-Aug-2018].

[126] M. Friberg, "'Plugin of the Month' Webinar with Ross Rowe on Bamboo," 2008. [Online]. Available: https://www.atlassian.com/blog/archives/plugin_of_the_m?_ga=2.251193414.158553986.1521466437-274383367.1521466437. [Accessed: 20-Mar-2018].

[127] A. Smith, "A shake n' Bake Recipe for Realistic Roadmaps with Jira Portfolio," 2015. [Online]. Available: https://www.atlassian.com/blog/jira-software/shake-n-bake-recipe-realistic-roadmaps-jira-portfolio?_ga=2.46788199.158553986.1521466437-274383367.1521466437. [Accessed: 03-Apr-2018].

[128] T. Crusson, "From Beta to GA: How to Build a Winning API," 2016. [Online]. Available: https://www.atlassian.com/blog/software-teams/how-to-build-an-api?_ga=2.154217976.158553986.1521466437-274383367.1521466437. [Accessed: 22-Mar-2018].

[129] L. Tay, "Atlassian - The Untold Story: How two Australian Young Guns built a $3 Billion Company, Headed for and American IPO," *Business Insder Australia*, 2014. [Online]. Available: https://www.businessinsider.com.au/atlassian-the-untold-story-how-two-australian-young-guns-built-a-company-headed-for-a-billion-dollar-ipo-2014-2. [Accessed: 06-Apr-2018].

[130] B. Darrow, "Business Software Maker Atlassian Says It's Going All-In With Amazon Cloud," 2017. [Online]. Available: http://fortune.com/2017/05/02/atlassian-goes-all-in-with-aws/. [Accessed:

03-Apr-2018].

[131] J. Silvers, “Jira Plugins and the Jira Platform,” 2007. [Online]. Available: https://www.atlassian.com/blog/archives/jira_plugins_an?_ga=2.46788199.158553986.1521466437-274383367.1521466437. [Accessed: 19-Mar-2018].

[132] M. Knighten, “Now available: online trials of Confluence and Jira,” 2009. [Online]. Available: https://www.atlassian.com/blog/archives/now_available_online_trials_of_confluence_and_jira?_ga=2.248245063.158553986.1521466437-274383367.1521466437. [Accessed: 22-Mar-2018].

[133] J. Clark, “Answers, meet Questions,” 2014. [Online]. Available: https://www.atlassian.com/blog/archives/answers-meet-questions?_ga=2.8974197.158553986.1521466437-274383367.1521466437. [Accessed: 20-Mar-2018].

[134] D. Meyer, “Atlassian Marketplace Top 10 Add-ons of 2012,” 2013. [Online]. Available: https://www.atlassian.com/blog/archives/atlassian-marketplace-top-10-add-ons-of-2012?_ga=2.49301094.158553986.1521466437-274383367.1521466437. [Accessed: 28-Mar-2018].

[135] S. Goff-Dupont, “Top 5 Plugins to Supercharge Atlassian Bamboo,” 2013. [Online]. Available: https://www.atlassian.com/blog/archives/bamboo-top-5-plugins?_ga=2.41117282.158553986.1521466437-274383367.1521466437. [Accessed: 22-Mar-2018].

[136] S. Farquhar, “Atlassian Customers: 40,000 Reasons to Celebrate,” 2014. [Online]. Available: https://www.atlassian.com/blog/archives/atlassian-customers-thank-you?_ga=2.209922261.158553986.1521466437-274383367.1521466437. [Accessed: 22-Mar-2018].

[137] “Salesforce - Products.” [Online]. Available: https://www.salesforce.com/products/. [Accessed: 18-Aug-2018].

[138] “All Products - Stage Equipment - DMX controller.” [Online]. Available: http://www.haosenlighting.com/productn.asp?ArticleID=2530. [Accessed: 02-Oct-2018].

Appendices

Appendix 1: Atlassian Products

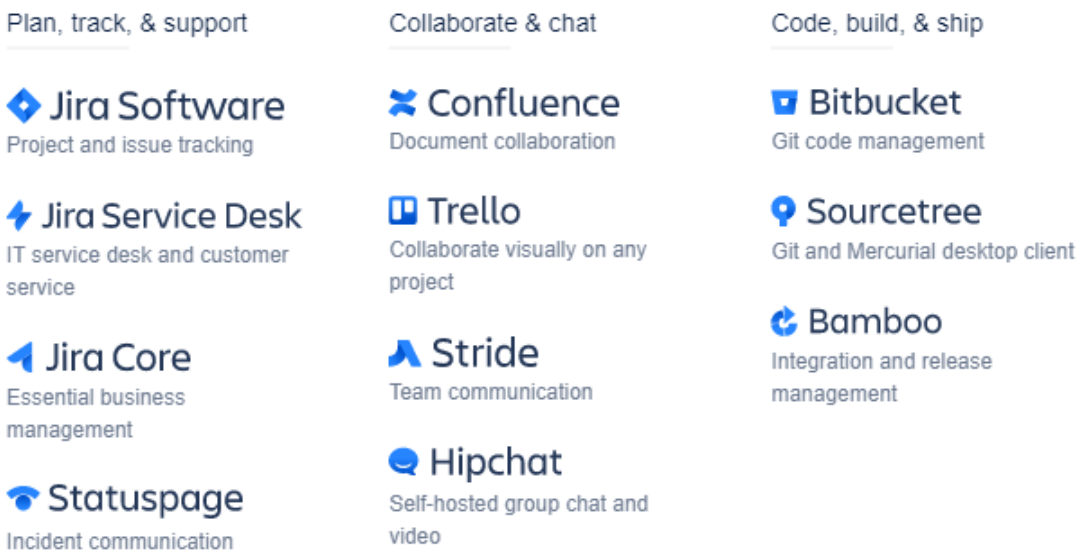


Figure 65: overview of Atlassian products, sorted by purpose [40].

Plan, Track and Support tools

Jira Software	Jira software is a tool for project management and issue tracking. [41]
Jira Service Desk	Jira Service desk is a tool for IT service desks and customer services. [41]
Jira Core	Jira core is specialized for essential business management [41]
Statuspage	Statuspage is an incident communication tool. [41]

Collaborate and Chat Tools

Confluence	Confluence is a documents collaboration tools which enables its users to organize their work, create documents and discuss those documents in a single place [41].
Trello	Trello offers boards on which projects can be organized in a flexible and visual way [41].
Hipchat	Hipchat is a tool for team communication in enterprise situations. It supports group chat, video and screen sharing [41] Hipchat is Atlassian's instant messaging platform for software teams [47]. It has a self-building buddy list, video integration and group chat [47].
Stride	Stride is a complete communication solution. It has group messaging, video meeting and collaboration tools to empower teams [41]

Tables: [41]

Code, Build and Ship Tools

Bitbucket is a code collaboration tools whish support inline comments and pull requests. It can be used to manage and share Git repositories and to build and ship software [41]

Sourcetree is Git and Mercurial desktop client for Mac and Windows [41].

Bamboo is a continuous integration tool for deployment and release management [41].
Tool for continuous integration [126]

Fisheye is a searching, monitoring and tracking tool for SVN, Git and Performance repositories [41].

Crucible is code peer review tool to find bugs and improve the quality of code [41]

Other

Crowd is a single sign-on and identify management tool for easy use, administration and integration [41].

Jira Portfolio

The perfect roadmap is data-driven, realistic and combined agile and long-term planning [127]. You can populate the Jira Portfolio backlog with epics and stories by importing them from Jira or entering them manually [127]. With Jira Portfolio you can use a default estimate of time for stories and epics [127]. The add the team velocity and Jira will automatically create a roadmap [127]. You can also create themes which reflect the strategic objectives of the organization [127]. With Jira Portfolio you can make a realistic, data-driven roadmap, use what-if scenario planning to react to change and create status overviews and reports [127].

StatusPage

StatusPage: companies can create a hosted page for communicating up-to-the-minute status to communicate the health of the service to their internal and external customers [83]. Internal status pages: communication about status across the company from the IT and operations department [83]. Redirecting customers to a status page can spare a company form getting thousands of phone calls and emails when a service is down [83]. StatusPage was acquired by Atlassian to accelerate its progress [46].

Hipchat Connect

Hipchat Connect is an API that enables developers to create apps for Hipchat that are conversational [128].

Appendix 2: Atlassian Annual Pricing

[70]

Annual Pricing

Annual User Tier	JIRA Software	JIRA Core	Confluence	Portfolio for JIRA	Capture for JIRA	Questions for Confluence	Team Calendars for Confluence
1 - 10*	$100	$100	$100	$100	$100	$100	$100
11 - 15	$1,050	$750	$750	$525	$345	$375	$375
16 - 25	$1,750	$1,250	$1,250	$875	$575	$625	$625
26 - 50	$3,500	$2,500	$2,500	$1,750	$1,150	$1,250	$1,250
51 - 100	$7,000	$5,000	$5,000	$3,500	$2,300	$2,500	$2,500
101 - 200	$11,000	$8,000	$8,000	$5,500	$3,600	$4,000	$4,000
201 - 300	$13,500	$10,000	$10,000	$6,750	$4,400	$5,000	$5,000
301 - 400	$14,500	$11,000	$11,000	$7,250	$4,700	$5,500	$5,500
401 - 500	$15,500	$12,000	$12,000	$7,750	$5,000	$6,000	$6,000
501 - 600	$16,500	$13,000	$13,000	$8,250	$5,300	$6,500	$6,500
601 - 800	$18,500	$15,000	$15,000	$9,250	$5,900	$7,500	$7,500
801 - 1,000	$20,500	$17,000	$17,000	$10,250	$6,500	$8,500	$8,500
1,001 - 1,200	$22,500	$19,000	$19,000	$11,250	$7,100	$9,500	$9,500
1,201 - 1,400	$24,500	$21,000	$21,000	$12,250	$7,700	$10,500	$10,500
1,401 - 1,600	$26,500	$23,000	$23,000	$13,250	$8,300	$11,500	$11,500
1,601 - 1,800	$28,500	$25,000	$25,000	$14,250	$8,900	$12,500	$12,500
1,801 - 2,000	$30,500	$27,000	$27,000	$15,250	$9,500	$13,500	$13,500

Appendix 3: Atlassian Time Line

2001

- Atlassian name is registered with ASIC c.

2002

- October 2002 office is set up in Sydney [129].
- Company is founded with a $10,000 credit card[84], [129].
- 2 people work at Atlassian [84].

2003

- Applications are open and extensible from day 1 [79].
- Atlassian products are on-premise only [130].

2004

- 6 people work at Atlassian [84].

2005

- San Fran Sisco office is opened [84].

2006

- First version of Atlassian Plugin SDK is released [79].
- June: 4340 customers and 50 staff in San Francisco and Sydney [129].
- Mid-year: 40 staff [84].
- Atlassian US team has 10 people [129].

2007

- Over a 100 free plugins for Jira, some developed by Atlassian, some by third parties [131].
- Poland office is opened [84].
- June: 98 employees [129].
- Revenue $ 22.5 billion [129].

2008

- Introduction Jira Studio [April 2008] [43].
- Amsterdam office is opened [84].
- 70 staff [84].
- 11,000 customers worldwide, including 40 of the world's top 50 corporations [43].

2009

- Online trials of Confluence and Jira (august) [132]. Sign up and immediately start using without downloading or installing for 30 days [132].
- 90 staff [84].
- Over $USD 110m in lifetime sales [84].
- $USD 45m sales in 2008/9 [84].
- Sales in 116 countries [84].
- Around 40,000 teams use Atlassian tools [84].
- Approximately 1 million developers are using Atlassian products [84].
- Estimated to be 1 in 6 engineers globally who use Atlassian tools every day to build software [84].
- 90 people, 12 software teams, 10 products, 4 countries, 2 deployment models [84]

2010

- First ever capital injection of US$ 60 million is accepted, from Accel Partners [129]; the largest investment ever in a software company at that time for Accel [129].

2011

- OnDemand applications are introduced [44]. OnDemand: applications that run on their custom-built platform in the cloud [44]. OnDemand: applications as a subscription service in the cloud [79]
- Atlassian answers is launched, a public question and answer community, this replaces the old forum [133].

2012

- Atlassian Marketplace is launched [79]. Making it easier for customers to purchase add-ons and easier for developers to sell and profit from their add-ons [44], [79].
- Add-ons for Jira, Confluence, Stash, Bamboo and more [134].
- New this year: consolidation of Atlassian products and add-ons on a single invoice [134].
- HipChat is acquired [129].

2013

- Atlassian OnDemand: over 24,000 instances running and this number is growing [44]
- Atlassian Connect is announced [44]. Atlassian connect is a distributed add-on technology with which developers can extent Jira, Confluence and all other Atlassian applications [44].
- Confluence questions is announced [133].
- 145 add-ons and extensions for Bamboo [135]
- 2,000 registered developers [79]
- 1,500 public add-ons (plugins) build for Jira, Confluence, Stash and other Atlassian Developer tools [79]
- Developers have now sold well over $10 million dollars in third-party add-ons for Atlassian applications [79]
- %US 58 million spend on research and development [129].

2014

- Rebranding: old Atlassian OnDemand, new Atlassian Cloud, from now (15 December 2014) and in the coming months the swapping of the old name for the new one will be done [69].
- Introduction of Jira Portfolio [136].
- Introduction of Atlassian Enterprise [136].
- New offices opened in Manila and Austin, Texas [136].
- More than 40,000 organizations as customers [136]. [not users, organizations]
- Announcement of US stock exchange debut [129].
- Corporate headquarters are moved to London in preparation for the stock market listing [129].
- Jira is a global standard and used by companies such as NASA, universities, eBay and Twitter [129].
- 30,000 customer [129].
- 750 people work for Atlassian [129].
- Revenues of $US 150 million a year [129].
- January 28, there are 178,899,959 Atlassian shares. The co-founders have 78%.

2015

- No specific highlights found.

2016

- Atlassian acquires StatusPage [83]. StatusPage was founded in 2013 [83]. The mission of StatusPage was to create trust between service providers and their customers [83].

2017

- Jira seamlessly integrates with Hipchat in the free connector from Atlassian [47]
- Trello is bought in January [130].
- Atlassian is working on moving all their operations to Amazon Web Services, a public cloud [130]. This way they can focus more on their core business [130].
- 75% of new customers are on the cloud [130].
- Customers in 170 countries [130].

2018

- Slack: more than 6 million active users [39].
- 2,500 employees [39].
- 9 offices in 6 countries [39].
- More than 125,000 customers [39].
- 2 planets with their software on them [39].
- 3,500 apps available in the marketplace [39].

Appendix 4: Salesforce Products

Products: [137].

Sales Cloud

The Sales Cloud product is intended to be used for the tracking of opportunities and can be used on any desktop or device. It delivers quotes, proposals and contracts. It also offers automated billing, subscription and revenue recognition capabilities. Sales Cloud also includes Einstein capabilities for predictive sourcing, actionable insights, intelligent automation and improved forecasting.

Service

Service Cloud is a tool for customer support. Its intention is to make customer support smarter and faster. It can be also be used in the field to deliver onsite service.

Marketing

Marketing Cloud is used to create personalized customer journeys, powered by an intelligent marketing platform for email, mobile, social, digital advertising and DMP. Pipelines can be accelerated and sales can be driven by B2B marketing automation.

Commerce

Create personalized shopping experiences with predictive intelligence on different channels and devices. It can be used for marketing, merchandising, promotions, fulfilment and more. It can create visibility for customers into real-time inventory visibility, allocation logic and order lifecycle management across channels.

Platform

The platform is the place for building and buying applications to streamline, automate or mobilize any business process. Apps can easily be bought or created on the lightning platform.

Einstein – Analytics

With Einstein analytics new insight can be collected from data with a few clicks. Historical information can be gathered and trends can be predicted.

Communities

Create online collaboration environments to connect with customers and build more personal relationships with them. Boost employee productivity and empower customers and employees with the collaboration platform.

IoT

Internet of Things devices can be connected to Salesforce data. Big data can be transformed into real-time actions.

Industries

Products can be built on Salesforce to meet specific industry needs and customer expectations.

Integration

All apps, data and devices across multiple clouds, organizations and systems can be connected with the integration builder. Personalized end-to-end customer experiences can be created.