



On-premise to SaaS Integration Risk Mitigation

A framework to gain insight in the risks, and recommend mitigation strategies, specific to on-premise to SaaS integration

Stijn Vorstenbosch

*"I promise nothing complete; because any human thing supposed to be complete,
must for that very reason infallibly be faulty."*

Moby Dick (Melville, 1851)

Master's thesis Business Information Technology
Faculty of Electrical Engineering, Mathematics, and Computer Science
&
Faculty of Management and Governance
University of Twente

A framework to gain insight in the risks, and recommend
mitigation strategies, specific to on-premise to SaaS integration

Utrecht, February 2015

Author

Name: Stijn Vorstenbosch
Email: vorstenbosch.stijn@gmail.com
Student number: s0169021

Supervisory committee

Internal supervisors
Dr. Ir. Maya Daneva
Prof. Dr. Jos van Hillegersberg

External supervisor
Rosanne Sigmans
Managing consultant
Deputy Clustermanager Fusion Technology J41

UNIVERSITEIT TWENTE.

PO Box 217
7500 AE Enschede
the Netherlands
+31 534 89 91 11
info@utwente.nl
www.utwente.nl



Reykjavikplein 1
3543 KA Utrecht
the Netherlands
+31 306 89 00 00
info@capgemini.com
www.capgemini.com

MANAGEMENT SUMMARY

Large software systems, such as Enterprise Resource Planning (ERP) systems, are considered notoriously hard to implement successfully. After years of experience in practice and scientific research towards critical success factors, organization impact, and economic impact only a small number of large implementation projects is a success. This research contributes to the definition of solution strategies to this challenge in order to increase the success rate of large implementation projects by addressing a small and new piece of the puzzle: Software-as-a-Service (SaaS) integration for ERP. The research has been specifically scoped to answer the following research question:

What are the risks specific to on-premise to SaaS integration and mitigation strategies to reduce these risks?

Integration between an on-premise ERP system and a SaaS solution results in a unique set of integration dynamics when compared to the more traditional integration scenarios such as on-premise to on-premise or on-premise to hosted. The difference is fundamentally grounded into two characteristics: ownership of the software and whether or not the integration crosses organizational boundaries. These two characteristics of 'on-premise to SaaS' integration provide insight in the specific integration risks that can be encountered.

To answer the research question we have adopted a multi-method research approach combining an explorative literature search, a rigorous systematic literature search, interviews with experts from practice and a workshop. By investigating the key drivers to ERP-as-a-service adoption and designing an integration risk factor framework, we have explicated multiple risks and formulated numerous mitigation strategies. The joint application of all research methods resulted in 36 specific on-premise to SaaS integration risks and 29 strategies to mitigate these risks. The 36 risks have been clustered into 12 overarching risk themes. These resulting themes are: complexity, compliancy, integration, support, security, release planning, change requests, system performance, maintainability, development costs, data confidentiality & integrity and user adoption. All these themes represent two different perspectives covering the same risk set. The perspectives have been compared and the themes have been matched to mitigation strategies that reduce the specific risk theme.

We have concluded that integration between an on-premise ERP system and a SaaS solution is not a new technical challenge, but rather more a governance-oriented matter. Drawing upon this conclusion we argue that it is not just a matter of orientation, but that governance becomes more important and complex due to the unique dynamics of on-premise to SaaS integration. This, in turn, leads to the hypothesized claim that when a SaaS solution is truly embedded in the application landscape of a client organization, the costs savings made by avoiding initial investments costs are reduced due to the increased integration effort. This weakens many statements claiming SaaS solutions are considered cheaper than the tradition on-premise system.

Furthermore, the themes revolving around change, maintainability, development, performance, integration support and security consist of a relatively high number of risks in comparison with the mitigation strategies that we found. Creating a dedicated role for overseeing SaaS integrations, using a canonical model, having SaaS ready on-premise architecture, setting up (WSDL) service contracts in a smart way, making use of dev-ops teams and creating back up strategies for broken interfaces are considered to be the most versatile mitigation strategies. Testing the SaaS, in particular regarding its non-functional requirements, before use is considered to be the most cost-effective mitigation strategy found.

The results of this research are the first steps towards developing a deeper understanding of the phenomenon of on-premise to SaaS integration. We argue the results help reveal the unique dynamics within an on-premise to SaaS integration projects that are created by the cloud computing

model. This understanding provides a powerful reasoning tool considering these types of integrations for both practitioners and researchers.

While we have identified numerous risks and mitigation strategies, we have not collected data on the probability and impact of the risks and mitigation strategies. This is necessary to make quantitatively substantiated claims concerning the impact of risks or the potential of mitigation strategies. We have also concluded that for every different integration environment, which includes the organization that uses the integration, the laws and standards they have to uphold, the application and infrastructure landscape including integration targets and the integration technologies used, the probability and impact of risks and the potential mitigation strategies can differ.

To build upon this research and obtain more insights into the risks of on-premise to SaaS integration we argue that undertaking case studies can break the shroud surrounding the probability and impact of risks and mitigation strategies and can provide the next step in validating our findings. Furthermore research towards different integration environments and how these affect probabilities and impacts of risks and the potential of mitigation strategies, detailed studies of the found risks and mitigation strategies, and collecting more risks and mitigation strategy to provide more generalizable themes and perspectives are considered valuable next steps. Another compelling 'next step' would be to approach our findings from the perspectives of the SaaS vendor. The findings of this research can be used as requirements to create a SaaS solution that reduces integration risks from within its design. Last but not least our findings could be used to improve existing IT governance frameworks such as ITIL in order to increase their resilience to SaaS to on-premise integrations.

ACKNOWLEDGEMENTS

This document marks the end of my time as a student at the university of Twente. In the following pages I present the results of my final project as a student, but before I do that I want to take a moment and reflect on my journey as a student.

My journey started in 2007 when I decided that studying Business and IT was the logical next step, but in retrospect you could also state I was lacking any real plans. Due to my 'decision' I tumbled down a hole that would even impress Alice herself. During my tumble I have met the most amazing people, embarked on countless adventures and even learned a thing or two about business and IT. Without these experiences I would not have been half the man I am today.

Writing this graduation thesis gave me the opportunity to use all these experiences to test my mettle in the 'real world'. During my project I learned much about working in a professional environment, IT consultancy, cloud computing, project management, and arguably most importantly: myself. Still it would be unfair to accept any credits for my work without mentioning a few people who have helped me during this project. First, I would like to thank Rosanne, my supervisor at Capgemini. Rosanne, it was a great pleasure working with you. You truly were my 'tour guide' during my stay at Capgemini. You are kind, smart and always ready to help. Your contributions to finding the right interviewees were invaluable. Furthermore I would like to thank my supervisors from the university of Twente. Maya, you are an awesome motivator and your contributions to my research design pushed my thesis to a higher level. Jos, your critical assessment of my work and creative mind meant sparring sessions with you always got me unstuck.

I also want to thank all other colleagues that helped me during my project and made my time at Capgemini unforgettable. I always went home with a positive feeling and healthy motivation to give my best the following day. Special mentions are deserved for the colleagues that were involved in exploring potential research subjects, took the time to answer my questions during the interviews and attended or helped organize my workshop.

Besides the people that helped me write my graduation thesis I would also thank my family and friends for providing me with much needed relaxation. I am lucky to say that you are too numerous to mention all by name and I hope you will be just as present in my next journey as you were in this one.

Last but not least I want to thank Alina for putting up with me all these years and thank my parents for their unlimited support and love. You kept faith and allowed me to experience life to the fullest even at times when the progress I made was not represented by passing exams. Without you my achievements would not have been possible.

Stijn Vorstenbosch

Utrecht, 2015

TABLE OF CONTENTS

Management summary	1-2
Acknowledgements	1-4
Table of contents	1-5
List of Figures	1-8
List of tables.....	1-10
1 Introduction	1-10
1.1 Capgemini.....	1-10
1.2 Motivation.....	1-11
1.3 Problem description	1-12
1.4 Research scope	1-13
1.5 Research goal.....	1-13
1.6 Research questions	1-13
2 Research approach	2-14
2.1 Explorative literature research	2-15
2.2 Systematic literature research	2-15
2.2.1 Search terms.....	2-15
2.2.2 Search tools	2-16
2.2.3 Inclusion and exclusion criteria	2-16
2.2.4 Validity	2-17
2.2.5 Search process	2-17
2.3 Semi-structured interviews.....	2-17
2.3.1 Interview plan.....	2-18
2.3.2 Data Collection & analysis	2-19
2.3.3 Validity	2-21
2.4 Workshop.....	2-21
2.4.1 Workshop plan	2-21
2.4.2 Data collection and analysis	2-23
2.4.3 Validation	2-23

3	Drivers AND Barriers of ERP-as-a-serice adoption	3-24
3.1	Cloud computing	3-24
3.2	Enterprise Recourse Planning (ERP)	3-26
3.3	ERP-as-a-service.....	3-27
3.4	Drivers and barriers of ERP-as-a-service adoption	3-27
4	Factors influencing integration risks	4-30
4.1	Integration scenarios.....	4-30
4.1.1	ERP delivery models.....	4-30
4.1.2	ERP integration.....	4-32
4.2	Integration RISK level	4-36
4.2.1	Integration	4-36
4.2.2	Technical integration risk levels	4-37
4.2.3	Network connectivity	4-37
4.2.4	Data sharing.....	4-37
4.2.5	Application interoperability	4-37
4.2.6	Process coordination	4-38
4.2.7	integration governance	4-38
4.3	Application type.....	4-38
4.4	Conclusions from explorative research.....	4-39
4.5	Feedback on the integration factors framework.....	4-41
4.6	Changes made to the integration factor model.....	4-42
5	Risks and mititagation strategies	5-44
5.1	Findings from the scientific community	5-44
5.1.1	Risk found	5-45
5.1.2	Mitigation strategies found	5-46
5.2	Findings from experience experts.....	5-47
5.2.1	Risks found	5-47
5.2.2	Mitigation strategies found	5-50
5.3	Combination of views.....	5-52
6	Integration framework	6-54

6.1	VennMaster.....	6-54
6.2	User IT organisation perspective	6-55
6.3	Integration cost perspective	6-56
6.4	Themes and mitigation strategies	6-57
6.4.1	Complexity	6-57
6.4.2	Integration	6-58
6.4.3	Compliance	6-59
6.4.4	Security	6-60
6.4.5	Release planning	6-61
6.4.6	Change requests.....	6-62
6.4.7	Support	6-63
6.4.8	System performance.....	6-64
6.4.9	Data Confidentiality & integrity.....	6-65
6.4.10	Maintainability	6-66
6.4.11	Development costs	6-67
6.4.12	User adoption.....	6-68
6.5	mitigation strategies	6-69
6.6	Integration factor framework risk mapping.....	6-71
6.7	Overview and validation.....	6-72
7	Conclusion	7-74
7.1	Discussion, implications and limitations.....	7-76
7.2	Recommendations	7-78
8	Future research	8-80
9	References	9-81
Appendix A.	Overview of Research Approach	86
Appendix B.	Mitigation strategies and risks.....	87
Appendix C.	Workshop	91



LIST OF FIGURES

Figure 1: Overview Capgemini Group (“Capgemini,” 2014)	1-11
Figure 2: Visual representation of overview of research approach.....	2-14
Figure 3: Visual representation of the systematic literature research process	2-15
Figure 4: Search terms	2-17
Figure 5: Visual representation of the interview process	2-20
Figure 6: Screenshot of coding done in QDA miner lite.....	2-20
Figure 7: Visual representation of the workshop process.....	2-23
Figure 8: Cloud computing service model categories (Pearson, 2012).....	3-25
Figure 9: Cloud deployment options	3-26
Figure 10: Fictional company without integration	4-33
Figure 11: Fictional company with integrated information systems	4-34
Figure 12: Fictional company with a hosted ERP system.....	4-34
Figure 13: Fictional company with an ERP-as-a-service module	4-35
Figure 14: Multi-tenancy of the ERP-as-a-service component	4-35
Figure 15: Five different integration levels.....	4-37
Figure 16: Integration factors framework	4-40
Figure 17: Integration factors framework 2.0	4-43
Figure 18: Systematic literature research topic matrix.....	5-45
Figure 19: Visual representation of the risks clustering of the user IT organization perspective.....	6-56
Figure 20: Visual representation of the risks clustering of the integration cost perspective	6-57
Figure 21: Perspective comparison from the complexity viewpoint.....	6-58
Figure 22: Perspective comparison from the integration viewpoint	6-59
Figure 23: Perspective comparison from the compliancy viewpoint.....	6-60
Figure 24: Perspective comparison from the security viewpoint.....	6-61
Figure 25: Perspective comparison from the release planning viewpoint.....	6-62
Figure 26: Perspective comparison from the change requests viewpoint	6-63
Figure 27: Perspective comparison from the support viewpoint	6-64
Figure 28: Perspective comparison from the system performance viewpoint	6-65
Figure 29: Perspective comparison from the data confidentiality & integrity viewpoint	6-66

Figure 30: Perspective comparison from the maintainability viewpoint 6-67

Figure 31: Perspective comparison from the development costs viewpoint 6-68

Figure 32: Perspective comparison from the user adoption viewpoint 6-69

Figure 33: Risk factor framework with categorization of risks 6-71

Figure 34: Overview of the comparison of risks perspectives 6-72

Figure 35: GEIT framework from COBIT 5 (ISACA, 2012) 7-79

LIST OF TABLES

Table 1: Inclusion criteria	2-16
Table 2: Exclusion criteria	2-16
Table 3: Interview sample	2-18
Table 4: Interview timetable	2-19
Table 5: Interviews facts	2-19
Table 6: Transcription and coding facts	2-20
Table 7: Workshop sample	2-22
Table 8: Workshop timetable	2-22
Table 9: User opportunities and challenges (Juell-skielse & Enquist, 2012)	3-28
Table 10: Supplier opportunities and challenges (Juell-skielse & Enquist, 2012)	3-29
Table 11: On-premise ERP advantages (Duan et al., 2012)	4-31
Table 12: Hosted ERP advantages (Duan et al., 2012)	4-31
Table 13: ERP-as-a-service advantages (Duan et al., 2012)	4-32
Table 14: Application types	4-39
Table 15: Overview of resulting papers	5-44
Table 16: Integration risks found in literature	5-45
Table 17: Integration risk mitigation strategies from literature	5-46
Table 18: Risks found during the interviews	5-47
Table 19: Frequency of risks found during interviews	5-49
Table 20: Mitigations strategies found during the interviews	5-50
Table 21: Frequency of mitigation strategies found during interviews	5-51
Table 22: Risk themes from the user organization perspective	6-55
Table 23: Risk themes from integration cost perspective	6-56
Table 24: Perspective comparison from the complexity viewpoint	6-58
Table 25: Perspective comparison from the integration viewpoint	6-58
Table 26: Perspective comparison from the compliancy viewpoint	6-59
Table 27: Perspective comparison from the security viewpoint	6-60
Table 28: Perspective comparison from the release planning viewpoint	6-61
Table 29: Perspective comparison from the change requests viewpoint	6-62

Table 30: Perspective comparison from the support viewpoint 6-63

Table 31: Comparison Perspective comparison from the system performance viewpoint 6-64

Table 32: Perspective comparison from the data confidentiality & integrity viewpoint 6-65

Table 33: Perspective comparison from the maintainability viewpoint 6-66

Table 34: Perspective comparison from the development costs viewpoint 6-67

Table 35: Perspective comparison from the user adoption viewpoint 6-68

Table 36: Analysis of mitigation strategies and risks themes 6-69

Table 37: Number of mitigation strategies applications according to experts..... 6-70

Table 38: Integration factor framework risks categorization 6-71

Table 39: Categorization of risks 6-72

Table 40: Key drivers and barriers for ERP-as-a-service adoption form the user perspective 7-74

Table 41: Overview of risks themes..... 7-75

1 INTRODUCTION

Complex software systems such as an Enterprise Resource Planning (ERP) system are a firm's facilitators for competitive advantages. Successfully implementing a fully integrated ERP system is like having a solid foundation for distinguishing yourself from the competition. However large IT systems are rather unsung heroes these days. The news is filled with failed implementations and ICT consultancy firms are getting increasing amounts of bad publicity, especially when a project in the public sector is abandoned. A couple of months before the start of this project, "de Volkskrant" states that the Dutch government alone waste 4 to 5 billion euro on failing IT projects every year (ANP, 2014), which is estimated to cost the Dutch taxpayer an amount of €300 per capita per year.

In the third week of my research, the SVB (Sociale VerzekeringsBank) "kicked out" Capgemini according to the Dutch newspaper "de Telegraaf", after an unsuccessful Oracle implementation seemingly costing the Dutch tax payer roughly €50 million (van Bergen & Mos, 2014). Even considering the fact that "De Telegraaf" is not known for their tact, and responsibility for failure of such a grand project can hardly be placed at only one stakeholder, it still does not look good on Capgemini's rap sheet. This research sets out to reduce the failure rate of ERP software implementation projects by looking into a small and new part of the puzzle: cloud integration.

This chapter introduces the research by providing a short description of the company facilitating the project in Chapter 1.1, followed by the motivation behind the research in Chapter 1.2. Chapter 1.3 introduces the industry-relevant problem that this research deals with. Chapter 1.4 states the boundaries of this research. Chapter 1.5 presents the research goal. Finally the research questions are presented in Chapter 1.6.

1.1 CAPGEMINI

This chapter describes the company facilitating the present research. We will start by introducing the Capgemini Group and move towards the division that was the host of this research project.

The history of Capgemini is characterized by takeovers, separations and mergers, all of which makes it rather complex, therefore in this thesis we have chosen for the 'simple' version. To provide insight into the history of the Group we start our historic journey in 1967. The predecessor of Capgemini was founded in 1967 (SOGETTI, Grenoble) by Serge Kampf. In 1975, after acquiring both CAP and Gemini computer systems, the name changed to Cap Gemini Sogetti. Finally in 2004 the name of the firm was officially changed to what it is today: Capgemini.

"Capgemini is one of the world's foremost providers of consulting, technology, outsourcing services and local professional services with reported revenues above 10 billion euro's. Present in over 40 countries with almost 140,000 employees, the Capgemini Group helps its clients transform in order to improve their performance and competitive positioning" ("Capgemini," 2014).

Figure 1 shows an overview of the Capgemini Group. Our research took place within the Oracle division, which is part of the Application Services NL.

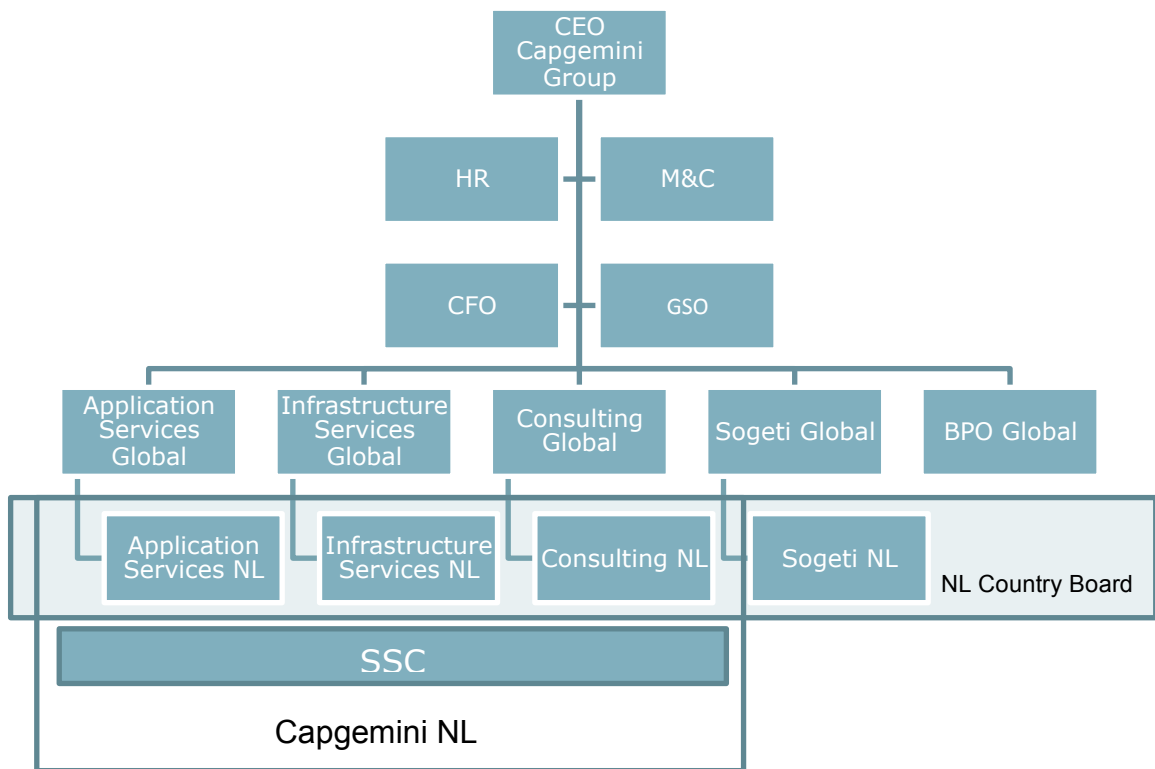


Figure 1: Overview Capgemini Group (“Capgemini,” 2014)

The part of the Oracle division the main researcher was an intern of was named J41 Oracle Solutions and is occupied with both providing Oracle technology solutions and Oracle application solutions. In total these two clusters account for about 160 consultants backed up by a number of staff people.

At the home base of J41 at Reykjavikplein 1 in Utrecht there were an average of 15 colleagues present at any given time during my stay. The less the number of people around, the better for the business, as those colleagues who were not at the office, were most likely working at the clients’ sites. The consultants on the so-called ‘bench’ were utilizing their networks in order to find new assignments, or improving and broadening their skills set by taking courses from one of the Capgemini learning environments.

1.2 MOTIVATION

ERP systems are among the largest, most complex, and most demanding information systems implemented by firms (Grabski, Leech, & Schmidt, 2011). ERP systems are considered companywide information systems that integrate all aspects of business. This means that an ERP system can cover everything from human resources, manufacturing, accounting, sales, distribution, management support and anything in between. The advantages of such an integrated system for firms are many fold, i.e. quicker reactions to competitive pressures and marketing opportunities, reduced inventories, ability to comply with regulations, and achieving a more efficient workforce (Bingi, Sharma, & Godla, 1999). The first ERP systems were introduced in the 80s and from that point ERP has been a focal point for both the scientific world and the workplace. Although the advantages these systems are renown, the process of actually achieving them could be called notorious. The scientific community has undertaken numerous studies into ERP systems that can be roughly placed into three categories: critical success factors, organizational impact, and economic impact (Anwar, 2011; Grabski et al., 2011). Due to the attention received, the research towards ERP quickly matured, but the results in the

field could be called under par (Aloini, Dulmin, & Mininno, 2007). When one looks into an everyday newspaper there is bound to be an article about some ERP project that went over budget, took longer than expected, or was abandoned entirely. From this perspective, the basis of my motivations stems to improve the success rate of ERP projects.

1.3 PROBLEM DESCRIPTION

Improving the success rate of large IT projects such as ERP projects is an ambitious goal to strive for, but taking on 35 years of research and experience in practice during a 6 month project could be called lunacy. However the field of ERP is changing due to new technologies and new business models. These changes present new opportunities for further research into the field of ERP, or large implementation projects in general, that for now remain under-researched. One of the changes that is particularly visible within Capgemini is cloud computing. Where once ERP systems were heavy on-premise systems that required high initial investments in both hardware and software, ERP systems are gradually becoming lighter and are offered as SaaS (Software As A Service) solutions. These lightweight ERP implementations, operating from the cloud are called ERP-as-a-service. This change opens up a variety of possibilities for SMEs (Small Medium Enterprises) and for LEs (Large Enterprises) alike. Research on this topic is limited and researchers have called for research towards the implementation, use, and risks in this new environment (Grabski et al., 2011).

When firms buy an ERP-as-a-service module, i.e. Oracle CRM on demand, many challenges are to be overcome in order to achieve the full advantages of the SaaS solution. Studies have been done towards these challenges of ERP-as-a-service (Juell-skielse & Enquist, 2012; Lechesa, Seymour, & Schuler, 2012; Lewandowski, Salako, & Garcia-Perez, 2013), but in-depth research towards specific challenges found by them is close to non-existent.

Integration challenges in particular are said to be one of the major barriers of ERP-as-a-service adoption (Addo-Tenkorank & Helo, 2011; Araujo, Vázquez, & Cota, 2014; Benlian & Hess, 2011; Dubey & Wagle, 2007; Juell-skielse & Enquist, 2012; Kolluru & Mantja, 2013; Lechesa et al., 2012; Lewandowski et al., 2013; F. Liu, Guo, Zhao, & Chou, 2010; Schubert & Adisa, 2011; Themistocleous, Irani, O'Keefe, & Paul, 2001; Vassiliadis, Stefani, Tsaknakis, & Tsakalidis, 2006).

While integration on a technical and architectural level is an important key to the puzzle, integration is also an important factor on management level as it is said to be responsible for 30%-45% of the costs of an ERP-as-a-service implementation (Bernstein & Haas, 2008; Hai & Sakoda, 2009).

From an explorative study the following conclusions can be made about the research field:

- There is no clear overview of cloud integration risks;
- It is unclear how to mitigate cloud integration risks;
- Some technical solutions are proposed for specific integration problems (i.e. Liu et al., 2010; Liu, Wang, Chou, Fazal, & Li, 2006);
- Studies towards SaaS integration “best practices” exist, but research has not gone further than high level guidelines (Hai & Sakoda, 2009; Kolluru & Mantja, 2013);

Resulting from these conclusions we define the following problem statement:

It is unclear what cloud integration risks exist during a cloud solution implementation, and how to mitigate these risks.

1.4 RESEARCH SCOPE

In order to arrive at a manageable research within the time span given for this project the following scoping decisions have been made:

- The research is positioned within the Oracle division at Capgemini in the Netherlands. The experience gathered from practice will originate from the Capgemini group and mainly from the Oracle community at Capgemini.
- The research will focus solely on integrating SaaS with on-premise systems. The reason for this decision is based on two arguments. Compared to cloud-to-cloud integration this type has the potential to be more challenging. Secondly, it is also more common in practice as many clients from IT consultancy firms are still using on-premise ERP systems and buy cloud solutions as add-ons.

1.5 RESEARCH GOAL

In Chapter 1.3 we have described the challenges in the current field of ERP research, specifically towards SaaS integration challenges. In order to provide more clarity in this emerging field we have defined the following research goal:

Develop a framework in order to gain insight in the risks, and recommend mitigation strategies, specific to integrating on-premise to SaaS systems.

1.6 RESEARCH QUESTIONS

In Chapter 1.5 we have stated a research goal that stems from the research problem discussed in Chapter 1.3. The following research question is a result of that goal.

What are the risks specific to on-premise to SaaS integration and mitigation strategies to reduce these risks?

In order to divide this main research question into manageable pieces we defined five sub-questions.

- I. What are the drivers and barriers of ERP-as-a-service adoption in Small and Medium-sized enterprises (SMEs) and Large Enterprises (LEs)?
- II. What factors influence information system integration risks?
- III. What on-premise to SaaS integration risks and mitigation strategies have been identified by the scientific community?
- IV. What on-premise to SaaS integration risks and mitigation strategies have been identified by experience experts working in the field of ERP and SaaS?
- V. Are the proposed risks and mitigation strategies useful in order to improve the ERP-as-a-service implementation success rate in practice?

In the following chapter we present our research approach. Our answers to the research questions can be found from Chapter 3 and onwards, starting with the answer to research question I. Research question II is answered in Chapter 4, followed by answers to research questions III and IV in Chapter 5. In Chapter 6 we complete the answers of research questions III and IV. We also provide our answer of research question V. In chapter 7 we present our overall conclusions, followed by a discussion of our work and its implications for practitioners and researchers. We mention potential projects for future research in Chapter 8.

2 RESEARCH APPROACH

To answer the questions presented in Chapter 1.6 we use a number of different research techniques. In this chapter we describe how we approached the research and what methods we have used in order to collect data required to answer the research questions. We will also discuss how we analyzed the data and argue on the validity of the different research methods.

During the design of our research we have specifically looked into combining multiple research methods to triangulate our data sources. According to Creswell (2003), this strategy probably originates from 1959 when the first multi-method research took place. Because all research methods have their limitations, scientists used multiple methods to counteract biases that are present in any single method. For this reason we have decided to use an explorative literature search, a systematic literature search, interviews and a workshop to gather our data in the most unbiased fashion. A visual representation of the complete research design is presented in the figure below, the full-scale version can be found in Appendix A.

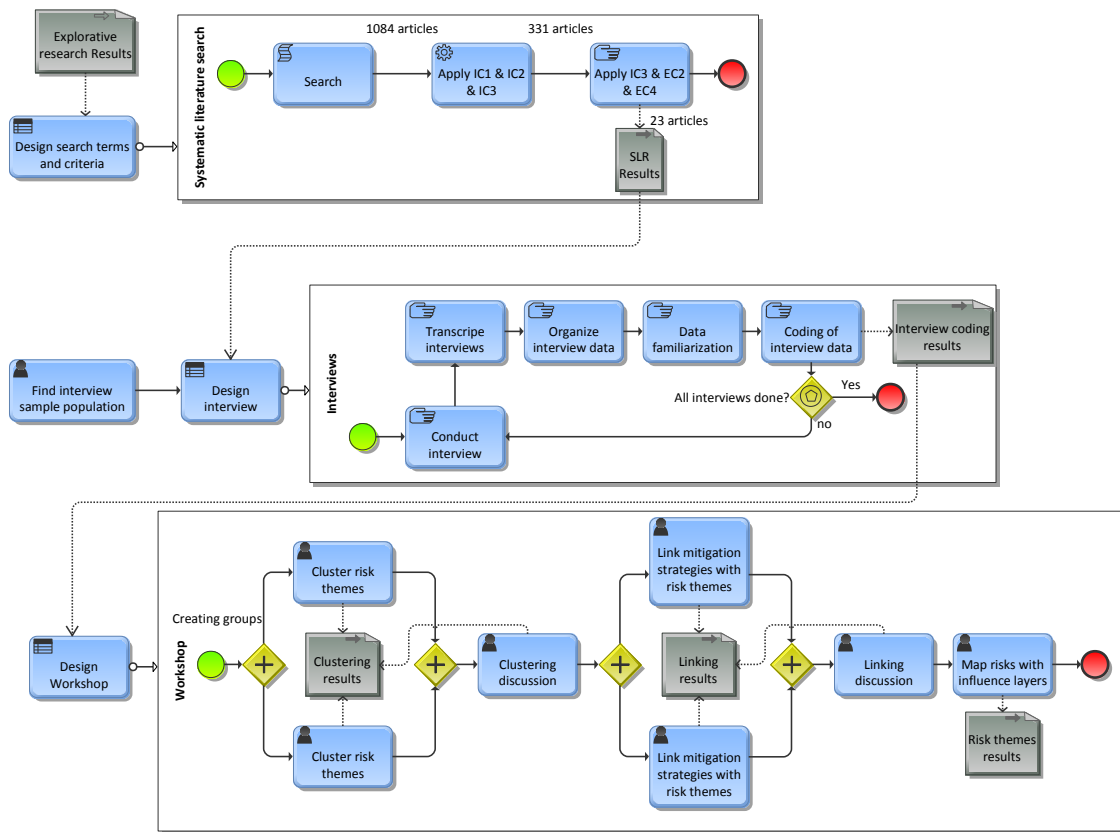


Figure 2: Visual representation of overview of research approach

In Chapter 2.1 we elaborate on the explorative literature research into the field of ERP and SaaS integrations that is used to answer research questions I and II. Chapter 2.2 explains the approach on the systematic literature research used to answer research question III, followed by the plan used to conduct interviews in order to provide the beginning of the answer of to research question IV. To complete the answer on research question IV we conducted a workshop. The workshop also provides an answer to research question V and its approach can be found in Chapter 2.4.

2.1 EXPLORATIVE LITERATURE RESEARCH

The explorative literature research has been conducted using Scopus, Google Scholar, and papers shared by supervisors and other researchers. We used the research to explore the scientific field of ERP, ERP-as-a-service, cloud computing, Enterprise Application Integrations (EAI) and software integrations general. Besides exploring the fields mentioned above we have also used the knowledge gained as a starting point for our in-depth research. The two research questions we answered during this phase can be considered a preliminary part of the research, the so-called foundation from which we dived deeper into the field of on-premise to SaaS integration.

2.2 SYSTEMATIC LITERATURE RESEARCH

In order to provide a complete unbiased overview of the on-premise to SaaS integration risks and mitigation strategies known in the scientific community, we conducted a systematic literature research. We have explicitly chosen to use a systematic literature research to capture all that is known on the subject.

“A systematic literature research is a means of identifying, evaluating and interpreting all available research relevant to a particular research question, or topic area, or phenomenon of interest” (Kitchenham, 2004). By using a structured protocol, as identified by Barbara Kitchenham, we can provide a literature search that is thoroughly and unbiased. It provides insight in the decisions made by the researcher, so that other researchers can approach the same challenge differently or reproduce the results in order to validate.

This chapter will be divided into different parts, representing the literature review protocol. The protocol consists of search terms, search tools used, and inclusion and exclusion criteria.

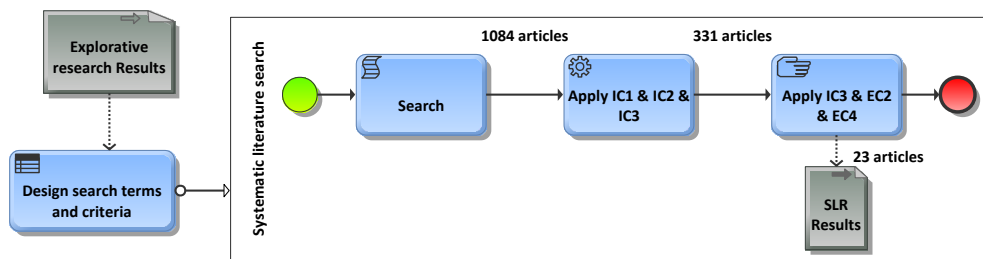


Figure 3: Visual representation of the systematic literature research process

2.2.1 SEARCH TERMS

In order to find useful results the following search term has been used: *cloud OR saas OR “cloud ERP” OR “ERP-as-a-service” OR “ERP as a service”) AND integration AND (risk* OR barrier* OR disadvantage* OR challenge* OR problem*)*. We have explicitly decided not to search for on-premise to SaaS integration as the combination SaaS and on-premise has not been widely used and considering the maturity of this research niche chances are we miss interesting studies. Another explicit decision that was made regarding the search terms was to withhold the sought after mitigation strategies from the actual search query as we argue that the field is in such a immature state this would lead a very small results pool. We will have a greater chance extracting the mitigation strategies from the studies in which risks have been identified.

2.2.2 SEARCH TOOLS

To find the best sources for the literature review we will use the Scopus search engine. Scopus distinguishes itself from the likes of Google scholar by providing strong tools for a thorough literature search. Although Google Scholar covers a wider variety of publications, it lacks important sources and has a high amount of search “noise” (Mikki, 2009). However, we are limited to the university of Twente library and Google Scholar is known to provide a greater possibility to find full texts (Mikki, 2009). We therefore use scholar to help us find full text of studies that we cannot access through Scopus.

2.2.3 INCLUSION AND EXCLUSION CRITERIA

In order to acquire a very specific and high quality subset of all papers resulting from the search query, inclusion and exclusion criteria have been deployed. The inclusion criteria govern the quality of a paper and whether or not it is relevant.

Table 1: Inclusion criteria

#	Inclusion criteria	Reasoning
IC1	Results must be a reviewed scientific article or a conference paper	Quality constraint for the input (handled by Scopus)
IC2	The subject area of the research is “computer science, social sciences and business & management”	Filtering out unwanted subject areas that cannot be applied within this research (handled by Scopus)
IC3	Results must have cloud-to-on-premise integration risks or mitigation strategy as a topic	Filtering out studies that do not discuss cloud-to-on-premise integration (handled manually)

The exclusion criteria provide boundaries, limiting the result pool on time of publishing, presence in the Twente University library and generally making sure no unnecessary work is done. An overview of the criteria can be found in Table 1 and Table 2.

Table 2: Exclusion criteria

	Exclusion criteria	Reasoning
EC1	Results must be from 2009 or more recent	The field of computer science and especially cloud computing is rapidly evolving. This constraint filters out old research that is no longer applicable (handled by Scopus)
EC2	The result is not available through the University of Twente library or via Google Scholar	This research will be conducted as part of a graduation thesis, therefore no funding is present to acquire papers that are not obtainable via the given sources (handled manually)
EC3	The older or longer case of multiple results on the same topic by the same author	To prevent doubling the work effort while the gains are minimal (handled manually)
EC4	The paper is not available in English	We are only able to interpret English studies (handled manually)

In Figure 4 the complete search query used in Scopus can be found. This is a combination of the search terms and automated criteria. The manually criteria are handled by the researcher.

```

TITLE-ABS-KEY((cloud OR saas OR "cloud ERP" OR "ERP-as-a-service" OR "ERP as a service") AND integration
AND (risk* OR barrier* OR disadvantage* OR challenge* OR problem*)) AND PUBYEAR > 2008 AND (
EXCLUDE(SUBJAREA,"ENGI" ) OR EXCLUDE(SUBJAREA,"MATH" ) OR EXCLUDE(SUBJAREA,"EART" ) OR
EXCLUDE(SUBJAREA,"PHYS" ) OR EXCLUDE(SUBJAREA,"MEDI" ) OR EXCLUDE(SUBJAREA,"ENVI" ) OR
EXCLUDE(SUBJAREA,"DECI" ) OR EXCLUDE(SUBJAREA,"ENGI" ) OR EXCLUDE(SUBJAREA,"MATH" ) OR
EXCLUDE(SUBJAREA,"EART" ) OR EXCLUDE(SUBJAREA,"PHYS" ) OR EXCLUDE(SUBJAREA,"MEDI" ) OR
EXCLUDE(SUBJAREA,"ENVI" ) OR EXCLUDE(SUBJAREA,"DECI" ) OR EXCLUDE(SUBJAREA,"MATE" ) OR
EXCLUDE(SUBJAREA,"ENER" ) OR EXCLUDE(SUBJAREA,"HEAL" ) OR EXCLUDE(SUBJAREA,"CHEM" ) OR
EXCLUDE(SUBJAREA,"BIOC" ) OR EXCLUDE(SUBJAREA,"CENG" ) OR EXCLUDE(SUBJAREA,"ECON" ) OR
EXCLUDE(SUBJAREA,"AGRI" ) OR EXCLUDE(SUBJAREA,"IMMU" ) OR EXCLUDE(SUBJAREA,"MULT" ) OR
EXCLUDE(SUBJAREA,"ARTS" ) OR EXCLUDE(SUBJAREA,"NEUR" ) OR EXCLUDE(SUBJAREA,"PHAR" ) OR
EXCLUDE(SUBJAREA,"Undefined" ) ) AND ( EXCLUDE(DOCTYPE,"cr" ) OR EXCLUDE(DOCTYPE,"re" ) ) AND (
EXCLUDE(DOCTYPE,"ch" ) )

```

Figure 4: Search terms

2.2.4 VALIDITY

In order to provide an unbiased and validated literature research we take validity measures into account. The first remark we want to make is that neither the main researcher nor his supervisors have conducted any significant research into the cloud integration field. This provides a clean sheet and unbiased view on the research field. Secondly, the inclusion and exclusion criteria that are not automatically supported by the used search tool will be checked by the main researcher. An internal validity check will be provided by a supervisor by applying the same manual criteria as the main researcher did on a subset of the raw results.

2.2.5 SEARCH PROCESS

When the search query was entered raw into Scopus on 7-10-2014, before any automated inclusion or exclusion criteria were taken into account, 1084 results were returned. After including the automated criteria as described in Chapter 2.2.3, 753 papers were excluded. The remaining 331 papers passed through our manual criteria based on abstracts, resulting in another 308 papers being excluded. The remaining 23 papers were analyzed in-depth and results are presented in Chapter 4. During the search process we have applied a snowballing technique to find more relevant papers by looking at the references of the resulting papers.

2.3 SEMI-STRUCTURED INTERVIEWS

Due to the perceived immaturity of the research towards cloud integration risks we have decided to use experts from practice to provide us with participatory knowledge claims. In this case, the experts will be senior implementation consultants (ranging from technical to more business-like roles) working at Capgemini. According to Creswell (2003), using open-ended questions fits with the quest for participatory knowledge claims, due to the explorative nature of the research. This fit is supported by Esterberg (2002) as she argues that semi-structured interviews allow the researcher to gather data about topics or phenomena we did not know interested us. Although we have some idea of what the risks of cloud integration will be, the field is in such an immature state that it is not wise to structure the interviews too rigidly and miss new perspectives. Results of the semi-structured interviews can be found in Chapter 5.2.

In the chapters below we present our detailed interview plan; explaining the sample and the interview script.

2.3.1 INTERVIEW PLAN

The first step towards data gathering through semi-structured interviews is to find a sample of people that can give the data that is needed for the research. The population we were interested in are senior implementation consultants that are experts on the topics of integration in general and SaaS solutions or a leading expert on one of the two topics. To arrive at a sample frame we have used expert sampling. This is a technique where respondents are chosen in a non-random manner based on their expertise that is necessary for the specific research subject (Bhattacharjee, 2012). The sampling frame was selected with the aid of the internal supervisor at Capgemini. Table 3 summarizes the data about our sample.

Table 3: Interview sample

Interviewee #	Project scenario	Project role
I1	Using Salesforce to try out a new project	Solution architect
I2	Salesforce integration with multiple on-premise systems	Data architect
I3	Migrating from an on-premise application to a SaaS component	Lead architect
I4	Migrating and standardizing multiple applications into a SaaS component	Lead governance
I5	Integrating legacy systems through a bus with a SaaS	Integrator
I6	Integrating multiple application using the Oracle Fusion middleware stack	Solution architect
I7	Integrating a Salary SaaS component with PeopleSoft ERP	Solution architect
I8	Integrating a HR SaaS component with on-premise ERP	Integration specialist on the application level
I9	Salesforce integration with on-premise ERP	Solution architect / Lead developer

When deciding how to design the interview-based research, we used our systematic literature research as a guide for creating the interview script. We knew beforehand that the scientific niche on integration risks and mitigation strategies of this particular type of integration was not a very mature field but after the literature research we made the following conclusions: First, we argue that the results of the literature study can be called only partially complete and therefore we need to find more risks instead of just trying to get them verified in the interviews. Secondly, we noticed that the scope of the sought after risks and mitigation strategies had to be made very clear in order to prevent discussing more general integration risks during the interviews. Table 4 shows the timetable for the interviews.

Creating the interview script meant making decisions. First, we have decided to use the integration factor framework that was developed as an answer to research question II as a tool to keep the interviewees within the scope of this research. Secondly we decided to let the interviewee come up with a project on their own that includes an integration between an on-premise system and a SaaS application. This is done for three reasons; first, due to the lack of hands-on experience with integration projects by the researcher, it could cripple the interview by coming up with a project that does not reflect reality. Secondly, in order to get as much data as possible we want the interviewee to be comfortable with the project we are discussing in terms of risks and mitigation strategies. Third, by allowing the interviewee not to speak strictly from past experiences, we also avoid any non-disclosure agreements that could hamper the flow of data. After agreeing upon a project we decided to create a

relatively unstructured part in which the interviewee was encourage to come up with as much risks and mitigation strategies as possible. The researchers aim was to keep the information flow going, and diving into mentioned risks or strategies if the interviewee remained too abstract.

Table 4: Interview timetable

Subject	Time
Introduction	2-3 minutes
Explanation of the integration framework	5 – 10 minutes
Finding a suitable project and role	5 - 10 minutes
Finding risks and mitigation strategies	20-30 minutes
Concluding remarks	2-3 minutes
Approximate total time	45 minutes

2.3.2 DATA COLLECTION & ANALYSIS

In order to guide us through the large amount of work and avoid potential pitfalls we have used available guides on qualitative analysis (Dierckx de Casterlé, Gastmans, Bryon, & Denier, 2011; Lacey & Luff, 2009). According to these works, collecting and coding qualitative data revolves around 5 major steps; transcription, organizing data, familiarization, coding, and themes. This chapter covers these steps in order to give full insight in the data collection and analysis process.

The recordings made during the interviews were used for transcription. We have decided to start with transcribing from the start of the ‘interactive’ part of the interview, referred to in the timetable as ‘finding a suitable project and role’ and stop transcribing at the concluding remarks in order to safe valuable time. However Dierckx de Casterlé et al., (2012) argue that this can create biases in the collected data, therefore we have strayed from this norm if we encountered feedback or other interactions from the interviewee during the ‘non-interactive’ parts of the interviews. We have transcribed all words of the interviews but not the non-verbal cues. We acknowledge that these cues can provide valuable insights but we have ignored them due to time constraints. The audio files of the interviews have been transcribed on the same day or on the day after the interviews to avoid the possibilities for misunderstandings in case of background noises or heated discussion. Although almost all interviews were conducted in private rooms, this tactic proved useful as it was rather remarkable to notice how conversations are sometimes incomplete or unclear when listening to recordings of these conversations afterwards. The organization of data was mainly done by the coding software used and digital storage facilities. We argue that more rigorous methods of organization, i.e. making the transcribed interviews anonymous, were not necessary due to the small scale project and the lack of negative consequences for the interviewees. As all of the work was done by a single researcher data familiarization has occurred rather easily by listening to the audio files once before transcribing and during the transcribing process. Facts on the interviews and transcribing can be found in Table 5.

Table 5: Interviews facts

Time period used for interviewing	From 27-10-14 until 20-11-14
Number of interviews	9
Total interview time (in minutes)	474
Amount of words transcribed (including headers)	54092
Average interview time (in minutes)	53
Average amount of words transcribed per interview	6010

In order to retrieve valuable information from the data, we have extensively coded the transcriptions. We have used an open coding strategy to code all risks and mitigation strategies that we collected. A logical next step would have been to look for categories of the codes and work towards higher level themes but due to the research set-up of having interviews combined with a workshop we have excluded this step. The clustering of risks will be done by the experts, in group work, in order to avoid any biases.

Table 6: Transcription and coding facts

Amount of words transcribed (including headers)	54092
Amount of different codes	127
Amount of coded text fragments	291
Amount of words coded	18980
Percentage of words coded	34,80%

In order to extract all the data from the interview transcription, we have coded the files. In Table 6 we have presented some interesting coding facts. The coding has been done using a freeware coding application called QDA Miner Lite (www.provalisresearch.com). An overview of the entire interview process is depicted in Figure 5.

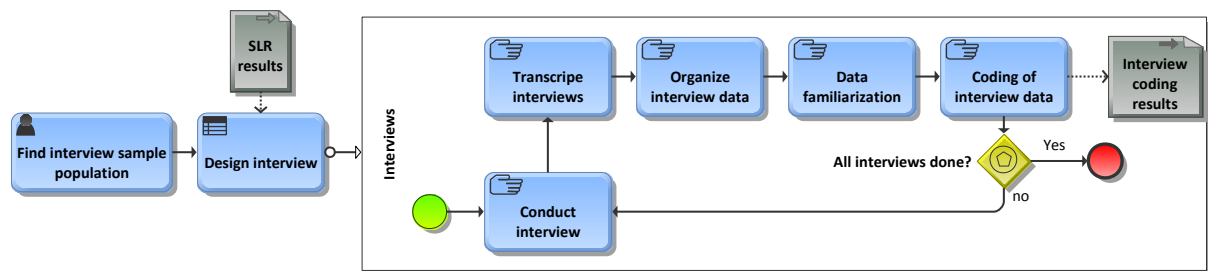


Figure 5: Visual representation of the interview process

A screenshot of how the program looks like can be seen in Figure 6. The application contains different sections, including a document manager, a code manager and a part in which the actual coding is done. The same software also allows for analyzing the codes created within the application in order to draw conclusions. I.e. all found risks can be presented in a table or diagram allowing the researcher to quickly find the segments of text that represent the found risks, the found risk can be checked on frequency of occurrence either in one transcription or in the set of all transcriptions. The coding is done by highlighting parts of the transcription and giving it a pre-defined code or creating an entirely new code for it. These codes can be grouped in order to provide a quick overview and can be used to manage all the highlighted pieces of the transcription.

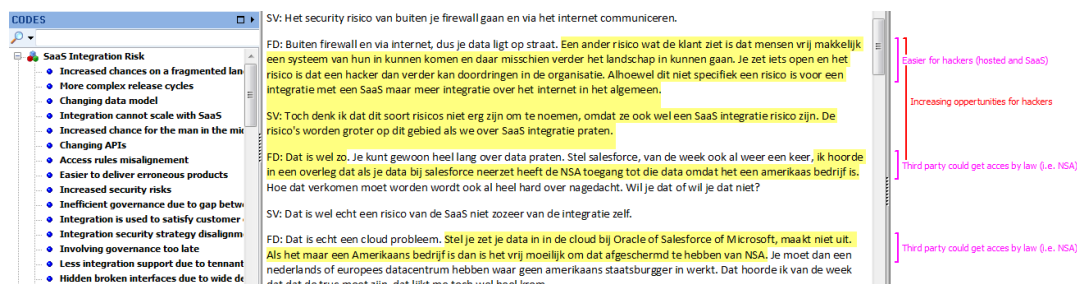


Figure 6: Screenshot of coding done in QDA miner lite

2.3.3 VALIDITY

To successfully argue that our findings from the interviews reflect reality, a number of validity measures have been taken into account. First, we have followed methodological literature on qualitative research (Dierckx de Casterlé et al., 2011; Lacey & Luff, 2009) in order to provide for a solid research approach and design for analysis. Secondly, we have used the different interviewees to check for deviant cases in order to test our interpretations. This process can be seen as part of respondent validation which is widely acknowledged as an improvement to the rigorousness of research (Lacey & Luff, 2009). Thirdly, we have guarded heavily for allowing too much interpretation during the coding process and tried to approach the data in an 'as-is' fashion as much as possible. Fourthly, we argue that the group of interviewees, while working at the same company, can be seen as diverse. Capgemini is a large company and just a small portion of the interviewees knew each other by name or had worked together in the past. At the time of the interviews all interviewees were operating on different projects. This was conducive to ensure triangulation in order to allow a more complete perspective on the research subject (Lacey & Luff, 2009).

Besides mentioning the validity measures we took in order to obtain valid results, we argue that it is fair to state the inherent weaknesses of our qualitative research approach. The first and foremost weakness is the fact that the qualitative data analysis of the interviews has been undertaken by only one researcher. Without having multiple researchers analyzing the data individually it is impossible to state if the findings are consistent, this often is referred to as inter-rater reliability (Lacey & Luff, 2009). The second inherent weakness lies in the respondent validation earlier stated to have a positive effect on the validity of the analysis of the interviews. Checking for deviations increases the validity but also the risk of influencing the interviewee and thus steering the interview in a certain direction.

2.4 WORKSHOP

We decided to organize a workshop to further enhance our findings from the interviews and to avoid potential biases in the process. This enhancement comes from identifying overarching risk themes (Lacey & Luff, 2009). This identification process can be done by the researcher but in order to get a broader support for the identified themes and by doing so avoid potential biases we let the workshop attendees do the clustering for us. In this chapter we elaborate on our workshop plan and the collection and analysis of data.

2.4.1 WORKSHOP PLAN

The same experts we have selected for our interviews have been invited to attend the workshop. The reason for this is twofold. First, this set of people is considered leading experts within Capgemini on the subject. For this reason we assume that they provide us input of the highest quality. It also implies that these experts are difficult to replace by another group of Capgemini professionals that possess the same level of knowledge. The second reason is more practical; we had limited time for our workshop and therefore needed people that were all ready 'up to speed' on the research. In the table below, we have listed the attendees for the workshop. The four attendees were hard fought over, as some invited interviewees were busy with projects at clients or had started working on projects abroad. We have rescheduled the workshop once due to low number of available attendees. Showing how difficult it is to get people of this caliber together in one place at the same time.

Table 7: Workshop sample

Interviewee #	Project role
I2	Data architect
I6	Solution architect
I7	Solution architect
I9	Solution architect / Lead developer

We had four goals to achieve during our workshop:

1. Find higher level risk themes
2. Link the found mitigation strategies with these themes
3. Map risks against integration framework layers
4. Validate findings

The first goal directly stems from the need to identify the risk themes. The second goal maps the found mitigation strategies against the risk themes. Such a mapping is interesting because it provides insight into which strategies can be applied to aid in mitigating what risks, but also provides information on the found set of mitigation strategies. The third goal requires the use of the integration factor model in order to find at which layer the risks are situated. This is interesting as it provides insight into the distribution of the risks amongst the different layers. Because every attendee is only responsible for a small portion of the result set, it is possible for the attendees to validate the findings from the interviews during the workshop. This will be done by addressing doubt on certain risks and mitigation strategies or adding new risks and mitigation strategies during the workshop. The timetable for the workshop is shown in the Table 8.

Table 8: Workshop timetable

Subject	Time
Introduction	10 min.
Clustering of risks (2 groups)	20 min.
Discussion of clustering	15 min.
Linking mitigation strategies to risks themes (2 groups)	20 min.
Discussion of linking	15 min.
Introducing framework 2.0	10 min.
Mapping risks towards framework	20 min.
Evaluation	10 min.
Approximate total time	120 min.

During the clustering of risks phase the attendees, who have been divided into two groups, will cluster the found risks into groups that represent the higher-level themes. We divide the group into two smaller groups to prevent that more introvert attendees have no say in how the clustering will be made. Secondly, we obtain more data as we receive results of two groups instead of one. After the clustering exercise the groups have time to discuss their clustering with the other group, and will try to collectively find one final clustering. This discussion is the third reason we divide the groups, as it is interesting to monitor. During the workshop this discussion did not lead to a collective final clustering but it lead to the acceptance of both perspectives as true and valuable. After the groups discussed both perspectives, we moved on towards the next part of the workshop. In this part we will use the risk themes originating from the clustering activity, to map which mitigation strategies help to reduce what risk themes. Afterwards we discussed the mitigation strategies, the discussion was kick started with group presentations of their results and questions concerning the quality of mitigation strategies, i.e. which strategies were considered to reduce a large number of risks, which were considered quick wins? When the discussion time limit was reached we moved the workshop forward towards the next

phase, mapping risk towards the integration factor framework. This was done in a single group, allowing open discussion on every risk that was placed into the factor framework. Results of the workshop can be found in Chapter 6. In Appendix C the presentation used during the workshop, pictures of results and pictures of the group work can be found. An overview of the entire workshop process can be found in Figure 7.

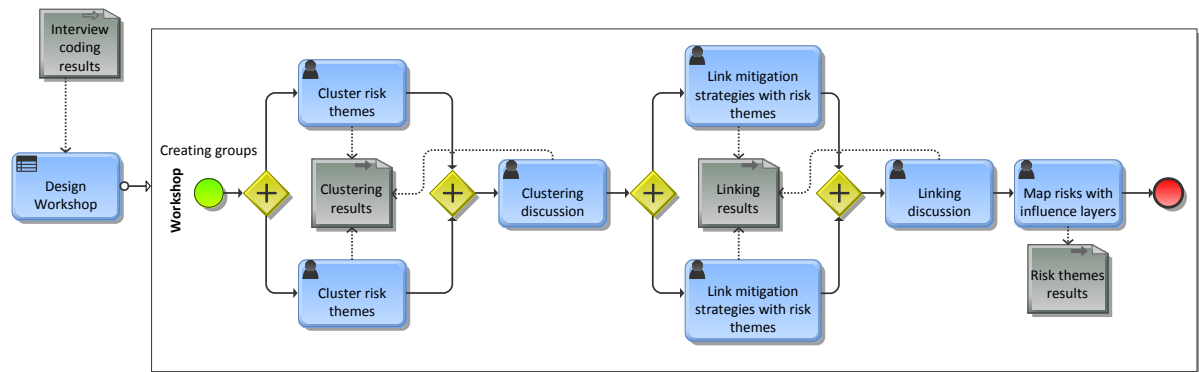


Figure 7: Visual representation of the workshop process

2.4.2 DATA COLLECTION AND ANALYSIS

The workshop attendees have received assignment envelopes prepared for every separate workshop activity. The envelopes contained activity-specific reference material and label sheets for the final result. The workshop attendees started working on their assignment and use basic workshop tools such as post-its, block notes and markers. When the groups reached a final answer they used the labels and flip-over sheets to present their results in a unified way. These sheets were collected as data from the activities. We used photos and notes to capture and describe all other notable happenings during the workshop. The two discussions and the final categorization of risks were recorded in order to recall these parts better when conclusion were made. The sheets with raw results have been photographed and can be seen in Appendix C in combination with workshop related material.

2.4.3 VALIDATION

The most important reason for organizing the workshop was to prevent potential biases because all data analysis was otherwise done by one person. The workshop was organized and facilitated by the researcher but any interference with the activities of attendees was carefully avoided. During the starting phase of every workshop activity envelopes were handed out with all the necessary tools and background information. I.e. the envelope for workshop activity one consisted of a label sheet with all the risks and documentation on every risk. This documentation of the risks exactly matches the descriptions of the risks in this document. Because all the information needed to complete the workshop activities was handed out beforehand the researcher could remain on the background and observe without interfering.

3 DRIVERS AND BARRIERS OF ERP-AS-A-SERVICE ADOPTION

In this chapter we answer our first research question: **what are the drivers and barriers of ERP-as-a-service adoption?** This question gives us insight in the reason behind the usage of SaaS solutions in the ERP landscape and provides an adequate starting point for this research. In order to provide a sufficient answer we first give a definition of cloud computing in general in Chapter 2.1, followed by an overview of ERP in general in Chapter 2.2. In Chapter 2.3 we combine the two definitions into ERP-as-a-service and we conclude with an overview of the drivers and barriers for ERP-as-a-service adoption.

3.1 CLOUD COMPUTING

The cloud in its essence a collection of IT service models, which in general can be placed into three categories: SaaS, Paas, and IaaS. In scientific literature there are other models mentioned that come forth out of the three categories i.e. CaaS, BPaas, XaaS, IaaS (Yong, Liang, & Kai, 2011) but for the purpose of this chapter we do not go into detail in all the different possibilities. Defining cloud is not an easy task because it is considered to be a strong IT hype, which results in a lot of marketing, opinions and thus different definitions. To make the situation more complex is that cloud is not a completely new thing. During interviews with experts the following was said about the situation:

“(...) there is a difference between the marketing story and what is actually happening. Cloud computing is nothing more than hosting version 5.0. We take small steps forward all the time, but if you act like nothing happened in between it looks like an enormous change.”

The quote makes it perfectly clear that we can speak of a spectrum from classic hosted solutions to cloud solutions. Because of this large gray area within the cloud and hosted spectrum, many different definitions of what cloud computing actually is have been formed. Another reason for the wide variety of definition stems from the orientation of research. To understand the range of these definitions we have presented a subset of the definitions of cloud computing that have been found in literature below. Yang, Wang, Liu, & Yu (2013) state that *“Cloud computing is the delivery of computing as a service rather than a product, whereby shared resources, software, and information are provided to computers and other devices as a metered service over a network”*. When looking at a research in the field of Building Information Management (BIM), *“Cloud computing is a centralized heterogeneous platform that enables different applications to be connected to each other through using remote data servers”* (Redmond, 2012). This definition does not even seem to describe the same concept as the definition we presented earlier. The existence of wide spread definitions of cloud computing is supported by research of Hmood & Al-Madi (2013) who found over 70 definitions of cloud computing in research.

To create clarity in this spectrum of cloud computing definitions, we have decided to adopt the essential cloud characteristics from the National Institute of Standards and Technology (NIST) together with its definition (Mell & Grance, 2011). The NIST definition is as follows; *“Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction”*. The NIST definition has been found in the majority of literature on cloud computing and is one of the few that states clear characteristics. On itself this definition might not provide the clarity we require, but in combination with the essential characteristics clarity is achieved. These characteristics of cloud computing can be found below (Mell & Grance, 2011):

- On-demand self-service
“A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider”
- Broad network access
“Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations)”
- Resource pooling
“The provider’s computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing, memory, and network bandwidth”
- Rapid elasticity
“Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time”
- Measured service
“Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service”

An overview of the cloud computing service model categories can be found in Figure 8.

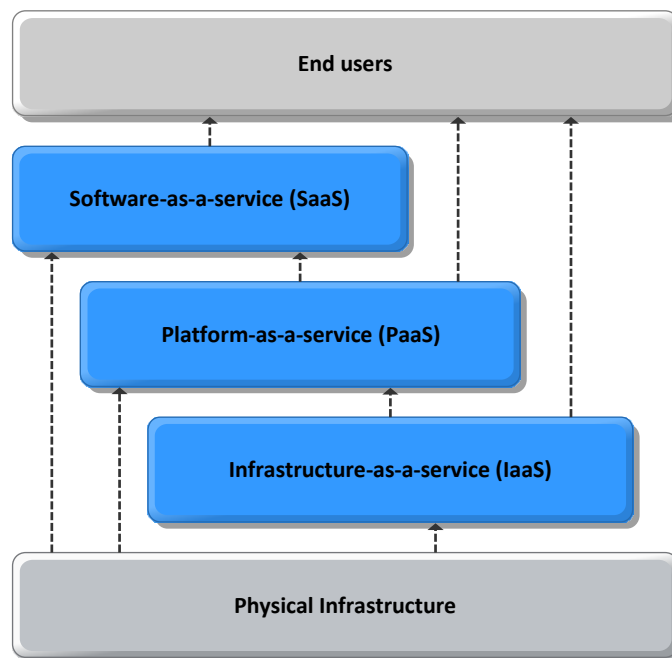


Figure 8: Cloud computing service model categories (Pearson, 2012)

Every service model takes the cloud principles and applies them their level in the stack. Infrastructure-as-a-service (IaaS) provides the Cloud user with hardware and infrastructure. Platform-as-a-service (PaaS) is one step higher in the stack and provides a platform, such as an operating system, which offers an environment for the user to use as they wish. Software-as-a-service (SaaS) only facilitates a specific application, which can be used instantly (Buyya, Broberg, & Goscinski, 2011; Raihana, 2012; Yang et al., 2013).

There are essentially four ways to organize your cloud; public, private, community or a hybrid form. When using a public cloud, everybody has access to the same cloud. Private means that the specific cloud has only one user. A community cloud is available for a group of users that have the same requirements. When you are using multiple forms of organizing your cloud you fit the hybrid form (Mell & Grance, 2011; Raihana, 2012). These ways of organizing is depicted in Figure 9. Other ways of organizing have been found but have been left out of this definition due to similar reasons as mentioned around the service models of cloud computing.

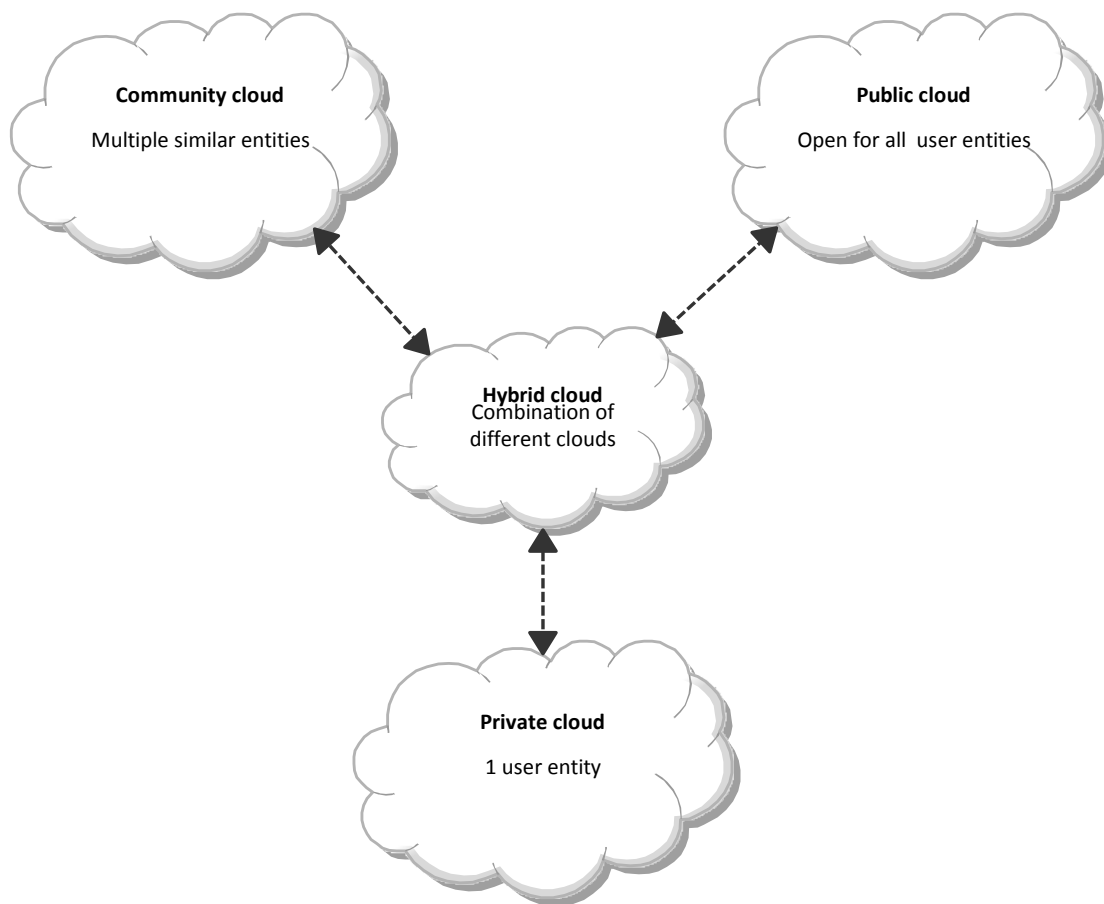


Figure 9: Cloud deployment options

3.2 ENTERPRISE RESOURCE PLANNING (ERP)

ERP systems are packaged business software systems that facilitate managing the efficient and effective use of resources (Aloini et al., 2007). Examples of ERP business software systems are finance, inventory control, customer relation management (CRM), and human resources. In our definition, an ERP system can be a combination of many software systems throughout the company or just a few, but it has to consist of at least two modules that are integrated.

The total made revenue within the ERP market was well over 25 billion U.S. dollars in 2013. During that year the market showed a slow growth through the economic crisis with a percentage of 3.8%. However, the total licensing revenue made declined slightly, showing the rise of the SaaS business model within the ERP market (Pang, 2014).

The conventional business model of the ERP system involved three actors: the using organization, the ERP vendor, and the consultant. When the using organization decides they want to invest in an ERP system they will select and implement an on-premise software package from an ERP vendor. The ERP vendor has designed the software and will maintain it. In order to help the using organizations with implementing (and sometimes selecting the right ERP vendor) ERP vendors partner up with consultants. These partnerships usually work on a certification basis, i.e. Capgemini is a diamond level partner with Oracle. To make money, the ERP vendors sell licenses that are needed by the using organization to operate the ERP system (Juell-skielse & Enquist, 2012).

3.3 ERP-AS-A-SERVICE

ERP-as-a-service is the SaaS variant of an ERP system. When we talk about ERP-as-a-service we are talking about ERP modules build from SaaS packages (Juell-skielse & Enquist, 2012). An ERP-as-a-service implementation in this sense can be seen as the process of integrating Oracle CRM on demand (SaaS CRM) with an existing on-premise ERP system. The SaaS packages are considered true cloud applications. This means that it uses a multi-tenant structure and all tenants use the same codebase and database, as well as the other cloud characteristics mentioned in the definition stated in Chapter 0.

This also implies that the ownership of the software is separate from its use, meaning that the initial investment for the software is placed at the ERP vendor instead of at the user organization. This way of delivering ERP systems is a game changer in many ways, i.e. SMEs are much more eligible as user organizations due to the lower initial investments, but it also reduces the possibilities to configure and integrate the software to specific needs (Juell-skielse & Enquist, 2012; Lechesa et al., 2012). As said in the previous chapter, ERP-as-a-service already has a measurable impact on the licensing revenues of the ERP market, indicating this is not just a marketing term.

3.4 DRIVERS AND BARRIERS OF ERP-AS-A-SERVICE ADOPTION

In the previous chapters we have stated our definitions of cloud computing, ERP systems and ERP-as-a-service. We use this chapter to present the drivers and barriers of ERP-as-a-service adoption.

To provide an insight into the reason why SMEs and LEs alike are starting to use the ERP-as-service alternative, we have discussed the possible reasoning with consultants from the fields as well as used literature to underpin these discussions. Some researchers argue that the adoption of software-as-a-service in general is simply because of technological advancements that now allow for a better usage of SaaS. Clearly technological advancements have lowered the costs of connectivity and the increased possibilities for users over the past few years (Dubey & Wagle, 2007). Consultants at Capgemini state that with the fast and more dynamic business environment, the speed with which an ERP-as-a-service model can be deployed or withdrawn from the market is a strong argument for the usage of SaaS like models. Other reasons can be allocated to less technical ones. Customers of the traditional license based business models are eager to shift due to frustrations around the original business model. Buying software licenses, paying for maintenance, and going through time-consuming and expensive upgrades is something not very enjoyable. Many customers think that paying monthly fees is the solution for the vendor lock-in the old business models instantiates. Another arguably sound reason is that the first SaaS products have been deployed successfully, i.e. Salesforce.com, and the

technologies used are becoming more and more proven (Dubey & Wagle, 2007). There are also indications that ERP-as-a-service opens up the ERP market for SMEs in particular due to the reduced upfront investments (Addo-Tenkorank & Helo, 2011; Lewandowski et al., 2013). In order to find out whether or not these arguments are supported by a wider group of experts we have undertaken an explorative research into the scientific literature written on the subject.

When looking at the drivers and barriers, or opportunities and challenges as they are often disguised, of ERP-as-a-service there are as many perspectives as there are actors. As mentioned the traditional actors are the user organization, the ERP vendor, and the consultant. The perspectives of the ERP vendor and the user differ the most, while the consultant assimilates perspectives of both parties (Juell-skielse & Enquist, 2012).

In the table below we have listed the drivers and barriers of ERP-as-a-service adoption from the user perspective found in scientific literature.

Table 9: User opportunities and challenges (Juell-skielse & Enquist, 2012)

User opportunities	User challenges
Reduced up-front investments*	An ERP project must be a business initiative
Decreased implementation costs and risks	Lack of senior management involvement
Predictable and lower costs*	Lack of detailed systems implementation plan
Productivity improvements	Project escalation and lack of control
Stock reductions	Lack of policies and laws*
More focus on IT-value	Poor use of consultants
Customer responsiveness	ERP-as-a-service requires local software
Complete service offerings from several vendors	Less customization and integration possibilities*
Increased bargaining power	Large dependency on vendor*
Access to and flexibility to choose between state of the art technologies*	Less availability, reliability and performance*
Access to reliable, secure and scalable infrastructure	Increased security risks*
Up-to-date software*	More rigid organizations
Remote access from anywhere at any time*	Structural changes*
Easier access to technical expertise*	Redistribution of responsibility
Easier version management	Lack of alignment
Improved processes*	Lack of project team expertise
Order cycle improvements	User resistance
Improved financial close cycle	High demands on process orientation
Improved information and transparency	
Increased integration of information	
Increased standardization	
More proactive purchasing behavior	
Simplified phasing of implementation	
Single point of contact	
Increased focus on core competencies	

The opportunities and challenges marked with an asterix sign have been directly found in other research (i.e. (Araujo et al., 2014; Benlian & Hess, 2011; Lechesa et al., 2012; Lewandowski et al., 2013; Raihana, 2012; Schubert & Adisa, 2011; Torbacki, 2008), and thus can be seen as the more widely accepted drivers of ERP-as-a service adoption. This also regards the table on the supplier perspective.

In our research question on the drivers and barriers of ERP-as-a-service adoption we have not made discrepancies between perspective, this is explicitly done due to the interest we have in both the user perspective and the vendor perspective. The consultancy firm that hosts this research has a need to understand both the vendors' perspective and the users' perspective. Therefore we present the former in the table below.

Table 10: Supplier opportunities and challenges (Juell-skielse & Enquist, 2012)

Supplier opportunities	Supplier challenges
More predictable revenue flows*	High initial investments for starting a SaaS business*
Potentially greater profit*	Initial reduction in turnover*
A need for standard applications in the market	Anticipate customer requirements
Expansion of potential customer base*	Serviticize software products
Improve supplier brand	Offer customizable services
Increased ability to offer more choices to customers	Contractual changes*
Shorter sales cycle	Increased demands on fast updates
More focus on IT-value	Manage service transitions
Improved transparency in pricing	Increase responsibility for customer operations
Strong lock-in effect of customers	Difficult to manage complex networks of SaaS suppliers
Lowered risk of pirated software	Address end-user presumption in service architecture
Economy of scale in development*	Manage development effectively
Improved economy of scale in distribution and operation	Develop for flexibility
Increased flexibility	Manage complexity of enterprise applications
Decreased risk	Manage service operation and maintenance effectively
Leverage domain area knowledge	Manage security effectively*
Knowledge aggregation	High requirements on service availability, performance, and scalability
Improved possibilities to build application expertise	Balance over- under capacity
Increased technological capabilities	Support several version of software
	New sales processes

It becomes interesting when we compare our findings from literature with the findings from our initial discussions or compare the different perspectives with each other. The customer described that using ERP-as-a-service reduces the risk for a vendor lock-in. When we think about the absence of high initial investments and standardized applications it seems this could be a fair assumption. After looking through the literature on the topic however, we can see that the risk of vendor lock-in still exists. It gets even more clear when we see that the vendor lock-in is considered an opportunity from the vendors perspective. The idea that with the SaaS business model ERP systems are becoming more affordable for SMEs and can be deployed more rapidly has been found in the literature and can be seen as viable.

4 FACTORS INFLUENCING INTEGRATION RISKS

This chapter gives the answer on research question II: ***What factors influence information system integration risks?*** It will be divided into two main sections that are the result of the research approach. The first chapter contains the results obtained from the explorative literature research. It will be divided into three parts that discuss the factors found during our study. Chapter 4.1 will discuss integration scenarios, followed by a chapter on the different integration risk levels, and finally Chapter 4.3 will present the different applications to be integrated. The method we used to gather data could be called an explorative literature study. Beforehand we were unable to predict what factors would be found and therefore using more refined research methods, such as a systematic literature study, were inappropriate. We used explorative interviews, our knowledge within the field of EAI, and the Scopus search engine to gather information on the factors.

The second part, starting from Chapter 4.5 represents what we learned during the interviews with experts. As stated in the research approach we have used the results of the first section in order to explain our scope to the interviewees. From this explanation feedback was collected and used to reshape our initial view on the factors that cause integration risks.

4.1 INTEGRATION SCENARIOS

In order to get a clear view of the different integration scenarios it is vital to understand the different ERP delivery models used in practice. We will use Chapter 4.1.1 to provide an overview of the known delivery models. In Chapter 4.1.2 we will elaborate on what these delivery models mean for integration.

4.1.1 ERP DELIVERY MODELS

There are three general delivery models for ERP: On-premise, hosted, and ERP-as-a-service. The most traditional form of ERP delivery is the on-premise model. On-premise means that the hardware and software of the ERP is inside the domain of the organization. When the hardware is being hosted at a service provider outside the organization domain we are talking about a hosted ERP. When the ERP is being offered as a SaaS we talk about ERP-as-a-service. Due to the fact that many hosted ERPs are marketed as a SaaS, we make a clear distinction between the last two delivery models.

In order to show the advantages of each delivery model we use the work of Duan, Faker, Fesak, & Stuart, (2012). They have conducted an explorative literature study and conclude with a comparison between different delivery models. The advantages that they found in more than one study and for which they found no contradictions have been stated below. There are many studies regarding the advantages of SaaS ERP but not many of them compare the three deliver models directly. To avoid proliferation of advantages and disadvantages we decided to use their research as our main source.

4.1.1.1 ON-PREMISE

The implications for having the hardware and software of an ERP on-premise means that the user organization owns the ERP and pays through a licensing model for the software. The user organization is responsible for maintaining the hardware, providing the space that the hardware requires, and handling disaster recovery (Duan et al., 2012).

Based upon Duan et al., (2012), Table 11 represents the advantages of having a on-premise ERP.

Table 11: On-premise ERP advantages (Duan et al., 2012)

On-premise ERP advantages
No non-depreciable subscription fees
Higher level of independency from the ERP provider
Functionality rich to satisfy the back-office needs of organizations in all types of industries
Enables extensive customization and complex integration
Low dependency on deficiency of network reliability and speed
Ease of retaining on-premise legacy systems
Easier integration with on-premise systems that require low latency
Enables high level of security and confidentiality
Easier compliance to data & environmental regulations

4.1.1.2 HOSTED

With a hosted ERP a service provider host the hardware for the user organization, while the user organization still owns the ERP system. The ERP can be connected to the user organization by a direct network connection or the internet (Duan et al., 2012). When the third party service provider hosts the ERP for multiple clients, they create a new environment for every client.

Based upon Duan et al., (2012), Table 12 represents the advantages of having a hosted ERP.

Table 12: Hosted ERP advantages (Duan et al., 2012)

Hosted ERP advantages
Functionally rich to satisfy the back-office needs of organizations in all types of industries
Enables extensive customization and complex integration
Allows hybrid deployment strategy
Ease of retaining on-premise legacy systems
Enables high level of security and confidentiality

The list for hosted ERP advantages is longer, if we would add the advantages found by Duan et al., (2012) in only one source, but according to their research the impact of the hosted ERP advantages are smaller than impacts of advantages of either on-premise ERP or ERP-as-a-service. This can be explained by acknowledging this delivery model as the middle ground between the two other types. Hosted ERP enjoys advantages of both more extreme delivery models at a lower impact, which could be an antecedent for the many contradicting advantages found in research.

4.1.1.3 ERP-AS-A-SERVICE

ERP-as-a-service means that the ERP system or module uses a SaaS service model. This implies that the user organization is not owning the software, and can be seen as 'renting' application and database space (Duan et al., 2012). When the SaaS provider has multiple client, or 'tenants', they all use the same database and codebase, amongst other features mentioned in our cloud definition. This is significantly different then the hosted ERP delivery model.

Table 13 represents the advantages of having a ERP-as-a-service, based upon Duan et al., (2012).

Table 13: ERP-as-a-service advantages (Duan et al., 2012)

ERP-as-a-service advantages
Lower upfront costs (hardware, user licenses, implementation, excluding training and customization)
Lower operating costs and efforts (energy, maintenance, configuring, upgrades, IT staff costs)
Scalability (highly elastic infrastructure capacity), faster time to market
Rapid implementation, easier to switch among IT providers
Enables enhanced focus on core competencies
Rapid acquisition of bug fixes and new functionality
Improved accessibility, mobility, and usability
Easier integration with other cloud services

4.1.2 ERP INTEGRATION

In the previous chapters the different ERP delivery models are explained. In this chapter, we elaborate on the different integration scenarios these delivery models are creating. We will do this by presenting the so-called integration 'archetypes'. These integration scenarios are the fundamentals for every possible integration scenario.

When defining what an ERP system is, the word integration to plays a vital role. In many studies ERP is linked with the ability for firms to integrate their primary business processes (Addo-Tenkorank & Helo, 2011; Duan et al., 2012; Giachetti, 2004; Hasselbring, 2000; Lee, Siau, & Hong, 2003; Malhotra & Temponi, 2009; Themistocleous, Irani, & Love, 2002; Themistocleous et al., 2001; Utomo, 2013). Without this integration, the ERP modules become a group of stand-alone silo applications without the power to provide any real sustainable advantages (Sumner, 2000). The hidden strength of the ERP systems lies in synergies; combining data for better reports and a more efficient process through data sharing are two examples of the advantages of an integrated system. Due to this reason, integration issues have been discussed for as long the ERP system has been in existence (Themistocleous et al., 2001). In order to place this study in perspective within the existing field of Enterprise Application Integration (EAI), we provide an overview of the different ERP integration scenarios that can be identified.

The examples used to explain the different scenarios will all cover the same fictional company. The company consists of three departments; an HRM department, a sales department, and a financial department. All of these departments use their own Information Systems (IS). In order to understand the differences per scenario, we have drawn the initial scenario without integrations and can be found in Figure 10. This scenario represents the silo structured organization with every department acting like an 'independent' organization.

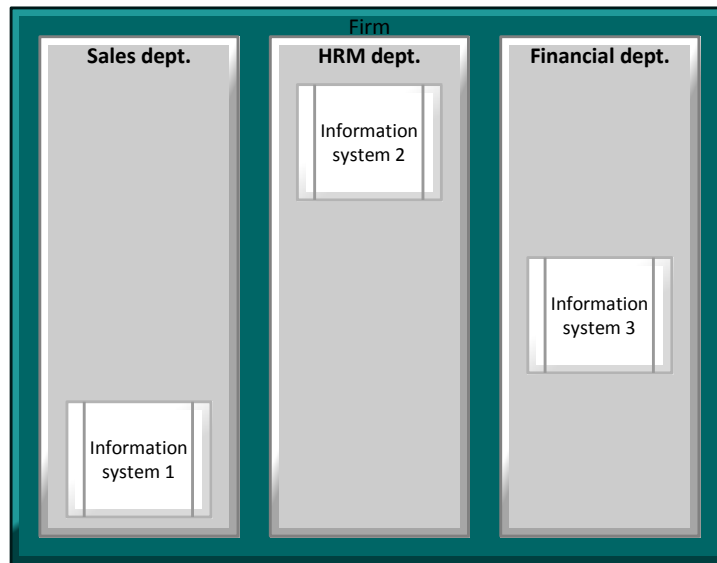


Figure 10: Fictional company without integration

4.1.2.1 ON-PREMISE TO ON-PREMISE

Arguably the oldest integration scenario is the on-premise to on-premise scenario. Challenges around this type of integration have been around as long as information systems have been built. In the early days of the ERP, systems were not implemented as an 'ERP system' but more or less evolved towards ERP because of the movement away from the siloed organizational structure (Giachetti, 2004; Puschmann & Alt, 2001; Utomo, 2013). In order to cope with increased competitive pressures and highly dynamic world, a different approach was necessary (Lee et al., 2003), which led to more integration between business functions. This evolution happened on all business levels, including Information Technology (IT), as it became apparent that it was inefficient for departments to work as separate entities. It is i.e. imaginably more efficient to enter data once and let it be used throughout the company, instead of having your sales department entering client data separately from the finance department. It could be argued that this change in strategy was the primordial sea of the ERP system. The point is that different systems, initially not made to work with other systems, had to be integrated. Later on, the same scenario occurred when additions were made for existing on-premise ERP systems, which in turn had to be integrated.

On-premise to on-premise integration means that both of the IS reside within organizational boundaries. In order to illustrate this scenario we use our fictional company. The advantages of breaking down the silo culture have affected firms in such a way that the departments operate less as separate units and have integrated their IS. In Figure 11 an overview of this situation can be found. It is important to note that the connections shown do not imply how the integrations are made, they just show what systems are integrated.

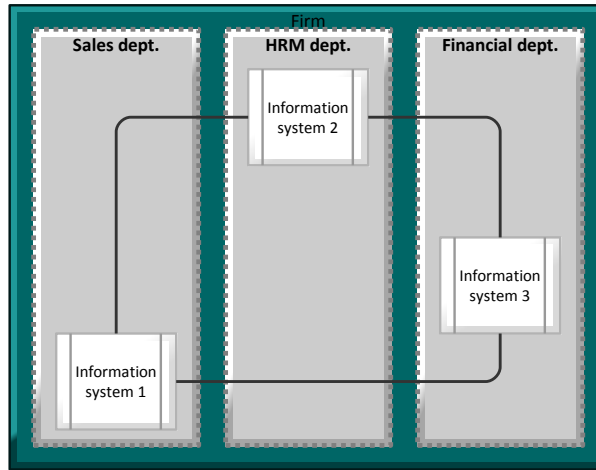


Figure 11: Fictional company with integrated information systems

4.1.2.2 ON-PREMIS TO HOSTED

On-premise ERP system require the using organization to control the infrastructure and platforms, absorb the costs for server maintenance and placing, and to be responsible for disaster recovery (Duan et al., 2012). All these responsibilities and resources needed to operate on-premise ERP systems led to a new business model, hosted ERP. As we recall from 4.1.1.2, hosted ERP means that the ERP system is offered to you as a service by a provider that hosts the servers and running the ERP system from somewhere else. The actual service offering can then happen over the internet or via a direct network connection. The technological difference of this scenario from the on-premise to on-premise scenario lies in the fact that a part of the ERP system lies outside the organizations domain.

In Figure 12 we can see the situation for our fictional company, which is now using a hosted ERP system to run their IS of the HRM department. Again the connections do not imply how the connections are made, they just show what entities are connected.

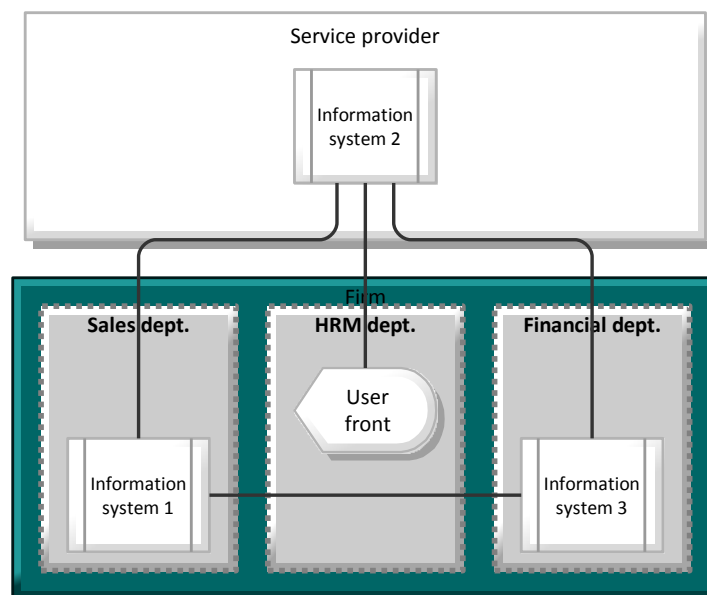


Figure 12: Fictional company with a hosted ERP system

4.1.2.3 ON-PREMISE TO SAAS

SaaS is one of the IT service models that originated from cloud technologies. It directly targets end users or business and delivers the same application to multiple customers that use the same database and object code (Duan et al., 2012).

Again, we have depicted the situation for our fictional company but now they are using a SaaS ERP to cover their HRM IS.

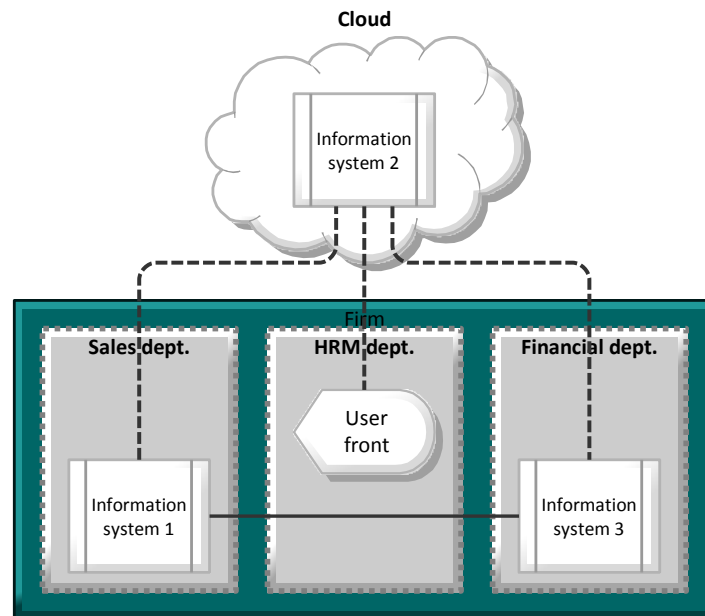


Figure 13: Fictional company with an ERP-as-a-service module

The differences with SaaS ERP and Hosted ERP is that the network connection is going through the internet, and the multi-tenant structure allows multiple users to use the same code- and databases. This is shown in Figure 14. This is fundamentally different than a hosted ERP where every client gets a private hosting environment. Another difference compared with the hosted ERP is that by using a SaaS the organization is no longer the owner of the software but merely a tenant.

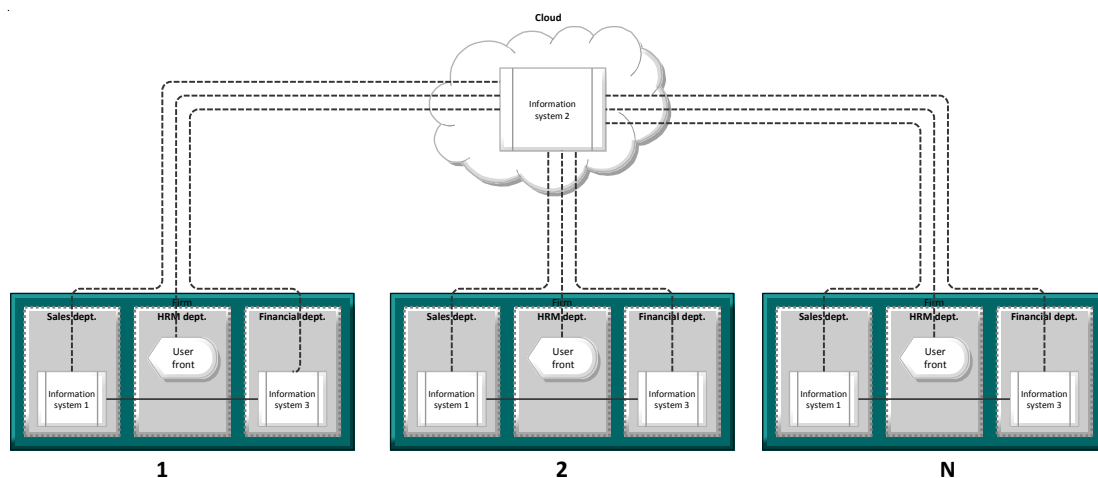


Figure 14: Multi-tenancy of the ERP-as-a-service component

4.1.2.4 DIFFERENT SCENARIOS

We used the integration scenario archetypes to explain how the integration scenario can affect the integration risks. The main pivots on which the archetypes differ from one another are ownership of the software and whether the ERP module is placed within or outside the organization borders.

By only explaining the archetypes, the tentative reader could have come up with combinations of delivery models that have not been mentioned. I.e. hosted ERP to SaaS is not explicitly mentioned but is not unheard-of in practice. We argue that the archetypes cover the integration risks present in every possible combination. For this particular example we argue that the organization owns one side of the ERP and has to connect across network boundaries with a SaaS application. This scenario is similar to the on-premise to SaaS archetype. This exercise can be done with all other possible combinations.

4.2 INTEGRATION RISK LEVEL

In the previous sections, different integration scenarios have been explained. In every figure lines were used to indicate integration, but notations were made to show these lines had nothing to do with how these ERP modules had been integrated. The second plane of ERP integration, apart from the scenarios, is on what level the integration risk takes place, or in short the integration risk level. The integration risk level can be divided into two main levels: Technical and organizational risks.

Before we dive directly into the different integration risk levels we take a moment to elaborate about integration itself to create a clear definition that will operate as a starting point of this chapter. After defining integration we will elaborate on the found integration risk levels.

4.2.1 INTEGRATION

Definitions of integration are widespread and are known to differ between studies. The reason for different definition of integration stems from the tendency of technology to evolve over time, and from the fact that integration can happen on multiple levels (Giachetti, 2004; Puschmann & Alt, 2001). The last argument is visible when you i.e. look at the difference between database integration and information system development research. In database research integration is seen as the activity to create a global database schema out of a group of distributed database schemas (Batini & Lenzerini, 1987). If we look at research regarding information system development, two types of integration are mentioned; strategic integration and operational integration. These types of integration govern respectively the integration between the business strategy and the IT strategy, and the integration between organizational infrastructure and processes (Henderson & Venkatraman, 1999). The difference between integration definitions lies on a spectrum ranging from technical towards functional. In order to achieve a clear understanding of what is meant with integration in this study, we use the levels defined by integration research from the pre-cloud era (Giachetti, 2004; Puschmann & Alt, 2001). According to these studies, there are four different levels of integration: network, data, application and process integration. This implies that integration in an enterprise system context can mean that system is physically connected, data can be shared between systems, applications can interoperate, and business processes are coordinated. We are using these layers to explain technical integration risks and therefore we add another layer on top of the initial four: the integration governance layer.

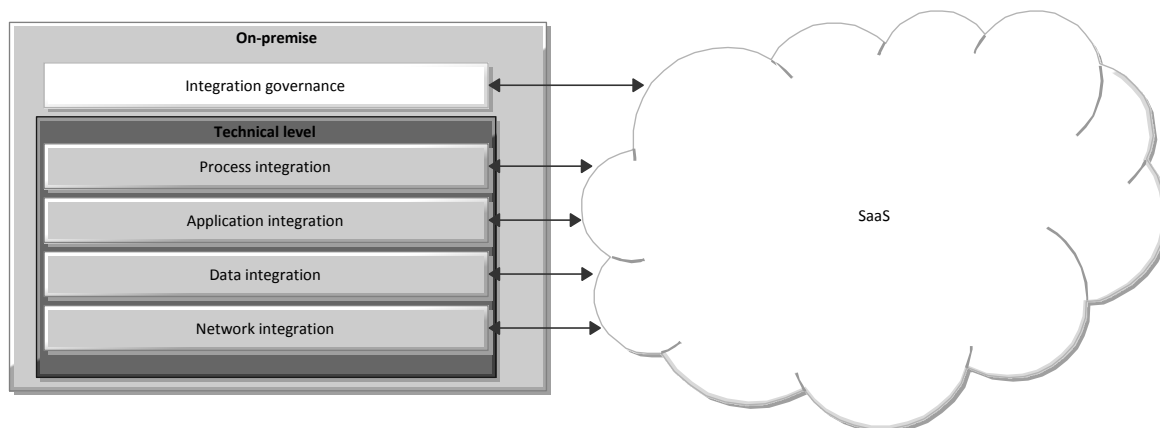


Figure 15: Five different integration levels

In the chapters below, the different levels of integration are explained.

4.2.2 TECHNICAL INTEGRATION RISK LEVELS

Many studies have shown that integration is a direct antecedent for the success of an ERP system and successful arguments can be made that more sophisticated integration means a better service (Hai & Sakoda, 2009; Ram, Corkindale, & Wu, 2013). In the sections below, we elaborate on the technical integration levels starting from network and ending at the process level, following the path from basic integration towards more sophisticated integration.

4.2.3 NETWORK CONNECTIVITY

Integration on the network layer revolves around connectivity. The goal of this layer is to establish a connection that ensures that data and/or messages can be sent from one system to another, it does not ensure that the data that is send can be interpreted. The issues relate around the physical heterogeneity of the hardware, devises and systems found in the physical network (Giachetti, 2004).

4.2.4 DATA SHARING

Integration on the data level revolves around data sharing. On this level, integration provides the necessary data for enterprise systems to utilize its business functions. The goal of this layer is to share data between two or more systems. According to Giachetti (2004), data sharing must overcome data schema diversity problems described by Batini & Lenzerini (1987). This data schema diversity can be subdivided into issues related to different perspectives, equivalence among constructs, inter-schema properties, and weak semantics (Giachetti, 2004).

4.2.5 APPLICATION INTEROPERABILITY

Integration on the application layer revolves around interoperability. Applications tend to use different programming languages, locally defined date and messages, which leads to heterogeneity. To overcome this problem, the goal on the application level is to allow one application to access and use data generated by another system. Issues related to the interoperability between applications are placed in the application layer (Giachetti, 2004).

Application interoperability is generally achieved by using middleware. Middleware is software that hides the complexities of integration from users and developers by presenting itself as the integrator. In short, a generic middleware allows applications to send and receive messages between each other by using services of the middleware. By using the middleware, applications do not need to understand other applications specific languages or data formats but simply let the middleware handle translation of the message and further routing of the message (Blair, Coulson, Robin, & Papathomas, 2009; Giachetti, 2004).

4.2.6 PROCESS COORDINATION

Integration on the process level revolves around coordination. Applications often support one or more business tasks that together form business processes. In these business processes separate functional units or even other entities in the supply chain together work towards the same goal. Issue related to the coordination of these processes i.e. task dependencies and goal alignment are placed on the process level (Giachetti, 2004).

4.2.7 INTEGRATION GOVERNANCE

The highest level of integration risk covers the integration risk governance. Due to the technical nature of papers of Giachetti (2004) and Puschmann & Alt (2001) we argue that there is one integration risk level missing. During discussion with consultants at Capgemini and the supervisors of this study some speculations were done on the outcome of our research. When we tried to place one of these speculated risks on the integration levels described above, we found that it did not fit the technical nature of the integration levels.

One of these speculated risks were the changes in APIs. This risk can be considered to be on the application level as it takes care of interoperability between applications. However, due to the tenant structure of SaaS solutions these changes happen without any control of the user organization. The risk that stems from this fact, for this example called 'not being ready', is something we could not place on any of the layers. This would be a specific risk the user organization is taking when using ERP-as-a-service, but it is rather organizational of nature instead of technical. Whether or not this example represents a real risk, it paved the way for a more organizational integration risk level on top of the more technical ones.

4.3 APPLICATION TYPE

The third factor that has been identified to influence the integration risks is the application type. In scientific literature, the following types have been identified; Homogeneous with one instance, homogeneous with several instances, and heterogeneous (Breiter & Naik, 2013; Puschmann & Alt, 2001). Many of the integration levels from Giachetti (2004) also refer to the heterogeneity of the different layers. In the table below, the difference between the application types are elaborated on.

Table 14: Application types

Application type	Explanation
Homogeneous with one instance	When an entire process is supported by one application and one database instance
Homogeneous with several instances	When identical processes are supported by several identical application that run on different environments and depend on separated databases
Heterogeneous	When several different processes in different business units are supported by several

4.4 CONCLUSIONS FROM EXPLORATIVE RESEARCH

To conclude our journey towards the factors that cause the integration risks, we see that the risks stem from three factors. First, it is important to get a grip on how the integration journey looks like; where do the systems that need to be integrated originate from? This factor is called **the integration scenario**. Secondly, we need to understand the depth of the integration; do we need to attain data integration, are the systems part of one seamless process, or is our organization affected due to the integration environment? This factor is called the **integration risk level**. The last factor that influences risks is the type of the applications or systems we are actually integrating. This factor is creatively called the **application type**.

In our search towards these factors that cause integration risks, we have widely discussed the results and some potential causes that have not made it into the results. Two of the topics in these discussions are worth mentioning. First, a strong argument can be made that the application type is not very valuable as there are not many homogeneous integrations that cross on-premise to SaaS boundaries. However, it is possible to mask the heterogeneous applications by homogeneous portals (Breiter & Naik, 2013). Still one could argue that the application type is lacking added value considering the orientation of this research. The argument to keep the application type as an integration risk is as follows. The mapped integration risk causes, span a wider scope than the scope of this research, as it is used to place this specific research topic in the overall field of integration risks. When one would undertake research into the risks of the other integration scenarios besides on-premise to SaaS we argue that the application type becomes more important. Therefore we argue that if we want to provide an overview of factors of integration risks we need to take the application type into account. The second topic of discussion was whether or not we should add the integration architectural pattern. Some more functional oriented consultants argued that there are different risks involved when different integration architectural patterns are used. Although we agree with these consultants, a more technical oriented consultant provided compelling arguments why the architectural pattern should not be taken into account in this specific research; although different architectural patterns provide different risks, these risks come forth from the specific pattern and are scenario and application type independent. Therefore it is argued that they should not be taken into account if we want to find specific scenario related integration risks.

In order to depict how this study is situated in the field of integration risks, we have mapped the three factors together in a three dimensional framework. Within the framework we have pin-pointed the scope, and the specific combination of factors this study focuses on.

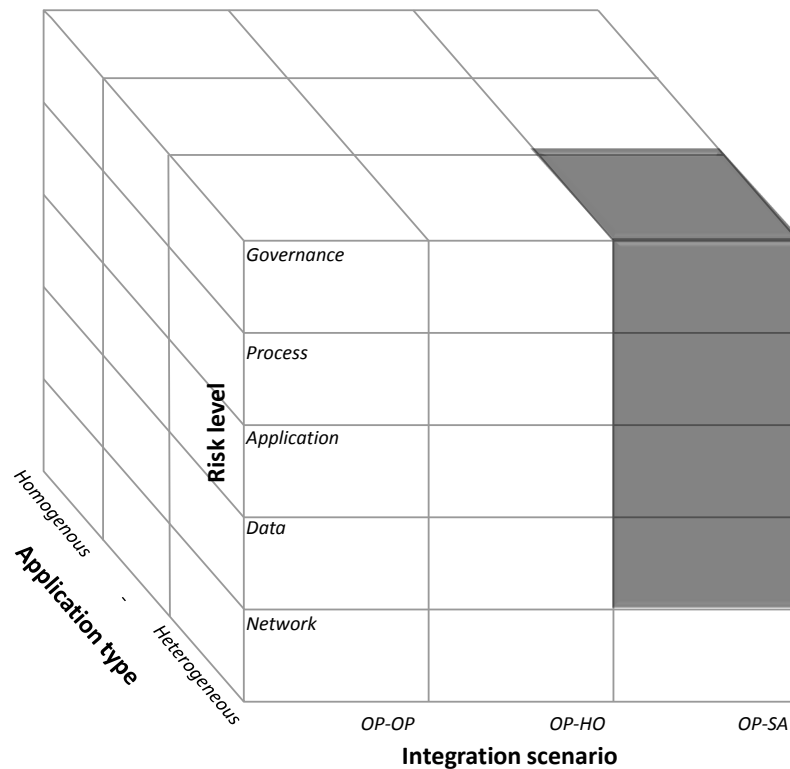


Figure 16: Integration factors framework

As can be seen in the model, depicted by the gray highlight, this study focuses on integration between on-premise ERP and SaaS ERP. The reasoning behind this decision is threefold. The first argument is due to the potential challenge in these types of integrations: the integration crosses organizational borders and encompasses both organizational owned software and ‘rented’ software. This combination of aspects is unique among the different scenario archetypes. The second reason stems from practice, especially from the environment this research was situated. Many requests from the clientele of the Oracle division consider SaaS ERP add-ons. Most of the known clientele have on-premise Oracle systems in place and now want to acquire new functionality and take advantage of benefits of the SaaS service model. This leads us to conclude that this particular scenario is a challenge that is faced every day. This conclusion is supported by researchers that argue that the future holds hybrid landscaped with on- and off-premise systems (Goyal, 2010). Third, on-premise to SaaS integration risks are an interesting research niche as it has received limited attention from research.

Another conclusion we can make from the gray highlight of the model is that we focus on the top three levels of integration, and thus neglect the network level. This decision has been made after receiving three independent but similar opinions of experts in the field of integration. Both my supervisor, Jos van Hillegersberg and Ronald Giachetti (2004), who both have done substantial research in the field of EAI, argue that the networking level is not as interesting as the other levels. This is due to the fact that eventually everything can be integrated at a network level (although this might mean a lot of headaches for some people). They argue that when this connection has been made, the ‘integrating’ can begin. The third expert was a technical solution architect at Capgemini, who used the exact same reasoning. The last axis of the model covers application types, which led to the easiest ‘decision’ that had to be made on the scope of the research project. Due to the fact that we can hardly speak of homogeneous application when we talk about on-premise to SaaS application, we can say that we only focus on the specific risks considering heterogeneous, on-premise to SaaS integration risks

stemming from the top three integration risk levels. The same conclusion on the heterogeneity of the application types has also been made by other studies (Breiter & Naik, 2013; Redmond, 2012).

4.5 FEEDBACK ON THE INTEGRATION FACTORS FRAMEWORK

During the interviews, we used the three dimensional framework that resulted from the explorative literature research described in the previous chapters, to describe the characteristics of the specific risks and mitigation strategies we were looking for. We asked the interviewees to give feedback to our theoretical model of factors influencing integration risks in order to improve the framework. In this chapter we describe what feedback we have received and how our initial framework morphed into a more accurate version.

Six out of nine interviewees criticized the application type axis as being the weakest of the three. This happens to be the axis discussed heavily among the researcher and supervisors before initiating with the interviews. The same arguments that were used in the initial discussion; questioning the impact of the dimension on the specific risks returned through the feedback of the interviewees. Some interviewees also mentioned other dimensions, such as the spectrum from batch to real-time integration, to be far more interesting in practice. I.e. one interviewee stated:

“(...) I mostly work in SOA environments, so homogeneous or heterogeneous is no issue for me. (...) the majority of communication is in an XML format, which makes it indifferent what kind of applications they are (...)”

More governance oriented interviewees argued that the differences between on-premise and hosted is small. The interviewee had a couple of compelling examples, such as large governmental agencies which ran applications on-premise but this ‘premise’ was divided into multiple sub-domains. This meant that for reaching the on-premise applications firewalls had to be crossed, making the scenario essentially the same as hosted integration. Still, during the interviews the different integration scenarios were proven to be useful in keeping the conversation on track and differentiating between general integration risks and specific SaaS integration risks.

Feedback on the integration levels was also given. Technically oriented interviewees stated that multiple layers were missing below the network layer, and even between the network layer and the data layer. As stated in Chapter 4.4, we were not interested in the network layer due to widely supported arguments. The cases made for the inclusion of these more technical layers revolved around auditing, laws and standards. One expert made the following argument.

“(...) we as Capgemini, have contracts which state that we and all of our partners are ISO270001 certified according to our interpretation of the ISO270001 security model. (...) This means that you need two-way authentication on the door of the datacenter. (...), the same goes for your network, infrastructure and all the layers above.”

The important point made here is that by integrating with a SaaS solution, the infrastructure, however not under your control, becomes part of your overall architecture. This means i.e. that when the user organization states its compliance with ISO270001, they have to be sure that their SaaS service provider is compliant as well. Thus ignoring these layers means missing important parts of the puzzle. More general feedback on the layers revolved around the governance layer. Although mentioned to be critical multiple times, seven of the nine interviewees mentioned in one way or another that technically the integration between an on-premise application and a SaaS application holds no real challenges. The following statement is an example of such a mentioning.

“For me these integration scenarios are technically not challenging, the integrations are more or less the same (..)”

On the other hand, seven interviewees mention that governance becomes more important. The following statement represents these opinions.

“The challenges lie in governance; how do you cope when Oracle decides to roll out a patch because of a security bug? How do I act? What is my next move? Ok, it will affect my services, but which processes in my organizations use these services? Or which departments are going to feel the consequences of these changes?”

They also mentioned that governance is covering all the layers and should not be depicted as a separate layer on its own. Another interviewee argued that the same goes for security as it was missing entirely in the model.

4.6 CHANGES MADE TO THE INTEGRATION FACTOR MODEL

The feedback on the model made clear that the views of practice and our abstraction from theory on integration differ substantially and gave us greater insight in what the different risk factors are. That being said the model made differences between different types of integration more clear and was overall considered useful during the interviews. We also argue that there is a significant theoretical value within the model as it allows us to map at what level specific integration risks of on-premise to SaaS integrations reside. Therefore we argue that the model, however in need of modification, is useful as a tool to discuss differences between different integrations.

We made a total of three changes to the model. The first change encompassed removing the application type axis of the model. During the interviews it became clear that this is not something influencing risks of integration projects. The second change was to reposition the governance layer more vertically to represent the presence of governance along all technical layers. The third change was to depict security as a new vertical layer. During the interviews, it became clear that security was something different than governance, but like governance spanned all layers.

Some decisions orienting from the feedback can only be admired due to their absence in the model. The so-called ‘missing layers’ have not been added to the model. These missing layers would not have improved the parsimoniousness of the model, and, more importantly, would not describe integration levels which are considered to be the topic of this research. We do however argue that challenges related around compliance, audits and standards are the results of integration and are therefore considered welcome in the vertical governance layer. Another decision covers the absence of architectural patterns such as batch or real-time integrations. We argue that these patterns do influence the risks of an integration project, but are considered to be too specific for our model. When it comes to on-premise to SaaS integration we consider the research towards risks specific to certain architectural types the ‘next step’.

In Figure 17 the new and improved model is presented. The black lines show the scope of the research; the rest of the model is grayed out. The model is considered to be the visual representation of the answer to research questions II.

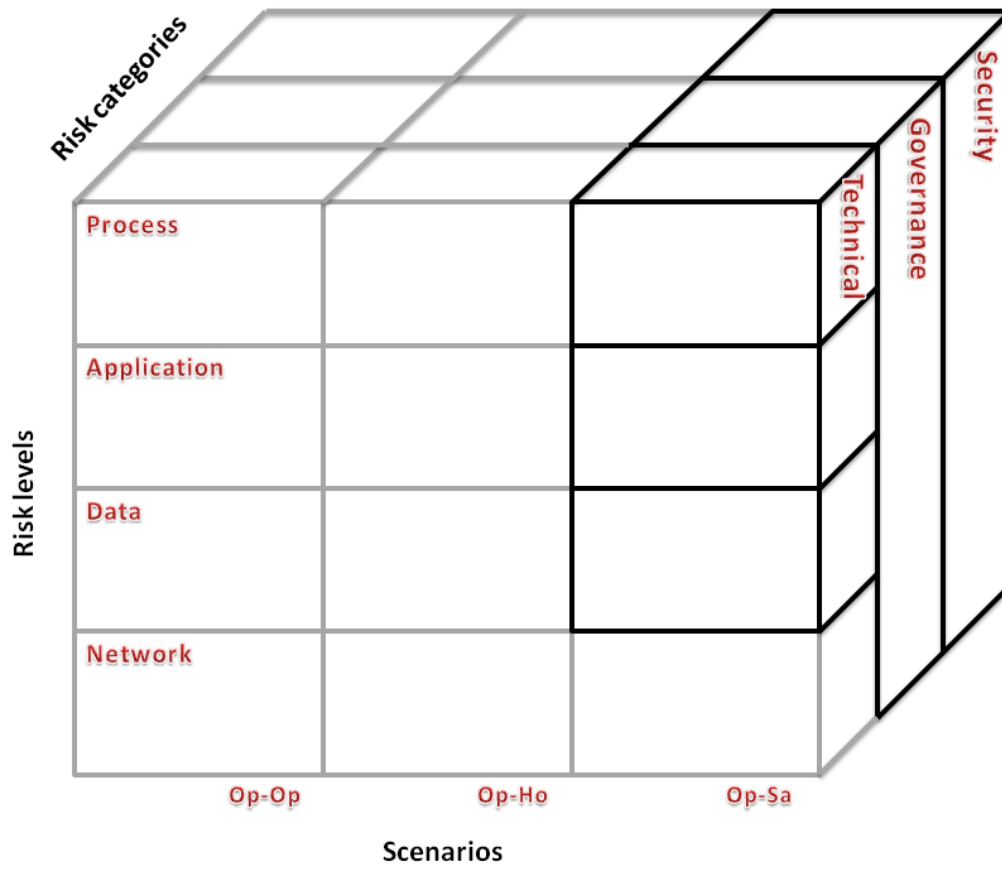


Figure 17: Integration factors framework 2.0

5 RISKS AND MITIGATION STRATEGIES

In this chapter we provide answers to research question III and IV. For the known risks and mitigation strategies from the scientific community we will use the systematic literature research method as described in Chapter 2.2. The results of this literature research can be found in Chapter 5.1. To collect the knowledge from practice we have conducted semi-structured interviews with experts as described in Chapter 2.3. The interview process and results obtained from the interviews can be found in Chapter 5.2. In Chapter 5.3, an overview is given with the found risks and mitigation strategies and our conclusion are made.

5.1 FINDINGS FROM THE SCIENTIFIC COMMUNITY

In our section describing the search protocol, we have argued that the research towards specific on-premise to SaaS integration risks is immature and, after conducting the systematic literature research, this statement can be confirmed. None of the studies that resulted from the research pool specifically focused on on-premise to SaaS integration, some even did not go further than mentioning general cloud risks. This is acknowledged in the topic matrix found in Figure 18.

In order to provide a better overview of the resulting papers we have categorized the papers on publication year. Together with the topic matrix, it gives a better insight in our results.

Table 15: Overview of resulting papers

Publication year	Papers
2014	(Kruize et al., 2014)
2013	(Adalety, Poppe, & Braa, 2013; Breiter & Naik, 2013; Hmood & Al-Madi, 2013; C. Liu, Yu, Zhang, & Guo, 2013; Neghina & Scarlat, 2013; Utomo, 2013)
2012	(Bolloju & Murugesan, 2012; Haimes & Chittister, 2012; Kotlarsky, Oshri, & Willcocks, 2012; Li, Shen, & Liu, 2012; Masiyev, Qasymov, Bakhishova, & Bahri, 2012; Mathew, 2012; Mislevics & Grundspenkis, 2012; Pahl & Zhu, 2012; Redmond, 2012)
2011	(Graupner, Basu, & Singhal, 2011; Lackermair, 2010; Yong et al., 2011; M. Zhu & Risch, 2011)
2010	(Aalmink, Gómez, & Schubert, 2010; Goyal, 2010; Z. Zhu, Chen, Song, & Liu, 2010)

We have mapped the papers against the topics to give insight in what topics we encountered during our systematic literature search. This is another piece of compelling evidence that the research field is not mature as none of the papers have on-premise to SaaS integration as a main topic. Most papers discuss cloud computing and some are integration oriented, and these papers made it possible for us to derive a number of risks and mitigation strategies.

Papers/ Topics	Specific cloud based platform	Cloud service architecture	Opportunities and challenges of cloud	SOA	Cloud solution implementation	Cyber security	Data integration	Integration middleware	IT management	Mobile agents	Service customer perspective	EAI
1	x	x										
2		x										
3		x										
4			x									
5			x	x				x				
6					x							
7						x						
8	x							x				
9				x			x					
10						x						
11			x									
12			x									
13			x						x			
14										x		
15	x							x				
16			x			x						
17							x					
18	x	x		x	x						x	
19		x	x					x				
20	x	x	x									
21			x									x
22	x	x	x									
23			x	x								
Results	6	7	11	4	2	3	2	4	1	1	1	1

Figure 18: Systematic literature research topic matrix

5.1.1 RISK FOUND

In the table below, the found risks are presented together with its source and explanation. Only risks that are perceived to fit the research scope as it is stated in Chapter 4, are included in the results.

Table 16: Integration risks found in literature

R#	Risk (source)	Explanation
R1	Fragmented infrastructure (Bolloju & Murugesan, 2012; Breiter & Naik, 2013)	Fragmented infrastructure, or device sprawl as it is sometimes called, refers to an IT landscape using different types of infrastructure, devices, and technologies. Due to the ad-hoc nature and heterogeneity of SaaS it is an increased risk to grow an infrastructure landscape that is so diverse that it becomes difficult to manage.
R2	Change in business requirements (Breiter & Naik, 2013)	The lack of ownership of the software makes the SaaS rather inflexible. What if the business requirements change slightly and therefore the integration has to alter? Working with a SaaS means limited flexibility, and therefore it is important to take changes in business requirements into account. Although this risk is a more general risk than just a specific integration risk. It creates integration challenges.
R3	Increased of cyber security risks (Graupner et al.,	When an organizations data crosses its own network boundaries, especially when passing through the internet, increased security

	2011; Neghina & Scarlat, 2013; Redmond, 2012; Yong et al., 2011)	risks are taken. Thus when an integration is made from on-premise to SaaS cyber security becomes a bigger issue.
R4	Semantic and syntactic differences in data models (Goyal, 2010; Pahl & Zhu, 2012)	Due of the tenant structure of a SaaS, users are no longer owning the software. The results for the data model are that there can be semantic and syntactic differences between on-premise and SaaS systems, creating a complex situation for the enterprise wide data semantics, especially when multiple SaaS modules are used, each having different semantics and syntaxes.
R5	Integration and SLAs (Kotlarsky et al., 2012; Lackermair, 2010; Yong et al., 2011)	SLAs are designed contracts to ensure quality of service (QoS) to the service subscribers and also cover integration. In order to enjoy the benefits of such a contract, it is important that the agreed upon service levels are carefully monitored. This monitoring needs to be part of SaaS management in order to achieve an effective SLA.
R6	Changing integration environment (Graupner et al., 2011)	Due to the tenant structure of a SaaS, it cannot be guaranteed that the integration environment remains the same. This has severe impact on process integration due to the fact that many process integration techniques are not suitable when only short term relationships are involved. Using a SaaS in a process chain increases this risks.
R7	Difficulties managing and monitoring cloud operations (Goyal, 2010; Yong et al., 2011)	When systems run on-premise, it is easier to monitor and manage the service than when using a SaaS. The possibilities for monitoring i.e usage is not always available, and creating your own fix, is almost impossible when compared to on-premise to on-premise integration.
R8	Business rules discrepancies (Goyal, 2010)	Just as data models can differ as described in R4, business rules can also differ or interpreted differently between SaaS application and on-premise applications.
R9	New integration technologies knowledge required (Goyal, 2010)	Cloud data integration requires new integration methods, meaning that the risk of a knowledge gap is present and it might be required to look for expertise outside the organization.

5.1.2 MITIGATION STRATEGIES FOUND

We used the same result pool that was used for section 5.1.1 to find mitigation strategies to reduce risks specific to on-premise to SaaS integration. In the table below we present the found strategies and, when possible, mapped them with found integration risks.

Table 17: Integration risk mitigation strategies from literature

MS#	Mitigation Strategy (source)	Explanation
MS1	Standardize data (Kruize et al., 2014)	Because it is often not possible to change the data model of the SaaS applications, it helps to look for common data standard. Change your on-premise data model towards this standard and look for data similarities in further SaaS additions.
MS2	Standardized and scalable architecture (Kotlarsky et al., 2012; C. Liu et al., 2013)	Integration problems can be partially overcome by simply having a good on-premise architecture in place. Therefore, when considering using SaaS systems, make sure your on-premise architecture is on the right track.
MS3	Before acquiring a SaaS, specifically take integration possibilities into account (Graupner et al., 2011; Redmond, 2012)	The service procurement process is important. Being ill informed about different integration possibilities amongst different solution candidates, can increase the integration challenges that are encountered. Using a best practice method (i.e. ITIL supplier management) to find the best

		targets to acquire, can help you avoid problematic integration projects.
MS4	Unified semantic data model and business rules to automate integration (Pahl & Zhu, 2012)	With the aid of unified semantic data models and business rules, mediator services can dynamically perform transformation on top of individual data models of heterogeneous environments.
MS5	Appoint a SLA monitor agent (Kim, Song, & Koo, 2008; Kotlarsky et al., 2012)	Holding suppliers to account against existing service contracts is crucial to the success of a vendor – user organization relationship. Cloud sourcing potential issues tend to unpredictable at the onset of a relationship. Furthermore, when issues arise, the contract will be subject to different interpretations. Contract in cloud sourcing tend to be unique (although might contain the same headers) for every deal and more diverse than other contracts. The contracts also require a faster response time than other types of sourcing contracts and need to deal with the immaturity of contracting in the cloud ecosystem. All these reasons create the need for a dedicated SLA monitor role.
MS6	Use IPaaS/CaaS to reduce the amount of individual interface between on-premise and SaaS systems (Bolloju & Murugesan, 2012; Yong et al., 2011)	IPaaS, CaaS or other similar services can offer help with building consolidated integration solutions, removing the need for inefficient ad-hoc integrations, and creating a more manageable system landscape. It can aid in standardization, atomization and even help with integrating on-premise to on-premise.

5.2 FINDINGS FROM EXPERIENCE EXPERTS

In this chapter, we present the results of the semi-structured interviews we have described in Chapter 2.3. We have found many risks and mitigation strategies and have presented them in a simple fashion containing the risks code, name and a short description.

5.2.1 RISKS FOUND

In Table 18, we present all the found risks that were obtained from the interviews, together with a short description.

Table 18: Risks found during the interviews

IR#	Risks	Explanation
IR1	Increased chances on a fragmented landscape	The risks of getting a “wild growth” of applications and integrations increases when making use of SaaS applications.
IR2	More complex release cycle planning	Adding uncontrolled SaaS release cycles to the set of existing release cycles creates a more complex release plan overall.
IR3	Increased chances of changes in the data model	The chances on uncontrolled data model changes is increased when using SaaS applications.
IR4	Integration cannot scale with the SaaS application	The integration capacity does not scale up or down with SaaS usage.
IR5	Increased chance for the man in the middle	The risks that something is trying to change or read the data that is send over an integration is increased
IR6	Increased chances of changing APIs	The chances on uncontrolled API changes is increased when using SaaS applications.
IR7	Access rules misalignment	Technical limitation or different access management structures of SaaS create a misalignment with organizational access management strategy.
IR8	Easier to deliver erroneous	SaaS makes is easier to start using an application, makes

	products	changes to the application, create connections etc. (especially for non-technical people) without looking at the rest of the landscape.
IR 9	Increased security risks	Integrating with SaaS increases the chance of security issues. This mainly stems from reaching outside the user organizations domain and passing through the internet.
IR 10	Inefficient governance due to gap between business and maintenance	The organizational gap between business and maintenance results in delayed action. This delayed action leads to a lower service quality of the SaaS application.
IR 11	Integration is used to satisfy customer customization needs	The lack of customization possibilities around SaaS can entice the customer to demand customization in the integration. This could lead to unstable integrations.
IR 12	Integration security strategy misalignment	Security options of SaaS do not fit in the organization security strategy.
IR 13	Involving governance too late	Not involving governance in the early processes of SaaS acquisition and implementation could lead to increased problems in time.
IR 14	Less integration support due to tenant relationship	Vendor integration provides limited specific support because the user organization is "one-of-many".
IR 15	Hidden broken interfaces due to wide defined WSDL service contracts	Allowing any type of variables in a WSDL contract creates "hidden" broken interfaces.
IR 16	SaaS integration environment influences your interface design patterns	The user organization needs to adapt to the SaaS which can influence your interface design patterns.
IR 17	Integration breaks due to uncontrolled upgrade	Integration stops working due to an upgrade from the SaaS vendor.
IR 18	Increasing opportunities for hackers	Integrating with SaaS creates more possibilities for people outside the user organization to misuse the integrations.
IR 19	New SaaS vendors have more frequent changes	During the beginning phases of the lifecycle of a SaaS vendor changes happen more often.
IR 20	Ad-hoc change due to security bug	In case of security bug, changes from the vendor side will happen with limited to no time to react.
IR 21	More complex integration management	Integration becomes more complex in a SaaS to on-premise integration.
IR 22	Technical limitations have a negative effect on compliancy	Technical restriction of SaaS can prevent compliancy to certain standards.
IR 23	ITIL is not made for SaaS	ITIL best practices are created in a time were cloud computing did not exist.
IR 24	More complex contract/SLA management	The complexity of contract/SLA management is increased.
IR 25	Less performance due to more security	Security measures have a negative effect on performance.
IR 26	Increased reactive handling of integrations due to lack of ownership	Proactive handling is decreased due to the lower feeling of responsibility over the SaaS application.
IR 27	Limited SLA options	The options for SLA arrangements are limited with a SaaS vendor.
IR 28	Increasing difficulties for auditing	Audit difficulties are increased when a organization makes use of SaaS applications.
IR 29	Image damage due to integration problems, even if you are not responsible	The risks of suffering image damage because of integration problems that are caused by uncontrolled factors is increased.

IR 30	Poor change communication and documentation from the SaaS vendor	Changes that happen without proper communication and documentation from the SaaS vendor.
IR 31	User is not ready for SaaS upgrade	The user is unable or unwilling to upgrade when the SaaS vendor does the upgrade.
IR 32	Logging and auditing dependencies on uncontrolled actors	Logging and auditing is dependent on the SaaS integration. The user organization is not in complete control over the integration.
IR 33	Dependent on the integration options of the SaaS vendor	SaaS vendor decides what integration options are available.
IR 34	Unstable interfaces due to too strictly defined WSDL contracts	Defining WSDL contracts too strict, creates interfaces that break unnecessary often.
IR 35	No control over changes	The user organization has no control over changes to SaaS from the vendor side.
IR 36	More frequent changes in a SaaS integration environment	On-premise to SaaS integration is a more dynamic environment.

Not all of the risk have been mentioned as often as others. To give insight in the number of mentions every risks has received, we created Table 19, mapping risks to the number of interviews in which they have been mentioned.

Table 19: Frequency of risks found during interviews

IR#	Found in % of interviews	IR#	Found in % of interviews
IR2	77,80%	IR12	22,20%
IR8	55,60%	IR36	22,20%
IR35	44,40%	IR3	22,20%
IR1	44,40%	IR22	22,20%
IR17	44,40%	IR5	22,20%
IR14	33,30%	IR13	11,10%
IR6	33,30%	IR21	11,10%
IR19	33,30%	IR26	11,10%
IR7	33,30%	IR10	11,10%
IR9	33,30%	IR16	11,10%
IR4	33,30%	IR28	11,10%
IR25	22,20%	IR29	11,10%
IR11	22,20%	IR30	11,10%
IR25	22,20%	IR31	11,10%
IR32	22,20%	IR15	11,10%
IR27	22,20%	IR33	11,10%
IR23	22,20%	IR34	11,10%
IR18	22,20%	IR20	11,10%

This list of risks frequency cannot be considered as a generalizable list that shows what risks have the most impact or occur more often than others. Still we argue that showing the frequency of mentioning shows that some risks are more present in the eyes of our experts than others. This becomes increasingly interesting because we have a diverse sample of interviewees. We elaborate on this in

Chapter 2.3. Especially IR2, IR8, IR35, IR1, and IR17 deserve a special mentioning as risks that are mentioned in a large portion of the interviews.

5.2.2 MITIGATION STRATEGIES FOUND

In Table 20, we present all the found mitigation strategies that were obtained from the interviews, together with a short description.

Table 20: Mitigations strategies found during the interviews

#	Mitigation strategy	Description
IM1	Use a canonical model	Use a service bus with canonical model in your middleware layer.
IM2	SaaS ready on-premise architecture	Have a stable and flexible architecture on premise that allows easy integration and disintegration. I.e. SOA architecture.
IM3	Create a specific governance strategy for SaaS integration	Specifically create a strategy for SaaS integration, apart your general integration governance strategy, in order to cope with the specific SaaS integration risks.
IM4	Use specialized SaaS security services	There is security software available special made to cope with SaaS dynamics. Use these specialized tools to get a secure solution.
IM5	Create a specific SLA management strategy for SaaS integration	Create a strategy on managing SLAs for SaaS integrations, apart from the general SLA management strategy, in order to cope with specific SaaS integration related SLA challenges.
IM6	Actively check compliancy with standards and audits of SaaS vendors	Go to your SaaS vendor and check if they are complying to standards and audits, or ask for independent, up-to-date certificates stating that the vendor is complying to standards and audits.
IM7	Create a dedicated role for overseeing SaaS integrations	Use a dedicated integration specialist role that is involved from the very beginning with SaaS integration. The dedication is necessary due to need for proactive management, i.e. monitoring and anticipating future changes.
IM8	Use governance when technical possibilities of SaaS integration are limited	When technical possibilities of the SaaS application are limited, i.e. around access management, use extra governance to overcome risks.
IM9	Encrypt messages and use secure channels	Use both data encryption and secure channels when sending data outside the organizations domain.
IM10	Use service registries	Use service registries to have a better and faster understanding of the impact of services and changes to the services.
IM11	Classify business objects towards SaaS availability beforehand	Business objects and their data should be classified beforehand on whether or not the organization is ok with potential storage or usage in the cloud.
IM12	Create a release cycle strategy aimed to cope with SaaS dynamics	The release cycle strategy should incorporate the dependencies on uncontrolled release plans of SaaS, and aim the on-premise (controlled) release cycles with these dependencies in mind.
IM13	Set up service contracts in a smart way	Service contracts (WSDL) need to be setup in a way that creates stable interfaces that do not turn into hidden broken interfaces.
IM14	Make use of more senior requirement management	An increased effort should be made considering requirement management, this calls for more influential management.

IM15	Make use of dev-ops teams	Use teams that remove or reduce the gap between development and operations.
IM16	Create a back-up strategy for broken interfaces	If an interface breaks, you should have a back-up strategy (i.e. raw data interface) to allow the organization reach its data.
IM17	Use a governance champion	Governance needs a strong mandate from higher management to create the means to act quicker.
IM18	Use a proactive SaaS procurement strategy	Be proactive around potential challenges when procuring a SaaS applications.
IM19	Use a more senior release; manager	To manage the release plan of a landscape including SaaS integrations, it is smart to have an influencing release plan manager.
IM20	Use available security options	Every SaaS has its own security options that, when used, provide the safest SaaS integration solution.
IM21	Put your middleware in the cloud	Make use of cloud based middleware between SaaS and on-premise applications.
IM22	Use stricter governance on changes	Governance needs to be more strict on changes in the integration environment.
IM23	Give special clearance for hotfixes	Hot fixing requires a need to be able to work quickly and thus having a strong mandate.
IM24	Use more proactive integration management	Use proactive integration management to prevent integration problems, i.e. by actively looking for potential future challenges or changes.
IM25	Test the SaaS application before using it	Test the SaaS, especially on non-functional requirements, before investing in an integration.
IM26	Specially focus on non-functional requirements	Specially focus on non-functional requirements as they are potential deal breakers with SaaS solutions.
IM27	Use a widely available and known role that is responsible for information on SaaS	Make sure the organization has a place where up-to-date information on SaaS applications, organization requirements and procedures can be found.

Again, as before, we have mapped the found mitigation strategies to the number of interviews in which they have been mentioned. This mapping can be found in the table below.

Table 21: Frequency of mitigation strategies found during interviews

IM#	Found in % of interviews	IM#	Found in % of interviews
IM1	88,90%	IM15	11,10%
IM3	66,70%	IM17	11,10%
IM2	44,40%	IM19	11,10%
IM12	44,40%	IM23	11,10%
IM18	33,30%	IM9	11,10%
IM7	33,30%	IM8	11,10%
IM19	22,20%	IM22	11,10%
IM10	22,20%	IM6	11,10%
IM24	22,20%	IM5	11,10%
IM20	22,20%	IM4	11,10%
IM11	11,10%	IM25	11,10%
IM16	11,10%	IM26	11,10%
IM13	11,10%	IM27	11,10%
IM21	11,10%		

The same arguments made considering the frequency of risks in Chapter 5.2.1 can be made for the mitigation risks found during the interviews. IM1, IM3, IM2, IM12 are mitigation strategies that are named exceptionally often and deserve a special mentioning.

5.3 COMBINATION OF VIEWS

In order to provide a holistic view of all the found risks and mitigation strategies, we use this section to combine the findings from our systematic literature research and the expert interviews and conclude on our initial findings. We need to make notion of a few risks found in literature that have been contradicted during the interviews with experts. This can be due to multiple reasons, i.e. there is a gap between practice and the scientific world or interpretation mistakes made by the researcher. We do not know what caused these contradictions exactly, but we find them interesting and argue that risks and mitigation strategies that have been contradicting of nature, at least have to be mentioned. In some cases these contradictions were so strong that we have decided to remove risks or mitigation strategies from the final list altogether.

First, we have found that R1, R3, R5, R6, and R7 have been mentioned during the interviews and thus can be seen as solid finds. This leaves R2, R4, R8, and R9 as the 'special' cases. When we checked for deviant cases during the interviews, the mentioning of R2, the risks of changing business requirements in combination with 'not owned' SaaS software, received the feedback of not being on-premise to SaaS specific. The following simple answer displayed the naivety of the found risk.

"Yes, but you also have to account for these changes when you work on-premise."

When the researcher tried to explain the R2 further by elaborating on the difference between a SaaS and a self owned application with respect to making changes to it the following answer was received, making it clear that R2 was not to be considered a specific on-premise to SaaS risk, nor did its impact increased in that specific scenario.

"(...) you can buy an on-premise application which does not allow you to customize it. On the other hand you can procure SaaS that is partially customizable. You also see on-premise applications that can be customized and SaaS solutions that are not customizable. It is a choice you deliberately make. (...) I do not see this as a risk but merely a choice you make in your business case. (...) you make a deliberate choice. If you choose for an on-premise or SaaS, you can still choose in both forms if you want to be able to customize it or not"

R4, the risk of having differences between data models, has received a similar faith. Although this risk is present during on-premise to SaaS integration, the argument for it not being specific was again simple.

"It is a standard integration problem, which you always have to take care of."

The argument of SaaS solutions having a bigger difference in data model than on-premise solutions is considered to be a long shot, especially as SaaS applications often adopt best practices.

R8, the discrepancies in business rules, gets the same treatment as the differences in data models, although it fueled more elaborate discussion, as business rules discrepancies is not so easily covered as data model differences. Following the arguments made in the quotation above, this is again a choice made during the procurement phase.

The last 'special' case is R9, stating that cloud technology requires different integration methods and thus a different set of expertise. When talking with the integration experts, it quickly became clear that the technological challenge of on-premise to SaaS integration can be considered non-existent.

"In principle, I do not see any difference worth mentioning, technology wise, in the entire SaaS story. Integration remains integration. The challenge mainly lies in governance"

Especially people working with SOA will not notice any differences as they already operate using technologies such as XML, web services, and APIs. If a SaaS solution exists that requires completely different integration technology and expertise, it would again be considered a procurement choice, not something that can be considered an on-premise to SaaS integration risks. Due to the reasons presented above, the 'special' cases have been removed from our final list of risks.

On the mitigation side, we have MS2, MS3, MS5, MS6 that have been mentioned during the interviews and thus can be considered as solid finds as their risks counterparts. MS1 and MS4 are considered viable mitigation strategies, although they have not been mentioned during the interviews. These two are added to the final list of mitigation strategies as IM28 and IM29. This results in 36 risks and 29 mitigation strategies. The full list of risks and mitigation strategies can be found in Appendix B.

6 INTEGRATION FRAMEWORK

The risks found during the systematic literature research and the interviews can be considered a subset of an inexhaustible set of risks. This set is inexhaustible because of three reasons. First, we have interviewed a group of various experts that operate on different levels. Secondly, we allowed them to come up with different projects scenarios from their own experience in order to identify the risks. Finally we accepted all risks that fitted into our scoping framework, ranging from very technical towards more business oriented layers. These three reasons result in a set of risks that vary in abstraction level and perspective. Therefore we cannot claim we have found all of the on-premise to SaaS integration risks as it would be possible to rephrase, abstract upon, approach a risk from a different perspective, or have different risks because of different projects and experiences all together. We do however argue that we have found enough risks to uncover overarching risk themes. The actual gathering of these themes has been done with a clustering assignment during the workshop as described in Chapter 0.

In order to present our risk themes we have used a tool to create area-proportional Euler diagrams, we briefly elaborate on this tool in Chapter 6.1. Due to the set-up of the workshop we have derived on two different perspectives, and thus two different sets of themes were obtained from the same risk set. In Chapter 6.2 and 6.3 we elaborate on both of the perspectives before we dive into the individual risk themes. The perspectives are compared in order to understand their relationship and we present mitigation strategies that can reduce the risk themes. We have abstained from generating one area-proportional Euler image in which all themes of both perspectives are placed, due to limitations of a two dimensional Euler diagram. In Chapter 6.5 we analyze the mitigation strategies and in Chapter 6.6 we present the categorization of risks.

In this chapter we have made use of allot of the early presented risks codes. Therefore certain part can be though to read. We advice using the complete overview of risks and mitigation strategies presented in Appendix B as a reference guide.

6.1 VENNMASTER

With an area-proportional Euler diagram we can show how big a risk theme is compared to other risks themes in terms of the amount of risks it represents. We can also use Euler diagrams to show how the risk themes intersect. These two reasons provide a compelling case to use area-proportional Euler diagrams as a way of visually represent the risk themes we found during the workshop.

Drawing Euler diagrams is relatively simple when we have small amount of sets, and when the sets have limited amount of intersections. In our case we had a total of 12 sets, which had vast amounts of intersections. After numerous attempts the computing power of a human brain was considered unable to create Euler diagrams that were considered correct, sometimes it was so complex we even had difficulties verifying if a particular Euler diagram was incorrect or not. These difficulties led to the adoption of the VennMaster application.

The VennMaster application is the result of a science project that tries to visualize Microarray experiments during Gene Ontology analysis (GO). Instead of long tables of genes it provides an alternative and visual representation of relationships and semi-quantitative size information to support biological hypothesis formulation (Kestler et al., 2008). We use the application to generate Euler diagrams of our risk themes. This required the creation of a unique database for our risks and risk themes as we of course were dealing with on-premise to SaaS integration risks instead of a GO database. Besides the database it also required lots of trails with different configurations as the VennMaster application uses a complex algorithm to calculate how to place the clusters.

6.2 USER IT ORGANISATION PERSPECTIVE

The first perspective is named the 'User IT organization perspective'. It is considered to be the more traditional perspective of the organization that uses the SaaS solution in combination with their on-premise systems. This perspective focuses on the increased complexity, uncontrolled changes, technical integration, compliancy, security and integration support. It consists of 7 different themes and overview of the results can be found in Table 22.

Table 22: Risk themes from the user organization perspective

Theme	Description	Risk#	Percentage of total risks
Complexity	The complexity theme encompasses risks that lead to a more complex situation when an application landscape consist of a combination of on-premise systems and SaaS solutions.	IR1, IR8, IR11	8.33%
Integration	Risks that are placed into the integration theme are considered to be risks that are mostly technical of nature, dealing with integration options, integration management and the technical integration.	IR4, IR15, IR16, IR21, IR29 , IR33 , IR34	19.44%
Compliancy	The compliancy risk theme gathers all risks that are considered challenges related to auditing and compliancy with standards.	IR12, IR22, IR28, IR32	11.11%
Security	Security risks represent the challenges that security creates for an integration between on-premise systems and a SaaS.	IR5, IR7, IR9, IR12, IR18, IR25	16.67%
Release planning	The release planning theme encompasses risks that revolve around uncontrolled changes in the integration environment and managing release cycles.	IR2, IR3, IR6, IR17, IR20, IR36	16.67%
Change requests	Change requests has a big overlap with release planning but shows a specific focus towards changes that are more ad-hoc instead of planned upgrades.	IR3, IR6, IR10, IR19, IR31, IR35, IR36	19.44%
Support	Risks placed in the support theme are risks that revolve around integration support such as SLA management or change documentation from the SaaS vendor.	IR13, IR14, IR23, IR24, IR26, IR27, IR30	19.44%
Total (7)		All risks used at least once (IR3, IR6, IR12, IR36 used twice)	111.11%

The percentage of the total risk used in Table 22 is higher than 100%, this is because four risks have been placed into multiple clusters. We have created a visual representation of the perspective, which can be found in Figure 19.

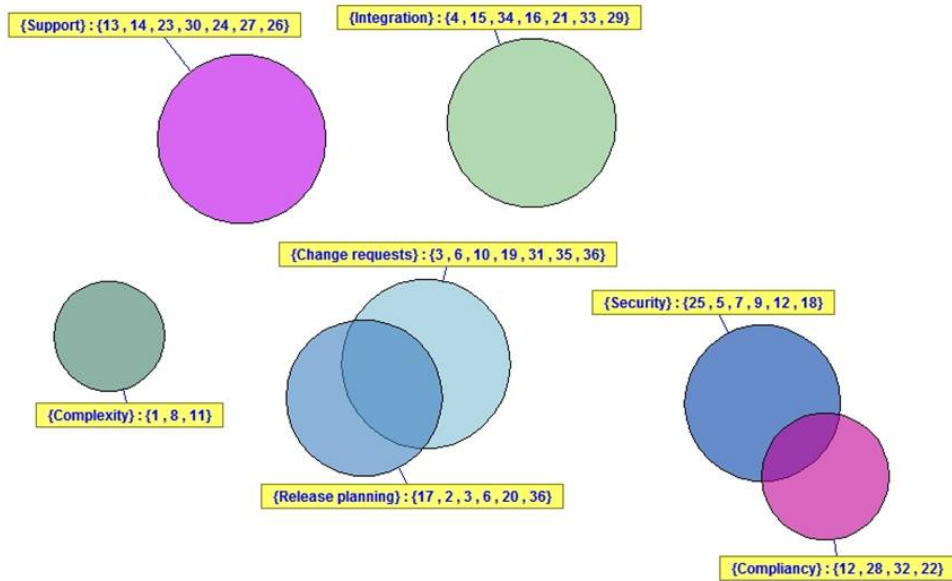


Figure 19: Visual representation of the risks clustering of the user IT organization perspective

6.3 INTEGRATION COST PERSPECTIVE

The second perspective obtained has been named the ‘integration costs perspective’; this is done because it is more cost oriented. According to one of the workshop attendees if you abstract upon all the risks of integration the one risk that remains is an increase of costs of the integration. The perspective consists of five themes that together represent risks that increase the costs for development, maintenance, data confidentiality and integrity, user adoption and system performance. An overview of how the risks have been clustered can be found in Table 23.

Table 23: Risk themes from integration cost perspective

Theme	Description	Risk#	% of total risks
System performance	Risks that are clustered under system performance are risks that negatively influence performance. Most of the risks result in broken interfaces.	IR4, IR15, IR17, IR25, IR34	13.89%
Data confidentiality & integrity	The data confidentiality theme is almost the same as the security theme and encompasses all risks that are related to this topic.	IR5, IR7, IR9, IR12, IR18	13.89%
Maintainability	Maintainability risks drives up the costs for maintaining the resulting integration. This encompasses overall architecture, integration support, auditing problems and integration management risks.	IR1, IR2, IR3, IR6, IR10, IR13, IR14, IR17, IR20, IR23, IR28, IR30	33.33%
Development costs	The high development cost theme combines all risks that result in development costs and has some overlap with maintainability but focuses on the development	IR1, IR8, IR11, IR12, IR13, IR16, IR19, IR21, IR24, IR27, IR30, IR32, IR33, IR34	38.89%

	phase of the integration.		
User adoption	Risks that are allocated to the user adoption theme are of a wide variety of risks, ranging from changes to the integration towards auditing problems. The risks create difficulties for users to adopt a SaaS solution. This leads to an increase in costs to achieve a desired user adoption level.	IR2, IR3, IR6, IR7, IR11, IR14, IR15, IR22, IR26, IR29, IR31, IR32, IR35, IR36	38.89%
Total (5)		All risks used at least once (IR1, IR2, IR3, IR6, IR7, IR11, IR12, IR13, IR14, IR15, IR17, IR30, IR32, IR34 used twice)	138.89%

The percentage of the total risk used in Table 23 is higher than 100%, this is because 14 risks have been placed into multiple clusters. We have created a visual representation of the data that can be found in Figure 20.

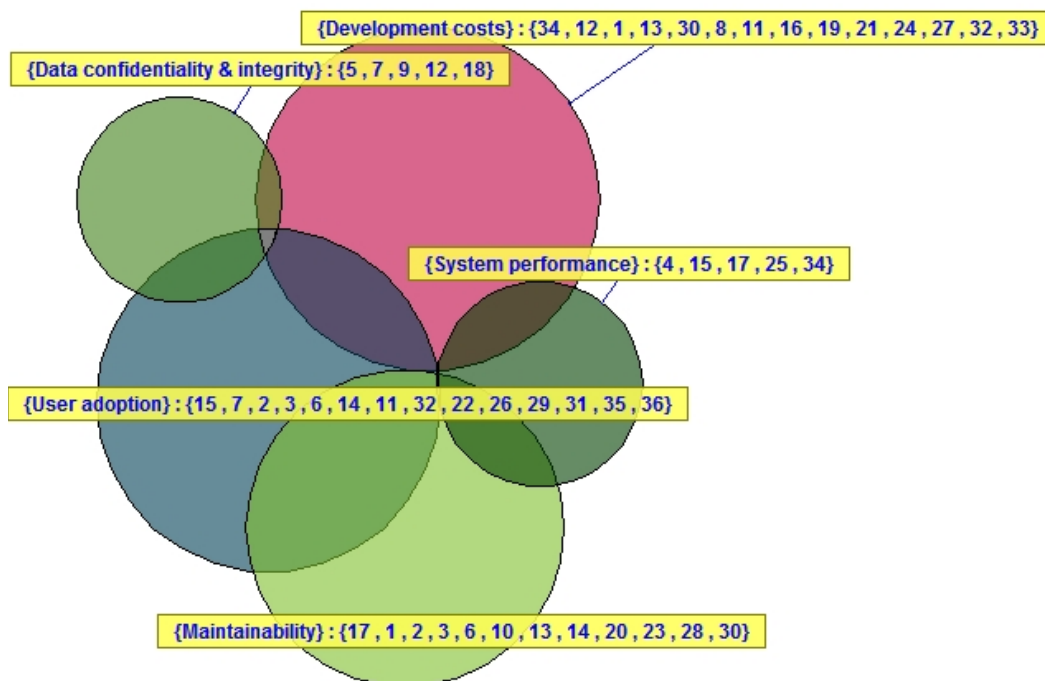


Figure 20: Visual representation of the risks clustering of the integration cost perspective

6.4 THEMES AND MITIGATION STRATEGIES

In order to gain a greater understanding of the found perspectives and themes we use this section to elaborate on the individual themes. We compare the individual themes with themes from the other perspective to gain insight into both perspectives and how the two perspectives co-exist in the found risk set. We also elaborate on the linkage between risk themes and mitigation strategies.

6.4.1 COMPLEXITY

The complexity theme encompasses risk that lead to a more complex integration situation. When we compare both perspectives from viewpoint of the complexity theme we find that complexity shares

risks with both development costs and maintainability. The overview of what risks are shared among the themes is presented in Table 24.

Table 24: Perspective comparison from the complexity viewpoint

Overlap with theme	Percentage of overlap	Risk#
High development costs	100.00%	IR1, IR8, IR11
Maintainability	33.33%	IR1

In Figure 21 we have presented a visual representation of the comparison.

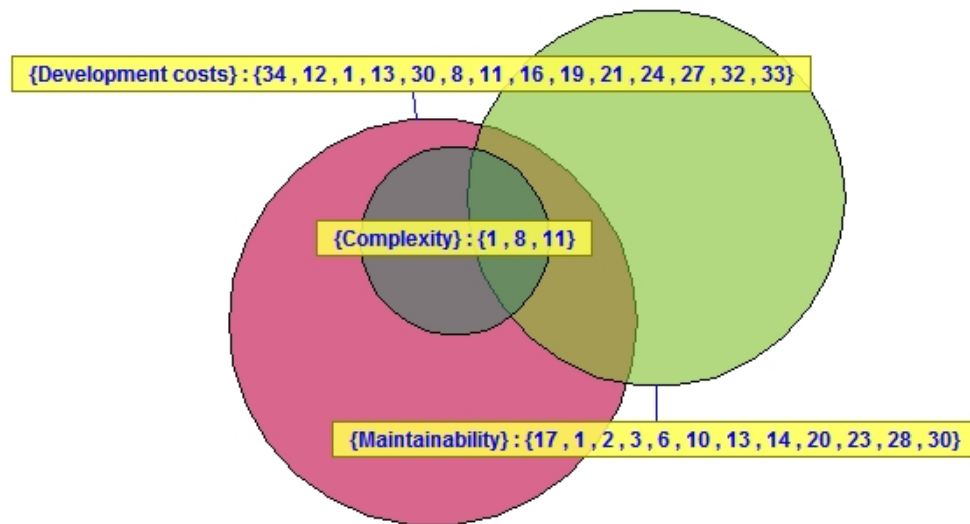


Figure 21: Perspective comparison from the complexity viewpoint

From this figure we can conclude that due to the complexity of integrations between on-premise systems and SaaS solutions both the development costs and the maintainability costs increase.

According to experts we can reduce the risk on increased complexity by applying IM1, IM2, IM7, IM10, IM13, IM27, IM28, and IM29.

6.4.2 INTEGRATION

The integration theme encompasses risks that come forth out of the technical integration aspects, integration options and management. When we compare both perspectives from the integration risk theme we find that the theme shares risks with development costs, user adoption and system performance themes. The overview of what risks are shared among the themes is presented in Table 25.

Table 25: Perspective comparison from the integration viewpoint

Overlap with theme	Percentage of overlap	Risk#
System performance	42.86%	IR4, IR15, IR34
High development costs	57.14%	IR16, IR21, IR33, IR34
User adoption (not meeting the customer needs)	28.57%	IR15, IR29

In Figure 22 we have presented a visual representation of the comparison.

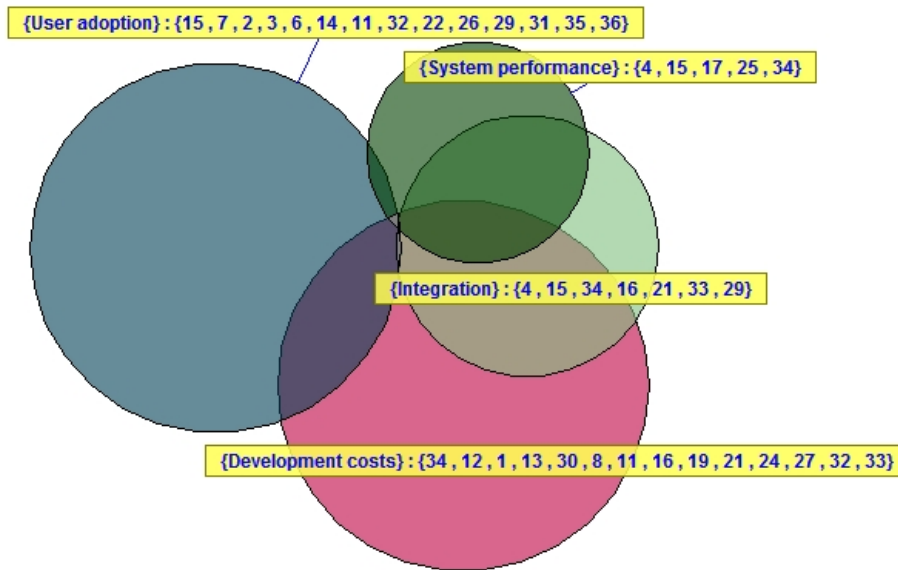


Figure 22: Perspective comparison from the integration viewpoint

We can conclude that integration risks affect system performance, increase the development costs and increase difficulties for user adoption.

According to experts we can reduce the integration risks by applying IM1, IM2, IM7, IM10, IM13, IM16, IM24, IM28, and IM29.

6.4.3 COMPLIANCY

The compliancy theme encompasses risks that come forth out compliancy with standards and auditing difficulties. When we compare both perspectives from the compliancy theme we find that the theme shares risks with development costs, user adoption, data confidentiality & integrity and maintainability themes. The overview of those risks that are shared among the themes is presented in Table 26.

Table 26: Perspective comparison from the compliancy viewpoint

Overlap with theme	Percentage of overlap	Risk#
Data confidentiality & integrity	25.00%	IR12
User adoption (not meeting the customer needs)	50.00%	IR22, IR32
High development costs	50.00%	IR12, IR32
Maintainability	25.00%	IR28

In Figure 23 we have presented a visual representation of the comparison.

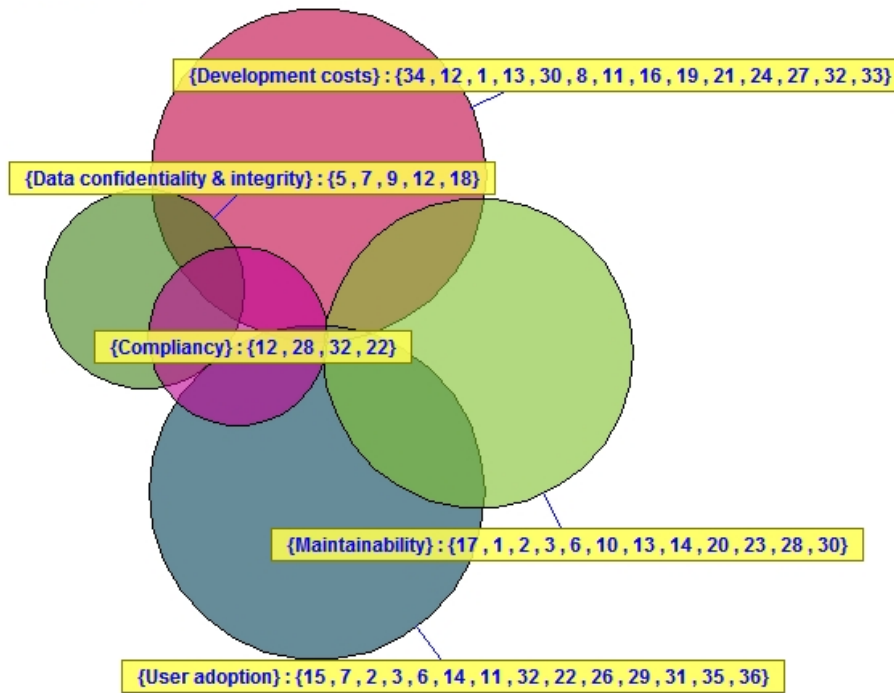


Figure 23: Perspective comparison from the compliancy viewpoint

From this figure we can conclude that compliancy risks increase the costs for development, maintainability, user adoption and data confidentiality & integrity.

According to experts mitigation strategies that reduce the compliancy risks are IM3, IM6, IM8, IM17, IM26, IM11, and IM25.

6.4.4 SECURITY

The security theme encompasses risks that come forth out security challenges. When we compare both perspectives from the security theme we find that the theme shares risks with development costs, data confidentiality & integrity, user adoption and system performance themes. The overview of what risks are shared among the themes is presented in Table 27.

Table 27: Perspective comparison from the security viewpoint

Overlap with theme	Percentage of overlap	Risk#
Data confidentiality & integrity	83.33%	IR5, IR7, IR9, IR12, IR18
User adoption (not meeting the customer needs)	16.67%	IR7
System performance	16.67%	IR25
High development costs	16.67%	IR12

In Figure 24 we have presented a visual representation of the comparison.

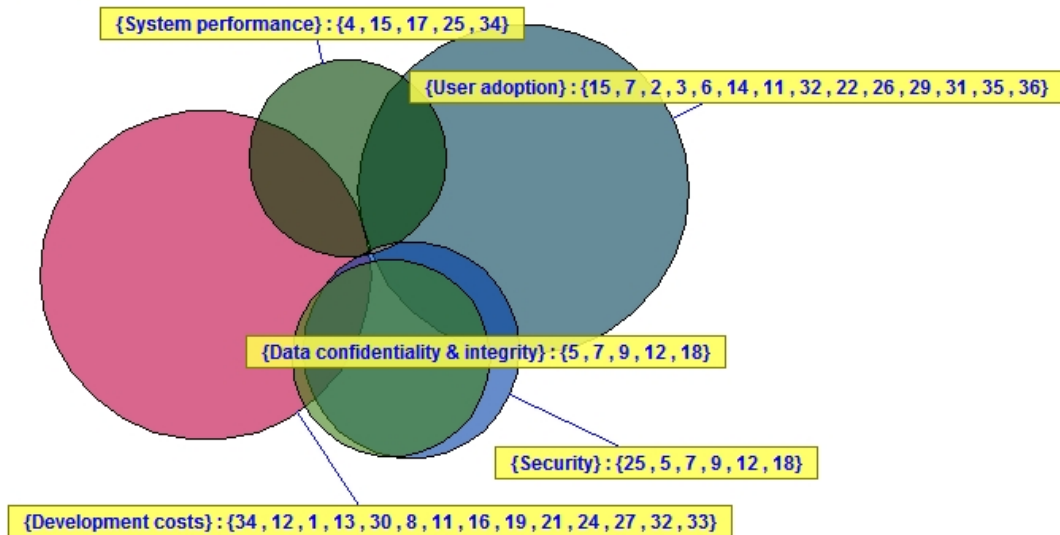


Figure 24: Perspective comparison from the security viewpoint

From this we can conclude that security risks lead to more cost concerning development, data confidentiality & integrity, maintenance and system performance.

According to experts mitigation strategies that reduce the security risks are IM4, IM8, IM9, IM11, IM20, and IM25.

6.4.5 RELEASE PLANNING

The release planning theme encompasses risks that come forth out challenges surrounding the release planning. When we compare both perspectives from the release planning theme we find that the theme shares risks with maintainability, system performance and user adoption. The overview of what risks are shared among the themes is presented in Table 28.

Table 28: Perspective comparison from the release planning viewpoint

Overlap with theme	Percentage of overlap	Risk#
Maintainability	83.33%	IR2, IR3, IR6, IR17, IR20
System performance	16.67%	IR17
User adoption (not meeting the customer needs)	66.67%	IR2, IR3, IR6, IR36

In Figure 25 we have presented a visual representation of the comparison.

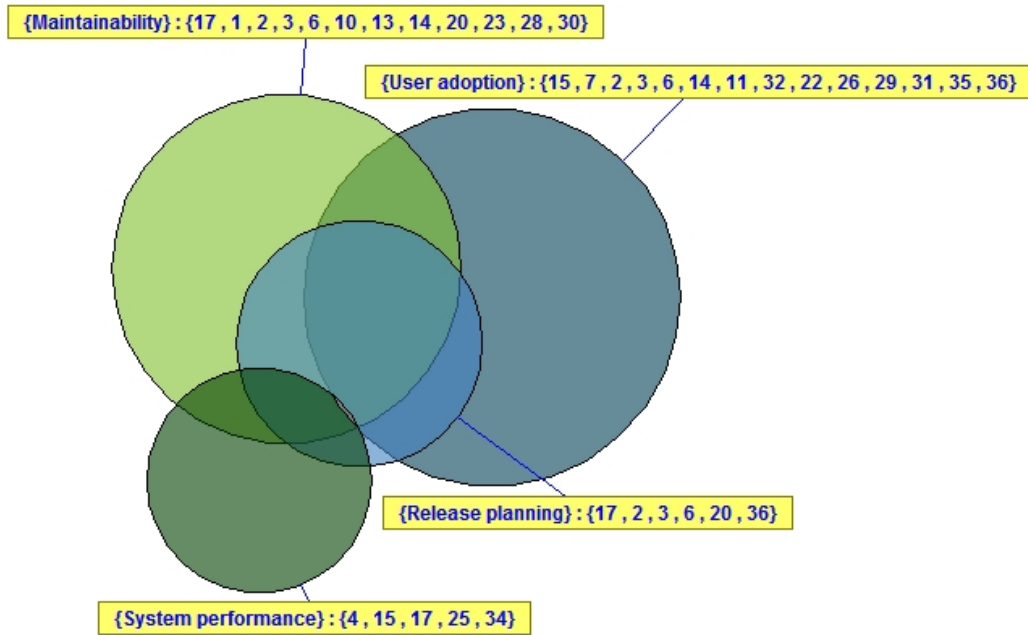


Figure 25: Perspective comparison from the release planning viewpoint

We can conclude that a more dynamic and less controlled release planning results in increased costs of maintainability, user adoption, and system performance.

According to experts mitigation strategies that reduce the release planning risks are IM2, IM12, IM14, IM15, IM19, and IM23.

6.4.6 CHANGE REQUESTS

The change request theme encompasses risks that come forth out challenges uncontrolled changes outside the release planning. When we compare both perspectives from the change request theme we find that the theme shares risks with maintainability, development costs and user adoption themes. The overview of what risks are shared among the themes is presented in Table 29.

Table 29: Perspective comparison from the change requests viewpoint

Overlap with theme	Percentage of overlap	Risk#
Maintainability	42.86%	IR3, IR6, IR10
High development costs	14.29%	IR19
User adoption (not meeting the customer needs)	71.43%	IR3, IR6, IR31, IR35, IR36

In Figure 26 we have presented a visual representation of the comparison.

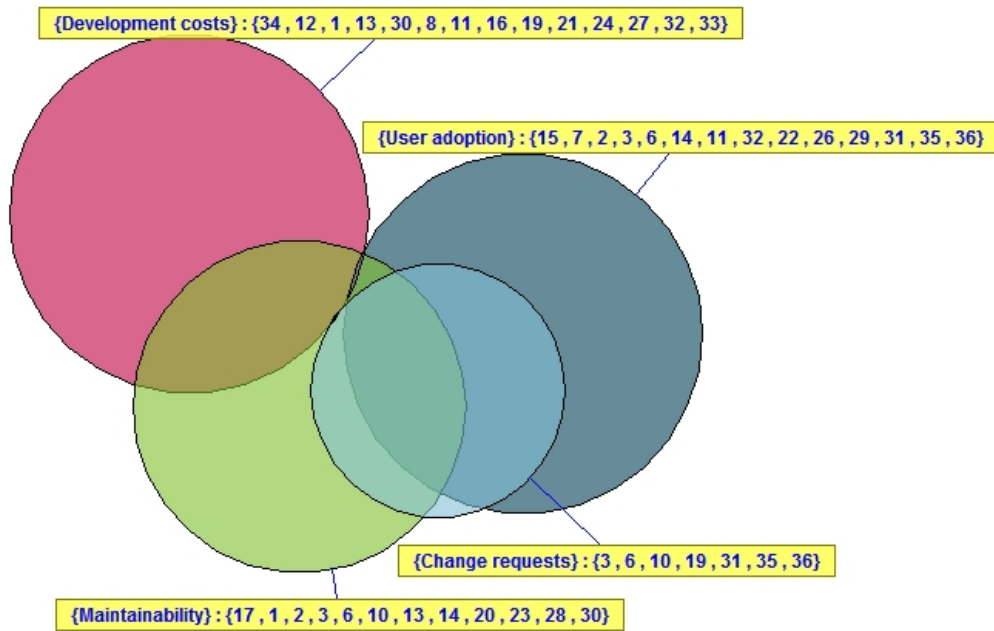


Figure 26: Perspective comparison from the change requests viewpoint

We can conclude that uncontrolled changes outside the release planning result in increased costs of maintainability, user adoption, and development.

According to experts mitigation strategies that reduce the change request risks are IM14, IM15, and IM22.

6.4.7 SUPPORT

The support theme encompasses risks that come forth out challenges surrounding integration support. When we compare both perspectives from the support theme we find that the theme shares risks with maintainability, development costs and user adoption themes. The overview of what risks are shared among the themes is presented in Table 30.

Table 30: Perspective comparison from the support viewpoint

Overlap with theme	Percentage of overlap	Risk#
Maintainability	57.14%	IR13, IR14, IR23, IR30
High development costs	57.14%	IR13, IR24, IR27, IR30
User adoption (not meeting the customer needs)	28.57%	IR14, IR26

In Figure 27 we have presented a visual representation of the comparison.

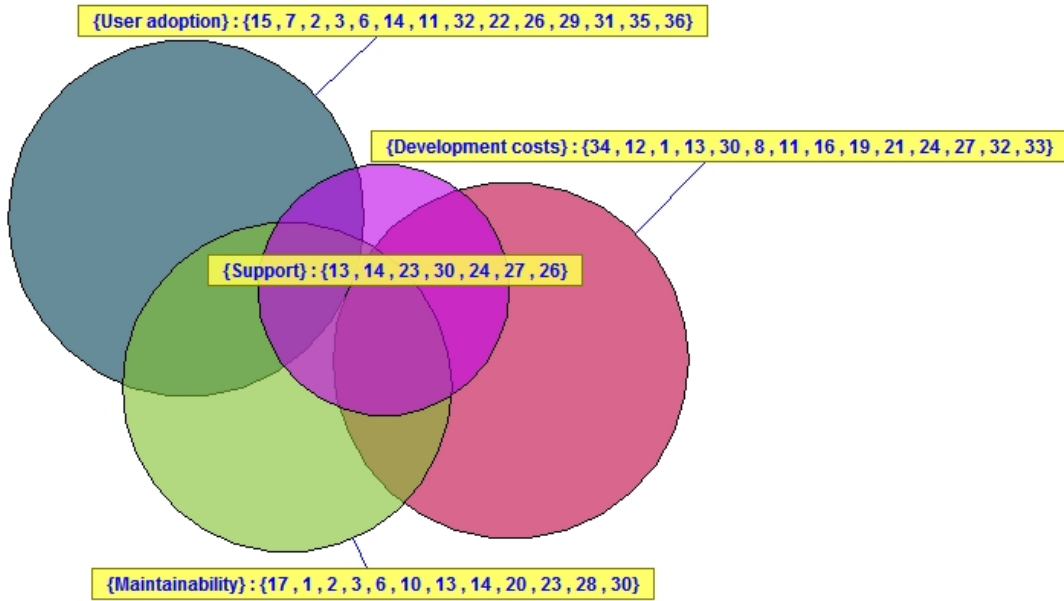


Figure 27: Perspective comparison from the support viewpoint

We can conclude that on-premise to SaaS integration support risks result in increased costs of maintainability, user adoption, and development.

According to experts mitigation strategies that reduce the support risks are IM3, IM5, IM15, IM16, IM23, and IM25.

6.4.8 SYSTEM PERFORMANCE

The system performance theme encompasses risks that increase the costs of achieving the wanted system performance. When we compare both perspectives from the system performance theme we find that the theme shares risks with integration, security and release planning themes. The overview of what risks are shared among the themes is presented in Table 31.

Table 31: Comparison Perspective comparison from the system performance viewpoint

Overlap with theme	Percentage of overlap	Risk#
Integration	60.00%	IR4, IR15, IR34
Security	20.00%	IR25
Release planning	20.00%	IR17

In Figure 28 we have presented a visual representation of the comparison.

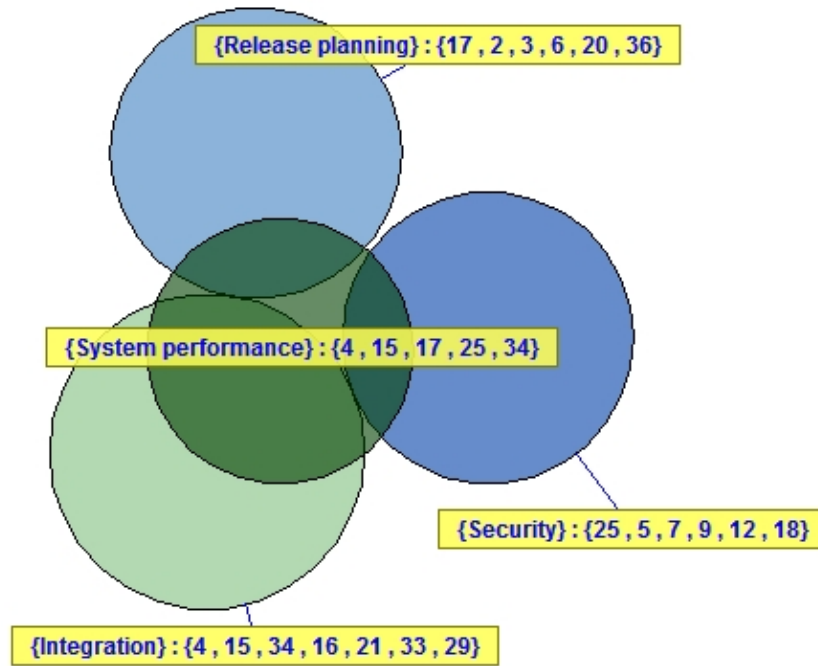


Figure 28: Perspective comparison from the system performance viewpoint

We can conclude that to achieve the desired system performance cost go up because of risks related to security, release planning and integration.

According to experts mitigation strategies that reduce the risk on a higher system performance costs are IM16 and IM21.

6.4.9 DATA CONFIDENTIALITY & INTEGRITY

The data confidentiality & integrity theme encompasses risks that increase the costs of achieving the wanted security level. When we compare both perspectives from the data confidentiality & integrity theme we find that the theme shares risks with compliancy and security themes. The overview what risks are shared among the themes is presented in Table 32.

Table 32: Perspective comparison from the data confidentiality & integrity viewpoint

Overlap with theme	Percentage of overlap	Risk#
Compliancy	20.00%	IR12
Security	100.00%	IR5, IR7, IR9, IR12, IR18

In Figure 29 we have presented a visual representation of the comparison.

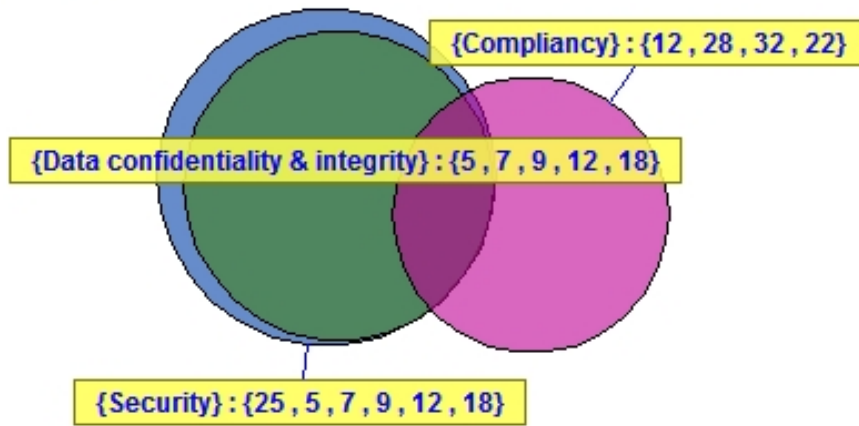


Figure 29: Perspective comparison from the data confidentiality & integrity viewpoint

We can conclude that to achieve the desired level of data integrity & confidentiality cost go up because of risks related to security and compliancy.

According to experts mitigation strategies that reduce the risk on higher data integrity & confidentiality costs are IM4, IM9 and IM20.

6.4.10 MAINTAINABILITY

The maintainability theme encompasses risks that increase the costs of maintaining on-premise to SaaS integrations. When we compare both perspectives from the maintainability theme we find that the theme shares risks with compliancy, complexity, release planning, change requests, and support themes. The overview of what risks are shared among the themes is presented in Table 33.

Table 33: Perspective comparison from the maintainability viewpoint

Overlap with theme	Percentage of overlap	Risk#
Complexity	8.33%	IR1
Compliancy	8.33%	IR28
Release planning	41.67%	IR2, IR3, IR6, IR17, IR20
Change requests	25.00%	IR3, IR6, IR10
Support	33.33%	IR13, IR14, IR23, IR30

In Figure 30 we have presented a visual representation of the comparison.

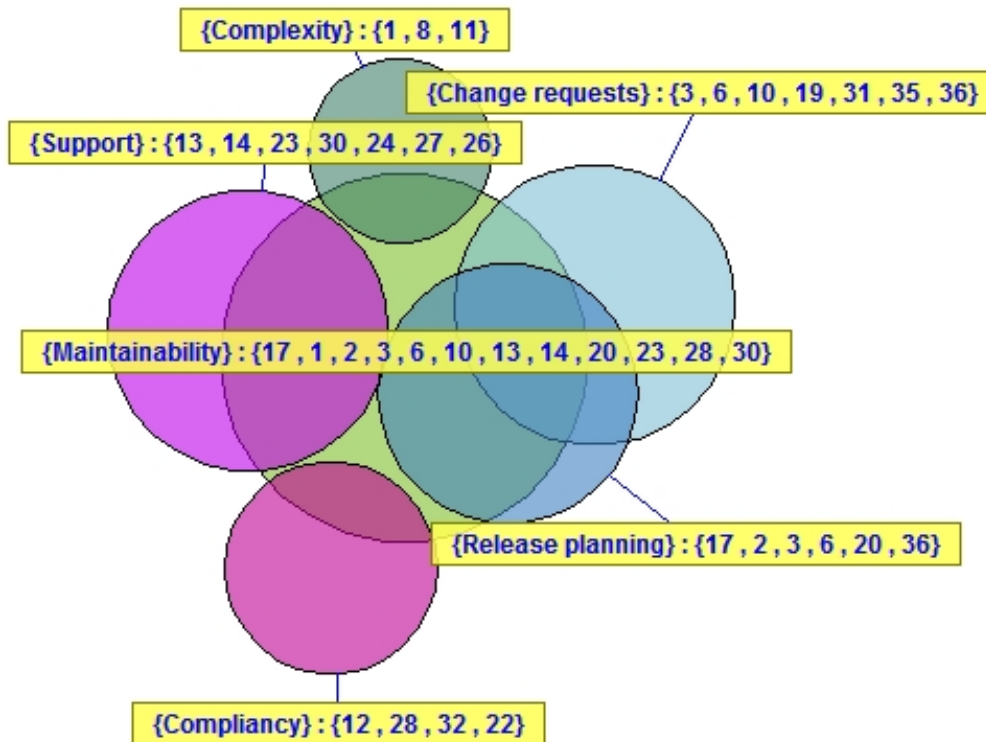


Figure 30: Perspective comparison from the maintainability viewpoint

We can conclude that to maintainability costs go up because of risks related to integration support, complexity, release planning, change request, and compliancy.

According to experts mitigation strategies that reduce the risk on higher data integrity & confidentiality costs are IM3, IM5, IM6, IM7, IM8, IM10, IM15, IM19, IM23, IM25, and IM26.

6.4.11 DEVELOPMENT COSTS

The development costs theme encompasses risks that increase the costs developing the desired solution. When we compare both perspectives from the development costs theme we find that the theme shares risks with compliancy, complexity, integration, change request, support and security themes. The overview of what risks are shared among the themes is presented in Table 34.

Table 34: Perspective comparison from the development costs viewpoint

Overlap with theme	Percentage of overlap	Risk#
Complexity	21.43%	IR1, IR8, IR11
Compliance	14.29%	IR12, IR32
Integration	28.57%	IR16, IR21, IR33, IR34
Change requests	7.14%	IR19
Support	28.57%	IR13, IR24, IR27, IR30
Security	7.14%	IR12

In Figure 31 we have presented a visual representation of the comparison.

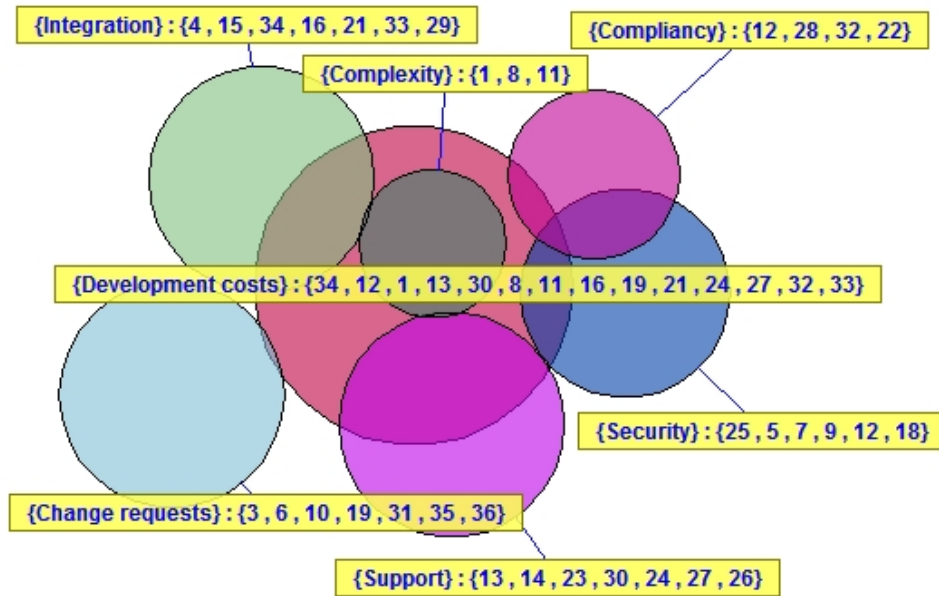


Figure 31: Perspective comparison from the development costs viewpoint

We can conclude that to development costs go up because of risks related to integration support, complexity, change request, integration, security and compliancy.

According to experts mitigation strategies that reduce the risk on higher data integrity & confidentiality costs are IM1, IM2, IM6, IM7, IM8, IM13, IM20, IM28, and IM29.

6.4.12 USER ADOPTION

The user adoption theme encompasses risks that increase the costs for achieving user adoption. When we compare both perspectives from the user adoption theme we find that the theme shares risks with compliancy, complexity, integration, change request, release planning, support and security themes. The overview of what risks are shared among the themes is presented in Table 35.

Table 35: Perspective comparison from the user adoption viewpoint

Overlap with theme	Percentage of overlap	Risk#
Complexity	7.14%	IR11
Compliancy	14.29%	IR22, IR32
Integration	14.29%	IR14, IR16
Release planning	28.57%	IR2, IR3, IR6, IR36
Change requests	35.71%	IR3, IR6, IR31, IR35, IR36
Support	7.14%	IR14
Security	7.14%	IR7

In Figure 31 we have presented a visual representation of the comparison.

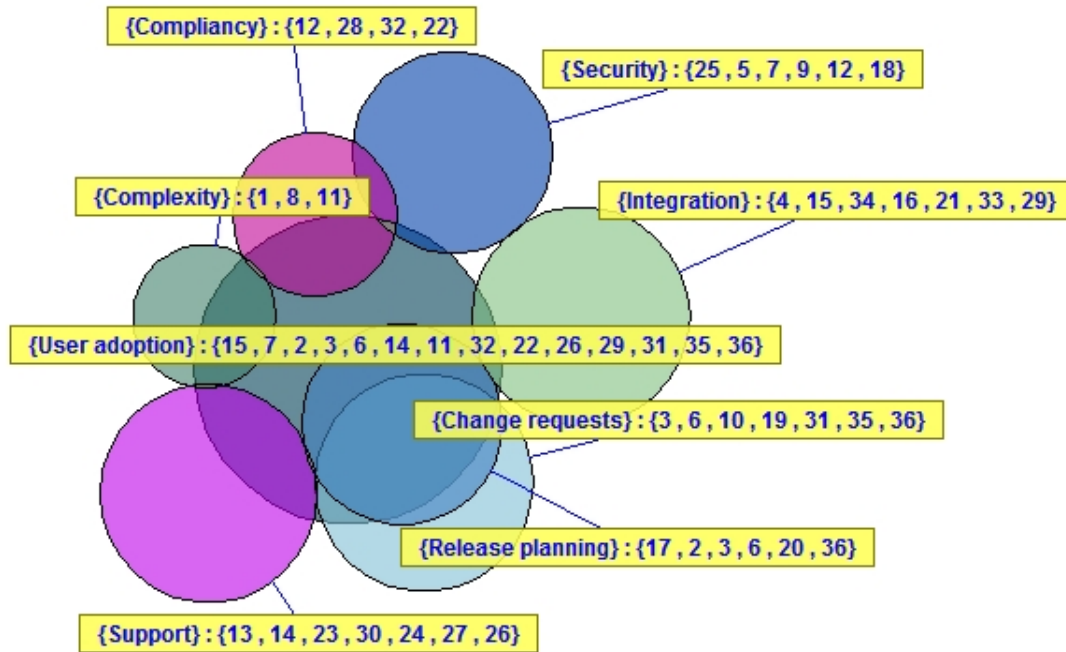


Figure 32: Perspective comparison from the user adoption viewpoint

We can conclude that to development costs go up because of risks related to integration support, complexity, change request, release planning, integration, security and compliancy. The attentive reader might notice that this risk theme overlaps with all risk themes from the user IT organization perspective. We can therefore argue that all risks themes from this perspective affect user adoption levels.

According to experts mitigation strategies that reduce the risk on higher data integrity & confidentiality costs are IM7, IM11, IM12, IM13, IM14, IM16, IM17, IM18, IM19, IM22, IM24, IM25, IM26, and IM27.

6.5 MITIGATION STRATEGIES

In the previous chapters we have elaborated on the individual risk themes and stated what mitigation strategies are considered to reduce the risk themes. In this chapter we further elaborate on the mitigation strategies and provide an overview of strategies and their impact on the found risks. In Table 36 we present an analysis on the mitigation strategies with respect to the risk themes.

Table 36: Analysis of mitigation strategies and risks themes

Risk theme	Is mitigated by # strategies	%of total mitigation strategies	consists of #risks	%of total risks	Difference of %
Complexity	8	27.59%	3	8,33%	19,25
Release planning	6	20.69%	6	16,67%	4,02
Integration	9	31.03%	7	19,44%	11,59
Change request	3	10.34%	7	19,44%	-9.10
Compliancy	7	24.14%	4	11.11%	13.03
Support	6	20.69%	7	19,44%	1.25
Security	6	20.69%	6	16,67%	4.02
System performance	2	6.90%	5	13,89%	-6.99

High development costs	9	31.03%	14	38,89%	-7.85
Data confidentiality & integrity	3	10.34%	5	13,89%	-3.54
User adoption	14	48.28%	14	38,89%	9.93
Maintainability	11	37.93%	12	33,33%	4.60

In the table we compare the risks themes and the amount of mitigation strategies that are found to mitigate these themes. We also look at the relationship between the amount of risks a risk theme consist of and how much mitigation strategies have been found to apply to that specific risk theme. We can see that four themes are having a negative difference and four other themes are below five difference percentage points. This means that the risk themes encompasses a relatively high amount of risks in relation to the complete risks set compared to the mitigation strategies in relation to the complete mitigation strategy set. We conclude from this that these themes are not covered as well as the other themes and might be considered problem areas. However, we do not know the impact or probability of individual risks and can therefore make no hard claims.

Another interesting view on the data we collected is what mitigation strategies are used more often than others. In Table 37 we have stated these statistics.

Table 37: Number of mitigation strategies applications according to experts

Mitigation Strategy	#mitigations	Mitigation Strategy	#mitigations
IM1	4	IM16	4
IM2	4	IM17	2
IM3	3	IM18	1
IM4	2	IM19	3
IM5	2	IM20	3
IM6	3	IM21	1
IM7	5	IM22	2
IM8	3	IM23	3
IM9	2	IM24	2
IM10	3	IM25	5
IM11	3	IM26	3
IM12	2	IM27	2
IM13	4	IM28	3
IM14	3	IM29	3
IM15	4		

From this information we can see that both IM7 and IM25 are used to mitigate five risk themes. Together with IM1, IM2, IM13, IM15, and IM16, that are used to mitigate four risk themes, these mitigation strategies can be seen as the best strategies when it comes to the amount risks themes they help to mitigate. IM18 and IM21 are only used once as one workshop group argued that they did not mitigated risks. During the discussion it became clear that both mitigation strategies were considered viable mitigation strategies but where misunderstood. When we asked about what mitigation strategies are considered to be a quick win, IM25 was general accepted as a good candidate. Another workshop attendee argued that shifting responsibility of on-premise to SaaS integration towards higher level management can also be very rewarding. Others agreed but argued that the costs would also be significant. It was also argued that the impact of governance related mitigation strategies would differ between organizations. IM1, which was mentioned by eight of the nine original interviewees, was considered to be very useful but it was difficult to assess whether it was cost effective. If a canonical model is already in place it would be relatively inexpensive to use, but building one explicitly for an on-premise to SaaS integration is considered to be expensive.

6.6 INTEGRATION FACTOR FRAMEWORK RISK MAPPING

We used the integration risk factor framework from Chapter 4 in combination with the risks we have identified to categorize the risks. In Table 38 we present our findings.

Table 38: Integration factor framework risks categorization

Factor	Percentage of risks
Technical	47.22%
Process	2.78%
Application	25.00%
Data	19.44%
Governance	58.33%
Security	13.89%
Total	119.44%

The total percentage is higher than 100% due to risks IR1, IR4, IR12, IR16, IR17, and IR33 have spanned multiple layers. When we translate this to the integration factor framework we get the following figure. The data shows that our earlier conclusions based on the interviews, concerning governance in on-premise to SaaS integrations, was correct.

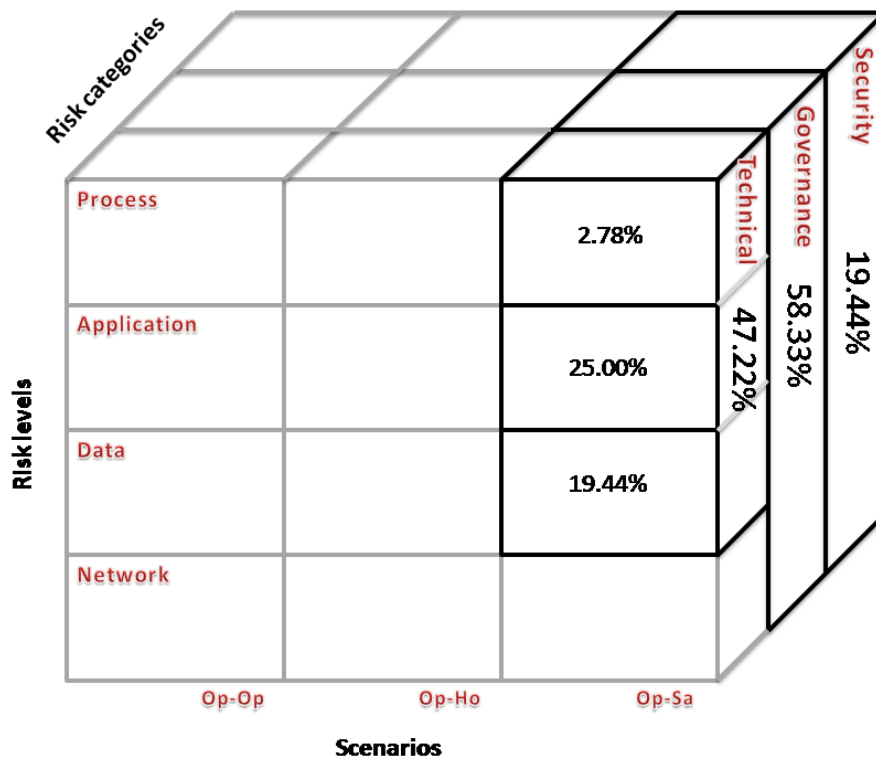


Figure 33: Risk factor framework with categorization of risks

To give better insight in this mapping of risks we have categorized risks per layer in Table 39.

Table 39: Categorization of risks

Non-technical			Technical		
Security	Governance	Data	Application	Process	
IR5, IR7, IR9, IR12, IR18	IR1, IR2, IR10, IR12, IR13, IR14, IR17, IR19, IR20, IR21, IR22, IR23, IR24, IR26, IR27, IR28, IR30, IR31, IR32, IR35, IR36,	IR1, IR3, IR4, IR11, IR15, IR33, IR34	IR1, IR4, IR6, IR8, IR16, IR17, IR25, IR29, IR33	IR16	

6.7 OVERVIEW AND VALIDATION

In this chapter we have combined the data we obtain from the systematic literature research, interviews and workshop in order to answer research questions IV and V. We identified twelve risks themes originating from two perspectives. We argue that the risks of integrating an on-premise system with a SaaS solution revolve around complexity, compliancy, release planning, change requests, security, integration and support when we look from a more traditional IT perspective. When we approach the risks from another perspective, the costs perspective, we argue that when we abstract upon all risks, the final risk would be that the integration costs are increased. Not fixing a broken integration is not an option, and therefore cost is a viable final risk that is also very useful when we look at the risks from a practical viewpoint. The themes of on-premise to SaaS integration that revolve around the increase of costs can be divided in development, maintainability, data confidentiality and integrity, user adoption and system performance.

In previous chapters we have compared the two perspectives and Figure 34 is overview of this comparison. In the figure we have approach on-premise to SaaS integration from the costs perspective and placed the other themes as sub-domains.

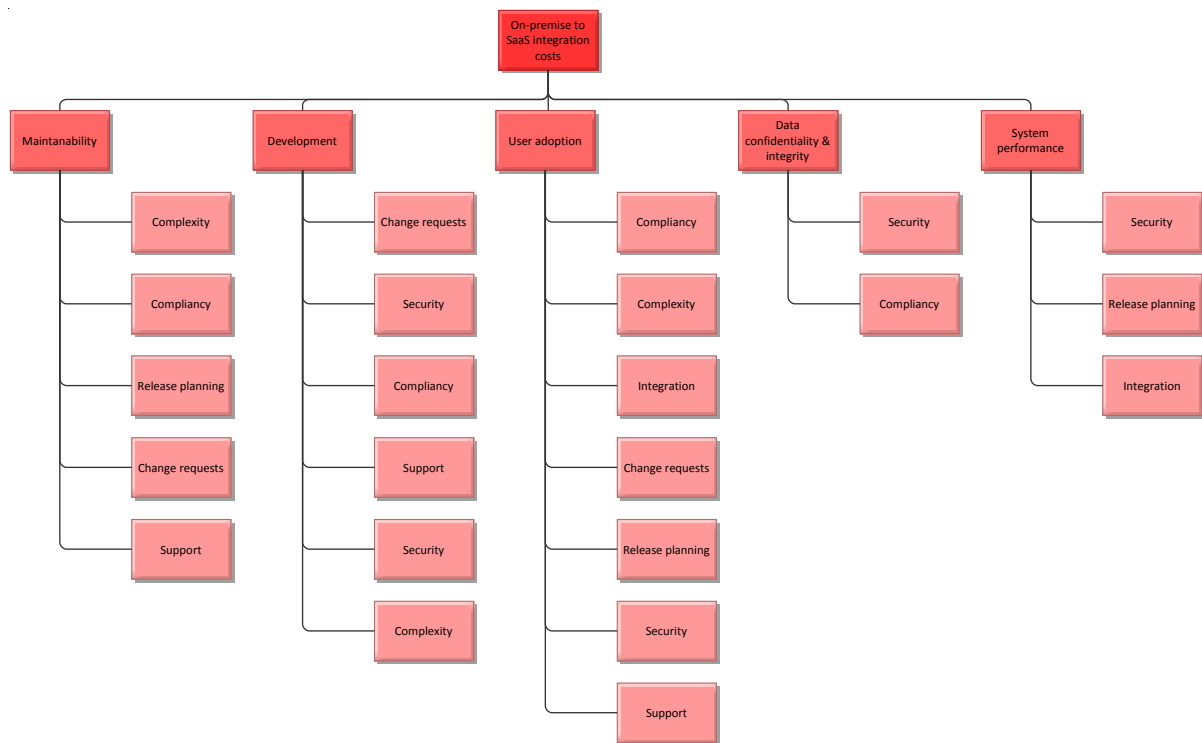


Figure 34: Overview of the comparison of risks perspectives

Throughout the workshop we have validated our research findings. First, we asked experts for risks that were considered not on-premise to SaaS specific or risks that were not considered risks at all. Secondly, we asked expert to critically asses the mitigation strategies we have found. We can conclude that all risks and mitigation strategies have passed the test but we do like to point out a few anomalies. All SaaS solutions are considered to be different from the other, i.e. some SaaS solutions can be more mature, meaning i.e. less uncontrolled changes and a more rigid release cycle. The risks we found are gathered from different experts, projects and experiences. This means that all risks are considered valid, but the impact and probability of encountering specific risks will vary between SaaS solutions and integration environments. The mitigation strategies we found can be roughly divided into two categories: technical and governance. The technical mitigation strategies can be considered not on-premise to SaaS integration specific, this can simply be explained by an earlier finding that technically integrating with a SaaS presents little challenges. The governance strategies can be considered more tailored towards the specific dynamics of on-premise to SaaS integrations. Both sets of mitigation strategies are considered valid, but experts argue that the governance strategies are influenced heavily by the overall governance structure of the implementing company.

7 CONCLUSION

In order to find the risks and mitigation strategies specific to on-premise to SaaS integrations we have conducted an explorative literature study, a systematic literature study, interviews, and have organized a workshop. In the previous chapters detailed reports of these activities have been written down in order to answer the underlying topics of our research. In this chapter we reflect on our previous chapters and answer our main question in one parsimonious answer.

Finding drivers and barriers for ERP-as-a-service adoption gave us insight in the reasons behind the adoption of a SaaS solution in the ERP landscape, and why ERP-as-a-service is not yet adopted by all companies. Before we could dive into these drivers and barriers of ERP-as-a-service we had to define key concepts used in our research. This led to the ‘discovery’ that cloud computing might be one of the world’s biggest misunderstandings. The term cloud computing is surrounded by such a thick layer of marketing that it was hard to find an agreeable definition. After many discussions with colleagues and reading numerous scientific articles, including the work of Hmood & Al-Madi (2013) who identified over 70 definitions of cloud computing, we argue that the NIST definition of cloud computing is the most elegant solution. Due to the use of five essential characteristics the NIST definition is both simple and complete and allows for the so-called ‘gray area’ that can be found between cloud computing and the more traditional IT solutions such as hosting. We also conclude that the technology behind integration with the SaaS solution is not new, as cloud computing in general is not a completely new technology but can be merely seen as hosting 5.0. In Table 40 we have stated the drivers and barriers from the user perspectives that were mentioned in multiple sources. For a complete overview of all drivers and barriers for adopting ERP-as-a-service, including those of the SaaS vendor, we refer to chapter 3 as they are too numerous to mention here.

Table 40: Key drivers and barriers for ERP-as-a-service adoption from the user perspective

User drivers	User barriers
Reduced up-front investments	Lack of policies and laws
Predictable and lower costs	Less customization and integration possibilities
Access to and flexibility to choose between state of the art technologies	Large dependency on vendor
Remote access from anywhere at any time	Increased security risks
Easier access to technical expertise	Structural changes
Improved processes	

The most important conclusion that can be made from these drivers and barriers, considering the scope of this particular research, is that **integration is considered a direct barrier from the user perspective and challenges on the vendor side of the spectrum show that suppliers are in need to develop applications that allow for integration in order to achieve customer satisfaction**, i.e. using a service architecture. This complimented the sources we had already, stating that integration was an important piece of the puzzle. We argue that the drivers and barriers of ERP-as-a-service adoption have been researched extensively, but taking a step further, i.e. focusing on integration results in a very immature research niche.

From the found drivers and barriers of ERP-as-a-service we progressed to defining integration and what factors of integrations influenced risks, results of this work can be found in Chapter 4. This was an essential step in finding the specific risks we required, without these factors we could have never scoped this research as sharp as we needed in order to avoid drowning in every integration risk that exists. We used the different ERP delivery models as a starting point to define three different integration archetypes. From our research we conclude that integration from on-premise to on-premise, from on-premise to hosted, and on-premise to SaaS, are considered the main ‘flavors’ of

integration scenarios. The difference between these flavors pivoted on two essential characteristics. First, integration could pass through organizational boundaries, or remain on-premise. Secondly integrations can be made with applications that are owned by the user organization or that are 'rented' in a multi-tenant model. These two characteristics create the interesting dynamics that exists when integrating between on-premise systems and SaaS solutions. Besides the integration scenario we have identified three different integration layers on which risks can occur: Technical, governance, and security. Within the technical level we have identified four sub-layers, consisting of the network layer, data layer, application layer and the process layer. We have excluded the network layer from our research scope, we argue that it allows integration to occur, but the actual integration takes places on the other levels. We have created a three dimensional cube that visually represents the integration risk factors and can be found in Figure 17.

After defining the integration risks this research focused upon, we started collecting these specific risks and strategies to mitigate these risks from the scientific community, the complete results can be read in Chapter 5.1. During a rigorous systematic literature research, covering over 1000 articles, we set out to obtain all specific risks and mitigation strategies known to the scientific community. This systematic literature research resulted in nine risks and six mitigation strategies, making it clear that research towards on-premise to SaaS integration risks was immature. The next step was to interview experts at Capgemini to obtain data for practice. The interview approach can be found in Chapter 2.3. After combining the results from both the interviews and the systematic literature research we collected 36 specific on-premise to SaaS integration risks and 29 mitigation strategies. These risks include specific risks that are not completely new but are augmentations from more general integration risks. These augmented risks have a greater impact due to the specific dynamics of on-premise to SaaS integrations. An overview of the interview results can be found in Chapter 5.2. Besides the identified risks and mitigation strategies the following key conclusions were made by interviewing experts:

- On-premise to SaaS integrations are technically not challenging, in the sense that those integrations do not require any substantially new knowledge.
- Integration governance becomes more important and more complex due to the unique dynamics of an on-premise to SaaS integration.
- When embedding SaaS solutions into a complex application landscape the savings made on initial investment can be reallocated to integrating the solution. This goes against popular claims that state a SaaS solution is cheaper than an on-premise solution.

In Chapter 0 the workshop approach is explained. This workshop was used to cluster the found risks into risk themes, link risk themes with mitigation strategy and categorize the found risks with the aid of risk factor framework. We obtained two different perspectives from the clustering activity, overall resulting in 12 risk themes.

Table 41: Overview of risks themes

User IT organization perspective	Integration costs perspective
Complexity	System performance
Integration	Data confidentiality & integrity
Compliance	Maintainability
Security	Development costs
Release planning	User adoption
Change requests	
Support	

The seven themes from the first perspective (see the left column of Table 41) one represent the traditional view from the user IT organization. The other five themes (see the second column) are part of a cost orientated perspective. These perspectives help us understand what risks can be encountered during an integration between on-premise and SaaS solutions, without us having to exhaust the list of potential risks. In Chapter 6 we elaborate on both of the perspectives, the individual themes, and their comparison. We have collected various kinds of data on the mitigation strategies and have described which of the 29 mitigation strategies reduce what risk themes. From this data and comments during the workshop, we have made a number of key conclusions:

- IM7, IM25, IM1, IM2, IM13, IM15, and IM16 are mitigation strategies that reduce the most risk themes.
- Themes revolving around change, maintainability, development, performance, integration support and security are considered key themes because they host a large quantity of risks compared to the amount of mitigation strategies that have been identified that can help to reduce these risks themes.
- IM25 is considered the most cost effective mitigation strategy.
- The impact and probability of risks and mitigation strategies differ per integration environment.

The individual risks that are placed in the risks themes have also been mapped on the risk factor framework. From this activity we can conclude that on-premise to SaaS integration risks are mainly governance related and not technical. We have not officially categorized the mitigation strategies but from our point of view these strategies show the same kind of categorization. This leads us to conclude that on-premise to SaaS integration is a mainly a governance challenge. Technical there are no 'unknowns'. A complete overview of the mapping can be found in Chapter 6.6.

In Chapter 6.7 we elaborate on the validation of our findings. Based upon the validation efforts of our workshop attendees we argue that all risks and mitigation strategies are considered valid but, as said above, the impact and probability of the risks and the potential of the mitigation strategies is unknown and can differ between integration environments.

7.1 DISCUSSION, IMPLICATIONS AND LIMITATIONS

We set out to identify the risks specific for on-premise to SaaS integrations and mitigation strategies to cover the knowledge gap that existed in scientific literature. Covering this gap in literature has some implications for practitioners and researchers. First, it helps practitioners think of practical ways to increase the implementation success ratio of large implementation projects that contain on-premise to SaaS integrations. The implementation success ratio can increase because we now know what risks we can encounter and how we can reduce these risks. Secondly, besides identifying risks we have also concluded that cloud computing is not a completely new 'computing recipe' but more closely resembles another 'computing flavor', however it does revolve around a set of interesting dynamics that create a unique situation that requires a different governance approach than traditional integrations. To ERP integrator practitioners this means that their role in such project changes from a mostly technical role to a more general consulting role. The more 'traditional' integrations required the consultant to initially set-up the integration during the implementation of the ERP system. When the client organization planned on updating their ERP systems to a new version the consultants returned to aid the client organization in updating, including keeping the integrations operational. In the 'traditional' situation both occasions can be seen as separate projects. When consultants are helping the client organization to integrate with a SaaS solution they have to understand the implication of the specific on-premise to SaaS dynamics. We argue that these projects still contain a technical core but require a more continuous approach instead of handling separate projects. For example the client organization must be ready for the implications of having little influence in the SaaS updates that are

'pushed' from the vendor side. The risks found during this research present many potentially problematic areas that require a more governance oriented approach than a 'traditional' integration.

This research has some implications for researchers engaged in empirical studies on ERP phenomena. We argue that in the future ERP systems will more than ever consist of multiple different modules that can not only originate from different vendors but can also consist of different computing models such as cloud computing. Because of this trend towards a more heterogeneous ERP landscape the overall application landscape on the client side becomes more complex. The increase in complexity originates from the specific on-premise to SaaS integration dynamics presented in this research. In example when we have multiple applications that operate in a closely integrated fashion it becomes clear that the lack of control on release cycles of some of these application increases complexity for the overall release planning. This change requires the focus of ERP phenomena research to move more towards continuity of the ERP system instead of mainly focusing on implementation. We argue that the continuity of the heterogeneous ERP landscape is threatened by the specific on-premise to SaaS integration dynamics.

Last, we address the possible threats to validity and limitations to the results obtained in this research project. A central question in any empirical study that employs qualitative research methods (e.g interviews, workshops) is about the extent to which the observations in our project are generalizable to other organizational settings. Clearly, we cannot claim universal generalizability of our results as it is impossible to cover all possible ways in which ERP SaaS systems could be integrated in client organizations. However, following research methodologists it might be possible to expect that our findings would be observable in organizations that operate in similar ERP markets, with similar organizational culture, similar values and level of maturity in project delivery (Seddon & Scheepers, 2012; Wieringa & Daneva, 2015). For example, it may be likely to obtain similar risk concepts in the other large consulting companies in the Netherlands committed to integrate other packages, e.g. SAP with the cloud. As Seddon and Sheppers suggest: "If the forces within an organization that drove observed behaviour are likely to in other organizations, it is likely that those other organizations, too, will exhibit similar behaviour" (p.12). We like to add to this that during our workshop none of the risks or mitigation strategies collected during our interviews was considered invalid, even though the data was collected from multiple different projects and experiences. This indicates the possibility that our findings are at least generalizable in the group of diverse experts we have interviewed.

Besides generalizability we have taken the utmost care during the design phase of this research to avoid unnecessary biases. We have adopted a multi-method approach that evaluated more than 1000 scientific articles, interviewed 9 experts from practices and conducted a workshop to acquire the data for our analysis. We have stated the limitations of every individual research method in Chapter 2, but the combination does not guarantee perfection, in the following paragraphs we present our self proclaimed limitations in order to allow the reader to interpret our work accordingly. Even though our sample of interviewees and workshop attendees is considered to contain a wide variety of different people, roles and experiences it is still possible that their combined view is biased. This is partially covered by the comparison we have made with the views from scientific literature but due to the immaturity of this research niche we cannot guarantee an unbiased view. We also argue, besides the possibility for biases, that list of findings is not exhaustive. Interviewing more experts and conducting more workshops could increase the number of risks and mitigation strategies. This increase in found risks and mitigation strategies could affect the identified themes and perspectives. The reasoning is simple: every integration environment is different. An integration environment contains the organizations that uses the integration, the laws and standards they have to uphold, the application and infrastructure landscape including integration targets and the integration technologies used. This is the main argument for clustering risk into themes as it allows us to grasp what kind of challenges await, while acknowledging we have not found all risks and mitigation strategies that exist. Besides the

challenges revolving our expert sample, the difference between integration environments, and the rigor and effort that has been put in our labor, we also need to take mistakes made by the researcher into account. If we include the supervisors, the group of researchers consists of four individuals but all the actual research is done by a single graduating student. This main researcher had limited experience in the research domain and if we combine this with the vast amounts of work undertaken it is possible that we have missed some interesting scientific article, part of transcription or comment during the workshop.

When we focus on the findings of our research we are proud to say we have taken the first leap to identify specific risks and mitigation strategies of on-premise to SaaS integration. The research approach we have taken proved to be a great fit with an under-researched research field but it comes at a price. The risks and mitigation strategies we have identified lack detailed specifications regarding their impact and probability. Therefore we cannot provide a definite answer to what risks or mitigation strategies are the stars of the show. We can only state how many expert mentioned specific risks and mitigation strategies and how many themes are affected by what risks or are reduced by what mitigation strategies. This limitation becomes even more complex regarding the fact that the integration environment differ every project, which affect impact and probability of risks and mitigation strategies.

7.2 RECOMMENDATIONS

In previous chapters the conclusion, discussion, implications and limitations can be read. In this chapter we present our recommendations how this work can aid practitioners. We have divided the recommendations into three stakeholders: the SaaS vendor, the user organization, and the integrator.

The SaaS service model is considered to be an improvement from the more traditional service models. By far the biggest improvement can be allocated to the SaaS vendor. The more traditional computing models i.e. required the software vendor to support multiple versions of their software and every new client required a new software environment that was often tedious to implement, especially if the user required customizations. The SaaS service model makes it simple for the vendor to enforce a single version and allow for a quick and easy 'implementation'. The risks that have been identified during this research view the integration from a user organization and integrator perspective. However much can be learned from the identified risks and mitigation strategies. The identified risks show clear challenges revolving around uncontrolled changes, complying with standards and audits, integration support, SLA management and communication surrounding these topics. These topics should be the basic building blocks of the non-functional requirements of the SaaS solution. Besides considering these requirements in the designing phases it is important to keep a long term perspective. Every change that has to be made to the SaaS directly affects the value proposition the SaaS vendor offers to the client, including potentially uprooting business critical integrations.

The user organization can use our work to obtain insight in the unique dynamics of on-premise to SaaS integrations, and the specific risks and strategies to mitigate these risks. This allows the user organization to create a stronger business case for acquiring SaaS modules to supplement their on-premise ERP system. Furthermore it allows the user organization to understand how changes to the governance strategy can increase the possibility on successful integrations.

In the section on the implication of this research we have stated that the role of the integrator changes from a mainly technical role towards a more general consultant role, due to the increasing complexity and importance in governance. We also argue that integrations are becoming more and more continuous projects instead of separate projects that return when implementing a new ERP module or when an upgrade is done.

The increase in complexity and importance of governance is one of the main conclusions this work presents but it can be difficult to grasp how the user organization or the integrator should use this conclusion. One of the risks identified stated that ITIL, an IT governance framework, is not 'cloud ready' and therefore could have problematic results when used blindly. We argue that when these IT governance frameworks are used in collaboration with the conclusions of our work they are good tools to help both the user organization and the integrator to provide the necessary governance improvements to cope with SaaS to on-premise dynamics. The IT governance frameworks provide a solid foundation of best practices, are widely adopted and come with tools, such as performance metrics, to increase the manageability of the governance processes. To make things even more concrete we use COBIT 5 to give an example. COBIT 5 contains the governance and management of enterprise IT (GEIT) framework that shows multiple processes that is used to structure IT governance (ISACA, 2012). We have presented an overview of these processes in Figure 35. From these processes we can clearly see similarities with our findings. If we use the GEIT framework in collaboration with our conclusions made, we argue that certain processes deserve more attention because they cover aspects of the integration that are more complex and important during an integration between an on-premise system and a SaaS solution. For example, uncontrolled changes are considered one of the challenges that need to be overcome when we integrate on-premise systems with SaaS solutions. We have defined technical mitigation strategies, such as the use of canonical model but mostly it is considered to be a governance challenge. In the GEIT framework we can find several processes, such as manage enterprise architecture, manage changes, manage change acceptance and transitioning, manage service request and incidents and manage continuity that can aid the user organization and integrator alike to govern uncontrolled changes better. The lessons we learned from this exercise is that although the governance complexity and importance has increased and on-premise to SaaS integration consists of a new and unique integration environment, we can use existing tools in combination with our findings to aid practitioners to control these new environments.

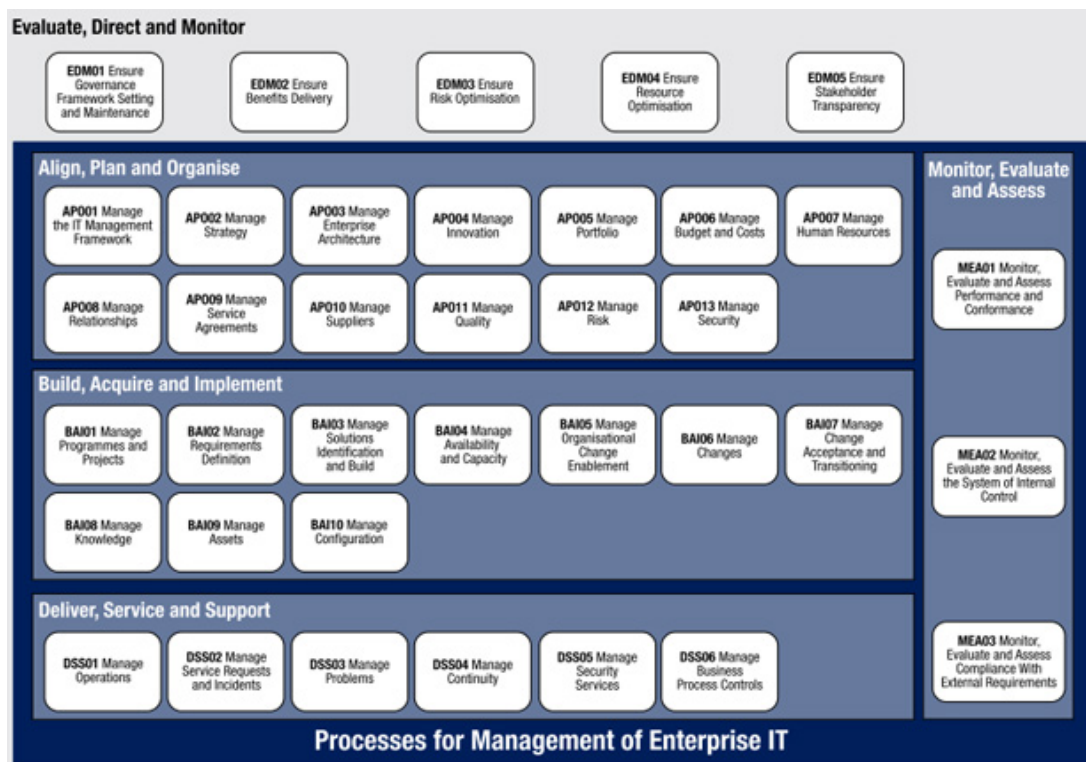


Figure 35: GEIT framework from COBIT 5 (ISACA, 2012)

8 FUTURE RESEARCH

In Chapter 7 we have discussed the research results and have stated the implications, limitations and recommendations of our work. In this chapter we elaborate on what future work can be done to increase the insights gained from this research and in the end provide more tools for practice to increase the implementation success ratio of large software projects. There are numerous potential new research projects that can build on our work. One of the potential research project that interest us is what changes to themes or perspectives occur when we interview more experts and organize more workshops, this could increase the strength of the claims made resulting from this research. During this work we have also validated the findings by way of expert validation; this could be complimented by doing case studies towards different integration projects. We argue that the probabilities and impact of risks, but also what the impact of the mitigation strategies is on specific risks, is an important next step. We have mentioned in previous chapters that these impact and probability figures differ per integration environment making a study towards them potentially complex. We argue that insight in these figures from one or two case studies could all ready provide enough insight to make strong claims about which risks and mitigation strategies deserve more detailed attention. At the same time it is interesting what exact factors of the integration environment influence the impact and probability of risks and mitigation strategies. If we take one step further back we argue that our study does not provide detailed insights into specific risks or mitigation strategies. Having these insights could improve the chances of successfully avoiding risks or applying mitigation strategies. This could mean that certain organization types should focus more heavily on governance related risks as their organizational structure is considered to be augmenting these risks. These insights could also help implement mitigation strategies, as we have only scratched the surface when it comes to the strategies or risks we have found. We argue that individual mitigation strategies could be subjected to a complete research project. I.e. it is valuable to know how a canonical model can be set-up in a way that makes coping with SaaS integration dynamic more easily.

Taking on the SaaS vendor perspective also creates a compelling case for future research. It is very interesting how all these risks can be used as a foundation for future SaaS development. If these risks are taken into account at the design phase of a SaaS many of them can be diverted.

Last but not least we argue that using our findings to improve IT governance frameworks could improve the overall success of the IT governance framework in coping with on-premise to SaaS integrations.

9 REFERENCES

- Aalmink, J., Gómez, J. M., & Schubert, A. (2010). Enterprise Tomography Driven Integration Lifecycle Management of Federated ERP in Green Clouds, 1–9.
- Adalety, D., Poppe, O., & Braa, J. (2013). Cloud Computing for Development – Improving the Health Information System in Ghana, 1–9.
- Addo-Tenkorank, R., & Helo, P. (2011). Enterprise Resource Planning (ERP): A Review Literature Report, //.
- Aloini, D., Dulmin, R., & Mininno, V. (2007). Risk management in ERP project introduction: Review of the literature. *Information & Management*, 44(6), 547–567. doi:10.1016/j.im.2007.05.004
- Alongi, V. (2008). *Sunset on a field of clouds (cover photo)*. Uruguay. Retrieved from <https://www.flickr.com/photos/vincealongi/>
- ANP. (2014, April 25). “Nederland Milkarden kwijt door falend ICT.” *Volkskrant*.
- Anwar, S. (2011). ERP Project Management in Public Sector – Key Issues and Strategies Rubana Mohsin, 1–10.
- Araujo, V. M., Vázquez, J. A., & Cota, M. P. (2014). A Framework for the Evaluation of Saas Impact. *International Journal in Foundations of Computer Science & Technology*, 4(3), 1–16. doi:10.5121/ijfcst.2014.4301
- Batini, C., & Lenzerini, M. (1987). A Comparative Analysis of Methodologies for Database Schema Integration, 18(4).
- Benlian, A., & Hess, T. (2011). Opportunities and risks of software-as-a-service: Findings from a survey of IT executives. *Decision Support Systems*, 52(1), 232–246. doi:10.1016/j.dss.2011.07.007
- Bernstein, P. A., & Haas, L. M. (2008). information integration in the enterprise.
- Bhattacharjee, A. (2012). *Social Science Research: Principles, Methods, and Practices*.
- Bingi, P., Sharma, M. K., & Godla, J. K. (1999). Critical Issues Affecting an ERP Implementation. *Information Systems Management*, 16(3), 7–14. doi:10.1201/1078/43197.16.3.19990601/31310.2
- Blair, G. S., Coulson, G., Robin, P., & Papatomas, M. (2009). An Architecture for Next Generation Middleware.
- Bolloju, N., & Murugesan, S. (2012). Cloud-based B2B Systems Integration for Small-and-Medium-sized Enterprises. *Proceedings of the International Conference on Advances in Computing, Communications and Informatics - ICACCI '12*, 477. doi:10.1145/2345396.2345475
- Breiter, G., & Naik, V. K. (2013). A Framework for Controlling and Managing Hybrid Cloud Service Integration. *2013 IEEE International Conference on Cloud Engineering (IC2E)*, 217–224. doi:10.1109/IC2E.2013.48
- Buyya, R., Broberg, J., & Goscinski, A. (2011). CLOUD COMPUTING.

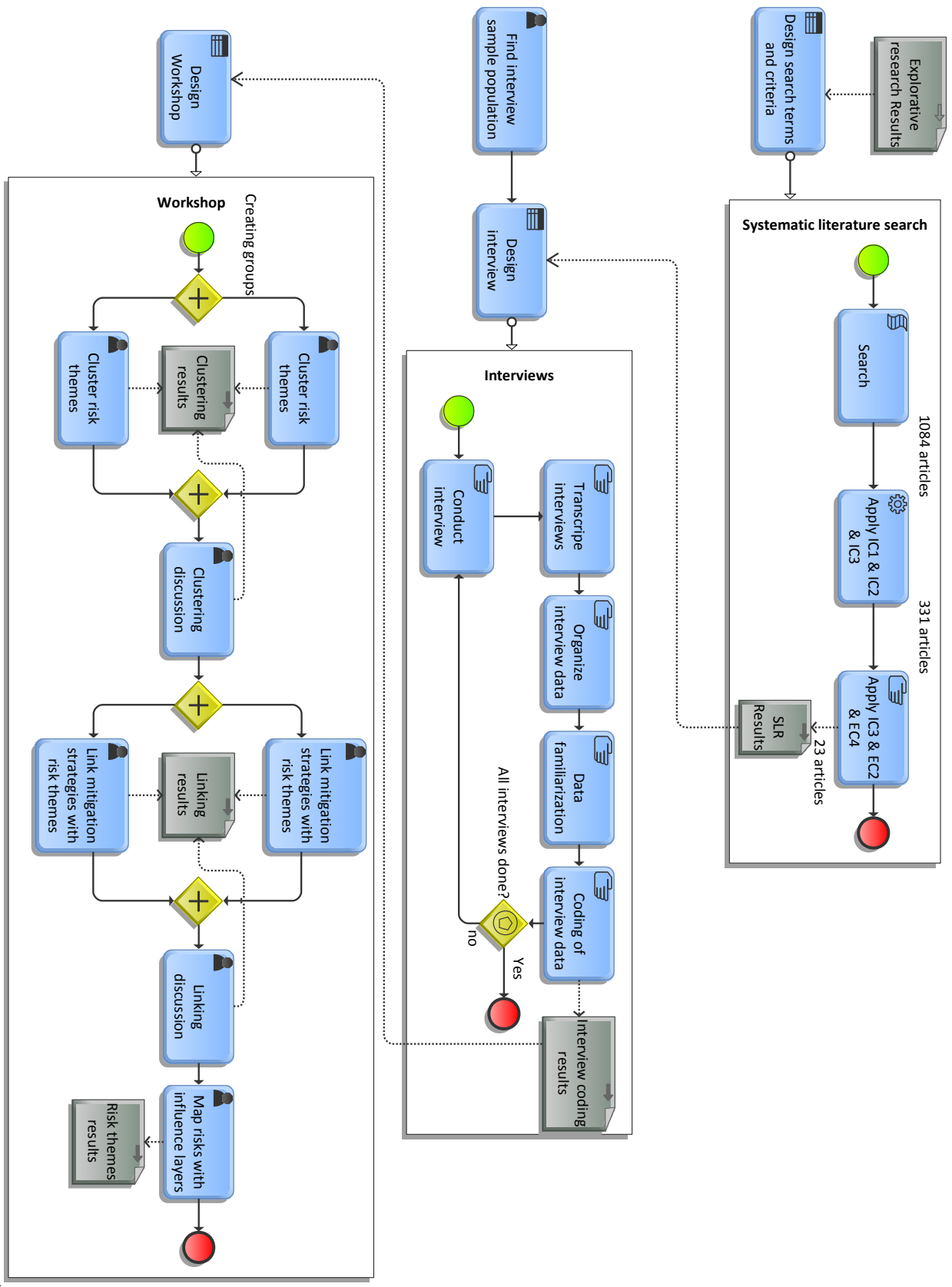
- Capgemini. (2014). Retrieved from <http://www.capgemini.com/>
- Creswell, J. W. (2003). Research design.
- Dierckx de Casterlé, B., Gastmans, C., Bryon, E., & Denier, Y. (2011). QUAGOL: a guide for qualitative data analysis. *International Journal of Nursing Studies*, 49(3), 360–71. doi:10.1016/j.ijnurstu.2011.09.012
- Duan, J., Faker, P., Fesak, A., & Stuart, T. (2012). Benefits and Drawbacks of Cloud-based Versus Traditional ERP Systems.
- Dubey, A., & Wagle, D. (2007). Delivering software as a service, (May).
- Esterberg, K. G. (2002). Qualitative Methods in Social Research.
- Giachetti, R. E. (2004). A framework to review the information integration of the enterprise. *International Journal of Production Research*, 42(6), 1147–1166. doi:10.1080/00207540310001622430
- Goyal, P. (2010). Enterprise Usability of Cloud Computing Environments: Issues and Challenges. *2010 19th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises*, 54–59. doi:10.1109/WETICE.2010.15
- Grabski, S. V., Leech, S. a., & Schmidt, P. J. (2011). A Review of ERP Research: A Future Agenda for Accounting Information Systems. *Journal of Information Systems*, 25(1), 37–78. doi:10.2308/jis.2011.25.1.37
- Graupner, S., Basu, S., & Singhal, S. (2011). Business Operating Environment for Service Clouds. *2011 Annual SRII Global Conference*, 1–10. doi:10.1109/SRII.2011.11
- Hai, H., & Sakoda, S. (2009). SaaS and Integration Best Practices.
- Haimes, Y. Y., & Chittister, C. C. (2012). Risk to cyberinfrastructure systems served by cloud computing technology as systems of systems. *Systems Engineering*, 15(2), 213–224. doi:10.1002/sys.20204
- Hasselbring, W. (2000). Information System Integration, 43(6), 33–38.
- Henderson, J. C., & Venkatraman, N. (1999). Strategic Alignment: Leveraging information technology for transforming organisations, 32(1), 472–484.
- Hmood, K. K., & Al-Madi, F. N. (2013). Impact of cloud computing on today's market Facilitating the move from local to international business.
- ISACA. (2012). COBIT 5. Retrieved from www.isaca.org/cobit5
- Juell-skielse, G., & Enquist, H. (2012). Implications of ERP as Service, 129–151.
- Kestler, H. a, Müller, A., Kraus, J. M., Buchholz, M., Gress, T. M., Liu, H., ... Weinstein, J. N. (2008). VennMaster: area-proportional Euler diagrams for functional GO analysis of microarrays. *BMC Bioinformatics*, 9, 67. doi:10.1186/1471-2105-9-67

- Kim, Y. J., Song, J., & Koo, C. (2008). Exploring the effect of strategic positioning on firm performance in the e-business context. *International Journal of Information Management*, 28(3), 203–214. doi:10.1016/j.ijinfomgt.2008.02.004
- Kitchenham, B. (2004). Procedures for Performing Systematic Reviews.
- Kolluru, N. V. S., & Mantja, N. (2013). Cloud Integration - Strategy to Connect Applications to Cloud, 2–7.
- Kotlarsky, J., Oshri, I., & Willcocks, L. (2012). The Dynamics of Global Sourcing. Retrieved from <http://link.springer.com/content/pdf/10.1007/978-3-642-33920-2.pdf>
- Kruize, J. W., Wolfert, S., Goense, D., Scholten, H., Beulens, A., & Veenstra, T. (2014). Integrating ICT Applications for Farm Business Collaboration Processes Using FI Space. *2014 Annual SRII Global Conference*, 232–240. doi:10.1109/SRII.2014.41
- Lacey, A., & Luff, D. (2009). Qualitative Data Analysis.
- Lackermair, G. (2010). Hybrid cloud architectures for the online commerce. *Procedia Computer Science*, 3, 550–555. doi:10.1016/j.procs.2010.12.091
- Lechesa, M., Seymour, L., & Schuler, J. (2012). ERP Software as Service (SaaS): Factors Affecting Adoption in South Africa, 152–167.
- Lee, B. J., Siau, K., & Hong, S. (2003). Enterprise Integration with ERP and EAI, 46(2), 54–60.
- Lewandowski, J., Salako, A. O., & Garcia-Perez, A. (2013). SaaS Enterprise Resource Planning Systems: Challenges of Their Adoption in SMEs. *2013 IEEE 10th International Conference on E-Business Engineering*, 56–61. doi:10.1109/ICEBE.2013.9
- Li, Y., Shen, Y., & Liu, Y. (2012). Utilizing Content Delivery Network in Cloud Computing. *2012 International Conference on Computational Problem-Solving (ICCP)*, 137–143. doi:10.1109/ICCP.2012.6383505
- Liu, C., Yu, Q., Zhang, T., & Guo, Z. (2013). Component-Based Cloud Computing Service Architecture for Measurement System. *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, 1650–1655. doi:10.1109/GreenCom-iThings-CPSCom.2013.299
- Liu, F., Guo, W., Zhao, Z. Q., & Chou, W. (2010). SaaS Integration for Software Cloud. *2010 IEEE 3rd International Conference on Cloud Computing*, 402–409. doi:10.1109/CLOUD.2010.67
- Liu, F., Wang, G., Chou, W., Fazal, L., & Li, L. (2006). TARGET: Two-way Web Service Router Gateway. *2006 IEEE International Conference on Web Services (ICWS'06)*, 629–636. doi:10.1109/ICWS.2006.127
- Malhotra, R., & Temponi, C. (2009). Critical decisions for ERP integration: Small business issues. *International Journal of Information Management*, 30(1), 28–37. doi:10.1016/j.ijinfomgt.2009.03.001
- Masiyev, K. H., Qasymov, I., Bakhishova, V., & Bahri, M. (2012). Cloud computing for business. *2012 6th International Conference on Application of Information and Communication Technologies (AICT)*, 1–4. doi:10.1109/ICAICT.2012.6398514

- Mathew, G. (2012). Elements of application security in the cloud computing environment. *2012 IEEE Conference on Open Systems*, 1–6. doi:10.1109/ICOS.2012.6417637
- Mell, P., & Grance, T. (2011). The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology.
- Melville, H. (1851). *Moby Dick*. London: Harper & Brothers.
- Mikki, S. (2009). Google Scholar Compared to Web of Science : A Literature Review, *1*(1), 41–51.
- Mislevis, A., & Grundspenkis, J. (2012). Workflow based approach for designing and executing mobile agents. *2012 Second International Conference on Digital Information Processing and Communications (ICDIPC)*, 96–101. doi:10.1109/ICDIPC.2012.6257277
- Neghina, D., & Scarlat, E. (2013). Managing Information Technology Security in the Context of Cyber Crime Trends Cyber-Crime Trends Analysis, *8*(1), 97–104.
- Pahl, C., & Zhu, Y. (2012). Data integration in mediated service compositions, *31*, 1129–1149.
- Pang, C. (2014). Market Snapshot : ERP Software , Worldwide , (May).
- Pearson, S. (2012). Privacy , Security and Trust in Cloud Computing.
- Puschmann, T., & Alt, R. (2001). Enterprise Application Integration - The Case of the Robert Bosch Group, *00*(c), 1–10.
- Raihana, G. F. H. (2012). Cloud erp – a solution model, *2*(1), 76–79.
- Ram, J., Corkindale, D., & Wu, M.-L. (2013). Implementation critical success factors (CSFs) for ERP: Do they contribute to implementation success and post-implementation performance? *International Journal of Production Economics*, *144*(1), 157–174. doi:10.1016/j.ijpe.2013.01.032
- Redmond, A. (2012). The Use of Cloud Enabled Building Information Models – An Expert Analysis.
- Schubert, P., & Adisa, F. (2011). Cloud Computing for Standard ERP Systems : Reference Framework and Research Agenda, (16).
- Seddon, P. B., & Scheepers, R. (2012). Towards the improved treatment of generalization of knowledge claims in IS research: drawing general conclusions from samples. *European Journal of Information Systems*, *21*(1), 6–21. doi:10.1057/ejis.2011.9
- Sumner, M. (2000). Risk factors in enterprise-wide/ERP projects. *Journal of Information Technology*, *15*(4), 317–327. doi:10.1080/02683960010009079
- Themistocleous, M., Irani, Z., & Love, P. E. D. (2002). ENTERPRISE APPLICATION INTEGRATION : AN EMERGING TECHNOLOGY FOR INTEGRATING, 1087–1096.
- Themistocleous, M., Irani, Z., O’Keefe, R. M., & Paul, R. (2001). ERP Problems and Application Integration Issues : An Empirical Survey, *00*(c), 1–10.
- Torbacki, W. (2008). SaaS – direction of technology development in ERP / MRP systems, *32*(1), 57–60.

- Utomo, W. H. (2013). Integration of SME, Industry and Government Through Public Infrastructure of SOA and Cloud Computing.
- Van Bergen, W., & Mos, B. (2014). Tientallen miljoenen verspild bij automatiseringsdrama SVB. *De Telegraaf*, (2 september). Retrieved from http://www.telegraaf.nl/dft/nieuws_dft/23033292/___SVB_loost_Capgemini_in_miljoenenproject_.html
- Vassiliadis, B., Stefani, A., Tsaknakis, J., & Tsakalidis, A. (2006). From application service provision to service-oriented computing: A study of the IT outsourcing evolution. *Telematics and Informatics*, 23(4), 271–293. doi:10.1016/j.tele.2005.09.001
- Wieringa, R., & Daneva, M. (2015). Six strategies for generalizing software engineering theories. *Science of Computer Programming*, 101, 136–152. doi:10.1016/j.scico.2014.11.013
- Yang, J., Wang, C., Liu, C., & Yu, L. (2013). Cloud Computing for Network Security Intrusion Detection System. *Journal of Networks*, 8(1), 140–147. doi:10.4304/jnw.8.1.140-147
- Yong, N., Liang, X. C., & Kai, Z. (2011). Connectivity as a Service: Outsourcing Enterprise Connectivity over Cloud Computing Environment. *2011 International Conference on Computer and Management (CAMAN)*, 1–7. doi:10.1109/CAMAN.2011.5778899
- Zhu, M., & Risch, T. (2011). Querying combined cloud-based and relational databases. *2011 International Conference on Cloud and Service Computing*, 330–335. doi:10.1109/CSC.2011.6138543
- Zhu, Z., Chen, L., Song, J., & Liu, G. (2010). Applying SaaS Architecture to Large Enterprises for Enterprise Application Integration, (20092006), 1–4.

APPENDIX A. OVERVIEW OF RESEARCH APPROACH



APPENDIX B. MITIGATION STRATEGIES AND RISKS

IM#	Mitigation strategy	Description
IM1	Use a canonical model	Use a service bus with canonical model in your middleware layer.
IM2	SaaS ready on-premise architecture	Have a stable and flexible architecture on premise that allows easy integration and disintegration. I.e. SOA architecture.
IM3	Create a specific governance strategy for SaaS integration	Specifically create a strategy for SaaS integration, apart your general integration governance strategy, in order to cope with the specific SaaS integration risks.
IM4	Use specialized SaaS security services	There is security software available special made to cope with SaaS dynamics. Use these specialized tools to get a secure solution.
IM5	Create a specific SLA management strategy for SaaS integration	Create a strategy on managing SLAs for SaaS integrations, apart from the general SLA management strategy, in order to cope with specific SaaS integration related SLA challenges.
IM6	Actively check compliancy with standards and audits of SaaS vendors	Go to your SaaS vendor and check if they are complying to standards and audits, or ask for independent, up-to-date certificates stating that the vendor is complying to standards and audits.
IM7	Create a dedicated role for overseeing SaaS integrations	Use a dedicated integration specialist role that is involved from the very beginning with SaaS integration. The dedication is necessary due to need for proactive management, i.e. monitoring and anticipating future changes.
IM8	Use governance when technical possibilities of SaaS integration are limited	When technical possibilities of the SaaS application are limited, i.e. around access management, use extra governance to overcome risks.
IM9	Encrypt messages and use secure channels	Use both data encryption and secure channels when sending data outside the organizations domain.
IM10	Use service registries	Use service registries to have a better and faster understanding of the impact of services and changes to the services.
IM11	Classify business objects towards SaaS availability beforehand	Business objects and their data should be classified beforehand on whether or not the organization is ok with potential storage or usage in the cloud.
IM12	Create a release cycle strategy aimed to cope with SaaS dynamics	The release cycle strategy should incorporate the dependencies on uncontrolled release plans of SaaS, and aim the on-premise (controlled) release cycles with these dependencies in mind.
IM13	Set up service contracts in a smart way	Service contracts (WSDL) need to be setup in a way that creates stable interfaces that do not turn into hidden broken interfaces.
IM14	Make use of more senior requirement management	An increased effort should be made considering requirement management, this calls for more influential management.
IM15	Make use of dev-ops teams	Use teams that remove or reduce the gap between development and operations.
IM16	Create a back-up strategy for broken interfaces	If an interface breaks, you should have a back-up strategy (i.e. raw data interface) to allow the organization reach its data.
IM17	Use a governance champion	Governance needs a strong mandate from higher


		management to create the means to act quicker.
IM18	Use a proactive SaaS procurement strategy	Be proactive around potential challenges when procuring a SaaS applications.
IM19	Use a more senior release; manager	To manage the release plan of a landscape including SaaS integrations, it is smart to have an influencing release plan manager.
IM20	Use available security options	Every SaaS has its own security options that, when used, provide the safest SaaS integration solution.
IM21	Put your middleware in the cloud	Make use of cloud based middleware between SaaS and on-premise applications.
IM22	Use stricter governance on changes	Governance needs to be more strict on changes in the integration environment.
IM23	Give special clearance for hotfixes	Hot fixing requires a need to be able to work quickly and thus having a strong mandate.
IM24	Use more proactive integration management	Use proactive integration management to prevent integration problems, i.e. by actively looking for potential future challenges or changes.
IM25	Test the SaaS application before using it	Test the SaaS, especially on non-functional requirements, before investing in an integration.
IM26	Specially focus on non-functional requirements	Specially focus on non-functional requirements as they are potential deal breakers with SaaS solutions.
IM27	Use a widely available and known role that is responsible for information on SaaS	Make sure the organization has a place where up-to-date information on SaaS applications, organization requirements and procedures can be found.
IM28	Standardize on-premise data models	Standardize on-premise data models towards best practice standards.
IM29	Use a unified semantic data model and business rules	Unified semantic data model and business rules can increase automation during integration.

IR#	Risks	Explanation
IR1	Increased chances on a fragmented landscape	The risks of getting a “wild growth” of applications and integrations increases when making use of SaaS applications.
IR2	More complex release cycle planning	Adding uncontrolled SaaS release cycles to the set of existing release cycles creates a more complex release plan overall.
IR3	Increased chances of changes in the data model	The chances on uncontrolled data model changes is increased when using SaaS applications.
IR4	Integration cannot scale with the SaaS application	The integration capacity does not scale up or down with SaaS usage.
IR5	Increased chance for the man in the middle	The risks that something is trying to change or read the data that is send over an integration is increased
IR6	Increased chances of changing APIs	The chances on uncontrolled API changes is increased when using SaaS applications.
IR7	Access rules misalignment	Technical limitation or different access management structures of SaaS create a misalignment with organizational access management strategy.
IR8	Easier to deliver erroneous products	SaaS makes is easier to start using an application, makes changes to the application, create connections etc. (especially for non-technical people) without looking at the rest of the landscape.
IR9	Increased security risks	Integrating with SaaS increases the chance of security issues. This mainly stems from reaching outside the user organizations domain and passing through the internet.
IR 10	Inefficient governance due to gap between business and maintenance	The organizational gap between business and maintenance results in delayed action. This delayed action leads to a lower service quality of the SaaS application.
IR 11	Integration is used to satisfy customer customization needs	The lack of customization possibilities around SaaS can entice the customer to demand customization in the integration. This could lead to unstable integrations.
IR 12	Integration security strategy misalignment	Security options of SaaS do not fit in the organization security strategy.
IR 13	Involving governance too late	Not involving governance in the early processes of SaaS acquisition and implementation could lead to increased problems in time.
IR 14	Less integration support due to tenant relationship	Vendor integration provides limited specific support because the user organization is “one-of-many”.
IR 15	Hidden broken interfaces due to wide defined WSDL service contracts	Allowing any type of variables in a WSDL contract creates “hidden” broken interfaces.
IR 16	SaaS integration environment influences your interface design patterns	The user organization needs to adapt to the SaaS which can influence your interface design patterns.
IR 17	Integration breaks due to uncontrolled upgrade	Integration stops working due to an upgrade from the SaaS vendor.
IR 18	Increasing opportunities for hackers	Integrating with SaaS creates more possibilities for people outside the user organization to misuse the integrations.
IR 19	New SaaS vendors have more frequent changes	During the beginning phases of the lifecycle of a SaaS vendor changes happen more often.
IR 20	Ad-hoc change due to security bug	In case of security bug, changes from the vendor side will happen with limited to no time to react.
IR 21	More complex integration management	Integration becomes more complex in a SaaS to on-premise integration.
IR 22	Technical limitations have a negative effect on	Technical restriction of SaaS can prevent compliancy to certain standards.

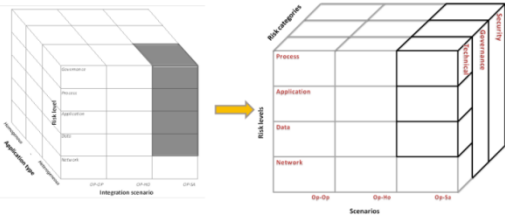
	compliance	
IR 23	ITIL is not made for SaaS	ITIL best practices are created in a time where cloud computing did not exist.
IR 24	More complex contract/SLA management	The complexity of contract/SLA management is increased.
IR 25	Less performance due to more security	Security measures have a negative effect on performance.
IR 26	Increased reactive handling of integrations due to lack of ownership	Proactive handling is decreased due to the lower feeling of responsibility over the SaaS application.
IR 27	Limited SLA options	The options for SLA arrangements are limited with a SaaS vendor.
IR 28	Increasing difficulties for auditing	Audit difficulties are increased when an organization makes use of SaaS applications.
IR 29	Image damage due to integration problems, even if you are not responsible	The risks of suffering image damage because of integration problems that are caused by uncontrolled factors is increased.
IR 30	Poor change communication and documentation from the SaaS vendor	Changes that happen without proper communication and documentation from the SaaS vendor.
IR 31	User is not ready for SaaS upgrade	The user is unable or unwilling to upgrade when the SaaS vendor does the upgrade.
IR 32	Logging and auditing dependencies on uncontrolled actors	Logging and auditing is dependent on the SaaS integration. The user organization is not in complete control over the integration.
IR 33	Dependent on the integration options of the SaaS vendor	SaaS vendor decides what integration options are available.
IR 34	Unstable interfaces due to too strictly defined WSDL contracts	Defining WSDL contracts too strict, creates interfaces that break unnecessarily often.
IR 35	No control over changes	The user organization has no control over changes to SaaS from the vendor side.
IR 36	More frequent changes in a SaaS integration environment	On-premise to SaaS integration is a more dynamic environment.

APPENDIX C. WORKSHOP

This appendix covers the workshop we have organized. First we will present the workshop presentation used to guide the workshop, secondly we will provide workshop and the raw workshop results.

 <p>Cloud integration risk mitigation workshop Leidsche rijn, 13 Januari 2015, Stijn Vorstenbosch</p> <p>People matter, results count.</p>	<h3>Introduction</h3> <ul style="list-style-type: none"> Research recap Organization Why are you here? Goals of today: <ol style="list-style-type: none"> Find higher level risk themes Link mitigation strategies towards these themes Map risks towards integration framework layers Validate findings 																		
<h3>Plan</h3> <table border="1"> <thead> <tr> <th>What?</th> <th>How long?</th> </tr> </thead> <tbody> <tr> <td>Introduction</td> <td>10 min.</td> </tr> <tr> <td>Finding risks themes</td> <td>20 min.</td> </tr> <tr> <td>Discussion of themes</td> <td>15 min.</td> </tr> <tr> <td>Linking mitigation strategies to risks themes</td> <td>20 min.</td> </tr> <tr> <td>Discussion of linking</td> <td>15 min.</td> </tr> <tr> <td>Introducing framework 2.0</td> <td>10 min.</td> </tr> <tr> <td>Mapping risks towards framework</td> <td>20 min.</td> </tr> <tr> <td>Evaluation</td> <td>10 min.</td> </tr> </tbody> </table>	What?	How long?	Introduction	10 min.	Finding risks themes	20 min.	Discussion of themes	15 min.	Linking mitigation strategies to risks themes	20 min.	Discussion of linking	15 min.	Introducing framework 2.0	10 min.	Mapping risks towards framework	20 min.	Evaluation	10 min.	<h3>Questions?</h3>
What?	How long?																		
Introduction	10 min.																		
Finding risks themes	20 min.																		
Discussion of themes	15 min.																		
Linking mitigation strategies to risks themes	20 min.																		
Discussion of linking	15 min.																		
Introducing framework 2.0	10 min.																		
Mapping risks towards framework	20 min.																		
Evaluation	10 min.																		
<h3>Finding risks themes</h3> <p>Rules: Time: 20 min.</p> <ul style="list-style-type: none"> Find higher level risks Minimal of 2 risks per theme Minimal of 4 themes Name the theme <p>Deliverable: Flipover sheet with risk theme stickers and theme names</p>	<h3>Risk themes discussion</h3> <p style="text-align: right;">Time: 15 min.</p> <p style="text-align: center;">Goal Find a single collective answer</p>																		
<h3>Linking of mitigation strategies</h3> <p>Rules: Time 20 min.</p> <ul style="list-style-type: none"> Link strategies to the risk Anything goes <p>Deliverable: Flipover sheet with areas representing the risk clusters filled with mitigation stickers</p>	<h3>Linking discussion</h3> <p style="text-align: right;">Time: 15 min.</p> <p style="text-align: center;">Goal What are key strategies?</p>																		

Framework 2.0



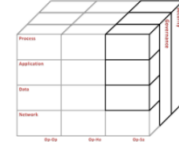
Capgemini

Mapping of risks

Rules:

- Map risks on the framework layers

Time: 20 min.



Capgemini

Evaluation

Contact information



Stijn Vorstenbosch
Email: stijn.vorstenbosch@capgemini.com

Capgemini

Capgemini

About Capgemini

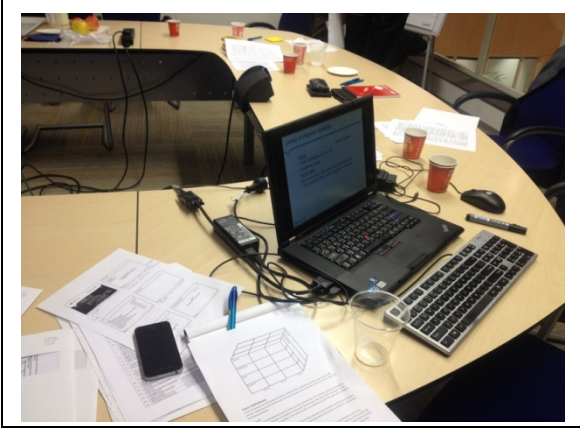
With almost 140,000 people in over 40 countries, Capgemini is one of the world's foremost providers of consulting, technology and outsourcing services. The Group reported 2013 global revenues of EUR10.1 billion.

Together with its clients, Capgemini creates and delivers business and technology solutions that fit their needs and drive the results they seek. In nearly 60 countries, Capgemini has developed its own way of working, the Capgemini Business Engine™, and driven its Employee 4.0™ worldwide delivery model.

www.capgemini.com

© 2013 Capgemini. All rights reserved. "Capgemini" is a registered trademark of Capgemini.





Sys Performance
16

Data confidentiality & integrity

Maintenance cost
6,7
25

Dev cost
8 20

User adoption
7 13 26

Sys Performance #2

Data confidentiality & integrity

Maintenance / Appl cost of operation
in \$, b .14, 17

High performance cost
1. 11, 12, 13 35
30, 32

User adoption, not obvious customer benefits
7 15

