University of Twente
School of Management and Governance
Chair of Technology Management

**MASTER THESIS**

FOR

MSC. BA PURCHASING & SUPPLY MANAGEMENT

# Procurement and asset management of commercial-off-the-shelf software

A case study of Thales Huizen

Submitted by: **Joël Benjamin Gugler**
s1480464

| | | |
|---|---|---|
| Supervisors: | 1st supervisor | **H.H Schiele (Prof. Dr.)** |
| | 2nd supervisor | **A.G Sigurðardóttir (Dr.)** |
| | Company supervisor | **D. Mladenova (Thales)** |

Contact e-Mail: jbgugler@gmail.com

Number of pages/words: **106/ 52295**
Bibliography program used: Evernote

Enschede, 17-07-2019 16:04:00

## ABSTRACT

Purchasing costs take up between 50 and 80 percent of an enterprise's operation. Such cost includes the acquisition of tangible – physical goods and intangible asset - software licenses.

Without a centralized system of asset management, such cost might not be optimized and there could be an additional cost from the operational risk of non-compliance in term of license usage. Since scientific literature in this field is lacking while there is a growing demand for a clear understanding and operationalization of software sourcing and management to reduce non-compliance risk, there is a need for theoretical research into this field and a practical solution for such problem.

The design approach is followed in which "problem identification and motivation, the definition of the objectives for a solution, design, and development" stages are explored. (Due to the scope of the research, the later stages of the design science approach are yet to be explored "demonstration, evaluation, and communication".) The central research question is "How can the software sourcing and asset management process be improved to lower risks?", following by three sub-questions:

SQ1) What is the current situation of software sourcing and management within Thales and their industry peers?

SQ2) What risks and problems occur with the current software sourcing (and management) process?

SQ3) What requirements and protocols would an improved software sourcing and management process need to have in order to minimise the risks and problems

In order to answer this question, a case study of Thales Huizen (and Thales Hengelo) is explored, followed by comparison to other industry partners (NS, Company A etc). The case study uses semi-structured interview and process modelling built on management theories found in the literature to identify the problems and suggest solutions. The interview process is done collaboratively and iteratively to ensure the rigour and trustworthiness of the study.

The research found that there is no centralized asset management system suitable for software purchasing currently exists in Thales Huizen nor does exist in their industry peers. The challenges are also the lack of synchronization across departments and across legacy system as well as there is a lack of accountability alongside with lack of a formal enforcement mechanism. These challenges pose a high operational risk and could lead to failure of software licensing compliance, resulting in an increase in operational cost and audit fines for non-compliance

Thus, a solution is proposed to create a SAM-centred software procurement (business) process in BPM. By utilizing the knowledge from interviews, industry insights and problem solving, a process is built that lays out a structured way for software to be procured. The outcome is a workable process that simplifies software procurement with clear software specific documents to fill in, to collect the data necessary to know precisely which software is sourced for which purpose and lays the data foundations for successful software asset management.

New introductions have been the COTS-Board, which is a group of software experts that oversee the software risks and help negotiate with the software resellers. Another new addition is the role of the software asset manager who is responsible for collecting the license specific data, and managing the software assets while executing accountability throughout the firm as the managers have to comply with supplying the required information. The research further on explores different possibilities for software that needs to be procured and/or managed. And lays out further positive possibilities (such as license-bundling, optimizing) for an integrative software asset management system.

What this research based heavily on is the utilization of existing infrastructure or processes to lower the eventual increased effort on the employees. There is a limited amount of change employees are willing to go through, and minimizing the actual change for them is crucial to minimize resistance.

**Keywords**

Purchasing process; Software; Commercial-off-the-shelf; License management; Software asset management; Business process modelling; Software sourcing; Risk; Compliancy

## ACKNOWLEDGEMENTS

Thanks to B.F. Postema for his usefulness in academic help through references examples from his PhD Thesis [1]. My father, Dr. Werner Gugler with his help with academic structures with his PhD [2] and Dr. H. Miri, for his help with the theoretical framework with his work. My academic friends N. Nguyen and A. Salac for giving me the feedback I needed to get going. Especially A. Salac, who helped me through many pizza infused work nights. Nga further on helped me so much with the editing and structuring of my thesis.

Professor Schiele receives praise for structural criticism and pushing me to work harder. I am eternally grateful for Dr. A.G. Sigurðardóttir for her help with the structure and teaching me negotiation, which I could use to negotiation the finishing of my thesis, amongst many other valuable life lesions. Without her, I would not have been able to do my Thesis.

Our newly formed "Nightly study association" *N.S.A. Sleepless* (Charles, Didre and Bryan) was responsible for a lot of motivating me to work harder and longer while finally embracing that I really live in the Ravelijn.

Thanks to the night-pass I could spend my nights working in the Ravelijn when the library had already closed, this not only provided a place to work, but also a home. Coming to the wonderful words of Shakespeare:

*"Parting is such sweet sorrow, that I shall say good night till it be morrow"*[3], as me finishing up my Thesis would mean that I would lose my night-pass (and my home in the Ravelijn), but also that I would finally graduate from a Masters of Science.

I further on thank the night guard of the Ravelijn for keeping me safe through my long and lonely study nights, as well as an pizza delivery guys that somehow always found a way to the door in the Ravelijn. Domino's, if you are reading this: Thank you!

D.C. Van der Griend helped tremendously with making the thesis academically.

The most significant help came from the wonderful employees of Thales Huizen, Thales Hengelo and Thales Paris. Thanks for the great trust you put in me and the freedom to create a strategy for software procurement and Asset Management. It is not usual that an intern gets to represent the company at headquarters after a month. Special thanks go to my colleagues, Robert Hodkinson and my supervisor Dess, without them I would not have been able to do this research.

[1] See Postema (2018), pp. 1-225

[2] See Gugler (1996), pp. 70-77

[3] Shakespeare (1806), p. 43

The Phd student V.F. Delke, is to thank for the optimized structure, the method section and the teachings in academic referencing.

Last but not least, I thank my now retired Olympus EPL-7, which gave me relief and career opportunities that I needed, and taught me that life is more than Software sourcing and Software asset management. Even though the research made it clear to me that Olympus's expenditures on Purchasing appear to be significantly high.[4]

Thank you for reading my thesis,

***Joël Gugler***

[4] Olympus_Group (2018), p. 92

# INDEX OF FIGURES

# INDEX OF TABLES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| COTS | Commercial-Off-The-Shelf |
| SAM | Software Asset Management |
| SW | Software |
| EULA | End-User-License-Agreement |

# GLOSSARY

**KIA:** *“Kwaliteit inkoop aanvraag” [Quality, purchase, and inquiry] KIA is a process used in Thales Huizen that introduces a new product into the configuration management system. More of a checklist than a process as the product specifications and export control are checked. After approval, the item will get an article number.*

**WITTE BON:** *(‘white ordering tickets’) are used to buy third-party supplied commodities, outside the regular Thales Huizen SIX / TTS ordering process in the QAD (‘ERP’) system.*

*RFQ: Request for Quotation*

***DTAP-ENVIRONMENT:*** *(Development, Testing, Acceptance, production)-Environment. Engineering companies typically work with a development-production process that starts with developing a product, testing it, accepting it and then producing it.*

***COTS-BOARD:*** *The Commercial-off-the-shelf or COTS-Board is a group of people with expertise on COTS-software that acts as a company advisory board for the procurement of software as a first point of contact.*

***SAM-TEAM:*** *The Software Asset Management or SAM-Team is a group of people with expertise in software and how to manage it (software asset management). Consisting of both engineers, lawyers and, managers they (in the context of this paper) are responsible for organizing, integrating and managing a software database into the company and handling Software audits.*

**HENGELO**: *The Hengelo department of Thales*

**HUIZEN:** *The Huizen department of Thales*

**MILITARY EXPORT RESTRICTIONS:** *e.g. The American* **ITAR** *(International Traffic in Arms Regulations),* **EAR** *(Export Administration Regulations) or* **Article 9** *of the Constitution of Japan, are export control regulations. All of them are designed to help ensure that defense-related technology does not get into the wrong hands or entirely outlaws military utilization of Japanese products.*[5]

**WINDCHILL PDMlink®:** *Configuration management system used by Thales Netherlands which includes all parts that are free to use as building blocks for (future) designs*

[5] See Pyman, Wilson, and Scott (2009), p. 224

# TABLE OF CONTENTS

# 1 INTRODUCTION: Problem and research identification

## 1.1 COMPLEXITY OF SOURCING SOFTWARE: Non-compliancy of intangible assets

Between 50 and 80 per cent of the costs of a company come from sourcing of the materials. [6] Purchasing management is a company's greater priority in order to maintain a competitive advantage [7]. Everything a uys are assets: tangible (Physical) or intangible (nonphysical) assets [8]. One of these intangible assets is software licenses, which is what the research will be about.

Throughout the years the purchase of commercial-off-the-shelf (COTS) software has sharply risen[9] as more companies switch to readily available (COTS) standard software solutions. COTS has the advantage of having a lower price than customised software solutions while being easier/faster to acquire.

Although software is easier to acquire than hardware, it is more difficult to buy correctly. This is mainly because software is different from hardware as it 'lives', is usually licensed, and – more importantly – comes with End User Licence Agreements (EULA) [10].

John Doe encounters the EULA when installing software and having to *'accept the terms and conditions'* to continue. Those terms and conditions are important for the end user of an intangible asset (software), and can be complex in nature while telling *who* can use it in *which way* for *how long.* If the user does not adhere to the agreement, he is non-compliant and there might be consequences. [11]

One example of non-compliancy is license overuse. Overuse happens when the ought the license for a certain number of users/nodes/machines and exceed this number upon audit. Overuse can also happen as access to third-party application. Alternatively, if a company is reselling the software (now embedded in a product), where it should not have been resold according to the End User Licence Agreements. To prevent problems from happening in the first place, it is therefore essential to know the risks and ways to do better.

Tangible goods on the other hand do not usually require a (renewed) license to (continue) using it. One cannot accidentally use more (physical) screws or engines than bought[12], but one can very easily let *more* people use the software than paid for. One can embed a screw in any

[6] See Akech (2005), p. 8; Bender, Brown, Isaac, and Shapiro (1985), p. 106; Chapman, Dempsey, Ramsdell, and Reopel (1997), p. 1; Day (2002), p. 2; Ellram (1996), p. 11; Gao and Tang (2003), p. 325; Waters (2006), p. 434

[7] See Ellram and Carr (1994), p. 12 also see Pearson and Gritzmacher (1990), p. 92

[8] See Kaplan and Norton (2004), pp. 10-11; Daniel and Titman (2006), p. 1607

[9] See Abts, Boehm, and Clark (2000), p. 1

[10] See Gomulkiewicz (1998), p. 910.

[11] See Wei, Zhang, Ammons, Bala, and Ning (2009), p. 92

[12] See Evans and Morriss (1984), p. 301

product, yet this may not be so simple or even allowed for software as it comes with a license agreement.

## 1.2 SOFTWARE EULAS AND AUDITS: A costly, but overlooked problem

In order to understand the objective relevance of software sourcing done correctly, and the problems and consequences of non-compliance, one must dive into the legal cases which have come to light in recent years as a result of software audits.

Recently Anheuser-Busch InBev settled with SAP for **$600M** over *'software non-compliancy'*, after unlicensed software was found on its systems which could be indirect accessed through interfaces with other software [13]. Diageo was forced to pay £57M to SAP over a similar problem, which was more than they had already paid for their license.[14] Mars struggled with Oracle over licenses[15], handed over 233.000 pages in documents[16] and the case was settled outside of court in 2015 for an undisclosed sum of money after a legal battle[17].

According to consultancy firms, it is estimated that **5-10 per cent of the revenue** from software manufacturers come from non-compliancy fines [18] and this number is only expected to grow. [19] The fines arise from audits from the software manufacturers who investigate if the customer is keeping true to the license agreement (not using more licenses in any way than agree upon).
Having to pay $600M after a legal battle or having to hand over almost a quarter million pages of documents is time-consuming, costly and simply unwanted.
In order for a firm to stay competitive in regards to its purchasing department, *software* should be taken into account in purchasing process optimization as it is part of the 50-80% of manufacturing costs. The extra cost dimension for software is the risk of audit fines which could jeopardize the entire firm. Risk assessment could soften the impact of the audit.
Using the risk matrix[20], risk assessment is done by assessing the likelihood and severity of the impact. In this case the likelihood of audit is dependent on the software vendors and not dependent on the customer per se. The variable that is possible to be reduced is the severity. In order to reduce de risk impact, the company has to reduce the severity, prepare for the audit or be fully

[13] See Anheuser-Busch-InBev (2018), p. 182
[14] See O'Farrell (2017), p. 2
[15] See MARS, INCORPORATED, ET AL VS. ORACLE CORPORATION, ET AL 2015), p. 2
[16] See MARS, INCORPORATED, ET AL VS. ORACLE CORPORATION, ET AL 2015), p. 4
[17] See DISMISSAL OF MARS, INCORPORATED, ET AL VS. ORACLE CORPORATION, ET AL 2015), p. 2
[18] See Lamoureux, Brill, and Joshi (2007), p. 8
[19] See Lamoureux et al. (2007), p. 8; Marquis, Spivak, and Barber (2016), pp. 3, 8
[20] See Hussey (1978), p. 7

compliant. Lowering the operational risk of non-compliance (which actually results in fines from audits) is the aim of the thesis.

Notwithstanding the rise in number of significant fines, there is seemingly a lack of empirical research and industry insights on how to cope with these modern purchasing problems of accounting for fines[21] for software non-compliance.
A search on Scopus with the keywords *"Software audit fines", "Commercial off the shelf software license"*, *"Software sourcing Audit", "software licensing audit",* or *"Software non-compliance audit"* from any year on, give no meaningful results, which might be because of its practical nature. The only information regarding auditory fines for software non-compliancy is found in grey literature from consultancies in the field. Although the legislative cases indicate that the problem is significant enough to be taken seriously and to be studied to utilize for consulting purposes. The legislative cases listed above (e.g. Mars vs Oracle or InBev vs SAP) were found not through academic sources.
Running the biggest software companies in the world [22] through a search engine with a legal case and a joker (e.g. {[Oracle]vs *} ) gave the legal cases found in legal databases needed to give an indication of the treat and seriousness of the problem.
Academic research on how to handle the new threat of having to pay significant amount of money to software auditors is lacking, therefore, this research is attempting to fill in that gap. The research consists of two elements: software and procurement. As there is a lack of academic research in the combination of the two elements, utilized sources can come from either of the fields to shed light on the problems surrounding software procurement. In the theoretical part those fields are further explored and attempt is made to combine them.

### 1.3 RESEARCH GOAL: Researching a new software sourcing process

As previously mentioned, the goal of this thesis is to examine *how* a company can innovate on a software sourcing and management process and the reasoning *why* it is relevant to pay attention to software sourcing. Due to software procurement being an under-researched field as for recent literature, possible research and insights into the problem must come from the case studies and interviews with companies. [23] As the focus is on a contemporary problem within real-life context, whereas the researcher does not have control over the elements.[24]
The obtention of the research goal will go in three steps, as based on the literature.[25]

[21] See Wei et al. (2009), p. 92
[22] See McCaffrey (2019), p. 1
[23] See Merriam (1998), p. 31
[24] See Yin (2017), p. 2
[25] See Chen (2009), p. 60

1) First, the current purchasing process of software needs to be seen in a holistic view.
2) Upon that, the risks of the current situation will be explored and evaluated.
3) Finally, a new process will be designed in regards to the previous obtained knowledge.

As employees in companies need to (be able to) use the designed or improved processes, a larger case study is done in Thales Huizen (see **chapter 1.3.1**) as a 3 month data gathering could be done in the company. As well as several smaller case studies of industry-peers where no long term data gathering could be done, for a holistic view of the situation

The design science research method is applied because it aligns with the quest for understanding and improving human performance while developing knowledge that can '*actually'* be used by the experts in the field to design solutions for their problems.[26]

From an academic view, the information and deep insights gathered from interviews with employees, resellers and industry peers on the practical applicability of an improved software sourcing and management process in a holistic view are the scientifical contributions of this thesis. The holistic view, analysis, and solutions are beneficial for other companies as it might assist them in lowering their risks on software audit fines and maximizing the benefits of both an optimised software procurement process and solid software asset management.

### 1.3.1 SCOPE: Commercial-off-the-shelf software for (complex) projects

**Thales** is a big complex Product & system supplier active in defence, aerospace, transportation, and communication. It supplies solutions in the range of a few million to a few billion Euros. It creates and sells highly complex systems, in which they embed software into the hardware to either manage the systems or to perpetuate them.

Triggered by the increase plausibility of audits due to recent fines at industry peers, a seemingly unstructured software purchasing process and increased software complexity of upcoming projects, Thales Huizen was orientating on what to improve in not only their purchasing department but in the processes that manage the entire life-cycle of software. This encompasses the documentation and configuration required for the *first-time* software are selected for tendering, the actual *purchase* of the software license, and the *management of the software assets* in the company. There were no indications that Thales Huizen was forced to pay auditory fines for non-compliance, yet the lingering danger of audits and self-knowledge that their software procurement process might not fully be formalized lead to the interest in conducting this study, to formalize and possibly inprove the situation.

[26] See van Aken (2004), pp. 155, 178

Regarding acquired software, there can be a differentiation of companies that use procured software for the sole cause of supplying its employees with software tools, and companies that also *embed software* into products that are resold. The last group of companies is dealing with more complicated issues. With increased complexity on the use of software, the risk of significant fines that come with non-compliancy and the nature of software being an intangible, yet valuable asset, rises. It is of great importance for companies to manage their software assets.

*Software asset management* (**SAM**) solutions that companies use diverge from simple spreadsheets to highly complex and dedicated (outsourced) software services.[27] As one solution might work for one company, but not the other, it is essential to look at every individual case in order to maximise efficiency and minimise the cost and effort a company has to spend in order to comply with the rules of audits.

### 1.3.2 DEVELOPING THE RESEARCH QUESTION: What makes a good software sourcing process?

To advice the company on their software procurement and software asset management, a research question and sub-questions will be asked even though one could see software procurement and software asset management as two different subjects that should be covered separately. The focus of this thesis is the integration of the two subjects into the software sourcing process. Starting with the need for software, and laying the foundation for the management of the software assets: software asset management.

As the goal is to design a new software sourcing process, the questions must be laid out for that. As they are linked to design science [28] and a new process needs to be designed, one can find in the most referenced paper about the subject that there are six steps of the design science process. Namely *"problem identification and motivation, definition of the objectives for a solution, design and development, demonstration, evaluation, and communication."*[29] With time constraints and slow development cycles, the research will only focus on the first three, and further research will have to evaluate the results of the renewed process. Coming back to the three steps of the research goal, the following sub questions are compiled.

**RQ)** *How can the software sourcing and asset management process be improved to lower risks?*

**SQ1)** *What is the current situation of software sourcing and management within Thales and their industry peers?*

[27] See Mackie (2014), p. 2
[28] See Peffers, Tuunanen, Rothenberger, and Chatterjee (2007), pp. 44-77
[29] See Peffers et al. (2007), p. 4

**SQ2)** *What risks and problems occur with the current software sourcing (and management) process?*
**SQ3)** *What requirements and protocols would an improved software sourcing and management process need to have in order to minimise the risks and problems?*

# 2 THEORETICAL FRAMEWORK: Software procurement processes and management

## 2.1 PURCHASING PROCESS: Current focus is hardware

Purchasing is defined as *"obtaining from external sources all goods, serices, capabilities and knowledge which are necessary for running, maintaining, and managing the company's primary and support activities in the most favourale conditions"* [30]. Purchasing is more focused on its objectives, which are *"To buy materials of the right quality, in the right quantity, from the right source delivered to the right time at the right place*" [31]
Alternatively, as Telgen forumulates it; Purchasing is*"anything resulting in an invoice"*[32]
When looking at the current literature about Purchasing processes[33], one finds that most build upon the research[34] of Van Weele.
The process that Van Weele designed and frequently cited is found in **Figure 1**
As is shown in Figure 1, the purchasing process includes both Strategic Sourcing and operative procurement, all cover three tasks.

**Strategic Sourcing** concerns three things.[35]

**1) Defining the Specifications** of a product - where Van Weele mentions [36] that functional and technical knowledge and specification should be looked at and documented. The importance of bringing supplier knowledge to engineering is also highlighted. This later accounts for 70% of the costs.[37]
**2) Select Supplier;** The process then talks about the supplier who will need to be selected from a list of possible suppliers via prequalification parameters and a request for quotation
**3) Contracting** (making sure everything arrives at the right price)
*Those are the tasks of Strategic Sourcing.*

[30] See Van Weele (2010), p. 14
[31] See Cavinato and Kauffman (1999), p. 61
[32] See Telgen (1994), pp. 87-88
[33] See De Boer, Harink, and Heijboer (2002), p. 26; Schiele (2017), pp. 4-5; Telgen (2005), pp. 1-2; Van der Valk and Rozemeijer (2009), p. 5; van Weele and Eßig (2017), p. 22
[34] See Van Weele (2010), p. 9
[35] See Van Weele (2010), p. 29
[36] See Van Weele (2010), p. 29
[37] See Telgen (1994), p. 2

**Operative Procurement** also covers three things.

4) **Ordering**; The *actual* paying and ordering of the product or service. [38]
**5) The expediting**; where the securement of the quality and timely delivery of goods and components is time guarded.
**6) Follow-Up and Evaluation;** Also in their job packages is the follow-up and evaluation of the product and delivery, wherein the usual case a feedback loop arises. The quality, delivery time and other factors of the product or service are evaluated, and it can be decided on whether or not to use this supplier in the future.[39]

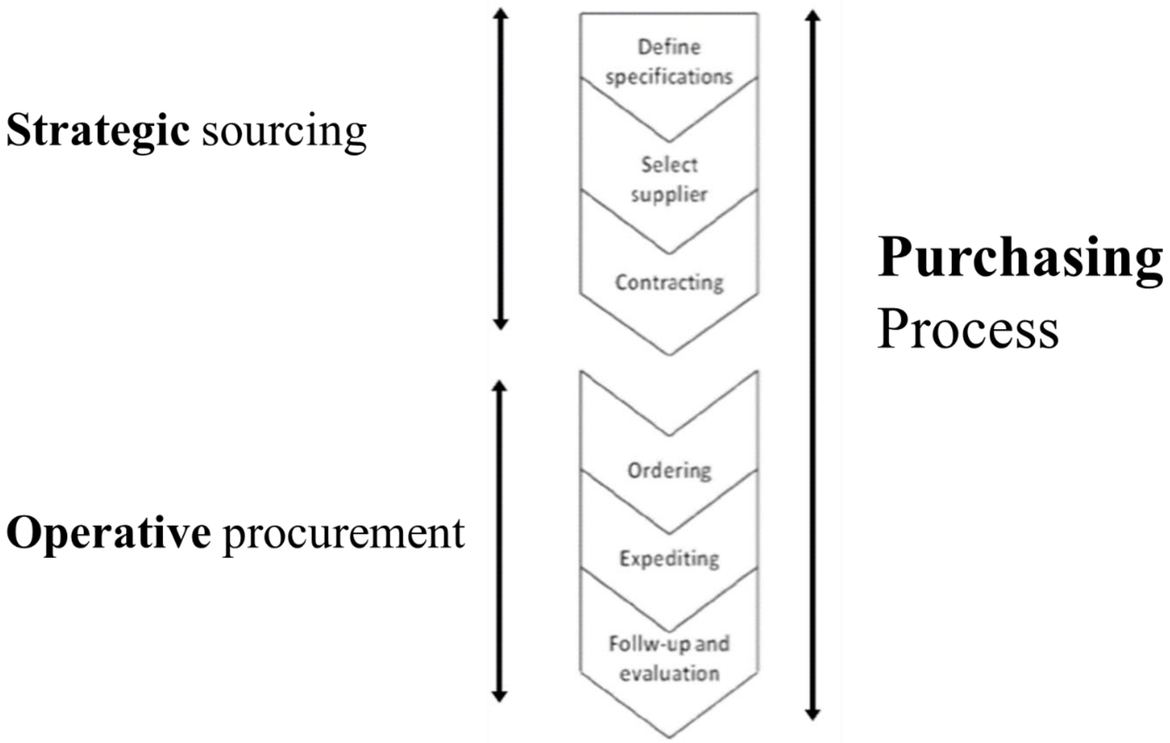


**Figure 1 Differentiation of Strategic Sourcing and Operative Procurement in the purchasing process.** [40]

As can be seen in the description of the process, the focus remains broad and is mainly for hardware. This can be seen in the fact that this process is designed to go from one step to the other. This procedure appears logical for hardware [41] – as hardware cannot be at two places at the same time, or quickly be downloaded from the internet and later be paid for. But is illogical

[38] See Van Weele (2010), p. 29
[39] See Van Weele (2010), p. 29
[40] Based on Van Weele (2010), p. 9
[41] See Van der Valk and Rozemeijer (2009), p. 3

for software. Unlike hardware, one can’t simply buy it, and do with it as one requires. The lack of flow can be a problem due to the almost fluid state of software. Next to that, software is not being mentioned in context with the process in any of the recent papers. This can be seen as a lack of focus towards software which needs to be addressed in this paper.

## 2.2 PURCHASING PROCESS INNOVATION: Best performance with full process revision and maturity

Van Weele [42] discusses the *three* types of purchasing which were identified by Robinson et al. [43] namely**, straight rebuy**, **modified Rebuy** and **new task.**

Based on this literature, **Table 1** was created. Here the numbers correlate with the six steps of the *purchasing process*. “1” being ‘define specifications’. “2” ‘Select supplier’ etc.

| **Type of Purchase** | **Description** | **Purchasing process steps needed** [44] |
|---|---|---|
| **New Task** | Which is a situation of which a product or service is bought for the first time | 1, 2, 3, 4, 5, 6 |
| **Modified Rebuy** | Where either an *alternative* product with or without an alternative price is bought from the same supplier, or the same or similar product is bought from an alternative supplier or a combination from the two. | 3, 4, 5, 6 |
| **Straight Rebuy** | Where the *same* product is just repurchased from the same supplier | 5, 6 |

**Table 1 Types of Purchases**[45]

(Schiele, 2007) [46] claims that purchasing departments can have different **maturity** levels. Purchasing maturity can be defined as *“The level of professionalism in the purchasing function”*. [47] Academics state that maturity level has a positive impact on Purchasing department performance. [48] This means that in most cases, a purchasing department that is highly structured, has feedback loops and high supplier integration will lead to more cost-savings. It is claimed that

[42] See Van Weele (2010), p. 31
[43] See Robinson, Faris, and Wind (1967), p. 126
[44] See Bildsten (2013), p. 4
[45] Based on Robinson et al. (1967), p. 126
[46] See Schiele (2007), p. 276
[47] See Rozemeijer, Van Weele, and Weggeman (2003), p. 7
[48] See Rossler and Hirsz (1996), p. 40; Rozemeijer et al. (2003), pp. 10-11; Schiele (2007), p. 1

purchasing development and thus maturity enhances purchasing performance [49], the performance of suppliers and the success of a firms.

With maturity comes a clear and easy to follow purchasing-process. Schiele lays out the dangers of not having a matured purchasing department, thus having an unstructured purchasing process in his paper of 2007, where Schiele argues that if the maturity is low *"introduction of best practices, such as an innovative cost-reduction method, may fail."* [50]

In order to analyze and possibly improve the purchasing process with or without maturities, one will need to know who is responsible for the parts, thus coming to the stakeholders.

Van Weele in his paper of 2010 [51] identified the **key shareholders** in this process, listed:

The **user,** which will work with the product.

The **influencers,** which influence the outcome by giving advice. Those are people who don't necessarily work with the product but might be the designer of a building, software architects or an expert who can influence the decision with advice.

**Buyers** are the people that negotiate with the supplier about the Terms and Conditions of the contract.

Next come the **Decision-makers,** which decide which suppliers to include and controls the budget. According to van Weele, they can also act as a designer who writes the specifications towards a specific supplier because of positive past experiences.

The author describes the role of the **Gatekeeper**, who oversees the information flow in the process, as they screen the contracts. Sometimes the role is assigned to a technical director's secretary, but a buyer can also be a gatekeeper.

Pearson and Gritzmacher [52] suggest that the roles of purchasing in the strategic management decision process are not highlighted well enough. They include purchasing in that process because it is crucial to source innovative parts to stay ahead of the competition and because purchasing plays a central role in identifying and analyzing supply trends.

The involvement and cooperation with purchasing are what the Pearson and Gritzmacher really focus on.[53] Seeing purchasing as a profit generation, rather than a cost-cutting function is the key to their research. In an ever-changing environment, software procurement could profit from the purchasing department. If all that purchasing is doing is trying to cut prices, there is a loss of potential gains such as supplier development and innovation. Integration will improve the motivation and efficiency of the employees who are engaged in purchasing and it would widen

[49] See Foerstl, Hartmann, Wynstra, and Moser (2013), pp. 691-692, 707; Paik (2011), p. 20

[50] See Schiele (2007), p. 1

[51] See Van Weele (2010), p. 28

[52] See Pearson and Gritzmacher (1990), p. 92

[53] See Pearson and Gritzmacher (1990), pp. 92, 93, 96

the fruits of their work. But in order to be successful, training the purchasing employees and improved communication with other departments is needed.

### 2.3 SOFTWARE: FunCompany Atally different from hardware

In order to design a purchasing process of software, it first has to be defined what software actually is.

The Institute of Electrical and Electronics Engineers (IEEE)) defines software as: *“Computer programs, procedures, and possibly associated documentation and data pertaining to the operation of a computer system*[54]” Whereas the ISO/IEC 9000-3 definition in software four components sees: “*Computer code, Procedures, Documentation, and the data which is necessary to operate the software system”*[55][56]

In its essential executive state, software is immaterial. The link to the tangible world comes from it’s the way that software can change material things [57]. One could say that software is bound to a hardware item.

Software is not manufactured; it is engineered or developed from materials that are amorphous and universal. “*It is a logical, rather than a physical system element”*[58] which leads to software being easily copied whereas the copy and original are identical. Further on software doesn’t wear out, but it evolves. [59] Updates come and go, and the one does not always have the right to use the software indefinitely and in all forms and features. It frequently has modules with extra functionalities that can be paid for or not. That may include service, only support, warranty, usage rights (per core, node, processor) or other functionality.

In contrast to many tangible products, the *costs* for software come from the *development* of it. The revenue comes from selling as many copies as possible, in a myriad of different license possibilities.

[54] See IEEE-Standards-Committee (1990), p. 66

[56] See Committee (1997), p. Sec. 3.11

[57] See Leveson and Weiss (2009), pp. 477, 484

[58] See Pressman (2005), p. 5

[59] See Pressman (2005), p. 11

Product-wise, software possesses some unique characteristics. The authorization of licensing comes down to that the manufacturer is not responsible for failures. When accepting the terms and conditions, one accepts an imperfect product that might include glitches, crashes or bugs which can cause considerable damage to the company, [60] something unimaginable for tangible goods. This all concludes that software is a tough thing to produce, and distinctly completely different from hardware. Software can also be tied to export restrictions. For example, Japanese software cannot be used for military purposes because of an article in their constitution.[61] Or American software isn't allowed to be *shipped* to Iran because of the embargo. Closer to home, software must have a European registration (EC), to be sold in the EU.[62]

The question can be asked on whether software is an asset. Because in contrast to hardware (assets), software is something intangible as one cannot hold it. However, there is a clear difference:

*If one buys a hardware item*, one possesses it, while the item in itself represents value.

*If one buys a software item*, one buys a license.

This license represents value. The license is the asset, as it gives you the right to use something. One can buy the right for 100 users, or 20 simultaneous users, 10 cores, several nodes, 40 company divisions, 20 computers etc.

The definition of an asset by academia is that has service potential or future economic benefit, is controlled by the organization and is the result of past transitions [63]

As one does not own the software, this means that the software is not the asset. But one owns the license code which can be controlled by the company and is the result of a past transaction, which makes that the asset.

In the end it has to be noted that there have been recent examples of companies experimenting with adding an end-user-license-agreement on hardware[64]

[60] See Charette (2005), pp. 1-2

[61] See Pyman et al. (2009), p. 224

[62] See Hanson (2005), p. 3

[63] See Schuetze (2004), p. 55

[64] See Hiltzik (2018), p. 1; Koebler (2018), p. 1; Wiens (2014), p. 1

## 2.4 COMMERCIAL-OFF-THE-SHELF SOFTWARE: Competitive advantage through mass production

Braun claimed more than 20 years ago that using commercial-off-the-shelf was not something new. [65] He wrote a lifecycle process for the effective reuse of commercial-off-the-shelf software. In this document, foundations of what can be nowadays be called 'Software asset management' can be found. [66]

In order to save money, companies don't develop much of their software but attempt to buy standard made packages (**commercial-off-the-shelf**), of which they can buy a license. Abts et al. () [67] mention the many advantages of using commercial-off-the-shelf software, and why companies use them in their development. A Commercial-of-the-shelf software product is defined as" a *commercially available or open source piece of software that other software projects can re-use and integrate into their own products"* [68]

According to Torchiano & Morisio () [69], the main characteristics of commercial-off-the-shelf software are that it's made exclusively for a project, closed or open source, non-commodity, integration in the final delivered system (but not in the development tool) and finally that its features and evolvement are not controllable.

Developing Commercial-off-the-shelf-Bases Systems is a method of constructing software systems by integrating multiple pre-existing commercial-off-the-shelf software components, each of which satisfies part of the system requirements. [70] The creation of systems from commercial-off-the-shelf software components offers the opportunity to reduce costs by sharing them with other users and has the potential to reduce training and infrastructure costs.[71] Consequently, by utilizing Commercial-off-the-shelf-Bases Systems, companies do not have to spend resources developing and maintaining costly systems and leave the development over to the manufacturer. In addition, such systems offer the possibilities of apartment and installation of modules and scripts to tailor the utilization, the so-called modified-off-the-shelfs[72]

[65] See Braun (1999), p. 29
[66] See Braun (1999), p. 29
[67] See Abts et al. (2000), p. 2
[68] See Torchiano and Morisio (2004), p. 91
[69] See Torchiano and Morisio (2004), p. 91

[70] See Brown and Wallnau (1996), p. 414; M. Vigder, Gentleman, and Dean (1996), p. 14
[71] See Braun (1999); Oberndorf (1997), p. 1
[72] See M. R. Vigder and Dean (1997), p. 7

*To summarize this chapter 2.3 and 2.4 regarding characteristics of software, get the following:*

**Table 2 Summary of the characteristics of (Commercial-off-the-shelf) Software**

| *Side* | **Explanation** |
|---|---|
| **Technical** | A set of functionalities, media and the basic lines of code. |
| **Legal/export** | The rights to use the product comes with terms and conditions. Also, one owns the license, the right to use the software, not the software itself. The products are potentially subject to export control restrictions specific to the classification defined by the publisher or the relevant national legislation |
| **Commercial** | The solution is sold in the form of several products (licenses, maintenance contracts, separate modules etc.). Further on it is an asset which can be taxed, and maintenance of it a liability |
| **Living** | Software is alive, it evolves and changes over time with updates |

**Source: Own summary of current chapter**

### 2.5 COMMERCIAL-OFF-THE-SHELF SOFTWARE PROCUREMENT: Premade comes with post-risks

In the previous chapters, both procurement processes and the nature of software have been discussed. In this chapter, it will be discussed what is known in the literature about software procurement. Software procurement is an understudied field for what purchasers use. Maria Ponisio writes that *“Despite a large number of studies, global IT sourcing projects are, in practice, performed ad-hoc and rely mostly on the manager’s experience”* [73]

The foundations of a process of the evaluation and selection process of Commercial-off-the-shelf software can be found in Al-Mahmood & Al A’ali (2011) [74] where they define the process as having *four* phases.

1. *Formation of Evaluation and Selection team and User requirements review* .
   In his phase one has to form an ‘evaluation and selection team’ and review and understand the user’s requirements and Business objectives
2. *Market research and Request for information issuance*
   In this phase, one has to do research on possible commercial-off-the-shelf software solutions and vendors, ask for information and create a list of qualified commercial-off-the-shelf software solutions and vendors
3. *Evaluation of commercial-off-the-shelf software solutions and Vendors*

[73] See Ponisio and Vruggink (2011), p. 1
[74] See Al-Mahmood and Al A’ali (2011), p. 296

In this phase one has to define selection criteria of the product and vendor, prepare, send and later evaluate the request for proposal and conduct the presentation.

4. *Selection of commercial-off-the-shelf software-solutions*
   In this phase, one has to analyse and review the proposals, make the selection of solutions and issue a letter of intent

**Table 3 commercial-off-the-shelf software selection process** [75]

This research attempts to combine the two common methodologies of the commercial-off-the-shelf software selection process [76] and solution contract management into a workable acquisition process of commercial-off-the-shelf software.

The focus of this report will be in the first part of this presented methodology, the selection process.

It comes down to evaluating, reviewing, fully understanding the software, its vendors and the End User Licence Agreement. Perform market research, evaluate those results and review the proposals based on criteria.

In the purchasing process, it comes down to selecting an item via attached *attributes* which are evaluated and compared. As concluded in chapter 2.3, software is funCompany Atally different to hardware; thus different attributes need to be assessed before making the decision. As a basis of this, Abts et al. (2000) and his co-author Boehm (1996) compiled a list of software attributes based on the IEEE standards on Software engineering. [77] The list contains of *correctness, flexibility, training, upgrade regulations, security, vendor concessions, product performance, functionality, intercomponent compatibility, version compatibility, vendor support, and maturity* [78]

Initially, this list included availability, but as times changed, software moved to an item that is always available, and ‘availability’ was therefore removed from the table. As was ‘understandability’ and ‘ease of use’ For a high-tech company to even consider a software to be used in their

[75] Derived from Al-Mahmood & Al A’ali (2011)
[76] See Comella-Dorda, Dean, Morris, and Oberndorf (2002), p. 87
[77] See Abts et al. (2000), p. 6; Boehm (1996)
[78] See Abts et al. (2000), p. 6

system, the engineers must understand it and be able to use it, which was a reason to exclude 'understandability' from the list'

A proper evaluation process of commercial-off-the-shelf software as a premise of successful implementation is supported by theory *"[An] organization will need to evaluate new product versions and potentially identify product replacements over the life of your system. If [one has] a foundation of good evaluation processes and practices, along with good documentation of the characteristics of products and the rationale for decisions, [one has] a good start at making COTS products work."* [79]

As the world has changed over the past two decades since the attributes were written down, the list will only be used as a reference point.

Another model that helps with understanding the purchasing items and the strategy needed for them in the context of vulnerability, complexity and importance to the company is the one presented by Kraljic.[80] In his 1983 Matrix, he differentiates purchasing items in *leverage* items, *strategic* items, *non-critical* items and *bottleneck* items.

The importance of software procurement or any procurement in general is to know your product and the market thoroughly to take opportunities and prevent risks.[81] Or as Krajic puts it, do a correct estimation of its importance of purchasing and the complexity of the supply market.[82]

As already concluded, the software can have a significant influence on the company result. Further on, software is atypical for Kraljic, as it has a considerable profit impact when the wrong software is ordered, and when non-compliant, but there are many suppliers due to resellers. As most companies in the industry don't buy from the manufacturers directly[83], there is usually one manufacturer, but several resellers.[84]

More recent research [85] on where to place on the Kraljic matrix have found out that there are two main categories, as the supply risk and profit impact were both irregular.. One cohort is the strategic items – which are the unique, innovative and technology leading products. Many of those are 'commodity' goods, but were mission critical and had a high buyer lock-in. The

[79] See Comella-Dorda et al. (2002), p. 96
[80] See Kraljic (1983), p. 1
[81] See Wang and Hazen (2016), pp. 2-3
[82] See Kraljic (1983), p. 3
[83] See Overdijk, van der Putt, de Vries, and Schafft (2011), p. 36
[84] Nettsträter, Geißen, Witthaut, Ebel, and Schoneboom (2015), pp. 2-3
[85] See Callagy (2007), p. 48

strategy recommended was to have a strong relationship with the suppliers, to prevent them from having too much power and possibly jeopardizing the supply lines.

Another cohort of commercial-off-the-shelf software were the smaller, more straightforward and well-defined software which had the characteristics that they were transferable across software domains and could be defined as ‘Non-critical Products’ The strategy required was efficient processing[86].

In conclusion, it can be stated that commercial-off-the-shelf software is not an easy commodity and should be treated with care and documented well.

## 2.6 SOFTWARE ASSET MANAGEMENT – From Excel sheets to automatic ordering

Software asset management is a subset of IT asset management.[87] The term used to describe a process in companies that ensure the “*legal and efficient purchase, maintenance[88], deployment, utilization, and disposal of software applications within an organization”[89]*. Software Asset Management influences all areas of a company, from procurement of every workstation to management of the assets. Those assets can be licenses, related services or even the digital or physical medium on which the software is delivered. The goal is not only to stay compliant but to manage the software further and optimize the process.[90]

Software Asset Management can be further divided into ***three*** cumulative subsets: [91]

**Software license compliance** [92] focuses on avoiding having too few procured licenses compared to the actual licenses in use (software deployments) and therefore being incompliant. In the End User Licence Agreement, more parameters can define compliance, such as the geo-location, configuration, internal/external usage and active or non-active usage. The company can ask itself if they are entitled to use the software if they have enough licenses, where it is deployed and by whom. Failing one of those makes the company noncompliant. It attempts to ensure that the likelihood and impact of a software audit are kept low and unbudgeted expense, legal and reputational risk are avoided.

[86] See Callagy (2007), p. 48
[87] See McCarthy and Herger (2011), pp. 561-562
[88] See Jakubička (2010), p. 1
[89] See Kim et al. (2014), p. 2
[90] See Jakubička (2012), p. 6
[91] See Irwin (2016), p. 1
[92] See Gangadharan, D’andrea, De Paoli, and Weiss (2012), p. 148; Irwin (2016), p. 2; Van Der Burg et al. (2014), pp. 731-732

**Software license management** [93] goes beyond software license compliance. It adds in the managing side of licenses, utilizing the right license for the right type and ensuring that no unnecessary licenses in type, quantity and function are used. As many organisations are over licensed in either quantity or functionality, massive cost savings can be achieved when managed correctly by for example move a software to a smaller server to ensure it’s licensed correctly and to avoid unnecessary direct costs and complexity while considering the possession of correct licenses, the reasonability of the management of the licenses and direct costs, as well as if the licenses are actually used and also optimally deployed.

**Software asset management** [94] goes even beyond the focus of licenses and the management of that, and also focusses on the software product (in nature, the asset), its function, currency, standardisation and the product lifecycle, from selecting, purchasing till possible renewal. Looking at reducing the technological complexity, lifecycle costs strategic failures while maximizing standardisation, the reaping of the benefits and supporting other IT functions. The considerations that a company has to take can be the search for a product roadmap, lifecycle cost consideration, an optimisation of the appropriate support, deployment/usage and version/alternatives of the current software. As well as standardizing versions, looking into functional redundant products and the funCompany Atal question if the right product is used. [95]

Software Asset Management helps to manage software licenses and the use of it.[96] But in order to manage something, one has to know what is there to manage. It starts with a catalogue of software that is used in the company with the appropriate license, and from there on, management is applied while looking at purchasing orders.[97] For many companies, this requires effort and a software package, and it has been seen in many companies that ‘Software Asset Management’ is done in excel, where licenses are kept track off.[98] The benefits of such a system can be *enormous*, automatic license management with a link to procurement. Bulk purchases, prevention of buying unnecessary licenses are only the beginning. For many companies, it comes down to limiting the effect that audits have on the firm, as during an audit the company is required to show that it is compliant. The software can be licensed in many ways; it can be per user, core, node, active user, business department etc.[99] Other forms are also possible, one can have a license for the development of a product, but not a product license or a testing license. Usually,

[93] See Moyle (2004), pp. 8, 51, 66 Irwin (2016), p. 3; Jakubička (2010), p. 1
[94] See Irwin (2016), pp. 3-5; Kim et al. (2014), p. 1
[95] See Irwin (2016), p. 4
[96] See Jakubička (2012), p. 5
[97] See Ben-Menachem (2008), pp. 241-242
[98] See Williams and O’Connor (2011), p. 2
[99] See Raman, Livny, and Solomon (2000), p. 290

testing or developmental licenses are cheaper or even free compared to product licenses. The danger with this is that contamination could happen. For this Software Asset Management is valuable, being able to keep track of the license and all the attributes of it.

Braun speaks of a commercial-off-the-shelf-software library and the organization of it. [100] This is a complete library of all the commercial-off-the-shelf software products in a company, which is automatically identified whereas parameters can be added into the bigger database. Parameters such as source code escrow, the effectiveness of the software, vendor liaison and support to users. The software in this system would be pre-approved and also available for other business parts to use.

Advantages are listed as easy to use and the possibility of cost savings due to planning. Also, requirement listings and costs to adapt the product are listed. As one product might have been commercial-off-the-shelf, but in order for it to be implemented into the product, it has to be adapted and this requires financial resources. [101]

Managing a database full of software is what Software Asset Management is about. [102]A database is only useful when it's accurate. To keep it accurate *maintenance* comes into place. Correct dataflow towards a database is of importance. One of the dataflows can come from the software reseller, which is a channel of obtaining updates and fixes of the software and the origin of the sourced software with knowledge of the product. Information like this can be the time it takes to fix problems, the failure rate, integration/adaption costs and experiences of the software including ease of use. Another source can be companies own administration, as software is being used somewhere, and it likely is registered at some place where it is located.

When enough data is put correctly into a database, the system can work and manage the assets. Software Asset Management maintenance[103] is not prioritized in many companies, and this is the reason why in many companies, Software Asset Management is being done in excel [104], where manually the number of licenses tracked[105]. An upgrade to this would be wandering agents who scan the systems for software and automatically adds those changes and found licenses to a database which it manages[106]

[100] See Braun (1999), p. 33
[101] See Braun (1999), p. 31
[102] See McCarthy and Herger (2011), p. 562
[103] See Merola (2006), p. 4
[104] See Ben-Menachem (2008), p. 244
[105] See Appendix K
[106] See Ben-Menachem (2008), p. 254

## 2.7 BUSINESS PROCESS IMPROVEMENT: Clearly defined roles and small steps

### 2.7.1 Having a clearly defined process, with core team that is willing to 'stop production'

Cooper identified the three cornerstones of successful product development, which are Process, strategy, and resources[107]. Having a high-quality new product process was the most significant one as for effect on profitability.

In the research this meant *"[...] an emphasis on* ***up-front homework****; sharp****, early product definition****; the voice of the customer evident throughout; tough go/kill* ***decision points****; a focus on* ***quality*** *of execution; and a thorough yet* ***flexible*** *process."* [108] [highlighted by the writer]. Further points of importance were, having the necessary people in place, and time freed up as well as both a cross-functional team and a defined and accountable leader in a team.

This means that for a company to function well, the software procurement process should be redefined in such a way that the homework is being done. This 'Homework' can be formulated into the introduction of products. Knowing what the product, market, manufacturer and financial situation is while clearly defining them in all its forms is part of that homework.[109] If the list of points is followed, the introduction of a new product must be done by a *cross-functional team* that is dedicated to the task while having dedicated knowledge about the subject. Thus, preventing people from slacking off for not having dedicated time to fulfil their process duties.[110] This will leave time for interaction and collaboration

Management plays a role in this as well, as one person needs to be accountable and manage the flow to make sure decision points are kept (in place). Those decision points have to be build into the process, where it prevents a project from getting too little oversight and the process becomes more like a *tunnel* rather than a *funnel*. Having the authority to stop a process if the requirements for a decision point are not met (thus giving it a 'kill'-decision), is therefore part of the essential managerial tasks.[111]

In the end, there is a remark that the process should be thorough, yet flexible. Indicating that the process should not be so inflexible that if a situation arrises where standard protocol is not sufficient, it will not be bypassed, but merely be adapted with managerial oversight. Being able to take a step back and re-evaluate the options and the process might prove to be benefitial, as a

[107] See Cooper (1996), p. 465
[108] See Cooper (1996), p. 465
[109] See Cooper (1996), pp. 468-469
[110] See Cooper (1996), p. 473
[111] See Cooper (1996), p. 471

new product process is merely a risk management model[112]. It is up to the mangers to asses that risk correctly and not dive inty bureaucracy too deeply.

Dedicated leadership is highlighted in the research; having a manager that is present throughout the entire time of a project helps with oversight while bringing in experience from the beginning of the project. It is suggested that the leader should be accountable from beginning to the end and has the responsibility when something goes wrong.[113]

Team membership is suggested as consisting of a small core group of responsible, committed and, accountable team players from different departments. Similarly, to the manager they should be present from start to finish of the project.[114] The rest of the team can be fluid, as work requirements can chance throughout the project.[115]

Having a focused core team, led by a leader that is present throughout the entire project that has a clearly defined process to follow and has the managerial leadership to stop the proceeding to a next step is what Cooper recommends.

### 2.7.2 BUSINESS PROCESS CHANGE: Change comes in little steps with involvement

With a new process comes change. Users of a process need not only follow the process, but also accept it. Resistance is regarded as the number one failure of new processes, more so than technical problems[116] To counter resistance, Salvendy [117] has compiled guidelines for reducing resistance which are supported with theories of other professors  (e.g. Kotter (1995))

1) Involvement of workers in the change. If the employees participate in the change and feel like the project is their own, they will cooperate.[118]

2) The top managers should support the change. If the employees feel like not everyone is committed, they will take it less seriously[119]

3) Let the employees see the advantage in the change.[120] This while keeping the worker's needs and values high. If the workers don't feel threatened and see an advantage, they will support the change.

[112] See Cooper (1996), p. 472
[113] See Cooper (1996), p. 474
[114] See Cooper (1996), p. 477
[115] See Cooper (1996), p. 477
[116] See Lea Hyer (1984), p. 201; Majchrzak (1988), pp. 248-250; Turnage (1990), p. 175
[117] See Salvendy (2001), p. 889
[118] Apsler and Sears (1968), p. 162
[119] See Kotter (1995), p. 60
[120] See Kotter (1995), p. 63

4) Free and open communication. When the employees feel like they are listened to and supported, there will be fewer misunderstandings and conflicts. [121]

5) Flexibility over bureaucracy. When not, everything is formally defined, and workers feel like there is space for improvements and reconsideration, resistance is reduced.

The lower the amount of extra work the employees must do to adopt the new process, the better. [122] No more than an extra 10% work is recommended as well as a transition phase that is as short as possible. [123] Effort, on the other hand, can mean both extra work, and different work that is to be done. It is therefore suggested to utilize as much of the existing process as possible.

## 2.8 SUMMARY OF THEORETICAL FRAMEWORK

The costs associated with purchasing materials take up between 50 and 80 per cent of a company and therefore has a big impact on its profitability (see **chapter 1.1).** If a company wants to be competitive, it needs to control purchasing. With a matured and structured process, it can perform better (see **chapter 2.2).** The most commonly used process is the one by Van Weele[124]. It starts with defining the specification's, then selecting the supplier, do the contracting, order the product and finally do the follow-up and evaluation. (see **chapter 2.1)** When focusing on the process of purchasing software, it becomes clear that the current processes are not optimized for software, as it encompasses a new dimension. Direct research is limited but correlates with the process popularized by van Weele. Although additional process attributes have to be found and implemented for software procurement, in order for the process to flow steadily while collecting all the necessary data (see **chapter 2.5).**

Software is very different from hardware. The cost of software is the development, not the production, and due to its intangible nature, it can be easily copied whereas those copies can be sold to regain the cost. Contrary to hardware, one does not own the product when one buys it. With a license, one possesses the right to use it in a certain way for a certain time. This license of these commercial-off-the-shelf-products comes with a numerous amount of conditions called the End User Licence Agreement and can be easily broken (**see chapter 2.3** and 2.4).

The usage of commercial-off-the-shelf-products have steadily increased through the years, as they offer companies a way to simply buy the required software, while not having to build it themselves. As mentioned before, this brings issues and requires management. (**chapter 2.5)**

Not only does a firm have to keep track that it is not using more licenses than bought, it is also responsible for the usage of it. A license could be bought per computer, core, virtual core, node, 100 users, or even per department. The complexity of managing this, and ease to be in violation,

[121] See Kotter (1995), p. 63

[122] See Sirkin, Keenan, and Jackson (2005), p. 11

[123] See Sirkin et al. (2005), p. 11

[124] See Van Weele (2010), p. 31

leads to software manufacturers making a lot of money by auditing firms for compliance. In order for a firm to limit non-compliancy, Software asset management is important Software Asset Management (see **chapter 2.6)** is the method of managing and keeping track of the software assets. This can take on different forms. Frequently used is the primary form of tracking licenses in Excel. A more advanced version is a dedicated software service that runs agents on the firm's servers to keep track of actual usages. (see **chapter 2.6).**
In order to improve a (software sourcing) process successfully, employees have to be involved in the change, showing them that the changes will bring prosperity while not bringing in more than 10% extra workload. (see **chapter 2.7.2).** A new process should be clearly defined and managed well, with kill-decisions in place that will help funnel, not tunnel a process while helping not to bypass it. Dedicated leadership with a core group of people present in the process is of importance, as well as doing all the homework for a new product (see **chapter 2.7.1)**

# 3 METHODS: Explorative qualitative research with semi-structured interviews with process modelling

## 3.1 PROBLEM ASSESSMENT: Interviews and Business process modelling

### 3.1.1 INTERVIEWS: Explorative Semi-structured to find the actual problem

To find the information for the current situation, the origin of the problem and possible solutions, interviews are conducted with employees of Thales. The purchasing processes of other companies (industry peers) were gathered and compared. The resellers were also be interviewed, as they had experience with selling software to many customers.

The guideline interview questions can be seen in APPENDIX KK[125]. The list of people interviewed can be found in APPENDIX Q[126].
Out of 160 interviews (150 of which originated from the employees of Thales and 10 from employees of industry peers/suppliers), a total of 44 interviews were found to be relevant and were transcribed into the Appendix, some are merged together. 21 interviews were conducted to compile the Antecedents listed in Table 6, and Table 8 and prior to everything 30 orientated explorative interviews were held to understand the context of the thesis, leaving 70 interviews out of 160 interviews not relevant enough for the thesis which were excluded.
Most (93%) of the 150 interviews from Thales came from Thales Huizen, whereas 8 came from Thales Hengelo and 3 came from Thales Paris. Due to the flexibility of the employees, there are several cases where employees work for both the Hengelo as well as the Huizen entity. They have been counted to the Huizen entity.

[125] See Appendix KK
[126] See Appendix Q

In the beginning of the research, a myriad of orientated interviews has been conducted to find the context for which the research was to be conducted and to obtain a holistic view of the situation and the people to interview. From the holistic view it became clear that software procurement and asset management was a complex theme and needed a wide area of industry experts. Solely looking at the purchasing department for information would not have given the report and its solution justice. A broad investigation into the entire supply chain of software was needed. Taking Thales Huizen as a big case study and adding additional smaller case studies for comparison would be sufficient to get an overview of the possibilities. Figure 2 gives a visualized overview of the methodology in answering the research questions, going from the current process, to the problems and risks of that process towards a renewed process, by utilizing not only interviews, but also industry insights and theoretical knowledge.

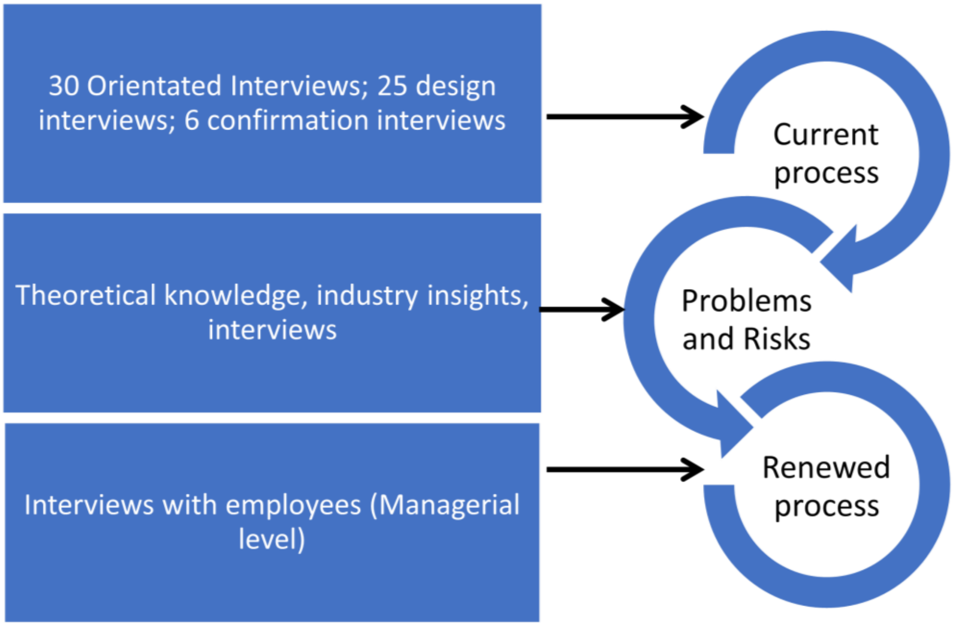

30 Orientated Interviews; 25 design interviews; 6 confirmation interviews
Theoretical knowledge, industry insights, interviews
Interviews with employees (Managerial level)
Current process
Problems and Risks
Renewed process

**Figure 2 Methodology of designing the new sourcing process with interviews and industry insights**

As it is usual with interviews in big companies, one might look for an answer but doesn't know the question to ask. The interview questions will thus be used as a guideline and occasionally guide the interviewee towards talking about the problems and solutions of Software Asset Management or software procurement.

Due to the nature of the industry (military, governmental contractor), severe limitations apply to the recording of formal interviews. Informal interviews are therefore conducted, and notes were made of the conversations in either Dutch or English, or a combination of the two, as some interviewees switched languages. While creating the notes, the interviewee was allowed to participate in note-taking and confirming the accuracy. As company rules had to be respected, no exact transcript was recorded, but the meaning and implication of each interview were checked by the interviewees. The list of employees interviewed can be found in **Error! Reference source not found.**, and a central overview of the outcome of each interview can be found in APPENDIX G.

### 3.1.2 BUSINESS PROCESS MODELING: Icons give a holistic view to understand complex problems

Peffers et al. define six steps for **design research:** *“problem identification and motivation, definition of the objectives for a solution, design and development, demonstration, evaluation, and communication.”* [127]

As this paper consists of orientated design research in a field, this methodology will be used. The search for a problem that is not entirely understood and finding a way to a working solution fit with the methodology that Peffers created, as it has to be used in practice by the people, it includes evaluation and communication, which can be considered as a feedback loop. With business process improvement research, having the theory aligned with praxis is of importance, as the new process must be usable.

For a modelled business process to be useful, it must be understandable. Therefore, standard symbols are used to describe specific steps in the process; the pictures came from the standard icons of Microsoft Visio, they are listed in **Error! Reference source not found.**.

## 3.2 MODELLING OF PROCESSES WITH THE HELP OF EMPLOYEES

### 3.2.1 SETUP: Let employees feel heard to solve the problem

In the paper, two figures were modelled; the software procurement process (current situation and proposed situation) and activities overview regarding Software Asset Management (current situation and proposed situation).

Due to the difference in nature of both and the focus of the research, the two items were modelled differently. As the procurement of software is - or should be a direct *complex* process with steps while Software Asset Management is more an activity, it was chosen that software procurement would be a modelled process with actors and arrows.

For Software Asset Management the importance is the laying out of a plan and strategy for a company, and due to the low maturity of the activities and freedom of the departments regarding the activities, a new activities model was proposed and modelled. This in order for the employees to quickly see what needs to be happening.

[127] See Peffers et al. (2007), p. 1

This was all line with the problem identification. The previous explorative interview or business process model was mentioned or shown in the next interview whereas employees could give feedback on it and this feedback could be immediately processed into changes on the process For change management to go as smooth as possible, as described by Salvendy [128], the design and problem identification were left to the employees. By applying a semi-structured interview method, it left space for employees to feel 'heard' and do their say with the current problem. [129] A set of framed questions didn't limit the discussion which was kept as open as possible while only minimally steering the theme towards software procurement and/or asset management. Meanwhile, this gave an opportunity to explore a complex problem with a broad, holistic view. As the employees that were interviewed would most likely utilize the new process, they should have a high say in it and give their opinion about the outcomes. According to the theory [130], there is less resistance if the employees see an advantage in the new process. When the employees have a big influence on the final product, *they* can mention a solution to problems they are facing, creating their own advantage. It is up to the interviewer to filter the relevant information into a case study and mould it into a solution.

In order to not miss relevant information from unlisted interviewees that the author does not know he does not know about, during most interviews, the interviewee was informally asked which people should be talked to next.

### 3.2.2 ATTRIBUTES: Finding new information by asking the experts

After several pre-interviews about the topic were held to uncover a holistic view about the situation, new attributes were to be found. In order to find the specific attributes for software procurement and asset management, a list of relevant stakeholders and players regarding the software sourcing process was compiled (see Appendix Q).

As these would be the people that would work with the process, it was assumed that they had in-depth knowledge of the inner workings of data and what data was required for a well-oiled process to lower the risks and be usable. It was essential to listen to these interviewees, especially when compiling lists of what needed to be registered into a renewed system. They can be formulated as attributed, which needed to be compiled.[131] The list was there started with Appendix W, where the main license manager of the transportation department of Thales elaborated on which data attribute points the license manager would like to have in such a system. With this data, short interviews were conducted with employees along the supply and management chain of software what they thought was vital information to have in such a system. As

[128] See Salvendy (2001), p. 889
[129] See Salvendy (2001), p. 889 ; Apsler and Sears (1968), p. 162
[130] See Salvendy (2001), p. 889 ; Apsler and Sears (1968), p. 162
[131] See Duke and Persia (1994), pp. 40-41

the supply chain started with the purchasing department doing its bidding, then the engineers working on the software embedding and finally having a product leave the premises where it passes configuration management and export control – this process direction was also to be taken into account when interviewing people and compiling the list of antecedents. First purchasers and their managers were asked, then engineers and their managers, as well as project managers and finally configuration mangers and export control were asked. The initial lists were passed on from interview to interview and expanded each and every time, as well as inspected and confirmed. This was to make sure no aspect was missing that could be of later use for auditing requests or management, as data is only valuable if it is complete. The departments in that order were keeping a check upon the works of the previous department during the supply chain, its therefore implied that this might also be true for a purchasing process of software and possibly beneficial for security.

Using a case study from Thales Huizen would not be sufficient to obtain a holistic view of the situation, therefore information streams also came from interviews of the industry peers. Initially, the thesis would focus more on the software resellers of the company and hear what they would advise, but the interviews did not turn out to be fruitful and were excluded.

Figure 3 visualizes the methodology of the compilation of antecedents of software sourcing and asset management in a step-up process.

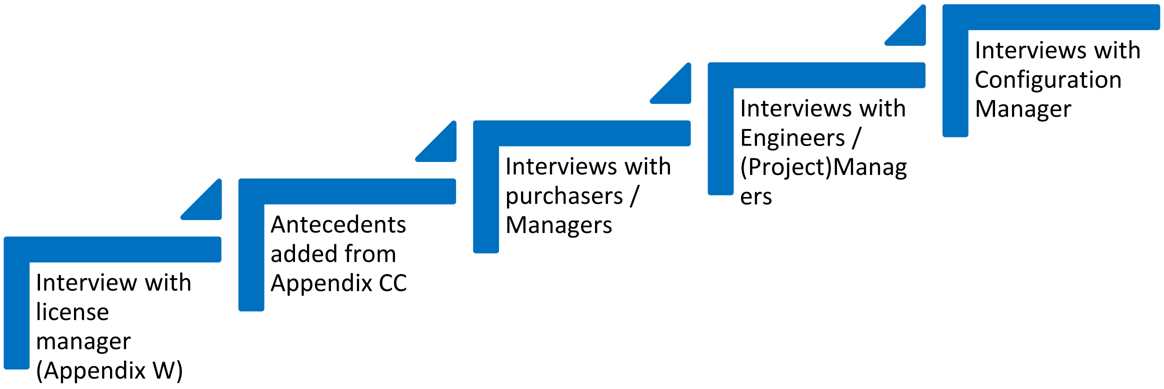

Interview with license manager (Appendix W)
Antecedents added from Appendix CC
Interviews with purchasers / Managers
Interviews with Engineers / (Project)Manag ers
Interviews with Configuration Manager

**Figure 3 Methodology of compilation of Software antecedents further used in the study**

After a large amount of data was gathered, it needed to be filtered and ordered. Utilizing the method of involving employees as much as possible, the interviewees were informally asked in which part of the solution the raw data should be included, steering towards a complete solution. This information was ultimately compiled in **Error! Reference source not found.** Table 6, and Table 8.To make sure the research method is reliable, validity and reliability has to be checked. In qualitative research this is equivalent to rigor and trustworthiness[132]. The threats of trustworthiness in relation to qualitative methods are reactivity, researchers bias, and respondents bias.[133] Those can be tackled with triangulation[134]. Which means that several research methods. are combined in a study about the same phenomenon to reduce bias and other reliability issues. E.g. interviews with workers, brainstorm sessions with mangers in order to map out, or map out and explain the complexity and richness of human behaviour by studying it from more than one standpoint. Which is what this study has done, by interviewing not only the employees regarding one centralized theme that has its roots in most of the departments (software procurement). As data triangulation [135] was done (different people and space) and theory triangulation, utilizing different theories regarding the attributes of software procurement. Combined with minimizing the bias due to not asking too many direct questions in the interview, the process has been designed to be iterative and collaborative, utilizing the workers to collaborate with the interviewer.

### 3.2.3 SOLUTION: asking employees for feedback while building a process

After several interviews, it became clear that several employees had already thought about solutions for the problems but felt not heard by their manager. In order to utilize this knowledge with a complex problem, the process was modelled in business process modelling and then remoulded bit by bit after asking the employees questions such as
*"How can we do it differently?"* or *"Look at the current process that I constructed, can I get your expert knowledge on it?"*.
Upon which employees frequently mentioned what could be improved and where the problems were.
The original process was therefore used as a basis of the new process, as the employees were to change their behaviour as little as possible. Change management is difficult for a firm, and the fewer changes needed to be done, the better employees would react to it. By specifically asking employees about the solutions and improvements for the process, expert knowledge and deep insight were used.

Due to this explorative nature, coding the interviews was not possible, and it had to be relied on building upon previously obtained knowledge.

[132] See Padgett (2016), p. 299
[133] See Padgett (2016), p. 304
[134] See Rothbauer (2008), p. 893
[135] See Denzin (2017), p. 350

## 3.3 SUMMARY OF METHODS

In order to find the current situation of the firm and the problems related to that. Interviews will be done in the firm and several industry peers. Due to the explorative nature, explorative semi-structured interviews are a solid choice. From those interviews, whose note was transformed into the case study after being checked by the interviewee, several were of explorative nature. Where business process management was used to visualize processes and receive immediate feedback from the employees. (See **chapter 3.1)**

By interviewing employees of the firm that even the smallest interaction with the software sourcing process, a holistic view of the situation was obtained. The new process was designed with the theory in mind of minimizing the change that employees were to go through, so the involvement of employees was high. In theory, the methods used would lead to less resistance in the firm (see **chapter 3.2.1** and theoretic **chapter 2.7.2)**

Using triangulation makes sure the data is valid and reliable. Attributes that were needed for the software sourcing process were directly sourced and criticised by numerous employees, as well as finding the best solutions. (see **chapter 3.2.2)** Although many employees were involved in designing the solution, it was up to the researcher to find the relevant bits and create a process that others would use and love. (see **chapter 3.2.3)**

# 4 RESULTS: Current situation analysis with solutions to problems and risks

## 4.1 CURRENT SITUATION OF SOFTWARE SOURCING AND MANAGEMENT: Thales and industry peer

<13 pages redacted due to confidentiality>

## 4.2 PROBLEMS AND RISKS WITH THE CURRENT SOFTWARE SOURCING PROCESS: No process or protocol (followed)

### 4.2.1 PROBLEMS AND RISKS ASSOCIATED WITH THE INTRODUCTION OF SOFTWARE: Bypassing can lead to losing track and audit fines

<9 pages redacted due to confidentiality>

## 4.3 REQUIREMENTS AND PROTOCOLS OF SOURCING COTS SOFTWARE

### 4.3.1 DEVELOPING A NEW PURCHASING PROCESS WITH SOFTWARE ATTRIBUTES

When looking at the purchasing process of Van Weele, one can see that the process starts with the “definition of the specifications” of the product or service.[136] As this concerns a ‘new task’ the full investigation shall be done. With Software Asset Management in hindsight, and the problems from Software Asset Management were previously that the data isn’t in the system - it is advised to fill in the data right into the management system which is to be described in chapter 4.3.4. It is an opportunity to collect the data right away and keep it central so it can be managed and be of use.

As one can see from the previous sub-question, the specifications for software are not clear and should be investigated. This was done by asking the employees of Thales and industry peers the following question:

*“What data should be necessary to know when dealing with or managing the procurement of software?”*

Before each short interview, the previous answers were shared with the employee where he was asked which the information was compiled into a big table and sorted. The interviewees were also asked to look at the previous answers and decide if they thought they were relevant or should be deleted. During the process, none of the interviewees mentioned that any previous answers were not good and only confirmed them. The sorting of the attributes, which would become data entrees somewhere in the new purchasing process was done by the relevant departments.

As there are several sides to software (chapter 2.5), a process will have to look at every side of it — namely the technical, legal, commercial, accounting, ‘alive’ and exportable side. When considering the specifications of software that chapter 2.3 compiled, one can compile a list of linking them together to get a basic understanding of what needs to be understood. Utilizing that as a semi-structured basis for the short interviews gives them an excellent basis to proceed with general knowledge about the further side of software regulation.

**Table 4 Software characteristics and their corresponding attributes, constructed from Abts et al. 2000 [137] and own elaboration**

| *Side* | *Attributes* |
|---|---|
| **Technical** | Correctness, functionality, version compatibility, maturity, performance |
| **Legal/exportable** | Vendor concessions, flexibility, security |
| **Commercial** | Price aspects, training |
| **Living** | Upgrade regulations, Intercomponent Compatibility, |

For the technical side, it is vital to look if the software does what it is supposed to do, and will do so in the future with newer versions and right now with version capabilities.
The legal side looks at what is written in the End User Licence Agreement, e.g. can the re-seller/manufacturer have access to the software to ensure it stays compliant, or if this software be used where one is planning on using it in that specific country.
The commercial side looks at the price and tax regulations and what is included. It can happen that training of the software is included in the price.
The 'living' side looks at what will happen in the future. Software is implemented in the beginning, but it will change over time. Free updates, future changed that might affect interconnecting software are but a few of the examples.

#### 4.3.1.1 USERS: High utilization of internal experts

Van Weele discusses the users [138] who will partake in the purchasing process. For the renewed process, users will have dedicated tasks that they will need to fulfil. When taking the standard rules of the process and taking them into the context of the purchasing process, one can state that there is one crucial aspect missing.

**COTS-Board:** In Thales as well as in Company A shipyards, there is an internal organization that concern itself with obtaining expert knowledge about commercial-off-the-shelf software and advising the firm about this. In this thesis, this will be named 'COTS-Board'. Due to the complexity of software in contrast to hardware, it is of logic to let an organization play a role in it. As they are currently influencing the procurement, they will be placed under the 'influencers. Letting an expert organization run the contact with the suppliers compared to a purchasing department that generally does not know too much about software might be a wise decision. The  COTS-board falls in line with the first step of Al-Mahmood and Al A'ali (2011), p. 296's [139] process of creating a team of experts.

**User**: The people that will use the software will be the **engineers** as they will implement the software and use it directly or indirectly

**Influencers**: COTS-Board supplier development, can help with this, but let it run through the COTS-Board for increased accurate data flow, and they have the experience with this and can pick out Biases, they are experts, and experts amongst each other are more effective than a purchaser talking with a supplier

**Buyers**:  The purchasing department, the purchaser with the most IT skills should handle this.

[136] See Van Weele (2010), p. 287
[137] See Abts et al. (2000. p 2)
[138] See Van Weele (2010), p. 28
[139] See Al-Mahmood and Al A'ali (2011), p. 296

**Decision makers:** The budget maker in combination with management

**Gatekeeper**: A person with in-depth knowledge about software who will oversee the department.

### 4.3.2 INTRODUCTION OF SOFTWARE: Compiling the attributed

The sourcing process starts with the introduction of software; this is in a normal production environment triggered by the need for software by the Research and Development department. In Appendix Z this is described as “D” – for “Development”.

In Van Weele (2010) one finds that there are technical and commercial specifications at the first step of his purchasing process[140]. As this renewed purchasing process is aiming to adapt to the current process of Van Weele (2010) towards software, the detailed specifications for software need to be gathered for those two items.

Before those items should be discussed, a **pre-check** should be done with software.

This is a check where only the known problems with the software are checked.

Filling in all the specifications and later find that one is not allowed to use this software in this setting would be very time-consuming. A simple check by the COTS-Board will be enough. The knowledge and experience of the Board will be helpful with figuring out what to look for. Generally, the question that will have to be answered once a developing department wishes a certain software is:

**Immediate concerns**

**Table 5 Immediate concerns question**

| |
|---|
| *“Does this particular software that is going to be used for this purpose have any immediate concerns?”* |

A much more extended example form is included in APPENDIX SS`

When no immediate concerns are posed for the software, and the ‘Testing’ department wants to proceed in obtaining this software, the next step is obtaining the commercial end technical specifications. Compiled they are as follow:

[140] See Van Weele (2010), p. 9

**Commercial specifications:**

| *Name of item* | **Example** | **Person to mainly fill it in** |
|---|---|---|
| **Manufacturer** | Hewlett Packard | Software engineer support with knowledge of Purchasing |
| **Manufacturers product code** | HPN03281953-ADV | ‘’ |
| **Name** | HP Network node manager 3.1.65 | ‘’ |
| **Open source/Commercial-off-the-shelf?** | Commercial-off-the-shelf | ‘’ |
| **License type** | Per server, per node, per core, per user, Unlimited, per year, Freeware etc. | ‘’ |
| **Risk of fines** | Big fines in the past for American airlines | Legal expert |
| **Manufacturer right to verify actual licenses in use** | Yes/No/Monitoring tool | Legal expert; Software engineer with knowledge of purchasing |
| **Export control** | “Not export controlled”, “Only allowed in non-military applications”<br>‘Origin from Japan, not to be used for military purposes.’ | Export control |
| **Notes** | “Watch out, buy in packs of 40”, “Little support, but at the time the only supplier, look for alternatives”, | Export; Purchasing; Whoever has information |

The table also lists who is recommended for the task of gathering the specific data.

*A much more detailed version of the table is centred in* APPENDIX TT *where an example form has been created*.

The question remains on whom will fill it out, as this was a problem in the past.
From the interview in APPENDIX U[141] one finds that knowledge of all the subjects is not with one person. It has to be spread out and given to the field experts. When software is being requested from management, the new ‘Introduction of a product’ will be sent to a committee that has to fill in the information. This committee will consist of experts in their specific field. This can be different for any firm, but generally, it would have an IT-purchaser, someone from export control, a general software engineer, a legal expert, a manager and the Software asset manager. In the end, there is the Gatekeeper to keep everything in check. This list was compiled by looking at the people who would need to fill in the data of the new introduction of software and finding the right person to fill in the information**. For every data entrée, it will be a team**

[141] APPENDIX U

**effort**, but there will be one person who is responsible for filling it in, thus minimising the limitations on people with valuable knowledge.

In order to complete the introduction of a new product, documents are required, one of which is the product description. This was selected for a change due to the complex nature of software and the difference in hardware.

**Technical specification** includes, but are not limited to:

| *Name of item* | **Example** | **Person to mainly fill it in** |
|---|---|---|
| **Support YES/NO** | Yes/No | Software engineer support with knowledge of Purchasing |
| **Support type** | 24/7 or 9/5, Mandatory payment/ Training is paid for | Software engineer support with knowledge of Purchasing |
| **Purpose** | Monitoring software/ Telnet / Virtual machine etc. | Manager |
| **Updates/Evolution** | Included are updates for 10 years; | Software engineer support with knowledge of Purchasing |
| **System interaction** | The software interacts with software x. | ‘’ |
| **Installation manual** | First, install software, then add license via the local website | ‘’ |
| **End User Licence Agreement** | Copy of the End user license agreement | ‘’ |
| **Functionality** | Can create databases out of Excel sheets | ‘’ |

**Table 6 Technical specifications of software required for the “Technical requirements”-document[142]**

| | | |
|---|---|---|
| **Operating system** | Only MS/Linux distribution x/OSX; Non-compatible with Windows below 8.1 | ‘’ |

*A much more detailed version of the table is centred in* APPENDIX UU *where an example form has been created*.

The gatekeeper has to control if the information is correct and his (online) signature will lead to acceptance of the introduction of the product.

After acceptance, the software will get an article code, and the item will be available in their configuration management system.

**Table 7 Example of an article code of a large technology company**

| | | |
|---|---|---|
| Article code | **12NC-TNL** | ASDF1234546809 |

As it now has an article code, it can be purchased.

[142] See *Appendix W, further interviews and own elaboration*

### 4.3.3 PURCHASING SOFTWARE: Developing a new process

#### 4.3.3.1 PROCUREMENT OF SOFTWARE: Gathering relevant logistical data

Now that a software item is in the configuration management system, it can be ordered.

In the development, testing, acceptance and Production. (DTAP)-Environment, the ACCEPTANCE phase can only have parts in it that have been configured. Once managerial approval is achieved for purchasing the software, there is a will and a budget to purchase the item. Previously this would happen on a 'Witte Bon' without proper documentation. With Software Asset Management in hindsight, it is wise to gather as much data as possible by the right people. As learned from the past, the expert knowledge is not always with Purchasing, therefore for the pure procurement of software regarding the Logistical data to not miss essential items. Cooperation with the COTS-Board appears to be reasonable.

There are many more logistical attributes for software than for hardware as it is of relevance whether or not the software is going to be used for own usage or somewhere in a foreign country for another company. There are many aspects that need to be covered by a document; Table 8 gives an attempt to cover the relevant factors as a document that was compiled when asking employees what logistical data needs to be known for a system to work well. [Elaboration by author]

**Logistical data:**

| *Item* | **Example** | **Reason for relevance** |
|---|---|---|
| **Purchasing order Number** | PO235098235089 | Standard purchasing data |
| **Purchasing order** | Attached Purchasing order | ‘’ |
| **Cost per license** | 237 Euro | ‘’ |
| **Date of purchase** | 03-03-2018 | ‘’ |
| **Licenses bought** | 10 licenses | ‘’ |
| **Quotation/tender** | Document of Tender | The quote from the reseller is relevant for registrational issues prevention. |
| **Contract** | “We are allowed to pass the software on to the customer, but only after 6 months.” | **“Purchasing document”** The contractual limitations can include anything in the flexible software license world. It is relevant to know exactly what one can and cannot do with the product. And include the entire contract into the configuration management; especially if one decides to reuse a license for another department. |

| | | |
|---|---|---|
| **Proof of Purchase** | Receipt | As a way of gathering information to cover for audits, the receipt has to be collected. |
| **Start of License** | 01-01-2019/ When in use | Hardware doesn’t have a starting date. A software can have it, and it is relevant to know from what moment on the software is usable. |
| **End of License** | 03-03-2036/Unlimited/x years | Hardware doesn’t have an end date; software licenses frequently have an end date. This date is relevant because from that moment on one can no longer use the license and it has to be renewed. |
| **Support** | Only bugfixes, the next version will cost money | Not only the license will be bought, but frequently support is also included and should be registered as such. |
| **End of support** | 03-03-2023 | Frequently a license is procured with additional support for failure or updates; it is essential to know the exact date when the support ends. |
| **Own use/**<br>**Customer use** | Customer use | It is relevant to know for what purpose the software is for, on a macro level. Frequently there are different regulations for reselling of software than for own use; it should, therefore, be written down. |

**Table 8 Logistical data for the renewed software purchasing process. [143]**

| | | |
|---|---|---|
| **For Project …** | TNL-Higher purpose to be specified if more than one. | Similarly, to the *“Own use/customer use*”, the ‘*For Project…”* concerns the detailed purpose of the software, but then at a micro level. It is relevant for contractual reasons, yet also for administrative purposes, to keep track of which software is used in which project. |

*A more detailed version of the table is centred in* APPENDIX WW *where an example form has been created*.

This information will go through the ERP-system towards the later-described software asset management.

#### 4.3.3.2 SUPPLIER DEVELOPMENT: Supplier has expert knowledge

The process has so far relied on internal experts of certain products or systems. There is also a possibility to utilize the supplier, as they are usually an expert on their product. In the world of software procurement, software is usually resold by a reseller. As seen in the interviews, they were more than willing to advise the customer on the product, but this poses risks. Having a biased supplier that is leaning towards a particular software company could advise a purchaser that does not have enough knowledge about the subject to form a critical opinion about this, might pose a danger. The information is there and should be utilized, but instead of utilizing the purchasing department, the COTS-Board should play a more important role in this communication. As it consists of experts with relevant knowledge about the subject which might not be easily persuaded into a product that might be for example more expensive than another product.

Once this is filled in, and the contracts are signed, the software licenses are sent out to the customer.

[143] Interviews (W), further interviews and own elaboration

#### 4.3.3.3 MODIFIED REBUY, STRAIGHT REBUY: Easier with a clear process and readily available data

The previous chapters incorporated the process of buying a new product that has not yet been sourced in the firm. As discussed in the literature, there are also possibilities to buy a slightly adapted version of the product, or simply buy the same product again. With a unified system, this will become easier as the access to relevant data is available.

For **Straight rebuy,** the authorized purchaser can rebuy the product in the ERP-system, as the previous contract, conditions and contact of the supplier are readily available.

It becomes slightly more complicated for a **Modified Rebuy**. As discussed, the same software can be bought in a myriad of different licence constructions and packages and configurations. The general information of the software is already in the ERP-system and can be easily cloned for an ‘adapted version’ of the product, with a different license structure. New negotiations with the supplier regarding the prices for a different license structure are most likely going to happen, but this can be added to the database.

Further on, new projects that share configuration of older projects can be sourced easier, as the data of parts of the project will already be available to use.

#### 4.3.3.4 SOFTWARE LICENSE DELIVERY: Unification of delivery and retrieval method

Once a software is bought, it is essential to have a unified way of retrieval. If the customer can’t keep track of the licenses nor find them after time has passed. Thales once had the idea of utilizing one single E-mail address for the retrieval of licenses for a certain department. This appears to be a simple yet effective method to start collecting the licenses (once implemented into the daily workings of the company). As they arrive in a myriad of ways, per document, are downloadable over the manufacturer’s website or come in stickers, it can all be initially managed in a secured email inbox. For further management, Software Asset Management is a viable solution that will be discussed in chapter 4.3.4 and 4.3.5.

### 4.3.4 MANAGING SOFTWARE ASSETS: Intranet, internet connected or off the grid products require a different approach

Incorporating Software Asset Management, there can logically be three types of products sold that could incorporate software. These were also the types that were found in the case studies.

Either an item is internet connected and can run a Software Asset Management-agent, or an item is not internet connected and won’t be able to run an agent. A third category comes up with (governmental) security, an intranet connected product that is not reachable on the normal internet but remains in a closed (secure) environment. The agent can detect code-tags in software to identify which software it is and which license fits to it.[144]

**Table 9 Types of products to be managed by Software Asset Management[145]**

| *Products to manage in Software Asset Management* | **Description** |
|---|---|
| **Off-the-grid products** | These are items that are not reachable over the internet, an example for Thales would be the communication boxes in tanks which will be in some desert. |
| **Internet connected Products** | These products can be retrieved over the internet and have the possibility to run a Software Asset Management agent to scan the systems for management. |
| **Intranet connected products** | Agents can run on this, but can't report back to a central command server |

A Software Asset Management agent is a software module that can be loaded on a computer to scan the usage of software[146], it can count how many licenses have been used or how many at the same time. This is reported back to a central server.

The **problems** regarding the current situation were that the current software management solutions were not central and there was no central overview of an inventory. It could therefore not be managed.

In order to cope with this problem, the solution would, therefore, need to be centralized, while being able to accept all the current databases

[144] See Ahmed et al. (2007), p. 1

[145] See **Error! Reference source not found.**, **Error! Reference source not found.**, **Error! Reference source not found.**, **Error! Reference source not found.**, and own elaboration

[146] See ServiceNow (2019), p. 1

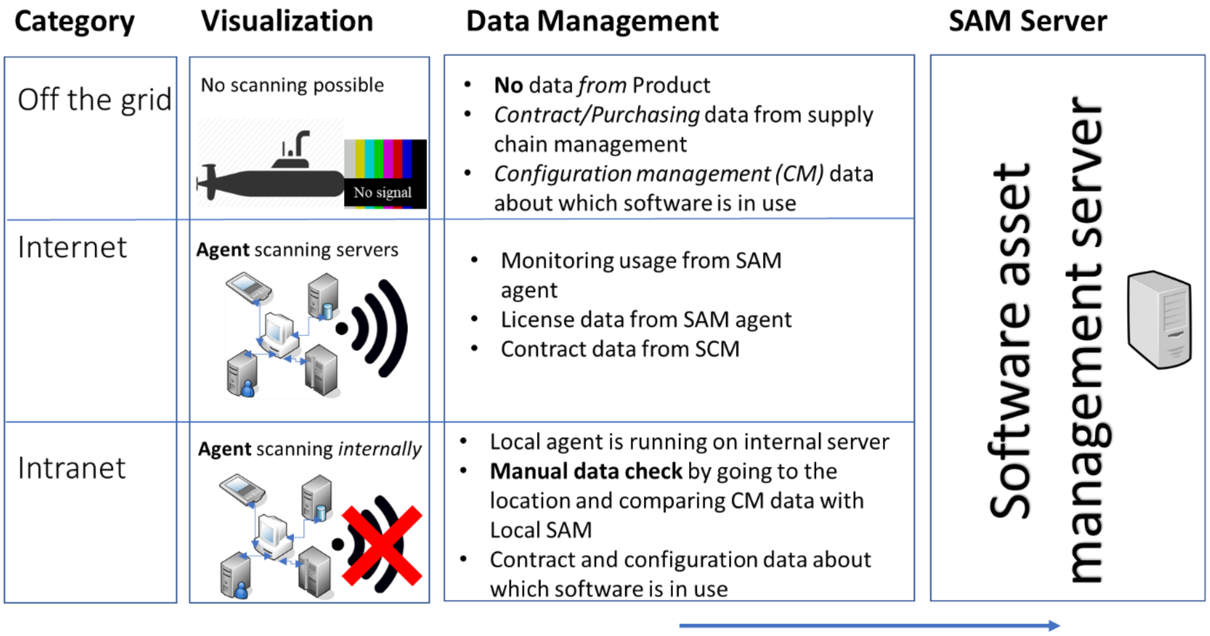
- For **off-the-grid-products, t**here is no possibility to install an agent on and reporting back to a central server. Because the producer can’t keep track of the item, a lower control over the exact data is also sufficient. Utilizing the purchasing data from the ERP system should be enough to satisfy the auditor and keep track of the product.

- For **internet-connected products** this is a matter of installing an agent on the product and letting it send back information which stocks up the databases to further manage it.

- **Intranet connected products** are connected on an internal network at either the customer or the firm itself. It is not possible or allowed to be connected to ‘the internet’ to transfer agent data over towards Software Asset Management. What is frequently possible is to run a local Software Asset Management agent to create a local centralized database on the intranet and manually comparing it to the Software Asset Management database that is constructed of configuration management and Purchasing data.

All three categories have ways of getting information towards the Software Asset Management server. Yet data does not always have to be generated on the locations; it can also come from a myriad of other sources. Utilizing contracts and configuration management data that describe what software is used in which product will give a valuable contribution to a working Software asset management system.
The further advantage of having this data available is that it allows for live feedback and a check if what is written to have been installed on that site, corresponds to what is currently running on that site.

Centralized information leads to efficient management of the data with its benefits. The three categories with a centralized Software Asset Management server are visualized in Figure 4.



| Category | Visualization | Data Management | SAM Server |
|---|---|---|---|
| Off the grid | No scanning possible<br>No signal | • **No** data *from* Product<br>• *Contract/Purchasing* data from supply chain management<br>• *Configuration management (CM)* data about which software is in use | Software asset management server |
| Internet | **Agent** scanning servers | • Monitoring usage from SAM agent<br>• License data from SAM agent<br>• Contract data from SCM | |
| Intranet | **Agent** scanning *internally* | • Local agent is running on internal server<br>• **Manual data check** by going to the location and comparing CM data with Local SAM<br>• Contract and configuration data about which software is in use | |

**Figure 4 Visualization of proposed Software Asset Management (SAM) data transfer for different categories**[147]

### 4.3.5 HOW TO MANAGE SOFTWARE ASSET MANAGEMENT: Utilizing a centralized system that can accept the current databases

#### 4.3.5.1 EASE OF USE OF SOFTWARE ASSET MANAGEMENT: Linking databases makes for easy management

As mentioned in chapter 2.6, many companies still use an Excel sheet or similar spreadsheet or word-processing programs as 'Software Asset Management' It is seen as an easy-to-use 'system' that has been used for many years.

Phrases such as *"But we have always done it this way"*[148] indicate a lack of willingness to change.

[147] Own elaboration and Creative Commons icons

[148] See APPENDIX R, as well as APPENDIX B

A system that consists of a better way of doing it while maintaining a likeness of the old ‘system’ is therefore properly advised. Taking into account the theory that one should not increase workload by more than 10%[149] and the theory[150] of resistance to change, it is advisable to use what one already has and implement that towards a working system.
Regarding the actual products that are to manage the following can be regarded;
If *the company* has access to it, so can an auditor. If the company doesn’t have access to it, neither does an auditor.
In practice, this means that if a product is unreachable for Software asset management, it is therefore no problem if only the configuration data is collected.

In the case study, it has been seen that Thales has many different databases where information is stored. From excel sheets, Configuration management on current systems, data dumps and more. For software asset management, it is important to use as much data as is available and centralize this data into a central system. When choosing a solution, possibilities of linking the current systems and databases with the new Software Asset Management should be thought of.

#### 4.3.5.2 USERS: Introducing the software asset manager

In the interviews, it became clear that it wasn’t clear who was tasked with the management of software licenses while it relied on employees to keep track themselves.
In an organized process the tasks have to be clearly defined; therefore someone should have the responsibility that the data is transferred to the central system. For, if no data reaches the central system, the central system is useless.
The most logic person to be in charge of this is the person with a holistic view of a project. Namely the project manager of the project where the data should be flowing from.
This way there is a person that is responsible and already available while leaving open the possibilities of scaling up by adding more departments to Software Asset Management.
Instead of filling in Software Asset Management data in Excel sheets, the managers will now fill it in in an excel-like environment in Software Asset Management.
As not every manager is knowledgeable of Software Asset Management, and there needs to be general oversight, a manager will be appointed that is responsible for the software asset database and system – the **software asset management.** He or she is responsible for accurate and up to date data on the system. The **local managers** of the projects will report to him with data. As visualized in **Figure 5**

[149] See Sirkin et al. (2005), p. 98
[150] See Salvendy (2001), p. 889

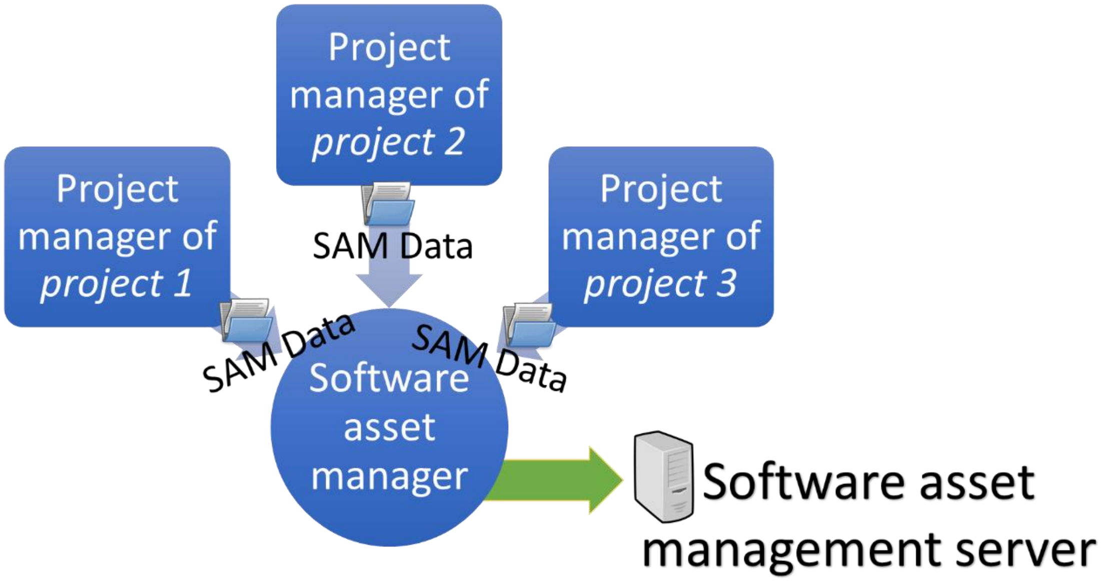

Project manager of *project 2*
Project manager of *project 1*
Project manager of *project 3*
SAM Data
SAM Data
SAM Data
Software asset manager
Software asset management server

**Figure 5 Overview of users in Software Asset Management: Managers and software asset managers[151]**

#### 4.3.5.3 POSSIBILITIES OF Software Asset Management: Automatic ordering and management

Now that the data is in the system, it can be managed. It will depend on the software what kind of data will be in the system, but generally, it will refer to the license structure and the current stock of software. For a certain software, it will be known how many licenses were bought at what time and how many are currently in use for which projects. Also known is when the licenses are expiring.

With this data one can optimize software purchasing, one can add reminders when licenses run out or even automatically order new licenses when the time comes. Further possibilities include stopping licenses from working if the maximum number of users had been reached and any further user leads to non-compliance. Data gathered by an agent will be the most accurate and can even provide insights in the daily usage of a software, which is data that is valuable for an auditor.

Another possibility is the bundling of purchases. There is an overview of the current software in use amongst departments, most of which needs to be renewed at some point in the future. If planned carefully, the purchase can be bundled and money can be saved. Going even further, a

[151] Own elaboration and creative commons icons

central overview of the software in use might help unify the software portfolio and make sure different departments of a company are using the same software for similar problems.

### 4.3.6 AUDIT POSSIBILITIES: Outsourcing or AUDIT-protocol

#### 4.3.6.1 OUTSOURCING SOFTWARE ASSET MANGEMENT: Supplier development comes with a price

As can be seen with the case of NS, it is also possible to outsource Software Asset Management to the software reseller.

With direct interviews from NS, they initially had their Software Asset Management outsourced. For a company that does not have the capacity to manage software on its own or does not want to deal with the risks, outsourcing Software Asset Management can be a viable, yet expensive option. The interviewed did disclose the possibilities of gradual out phasing of outsourced Software Asset Management if the maturity of the purchasing department and software management 'department' had increased, as they could retake certain task and save money while making gradual improvements. If the improvements did not turn out great, then scaling back is also a possibility.

#### 4.3.6.2 SURVIVING AN AUDIT: Protecting oneself with a SAM-Team and house rules

The gathered data so far have been to improve the data flow and gathering, leading towards a manageable system that is able to present itself to an auditor in the case of an audit.

In the interviewed firm there were no formal protocols in place regarding what to do when an audit comes. If one knows what an audit will look like, one can prepare itself for the audit and prevent risks.

From the interview of NS [152] one finds that an auditor will always find something. It is therefore important to protect itself as much as possible. Audits happen for four main reasons.

1) A software manufacturer *wants more money*, so they look at their customers.
2) At *random*, with a received list of companies to audit.
3) After a *company transfer.*
4) *Treason*; An (project) employee leaves the company and spreads the word that things aren't managed well with software.

With these reasons in mind, an audit can happen at any time. To minimize the impact, a protocol with house rules should be in place to be prepared.

NS gives a view on that, as they only allow the auditor to speak to the software asset manage-

[152] Appendix CC

ment team, and no one else. Those are the employees that are responsible for the software asset management of the firm.

They also show a presentation where they present the following house rules:

- Random Audits shall only be done by **an independent 3rd party** audit office. Not the ones that are paid by the software manufacturer.
- **All communication** shall happen with the **SAM-team**, no one else shall be questioned. If it doesn’t go via Software Asset Management, the Audit will stop immediately. If there is no SAM-team, it should be created ad-hoc.
- The business departments (infra, admin) collect their own required information to be delivered to the auditor via the SAM-team

*Next come two phases for the audited company*

*Phase 1* starts with the data gathering, as presented in the presentation. This will lead to a report by the SAM-team that is to be presented

*Phase 2* is the commercial negotiation which is the responsibility of purchasing. The SAM-team keeps out of it.

In general, there are 2 options to do Software Asset Management. The first one is to not manage anything and just move over to Phase 2 where one paid money and keeps on going, and the second option is to record as much information as possible and adjust the systems, preventing high fines from Non-compliance. Leaving expert teams over to internal expert might prove suitable for companies.

**Minimal required data**

It is essential to have as much data available, but there is insight on what the minimal data is that needs to be collected in order to survive an audit. Which is the **Proof of purchase**, the **Request for quotation**, the **purchasing order** and the **invoice** according to the interview in Appendix CC (NS).

## 4.4 NEW SOURCING PROCESS: Providing an overview of the new process.

### 4.4.1 NEW PROCESS: Software fitted introduction, purchasing and management while utilizing current processes for limited change management.

In chapter 4.3, the protocols and documents regarding the new process were researched and written down. It is now in this chapter that they are brought together into a full pragmatic system for an overview to be used for managerial purposes. To keep the size under control, and the chapter structured, explanations and compilations of documents that were done in the past will be referred to. Screenshots from figure 12 will be used throughout the text., To prevent unnecessary confusion, they are not named as figures and are merely regarded as "In-text-shapes".

In Figure 6, the renewed software sourcing process is summarized, which will be used for describing the steps in the process. In the figure, the is visualized from the beginning till the end. Arrows are going from phases to actors in the process, all the document icons are documents that need to be filled in, in order to continue. There are hard stops between every step, with the **gatekeeper** making sure that no step is (partially) bypassed.

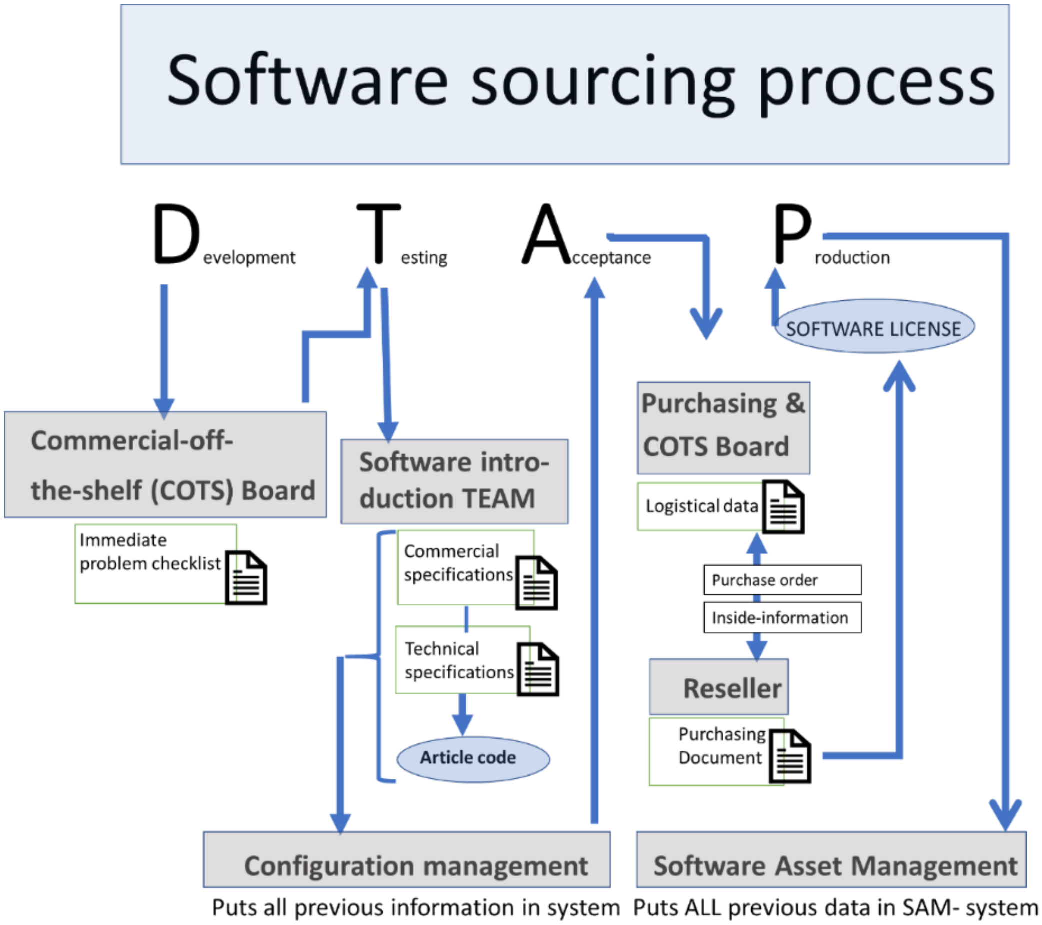

Software sourcing process
Development
Testing
Acceptance
Production
SOFTWARE LICENSE
Commercial-off-the-shelf (COTS) Board
Immediate problem checklist
Software intro-duction TEAM
Commercial specifications
Technical specifications
Article code
Purchasing & COTS Board
Logistical data
Purchase order
Inside-information
Reseller
Purchasing Document
Configuration management
Puts all previous information in system
Software Asset Management
Puts ALL previous data in SAM- system

**Figure 6 Renewed software sourcing process (simplified)**

The full version of the business process can be seen in **APPENDIX PP** while a miniature version is shown in Figure 7.

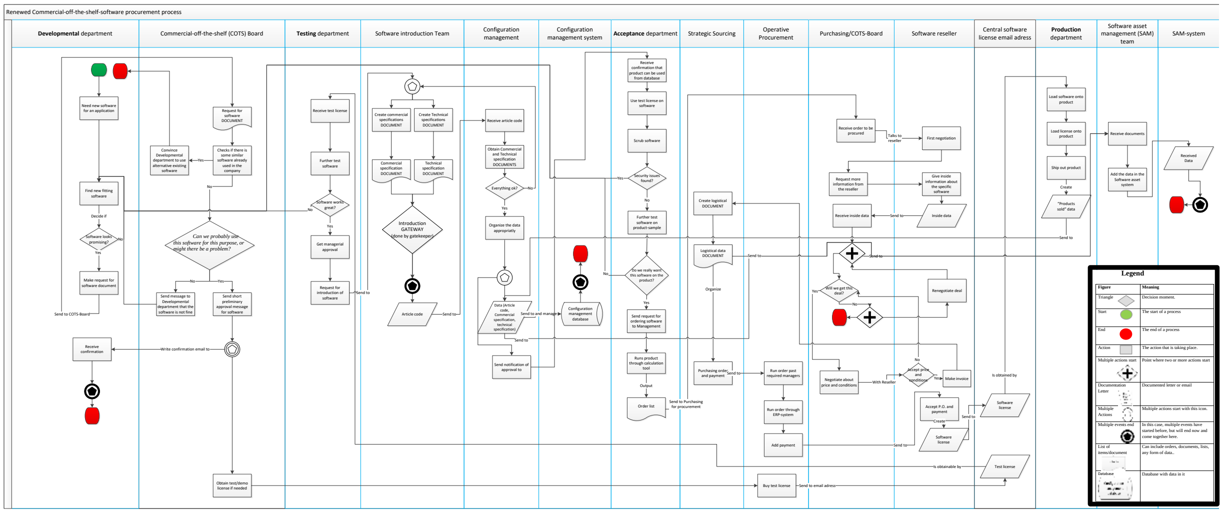
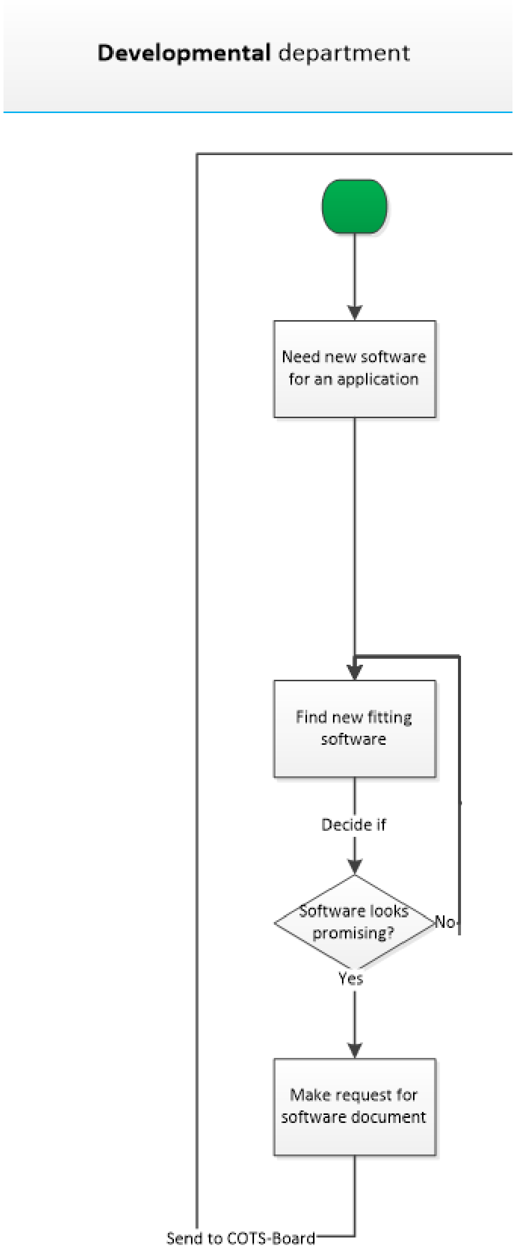

Renewed Commercial-off-the-shelf-software procurement process
Developmental department
Commercial-off-the-shelf (COTS) Board
Testing department
Software introduction Team
Configuration management
Configuration management system
Acceptance department
Strategic Sourcing
Operative Procurement
Purchasing/COTS-Board
Software reseller
Central software license email adress
Production department
Software asset management (SAM) team
SAM-system
Legend
Figure
Meaning
Triangle
Decision moment.
Start
The start of a process
End
The end of a process
Action
The action that is taking place.
Multiple actions start
Point where two or more actions start
Documentation Letter
Documented letter or email
Multiple Actions
Multiple actions start with this icon.
Multiple events end
In this case, multiple events have started before, but will end now and come together here.
List of items/document
Can include orders, documents, lists, any form of data..
Database
Database with data in it

**Figure 7 Renewed software procurement process for Commercial-Off-The-Shelf-Software**

It begins with the development of a **new product,** where there are normally **four** phases, *Development*, *Testing*, *Acceptance* and *Production*.  The **fifth phase** is the management of the further data.

1. Development

For a project, the engineers or the customer decide that they a new software with specific requirements. After looking around on conferences and the internet, they will download some software and experiment around, at this stage they will have all the freedom to test what they want. But once the software becomes a viable option for the solution, they will report to the **Commercial-off-the shelf board (COTS-Board)** if the software has not been ordered before.


Developmental department
Need new software for an application
Find new fitting software
Decide if
Software looks promising?
No
Yes
Make request for software document
Send to COTS-Board

This will go via a document that is given as an example in

APPENDIX RR (Request for introduction of software)

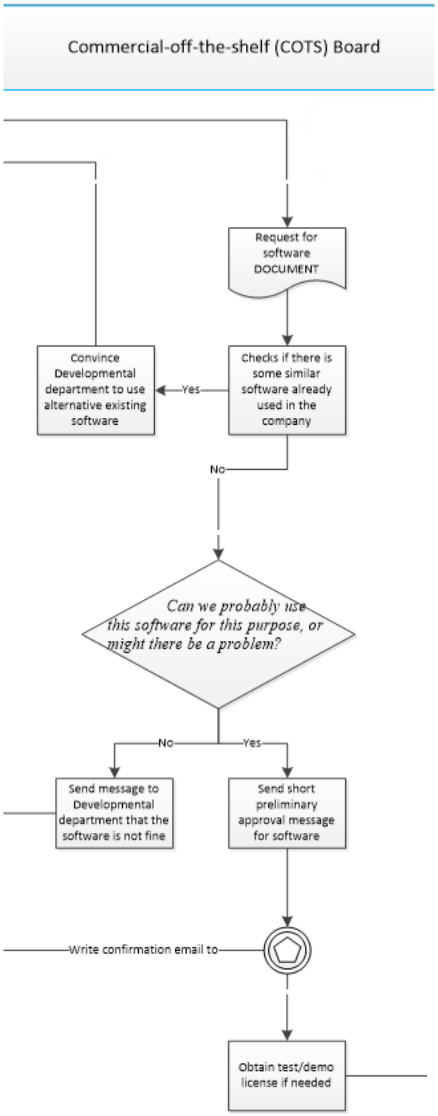
The newly introduced **Commercial-off-the-shelf-Board**, which is further described in **chapter 4.3.1.1** as derived from **chapter** Error! Reference source not found.**,** will run an immediate problem checklist on the software, where a group of experts in their field will try to identify the potential dangers of using this software in this setting. As every project and product is going to be different, the expertise and experience of the group will have to be used to create an immediate checklist what is relevant for that project. Questions related to

*"Can we probably use this software for this purpose, or might there be a problem?"*

will created by the COTS-board.

*This document is further described (with examples) in* ***APPENDIX SS.***

When there appears to be no immediate threat, the approval for testing will be given by the COTS-board. When necessary, the COTS-board will help provide a test, demo or developmental license for the testing department.

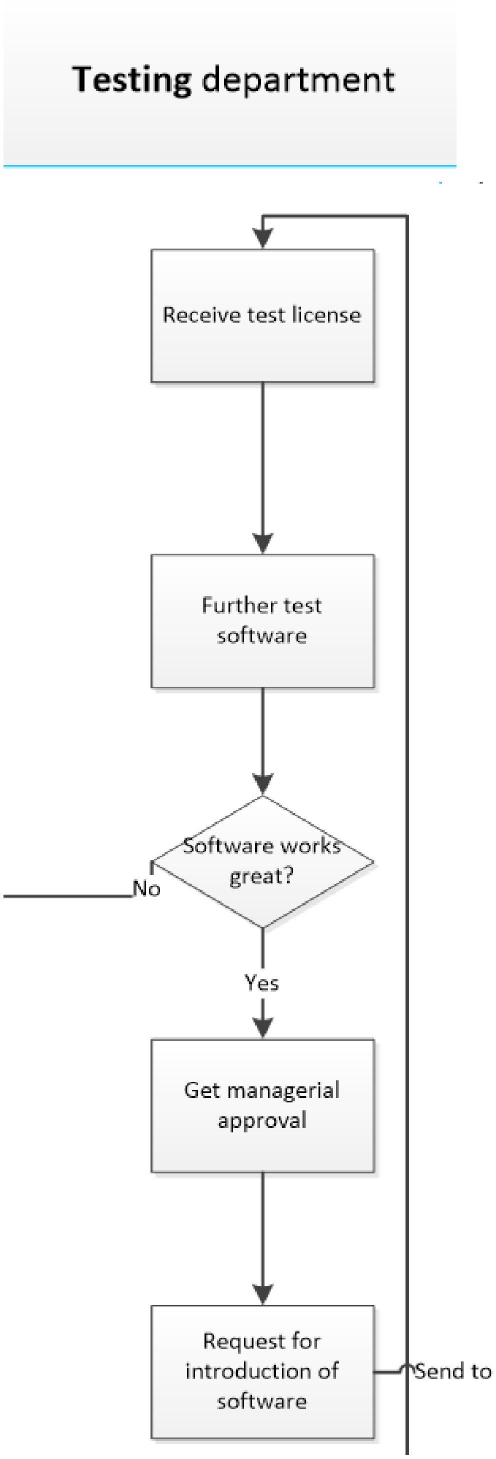

Commercial-off-the-shelf (COTS) Board
Request for software DOCUMENT
Checks if there is some similar software already used in the company
Yes
Convince Developmental department to use alternative existing software
No
Can we probably use this software for this purpose, or might there be a problem?
No
Yes
Send message to Developmental department that the software is not fine
Send short preliminary approval message for software
Write confirmation email to
Obtain test/demo license if needed

## 2. Testing

In this stage, the software will be tested in a similar setting as future usage of the software in that project.

If testing the software in the environment proves to be successful, management and engineers will say that they wish to get this software to get the process to continue.

As this concerns a software that has not yet been introduced before, it first has to be introduced.



The **Introduction of new software** will go in two steps as descrived in **chapter 4.3.2.**

It will be done by a team similar or identical to the COTS-Board, a specialized group of people called the **software introduction team**, that need to check the requirements of the software that will be entered into the company. Like hardware introduction teams, there has to be an expert on each aspect of software that individually is responsible for those relevant aspects.

The export-expert, IT-Purchaser, manager, legal expert and software engineer with knowledge of purchasing all have their own data to gather as shown in **chapter 4.3.4**. In smaller firms it is recommended to have the COTS-board and software introduction team are the same.

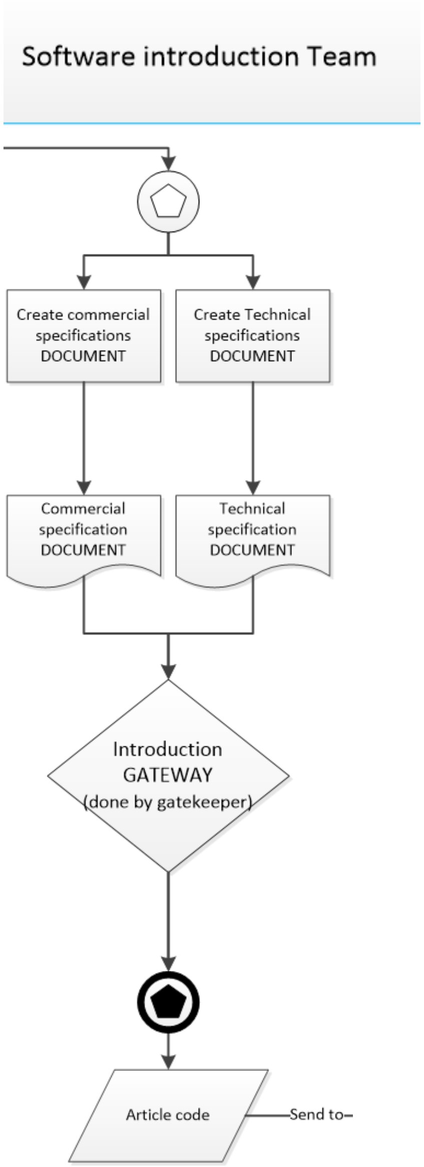
A) The **commercial data** will be gathered of the software. A document will be used that is made of the attributes listed in

**APPENDIX TT** *which comes with a filled out example document.*

B) The technical specification of software needs to be gathered by the relevant experts. Both are listed in

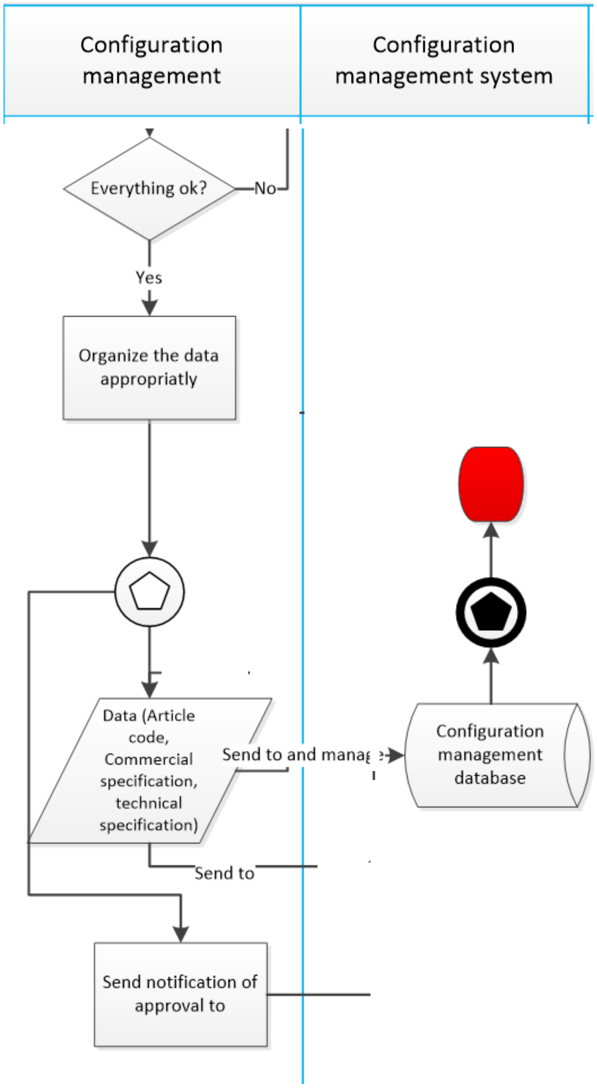
***APPENDIX UU** which comes with a filled out example document.*

After both documents are *fully* filled out, approved with no animalities in sight, there is enough data available to enter the product into the *configuration management system.*
This happens after the gateway, as described and exemplified in **APPENDIX VV.**


Software introduction Team
Create commercial specifications DOCUMENT
Create Technical specifications DOCUMENT
Commercial specification DOCUMENT
Technical specification DOCUMENT
Introduction GATEWAY (done by gatekeeper)
Article code
Send to

It is then registered as a component to use in products where the previous documents will be included into the registration.

It will then also obtain an **article number** which allows an item to be ordered.


Configuration management
Configuration management system
Everything ok?
No
Yes
Organize the data appropriatly
Data (Article code, Commercial specification, technical specification)
Send to and manag
Configuration management database
Send to
Send notification of approval to

## 3. Acceptance

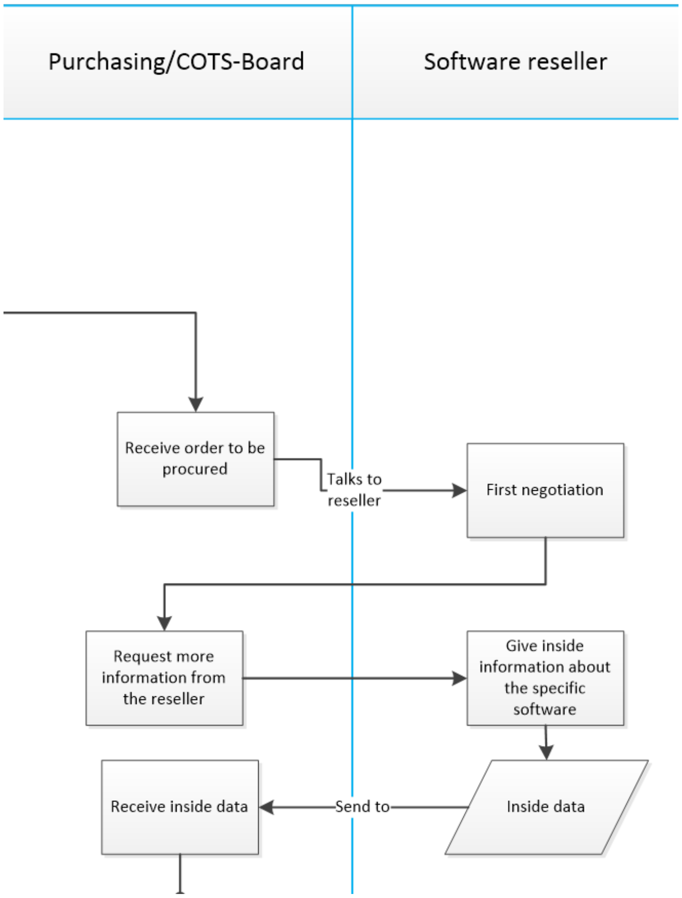
For the next developmental phase; **Acceptance,** the registration and configuration need to be in order. This phase consists of having a working prototype to show to a customer. Once this is accepted, the parts for the order need to be bought, including the software.

Therefore, the next users in the process will be a combination of the purchasing department and the COTS-board as items need to be procured, but with expert knowledge from a department, the COTS-board.

As software resellers might not be unbiased, the main contact between the reseller and the firm will be the COTS-board, which has expertise in software. The reseller might have valuable *inside information* regarding the placement and procurement of the software, and it should be discussed with a mutual expert. This exchange should be documented and discussed internally.

The role of the purchasing department is there to negotiate deals and maintain relationships with the reseller. The data flow goes both ways, from the reseller to the firm and back. This allows for many discussions to be done regarding the potential deal.

Once an agreement is reached regarding the price and conditions of the software, the logistical data need to be filled out by the purchasing department.


Purchasing/COTS-Board
Software reseller
Receive order to be procured
Talks to reseller
First negotiation
Request more information from the reseller
Give inside information about the specific software
Inside data
Send to
Receive inside data

*The logistical data document with filled out examples is described in* ***APPENDIX WW***

Once the documentation is done, the **purchasing order** can be shipped out to the reseller, and the deal can be fulfilled.

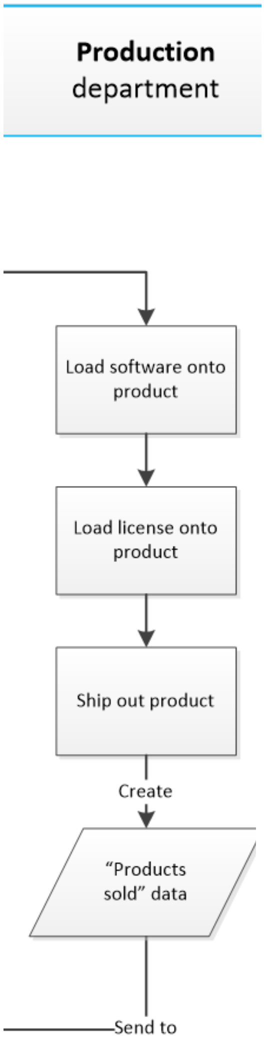
Upon receiving the purchasing order, the software reseller will create their own **purchasing document**, which at least includes a *receipt* and *proof of payment*; as well as the license keys which are ordered (the product).

Those keys should be delivered at a centralized entity, a dedicated email address to store temporarily store them and get an overview is enough for most companies. Here the resellers can send all their software license retrieval information to, where the company can decide how to link this to their software asset management system. As further described in chapter 4.3.3.4.

Those licenses are now in stock and can be used for production or for later use. The data gathered, as well as the stocks will (eventually) also go towards a centralized software asset management system, which will be explained in step 5.

4. Production

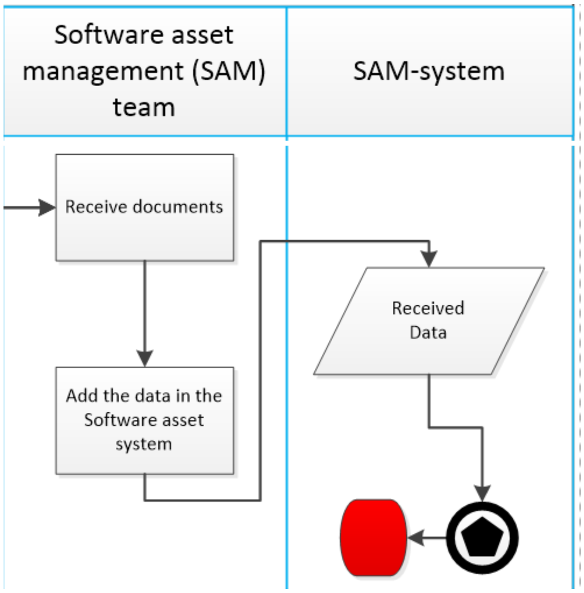
During the production of a product, many components are needed. Amongst those components, there is software. Similarly, to hardware, software (licenses) can also be in stock and expire. During the production, licenses will be taken from the stock when available, or reordered when necessary (see chapter 4.3.3.3 for that).



The licenses will then be added/embedded to the products and will be shipped out. The moment licenses are used, software asset management should act. This begins with the counting of licenses used, against the current stock, which is the most basic way of Software Asset Management.

5. Software asset management

In the end, all previous data and gathered documents will have to get into a centralized software asset management system.


Software asset management (SAM) team
SAM-system
Receive documents
Add the data in the Software asset system
Received Data

All the data will then go towards a software asset management system that gathers data from all departments. The flow of data is further described in chapter 4.3.4.

As purchasing orders, technical specifications, contracts and information regarding the supplier all come together in a centralized system, this needs to be managed.

The **project manager** that initially requested the software is going to be responsible for the dataflows of all the software that is used in his department. The manager of those project managers is the **Software Asset Manager** who is responsible for a central database of Software Asset Management. (see chapter 4.3.5.2 and Figure 5 Overview of users in Software Asset Management: Managers and software asset managers for more information)

This Software Asset Management-system is preferably capable of scanning the current systems and maintaining license control. Another useful feature is to be able to read the purchasing data and existing databases, to store data into a system and keep the software assets (licenses) in check and have data available for audits.

Comparing all the data in the centralized system is beneficial for checking of existing databases and keeping the data correct.

Now that the data is collected in a centralized Software Asset Management system, possibilities of synergy can be explored to let the firm become more efficient.

The purchasing department will have to automatically send over the purchasing data to the Software Asset Management With further automation this can be automatically recognized, decoded and inserted into the system.

As for Software Asset Management, it can report back to purchasing whenever a license is about to expire, or a license is overused (e.g. more computers than allowed, more people online than allowed). To trigger purchasing to take action upon the matter. The Software Asset Management database can also be used to add feedback and quality check of the product and supplier. Utilising Software Asset Management as an expediting, follow-up and evaluation method, fulfilling step 5 and 6 of van Weele's purchasing process.[153]

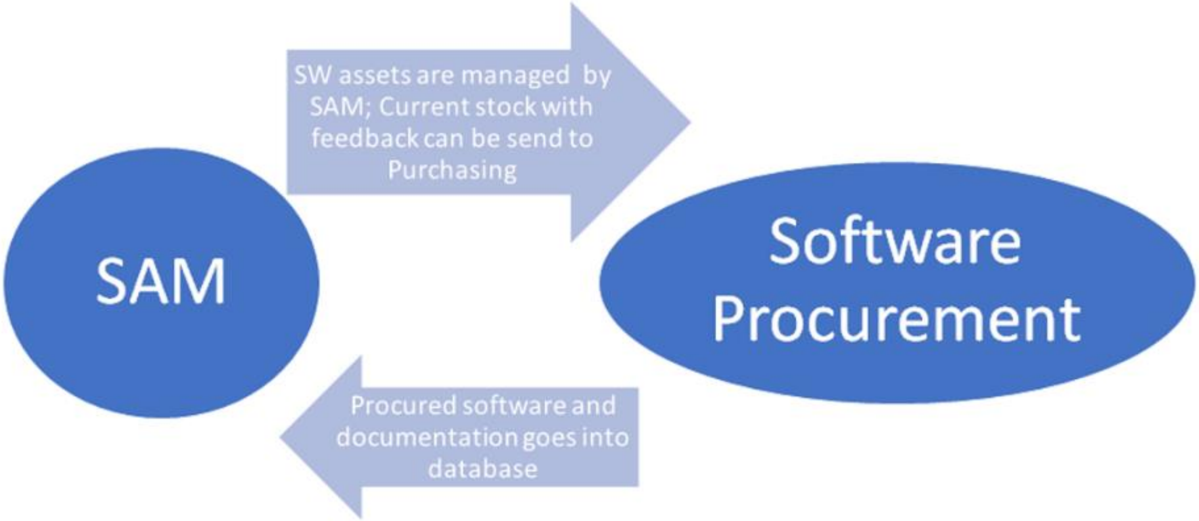

SW assets are managed by SAM; Current stock with feedback can be send to Purchasing
SAM
Software Procurement
Procured software and documentation goes into database

**Figure 8 Synergy between Software-asset-management and Software procurement**

A long-term task of Software Asset Management is to prevent the company from getting fined by auditors. Prevention is the first step, and this means collecting proof (data). Mandatory data encompass the **Proof of purchase**, the **Request for quotation**, the **purchasing order** and the **invoice** according to information compiled in chapter 4.3.6.2, to show to an auditor. House-rules should be in place regarding Software Asset Management. When an audit comes, they are only allowed to communicate with Software Asset Management.

An auditor should only be allowed to talk to the people that run Software Asset Management. They will direct the local departments to gather the information that is needed to prove the compliance. Further house rules are that random audits are only done by an independent 3rd party audit office and that the business departments collect their own required information which is then to be delivered to the SAM-team, not the auditor.

[153] See Van Weele (2010), p. 31

### 4.4.2 OLD VERSUS NEW: Solving the problems of the past

Coming back to the four main problems of the procurement of software: Lack of formalization, synchronization, enforcement and accountability. The first three are directly solved by the new formalized software procurement and asset management process, yet accountability is not directly solved. However, the roles of the software asset manager and the gatekeeper, who all control and oversee parts of the process can be held accountable or hold people accountable for not following the formalized process.

In the old process, it was possible to bypass the entire process with a WITTE BON, therefore risking not registering anything in the company and possibly misusing software which may not even be safe to use. Unchecked ordering of software is no longer possible, as there are now at least two people accountable, the software asset manager, and the gatekeeper. The latter one controls the documents side, and the first one the database and software license side while working with the managers (which can be held accountable).

# 5 DISCUSSION

## 5.1 THEORETICAL IMPLICATIONS: Procuring software is bound to practical problems

### 5.1.1 PURCHASING MATURITY: Higher performance with a higher maturity

Schiele [154] states that the maturity of the purchasing department affects performance. For software purchasing, it could be seen that in many of the cases discussed, the software was bought ad-hoc – which shows no real maturity. In order to increase its performance, a calculated new process should be in place. Not only focusing on getting the best price and conditions which is more relevant for hardware, but also to obtain a working management system. With millions of dollars in potential audit fines, it gives a new dimension to purchasing and supply chain management. With software purchasing maturity comes better asset management, which leads to many possibilities. The theoretical implications of the new possibilities of utilizing Software Asset Management into the purchasing process were not explored, but the practical examples given by the employees gave reasons to believe that an automated system might prove beneficial in moving purchasing to a higher level and might leave room for more maturity possibilities.

### 5.1.2 PURCHASING SOFTWARE: Still under-researched

The research brought upon new attributes for software procurement and a pragmatic way of implementing them into a purchasing process. A holistic view of the current ways of buying software is given in the high-tech industry. Compared to the theory of Van Weele [155]and Kraljic[156], one can conclude that companies still don’t know how to treat software right. The implication of this thesis is that software is a highly complex good that can be a threat to the supply chain if treated lightly.

The amount of solutions and support from the employees gathered by conducting the thesis gave an indication that involving the employees in the improvement of the process leads to less friction and resistance[157] and it is indicated that this has been fruitful to produce the process.

[154] See Schiele (2007), p. 276
[155] See Van Weele (2010, p. 9)
[156] See Kraljic (1983), p. 1
[157] See Salvendy (2001, p. 889)

### 5.1.3 AUTOMATIC PURCHASING: Possibilities become feasible

For purchasing management, the research into Software Asset Management opens up new possibilities for a tighter grip on the process and the software assets. With new developments such as Software Asset Management, utilization of ways to automate purchasing become easier accessible for companies. This can lead to the further automation of purchasing itself. A product with embedded software that can repurchase itself once it notices that the license will expire soon falls in line with (Schiele, 2016)[158] who talks about industry 4.0 in purchasing and the possibilities in it. Further on, bundling possibilities are easier, and new possibilities for automatic planning are the further implications of this research.

### 5.1.4 SIMPLE DESIGN: Behaviour of people

As could be seen in the case study, companies were reluctant to change their behaviour. Processes were not followed and changing into a better process proved difficult. This falls in line with the theory [159]. A solution should, therefore, be kept simple and not require increased workload of more than 10%[160]. Moving from Excel sheets what many companies are using for Software Asset Management, towards a centralized management system where the employees fill in spreadsheets in Software Asset Management falls in line with the theory. Simplicity is what a new process should require, as well as clear possibilities for advantages. Making the lives of employees simpler would generally be in favour of the employees themselves and letting them help make their lives better might have helped even more.

With a clear structure of project managers that take responsibility and a Software Asset manager that manages all the data, this falls in line with taking the path of least resistance as for change.

## 5.2 MANAGERIAL IMPLICATIONS: A structured, simple process is more functional than a AD-Hoc process

## 5.3 GENERAL MANAGERIAL IMPLICATIONS

The managerial implications for this research fall in line with the theoretical implications. Regarding a simple, easy-to-use process that manages the data well enough for the requirements of Software Asset Management. Advantages of automation in software procurement and software asset management might seem abstract at the beginning. But decreased time spent on audits and a central overview of the current usage and stock will provide possibilities for future cost

[158] See Schiele (2016), p. 1
[159] See Sirkin et al. (2005), p. 104
[160] See Sirkin et al. (2005), p. 98

savings. Good data management can save time and money in the future. It is also advised to use the institutions that one already has regarding purchasing, analyse the problems with them and extend from that point on. Reshaping the entire supply chain for the promise of Software Asset Management might not be the easier solution.

Gradual changes over time might provide less resistance than substantial changes and to 'real change'.[161] Going from employees filling in data on in whatever format they prefer to a unified spreadsheet system with oversight of a Software Asset Manager is not extensively different from each other, yet will give the company the data it needs to survive audits and optimize their supply chain.

Further on, in smaller firms, it would be recommended to have the COTS-Bard and the software introduction team merge into one entity, due to a lack of more employees to fill these posts. What counts for both SME and large firms, membership of the team does not have to be a fulltime job but can be several hours of a week next to an employee's regular tasks.

Regarding follow up of processes. During the entire flow of the chain, from buying software for the first time to managing licenses – protocols need not only be in place like the introduction of new software but also be used, as <redacted>. Strong leadership while showing the employees that there is an advantage to utilizing the new process might turn fruitful.

### 5.3.1 IMPLICATIONS FOR THALES

Thales has struggled with its software asset management for some time. Because the company is not used to managing projects and did not require purchasing and management protocols for software, it struggled with the transition to a more project related business.

While the general managerial implications are very relevant for Thales, there are some specific implications that imply only to Thales.

The implication for the company is that it would benefit significantly with a software asset management system. While the internal software is already running on Software x, the product software remains unmanaged. While a license was already bought for the security project and general knowledge of the program (although different module) exists in the company, it is advised to use what one already has and link it together as much as possible. The software would fit their requirement. Software Asset Management has many advantages; such as having a centralize inventory of sofware, knowing exactly what is used where, and when it is about to expire. This leads to the next advantage: automatic ordering of expired or overused software. All lead to the company attempting to be compliant, and providing the data for possibly dangerous and expensive audits. Although Software Asset Management was not necessary in the past, thinking of the future would have been a great time and money saver, and Thales has a high chance of saving money in the future with Software Asset Management.

[161] Hage and Aiken (1970), p. 105; Kickert and van der Meer (2011), p. 475; Van De Ven (1993), p. 222

# 6 LIMITATIONS: Small case study in a scoped field

Limitations of the study concern mostly the amount of companies interviewed and the effectiveness of the solution.

In the study, a big case study was made of Thales Huizen, and four industry peers were interviewed about their software sourcing and management. For a full cross-case analysis, this is too few, as the case studies with the industry peers were firstly too short and not numerous enough. A follow-up study would have to look at more companies.

Further on, the research has been written down, but not tested yet. Although every step and partial solution has been informally discussed with employees who would have to use that solution, it is untested if it will first be used, and secondly, be better than the original processes. Due to time constraints and the years-long process of implementing new processes in tech-companies, putting the solution into praxis was not possible.

As most of the information that turned into the solution came from Thales, it might be biased towards Thales and the solutions that work for them. Even though the effort has been put into making the research as generally applicable as possible, this remains a limitation. Data origin has an effect on the outcome.

# 7 FUTURE RESEARCH: Testing the effectiveness of the process and in-depth analysis of software and purchasing

This case study delivered a concept of a software sourcing process based on case studies. Due to the time constraints it has not been tested. Further research could take the model and apply it to a company to test its functionality and limits. As well as test the managemental advisory approach listed in the thesis and extending it towards a practical approach of finalizing Van Weele's purchasing process in the context of software sourcing.

The attributes created were based on a limited number of interviews, from a handful of companies. Software sourcing and management specific attributes could be studied further, specifically by diving into the technical side of software. As this thesis consisted of a combination of purchasing and software, studying them separately in dept would further provide the process with valuable knowledge for improvement.

The thesis has explored the possibilities of utilizing a companies own employees to find solutions for problems, which were initially identified by the employees themselves with the help of the author. It is not known if this was more fruitful than if it had been done on a sole theoretical approach, and further research could test the effectiveness of this method, taking into account more factors.

# BIBLIOGRAPHY

1. **Abts, C., Boehm, B. W., & Clark, E. B. (2000**). *COCOTS: A COTS software integration lifecycle cost model-model overview and preliminary data collection findings.* Paper presented at the ESCOM-SCOPE Conference.
2. **Ahmed, S., Jones, A., Christiansen, D., Probert, D., Sethuraman, N., Grayson, L., . . . Moore, V. (2007)**. United States Patent No.: U. S. P. a. T. O. (USPTO).
3. **Akech, J. M. M. (2005)**. Development partners and governance of public procurement in Kenya: enhancing democracy in the administration of aid. *New York University Journal of International Law and Politics, 37*, 34.
4. **Al-Mahmood, S., & Al A'ali, M. (2011)**. COTS software procurement methodology. In *E-Strategies for Resource Management Systems: Planning and Implementation* (pp. 288-305): IGI Global.
5. **Anheuser-Busch-InBev. (2018)**. *UNITED STATES SECURITIES AND EXCHANGE COMMISSION WASHINGTON, D.C. 20549 FORM 20-F Anheuser-Busch InBev SA/NV*. Retrieved from https://www.ab-inbev.com/content/dam/universaltemplate/ab-inbev/investors/reports-and-filings/sec-filings/SEC_Filings_20-F_20180319.pdf
6. **Apsler, R., & Sears, D. O. (1968)**. Warning, personal involvement, and attitude change. *Journal of Personality and Social Psychology, 9*(2p1), 4.
7. **Ben-Menachem, M. (2008)**. Towards management of software as assets: A literature review with additional sources. *Information software technology, 50*(4), 241-258.
8. **Bender, P. S., Brown, R. W., Isaac, M. H., & Shapiro, J. F. (1985)**. Improving purchasing productivity at IBM with a normative decision support system. *Interfaces, 15*(3), 106-115.
9. **Bildsten, L. (2013**). *Purchasing decisions: enablers and barriers to innovation in construction.* Paper presented at the Proceedings of the CIB World Congress, Brisbane, Australia.
10. **Boehm, B. (1996)**. Anchoring the software process. *IEEE software, 13*(4), 73-82.
11. **Braun, C. L. (1999**). *A lifecycle process for the effective reuse of commercial off-the-shelf (COTS) software.* Paper presented at the Proceedings of the 1999 symposium on Software reusability.
12. **Brown, A. W., & Wallnau, K. C. (1996)**. A framework for evaluating software technology. *IEEE software, 13*(5), 39-49.
13. **Callagy, C. (2007)**. *Commercial Off-The-Shelf Software: Using Purchasing Portfolio Management to Gain Competitive Advantage.* National University of Ireland,, Galway.
14. **Cavinato, J. L., & Kauffman, R. G. (1999).** *The purchasing handbook: A guide for the purchasing and supply professional*. Tempe, Arizona: McGraw-Hill.
15. **Chapman, T. L., Dempsey, J. J., Ramsdell, G., & Reopel, M. R. (1997)**. Purchasing: no time for lone rangers. *The McKinsey Quarterly*(2), 31-32.
16. **Charette, R. N. (2005)**. Why software fails [software failure]. *IEEE spectrum, 42*(9), 42-49.

17. **Chen, C. J. (2009)**. Apply Project Quality Management in the Process of New Product Development. *Law and Economy, 18*(11), 58-62.
18. **Cheng, K., & Finney, S. (2016)**. The Tangle of Intangible Assets and Business Combinations. *The CPA Journal, 86*(1), 40.
19. **Comella-Dorda, S., Dean, J. C., Morris, E., & Oberndorf, P. (2002**). *A process for COTS software product evaluation.* Paper presented at the International Conference on COTS-Based Software Systems.
20. **Software Engineering: Guidelines for the Application of ISO 9001:1997 to Computer Software, (1997**).
21. **Cooper, R. G. (1996)**. Overhauling the new product process. *Industrial marketing management, 25*(6), 465-482.
22. **CrossFitGames. (2019**). DÓTTIR is the Icelandic word for DAUGHTER. Retrieved from https://twitter.com/CrossFitGames/status/1124878274285301761
23. **Daniel, K., & Titman, S. (2006)**. Market reactions to tangible and intangible information. *The Journal of Finance, 61*(4), 1605-1643.
24. **Day, M. (2002).** *Gower handbook of purchasing management*: Gower Publishing, Ltd.
25. **De Boer, L., Harink, J., & Heijboer, G. (2002)**. A conceptual model for assessing the impact of electronic procurement. *European Journal of purchasing supply management, 8*(1), 25-33.
26. **Denzin, N. K. (2017).** *The research act: A theoretical introduction to sociological methods*: Routledge.
27. **DISMISSAL OF MARS, INCORPORATED, ET AL VS. ORACLE CORPORATION, ET AL**, No. CGD-15-548606 [2017] EWHC 189 (TCC) (Superior court of California 2015).
28. **Duke, C. R., & Persia, M. A. (1994)**. Effects of distribution channel level on tour purchasing attributes and information sources. *Journal of Travel and Tourism Marketing, 2*(2-3), 37-56.
29. **Ellram, L. M. (1996)**. A structured method for applying purchasing cost management tools. *International Journal of Purchasing and Materials Management, 32*(4), 11-19.
30. **Ellram, L. M., & Carr, A. (1994)**. Strategic purchasing: a history and review of the literature. *International Journal of Purchasing, 30*(1), 9-19.
31. **Evans, D. J., & Morriss, O. (1984)**. Non-Newtonian molecular dynamics. *Computer Physics Reports, 1*(6), 297-343.
32. **Foerstl, K., Hartmann, E., Wynstra, F., & Moser, R. (2013)**. Cross-functional integration and functional coordination in purchasing and supply management: Antecedents and effects on purchasing and firm performance. *International Journal of Operations, 33*(6), 689-721.
33. **Gangadharan, G., D'andrea, V., De Paoli, S., & Weiss, M. (2012)**. Managing license compliance in free and open source software development. *Information Systems Frontiers, 14*(2), 143-154.
34. **Gao, Z., & Tang, L. (2003)**. A multi-objective model for purchasing of bulk raw materials of a large-scale integrated steel plant. *International Journal of Production Economics, 83*(3), 325-334.
35. **Gomulkiewicz, R. W. (1998)**. The license is the product: comments on the promise of article 2B for software and information licensing. *Berkeley Tech. LJ, 13*, 891.

36. **Gugler, W. (1996).** *Jehu und seine Revolution: Voraussetzungen, Verlauf, Folgen*. Kampen: Kok.
37. **Hage, J., & Aiken, M. (1970).** *Social change in complex organizations* (Vol. 41). New York: Random House Trade.
38. **Hanson, D. (2005).** *CE marking, product standards and world trade*: Edward Elgar Publishing.
39. **Hiltzik, M. (2018, NOV 16, 2018).** How Apple and other manufacturers attack your right to repair their products. . *Latimes*. Retrieved from https://www.latimes.com/business/hiltzik/la-fi-hiltzik-right-repair-20181116-story.html
40. **Hussey, D. E. (1978)**. Portfolio analysis: practical experience with the directional policy matrix. *Long Range Planning, 11*(4), 2-8.
41. **IEEE-Standards-Committee. (1990)**. IEEE std 610.12-1990 ieee standard glossary of software engineering terminology. *IEEE standards*.
42. **Irwin, S. (2016**). SAM – More than License Compliance & Management. Retrieved from http://itak.iaitam.org/sam-license-compliance-management/
43. **Jakubička, M. (2010**). *Software asset management.* Paper presented at the 2010 IEEE International Conference on Software Maintenance.
44. **Jakubička, M. (2012)**. *Software asset management in a Large (Academic) Organization.* (PhD), Masarykova univerzita, Fakulta informatiky,
45. **Kaplan, R. S., & Norton, D. P. (2004)**. The strategy map: guide to aligning intangible assets. *Strategy leadership, 32*(5), 10-17.
46. **Kickert, W. J. M., & van der Meer, F.-B. (2011)**. Small, Slow, and Gradual Reform: What can Historical Institutionalism Teach us? *International Journal of Public Administration, 34*(8), 475-485. doi:10.1080/01900692.2011.583768
47. **Kim, D., Moon, J., Cho, S., Choi, J., Park, M., Han, S., & Chung, L. (2014**). *A birthmark-based method for intellectual software asset management.* Paper presented at the Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication.
48. **Koebler, J. (2018).** Tractor-Hacking Farmers Are Leading a Revolt Against Big Tech's Repair Monopolies. *Motherboard Vice*. Retrieved from https://motherboard.vice.com/en_us/article/kzp7ny/tractor-hacking-right-to-repair
49. **Kotter, J. P. (1995)**. Leading change: Why transformation efforts fail. *Harvard Business Review, March-April 1995*.
50. **Kraljic, P. (1983)**. Purchasing must become supply management. *Harvard Business Review, 61*(5), 109-117.
51. **Lamoureux, T., Brill, R., & Joshi, A. (2007)**. *Is unlicensed software hurting your bottom line? Leading practices to Reduce Revenue Loss*. Retrieved from https://advisory.kpmg.us/content/dam/advisory/en/pdfs/risk-assurance/unlicensed-software-increase-revenue.pdf
52. **Lea Hyer, N. (1984)**. The potential of group technology for US manufacturing. *Journal of Operations Management, 4*(3), 183-202.
53. **Leveson, N. G., & Weiss, K. A. (2009)**. Chapter 15 - Software System Safety. In G. E. Musgrave, A. M. Larsen, & T. Sgobba (Eds.), *Safety Design for Space Systems* (pp. 475-505). Burlington: Butterworth-Heinemann.

54. **Mackie, k. (2014)**. *Microsoft Found To Issue the Most Software Audits.* Retrieved from Redmond: https://redmondmag.com/articles/2014/05/02/microsoft-software-audits.aspx
55. **Majchrzak, A. (1988).** *The human side of factory automation: Managerial and human resource strategies for making automation succeed*: Jossey-Bass.
56. **Marquis, H., Spivak, G., & Barber, V. (2016)**. *Cut Software Spending Safely With SAM*. Retrieved from
57. **MARS, INCORPORATED, ET AL VS. ORACLE CORPORATION, ET AL**, No. CGC15548606, Arlene Ramos (The superior court of California 2015).
58. **McCaffrey, M. (2019)**. *Global 100 Software Leaders by revenue* Retrieved from https://www.pwc.com/gx/en/industries/technology/publications/global-100-software-leaders/explore-the-data.html
59. **McCarthy, M. A., & Herger, L. M. (2011**). *Managing software assets in a global enterprise.* Paper presented at the 2011 IEEE International Conference on Services Computing.
60. **Merola, L. (2006**). *The COTS software obsolescence threat.* Paper presented at the Fifth International Conference on Commercial-off-the-Shelf (COTS)-Based Software Systems (ICCBSS'05).
61. **Merriam, S. B. (1998).** *Qualitative Research and Case Study Applications in Education. Revised and Expanded from" Case Study Research in Education."*: ERIC.
62. **Montero, M. J. (2007)**. *Virtualisatie en IT-auditing.* (PhD Postgraduate dissertation), Vrije Universiteit, Amsterdam, The Netherlands.
63. **Moyle, K. (2004)**. *What place does open source software have in Australian And New Zealand schools' and jurisdictions' ICT portfolios?* Retrieved from
64. **Nettsträter, A., Geißen, T., Witthaut, M., Ebel, D., & Schoneboom, J. (2015)**. Logistics software systems and functions: an overview of ERP, WMS, TMS and SCM systems. In *Cloud Computing for Logistics* (pp. 1-11): Springer.
65. **England and Wales High Court (Technology and Construction Court) Decisions between SAP UK LIMITED AND DIAGEO GREAT BRITAIN LIMITED**, No. HT-2015-000340 (Royal Courts of Justice 2017).
66. **Oberndorf, P. A. (1997**). *Facilitating component-based software engineering: COTS and open systems.* Paper presented at the Proceedings Fifth International Symposium on Assessment of Software Tools and Technologies.
67. **Olympus_Group. (2018)**. *Integrated Report from Fyscal Year ended Mar. 31, 2018*. Retrieved from Tokyo: https://www.olympus-global.com/ir/data/integratedreport/pdf/integrated_report_2018e_A3.pdf
68. **Overdijk, T., van der Putt, P., de Vries, E., & Schafft, T. (2011**). *Exhaustion and software resale rights.* Paper presented at the CRi.
69. **Padgett, D. K. (2016).** *Qualitative methods in social work research* (Vol. 36): Sage Publications.
70. **Paik, S.-K. (2011**). *Supply management in small and medium-sized enterprises: role of SME size.* Paper presented at the Supply Chain Forum: An International Journal.
71. **Pearson, J. N., & Gritzmacher, K. J. (1990)**. Integrating purchasing into strategic management. *Long Range Planning, 23*(3), 91-99.

72. **Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007)**. A design science research methodology for information systems research. *Journal of management information systems, 24*(3), 45-77.
73. **Ponisio, L., & Vruggink, P. (2011)**. Effective monitoring and control of outsourced software development projects. In *Information Systems Development* (pp. 135-147): Springer.
74. **Postema, B. F. (2018).** *Energy-efficient Data Centres: Model-based Analysis of Power-performance Trade-offs*. Enschede: University of Twente.
75. **Pressman, R. S. (2005).** *Software engineering: a practitioner's approach* (7 ed.). New york, New York: McGraw-Hill.
76. **Pyman, M., Wilson, R., & Scott, D. (2009)**. The extent of single sourcing in Defence Procurement and its relevance as a corruption risk: A first look. *Defence and Peace Economics, 20*(3), 215-232. doi:10.1080/10242690802016506
77. **Raman, R., Livny, M., & Solomon, M. (2000**). *Resource management through multilateral matchmaking.* Paper presented at the Proceedings the Ninth International Symposium on High-Performance Distributed Computing.
78. **Robinson, P. J., Faris, C. W., & Wind, Y. (1967).** *Industrial buying and creative marketing*: Allyn and Bacon.
79. **Rossler, P. E., & Hirsz, A. (1996)**. Purchasing's interaction with customers: The effects on customer satisfaction—A case study. *International Journal of Purchasing Materials Management, 32*(4), 37-43.
80. **Rothbauer, P. J. T. S. e. o. q. r. m. (2008)**. Triangulation. *1*, 892-894.
81. **Rozemeijer, F. A., Van Weele, A., & Weggeman, M. (2003)**. Creating corporate advantage through purchasing: toward a contingency model. *Journal of Supply Chain Management, 39*(4), 4-13.
82. **Salvendy, G. (2001).** *Handbook of industrial engineering: technology and operations management*. West Lafayette, Indiana: John Wiley & Sons.
83. **Schiele, H. (2007)**. Supply-management maturity, cost savings and purchasing absorptive capacity: Testing the procurement–performance link. *Journal of Purchasing and Supply Management, 13*(4), 274-293.
84. **Schiele, H. (2016)**. Industrie 4.0 in der Beschaffung. *Wingbusiness*, 15-18.
85. **Schiele, H. (2017)**. *Handbook of Operations*. DRAFT 170109 Purchasing and supply Management. University of Twente. Enschede.
86. **Schuetze, W. P. (2004).** *What is an Asset?* : Routledge.
87. **ServiceNow. (2019**). Scan software installations with the system scheduler. Retrieved from https://docs.servicenow.com/bundle/london-it-service-management/page/product/asset-management/concept/c_ScanSWInstWithSysSched.html
88. **Shakespeare, W. (1806).** *Romeo and Juliet*: John Cawthorn.
89. **Sirkin, H. L., Keenan, P., & Jackson, A. (2005)**. The hard side of change management. *HBR's 10 Must Reads on Change*, 99.
90. **Telgen, J. (1994)**. Inzicht en overzicht : de uitdagingen van besliskunde en inkoopmanagement. In: Universiteit Twente.
91. **Telgen, J. (2005)**. Purchasing control, compliance and the box: purchasing management based on hard data. In *Perspektiven des Supply Management* (pp. 413-423): Springer.
92. **Torchiano, M., & Morisio, M. (2004)**. Overlooked aspects of COTS-based development. *IEEE software, 21*(2), 88-93.

93. **Turnage, J. J. (1990)**. The challenge of new workplace technology for psychology. *American psychologist, 45*(2), 171.

94. **van Aken, J. E. (2004)**. Management research based on the paradigm of the design sciences: the quest for field-tested and grounded technological rules. *Journal of management studies, 41*(2), 219-246.

95. **Van De Ven, H. (1993)**. The development of an infrastructure for entrepreneurship. *Journal of Business Venturing, 8*(3), 211-230.

96. **Van Der Burg, S., Dolstra, E., McIntosh, S., Davies, J., German, D. M., & Hemel, A. (2014**). *Tracing software build processes to uncover license compliance inconsistencies.* Paper presented at the Proceedings of the 29th ACM/IEEE international conference on Automated software engineering.

97. **Van der Valk, W., & Rozemeijer, F. (2009)**. Buying business services: towards a structured service purchasing process. *Journal of Services Marketing, 23*(1), 3-10.

98. **Van Weele, A. J. (2010).** *Purchasing & supply chain management: analysis, strategy, planning and practice*: Cengage Learning EMEA.

99. **van Weele, A. J., & Eßig, M. (2017)**. Rolle der Beschaffung im Wertschöpfungsprozess. In *Strategische Beschaffung* (pp. 13-38): Springer.

100. **Vernede, R. (2017)**. To segment or not to segment: Een overzicht van nut en noodzaak van netwerksegmentatie. *Informatie Beveiliging Magazine, 17*(3), 4-10.

101. **Vigder, M., Gentleman, W. M., & Dean, J. (1996)**. *COTS Software Integration: State of the art*. Retrieved from Montreal:

102. **Vigder, M. R., & Dean, J. (1997**). *An architectural approach to building systems from COTS software components.* Paper presented at the Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research.

103. **Wang, Y., & Hazen, B. (2016)**. Consumer product knowledge and intention to purchase remanufactured products. *International Journal of Production Economics, 181*, 460-469.

104. **Waters, C. D. J. (2006).** *Operations strategy*. London: Cengage Learning EMEA.

105. **Wei, J., Zhang, X., Ammons, G., Bala, V., & Ning, P. (2009**). *Managing security of virtual machine images in a cloud environment.* Paper presented at the Proceedings of the 2009 ACM workshop on Cloud computing security.

106. **Wiens, K. (2014).** You Gotta Fight For Your Right to Repair Your Car. . *The Atlantic*. Retrieved from https://www.theatlantic.com/technology/archive/2014/02/you-gotta-fight-for-your-right-to-repair-your-car/283791

107. **Williams, C., & O'Connor, S. (2011)**. Four best Practices for Software Asset Management. *BCM Software Industry Insights*.

108. **Yin, R. K. (2017).** *Case study research and applications: Design and methods*: Sage publications.

# APPENDIX

## APPENDIX G

Summarized overview and results of interviews

| Interview style/ Subject | Company | Person | Date | Length (hours) | Appendix/Table |
|---|---|---|---|---|---|
| **Email communication regarding software procurement in their company** | Company D | Purchasing (manager) | 04 June 2018 14:24 | N.a. | A |
| **Informal interview regarding the introduction process, its difficulties with bypassing and configuration management** | Thales Huizen | Employee (Export control) | 06 June 2018 12:00-15:00 | 3 | B |
| **One question interview about which software is used in Thales Huizen** | Thales Huizen | IT purchaser | 07 June 2018 09:00-11:00 | 2 | C |
| **One question interview about which software is used in Thales Huizen** | Thales Huizen | License Manager | 07 June 2018 14:00-15:00 | 1 | C |
| **One question interview about which software is used in Thales Huizen and double check the previous answers** | Thales Huizen | IT Infrastructure designer (Contractor) | 07 June 2018 16:00-18:00 | 2 | C |
| **Talk regarding the defense project, its struggles and background.** | Thales Huizen | IT infrastructure designer (contractor) | 13 June 2018 13:30-15:00 | 1.5 | H |
| **Phone call about the COTS-board and COTS software use in Hengelo** | Thales Hengelo | Member of COTS board | 18 June 2018 11:00-12:30 | 1.5 | I |
| **Talk about the inner workings of non-structural software asset management** | Thales Huizen | Software license manager | 20 June 2018 16:00-16:30 | 0.5 | K |
| **Informal talks regarding the simplification of software request via email** | Thales Huizen | 8 engineers | 20-22 June 2018 | 1 | N |
| **Interview regarding the complexity of software in the product line and the (historical) need for SAM** | Thales Huizen | (license) Manager product line | 25 June 2018 | 1 | O |

| | | | | | |
|---|---|---|---|---|---|
| | | | 09:00-10:00 | | |
| **Interview regarding Configuration management, risks of noncompliance and big projects** | Thales Huizen | Configuration/obsolescence manager | 25 June 2018 11:00-14:00 | 3 | P |
| **Interview about compliancy issues in Huizen, procedure of new functionality and open-source software** | Thales Huizen | COTS-Board member, project manager | 27 June 2018 15:00-18:00 | 3 | R |
| **Interview about what software is used in projects and how the project is being managed in regards to SAM** | Thales Huizen | Infrastructure project manager/engineer | 28 June 2018 13:00-15:30 | 2.5 | S |
| **LinkedIn interview regarding Company C's software sourcing and asset management** | Company C | IT purchaser responsible manager | 02 July 2018 20:44-23:39 | 3 | JJ |
| **Talk about the COTS-board in regards to deeper workings of the board and configuration management** | Thales Huizen | COTS-board member | 04 July 2018 09:00-12:00 | 3 | T |
| **Phone call regarding software procurement, the COTS-board, reasons why the system is bypassed** | Thales Hengelo | Process manager (responsible for processes) | 09 July 2018 08:30-10:00 | 1.5 | U |
| **Interview regarding the problems with the current software procurement** | Thales Huizen | IT purchaser | 13 July 2018 13:00-15:00 | 2 | V |
| **Discussion about using the ServiceNow SAM module as SAM and the limitations and future problems of such a system. As well as a confirmation of the previous mentioned antecedents** | Thales Huizen | Configuration/obsolescence manager | 19 July 2018 17:00-19:00 | 2 | X |
| **Interview regarding the supply chain of projects, definition of an Asset within Thales and the central mailbox for software purchases** | Thales Huizen | Manager supply chain engineering | 23 July 2018 8:30-10:00 | 1.5 | Y |
| **Online chat about the development of products in an DTAP environment and its implications on software asset management and purchasing** | Thales Huizen | IT infrastructure designer (contractor) | 23 July 2018 13:30-14:30 | 1 | Z |
| **Explorative interview regarding the problems with purchasing software for projects** | Thales Huizen | Project leader (infrastructure) | 27 July 2018 | 1.5 | AA |

| | | | | | |
|---|---|---|---|---|---|
| **and the need for automation and a structured, standardized process with ServiceNow with the limitations in place** | | | 15:00-16:30 | | |
| **Phone call regarding software procurement and software asset management in NS and how to prevent Audits** | Company B | IT purchaser | 31 July 2018 12:00-15:00 | 3 | CC |
| **Open interview regarding the setup of the defense project, its planning and possibilities with software procurement via automation** | Thales Huizen | Infrastructure engineer | 31 July 2018 15:00-16:30 | 1.5 | BB |
| **Interview about the inner workings (management, CM, purchasing) of the defense project; Possibilities and limitations of SAM; Solutions of SAM** | Thales Huizen | Project manager | 01 Aug 2018 08:00-10:30 | 2.5 | F |
| **Phone call interview regarding software procurement and asset management in Company A with regards to the setting up of a process and utilizing suppliers** | Company A Shipyards | Vendor and contract manager IT | 01 Aug 2018 11:00-13:00 | 2 | DD |
| **Orientated interview about what to think of with Software asset management in regards to being able to reuse configuration easily in a SAM-system** | Thales Huizen | (Software) manager | 01 Aug 2018 15:00-16:00 | 1 | EE |
| **Multiple emails as reply on why software is bought via the wittebon; and he difficulties in management of it** | Thales Huizen | **6 people**: Project manager (2x), configuration manager (2x), export control, Head of purchasing, | 08 Aug 2018 13:54-18:45 | N.a. | FF, GG |
| **Interview about the bypassing of the purchasing process in the future and how to prevent it** | Thales Huizen | Supply chain officer | 09 Aug 2018 10:30-11:30 | 1 | HH |
| **Email regarding software procurement troubles and possibilities with bypassing security concerns** | Thales Huizen | Contractor executive project management | 09 Aug 2018 14:11 | N.a. | II |
| **Compilation of data (antecedents) that needs to be in a SAM system and therefore needs to be created at some point.** | Thales Huizen | License manager/engineer | 09 Aug 2018 16:00-17:00 | 1 | W , Table 6, Table 8 |
| **Compilation of data (antecedents) that needs to be in a SAM system and therefore needs to be created at some point. As well as a check on the previous antecedents and work on the solution.** | Thales Huizen | 5 Purchaser, One Purchasing manager, one purchasing officer | 10 Aug 2018 08:00-19:00 | N.a. | |

| | | | | | |
|---|---|---|---|---|---|
| **Compilation of data (antecedents) that needs to be in a SAM system and therefore needs to be created at some point. As well as a check on the previous antecedents and work on the solution.** | Thales Huizen | 6 Engineers (two of them contractors) | 13 Aug 2018 08:00-19:00 | N.a. | Table 6 |
| **Compilation of data (antecedents) that needs to be in a SAM system and therefore needs to be created at some point. As well as a check on the previous antecedents and work on the solution.** | Thales Huizen | Three engineer managers (infrastructure, IT and hardware), Two Project managers | 14 Aug 2018 08:00-19:00 | N.a. | Table 6, Table 9 |
| **Compilation of data (antecedents) that needs to be in a SAM system and therefore needs to be created at some point. As well as a check on the previous antecedents and work on the solution.** | Thales Huizen | Two Configuration managers, one export controller | 15 Aug 2018 08:00-19:00 | N.a. | Table 8; Table 6, Table 9 |

## APPENDIX J
## Microsoft Vision icons used and their meaning

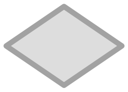
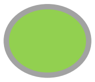
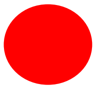
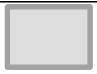
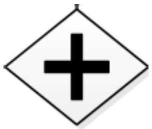
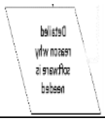
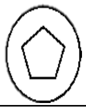
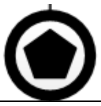
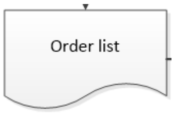
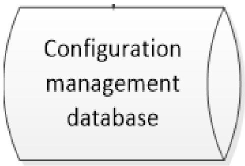
| Figure | Meaning |
|---|---|
| Triangle  | Decision moment. |
| Start  | The start of a process |
| End  | The end of a process |
| Action  | The action that is taking place. |
| Multiple actions start  | Point where two or more actions start |
| Documentation Letter  | Documented letter or email |
| Multiple Actions  | Multiple actions start with this icon. |
| Multiple events end  | In this case, multiple events have started before, but will end now and come together here. |
| List of items/document  | Can include orders, documents, lists, any form of data.. |
| Database  | Database with data in it |

**Table 10 Symbols used for business process modelling**

## APPENDIX KK

## INTERVIEW QUESTIONS

### General questions

- How do you interact with the purchasing process of software?
- How would you describe the current process?
- What are your experiences with the current process?
- Do you feel like you understand the current process?
- Could you guide me through the process of your task?
- Where is room for improvement?
- What kind of solution/system do you think would be better or benefit your task?
- What do you think is missing in the current process? Or, how should it look like?
- What makes you use or bypass the process?
- Who do you talk to if you have questions or get stuck?

### *Function/role specific questions:*

### Engineers

- How do you decide which software to buy?
- If we wanted to build a database (Software Asset Management) with a list of software (assets) for an easy overview and management, what information would need to be inside?

### Purchasers/engineers:

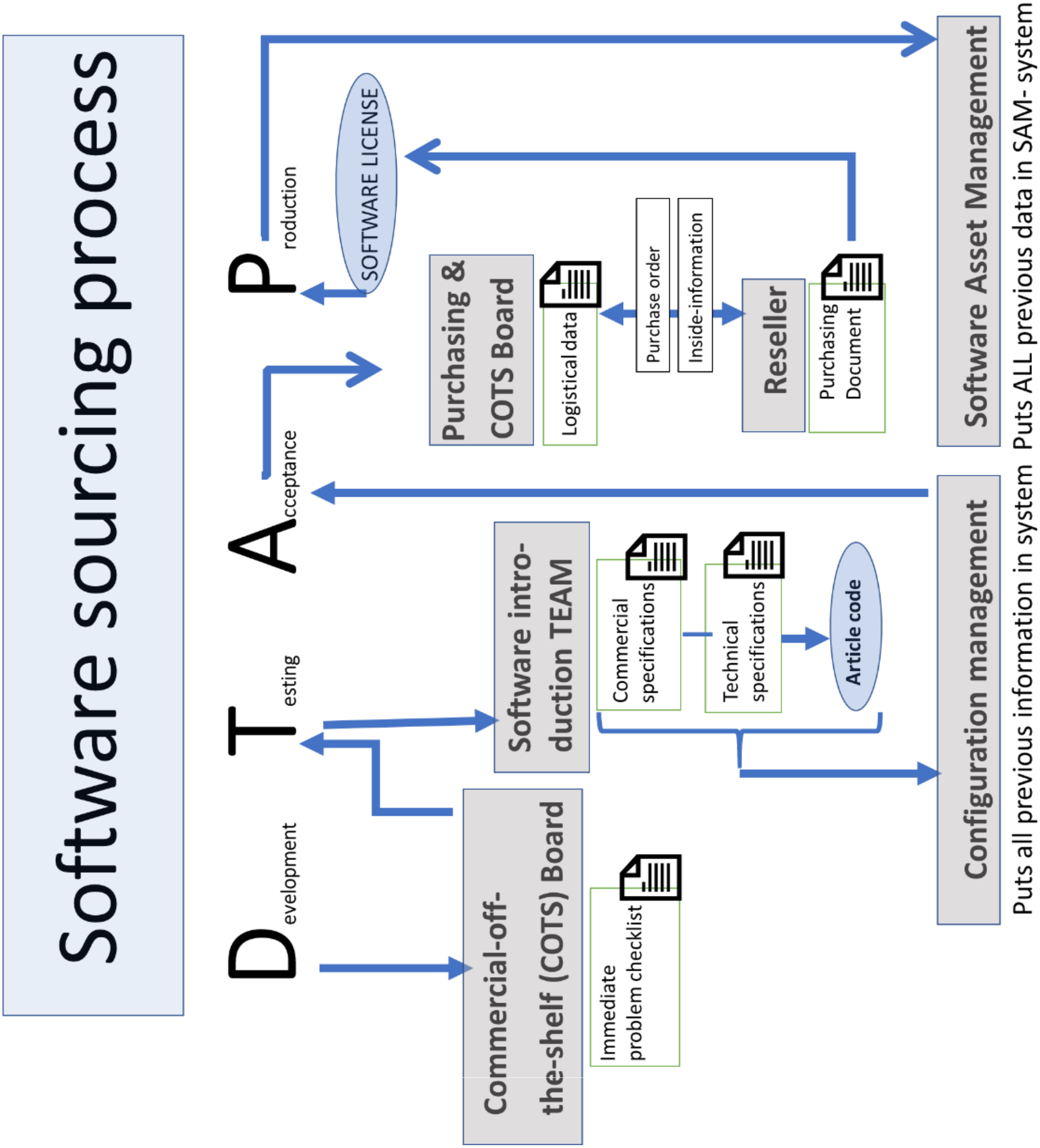
- How do you manage your software licenses? Do you use a database?
- Did you ever encounter problems with your licenses?

### Other companies:

- How is software bought in your company?
- How are the specific types of software (commercial-off-the-shelf, custom Build etc.) being bought in your company?
- If you use a reseller or direct link to the manufacturer, how is the process?
- How do you manage your software licenses? Do you use a database?

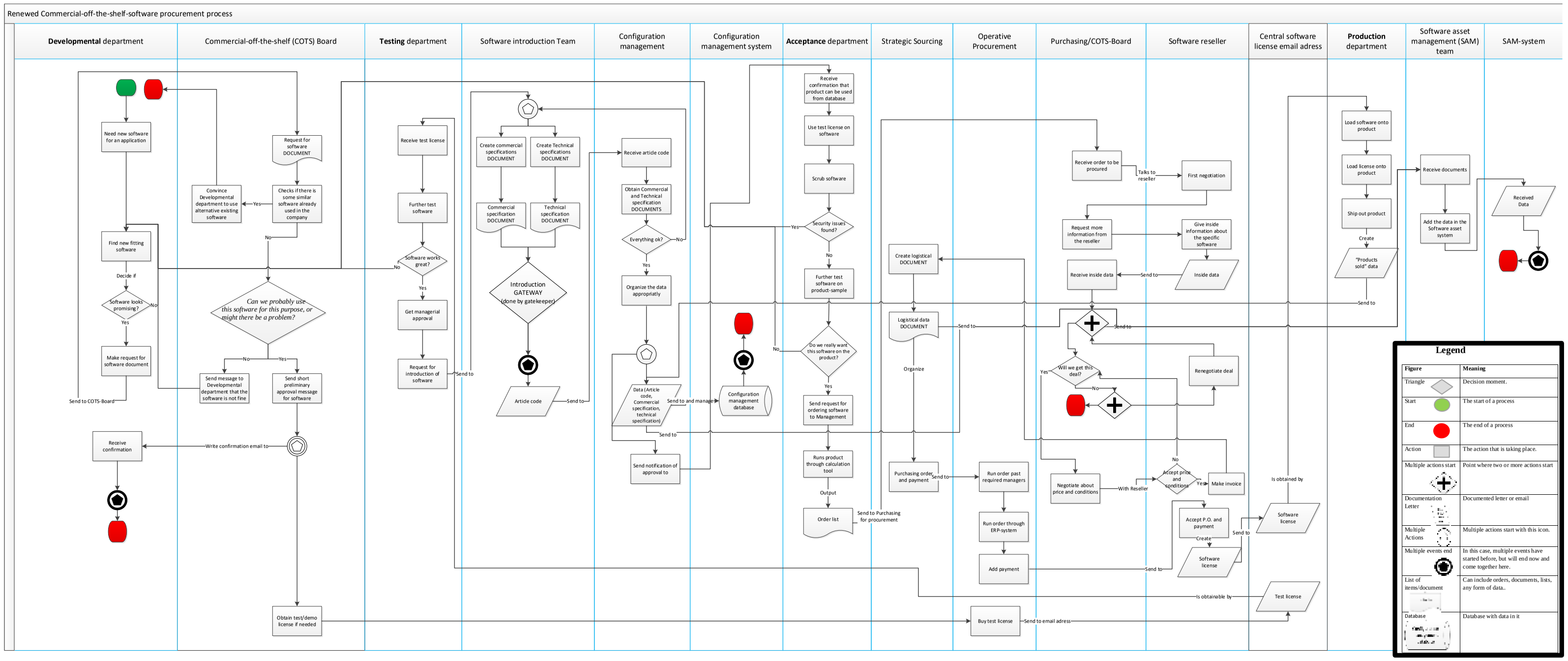
- Did you ever encounter problems with your licenses?
- What happens with software licenses/service that are about to be expired?
- Do you receive a notification to extend the license or support?
- Do you use a system for this, or a protocol/process?
- You load software on a product, what happens when you sell that product?
- Are you allowed to resell the license, or how do you manage that?

## APPENDIX LL
## New purchasing process for commercial-off-the-shelf software


Software sourcing process
Development
Testing
Acceptance
Production
Commercial-off-the-shelf (COTS) Board
Immediate problem checklist
Software intro-duction TEAM
Commercial specifications
Technical specifications
Article code
Purchasing & COTS Board
Logistical data
Purchase order
Inside-information
Reseller
Purchasing Document
SOFTWARE LICENSE
Configuration management
Puts all previous information in system
Software Asset Management
Puts ALL previous data in SAM- system

## APPENDIX PP

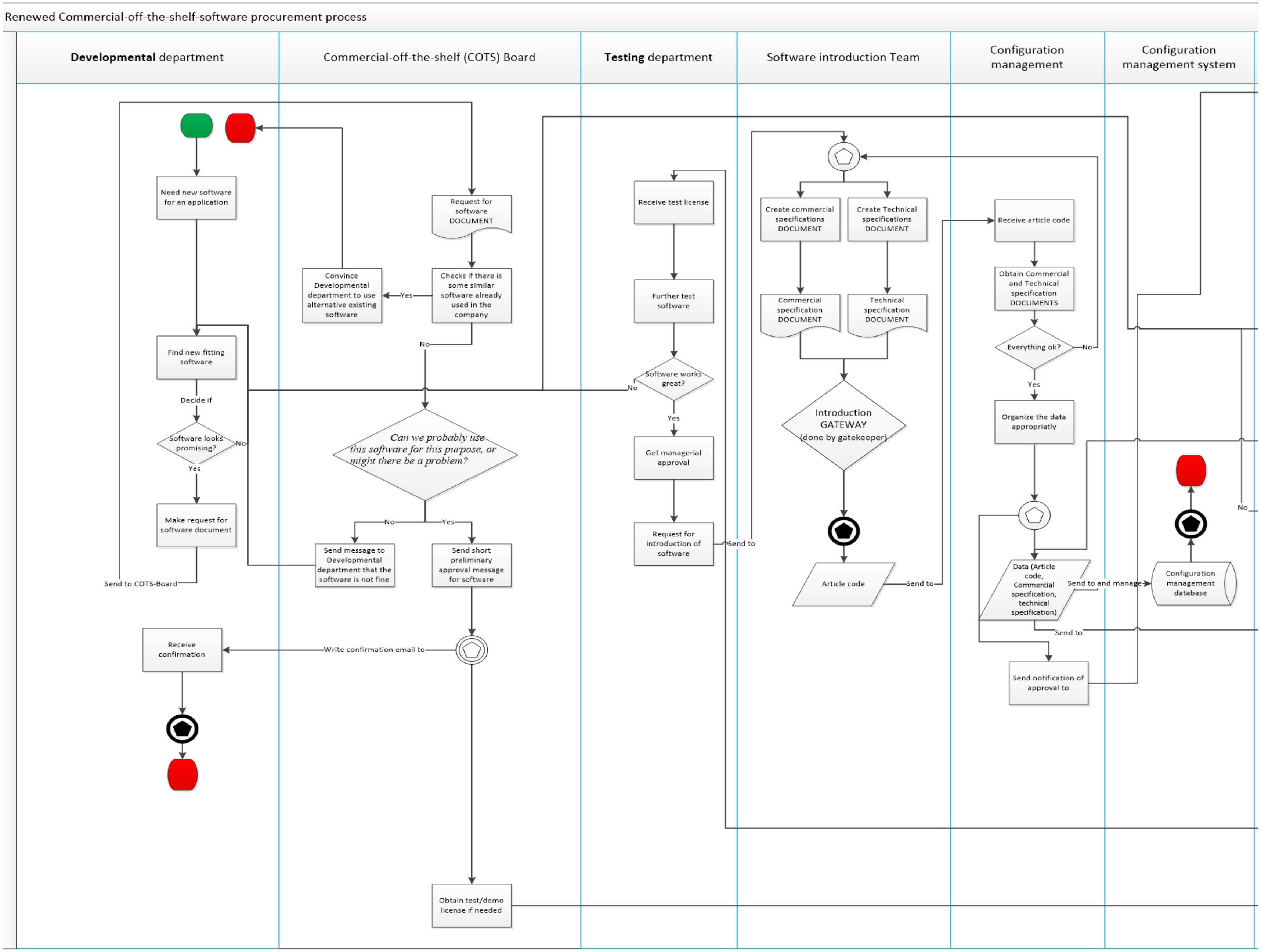
The renewed commercial-off-the-shelf procurement process (Microsoft Visio file)

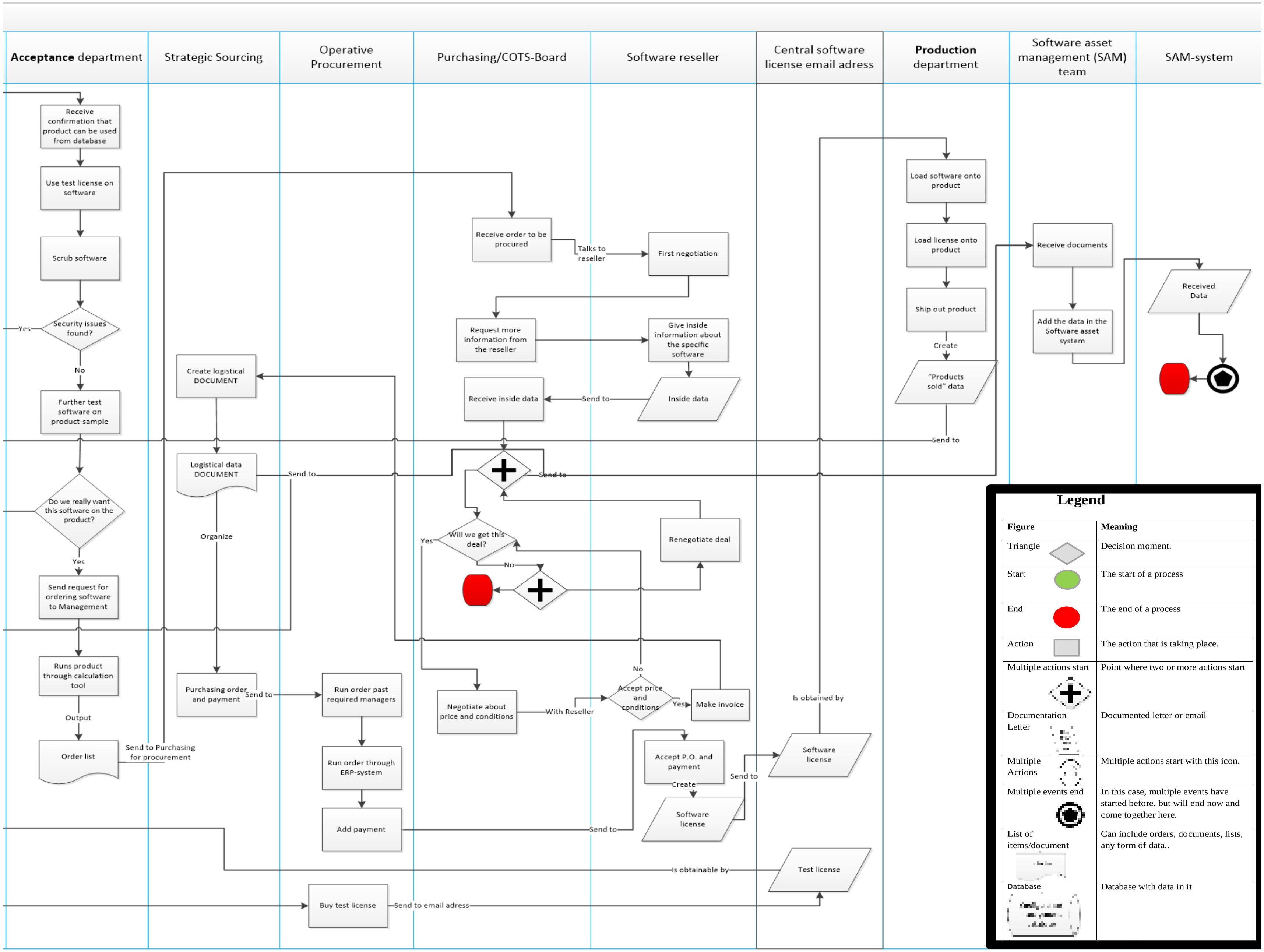

Renewed Commercial-off-the-shelf-software procurement process
Developmental department
Commercial-off-the-shelf (COTS) Board
Testing department
Software introduction Team
Configuration management
Configuration management system
Acceptance department
Strategic Sourcing
Operative Procurement
Purchasing/COTS-Board
Software reseller
Central software license email adress
Production department
Software asset management (SAM) team
SAM-system
Need new software for an application
Find new fitting software
Decide if
Software looks promising?
No
Yes
Make request for software document
Send to COTS-Board
Receive confirmation
Request for software DOCUMENT
Checks if there is some similar software already used in the company
Yes
Convince Developmental department to use alternative existing software
No
Can we probably use this software for this purpose, or might there be a problem?
No
Yes
Send message to Developmental department that the software is not fine
Send short preliminary approval message for software
Write confirmation email to
Obtain test/demo license if needed
Receive test license
Further test software
Software works great?
No
Yes
Get managerial approval
Request for introduction of software
Send to
Create commercial specifications DOCUMENT
Create Technical specifications DOCUMENT
Commercial specification DOCUMENT
Technical specification DOCUMENT
Introduction GATEWAY (done by gatekeeper)
Article code
Send to
Receive article code
Obtain Commercial and Technical specification DOCUMENTS
Everything ok?
No
Yes
Organize the data appropriatly
Data (Article code, Commercial specification, technical specification)
Send to and manage
Send to
Send notification of approval to
Configuration management database
Receive confirmation that product can be used from database
Use test license on software
Scrub software
Security issues found?
Yes
No
Further test software on product-sample
Do we really want this software on the product?
No
Yes
Send request for ordering software to Management
Runs product through calculation tool
Output
Order list
Send to Purchasing for procurement
Create logistical DOCUMENT
Logistical data DOCUMENT
Send to
Organize
Purchasing order and payment
Send to
Run order past required managers
Run order through ERP-system
Add payment
Send to
Buy test license
Send to email adress
Receive order to be procured
Talks to reseller
First negotiation
Request more information from the reseller
Give inside information about the specific software
Inside data
Send to
Receive inside data
Send to
Will we get this deal?
Yes
No
Renegotiate deal
Negotiate about price and conditions
With Reseller
Accept price and conditions
No
Yes
Make invoice
Accept P.O. and payment
Create
Software license
Send to
Software license
Is obtained by
Test license
Is obtainable by
Load software onto product
Load license onto product
Ship out product
Create
"Products sold" data
Send to
Receive documents
Add the data in the Software asset system
Received Data
Legend
Figure
Meaning
Triangle
Decision moment.
Start
The start of a process
End
The end of a process
Action
The action that is taking place.
Multiple actions start
Point where two or more actions start
Documentation Letter
Documented letter or email
Multiple Actions
Multiple actions start with this icon.
Multiple events end
In this case, multiple events have started before, but will end now and come together here.
List of items/document
Can include orders, documents, lists, any form of data..
Database
Database with data in it

## APPENDIX QQ

The renewed commercial off the shelf procurement process.

(On two pages) the photo output to make it readable in the printed version


Renewed Commercial-off-the-shelf-software procurement process
**Developmental** department
Commercial-off-the-shelf (COTS) Board
**Testing** department
Software introduction Team
Configuration management
Configuration management system
Need new software for an application
Find new fitting software
Decide if
Software looks promising?
No
Yes
Make request for software document
Send to COTS-Board
Receive confirmation
Request for software DOCUMENT
Checks if there is some similar software already used in the company
Yes
Convince Developmental department to use alternative existing software
No
Can we probably use this software for this purpose, or might there be a problem?
No
Yes
Send message to Developmental department that the software is not fine
Send short preliminary approval message for software
Write confirmation email to
Obtain test/demo license if needed
Receive test license
Further test software
Software works great?
No
Yes
Get managerial approval
Request for introduction of software
Send to
Create commercial specifications DOCUMENT
Create Technical specifications DOCUMENT
Commercial specification DOCUMENT
Technical specification DOCUMENT
Introduction GATEWAY (done by gatekeeper)
Article code
Send to
Receive article code
Obtain Commercial and Technical specification DOCUMENTS
Everything ok?
No
Yes
Organize the data appropriatly
Data (Article code, Commercial specification, technical specification)
Send to and manage
Send to
Send notification of approval to
Configuration management database
No


Acceptance department
Strategic Sourcing
Operative Procurement
Purchasing/COTS-Board
Software reseller
Central software license email adress
Production department
Software asset management (SAM) team
SAM-system
Receive confirmation that product can be used from database
Use test license on software
Scrub software
Security issues found?
Yes
No
Further test software on product-sample
Do we really want this software on the product?
Yes
Send request for ordering software to Management
Runs product through calculation tool
Output
Order list
Send to Purchasing for procurement
Create logistical DOCUMENT
Logistical data DOCUMENT
Send to
Organize
Purchasing order and payment
Send to
Run order past required managers
Run order through ERP-system
Add payment
Buy test license
Send to email adress
Receive order to be procured
Talks to reseller
First negotiation
Request more information from the reseller
Give inside information about the specific software
Receive inside data
Send to
Inside data
Send to
Will we get this deal?
Yes
No
Renegotiate deal
Negotiate about price and conditions
With Reseller
Accept price and conditions
No
Yes
Make invoice
Accept P.O. and payment
Create
Software license
Send to
Send to
Is obtainable by
Software license
Is obtained by
Test license
Load software onto product
Load license onto product
Ship out product
Create
"Products sold" data
Send to
Receive documents
Add the data in the Software asset system
Received Data

**Legend**

| Figure | Meaning |
|---|---|
| Triangle ◇ | Decision moment. |
| Start 🟢 | The start of a process |
| End 🔴 | The end of a process |
| Action ▭ | The action that is taking place. |
| Multiple actions start ✚ | Point where two or more actions start |
| Documentation Letter | Documented letter or email |
| Multiple Actions | Multiple actions start with this icon. |
| Multiple events end | In this case, multiple events have started before, but will end now and come together here. |
| List of items/document | Can include orders, documents, lists, any form of data.. |
| Database | Database with data in it |

**APPENDIX RR**
**Request for software DOCUMENT**

# Request for software DOCUMENT

| | |
|---|---|
| Name (Signature) | H. Schiele<br>~~*H.SchieLE*~~ |
| Manager (signature) | A. Sigurdatdottir<br>~~ASiGuRDarDottiR~~ |
| Date | 22-05-2019 |
| Department | Purchasing and supply management |
| Software | PlagScan GmbH plagiarism detector |
| Purpose | Will be used in the faculty to detect plagiarism in Master thesis's |
| Location | A virtual server on campus controlling all incoming online applications |
| Amount | 20-30 licenses |

**APPENDIX SS**
**You can probably use it for this purpose DOCUMENT**

# You can probably use it for this purpose DOCUMENT

| | |
|---|---|
| COTS-Board representative (signature) | F. Wynstra<br>~~F. Wynstra~~ |
| Date | 23-05-2019 |
| For Department | Purchasing and supply management |
| Software | PlagScan GmbH plagiarism detector |
| Purpose | Will be used in the faculty to detect plagiarism in Master thesis's |
| For Location | A virtual server on campus controlling all incoming online applications |
| Q1)<br>Q2)<br>Q3)<br>Q4)<br>Q5) | Is it allowed to use the software on a virtual server? – **No problem**<br>Is it allowed to use plagiarism software at all? **– according to X section B, no problem**<br>Is there a known better alternative? **- Not that we know of**<br>Can we find weaknesses or problems with this software? – **No, not on google**<br>Does the manufacturer allow the software to be used for the educational system? – **Within the EU there seems to be no problem, outside the EU it's a problem** |
| Status | Accepted |
| Note | If the product will be shipped outside the EU, there will be a problem. As the application is currently the university of Twente, and the estimated usage seems to be no within the scope of the current EULA, we can give the indication that there will probably be no problem. |

## APPENDIX TT
## Commercial specifications DOCUMENT

Appendix TT concerns a filled in template of the ‘commercial specifications DOCUMENT’ of software procurement.

## Commercial specifications DOCUMENT

| *Name of item* | **Example** | **Person to mainly fill it in/Support from** |
|---|---|---|
| **Standard specifications** | | |
| **Date** | 28-05-2019 | IT purchaser |
| **Manufacturer** | Hewlett Packard | IT-purchasing /Software engineer support with knowledge of Purchasing |
| **Manufacturers product code** | HPN03281953-ADV | ‘’ |
| **Name** | HP Network node manager 3.1.65 | ‘’ |
| **Open source/Commercial-off-the-shelf?** | [Commercial-off-the-shelf] [GPL 2.0] | ‘’ |
| **License type (per)** | [server], [node], [core], [user], [Unlimited use], [year], [Freeware] | ‘’ |
| **(Re)seller** | [Insight][Comsoft] [ Hewlett Packard] | IT-purchasing |
| **Reseller product code** | 030219894 | IT-Purchasing |
| **Expected cost price** | [30 Euro per license per node] [Two hundred euros for a department] | IT-Purchasing |
| **Name and Signature** | J. Telgen<br>J. Telgen | IT-Purchasing |
| **Risks and control** | | |
| **Risk of fines** | Big fines in the past for American airlines | Legal expert |
| **Manufacturer right to verify actual licenses in use** | [Yes][No] [Monitoring tool] | Legal expert/ Software engineer with knowledge of purchasing |
| **Export control** | [Not export controlled], [Only allowed in non-military applications]<br>[Origin from Japan, not to be used for military purposes.] | Export control |
| **Notes** | [Watch out, buy in packs of 40], [Little support, but at the time the only supplier, look for alternatives], | Export control/ Purchasing; Whoever has information |
| **Name and signature** | A. Van Weele<br>A. Van Weele | Export control |

**APPENDIX UU**
**Technical specifications DOCUMENT**

# Technical specifications DOCUMENT

| *Name of item* | **Example** | **Person to mainly fill it in** |
|---|---|---|
| **Support YES/NO** | [Yes]/[No] | Software engineer support with knowledge of Purchasing |
| **Support type** | [24/7]/ [9/5]/] Mandatory payment]/ [Training is paid for] | Software engineer support with knowledge of Purchasing |
| **Updates/Evolution** | [Included are updates for 10 years]/ [Updates not available for new Operating system] | Software engineer support with knowledge of Purchasing |
| **System interaction** | [The software interacts with Microsoft Paint so that it crashes the server] | Software engineer support with knowledge of Purchasing |
| **Installation manual** | 1) Install software<br>2) get to the manufacture's website<br>3) click on 'license codes'<br>4) Log in with "**eleihcsh**" as username and "**rittodradrugisga123'** as password<br>5) Download license code<br>6) Insert license code on product | Software engineer support with knowledge of Purchasing |
| **End User Licence Agreement (EULA)** | 📎<br>*Copy of the End user license agreement* | Software engineer support with knowledge of Purchasing |
| **Functionality** | 1. Can create databases out of Excel sheets<br>2. Cannot connect to China<br>3. Links servers together | Software engineer support with knowledge of Purchasing |
| **Operating system** | [Only MS]/[Ubuntu 18.04.02 LTS]/[OSX]/[Non-compatible with Windows below 8.1] | Software engineer support with knowledge of Purchasing |
| **Purpose** | [Monitoring software]/ [Telnet] / [Virtual machine] | Manager/infrastructure designer |
| **Name (Signature)** | F. Vos<br>~~F. Vos~~ | (IT-)Manager |

## APPENDIX VV
## Introduction gateway

| *Name of item* | **Example** | **Responsible person/Getting assistance from** |
|---|---|---|
| **Commercial specifications in line with other documents and complete?** | [Everything is good] | Gatekeeper/Export control |
| **Logistical data in line with other documents and complete?** | [Everything is good] | Gatekeeper/IT-Purchaser |
| **Technical specifications in line with other documents and complete?** | [Everything is good] | Gatekeeper/IT-Manager |
| **Notes** | There seems to be a risk with using windows server, but in the foreseeable future we don't expect to use windows server | Gatekeeper |
| **Item save to receive an article code** | Positive | Gatekeeper/Configuration manager |
| **Name (Signature)** | J. Gugler<br>~~J. Gugler~~ | Gatekeeper |

**APPENDIX WW**

**Example form for Logistical data document regarding software procurement**

# Logistical data DOCUMENT

| *Item* | **Example** | **Person to mainly fill it in** |
|---|---|---|
| **Purchasing order Number** | PO235098235089 | IT-Purchaser |
| **Purchasing order** | 📎<br>Attached Purchasing order | IT-Purchaser |
| **Cost per license** | 237 Euro | IT-Purchaser |
| **Date of purchase** | 03-03-2018 | IT-Purchaser |
| **Licenses bought** | [10] licenses | IT-Purchaser |
| **Quotation/tender** | 📎<br>Document of Tender | IT-Purchaser |
| **Contract** | “We are allowed to pass the software on to the customer, but only after 6 months.” | IT-Purchaser |
| **Proof of Purchase** | 📎<br>Receipt | IT-Purchaser |
| **Start of License** | [01-01-2019]/[When in use]/[When downloaded] | IT-Purchaser |
| **End of License** | [03-03-2036]/[Unlimited]/[23 years] | IT-Purchaser |
| **Support** | Only bugfixes, the next version will cost money | IT-Purchaser |

| | | |
|---|---|---|
| **End of support** | 03-03-2023 | IT-Purchaser |
| **Own use/<br>Customer use** | [Customer use] | IT-Purchaser |
| **For Project …** | TNL-Higher purpose to be specified if more than one. | IT-Purchaser |
| **Signature** | **R. Loohuis**<br>**~~R. Loohuis~~** | Purchasing manager |