NTNU – Trondheim
Norwegian University of
Science and Technology

A Study of Distribution of IT-work in Norwegian Organizations

Tor Kristian In'T Veld

Master of Science in Informatics
Submission date: June 2014
Supervisor: John Krogstie, IDI

Norwegian University of Science and Technology
Department of Computer and Information Science

Abstract

The distribution of work in an organization's IT-department has been changing the last 20 years. More time is spent on maintenance than development. Maintenance has become a great expense towards cost and time, but are often not prioritized. What is the reason for this? And are there other factors in the organizations influencing the distribution of work?
This thesis is part of a replication study performed every fifth year since 1993. A survey was conducted in 2013 with 68 Norwegian organizations, gathering data about and in relevance to distribution of work in the IT-departments. The thesis presents the results and compares different factors up against each other to find correlations. Results are also compared to the other surveys in this replication study, in order to find trends and reasons for change over time. This study has also had a focus on differences between private and public organizations
The main results of this thesis, was that maintenance has continued its increasing trend. When comparing maintenance and development work isolated, 78% was spent on maintenance. This is a significant increase since last study in 2008 when it was 66%. Application portfolio upkeep has also continued to increase, and was at 68% (2008 - 63%). The conclusion for these results, are the major use of outsourcing. Organizations outsource a large part of the development, but keep the maintenance in-house. This may be why there has been a decrease in application portfolio evolution and an increase in upkeep. This study has had a majority of public organizations, which outsourced more than the rest of the population. It is recommended to do further investigation on the use of outsourcing and on differences between public and private organizations.

Sammendrag

Fordelingen av arbeid i en organisasjons IT-avdeling har forandret seg de siste 20 årene. Mer tid er brukt på vedlikehold enn utvikling. Vedlikehold har blitt en stor bekostnig på både kostnader og tid, men er likevel ikke alltid prioritert. Hva er grunnen til dette? Og er det andre faktorer innad i organisasjonene som påvirker fordelingen av arbeid?
Denne avhandlingen er en del av en replikasjonsstudie utført hver femte år siden 1993. En spørreundersøkelse var utført i 2013 i 68 Norske organisasjoner, som samlet inn data om og i relevanse til fordelingen av arbeid i IT-avdelinger. Denne avhandlnigen presenterer resultatene og sammenligner forskjellige faktorer opp mot hverandre for å finne likheter. Resultatene er også sammenlignet opp mot andre undersøkelser i dette replikasjonsstudiet, for å finne trender og grunner til forandring over tid. Denne undersøkelsen har også hatt et fokus på forskjeller mellom privat og offentlig sektor.

Hovedresultatene i denne avhandlingen, var at vedlikehold har fortsatt sin stigende trend. Når isolert vedlikehold og utvikling sammenlignes, ble 78% av tiden brukt på vedlikehold, som er det høyeste resultatet målt de siste 20 år. Dette var en significant økning fra 2008 undersøkelsen hvor isolert vedlikehold lå på 66%.
andelen applikasjonsportefølje vedlikehold (upkeep) var på 68% (2008 - 63%), som er en liten økning etter å ha ligget relativt stabil i en periode. Konklusjonen for disse resultatene, var en større bruk av outsourcing av vedlikehold. Organisasjoner outsourcer en større del av utviklingen, men utfører vedlikehold innad i organisasjonen. Dette er årsaken til at applikasjoneportefølge vedlikehold (upkeep) har økt, mens applikasjonsportefølge evolusjon har blitt redusert.
Det er foreslått å undersøke dypere på bruk av outsourcing og forskjell mellom privat og offentlig sektor, samt teknologi som forandrer fordelingen av arbeid innenfor IT.

Preface

This master's thesis is written in the course IT3902 - Informatics postgraduate thesis: Information Management, written from August 2013 to May 2014 at the department of Computer and Information Science, and submitted to the Norwegian University of Science and Technology (NTNU).

My thesis is an empirical investigation on distribution of work in IT-departments. It is part of a replication study started by Professor John Krogstie in 1993. There has also in this study been a cooperation with 'IT i praksis'.

I would like to thank my supervisor Professor John Krogstie for his guidelines and feedback. His dedication to this study has been crucial for the investigation. I also have to thank him for his patience when answering all my questions in our long meetings and frequent mail correspondences.

Trondheim, June 1st 2014

____

Tor Kristian in't Veld

Contents

Chapter 1

Introduction

This chapter will serve as an introduction to this thesis by introducing the problem definition, motivation for performing a solution and a presentation of the rest of this thesis.

1.1 Motivation

What was before simply called a software program used as a tool to perform different tasks, are today complex and enormous systems. Organizations have systems integrated across departments, where each and one serve its own purpose, but together form the architecture of the organization. These complex systems have revolutionized software development. Dozens of system developers can spend years on a system, with a budget of hundreds of millions. However, when the system is put in production, the life cycle has just begun. A system can stay in production for 10-30 years. Maintaining the mission critical software of an organization is not an easy task. It requires existence of great maintenance management, as well as maximizing strategic impact and optimizing the cost of maintenance activities. To achieve this requires the organizations to be committed to the maintenance processes.
S. Colter wrote in 1987: "The greatest problem of software maintenance is not technical, but managerial". Earlier studies blame bad maintenance on lack of documentation and poorly designed code [21]. Others blame this on bad processes and management [41]. What ever the reason is, either if it is managerial or technical, internal or external, this replication study shows that for the last 20 years, the share of maintenance has exceeded the share of devel-

opment during an organization's distribution of work. Maintenance is often seen upon as a burden, or as negative work. But to keep the system in a competitive state and in accordance to business strategy, consistency is a necessity. The system must evolve in parallel with the market. If the system is not up to date, it must be replaced with a new system. Maintenance performed in accordance to keeping the system competitive is both positive and a necessity. On the other side, there are maintenance addressing system errors and minor modifications. Such maintenance could be avoided in the development process, and is therefore considered as a more negative share of work. This tells us that there is positive (evolutionary) and negative (unnecessary) maintenance.

Several problems in software maintenance are caused by a lack of knowledge on the area[29]. This includes the maintenance processes and the effect they have on software evolution. A survey reported that only two percent of empirical studies focus on maintenance[29]. To have a well performing maintenance process: 1. helps the organization spend less time and cost in the software lifecycle and 2. failure to change software quickly and reliably can influence the business processes[5].
When gathering and analysing the distribution of work in organizations, together with other relevant factors, it could be possible to both address a potential problem, but also map correlations to trends and results. These factors could be business or IT-strategy, technologies used, methodologies used or simply the structure of the organization.

1.2 Problem Formulation

In 1993, 1998, 2003 and 2008 surveys were performed to investigate development and maintenance of IT-systems in Norwegian organizations. Data collection for a similar survey is to be carried out in 2013. The assignment will be to perform data collection in a survey investigation and accompanying case studies in Norwegian organizations. The data obtained are to be analysed.
Together with a literature review, the survey investigation is expected to give us new knowledge about mechanisms affecting resource utilization related to information systems support

in organizations. The report is expected to form the basis for scientific publications

1.3 Context

This thesis is a replication study based on Krogstie's investigations in 1993, 1998, 2003 and 2008. All studies were built around a survey, sent out to Norwegian organizations. These studies have mainly focused on the share of maintenance at that time, and have been compared to previous studies. By doing this, trends and a better overview of the situation were obtained.

This study also contains a survey, but in addition there was established a collaboration with 'IT i praksis'. 'IT i praksis' is another national survey performed every year to identify the situation in the IT-business. Comparing our data with this survey could support the results and analysis obtained, because it expands the population and investigation-area. This study also has a bigger focus on public vs private sector, and the entire distribution of work.

1.4 Report structure

This thesis is organized into 11 chapters. The first chapter is an introduction to the research problem and the motivation behind the thesis. The second chapter presents relevant background information for the surrounding subjects that build this thesis. Chapter three is a presentation of the other studies in this replication study, namely related work, and chapter four gives an evaluation and a presentation of the used research method.

In the next part, namely chapter 6 and 7, descriptive results from this and the 'IT i praksis' survey are presented. In the final part, these results are discussed. In the hypothesis-testing chapter, analysis results are discussed towards specific hypothesis. The discussion chapter contains discussion on a higher level together with more comparisons to related studies. Finally, chapter 10 is a chapter dedicated to evaluation of the entire investigation. This chapter evaluates what was done, collected and discussed. The last chapter contains the conclusion, which is a summary of the main findings.

Chapter 2

Background

This chapter builds a foundation for this thesis by presenting relevant concepts and theory. It will also explain different terms used in the thesis. This chapter will give the reader an overall knowledge of the state-of-practice for why this investigation is performed.

2.1 Software evolution

Software Evolution is a term used to describe the continuous change in a software systems after it has been released. For an actively used software system, it is important that it adapts to its environment. This way it can stay useful and competitive in an environment that is in continuous change. All software that is useful and stimulates user-requests for change and improvements initiates software evolution. This means that software evolution only takes place when initial development was successful[5]. The software life cycle both pre and post release has been discussed since the 1960s. The well known Swanson & Lientz wrote in 1980 that maintenance consists largely of continued development[35]. As early as 1976 Lehman introduced Lehman's laws and their definition that software evolution is "The dynamic behaviour of systems as they are maintained and enhanced over their life times"[4]. Maintenance refers to activities that take place after the system is implemented. Software evolution is defined as examining the dynamic behaviour of systems and how they change over time[29].

Years of dynamic change by different developers, can cause disordered code and architecture.

Eventually, this can result in maintenance that is harder to conduct. C. Jones addressed this problem as *software entropy*[9]. He wrote: "All known compound objects decay and become more complex with the passage of time unless effort is exerted to keep them repaired and updated"[9]. If systems are poorly developed, they will more likely need more frequent maintenance. The work on maintenance will also be more time consuming, harder to conduct and will create more unstructured architecture. For every year, defect repairs and updates will degrade the original structure. Further, this will also make new changes more difficult to conduct. In the end, maintenance can only be performed by a few experts of the system[9]. It is therefore important to have knowledge of software architecture and on software team work[5]. This way they can make changes in the software without damaging the architectural integrity. When this is not possible anymore, the system goes into a new stage called servicing. In this stage, only minor changes like patches and code changes are executed. There may be different reasons for why a system goes into the servicing stage. To maintain a complex system requires knowledge necessary for evolution. The loss of knowledge is usually associated with loss of key personnel[5]. Another reason is that the system may not be the central product for the organization any more. The organizations have then moved on to different business strategies or obtained new systems. A system may also go into servicing stage because of business aspects, an example being that changes may result in the loss of architectural integrity.

All the different evolution stages are displayed in *Figure 2.1.* After the servicing stage there is the phase-out stage, where no more services are conducted. The patches are considered too costly considering the minor benefits it gives. The system is only alive because it is used as archive or used because it is integrated in other systems. The final stage is the close-down stage, where the system goes trough the process of a clean take-down (makes sure everything needed is saved and that it is not integrated in anything else).

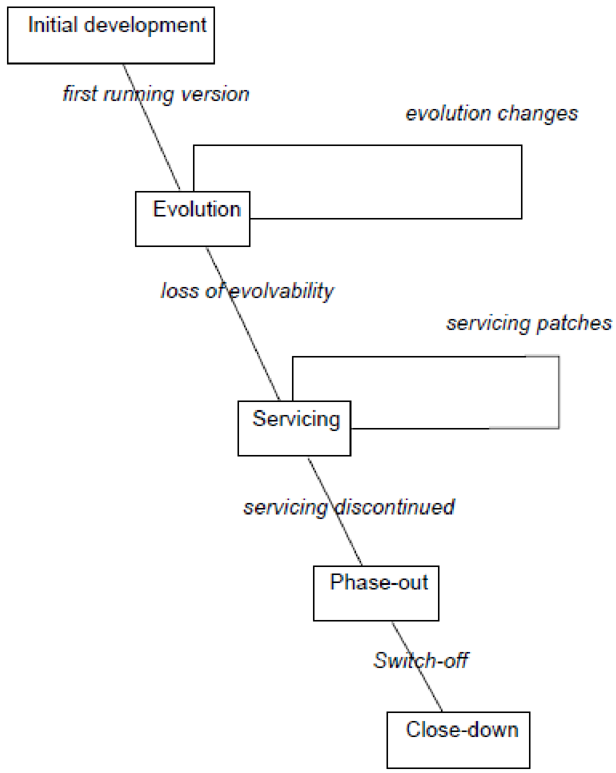

Figure 2.1: The simple staged model for software evolution [5]

To ensure well-performed evolution during a long period of time, the mentioned aspects in this section are important. Coherent architecture with well-thought patterns and a structured code are necessities for simplicity of maintenance. After the system is put in production, it is important to keep the knowledge in the team. If all developers are replaced, knowledge necessary for evolution is replaced with new unfamiliar developers. It is then even more important to use good methods and tools during both development and maintenance.

Lehman's laws

In the 1970s, Belady and Lehman studied releases of the OS/360 operating systems, twelve years after its release[4]. In this study they observed many observations about the size- and complexity growth of the system. They conducted that software is set to change during its existence. These observations made them deduct three laws about software evolution and how changes influence them. Over the years, Lehman has revisited the original laws, and made new ones. Some of the laws are experience based, and some may sound obvious. However, they are important for the integrity and as guidelines. Today there are a total of 8

Lehman's laws addressing evolutionary-type (e-type) systems. An e-type system is a system that operates in the real world and is bound to gain modifications during its life cycle[44]. Lehman's laws are listed below:

1. Continuing Change — An E-type system must be continually adapted or it becomes progressively less satisfactory.

2. Increasing Complexity — As an E-type system evolves its complexity increases unless work is done to maintain or reduce it.

3. Self Regulation — E-type system evolution process is self-regulating with distribution of product and process measures close to normal.

4. Conservation of Organisational Stability (invariant work rate) - The average effective global activity rate in an evolving E-type system is invariant over product lifetime.

5. Conservation of Familiarity — As an E-type system evolves all associated with it, developers, sales personnel, users, for example, must maintain mastery of its content and behaviour to achieve satisfactory evolution. Excessive growth diminishes that mastery. Hence the average incremental growth remains invariant as the system evolves.

6. Continuing Growth — The functional content of an E-type system must be continually increased to maintain user satisfaction over its lifetime.

7. Declining Quality — The quality of an E-type system will appear to be declining unless it is rigorously maintained and adapted to operational environment changes.

8. Feedback System (first stated 1974, formalised as law 1996) — E-type evolution processes constitute multi-level, multi-loop, multi-agent feedback systems and must be treated as such to achieve significant improvement over any reasonable base.

2.2 Definition of maintenance

Maintenance is part of the software evolution, and must therefore always be taken into consideration. However, there are many ways to define the concept. Software maintenance is by the The Institute of Electrical and Electronics Engineers (IEEE) standard defined as:

> *"The process of modifying a software system or component after delivery to correct faults, improve performance or other attributes, or adapt to a changed environment"[25].*

Maintenance can be categorized into different types. Already in 1976, Swanson categorized maintenance into adaptive, corrective and perfective maintenance [46]. These categories have been applied to the IEEE standards[26] and are still being practised today. The standard IEEE definition define corrective, adaptive and perfective maintenance as this:

1. *Corrective maintenance:* The reactive modification of a software product performed after delivery to correct discovered problems.

2. *Adaptive maintenance:* Modification performed after delivery, to provide enhancements necessary to keep the software product usable in a changing environment.

3. *Perfective maintenance* The modification of a software product after delivery, to detect and correct latent faults in the software product before they are manifested as failures. Examples can be to improve performance, maintainability, or other software attributes. Perfective maintenance can also be divided into *Enhancive maintenance*[11] and *non-functional perfective maintenance*[32].

   Enhancive maintenance extends or expands functionality, or adds new data flows to/from the system[11].

   Non-functional perfective maintenance is changes that are in favour of the developer or maintainer of the system. An example can be to improve modifiability.

Some also say that *Preventive maintenance* is a type of maintenance. Preventive maintenance is modifications performed to prevent problems before they occur [26]. Some say this is a maintenance type of its own, while others mean it can not be categorized as an individual maintenance type, but that it is part of perfective maintenance [15][10]. Both Swanson and IEEE standards have addressed the definition preventive maintenance, but never claimed it to be a maintenance type of its own.

In 1995 Krogstie introduced a new concept in addition to the traditional maintenance types (corrective, adaptive and perfective)[30]. The new concepts were *application portfolio upkeep* and *application portfolio evolution*. These were added because they can give a better indication of the efficiency of the application system support in an organization, compared to the traditional types[31]. The [30] definitions are listed below

1. Application portfolio evolution: Development or maintenance where changes in the application increase the functional coverage of the total application systems portfolio of the organization. This include development of new systems that cover new functions, and enhancive maintenance.

2. Application portfolio upkeep: Modifications done to keep up the functional coverage of the system portfolio of the organization. This includes the standard corrective, adaptive, non-functional perfective, and also development of replacement systems.

Figure 2.2 illustrates the relationships between maintenance and development, and between Application portfolio upkeep and Application portfolio evolution. As the figure illustrates, upkeep and evolution go across maintenance and development. Upkeep can be considered functional maintenance, and evolution is functional development.

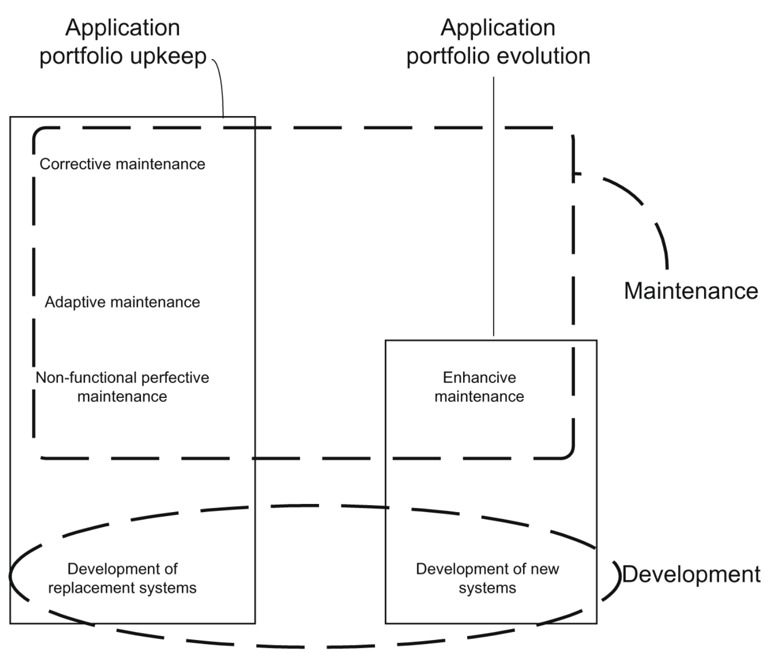

Figure 2.2: Relationships between development and maintenance [30]

2.3 Development models

A software process model is activities and associated information that are required to develop a software system[45]. Every organization that develops software uses some sort of development model, indirectly or directly. But there are some individual models that follow an abstract process. Some common process models will be presented in this section.

2.3.1 Code and fix

Code and fix, often called cowboys coding, is developing code without a design or plan. The developers immediately begin do produce code. And after some time, they start testing. This goes back and forth until the system is finished. It can be effective with minor software programs and with few and experienced developers. Using this model when developing complex systems in larger teams are almost impossible to do without complications. The finished product is most likely a challenge to maintain. Because there are no structure, design or planning in this model, some deny calling it a development model.[42]

2.3.2 Agile/Scrum

Scrum is an agile development model. Agile is a method based on iterative and incremental development. First, the team creates a plan for what needs to be done throughout the project. The project is then organized in cycles called sprints (2-4 weeks). Before every sprint, the team selects what tasks should be done in that sprint. Usually the highest prioritized tasks are implemented first. At the end of each sprint, the team makes an evaluation, a plan for the next sprint, and hopefully is able to send an updated product increment to the customer. The scrum team consists of members with different roles. These roles have different responsibilities throughout the project. There are also internal daily meetings and fixed weekly or biweekly meetings with the customer. This forces communication and keeps all associates continuously updated.[19]

2.3.3 Waterfall model

The waterfall model is a methodology where developers go through different phases during the entire process. They start with system requirements and end up in product release. When they are finished with one phase, they go to the next one. It is also possible to go back to a previous phase if necessary. The model is called waterfall because the progress "flows" from one phase to the next.
The staged processes enforce discipline, with a defined start and end point for every phase. With this the team always knows what stage they are in. Doing the requirements and design first, improves quality. The waterfall method is criticized on different points. The customer does not always know what he wants before the project. So finishing the planning stage first, and then go over to the implementation, there is no interaction with the customer who might change his mind as the stages go by. Because of this, some say that the waterfall is a model that only works in theory, but not in real life.[24]

2.3.4 Spiral model

The spiral model is a risk-driven process model. It got its name because the process goes through the same stages for every increment by going in a spiral[7]. The spiral model builds on the waterfall and incremental model. There are four main phases in an increment. Determine objectives, identify and resolves risks, development and test, and plan the next iteration. The idea is to build on a prototype for every increment, until the operational prototype is finished.

2.3.5 Kanban

Kanban is an approach to incremental, evolutionary processes and system changes for organizations. It is also considered to be a change management method. Kanban is a simple approach to follow. It does not have specific roles or process steps. However, it does follow four principles. It "starts with what you got", meaning that it starts with what roles and processes you already have, and then simulates incremental changes to the development. It then pursues these incremental changes until they are finished. Kanban works in theory with tasks moving through a pipeline. Requests move into the pipe from one side, and from there they go through different stages. These stages can be planning and development. At the end, the request leaves the pipeline as an improved software. In practice, this pipeline is a board, with the tasks written down on cards. The cards are then moved along the board until it is deployed as software.

2.4 Maintenance models

Maintenance models are different from development models. In development models, there is always a planning stage. What needs to be done is planned here. It is also known (approximately) when every process is starting and ending. In maintenance models, errors and modification requests come in randomly. They are not managed by using project management techniques. The only way to plan for maintenance is by either using queue management techniques or priority techniques.
The size or complexity of maintenance tasks are often so small that they can be executed by one or two persons. Bugs and errors may also be deeply integrated in the system. This means that the maintenance developers need much knowledge of the entire system.

The *Figure 2.3* from [1] illustrates the interfaces that the software maintainers interact with. The "Customers and Users interface" is the most important one. This interface is the client, and is central in Software development, Software maintenance, and Infrastructure and Operations. Infrastructure and Operations handles support and maintenance. It is also in control of backups, recovery and systems administrations.
Number 4 in *Figure 2.3* shows the interface between maintainers and suppliers. There is often a long list of suppliers to large systems. Examples are network systems, ERP vendors, cloud suppliers, outsourcing organizations or subcontractors. Maintainers have a relationship and understanding with all suppliers to be able to manage them efficiently, and to ensure that the tasks are performed[1].
There is a flow of requests circulating between users, help desk and maintainers (number 5 in *Figure 2.3*). For this interface to be effective, an efficient communication flow is needed for quick resolution of failures [1]. This could be done with a mechanized problem resolution.

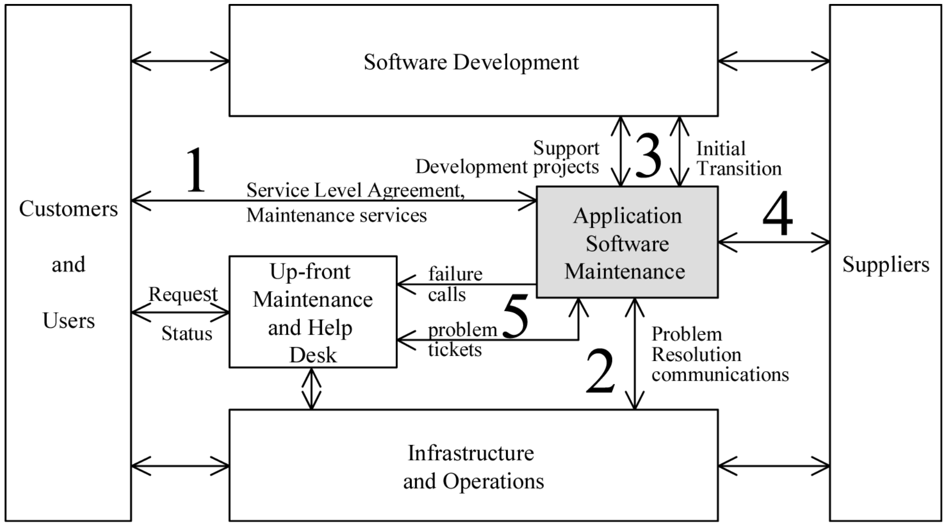

Figure 2.3: Software Maintainers Context Diagram[1]

The rest of this section will give a short presentation of some maintenance models.

2.4.1 Quick-fix model

The quick-fix model is a maintenance model where the developer simply takes the source code, locates the code in need of maintenance, performs the maintenance, and recompiles the system as a new version. The developer often do this without looking at requirements, design or documentation. Towards larger maintenance tasks, this model may do more harm than good. But for a minor bug fix, involving a single component, this method is efficient. [3]

Software Maintenance Maturity Model

Software Maintenance Maturity Model (SM^{mm}) is a maturity model for daily software maintenance activities[1]. SM^{mm} was designed as a customer-focused reference model. It has a focus on auditing the software maintenance capabilities of software suppliers, and to improve internal software maintenance organizations[1]. The model's structure is similar to the Capability Maturity Model Integration (CMMi)[1], and is designed to be a complement to it. In addition to CMMi, SM^{mm} has also taken best practices from other maturity models, including Camelia Maturity Model, Software Enginering Institute (SEI) and Related Technology

[1]CMMi: `http://en.wikipedia.org/wiki/Capability_Maturity_Model_Integration`

(CobIT) and Cm3-Corrective Maintenance Model. Combining so many models into one, results in a new large model. The four domains are:

1. Software maintenance process management
2. Software maintenance request management
3. Software evolution engineering
4. Support to software evolution engineering

This comprehensive structure gives an indication that this model may be very complex, and should be used accordingly. The SM^{mm} can result in lower maintenance and support costs, shorter intervals in maintenance, and increased ability to achieve service levels[1].

2.4.2 Full reuse model

The full reuse model starts with the requirements for the new system, and reuses as much of the old system as is practical. It then builds a new system using documents and components from the old system. It is also normal to use components and documents from other systems available. By reusing the best parts of many systems, the process will be more efficient and the new system better. [3]

2.5 What may affect work distribution

Earlier in this chapter there was a presentation of studies that indirectly may influence the distribution of work. Examples are how a system is developed, use of methods and models, or business strategies. This section gives a presentation of some more direct factors that may influence the distribution of work.

2.5.1 Dynamics of software maintenance

A software's evolution influences the software life-cycle in many aspects. Sometimes the system is developed by one group of people, and then maintained by another. In other cases, information regarding system size, complexity, reliability or maintainability is often missing. It is difficult to evaluate how much maintenance a system will need and how much it will cost after its implementation. The expenditures are far from over when the system is put in production. During a system's life cycle, only 25-33% of the total effort is conducted during the implementation [49].

Another factor that may affect distribution of maintenance is management attitude. Management allocates tasks, and maintenance is often the labour that is reduced when there is a time/budget pressure[6]. Organizations that are proactive in using maintenance tools and services can spend less than 30% of their software budgets on various forms of maintenance, while organizations that have not re-used any of these tools and services may spend more than 60% of their budget[9].

The code baseline may affect maintenance work for its entire life cycle. Bad architecture, program structure or documentation can make modification difficult and time consuming[6]. Programmers may often struggle with understanding the function of the code. An investigation discovered that developers spent the same amount of time studying the code, as they did on implementing the changes[18]. Code that is easy to understand and well documented may reduce this time.

Maintenance work can also be influenced by business factors. There may be changes in how the organization work that influence new changes in the system.

To evaluate maintenance year by year is not easy. This is because maintenance is not always linear with respect of time[6]. Maintenance is a dynamic process that is affected by external factors mentioned in this section. In *Figure 2.4* below, maintenance unit effort per day is presented over time. As time goes by, the system's maintenance is influenced by external factors. When the system is "out of date", it goes into a steady state until it is shut down.

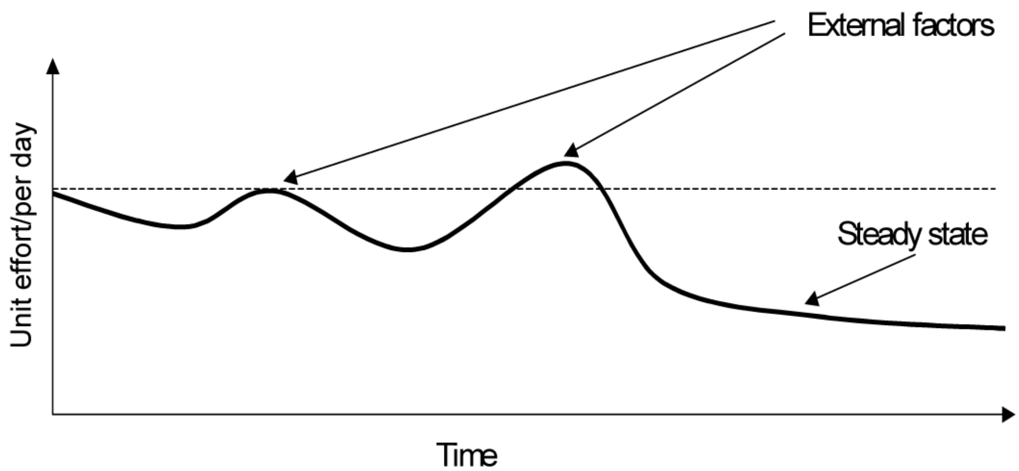

Figure 2.4: A graph displaying the dynamic process affected by external factors [6]

The differences in how a system is developed are crucial for further maintenance. If the project is lagging or having a tight schedule, the final product may be influenced by this. *Figure 2.5* displays the cost differences between a system developed during lagging projects, average projects and leading projects. Only the leading projects have a decrease in maintenance cost during the first five years of their life-time. A lagging project spend almost twice as much as a leading project.

	Lagging Projects	**Average Projects**	**Leading Projects**
DEVELOPMENT	$1,200.00	$1,000.00	$800.00
Year 1	$192.00	$150.00	$120.00
Year 2	$204.00	$160.00	$112.00
Year 3	$216.00	$170.00	$104.00
Year 4	$240.00	$180.00	$96.00
Year 5	$264.00	$200.00	$80.00
MAINTENANCE	$1,116.00	$860.00	$512.00
TOTAL COST	$2,316.00	$1,860.00	$1,312.00
Difference	$456.00	$0.00	-$548.00

Figure 2.5: Cost of software system. Costs are in USD per function point[9]

During development and testing, the costs of fixing an error are measured in units of time, effort, and personnel to locate and correct it. During maintenance, the real price of an error is the price to correct the damage it inflicted on the organization through violated data integrity[20]. This makes the speed of correcting errors in maintenance far more important than in development[20]. Even so, development is in most cases more prioritized than maintenance.

2.5.2 Service-Oriented Architecture

Service-Oriented Architecture (SOA) is a method for system integration. This is done by designing, developing, deploying and managing systems into modules to further provide functionality as a service to other applications. Implementing SOA in organizational systems can benefit business agility, adaptability, leverage of legacy systems and cost-efficiency, consistency and reduced redundancy[34]. In an IT-perspective SOA benefits issues related to reuse, maintenance and Enterprise application integration(EAI).

SOA is designed to enable interoperability over public networks, but it is also used on private networks with Internet-based transport protocols [47]. Large organizations can have systems where capabilities and raw intelligence data can be isolated and lose its context. SOA provides a way to discover these capabilities and combine them to meet a business user's needs. The clients can with SOA access capabilities (services) through a service interface in much the same way as a custom-made application. As organizations develop new services, the resulting service repositories provide a possibility for reuse. [47].

At a high level, service-oriented systems contain three different components. Services, service consumers and the SOA infrastructure. A representation of the high-level service-oriented system is illustrated in *Figure 2.6*.

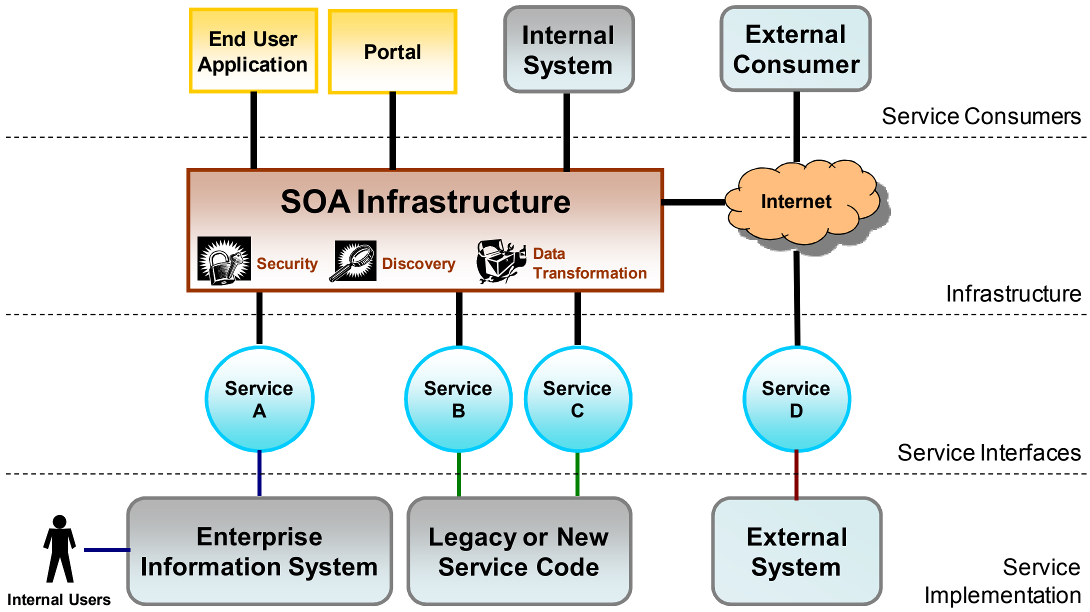

Figure 2.6: A representation of a high-level architecture of a Service-Oriented System [34]

Services are reusable components that represent business tasks. Services can be globally used across processes and the organization. It is possible to reconfigure them as new business processes are integrated [34]. *Service Consumers* are the clients in a service-oriented system. They consume the functionality provided by the services. Examples of service consumers are end-user applications, internal and external systems, and portals [34]. *Infrastructure* is the component that links service consumers to services. It is a message-based communication model. It can also contain a bus to support web-service environments[34].

The SOA design concept is to set the data-processing functionality so that the consumer does not need to know how or where the data processing was executed [47]. If service consumers can access the interfaces as seen in *Figure 2.6*, the service implementation is irrelevant to them. Developers can build services and easily add them as modules to the system. By doing this, the system can easily increase agility and introduce new business models [47].
50% of new operational applications and business processes was in 2007 designed with the use of SOA, believing it would rise to 80% by 2010[34].
SOA is different from traditional systems, which may result in new issues regarding maintenance and software evolution. These differences are 1. diversity of service providers. 2.

because SOA is supposed to rapidly adapt to always changing business needs, it needs shorter release cycles. 3. potential to leverage legacy investments with minimal change to existing systems. [34]. SOA can make the maintenance process more complex. A reason for this is because services are shared among multiple business processes and consumers. It may therefore be hard to control who is responsible for what. Also, all the different business units may have different requirements for the same service. [34]

2.5.3 Cloud

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of computing resources [37]. These resources can be networks, servers, storage, applications and services. An example on the use of cloud could be that an entire system is placed on the organization's servers instead of placed on every single device. The users can then log on to this system through a network. If the system grows bigger, it is only necessary to expand with servers or other hardware at the specific location. Also, if a client needs more resources, the system can allocate this to him internally.
Cloud can provide rapid releases with minimal management effort [37]. There may also be less service provider interaction. There are various degrees of the use of Cloud. Some base their entire systems on it, while others only use single applications running on cloud. Cloud computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the data centres that provide those services [2].

There are many different cloud models available. Some of the these are explained by [37], and presented below :

1. *Private cloud* is an infrastructure used by a single organization with multiple consumers. It can be owned, managed and operated by the same organization or a third party.

2. *Community cloud* is used by a specific community of consumers from different organizations that have shared interests.

3. *Public Cloud* is provisioned for open use by the general public. It may be owned by a business or by the government.

4. *Hybrid Cloud* is a merge of two or more different cloud infrastructures that are bound together. This enables data and application portability among them.

A negative side of cloud, is that the system is placed online, which makes the data more exposed to outsiders. Data is more accessible by online intruders, who can either obtain a user's access credentials or find a security breach in the software. If the system had not been online, it would be impossible to access it from the other side of the world, by either employees or intruders. Frequently, large corporations announce leaks of sensitive information of log-in credentials, examples are Ebay in 2014[2] and yahoo in 2014[3]. This issue has been addressed by organizations and governments, who wish to create both international and global laws for storing sensitive information on cloud systems. Several laws are already passed, in Norway laws addressing this is in the 'personal data act' (Norwegian translation: 'personopplysningsloven').

2.5.4 Outsourcing

Outsourcing is easily explained as setting out services to a third party vendor. This can include outsourcing entire departments like an IT-department or tasks like technical support. It is also possible to outsource defined services such as data-storage. This way there can be more focus on in-house competencies.

Large organizations will internally have to perform many different tasks. They may therefore struggle with tasks beyond their specialized field. They may also calculate that outsourcing of such tasks is cheaper than devoting an entire operational department to do it in-house. To outsource these tasks to other specialized organizations can be a solution. This can reduce costs, resources, and risks for the outsourcing organization [13]. Outsourcing to an organization who specializes in that task, will also increase the products quality. At first the motive for outsourcing was to reduce costs, in later years it was also to achieve technological flexibility, easier control on staffing, and to focus more on in-house competencies[12].

Every organization should have a good sourcing strategy, because sometimes it is not bene-

[2] Article about ebay passwords stolen: `http://www.reuters.com/article/2014/05/21/us-ebay-password-idUSBREA4K0B420140521`

[3] Article about Yahoo passwords stolen: `http://news.yahoo.com/yahoo-email-account-passwords-stolen-002044026--finance.html`

ficial to outsource a service. Drawbacks of outsourcing is that the organization lose control of the task. This can affect safety, quality and trust [13].
Outsourcing should not be mistaken with offshoring. These practices are related, but during offshoring organizations reduce labour costs by moving the work to other countries.

2.5.5 Developers experience

The amount of experience a developer has will influence his skills towards maintenance and development. When categorizing developers in experts and novices, studies indicate that experts implement changes more efficiently[16], organize information in larger chunks [36], and they use more principle based strategies when solving problems[48]. Even if these factors are greater for experienced developers, there is not a strong correlation between experience and skill[27]. What may improve skill is deliberate practice[17]. This can be training activities that are designed to improve specific aspects of the individual's performance.
A study by Jørgensen and Sjøberg supports these results[27]. 109 maintenance tasks solved by 54 developers in an Norwegian organization were studied. They concluded that there was a very large difference between brand new developers (0-1 years experience), and developers with moderate experience. But after reaching a moderate level of experience, there were no significant differences. It was also discovered that increased experience did not lead to higher accuracy when predicting maintenance problems.

2.6 Differences between public and private sector

Public and private organizations work in different sectors, and would naturally be different on some aspects. However, some also report that public organizations have a less effective/satisfying conduct during their systems life cycle (project planning, development and maintenance)[22][38]. An example is that public projects had an overrun of 67%, which is significantly higher than private projects who only had an overrun of 21%[38].
A functional aspect creating differences, is that private organizations follow a business or IT-strategy based on profit or strategic goals, but in public sector, it may be because of political reasons[28]. Because of these political overrides, the power is being taken away from

the organization's managers. By doing this, the decision is made based on political reasons, instead of what is a direct value to the organizations[28].

Something that needs to be considered, is that errors in public organizations are obliged to be unveiled. Private organizations are also making faults, but since they do not have to publish the errors, they are not unveiled to the public. Therefore, the rate of private errors may seem less frequent to the general person. If a private organization has bad results, it goes bankrupt. However, a public organization can in practice not go bankrupt, and can therefore continue to run poorly for a long period of time. This means that the results from private organizations are basically good, because they are not bankrupt.

Chapter 3

Related work

This thesis is part of a replication study executed every five years between 1993 and 2013. These studies are presented in this chapter.

3.1 Lientz & Swanson (1977)

Lientz and Swanson performed an investigation on maintenance in 1977[35]. Surveys were sent out to over 2000 American organizations and responses were received from 487 of them. The investigation was based on the distribution of labour between development and maintenance on application systems.

An observation from this study was that organizations who did not integrate work on maintenance and development spent less time on maintenance. It was also concluded that maintenance increased with the system's age.

3.2 Nosek and Palvia (1990)

In 1990 Nosek and Palvia performed an investigation on American organizations[39]. This study was based on the Lientz/Swanson investigation from 1977, having many of the same questions. The results were extracted from 52 survey responses.

3.3 Krogstie (1993)

J. Krogstie conducted a study on distribution of maintenance and development in 1993[32]. The main goals were to compare maintenance in Norway with similar studies, and to investigate new areas in order to assess the information system support efficiency in organizations. Data was gathered with surveys and 52 organizations responded.

Krogstie concluded that there were no significant differences in his study compared with previous studies. A large portion of time spent on functional maintenance is used on development of replacement systems. The reason for this may be that they are not able to keep up with the organizational and technical changes that come with maintenance.

3.4 Holgeid, 1998

K. K. Holgeid conducted a study on development and maintenance in Norway during the year 1998 [23]. This study was a follow-up and collaboration with Krogstie's study from 1993 [32]. Holgeid used a survey to gather information. The results were based on responses from 53 Norwegian organizations. The main concepts of the investigation were to map the distribution of work spent on maintenance and development in Norwegian organizations, and also to compare his results with similar studies to prove that there were no differences between them.

Some findings were that more work was spent on operation and support, and less on development as opposed to other studies. A conclusion was that distribution of work used for support and operations was taken from time spent on development.

3.5 Jahr (2003)

In 2005 A. Jahr wrote a master thesis on maintenance and development [31]. The investigation consisted of a survey sent out to Norwegian organizations. The data is gathered from 54 respondents. The survey is based on the works done by Holgeid [23] and Krogstie [32]. His main goal was to investigate organizations distribution of work, with emphasis on the categories maintenance and development. His results would also be compared to previous

studies in search for patterns and trends.
Conclusions created were that time spent on maintenance has increased. Also, maintenance was influenced by other factors. Examples were that organizations with complex portfolio spent significantly less time on maintenance. Organizations who did not use pre-defined methods during development and maintenance, would spend more time on maintenance.

3.6 Davidsen (2008)

M. K. Davidsen wrote a master thesis on maintenance and development [14]. This thesis was also a collaboration with Krogstie and was a continuation on that replication study [32], [23], [31]. Davidsen gathered information from 65 Norwegian organizations by sending out on-line surveys. His main goal was to map the distribution of work and then to compare these data to previous studies.
Discoveries found in the investigation were that maintenance has decreased, and left its previous increasing trend. A reason for this was that maintenance would decrease after the Y2K. Fewer variables affected maintenance and development in this study compared to previous studies. But the number of system developers did affect the maintenance variables.

3.7 Distribution of work from related studies

All of the previous studies mentioned in this chapter are based on each other. This means that many of the same questions and hypothesis have been used. The same variables can therefore be compared. The results of distribution of work is listed in *Table 3.1*. These results are discussed later in this thesis.

	Lientz/ Swanson (1977)	Nosek/ Palvia (1990)	Krogstie (1993)	Holgeid (1998)	Jahr (2003)	Davidsen (2008)
Total Maintenance	**49,0%**	**58,0%**	**40,0%**	**41,4%**	**35,9%**	**34,9%**
Corrective maintenance	9,8%	-	10,4%	12,7%	8,7%	8,2%
Adaptive Maintenance	12,3%	-	4,0%	8,2%	7,2%	6,2%
Functional Perfective Maintenance	20,6%	-	20,4%	15,2%	12,5%	11,3%
Non-functional perfective maintenance	6,3%	-	5,2%	5,4%	7,5%	9,1%
Total Development	**43,0%**	**35,0%**	**29,6**	**17,1%**	**21,9%**	**21,1%**
Replacement development	0	-	11,2%	7,7%	9,7%	9,7%
New development	0	-	18,4%	9,5%	12,2%	11,4%
Operation	-	-	-	23,0%	23,1%	23,7%
Support	-	-	-	18,6%	16,8%	20,1%
Other work	**8,0%**	**7,0%**	**30,0%**	**0,0%**	**2,3%**	**0,0%**
Application portfolio upkeep	-	-	44,3	62,0	61,0	63,0
Application portfolio evolution	-	-	55,8	38,0	39,0	37,0
Isolated maintenance	**53,0%**	**62,0%**	**59,0%**	**72,9**	**65,9%**	**65,7%**
Isolated development	**47,0%**	**38,0%**	**41,0%**	**27,1**	**34,1%**	**34,3%**

Table 3.1: Previous results from the replication study

Figure 3.1 is a graph, presenting the share of development and maintenance when you only look at development and maintenance (isolated). This illustrates an increasing trend of isolated maintenance, and a decreasing trend of isolated development. Something to take notice of is that the values are almost identical between 2003 and 2008.

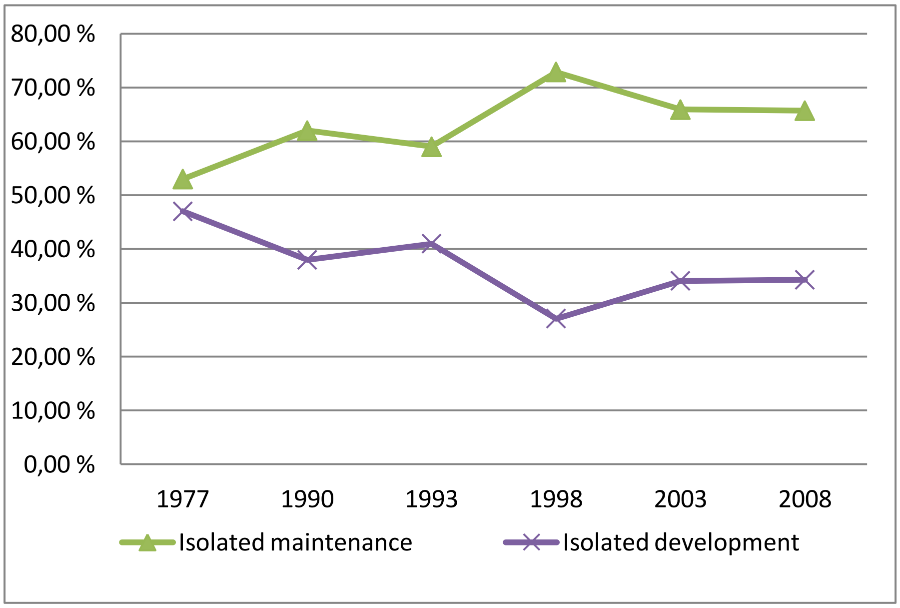

Figure 3.1: A graph displaying maintenance and development from previous studies

Chapter 4

Research methods

The research chapter gives an outline of how the thesis was conducted. It presents and explains the research methods used from pre-study to execution.

4.1 Replication studies

As mentioned earlier, this thesis is part of a replication study. As time passes, it is natural that research with similar subjects occurs. These studies often reference each other for comparison. The same researcher may also repeat a study after a few years, or want to do the study in a different environment. This replication of a study makes researchers see if the results have changed or remained the same. According to La Sorte, "A replication study refers to a conscious and systematic repeat of an original study"[33]. You could also say that it is a study to check the accuracy or truth of the original study. Replication of empirical studies is necessary to establish a more solid case for the results. A single study can prove a hypothesis, but replication of the study reflects knowledge based on separate factors[43]. These separate factors can be time, place, or persons. If these factors change, but the results remain the same, this will set ground for the results. If the factors change, it may also be possible to see trending changes in the results. This will help making new observations of the replication study.

A study is not a replication study if it does not have a relationship with another study. Studies that address the same hypothesis, but without the knowledge of the original study, are not replication studies[43].

Replication studies may change methods or technology that was used. A study can also evolve, meaning that the hypothesis can be modified and new areas of research may be added. Even if the "landscape" of the study changes, but the research questions and relationship to the previous studies remain, they are still a replication study.

Replication studies are classified as internal or external[8]. A study is internal if it is conducted by the original researcher and external if it was performed by an independent researcher[8]. It is also internal if the original researcher is involved in the new study. In 2011 there was an investigation on replication of empirical studies. 16 126 articles were investigated. Among these, there were 93 articles containing replication studies, performed between 1994 and 2010[43]. 71% of the studies were internal, and as much as 60% of the studies were performed in the last six years[43]. This means that replication studies are becoming more popular.

4.2 Choice of method

When doing research, there are a wide range of various research methods to choose from. Each method has its advantages and disadvantages. To chose the most suitable one, comes down to which questions you want to answer and what data you want to gather. This study is part of an internal replication study going back 20 years. John Krogstie performed the first study in 1993 [32], and has since then been the initiative taker with surveys performed every 5 years (1998, 2003, 2008). Because this is a replication study, some ground boundaries had to be adhered to. Choice of method was one of them. This thesis had to be built around a survey investigation. Some of the same hypothesis and questions from previous years would also have to be similar. This was necessary in order to make it possible to compare data against the previous investigations and keeping the gathered data consistent.

This study is built as an empirical research. Hypothesis were first defined based on organizational environment factors found in the pre-study (ex. popular techniques and tools, new technology). Then a survey was made to answer these hypothesis. Data was gathered, which further could be analyzed. The results were compared to the hypothesis and conclusions could be drawn.

4.3 Quantitative research

When working with data analysis, there are two main categories of data to collect. They are qualitative and quantitative data. Qualitative data is all non-numeric data. Examples are interviews, tapes and observations. This data is usually gathered through case studies, ethnography studies and action research. After the data is collected, the researcher extracts the relevant data. The method is often criticized because the researcher does not always manage to explain the process from data to results. Making the results look like they appear out of nothing. Qualitative data is also time consuming to analyse, and because of lack of "guidelines", it is subjective what data is measured as important. The advantage of qualitative data is that every answer is individual, making it possible to receive more detailed information about what the researcher asks. There is also a possibility to an alternate explanation, instead of an answer that is pre-defined.

Quantitative data literally means data, or "based on numbers". It is numbers (ordinal, interval or ratio data) and single words (nominal data), often describing something, and it is presented in tables or graphs. This is the main type of data gathered in experiments and surveys. The reason for this is that it is both efficient to collect, and to process afterwards. Statistical analysis is used on the gathered data, often through a statistical software[1]. The idea is to look for patterns and comparisons, and draw conclusions. The advantage of quantitative data is that it provides scientific respectability. Some even mean quantitative data is the only valid form of research[40]. The analysis are based on measured quantities, and not subjective impressions. Also, large volumes of data can effectively be analyzed (using software programs). The disadvantage of quantitative data can be that the use of statistic mathematics and software tools is for some a barrier. Important qualitative aspects may be missed, and bias the investigation.

[1] SPSS, Excel, Minitab

As explained, one of the categories is not better than the other, but the choice of method decides the data analysis. Because of this, it is often normal to use both quantitative and qualitative data. This is called triangulation. This gives the widest spectre of gathered data, and adds more weight to the conclusions. Naturally, this will also result in more work and time spent. The purpose of this study was to map work distribution, and compare it to surrounding factors. To do this, a natural selection of Norwegian organizations needed to be reached out to. A large amount of data was therefore required from all over the country. A survey collecting quantitative data was therefore a natural choice for this investigation. Qualitative data could have been used in the beginning of the study, in order to build theory and hypothesis. Further, quantitative data could have used to test the hypothesis, but pre-study of earlier studies and relevant articles were sufficient.

4.4 Pre-study

Prior to starting a thesis, a considerable amount of research material has to be gathered and analysed. The supporting literature is one of the most important aspects when writing a thesis. "One of the best ways to maintain an argument, is by presenting evidence from literature"[40]. A study must reference previous work to explain why this study is the same as, or different from, other studies. It is also possible to avoid mistakes by knowing strengths, weaknesses, omissions or bias from similar studies[40]. It is also possible to find gaps in previous research, which can be put into this research, or to find subjects suitable for this research. As pre-study is a necessity for the researcher, it may also help the reader. The researcher has an understanding of the study, because it was he who performed it. The reader on the other hand, usually has little or no knowledge in advance. By adding background information to the thesis, it will be easier for the reader to understand what is being studied and why.

Because of the longitude of this study, going back 20 years, there were a large amount of direct background studies and theory that had to be read. The supervisor provided all the previous studies and was encouraging to get an understanding of the surrounding theory. First when this knowledge was understood, it would be possible to build a new thesis around the objective of this study. After getting an understanding of the replication study, it would be possible to pull out subjects for this thesis, and start to narrow down the pre-study. It would be necessary to find out what was relevant to add and change, based on today's conditions.

Information was gathered mainly by searching online archives after relevant topics, but also by looking at references in similar studies. When finding relevant authors, it would also be possible to find other articles written by them. All the relevant articles and books were categorized, making it easier to find and access at a later stage. Notes of all the articles were also written to know what they contained. This way it would be quicker to re-find important information.

After the information retrieval, it would be possible to create new or modified hypothesis. These hypothesis were made in collaboration with the supervisor. When hypothesis were created, it would be possible to create questions for the survey that would lead to answering these hypothesis. Many questions from previous investigations were naturally used. But as new hypothesis were made, new question were needed.

4.5 Attendees

All 400 attendees were from Norwegian organizations. There were used two different mailing lists. One of the lists was from the Norwegian Computer Society. This list had also been used in previous surveys. The survey was sent out to 284 attendees from this list.

Because of the focus on public vs private organizations in this investigation, more public organizations were necessary. The study of 2008 had a share of only 18% of public organizations. The share of public organizations had to increase to get a comparable amount toward public organizations. A mailing list from 'public sectors data forum' (Norwegian translation: Offentlig Sektors Dataforum (OSDF)) was used. From this list, 116 attendees were deducted. It was necessary that the organizations had IT-departments and used software systems as a central tool in the business strategy. By using these lists, it was assured that all participants had some relation to IT and that many probably had their own IT-department.

4.6 The survey

The survey was sent out to organizations with differences in size, geography and field of work. The chance of getting a representative selection was larger using this method. Because the study was gathering quantitative data, and a large number of respondents was necessary, a survey would be the most efficient. The survey was attached to an email sent out to the recipients. The survey can be found in *Appendix A*. An online survey tool called surveymonkey[2] was used. All replies were sent directly to the registered account on this website. Surveymonkey was used because it is a tool that gives a good overview of recipients throughout the process. This included possibilities to see who had received the survey, who and how much they had answered at all time, and it contained analysing tools.

A rule of thumb in data gathering is to have at least 30 participants[40]. If not, statistical analysis are not reliable. If the sample is less than 30, displaying the data in percent should be avoided.

Two factors to be considered are accuracy range and confidence level. Researchers normally work towards a confidence level of 95 percent. This means that we can be sure that the true population value falls within the range of values obtained from the sample. The other factor, accuracy range (also called margin for error), tells us how close to the true population we are. To get perfect accuracy, the entire target population would need to participate in the survey. For our survey, this would mean all Norwegian organizations. In most cases that is impossible. Researchers therefore usually work with an accuracy range of +-3%.

As an example, if the population is 1 000 000, 1000 people would have to participate if the accuracy range and confidence level is supposed to be 95% and +-3%. Previous studies in this investigation indicate a response rate of about 20%. This was therefore the estimated percentage of responses for this survey. The survey was sent out to 400 recipients from different organizations all over the country. The survey was sent attached to a mail, as seen in *Appendix B*. The recipients got a deadline of two weeks to fill in the survey. After the deadline had passed, there were not enough answers. Two mail reminders were sent out before a satisfying number of replies were achieved. Of the 400 recipients on the list, 62

[2] `www.surveymonkey.com`

were fully filed out and 26 were partly filled out. A reason for partly-filled out answers was probably because the recipients were not able to answer all of the questions. This may be because the questions were not relevant for the particular organization or the recipient may have found the survey too time-consuming. Having 87 started replies, the answer percent was 21,7, which was the expected result. However, a total number of 68 replies were filled out in such a degree that it could be used in the data analysis. This was an answer-percentage of 17%, a bit smaller than expected, but sufficient.

4.7 Data analysis

All answers were automatically stored in our survey tool: surveymonkey. Surveymonkey has analysis tools, but they were not sufficient for our needs. All data was extracted as numerical data to an excel file. The data then had to be rendered manually to see if all data was consistent and correct. If values were incorrect, they were either corrected or removed. Some new variables were also calculated and added. E.g calculation of application portfolio and isolated maintenance. When this stage was finished, the data was ready to be imported to a proper analysis tool. In this study the chosen software was: Statistical Package for the Social Sciences (SPSS). SPSS is a software which provides the researcher to do statistical analysis on their own. An advantage is that it integrates statistical analysis, data management and data documentation in the same software. SPSS was selected because the supervisor had experience with it and therefore knew it would cover all functions for the necessary analysis. First, descriptive results were created. In this process, primarily frequency tables and descriptive tables were created. The main point of these results is to present the data that had been gathered. These results were further discussed and compared to previous studies.

To answer the hypothesis, more advanced statistical methods had to be used. These methods are listed below. Some of the hypothesis made it necessary to directly compare this study's data to the data collected 5 years ago. Raw data from the 2008 survey had to be combined with the data from this study, and was further analysed.

4.7.1 Spearman's correlation

Spearman's correlation[3] is a method that compares the linear relation between paired data. This is done by evaluating the monotonic function (either it never increases or decreases as its independent variable increases). When comparing the data, one variable (x) is sorted. The other variable should now either always increase or decrease as x grows. The monotonic relationship has a scale called correlation coefficient value. This value goes from from 0 (very weak), to 1 (very strong). In addition to the correlation coefficient value, a p-value is also used to evaluate the correlations significance. If p is higher than 0.05, the data is not significant. 0.05 means that there is a 5% chance that the correlation is just a coincident.

4.7.2 The Wilcoxon signed-rank test

The Wilcoxon signed-rank test is a paired difference test used when comparing two related samples to assess if their population mean ranks differ. Wilcoxon is equivalent to the dependent T-test. The test can make assumptions that the scale of measurement for the two related samples has the properties of an equal-interval scale. The method subtracts one of the datasets with the other. The results are then separated into positive numbers, negative numbers or ties. Finally, the method finds out if the differences have statistically significant differences. This is the Asymp. Sig. (2-tailed) number which is the p-value for the test.

4.7.3 Mann Whitney U-test

Mann Whitney U test is a T-test. It is used to compare differences between two-independent samples. The data is grouped in two categories (example public and private sector). Then the mean rank and sum of ranks are calculated separately in the two groups. The mean rank is useful to see the numerical difference of the two groups. It then calculates the test statistics. The U-value is the value indicating the statistically difference of the two groups. There is also a p-value presenting the significance.

[3]Spearman correlation: http://www.statstutor.ac.uk/resources/uploaded/spearmans.pdf

4.8 Collaboration with 'IT i praksis'

This years study had a collaboration with another study called 'IT i praksis'[4]. This is a yearly investigation performed by Ramboll[5] and the Norwegian Data Association (dataforeningen)[6]. Every year, a survey is delivered to 500 Norwegian public and private organizations. The questions are based on relevant topics at the time, so the questions may vary from year to year. The survey focuses on strategy, trends and experience towards IT. This collaboration involved that some of the survey questions were switched. The gathered data was then shared as soon as it was collected. This made it possible to compare our data to another study with more than 200 respondents, making a more solid case for our results. It is unknown if 'IT i praksis' will use data collected in this study.

Some of the questions in 'IT i praksis' were a bit different than in our study. This made the comparison of data not hundred percent valid. Even so, these data were a good assumption if the results were practicable.

[4] IT-i praksis: http://www.ramboll.no/services/management-consulting/it-ledelse-og-it-strategi/it-i-praksis

[5] Ramboll: http://www.ramboll.no

[6] Norwegian data association: http://www.dataforeningen.no/in-english.128921.no.html

Chapter 5

Hypothesis

This chapter presents the hypothesis used in this investigation. These hypothesis are discussed and analysed in later chapters.

5.1 Maintenance and development

H1. There are no differences in the amount of time spent on maintenance and development, when only looking at maintenance and development.

H2. There are no differences in the amount of time spent on maintenance and development.

H3. There are no differences between time spent on application portfolio upkeep and traditional maintenance, when only looking at development and maintenance.

H4. There are no differences between time spent on application portfolio evolution and traditional development, when only looking at development and maintenance.

H5. There are no differences between time spent on application portfolio evolution and application portfolio upkeep.

5.2 Type of organization

H6. There are no differences in the distribution of work between organizations with many employees and organizations with fewer employees.

h7. An organization's distribution of work is not affected by the top IT-manager role-priority.

5.3 Importance of IT

H8. There are no differences in the distribution of work between organizations where the size of the IT-department compared to the total number of employees is large, and the organizations where the size of the IT-department compared to the total number of employees is small.

H9. There are no differences in the distribution of work between organizations in which there are many system-developers in proportion to total number of internal users, and organizations with few system-developers in proportion to total number of internal users.

H10. There are no differences in the distribution of work between organizations in which there are many system-developers in proportion to total number of employees in the IT department, and organizations with few system-developers in proportion to total number of employees in the IT department.

H11. There are no differences in the distribution of work in organizations where IT- and business strategy are integrated, and where this is not the case.

5.4 Consultants and employees

H12. There are no differences in the distribution of work between organizations with high average experience among developers, and organizations with low average experience among developers.

H13. The number of hired consultants in an organization does not affect its distribution of work.

5.5 Complexity of the portfolio

H14. There are no differences in the distribution of work between organizations with many main systems and organizations with fewer main systems.

H15. There are no differences in the distribution of work between organizations with many end-users and organizations with fewer end-users.

H16. There are no differences in distribution of work between organizations with main-systems having a high average age, and organizations with main-systems having a low average age.

H17. There are no differences in the distribution of work between organizations that use many different programming-languages, and organizations that use fewer different programming-languages.

5.6 Use of methods and tools

H18. There are no differences in the distribution of work between organizations that use pre-defined methods throughout the system's life cycle, and the organizations that do not use this.

H19. There are no differences in the distribution of work in organizations with a high number of routines established for management and maintenance of IT-systems, compared to organizations with less routines for this.

H20. When developing replacement systems, it is easier to reuse specifications and design, than code.

5.7 Outsourcing

H21. There are no differences in the distribution of work between organizations that outsource much of the total IT-activity, and organizations that outsource less of the total IT-activity

H22. The use of outsourcing is not dependent on the size of the company

H23. There are no differences in the distribution of work in organizations that develop most of their main systems internally, through an external organization or use package solutions.

5.8 Service-oriented architecture

H24. There are no differences in the distribution of work between organizations that have deployed service-oriented architecture and organizations that have not deployed service-oriented architecture.

H25. The use of service oriented architecture is not dependent on the size of the company.

5.9 Comparisons with previous survey

H26. There are no differences between the percentage of maintenance time in our survey and what was reported in the previous survey.

H27. There are no differences between the breakdown of maintenance work (corrective, adaptive, enhancive and perfective) in our survey and what was reported in the last survey.

H28. There are no differences between the percentage of development time in our survey and what was reported in the last surveys.

H29. There are no differences between the percentage of time used on support and operation in our survey and what was reported in the last survey.

H30. There are no differences between the distribution of work among maintenance and development in our survey and what was reported in the last surveys when disregarding other work than development and maintenance.

H31. There are no differences between the distribution of application portfolio upkeep and application portfolio evolution in our survey and what was reported in the last surveys.

5.10 Replacement systems

H32. There are no differences in the share of total new systems being developed that is classified as replacement systems in our survey and what was reported in 2008 and 2003.

H33. The average age of a system that is being replaced, is the same in our survey and what was reported in 2008 and 2003.

5.11 Public and private differences

H34. There are no differences in the amount of outsourcing between public and private organizations.

H35. There are no differences between time spent on maintenance in public and private organizations.

H36. There are no differences between the percentage of time used on development in private and public sector.

H37. There are no differences between the distribution of work among maintenance and development between private and public sector when disregarding other work than development and maintenance.

H38. There are no differences between the distribution of application portfolio upkeep or evolution in private and public sector.

H39. There are no differences between the percentage of time used for operation and support between private and public sector.

5.12 Cloud

H40. There are no differences in the distribution of work between organizations with many main systems using cloud, compared to organizations with few main systems using cloud.

H41. There are no differences in the use of cloud between organizations having many employees and end-users, compared to organizations with few employees and end-users.

Chapter 6

Descriptive results

In this chapter, all descriptive results from the survey will be presented. This chapter contains only a visualization of current and previous data. This means that discussion and conclusion will be presented in later chapters. The results will be compared to previous studies in this replication study (1993, 1998, 2003, 2008). Old data will usually be presented in parentheses.

6.1 Respondents

The survey was addressed to contact persons in different organizations. It was preferable from our side that an IT-manager would reply. The reason for this was that they have often obtained the experience needed for good and plausible answers. It is also the IT-managers who have the overview of the department, and have the needed answers at hand. *Table 6.1* displays the distribution of employment among respondents.

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	IT-manager	57	83,8	83,8	83,8
	Business Manager	4	5,9	5,9	92,6
	System Developer	3	4,4	4,4	100,0
	Project Manager	2	2,9	2,9	95,6
	IT-Architect	2	2,9	2,9	86,8
	Total	68	100,0	100,0	

Table 6.1: Respondents work position

The total percentage of managers is 92,6% (when combining all three manager-categories). 4,4% were system developers and 2,9% were IT-architects. Compared to previous studies, the results were similar (2008 - 97%; 2003 - 82%;1998 - 91%; 1993 - 94%). The majority of IT-managers makes the answers consistent both in this study and toward previous studies.

The respondents' IT-experience was also asked for in the survey, because experience can influence answers. A person who has worked many years will answer based on experience, while a person with a shorter career may answer based on theory or common perception. The respondents' average years of experience is displayed in *Table 6.2*.

	N	Minimum	Maximum	Mean	Std. Deviation
Years experience IT	68	3	40	21,31	8,223
Valid N (listwise)	68				

Table 6.2: Respondents IT-experience

The average year of IT-experience for the respondents is 21,3 years, with none having less than 3 years of experience. This is an increase of years experience compared to previous studies (2008 - 17; 2003 - 15; 1998 - 14; 1993 - 17).

6.2 Organization

This study has tried to get a normal distribution of organizations. To get this, different contact lists were used. It was preferred to have participants from different sectors and of different size. This section presents results about the different organizations participating in this survey. If one type of organization is of a majority, this can influence the results. Even if it was not necessary (or a desire) to contact the same organizations as previous studies, they would still need to be of the same distribution.

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Public sector	44	64,7	64,7	64,7
	Consulting services	9	13,2	13,2	77,9
	Telecom & IT	4	5,9	5,9	83,8
	Retail	3	4,4	4,4	88,2
	Other	3	4,4	4,4	92,6
	Bank and Insurance	2	2,9	2,9	95,6
	Healthcare	1	1,5	1,5	97,1
	Travel and transport	1	1,5	1,5	98,5
	Construction	1	1,5	1,5	100,0
	Total	68	100,0	100,0	

Table 6.3: Organizations field of work

Table 6.3 presents the field of work for the participated organizations. More than half of the organizations work in public sector with 64,7% (2008 - 19%; 1993 - 14%) A reason for public increase is that more focus has been on public/private organizations in this study, resulting in more public organizations in the participants list. Public and private sector often have a different focus in fields of practice, which may affect this survey's gathered data. Therefore, the majority of public organizations may be a reason for some of the results being different from previous years. Even if this may affect some results, it will also give a possibility to compare data between public and private sector. This is an important and new topic of this survey.

The distribution of the other organizations was consulting with 13% (2008 - 23%; 2003 - 41%; 1998 - 19%) and telecom with 5,9% (2008 - 20%; 2003 - 46%; 1993 - 15%). The reduction of distribution among the organizations, is because of the focus in getting public organizations this year.

IT-leaders roles	Frequency	Percent	Cumulative Percent
Being proactive toward business management with new ideas and initiatives to change processes and applications	16	24,2	24,2
Ensuring cost-effective delivery of core IT services	15	22,7	83,3
Fire fighting and daily operations	11	16,7	100,0
Collaboration with business management on improvements to applications	10	15,2	39,4
Ensure that new projects are delivered on time, within budget and quality	9	13,6	60,6
Develope new business models that exploit technological opportunities	5	7,6	47,0

Table 6.4: Description of top IT-leaders role in daily operations

The assumption of the top IT leader's role of the organization may impact the daily operation of it. It was therefore asked what describes the top IT-leaders role today. The categories and frequencies of answers are visible in *Table 6.4*. A majority answered that the top leader who spend most of the time being proactive toward business management (24,2%) and ensuring cost-effective delivery of core IT-services (22,7%). Only 7,6% means the top leader spend most of the time on developing new business models that exploit technological opportunities.

It is possible to split the results in more narrow categories. One is "taking initiative to development and improvement to applications", and the other category as "other managerial tasks". This makes 53% of all top leaders having a focus on other managerial tasks, and 47% having a focus on development. This means that only half of the leaders focus on new development, while the rest is focusing on managerial tasks (e.g: fire fighting and deadlines).

	Frequency	Valid Percent	Cumulative Percent
Business- and IT-strategy do not integrate	15	22,1	26,5
Business- and IT-strategy are integrated	50	73,5	100,0
Total	68	100,0	

Table 6.5: Integration between IT- and business strategy

Table 6.5 displays the organizations' IT-integration. 73,5% have some sort of integration between IT- and business strategy. This could indicate that most organizations base their development on business factors, which should encourage application portfolio evolution. Evolution is needed for a system to stay competitive in a business environment.

	N	Minimum	Maximum	Median	Mean	Std. Deviation
Number of employees	68	2	12000	400	1083,54	2356,539
Valid N (listwise)	68					

Table 6.6: Average number of employees in organizations

In *Table 6.6* the mean number of employees is 1083, having a standard deviation of 2356,5 and a median of 400. The high SD and low median compared to the mean, normally means that there was a large variation in number of employees. To illustrate the distribution of employees among the participant organizations, a box plot is added in *Figure 6.1.* There are some outliers, but most are grouped around 2 and 1000 employees.

The mean in this study was exactly the same as in 2008 when the mean also was 1083, but then the standard deviation was almost twice as large (4521). The reason for that was that the 2008 investigation had even more outliers (largest organization had 35.000 employees). Earlier studies resulted in varying numbers (2008 - 1083; 2003 -181; 1998 - 656 ;1993 - 2347). The reason for the different numbers is hard to say, but it is clear that different contact lists have been used. If size of the organizations influence work distribution is therefore a hypothesis and will be discussed in a later chapter.

Because this study contains a large degree of public organizations, the difference in size between these sectors were also investigated. Public organizations had a median of 525, while private organizations only had a median of 82. The public organizations were 85% larger than private organizations in this population.

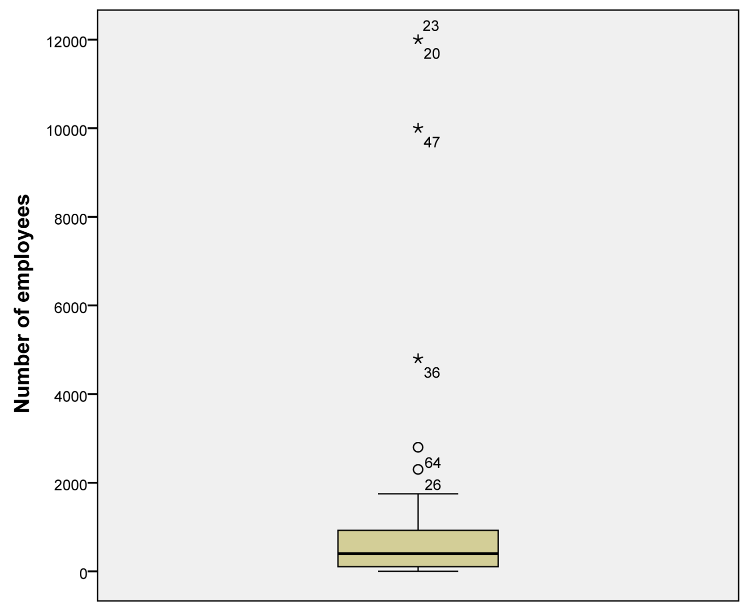

Figure 6.1: A box-plot showing number of employees in the organizations

Table 6.7 shows the organizations' IT-budget in all studies. With a simple calculation to find the average budget of the organizations[1], we find an average of NOK 21,5 millions (in millions; 2008 - 16,6; 2003 - 10,4; 1998 - 18,2; 1993 - 12,0). Besides from 1998, there has been an increase in the average budget. However, the majority of organizations have had a budget between NOK 1-10 million in all studies. In later years, more organizations with a budget of more than NOK 50 millions has occurred more often.

Comparing public and private budgets, the sectors were almost identical. Both sectors had a mean between NOK 15-25 millions.

IT-budget	2013		2008		2003		1998		1993	
	N	Percent	N	Percent	N	Percent	N	Percent	N	Percent
NOK 50 million	15	22,1%	10	15,4%	4	7,4%	9	17,0%	4	9,3%
Between NOK 40 and 50 million	0	0,0%	3	4,6%	2	3,7%	2	3,8%	1	2,3%
Between NOK 30 and 40 million	3	4,4%	1	1,5%	0	0,0%	2	3,8%	1	2,3%
Between NOK 20 and 30 million	9	13,2%	3	4,6%	1	1,9%	2	3,8%	2	4,7%
Between NOK 10 and 20 million	7	10,3%	9	13,8%	4	7,4%	8	15,1%	5	11,6%
Between NOK 1 and 10 million	25	36,8%	18	27,7%	28	51,9%	18	34,0%	13	30,2%
Less than NOK 1 million	9	13,2%	21	32,3%	15	27,8%	12	22,6%	17	39,5%
Total	68	100,0%	65	100,0%	54	100,0%	53	100%	43	100%

Table 6.7: Organizations IT-budget

[1] Average budget of organizations =((#*60)+#*45)+(#*35)+(#*25)+(#*15)+(#*5)+(#*0,5))/N

The growth of IT-budget is probably influenced by many internal and external factors. The general organizations are larger, and more money is invested in IT compared to earlier. Since the first study 20 years ago, inflation would also be assumed. The inflation between 1993 and 2013 was 48%[2]. Considering the budget compared to the inflation in 2013, we get in NOK millions: 2013 - 21,5; 2008 - 18; 2003 - 12;1998 - 24; 1993 - 18. Considering inflation, the IT-budget has therefore not grown as much as it first appeared.

6.3 Distribution of work

This section contains a presentation of the organizations' distribution of work, both the distribution of work internally and work being outsourced.

The increase in outsourcing the last years has made outsourcing a larger part of this replication study. However, outsourcing was only part of the 2008 and 2013 studies. *Table 6.8* shows that 85,3% of all organizations outsource some sort of services. This is an increase from 2008 when 79% of the organizations outsourced services.

	Frequency	Percent
Outsource	58	85,30 %
Do not outsource	10	14,70 %
Total	68	100,00 %

Table 6.8: Table displaying if the organization outsource services or not

Even if a large majority of the organizations outsource some part of their services, there are still differences in how much they outsource. *Table 6.9* displays how much of a service that is being outsourced.

[2]Central Bureau of Statistics consumer price index: `http://www.ssb.no/priser-og-prisindekser/statistikker/kpi`

	N	Minimum	Maximum	Mean	Std. Deviation
Outsourcing total IT-activity	68	0	100	32,68	30,817
outsourcing development	68	0	100	47,12	43,777
Outsourcing maintenance	68	0	100	34,49	36,698
Outsourcing operation	68	0	100	33,96	36,671
Outsourcing support	68	0	100	16,10	27,247
Valid N (listwise)	68				

Table 6.9: How much (in percent) the organizations outsource of different tasks

One third (32,7%) of the total IT-activity was outsourced. This was a almost the same as in 2008 when the mean was 29,7%.

In 2013 development and maintenance were outsourced 47,1% and 34,5%, while in 2008 both were lower and more even with 32% and 31% respectively. The outsourcing of development has increased the most since 2008.

The numbers from *Table 6.9* and *Table 6.8* indicates that the share of outsourcing in Norwegian organizations is not continuing to grow, but the distribution of what is outsourced has changed.

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Reasonably accurate, based on good data	15	22,1	22,1	22,1
	A rough estimate, based on minimal data	27	39,7	39,7	61,8
	A best possible guess, not based on any data	26	38,2	38,2	100,0
	Total	68	100,0	100,0	

Table 6.10: Quality of answers towards outsourcing

Table 6.10 shows the quality of the outsourcing data. The quality of this data is a bit low, with only 22% answering based on good data, and 40% answering based on a rough estimate. The answers are therefore not totally reliable, but reliable enough to give an indication on the outsourcing situation.

Table 6.11 contains the distribution of work in the organizations. This study had the most total maintenance and the least total development of all previous studies. Total maintenance is the sum of corrective, adaptive, enhancive and non-functional maintenance and took 41,48% of the distribution of work (2008 - 35%; 2003 - 36%; 1998 - 41%; 1993 - 40%). The maintenance category has throughout the studies been stable between 35 and 42 percent. Total development was in 2013 13,63% (2008 - 21%; 2003 - 22%; 1998 - 17%; 1993 - 30%). This is a new low for the share of total development, being almost half of the previous investigation. Development has had a more noticeable change than maintenance during these studies.

	N	Minimum	Maximum	Mean	Std. Deviation
Total maintenance	**67**		**100%**	**41,48%**	**20,25%**
Corrective maintenance	67		25%	9,27%	6,72%
Adaptive maintenance	67		30%	8,84%	6,00%
Enhancive maintenance	67		60%	12,24%	11,61%
Non-Functional perfective maintenance	67		87,5%	11,14%	12,15%
Total development	**67**		**70%**	**13,63%**	**14,23%**
Development of replacement systems	67		60%	6,84%	9,94%
Development of new systems	67		30%	6,80%	7,16%
Operations	67		70%	21,78%	13,86%
Support	67		100%	23,10%	20,37%
Isolated maintenance	**67**		**100%**	**77,73%**	**20,01%**
Isolated development	**67**		**100%**	**22,27%**	**22,08%**
Application portfolio upkeep	67		100%	68,07%	19,24%
Application portfolio evolution	67		69%	31,93%	19,23%
Valid N (listwise)	67				

Table 6.11: Distribution of work in IT-department

When disregarding other work than maintenance and development (isolated), maintenance had a share of 77,73% (2008 - 66%; 2003 - 66%; 1998 - 73%; 1993 - 59%) and isolated development had a share of 22,27% (2008 - 34%;2003 - 34%;1998 - 27%; 1993 - 41%), meaning that both isolated development and maintenance were at a record high and low compared to previous years. *Figure 6.2* below, is a graph of the distribution of maintenance and development the last 20 years. It is possible to see a trend of isolated maintenance increasing and isolated development decreasing between the investigations.

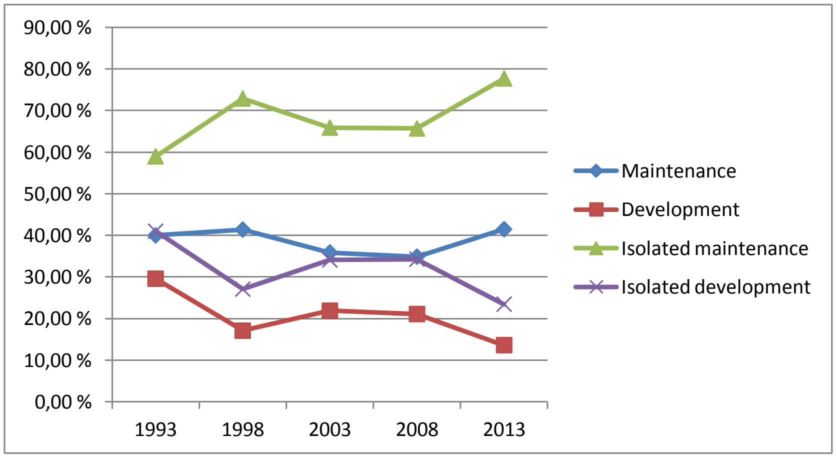

Figure 6.2: Distribution of maintenance and development the last 20 years

Maintenance was about the same, and development was lower compared to previous studies. This can have a connection with the distribution of outsourcing. There was a large difference in outsourcing of development and the about the same share of outsourcing of maintenance in this study compared to the previous study. It may be a larger trend to outsource development, and execute the maintenance internally.

Application portfolio upkeep (corrective, adaptive, and enhancive maintenance, together with development of replacement systems) was 68,1% (2008 - 63%, 2003 - 61%; 1998 - 62%; 1993 - 44%). Looking at 1998 to 2008, upkeep seemed to have stabilized, but this survey displayed a huge increase since the last study. Application portfolio evolution (develop new systems and functional perfective maintenance) was 31,9% (2008 - 37%, 2003 - 39%; 1998 - 38%; 1993 - 56%).

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Reasonably accurate, based on good data	8	11,8	11,8	11,8
	Rough estimate, based on minimal data	30	44,1	44,1	55,9
	A best possible guess, not based on any data	30	44,1	44,1	100,0
	Total	68	100,0	100,0	

Table 6.12: Quality of answers about distribution of work

Table 6.12 shows that the quality of answers on distribution of work is very low. Only 11,8% of the answers are reasonably accurate. The average was between a rough estimate and a best possible guess.

6.4 IT-department

This section gives a presentation of the organizations' IT-departments. Examples are number of developers and the use of consulting. This data can influence the results. An example is that the number of developers could influence the distribution of work.

	N	Min	Max	Mean	Std. Deviation
Number of employees in IT-departement	68	0	90	13,35	17,90
Number of system developers	68	0	60	4,22	9,89
Average number of consultants in IT-department	68	0	35	3,12	6,56
avg. numb consultants / numb employees IT-department	67	0	1,88	0,23	0,40
Valid N (listwise)	68				

Table 6.13: Number of employees in IT-department

Table 6.13 represents the average number of employees in the IT-department which is 13,35 calculated to full-time employees(2008 - 14; 2003 - 10; 1998 - 11; 1993 - 24). This could indicate that the IT-departments have grown the last 10 years. However, when comparing the employees in IT-departments to the total amount of employees in the organizations, only 1,23% work in the IT-department (2008 - 1,3%;2003 - 5,4%; 1998 - 1,7%; 1993 1,0%). These numbers vary so much that no trend was found and it could seem like the IT-department grows in parallel with the organizations.

The number of system developers in the IT-department has a mean of 4,22 (31,6%). This is a slight increase from 2008 when only 19,2% of the IT-department was developers. It is still lower than previous years, when in 2003 and 1998 it was 42%, and 39% in 1993. This decrease of developers the last years could be linked to the increase of outsourcing development.

There was a mean of 3,12 full-time consultants in the IT-department. This is the highest number of consultants throughout the studies (2008 - 2,82; 2003 - 0,7; 1998 - 2,7). The number of consultants may have been low in 2003 because of the "dot-com bubble" which appeared around 2001 making a though market for consultants. Consultants are usually the first people who are cut back. Besides from 2003, the number of consultants has been quite stable, with an increase this year. When comparing number of consultants to the total number of employees in the IT-department, the mean was 0,23.

What was unexpected, was that even if the number of consultants is at a record high, there was still a large percentage that did not hire consultants at all. As much as 48,5% of the organizations answered that they did not have any hired consultants (2008 - 45%; 2003 - 56%; 1998 - 30%). This could indicate that there was not a larger number of organizations that hired consultants, but the ones who did, hired more now than before.

	N	Minimum	Maximum	Percent	Mean	Std. Deviation
0-1 year experience	41	0	8	7,2%	,76	1,80
1-3 years experience	41	0	12	12,1%	1,27	2,48
3-6 years experience	41	0	50	25,6%	2,68	8,28
6-10 years experience	41	0	10	13,7%	1,44	2,62
More than 10 years experience	41	0	50	41,4%	4,34	10,79
Average years experience	**41**	**,50**	**15,00**		**8,38**	**4,65**
Valid N (listwise)	41					

Table 6.14: Years experience for employees in IT-departments

Table 6.14 presents the years experience of employees in the IT-department. The majority of employees had more than 10 years experience (41%). The average years experience is 8,4 years[3].

Comparing these numbers to the 2008 study, the numbers are less distributed. Only 0,81 of the employees have more than 10 years experience, resulting in an average experience of 5,8 years (2,6 years less than 2013).

In 2003 the average years experience was 5,4 years, 8,8 years in 1998 and 7 years in 1993. The average years experience was higher in 2013 than in 2008, but there were no trends to be found in the last 20 years of study.

[3]When calculating the average, we gave every category a weighted number. Example is 0-1 years = 0,5 and 1-3 years = 2 years etc. Finally we calculated the average of these factors. Formula: (0,5*A + 2*B + 4,5*C + 8*D + 15*E) / (A + B + C + D + E)

6.5 System portfolio

This section presents results based on the organizations' system portfolio.

	N	Minimum	Maximum	Mean	Std. Deviation
Number of main systems	66	1	100	11,62	17,258
Valid N (listwise)	66				

Table 6.15: Number of operative main systems

From *Table 6.15* it is possible to see that the mean number of main systems used in organizations was 11,6. This is the highest recorded number in the studies (2008 - 8; 2003 - 5; 1998 - 10; 1993 - 10). The large amount of main systems may have a connection with the fact that organizations in this study was larger in size.

	N	Minimum	Maximum	Sum	Percent	Mean	Std. Deviation
#systems 0-1 year	27	1	24	77	10,3%	2,85	4,622
#systems 1-3 years	42	1	25	157	20,9%	3,74	4,819
#systems 3-6 years	46	1	30	213	28,4%	4,63	5,953
#systems 6-10 years	42	1	30	215	28,6%	5,12	6,660
#systems 10+ years	25	1	15	89	11,9%	3,56	3,404
Average age main systems	**66**	**,50**	**15,00**			**5,92**	**2,915**

Table 6.16: Distribution of main systems age

Table 6.16 shows the distribution of age among the main systems. The average age was 5,9 years, which was pretty much in the middle of our age distribution (0-10 years). In 2008 the average age was 5 years and had a similar age distribution as in this study. Also in 1998 and 1993 the average age was 5 years, but in 2003 it was only 3,9 years. Y2K is a natural reason for why it was lower in 2003.

Besides from 2003, the average age has been stable for about 5 years, with a slight increase in this study. Even if there are more main systems in this study, the average age of them are still the same. There was also a major increase in systems older than 10 years (12% in 2013 and 6,6% in 2008), meaning that systems are in production longer now than before. In 1993, 51% of the systems were 0-3 years old, while in 2013 31% were in that category. Systems are longer in production now than before.

	2013			2008		2003		1998		1993
	Sum	Percent	Mean	Sum	Percent	Sum	Percent	Sum	Percent	Percent
Developed internally by IT-department	96	12,6%	1,45	53	12,0%	47	22,6%	132	26,8%	59%
Developed in the organizations user group	21	2,8%	,32	10	2,3%	4	1,9%	132	26,6%	1%
Developed by an external organization	236	31,0%	3,58	176	39,8%	73	35,1%	108	22,0%	12%
Package solution, with major internal adaptations	165	21,7%	2,50	100	22,6%	25	12,0%	47	9,6%	11%
Package solution, with minor internal adaptions	154	20,2%	2,33	78	17,6%	57	27,4%	72	14,6%	17%
Solutions that use web services/components developet externally	89	11,7%	1,35	25	5,7%	2	1,0%	2	0,4%	-

Table 6.17: Distribution of software development

From *Table 6.17* it is possible to deduct that most systems are developed by an external organization (31%) and only 12,6% of the systems are developed internally. These numbers are very similar to the 2008 study where 39,8% were developed externally and 12% were developed internally. It is peculiar that these numbers are similar to the 2008 study, when the results from outsourcing showed that there was a large increase of outsourcing development.

There are always changes in what type of software is popular, with changing trends between decades. Package solutions have had an increase since 1993 and up to 2008. Between 2008 and 2013, these numbers have not changed much. This could indicate the beginning of a new trend and popularity. An example is web services and components developed externally, which have had a major increase the last 10 years. The significant rise of package solutions with minor changes in 2003 may be because of the "dot-com-bubble" bursting around year 2000, making organizations look for simpler solutions.

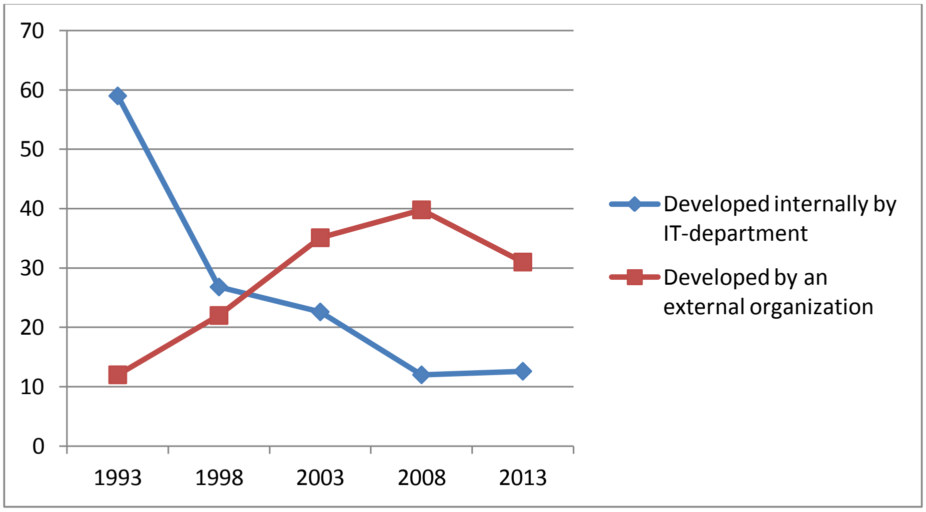

Figure 6.3: Relationship between internal and external organizational development

Figure 6.3 shows the relationship between internal and external development of systems. Internal and external development have had a parallel rise and fall, with an intersection around year 2000. External development may have reached its peak in 2008, but it is too early to draw a conclusion of that.

	N	Minimum	Maximum	Median	Mean	Std. Deviation
#internal end-users	66	2	12000	325	852,79	1757,595
Valid N (listwise)	66					

Table 6.18: Number of internal end-users

From *Table 6.18* we can see that the mean number of internal end-users was 853. The standard deviation was quite high, so there were probably some outliers. This is a large increase from previous years (2008: 559, 2003: 115, 1998: 498 and 1993: 541). 2008 had a standard deviation of 2000, meaning there was probably even more outliers than in this study. The reason for the last years increase of internal users can be that there are more public organizations participating in this study which of course usually are very large. Public organizations had a mean of 1073 internal end-user, while private organizations had only a mean of 468.

The median of number of end-users was 325, which may give a more accurate estimation, neglecting outliers.

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Reasonably accurate, based on good data	44	64,7	66,7	66,7
	Rough estimate, based on minimal data	14	20,6	21,2	87,9
	A best possible guess, not based on any data	8	11,8	12,1	100,0
	Total	66	97,1	100,0	

Table 6.19: Quality on answers regarding internal end-users

The data based on the number of internal end-users are reasonably accurate based on *Table 6.19*. It shows that as much as 87,9% answered based on reasonably accurate or based on a rough estimate.

	N	Minimum	Maximum	Median	Mean	Std. Deviation
#external end-users	66	0	3000000	30	91868,02	443049,417
Valid N (listwise)	66					

Table 6.20: Number of external end-users

The mean number of external end-users was 91 868, based on data from *Table 6.20*. However, looking at the box-plot in *Figure 6.4* this number is influenced by some major outliers. It is therefore not easy to compare these data with previous studies. When removing the two largest organizations (3 millions and 2 millions), the mean was 16 613. Even this number is drastically larger than all previous studies. That the median was only 30, shows that there were some huge outliers pushing the mean up.

In 2008 the mean number of external end-users was 3819 and in 2003 the mean was only 198. Different factors affect this. The main factor is probably this investigation's increase in public organizations. Public organizations often have a focus on the entire national or municipal population. Public organizations had a mean of 121 367 and private organizations had a mean of 40 244. Another reason is the expanding use of online systems, which makes them reach a lot of external users. All organizations have since 1994 put themselves on the Internet. However, in the last 10 years, the organizations have also tried to add user-interactions to the Internet (ex. online shopping, public services, social media, news).

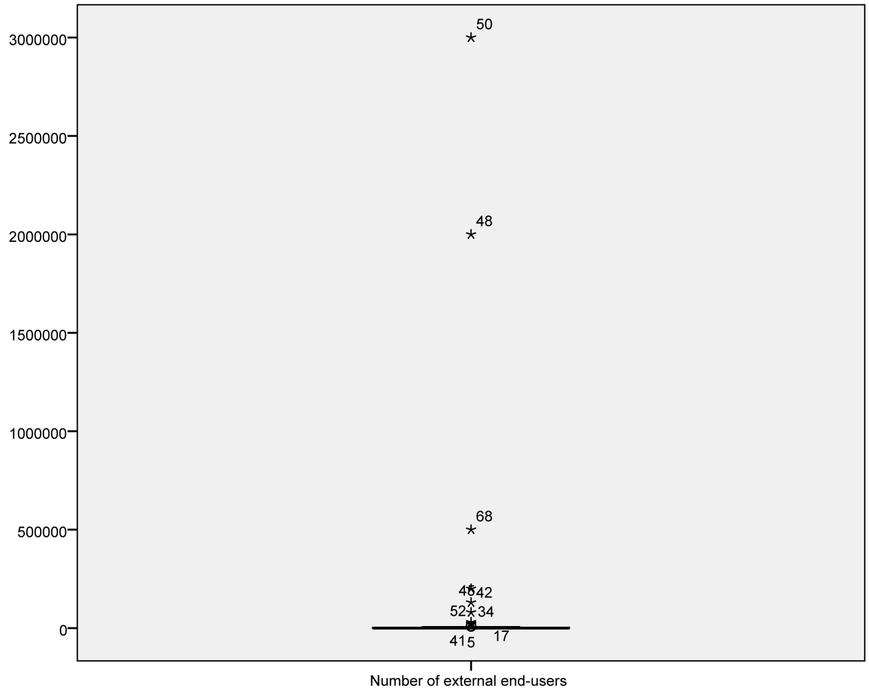

Figure 6.4: Box-plot of external end-users

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Reasonably accurate, based on good data	37	54,4	56,1	56,1
	Rough estimate, based on minimal data	17	25,0	25,8	81,8
	A best possible guess, not based on any data	12	17,6	18,2	100,0
	Total	66	97,1	100,0	

Table 6.21: Quality on answers regarding internal end-users

On answers concerning external end-users, *Table 6.21* shows that 56% are based on accurate data and 25% are based on a rough estimate. This makes these results very reliable.

6.6 Use of technology

Different technologies are used in the different organizations. This could be different programming languages, or the use of cloud or SOA systems. This section gives a presentation of the organizations' use of technology.

Language	#Org. using lang.	Percent org. using lang.	#systems using lang.	%-systems using lang.
Java	25	41,0%	100	18,2%
Script	21	34,4%	51	9,3%
C#	16	26,2%	63	11,5%
C++	12	19,7%	36	6,6%
4gl	12	17,6%	51	9,3%
Other	7	10,3%	202	36,8%
Cobol	6	9,8%	12	2,2%
C	4	6,6%	34	6,2%
Total	**103**		**549**	**100 %**

Table 6.22: Distribution of programming languages

The use of programming languages are always changing in popularity. In one decade a language may be the most used, in the next it may be the least used. From *Table 6.22* we see that Java is the most used programming language with 41% of the organizations using it. This was also the most used language in 2008 (40%). The least used programming language used is Cobol (9,8%). In 2008 it was also Cobol (5,0%) as well as Assembly (3,3%)(not asked in 2013 survey because of little usage).

It is possible to see the growing popularity of Java and script languages the last 20 years. Only 2% of all systems used Java in 1998, while in 2013 this had risen to 18,2%. In contrast, Cobol and 4GL have had a steady decrease since 1993 when 49% and 24% of the systems used them. In 2013 only 2,2% used Cobol and 9,3% used 4GL.
36,8% of the languages used in systems were categorized as "other". However this number may be exaggerated because many of the recipients did not know what language was used in some systems, and therefore put their answer in "other". Examples of this ignorance can be the case when systems are developed by an outsourced organization or when they use COTS packages.

	N	Minimum	Maximum	Mean	Std. Deviation
Total number of different languages	45	1	7	2,18	1,370
Valid N (listwise)	45				

Table 6.23: Mean number of different languages in an organization

According to *Table 6.23* the mean number of different languages in use by an organization is 2,18 (disregarding organizations with zero languages). The mean has varied between 2,0 and 2,7 in all studies (1993-2013). Regardless of what programming languages was used, the mean number of languages has been stable. The average organization focuses on 2-3 languages when it develops.

	2013		2008	
	Frequency	Percent	Frequency	Percent
Do not know	7	10,3%	6	9,2%
Not used	10	14,7%	16	24,6%
Seldom used	16	23,6%	18	27,7%
Used to some extent	13	19,1%	11	16,9%
Often used	7	10,3%	4	6,2%
Almost always used	9	13,2%	4	6,2%
Total	62		59	

Table 6.24: Use of SOA

Table 6.24 presents the distribution of the use of SOA. 66% of the organizations used SOA to some extent, and 14,7% did not use it at all. However, only 23,5% used SOA always or often. The use of SOA has had a minor increase since 2008 when it was used in some extent by 57% of the organizations. 12,4% used SOA always or often, meaning twice as many organizations used it often in 2013 than 2008.

SOA was only part of the investigation in 2008 and 2013, but the numbers indicate that the use of SOA has increased. Lewis and Smith stated in 2008 that by the year 2010, 80% of all operational applications and business processes would use SOA[34]. We have no numbers of how many systems use SOA, but our results indicate that the use of SOA has not increased that much.

	N	Minimum	Maximum	Sum	Mean	Std. Deviation
Number of main systems using Cloud	62	0	8	44	,71	1,508
Valid N (listwise)	62					

Table 6.25: Systems use of cloud

When asking about cloud, it was specified that we meant main systems running on cloud, and not minor applications (example e-mail and general storage applications). *Table 6.25* shows that organizations have a mean of 0,71 main systems in cloud. 44 main systems use cloud, which is 5,7% of all main systems. This is the first time questions about cloud were part of this survey.

	Neg. degree (1)		Minor degree (2)		Neutral (3)		Some degree (4)		Large degree (5)		
	Count	Row N %	Count	Row N %	Count	Row N %	Count	Row N %	Count	Row N %	**Avg.**
Easier with overall maintenance	0	0,0%	1	5,6%	3	16,7%	4	22,2%	10	55,6%	**4,3**
Reduction of operation	0	0,0%	0	0,0%	4	23,5%	9	52,9%	4	23,5%	**4,0**
Reduction of hardware costs	0	0,0%	4	22,2%	2	11,1%	7	38,9%	5	27,8%	**3,7**
More satisfied users	0	0,0%	2	11,1%	8	44,4%	5	27,8%	3	16,7%	**3,5**
Reduction of user support	0	0,0%	3	16,7%	8	44,4%	7	38,9%	0	0,0%	**3,2**
Reduction of software costs	0	0,0%	5	27,8%	8	44,4%	4	22,2%	1	5,6%	**3,1**
Easier to implement new functions	2	11,1%	3	16,7%	8	44,4%	3	16,7%	2	11,1%	**3,0**
Easier to correct minor errors	1	5,6%	3	16,7%	10	55,6%	3	16,7%	1	5,6%	**3,0**
Easier to correct major errors	1	5,6%	4	22,2%	10	55,6%	1	5,6%	2	11,1%	**2,9**

Table 6.26: Factors influenced by cloud

Table 6.26 presents how cloud systems influence different factors in the organization. The factors listed are characteristics that cloud are supposed to influence in a positive way. The question was categorized with a number of importance, ranging from 1 (negative degree) to 5 (large degree). All factors have an average of at least neutral degree (mean of about 3). Two organizations (11,1%) listed that it is harder to implement new functions, making that category the worst with a mean of 3,0, together with that it is easier to correct errors (mean of 2,9). It was expected that the reduction of hardware costs had a high average of 3,7. A guess is also that most of the organizations use special cloud-server vendors when putting their systems in cloud. "Easier with overall maintenance" had the highest mean with 4,3. This may be because the errors and maintenance tasks are often more centralized, and it is therefore easier to reach out to all clients at once.

There is no previous data on this subject because this question has not been part of previous studies.

6.7 Development of new systems

This section gives a presentation of how organizations plan and execute the development of new systems.

	2013		2008	
	Frequency	Percent	Frequency	Percent
No plan	22	36,1%	22	38,60 %
A desire	12	19,7%	18	31,60 %
A definite plan	7	11,5%	6	10,50 %
Already startet implementing	20	32,7%	11	19,30 %
Total	61		57	

Table 6.27: Organizations plan towards implementing SOA

From *Table 6.27* we see that 44% of the organizations have already started or have a definite plan to implement SOA, compared to 36% who have no plan to implement SOA at all. Compared to 2008, 39% had no plan to implement SOA, which is pretty much the same. But only 30% had already started or had a definite plan to implement SOA. This is an increase of one third between 2008 and 2013. These numbers correspond with Lewis and Smith[34] stating that SOA would increase between 2007 and 2010.

The same amount had no plan to implement SOA in 2013 and 2008. However, less had a desire to implement it in this study compared to 2008. The desire to implement SOA was reduced from 31,6% in 2008 to 19,7% in 2013.

		Frequency	Percent
Yes	**Total**	**14**	**22,6**
	Public	8	20,5
	Private	6	26,1
No, but have started introducing this	**Total**	**15**	**24,2**
	Public	9	23,1
	Private	6	26,1
No, and have no plan to implement this	**Total**	**33**	**53,2**
	Public	22	56,4
	Private	11	47,8
Total		62	100,0

Table 6.28: Use of cloud

Table 6.28 displays that about half of the organizations have no plan to implement cloud and the other half have implemented or started to implement cloud. Only 22,6% have already implemented cloud in their organizations. This is the first time cloud was part of this study and can therefore not be compared to the previous studies.

When separating private and public organizations, there were no major differences between these two sectors.

	N	Min	Max	Sum	Mean	Std. Deviation
#Systems being developed	61	0	5	96	1,57	1,431
#Replacement systems among total systems being developed	53	0	3	56	1,06	,989
Valid N (listwise)	53					

Table 6.29: Number of systems being developed

From *Table 6.29* we can see that the mean number of systems being developed in an organization is 1,57 (2008 - 1,53; 2003 - 0,74; 1998 - 1,58; 1993 - 1,92).

This number might be lower than it should, because it includes organizations that do not develop at all. But it is hard to sort the organizations who never develop, from organizations who just do not develop at that particular moment. When removing organizations with zero systems under production, the mean number was 2,29.

Of the 96 systems being developed, 56 (58%) of these were replacement systems. Since 1998, it seems that the share of replacement systems has stayed more or less around 60% (2008 - 64%; 2003 - 60%; 1998 - 57%; 1993 - 48%).

	N	Minimum	Maximum	#systems	Percent	Mean
#systems 0-1 year	0			0		
#systems 1-3 years	2	1	1	2	15,4%	1,00
#systems 3-6 years	12	1	5	19	24,4%	1,58
#systems 6-10 years	17	1	4	25	22,6%	1,47
#ystems 10+ years	9	1	10	22	37,6%	2,44
Average age system being replaced	**35**	**4,50**	**15,00**			**8,25**

Table 6.30: Distribution of age among main systems being replaced

Table 6.30 presents the distribution of systems based on age categories. The average age was quite high, with 8,25 years (2008 - 6,9; 2003 - 5,5; 1998 - 10,5; 1993 - 8,5). After a 10 year period when age of systems being replaced was low (1998 - 2008), the age has increased towards 2013. This is a reflection of that in 2003, the majority of systems were 3-6 years old (50%), in 2008 the majority were 6-10 years old (33,3%) and in 2013 the majority were 10+ years old (37%). This could indicate that today's systems are more solid and therefore last longer. Looking at the decrease in development, and the increase in maintenance the last years, it could also indicate that organizations prioritize maintenance on old systems, instead of developing new replacement systems.

	Not Relevant (1)		Less important (2)		Some important(3)		Important (4)		Very important (5)		
	Count	Row N %	Count	Row N %	Count	Row N %	Count	Row N %	Count	Row N %	**Avg.**
Integration with other systems	9	18,8%	2	4,2%	10	20,8%	13	27,1%	14	29,2%	**3,44**
Difficult to maintain existing systems	9	18,4%	3	6,1%	10	20,4%	15	30,6%	12	24,5%	**3,37**
Standardization with the rest of the organization	10	20,4%	3	6,1%	13	26,5%	12	24,5%	11	22,4%	**3,22**
Difficult to operate exisiting systems	9	18,4%	6	12,2%	18	36,7%	9	18,4%	7	14,3%	**2,98**
Difficult to use exisiting system	8	16,7%	6	12,5%	21	43,8%	8	16,7%	5	10,4%	**2,92**
Transition to a new technical architecture	14	29,2%	7	14,6%	12	25,0%	10	20,8%	5	10,4%	**2,69**
Transition to SOA	17	35,4%	8	16,7%	9	18,8%	9	18,8%	5	10,4%	**2,52**
There are alternative package solution	20	41,7%	3	6,3%	11	22,9%	11	22,9%	3	6,3%	**2,46**
Other	23	57,5%	1	2,5%	10	25,0%	2	5,0%	4	10,0%	**2,08**
There are alternative application generators	27	56,3%	7	14,6%	12	25,0%	2	4,2%	0	0,0%	**1,77**

Table 6.31: Reasons for why systems are replaced

In the survey it was asked why systems were replaced by new systems. *Table 6.31* gives a presentation of this data with a grade from 1 (not relevant) to 5 (very important). The two largest reasons for replacing a system were the difficulty maintaining existing systems (mean - 3,37) and integration with other systems (3,44). These reasons were also the largest in 2008 (3,7 and 3,7), and were at the top in many of the other studies. The least important reason for replacing a system was the alternate package applications (mean - 1,77). This reason has been one of the lowest throughout all of the studies (2008 - 1,9; 2003 - 1,9; 1998 - 1,6; 1993 - 1,8). By looking at these results, there are no major differences between the investigations for why organizations replace their systems.

	Do not develop		Almost nothing (1)		Little (2)		Some (3)		much (4)		Very much (5)		
	Count	Row N %	Count	Row N %	Count	Row N %	Count	Row N %	Count	Row N %	Count	Row N %	**Avg**
Spesification	25	41,0%	4	6,6%	5	8,2%	9	14,8%	11	18,0%	7	11,5%	**3,3**
Design	26	42,6%	10	16,4%	10	16,4%	6	9,8%	5	8,2%	4	6,6%	**2,5**
Code	27	44,3%	14	23,0%	6	9,8%	8	13,1%	4	6,6%	2	3,3%	**2,2**

Table 6.32: Re-use of design, code and specification

Table 6.32 displays how much the organizations re-used specification, design and code. The scale was set from 1 (almost nothing) to 5 (very much). All three categories were quite low with an average between 2,2 and 3,3. Specification was reused the most with an average of 3,3. This was also the highest and lowest in 2008 (specification - 2,58, design - 1,98, code 1,65).

A positive side to this is that all categories have increased since 2008. The reason for this is probably because in this study a check-box was added stating "we do not develop". In 2008 it was mandatory to answer, so their answers were probably put in "almost nothing" instead. If all answers from "do not develop" were put in "almost nothing", we get a mean specification - 2,37; design - 1,86; code - 1,68. These numbers are lower than 2008. Considering all this, reuse has probably been relatively stable since 2008.

In studies before 2008, specification and design was a combined category, and this years data is therefore hard to compare with those numbers. Combining specification and design, 24% reuses almost nothing or little of the specification and design (2008 - 47%; 2003 - 66%; 1998 - 53%; 1993 - 52%). 23% reuse almost no code in this study (2008 - 68%; 2003 - 36,9%; 1998 - 74%; 1993 - 86%). This is a huge difference, but again we have to consider that this in this study, participants could answer "do not develop" instead of putting their answer in "almost nothing".

6.8 Methods & tools

During development and maintenance it is normal to use some sort of method or tool. This can make the tasks easier, which saves time. It can also make results more robust, decreasing the work flow at later times. These tools can influence an organizations distribution of work by great numbers. That is why questions and tools are part of this study and are presented in this section.

	Frequency	Percent	Cumulative Percent
Not used	27	55,1	55,1
Used	22	44,9	100,0

Table 6.33: Use of methods during development or maintenance

During maintenance and development, *Table 6.33* displays that only 44,9% uses a pre-defined method. This was the same as in 2008, but a bit lower than previous studies (2008 - 45%; 2003 - 57%; 1998 - 51%).

Table 6.34 presents in which phases a method has been used. This year's numbers are drastically lower than all previous studies. A reason for this may be a change in the question format. In previous studies it was just a check-box, while this year it was a text-box where the recipient was asked to write in what method that was used. It is feared that even if a person knew that a method was used, but not the name of it, he would not write anything. Because almost the same amount wrote that they used the same method as in 2008, this seems to be a believable reason. There was a drop in the use of methods from 2003 (average 40%) to 2008 (average 36%), but a continuous drop from 2008 to 2013 (average 17,5%) is half as much. However, the distribution of use was very similar in this study as previous studies. E.g "Project management" had the most usage of methods in 2013, this was also much used in the previous studies. "Analysis" was the least used in this investigation, and has also been used little in previous investigations. It is with this possible to see similarities between studies, and that probably not much has changed.

	2013		2008		2003		1998	
	#used	Percent	#used	Percent	#used	Percent	#used	Percent
Planning	8	16,3%	17	31,5%	20	43,5%	18	34,0%
Analysis	6	12,2%	13	24,1%	11	23,9%	16	30,2%
Requirement specification	7	14,3%	26	48,2%	26	56,5%	27	50,9%
Design	7	14,3%	18	33,3%	21	45,7%	21	39,6%
Implementation	11	22,4%	21	38,9%	24	52,2%	23	43,4%
Testing	11	22,4%	24	44,4%	25	54,3%	18	34,0%
Deployment	10	20,4%	18	33,3%	15	32,6%	14	26,4%
Operation	13	26,5%	22	40,7%	17	37,0%	17	32,1%
Maintenance	12	24,5%	16	29,6%	13	28,3%	16	30,2%
Project management	16	32,7%	20	37,0%	16	34,8%	22	41,5%
Program management	4	8,2%	-	-	-	-	-	-
Benefits realization	5	9,4%	-	-	-	-	-	-

Table 6.34: Use of pre-defined methods in systems lifecycle

The survey's last question was about use of organizational controls towards development and maintenance. These results are presented in *Table 6.35*. It is positive to see that the use of testing before system production has increased again since 2008 (2013 - 81,5%; 2008 - 57,%;

2003 - 75%). The same with logging of user requirements (2013 - 61%; 2008 - 37%; 2003 - 49%; 1998 - 59%; 1993 - 77%). The organizational control that was least used was costs related to maintenance and operations are charged to the users (16,7%), which has been low throughout the replications studies.

	2013		2008		2003	1998	1993
Organizational controls	#use of 54	percent	#useof 54	percent	percent	percent	percent
All changes are tested before production	44	**81,5%**	31	57,4%	75,0%	59,0%	79,0%
All user requirements are logged	33	61,1%	20	37,0%	49,0%	59,0%	77,0%
Amendments are classified by type and importance	31	57,4%	30	55,6%	64,0%	59,0%	60,0%
Users requesting change will be notified both if the proposed amendment is carried or rejected	31	57,4%	22	40,8%	51,0%	51,0%	79,0%
All modifications are documented	29	53,7%	14	**26,0%**	57,0%	51,0%	67,0%
All amendments undergo analysis and cost estimation	26	48,1%	22	40,8%	55,0%	36,0%	54,0%
Except from operation mistakes, all changes are gatheredand periodically	26	48,1%	22	40,8%	**13,0%**	51,0%	40,0%
After modification, attached documentation are updated	22	40,7%	27	50,0%	34,0%	28,0%	25,0%
Equal routines for all changes	22	40,7%	32	59,3%	40,0%	40,0%	58,0%
Equipment related to operation and maintenance are chared to the users	11	20,4%	20	37,0%	17,0%	15,0%	40,0%
A formal review of the system is performed periodically	9	16,7%	7	13,0%	**38,0%**	17,0%	8,0%
Costs related to operation and maintenance are charged to the users	9	16,7%	7	13,0%	19,0%	13,0%	31,0%

Table 6.35: Use of routines during maintenance of systems

Chapter 7

IT i praksis

In this chapter there will be a presentation of the results from the data gathered in the survey executed by 'IT i praksis'. These results will also be compared to results from our study. Getting the same (or different) results when comparing data from different populations, can help support our own results and hypothesis.

7.1 Organizations

IT-managers roles	Frequency	Percent	Cumulative Percent
Ensuring cost-effective delivery to core IT-services	62	29,1	39,4
Being proactive with new ideas and initaitives to change processes and applications	50	23,5	94,8
Collaboration with business management to improve applications	35	16,4	55,9
Ensure that new projects are delivered on time, within budget and within quality	33	15,5	71,4
Fire fighting and daily operations	22	10,3	10,3
Developing new business models that exploit technological opportunities	10	4,7	99,5
Do not know	1	0,5	100,0
Total	213	100,0	

Table 7.1: Top It-managers daily role

The results are similar. A question asked was what describes the IT-managers' daily role in the organization. These results are presented in *Table 7.1*. A majority, with 29,1%, is "ensures cost-effective delivery to core IT-services", and 23,5% is "proactive with new ideas and initiatives to change processes and applications". These roles were also at the top in our survey (23% and 24%). There were no major changes in this distribution compared to our study.

	Frequency	Valid Percent	Cumulative Percent
No integration between business- and IT-strategy	40	18,8	18,8
Integration between business- and IT-strategy	172	80,8	99,5
Do not know	1	,5	100,0
Total	213	100,0	

Table 7.2: Integration between IT-strategy and business strategy

The results are partly similar. As shown in *Table 7.2*, a majority of the organizations have an integration between business and IT-strategy, with 81%. In our study there was also a majority of integration, but it is a bit lower with 73,5%.

#employees	Frequency	Percent	*Percent 2013*
Less than 100	16	8,0	*25,0*
101-250	32	16,1	*16,2*
251-500	28	14,1	*16,2*
501-2000	71	35,7	*33,8*
More than 2000	52	26,1	*8,8*
Total	199	100	*100*

Table 7.3: Number of employees

The results are partly similar. *Table 7.1* displays the number of employees in the organizations. The majority of organizations have more than 500 employees, with 62% (own; 42,6%). Our study had the same amount of organizations from 101-2000 employees, but less organizations with more than 2000 employees. 'IT i praksis' intentionally contacts large organizations, while our study focused on a more normally distributed selection.

7.2 Distribution of work

'IT i praksis' also asked the same distribution of work questions, which was our study's central concept. This section will present and compare the distribution of work in the organizations.

	N	Minimum	Maximum	Mean	Std. Deviation
Total maintenance	208	0,00	90,00	40,36	15,16
Corrective maintenance	208	0,00	70,00	10,43	8,34
Adaptive maintenance	208	0,00	37,50	9,89	6,53
Enhancive maintenance	208	0,00	60,00	13,09	9,95
Non-functional perfective maintenance	208	0,00	26,67	6,95	4,64
Valid N (listwise)	208				

Table 7.4: Distribution of maintenance

The results are similar. *Table 7.4* shows the distribution of maintenance. These data are very similar to our own study (total - 40,36; corrective - 9,27; adaptive - 8,84; enhancive - 12,24%, non-functional - 11,14%). The only slight difference is in non-functional maintenance, which is larger in our study.

	N	Minimum	Maximum	Mean	Std. Deviation
Total development	208	0,00	100,00	17,48	14,05
Development of replacement systems	208	0,00	66,67	8,74	9,58
Development of new systems	208	0,00	33,33	8,74	7,62
Valid N (listwise)	208				

Table 7.5: Distribution of development

The results are not similar. The distribution of development can be seen in *Table 7.5*. Total development had a mean of 17,5%, with development of replacement systems and new systems even at 8,74%. This was higher than in our study, where total development was only 13,6% and development of replacement systems and new systems were at an even 6,8%. These results show that the population from 'IT i praksis' spent more time on development.

	N	Minimum	Maximum	Mean	Std. Deviation
Operations	208	0,00	80,00	23,87	13,41
Support	208	0,00	70,00	18,29	11,43
Valid N (listwise)	208				

Table 7.6: Distribution of operations and support

The results are very similar. *Table 7.6* shows that operation had a mean of 23,9%. This was similar to our study, where operation was at 22%. Support was at 18,3% while in our study it was 23%.

	N	Minimum	Maximum	Mean	Std. Deviation
Isolated maintenance	207	0,00	100,00	71,76	18,58
Isolated development	207	0,00	100,00	28,24	18,58
Application portfolio upkeep	207	11,11	100,00	63,59	17,18
Application portfolio evolution	207	0,00	88,89	36,41	17,18
Valid N (listwise)	207				

Table 7.7: Distribution of Isolated maintenance and development and application portfolio

The results are similar. In *Table 7.7* the application portfolio upkeep has a mean of 63,6%, while evolution was at 36,4%. This is quite similar to our investigation where upkeep was 68% and evolution was 32%. Isolated maintenance (maintenance when only looking at maintenance and development) had a mean of 71,8%, and isolated development 28,2%. In our study, the difference was a bit higher with an isolated maintenance of 77,7% and development of 22,3%.

	Frequency	Percent	Cumulative Percent
Reasonably accurate, based on good data	12	5,6	5,6
A rough estimate, based on minimal data	81	38,0	43,7
A best possible guess, not based on any data	120	56,3	100,0
Total	213	100,0	

Table 7.8: Quality of answers toward distribution of work

The results are not similar. A reason for the differences between our study and 'IT i praksis' may have a connection with the quality of the answers. As shown in *Table 7.8*, only 5,6% based their answers on reasonably accurate data, while more than half of the answers were a best possible guess. In our study, 12% of the answers were reasonably accurate, and 44% was a best possible guess. This may be enough of a difference to affect some of the results.

7.3 Outsourcing

'IT i praksis' also asked questions regarding outsourcing. These questions were not formulated in the same way as in our own study, so it was hard to give an accurate comparison. This section gives a presentation of what organizations outsource.

	Frequency	Percent	Cumulative Percent
Yes	151	70,9	70,9
No	62	29,1	100,0
Total	213	100,0	

Table 7.9: Do the organizations outsource services

The results are partly similar. *Table 7.9* shows that 71% of all organizations outsourced some sort of services. This is a bit less than in our survey where 85% answered the same.

	N	#	Percent
Apllication management and maintenance	213	95	44,6%
Projects and development	213	34	16,0%
Operation	213	90	42,3%
Support	213	48	22,5%
Support processes (HR, economy)	213	35	16,4%
Core processes	213	10	4,7%
Other	213	26	12,2%

Table 7.10: Distribution of what was outsourced by the organizations

The results are not similar. When it comes to what the organizations outsource, the results are presented in *Table 7.10*. The most outsourced services were "application management and maintenance" (44,6% of all organizations) and operation (42,3% of all organizations). In our survey it was asked how much of each category was outsourced, which makes comparing these data hard. On the other hand, when counting all cases with zero outsourcing as "not outsourcing", and all above zero was categorized as "outsourcing", something comparable was achieved. In our study, 67% of all organizations outsourced development and 72% outsourced maintenance. 43% outsourced support and 76% outsourced operation. All the results from 'IT i praksis' were smaller than our own study.

7.4 Private vs public sector

Both studies had a focus on differences between public and private sector. The two sectors often focus on different aspects, which may cause a difference in other results collected from this survey. This section looks at data distributed on private and public participants to see if there are any differences between the two groups.

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Private	87	40,8	40,8	40,8
	Public	126	59,2	59,2	100,0
	Total	213	100,0	100,0	

Table 7.11: Distribution of public and private organizations

The results are partly similar. Of the 213 organizations in the survey, 126 (59,2%) worked in public sector and 87 (40,8%) worked in private sector. This gives a minor majority of public organizations. In our own study the difference was a bit larger with 64% public and 36% private. Both studies had large enough gaps between private and public sector to cause differences in other results.

Sector		Frequency	Valid Percent	Cumulative Percent
Private	Yes	67	77,0	77,0
	No	20	23,0	100,0
	Total	87	100,0	
Public	Yes	84	66,7	66,7
	No	42	33,3	100,0
	Total	126	100,0	

Table 7.12: Do the organizations outsource services

Table 7.12 shows that 77% of all private organizations outsourced some services. This is a little more than public, with 66,7%.

	sector	N	#outsourcing	Percent
Apllication management and maintenance	Private	87	37	42,5%
	Public	126	58	46,0%
Projects and development	Private	87	18	20,7%
	Public	126	16	12,7%
Operation	Private	87	51	58,6%
	Public	126	39	31,0%
Support	Private	87	32	36,8%
	Public	126	16	12,7%
Support processes (HR, economy)	Private	87	18	20,7%
	Public	126	17	13,5%
Core processes	Private	87	7	8,0%
	Public	126	3	2,4%
Other	Private	87	10	11,5%
	Public	126	16	12,7%

Table 7.13: Distribution of services outsourced split between public and private sector

The results about what services was outsourced are shown in *Table 7.13*, with a split between public and private organizations. A part from "other" and "application management and maintenance", there are significant differences in all categories, with private sector outsourcing the most.

	Mean private	Mean public	diff %
Total maintenance	**43,4**	**38,3**	**11,7**
Corrective maintenance	11,4	9,8	13,9
Adaptive maintenance	9,6	10,1	6,0
Enhancive maintenance	16,1	11,1	31,1
Non-functional perfective maintenance	6,4	7,3	14,9
Total development	**21,8**	**14,6**	**33,1**
Development of replacement systems	10,4	7,6	26,9
Development of new systems	11,4	6,9	39,5
Operations	20,5	26,2	27,8
Support	14,4	21,0	45,9
Isolated maintenance	**67,5**	**74,7**	**10,6**
Isolated development	**32,5**	**25,3**	**22,0**
Application portfolio upkeep	**57,6**	**67,7**	**17,6**
Application portfolio evolution	**42,4**	**32,3**	**23,9**

Table 7.14: Distribution of work, split between public and private sector

The differences between private and public distribution of work are presented in *Table 7.14*. The far right column displays the percentage difference between the two sectors. Total development was 33% larger for private sector than public sector. This category was also larger for private sector in the amount of outsourcing.

Support is almost twice as large in public sector than private (21% vs. 14%). A total of 36% of private organizations outsourced support, compared to public, with 13%, which could mean that public organizations wish to do this themselves, while private organizations outsource it instead.

Private sector spend less time on total maintenance (public - 38%; private - 43%). However, they did spend more time on application portfolio upkeep (public - 67%; private - 57%). This could indicate that private organizations are more successful with the allocated time spent on IT.

Sector		Frequency	Valid Percent	Cumulative Percent
Private	No integration between business- and IT-strategy	10	11,5	11,5
	Integration between business- and IT-strategy	77	88,5	100,0
	Total	87	100,0	
Public	No integration between business- and IT-strategy	30	23,8	23,8
	Integration between business- and IT-strategy	95	75,4	99,2
	Do not know	1	,8	100,0
	Total	126	100,0	

Table 7.15: Integration between IT-strategy and business strategy

Table 7.15 shows that more private organizations have an integration between IT-strategy and business strategy. 88,5% of the private organizations have this integration, while 75,4% in public have the same.

	Freq. Private	% Private	Freq. Public	% public	% diff
Fire fighting and daily operations	7	8,0	15	11,9	32,4
Ensuring cost-effective delivery to core IT-services	18	20,7	44	34,9	40,75
Collaboration with business management on improvement to applications	21	24,1	14	11,1	53,97
Ensure that new projects are delivered on time, within budget and quality	15	17,2	18	14,3	14,14
Being proactive with new ideas and initaitives to change processes and applications	20	23,0	30	23,8	3,45
Developing new business models that exploit technological oportunities	6	6,9	4	3,2	53,97
Total	87		126		

Table 7.16: Top IT-managers role in daily operations

The results are similar. *Table 7.16* presents that there are noteworthy differences between private and public sector, when it comes to the description of the IT-managers' daily roles. In private sector, most leaders collaborate with business management on improvement of applications (24,1%), while only 11,1% do the same in public sector. In public sector most leaders ensure cost-effective delivery to core IT-services (34,9%), which is done by 20,7% in private. These data display that there were a majority of public IT-managers who had a focus on administrative tasks, while private IT-managers often have a larger focus on tasks toward development. This is the same for our own studies. It can also explain some of the other differences in distribution of work between private and public organizations.

7.5 Conclusion of 'IT i praksis' results

This chapter has presented results from the survey by 'IT i praksis' and compared it with our own study. Most of the results were either similar or partly similar. Examples where the results were not similar, were on the share of development and outsourcing. There may be many reasons for these differences. First of all, the population was different. Also, different mailing lists were used, which can be affected by which sources was used to collect them.
In our study, there was a focus on getting a normal distribution of participants. 'IT i praksis' had a focus on contacting larger organizations (+2000 employees). There were also a larger amount of private organizations in 'IT i praksis'(41% vs own 36%). The size of organizations and which sector they are in, may influence results like outsourcing and distribution of work. The reason for these differences could be the different focus they may have compared to smaller organizations. E.g an organization with 5 employees and no external end-users do not spend much time on support.
The quality of the answers are also different. Only 43,6% answered questions about distribution of work reasonably accurate or with a rough estimate. 56% did the same in our study.

When looking at comparisons between public and private sector, the numbers were sometimes different, but the distribution between private and public was often the same. E.g our study had a larger share of organizations outsourcing development, but in both studies private organizations outsourced more development than public organizations.

Overall, both surveys gathered similar results.

Chapter 8

Hypothesis-testing

In this chapter, the investigation's hypothesis will be tested. The hypothesis will be tested with statistical analysis, and then evaluated accordingly. There will be presented some discussions in this chapter, but the overall discussion will be presented in the discussion chapter at the end of this thesis.

8.1 Normality test

Before testing the hypothesis, a test had to be run to see if the data was normally distributed. A test like this compares the shape of the sample distribution to the shape of a normal curve. If the shape is "normally shaped", the population is normally distributed. A test that is significant is not shaped like a normal curve, and the population is in that case not normally distributed.

Both Kolmogorov Smirnov and Shapiro-Wilks normality tests were run, but because the data consisted of less than 2000 samples, it was only necessary to look at the Shapiro-Wilks test. The test results are presented in *Table 8.1.* Only total maintenance was normally distributed with a significance of 0,340 in the Shapiro-Wilks test. This was also the only normally distributed variable in 2008, when the significance was 0,269.

	Kolmogorov-Smirnov[a]			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
Total maintenance	,092	64	,200*	,979	64	,340
Total development	,158	64	,000	,864	64	,000
Operations	,172	64	,000	,925	64	,001
Support	,179	64	,000	,868	64	,000
Isolated maintenance	,164	64	,000	,871	64	,000
Isolated development	,164	64	,000	,871	64	,000
Application portfolio upkeep	,092	64	,200*	,962	64	,044
Application portfolio evolution	,092	64	,200*	,962	64	,044

*. This is a lower bound of the true significance.

a. Lilliefors Significance Correction

Table 8.1: Normality test of the investigations most central results

8.2 Maintenance and development

H1. There are no differences in the amount of time spent on maintenance and development, when only looking at maintenance and development.

	N	Mean	Std. Deviation	Minimum	Maximum
Isolated maintenance	64	77,73	20,017	0	100
Isolated development	65	22,27	22,076	0	100

Ranks

		N	Mean Rank	Sum of Ranks
Isolated maintenance - isolated development	Negative Ranks	59[a]	31,61	1865,00
	Positive Ranks	3[b]	29,33	88,00
	Ties	2[c]		
	Total	64		

a. Development < Maintenance

b. Development > Maintenance

c. Development = Maintenance

Test Statistics[a]

	Isolated development - Isolated maintenance
Z	-6,257[b]
Asymp. Sig. (2-tailed)	,000

a. Wilcoxon Signed Ranks Test

b. Based on positive ranks.

Table 8.2: Isolated maintenance vs isolated development

H1 is rejeceted. *Table 8.2* illustrates that more time was spent on maintenance than development (when only looking at development and maintenance) in 59 of 64 cases, making it a significant amount. H1 was also rejected by results gathered in 'IT i praksis'.

H2. There are no differences in the amount of time spent on maintenance and development.

	N	Mean	Std. Deviation	Minimum	Maximum
Total maintenance	67	41,48	20,249	0	100
Total development	67	13,63	14,233	0	70

Ranks

		N	Mean Rank	Sum of Ranks
Total development - Total maintenance	Negative Ranks	59[a]	31,22	1842,00
	Positive Ranks	3[b]	37,00	111,00
	Ties	5[c]		
	Total	67		

a. Total development < Total maintenance

b. Total development > Total maintenance

c. Total development = Total maintenance

Test Statistics[a]

	Total development - Total maintenance
Z	-6,070[b]
Asymp. Sig. (2-tailed)	,000

a. Wilcoxon Signed Ranks Test

b. Based on positive ranks.

Table 8.3: Total maintenance vs total development

H2 is rejected. *Table 8.3* displays that total maintenance was larger than total development in 59 of 67 cases. H2 was also rejected with results from 'IT i praksis'.

H3. There are no differences between time spent on application portfolio upkeep and traditional maintenance, when only looking at development and maintenance.

	N	Minimum	Maximum	Mean	Std. Deviation
Isolated maintenance	64	0	100	77,73	20,02
Application portfolio upkeep	64	31	100	68,07	19,24
Valid N (listwise)	64				

Ranks

		N	Mean Rank	Sum of Ranks
Application portfolio upkeep - Isolated maintenance	Negative Ranks	38[a]	23,83	905,50
	Positive Ranks	9[b]	24,72	222,50
	Ties	17[c]		
	Total	64		

a. Application portfolio upkeep < Isolated maintenance
b. Application portfolio upkeep > Isolated maintenance
c. Application portfolio upkeep = Isolated maintenance

Test Statistics[a]

	Application portfolio upkeep - Isolated maintenance
Z	-3,615[b]
Asymp. Sig. (2-tailed)	,000

a. Wilcoxon Signed Ranks Test

b. Based on positive ranks.

Table 8.4: Application portfolio upkeep vs isolated maintenance

H3 is rejected. *Table 8.4* displays that in 38 of 64 cases, isolated maintenance was larger than application portfolio upkeep. H3 was also rejected with results from 'IT i praksis'.

H4. There are no differences between time spent on application portfolio evolution and traditional development, when only looking at development and maintenance.

	N	Minimum	Maximum	Mean	Std. Deviation
Isolated development	65	0	100,00	22,27	22,08
Application portfolio evolution	64	0	69,23	31,93	19,24
Valid N (listwise)	64				

Ranks

		N	Mean Rank	Sum of Ranks
Application portfolio evolution - Isolated development	Negative Ranks	9[a]	29,72	267,50
	Positive Ranks	43[b]	25,83	1110,50
	Ties	12[c]		
	Total	64		

a. Application portfolio evolution < Isolated development

b. Application portfolio evolution > Isolated development

c. Application portfolio evolution = Isolated development

Test Statistics[a]

	Application portfolio evolution - Isolated development
Z	-3,840[b]
Asymp. Sig. (2-tailed)	,000

a. Wilcoxon Signed Ranks Test

b. Based on negative ranks.

Table 8.5: Application portfolio evolution vs isolated development

H4 is rejected. *Table 8.5* shows that application portfolio evolution was larger than total development in 43 of 64 cases. H4 was also rejected with results from 'IT i praksis'.

H5. There are no differences between time spent on application portfolio evolution and application portfolio upkeep.

	N	Mean	Std. Deviation	Minimum	Maximum
Application portfolio upkeep	64	68,07	19,24	31	100
Application portfolio evolution	64	31,93	19,24	0	69

Ranks

		N	Mean Rank	Sum of Ranks
Application portfolio upkeep - Application portfolio evolution	Negative Ranks	10[a]	15,00	150,00
	Positive Ranks	50[b]	33,60	1680,00
	Ties	4[c]		
	Total	64		

a. Application portfolio upkeep < Application portfolio evolution

b. Application portfolio upkeep > Application portfolio evolution

c. Application portfolio upkeep = Application portfolio evolution

Test Statistics[a]

	Application portfolio upkeep - Application portfolio evolution
Z	-5,635[b]
Asymp. Sig. (2-tailed)	,000

a. Wilcoxon Signed Ranks Test

b. Based on negative ranks.

Table 8.6: Application portfolio upkeep vs application portfolio evolution

H5 is rejected. From *Table 8.6* the application portfolio upkeep was larger than application portfolio evolution in 50 of 64 cases, leaving application portfolio evolution larger in only 10 cases. H5 was also rejected with results from 'IT i praksis'.

8.3 Type of organizations

H6. There are no differences in the distribution of work between organizations with many employees and organizations with fewer employees.

			Total maintenance	Total development	Operations	Support	Application portfolio upkeep	Application portfolio evolution
Spearman	#employees	Correlation Coefficient	,045	-,111	,282*	-,009	,259*	-,258*
		Sig. (2-tailed)	,716	,371	,021	,939	,039	,039
		N	67	67	67	67	64	64

*. Correlation is significant at the 0.05 level (2-tailed).

**. Correlation is significant at the 0.01 level (2-tailed).

Table 8.7: Numbers of employees vs distribution of work

H6 is rejected. Correlation results presented in *Table 8.7* indicate that there was a difference in work distribution. Operations have a significant correlation coefficient of ,282 and the application portfolios a correlation of +-,259. This indicates that larger organizations spent less time on evolution. The correlation was low and only on some of the variables, but it was still existing.

h7. An organization's distribution of work is not affected by the top IT-manager role-priority.

Describe top managers role today		Total maintenance	Total development	Operations	Support	Application portfolio upkeep	Application portfolio evolution
Being proactive with new ideas and initiatives to change processes and applications	Mean	44,09	11,29	19,14	25,47	74,32	25,68
	N	16	16	16	16	15	15
	Std. Deviation	23,692	13,898	12,448	26,912	21,204	21,204
Collaboration with business management on improvements to applications	Mean	52,65	18,59	18,59	**10,18**	54,95	**45,05**
	N	10	10	10	**10**	10	**10**
	Std. Deviation	25,231	16,768	12,511	**6,941**	21,134	**21,134**
Developing new business models that exploit technological opportunities	Mean	42,86	**28,58**	**12,75**	15,82	76,75	23,25
	N	5	**5**	**5**	5	5	5
	Std. Deviation	20,917	**24,388**	**9,600**	17,063	16,978	16,978
Ensure that new projects are delivered on time, within budget and quality	Mean	40,60	19,66	20,95	18,79	59,54	40,46
	N	9	9	9	9	8	8
	Std. Deviation	18,134	12,478	12,723	14,715	13,590	13,590
Ensuring cost-effective delivery of core IT services	Mean	36,51	9,46	28,54	25,49	73,40	26,60
	N	15	15	15	15	15	15
	Std. Deviation	13,842	8,276	17,320	13,356	17,832	17,832
Fire fighting and daily operations	Mean	33,38	**5,85**	24,17	**36,60**	65,96	34,04
	N	11	**11**	11	**11**	10	10
	Std. Deviation	18,048	**7,742**	13,087	**24,700**	16,160	16,160
Total	Mean	41,31	13,52	21,80	23,37	68,01	31,99
	N	66	66	66	66	63	63
	Std. Deviation	20,356	14,312	13,970	20,408	19,385	19,385

Table 8.8: Distribution of work vs top IT-managers role-priority

H7 is rejected. *Table 8.8* displays organization's work distribution distributed on their IT-manager role-priority.
There are no significant differences on total maintenance.
On total development, "Developing new business models that exploit technological opportunities" has a mean of 28,58, which is significantly larger than the main average, which is 13,52. Also, "Fire fighting and daily operations", has a significant difference with a mean as low as 5,85 on total development. With support, "Top IT manager spend most time on collaboration with business management on improvements to applications" has a mean of 10,18, which is significantly lower than the mean of total support, which is 23,37. These leaders have also a major growth in application portfolio evolution with a mean of 45.

In *Table 8.9* below, the same results from the 'IT i praksis' survey occur. These results are very similar and have much of the same distribution as our survey, which builds support to our findings.

Describe IT-manager role today		Total maintenance	Total development	Operations	Support	Application portfolio upkeep	Application portfolio evolution
Being proactive with new ideas and initaitives to change processes and applications	Mean	39,85	19,03	22,98	18,13	61,56	38,44
Improvement to applications	Mean	46,00	19,53	18,39	16,08	54,47	45,53
Developing new business models that exploit technological oportunities	Mean	38,44	29,11	21,11	11,33	53,59	46,41
Ensure that new projects are delivered on time, within budget and quality	Mean	43,69	21,51	20,86	13,93	62,31	37,69
Ensuring cost-effective delivery to core IT-services	Mean	37,73	13,74	26,59	21,95	69,39	30,61
Fire fighting and daily operations	Mean	35,73	10,21	32,56	21,50	72,96	27,04

Table 8.9: Distribution of work vs top IT-managers role-priority from the 'IT i praksis'-survey

8.4 Importance of IT

H8. There are no differences in the distribution of work between organizations where the size of the IT-department compared to the total number of employees is large, and the organizations where the size of the IT-department compared to the total number of employees is small.

			Total maintenance	Total development	Operations	Support	Application portfolio upkeep	Application portfolio evolution
Spearman's	Employees IT / Total Employees	Correlation Coefficient	,116	,450**	-,090	-,356**	-,433**	,433**
		Sig. (2-	,349	,000	,467	,003	,000	,000
		N	67	67	67	67	64	64

**. Correlation is significant at the 0.01 level (2-tailed).

*. Correlation is significant at the 0.05 level (2-tailed).

Table 8.10: Employees IT-department / total employees vs distribution of work

H8 is rejected. *Table 8.10* shows that there was a significant correlation in all categories except total maintenance and operations. There was a large significant difference in total development and application portfolio evolution. This means that the more IT employees an organization has, the more development is executed. There was also less support and less portfolio upkeep.

H9. There are no differences in the distribution of work between organizations in which there are many system-developers in proportion to total number of internal users, and organizations with few system-developers in proportion to total number of internal users.

			Total maintenance	Total development	Operations	Support	Application portfolio upkeep	Application portfolio evolution
Spearman	system developers / Internal users	Correlation Coefficient	,152	,523**	-,109	-,453**	-,369**	,369**
		Sig. 2-tailed	,221	,000	,379	,000	,003	,003
		N	67	67	67	67	64	64

**. Correlation is significant at the 0.01 level (2-tailed).

*. Correlation is significant at the 0.05 level (2-tailed).

Table 8.11: Number of system developers / internal end-users vs distribution of work

H9 is rejected. *Table 8.11* indicates that the number of system developers compared to internal users correlates with total development, support and portfolio evolution/upkeep. The share of total development increases in parallel with the number of system developers. Support decreases when system developers increase.

H10. There are no differences in the distribution of work between organizations in which there are many system-developers in proportion to total number of employees in the IT department, and organizations with few system-developers in proportion to total number of employees in the IT department.

			Total maintenance	Total development	Operation	Support	Appl. portfolio upkeep	Appl. portfolio evolution
Spearman	Number of developers / Employees IT-departement	Correlation Coefficient	,155	,463**	-,150	-,397**	-,342**	,341**
		Sig. (2-tailed)	,215	,000	,230	,001	,006	,006
		N	66	66	66	66	64	64

**. Correlation is significant at the 0.01 level (2-tailed).

*. Correlation is significant at the 0.05 level (2-tailed).

Table 8.12: Number of developers / number of employees IT-department vs distribution of work

H10 is rejected. *Table 8.12* displays a clear correlation in total development, support and application portfolio. Total development increases and support decreases as the number of developers grow. It is natural that the more developers an organizations have, the more time is spent on development and application portfolio evolution.

H11. There are no differences in the distribution of work in organizations where IT- and business strategy are integrated, and where this is not the case.

Ranks

Describe top leader grouped		N	Mean	Mean Rank	Sum of Ranks
Total maintenance	No integration between business- and IT-strategy	15	43,4	32,8	492,00
	Integration between business- and IT-strategy	49	40,7	32,4	1588,00
	Total	64			
Total development	No integration between business- and IT-strategy	15	14,8	32,9	494,00
	Integration between business- and IT-strategy	49	13,3	32,4	1586,00
	Total	64			
Operations	No integration between business- and IT-strategy	15	19,5	31,6	474,50
	Integration between business- and IT-strategy	49	22,2	32,8	1605,50
	Total	64			
Support	No integration between business- and IT-strategy	15	22,3	32,5	487,50
	Integration between business- and IT-strategy	49	23,8	32,5	1592,50
	Total	64			
Application portfolio upkeep	No integration between business- and IT-strategy	15	70,2	33,7	505,00
	Integration between business- and IT-strategy	46	66,7	30,1	1386,00
	Total	61			
Application portfolio evolution	No integration between business- and IT-strategy	15	29,8	28,3	425,00
	Integration between business- and IT-strategy	46	33,3	31,9	1466,00
	Total	61			

Table 8.13: Presentation of mean of Top IT-leaders role-priority vs distribution of work

Test Statistics[a]

	Total maintenance	Total development	Operations	Support	Application portfolio upkeep	Application portfolio evolution
Mann-Whitney U	363,000	361,000	354,500	367,500	305,000	305,000
Wilcoxon W	1588,000	1586,000	474,500	1592,500	1386,000	425,000
Z	-,072	-,105	-,207	0,000	-,671	-,671
Asymp. Sig. (2-tailed)	,943	,916	,836	1,000	,502	,502

a. Grouping Variable: Describe top leader grouped

Table 8.14: Test statistics of mean ranks from *Table 8.13*

H11 is not rejected. In *Table 8.14* none of the work categories are significant. When looking at *Table 8.13* the mean of the two integration categories were very similar in every work category. The integration of business ant IT-strategy was not related to the distribution of work.

Neither in 'IT i praksis' were these results significantly different.

8.5 Consultants and employees

H12. There are no differences in the distribution of work between organizations with high average experience among developers, and organizations with low average experience among developers.

			Total maintenance	Total development	Operations	Support	Application portfolio upkeep	Application portfolio evolution
Spearman's	Average years experience	Correlation Coefficient	,054	-,186	-,186	,147	,096	-,096
		Sig. (2-tailed)	,740	,251	,249	,367	,555	,555
		N	40	40	40	40	40	40

**. Correlation is significant at the 0.01 level (2-tailed).

*. Correlation is significant at the 0.05 level (2-tailed).

Table 8.15: Developers average years experience vs distribution of work

H12 is not rejected. In *Table 8.15* there are no significant correlation between developers experience and the distribution of work.

H13. The number of hired consultants in an organization does not affect its distribution of work.

			Total maintenance	Total development	Operation	Support	Application portfolio upkeep	Application portfolio evolution
Spearman	Average number of consultants IT-department	Correlation Coefficient	,194	,287*	-,108	-,225	-,110	,111
		Sig. (2-tailed)	,116	,019	,384	,067	,388	,384
	Hired consultants / employees IT-department	Correlation Coefficient	,154	,218	-,237	-,081	-,110	,111
		Sig. (2-tailed)	,217	,079	,056	,519	,386	,384
		N	66	66	66	66	64	64

**. Correlation is significant at the 0.01 level (2-tailed).

*. Correlation is significant at the 0.05 level (2-tailed).

Table 8.16: Number of hired consultants vs distribution of work

H13 is not rejected. However, *Table 8.16* displays that there was a significant correlation in total development. More of the organization's resources seem to go to development when many consultants are hired. This is not the case when comparing hired consultants up against total numbers of employees in the IT department. Because the correlation is low (0,287) and not significant when including total number of employees, H13 is categorized as not rejected.

8.6 Complexity of the portfolio

H14. There are no differences in the distribution of work between organizations with many main systems and organizations with fewer main systems.

			Total maintenance	Total development	Operations	Support	Application portfolio upkeep	Application portfolio evolution
Spearman	#-main systems	Correlation Coefficient	,071	-,172	,265*	-,006	,313*	-,313*
		Sig. (2-tailed)	,572	,172	,033	,964	,013	,013
		N	65	65	65	65	62	62
	#-main systems / #-employees	Correlation Coefficient	,039	-,010	-,203	-,030	-,135	,135
		Sig. (2-tailed)	,759	,937	,105	,810	,296	,297
		N	65	65	65	65	62	62

*. Correlation is significant at the 0.05 level (2-tailed).

**. Correlation is significant at the 0.01 level (2-tailed).

Table 8.17: Number of main systems vs distribution of work

H14 is partly rejected. *Table 8.17* shows that operations (correlation of ,265) and application portfolio (+-,313) have a significant correlation with the number of main systems in an organization. It is natural that organizations, with many main systems, have allocated much time on application portfolio evolution.

H14 is only partly rejected because when looking at the number of main systems compared to the size of the organization, there are no correlations.

H15. There are no differences in the distribution of work between organizations with many end-users and organizations with fewer end-users.

			Total maintenance	Total development	Operations	Support	Application portfolio upkeep	Application portfolio evolution
Spearman	#internal end-users	Correlation Coefficient	-,009	-,176	,340**	,033	,261*	-,262*
		Sig. (2-	,944	,161	,006	,794	,040	,040
		N	65	65	65	65	62	62
	#external end-users	Correlation Coefficient	,183	,313*	-,055	-,352**	-,192	,193
		Sig. (2-	,144	,011	,664	,004	,135	,132
		N	65	65	65	65	62	62
	#total end-users	Correlation Coefficient	,187	,067	,085	-,262*	,018	-,017
		Sig. (2-	,136	,593	,499	,035	,888	,898
		N	65	65	65	65	62	62

**. Correlation is significant at the 0.01 level (2-tailed).

*. Correlation is significant at the 0.05 level (2-tailed).

Table 8.18: Number of end-users vs distribution of work

H15 is rejected. *Table 8.18* states that there was significant correlation on all types of users in the organizations.

The number of internal end-users correlates with operation (,340) and application portfolio (+-,261).

The number of external end-users correlates with total development (,313) and support (-,352).

When combining internal and external end-users (total number of end-users), there is a significant correlation on support (-,262).

It was unexpected that support decreases in organizations with many external end-users, one should have thought that more end-users would increase support.

H16. There are no differences in distribution of work between organizations with main-systems having a high average age, and organizations with main-systems having a low average age.

			Total maintenance	Total development	Operations	Support	Application portfolio upkeep	Application portfolio evolution
Spearman's	Average age main systems	Correlation Coefficient	-,207	-,087	,254*	,034	,026	-,027
		Sig. 2-tailed	,097	,492	,041	,790	,843	,837
		N	65	65	65	65	62	62

*. Correlation is significant at the 0.05 level (2-tailed).

**. Correlation is significant at the 0.01 level (2-tailed).

Table 8.19: Average age main systems vs distribution of work

H16 is rejected. From *Table 8.19* it is possible to deduct that operations were significantly correlated with ,254. The correlation was small, but existing. This indicates that when the average age on main systems grows, operations also grow.

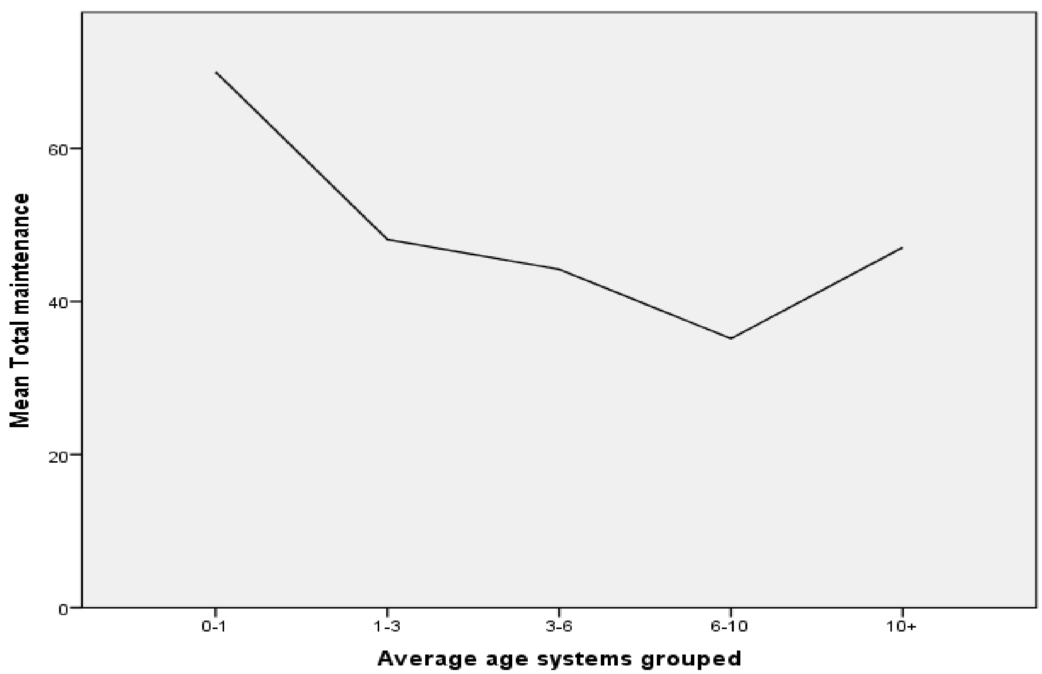

Figure 8.1: Time spent on total maintenance compared to the average age of main systems

As a side note, *Figure 8.1* displays a graph of the organizations' average age of all their main systems, compared to time spent on total maintenance. When a system is put in production, it is not always completely finished, but enough to be functional. Some errors not covered in the testing phase, are also fixed after launch. A moderate amount of modifications are therefore performed, and this is why a system usually goes through a lot of maintenance in its first year.

This is reflected in our investigation, where the maintenance is highest for organizations who have an average age of systems at 0-1 years. As the system gets older, less maintenance work is spent on it. Until it reaches 10+ years, time spent on maintenance increases again. This is discussed by Bhatt and Shroff[6], where maintenance gets more time consuming after some years because the structure and code get more untidy, together with key personnel leaving the project.

H17. There are no differences in the distribution of work between organizations that use many different programming-languages, and organizations that use fewer different programming-languages.

			Total maintenance	Total development	Operations	Support	Application portfolio upkeep	Application portfolio evolution
Spearman's	#different languages	Correlation Coefficient	,140	,234	,058	-,253	-,351*	,351*
		Sig. 2-tailed	,365	,126	,708	,097	,021	,021
		N	44	44	44	44	43	43

*. Correlation is significant at the 0.05 level (2-tailed).

**. Correlation is significant at the 0.01 level (2-tailed).

Table 8.20: Number of different programming languages vs distribution of work

H17 is rejected. *Table 8.21* displays that the total number of different languages correlates with application portfolio upkeep and evolution(+-,351). Organizations who use many languages usually have many systems, which explains the increase of application portfolio evolution.

8.7 Use of methods and tools

H18. There are no differences in the distribution of work between organizations that use pre-defined methods throughout the system's life cycle, and the organizations that do not use this.

			Total maintenance	Total development	Operations	Support	Application portfolio upkeep	Application portfolio evolution
Spearman	Have used method	Correlation Coefficient	,053	,307*	,137	-,309*	-,188	,188
		Sig. (2-tailed)	,719	,034	,354	,033	,211	,211
		N	48	48	48	48	46	46

*. Correlation is significant at the 0.05 level (2-tailed).

**. Correlation is significant at the 0.01 level (2-tailed).

Table 8.21: Organizations who have used a method during a systems life cycle vs distribution of work

H18 is rejected. From *Table 8.21* it is possible to see that there is a significant correlation in total development (,307) and support (-,309). This states that organizations that use methods, also spend more time on developing, and less time on support. We were hoping to see a decrease in maintenance here, but there was no correlation. However, the increase in development shows that the use of methods do help.

H19. There are no differences in the distribution of work in organizations with a high number of routines established for management and maintenance of IT-systems, compared to organizations with less routines for this.

			Total maintenance	Total development	Operations	Support	Application portfolio evolution	Application portfolio upkeep
Spearman's	#routines	Correlation Coefficient	,058	,172	,198	-,275*	,091	-,091
		Sig. (2-tailed)	,681	,219	,156	,046	,532	,532
		N	53	53	53	53	50	50

*. Correlation is significant at the 0.05 level (2-tailed).

**. Correlation is significant at the 0.01 level (2-tailed).

Table 8.22: Number of routines for management and maintenance of IT-systems

H19 is partly rejected. *Table 8.22* shows that organizations with many routines for management and maintenance, spend less time on support (correlation of -,275). Because routines only correlate with support, and the correlation is low, H19 is only considered partly rejected.

H20. When developing replacement systems, it is easier to reuse specifications and design, than code.

	Almost nothing (1)		Little (2)		Some (3)		much (4)		Very much (5)		Avg
	Count	Row N %	Count	Row N %	Count	Row N %	Count	Row N %	Count	Row N %	
Spesification	4	6,6%	5	8,2%	9	14,8%	11	18,0%	7	11,5%	3,3
Design	10	16,4%	10	16,4%	6	9,8%	5	8,2%	4	6,6%	2,5
Code	14	23,0%	6	9,8%	8	13,1%	4	6,6%	2	3,3%	2,2

Table 8.23: Re-use of specification, design and code based on numbers from 1-5

H20 is not rejected. In *Table 8.23*, the reuse was categorized from 1 (almost nothing) to 5 (very much). The mean for specification is 3,3. 3,3 can be categorized as much/very much reuse. Code had the lowest mean with 2,2 (little/some). Reuse of design is higher than code, but it is still relatively low with 2,5 (little/much). Code is reused the least, and the hypothesis is therefore not rejected.

8.8 Outsourcing

H21. There are no differences in the distribution of work between organizations that outsource much of the total IT-activity, and organizations that outsource less of the total IT-activity

			Total maintenance	Total development	Operations	Support	Application portfolio upkeep	Application portfolio evolution
Spearman	Outsourcing total	Correlation Coefficient	-,180	-,084	-,175	,220	,015	-,015
		Sig. (2-tailed)	,146	,498	,157	,074	,904	,903
		N	67	67	67	67	64	64
	outsourcing development	Correlation Coefficient	-,059	-,302*	-,026	,296*	,322**	-,321**
		Sig. (2-tailed)	,634	,013	,835	,015	,009	,010
		N	67	67	67	67	64	64
	Outsourcing maintenance	Correlation Coefficient	-,153	-,241*	-,201	,271*	,184	-,183
		Sig. (2-tailed)	,216	,049	,103	,026	,145	,147
		N	67	67	67	67	64	64
	Outsourcing operation	Correlation Coefficient	-,082	-,091	-,356**	,019	-,030	,030
		Sig. (2-tailed)	,510	,463	,003	,876	,814	,817
		N	67	67	67	67	64	64
	Outsourcing support	Correlation Coefficient	-,173	-,193	-,290*	,188	,277*	-,278*
		Sig. (2-tailed)	,160	,118	,017	,128	,027	,026
		N	67	67	67	67	64	64

**. Correlation is significant at the 0.01 level (2-tailed).

*. Correlation is significant at the 0.05 level (2-tailed).

Table 8.24: The use of outsourcing vs distribution of work

H21 is rejected. From *Table 8.24* it is possible to deduct these numbers to be significant:

- Outsourcing development is significantly correlated with total development (-,302), support (,296), and application portfolio (+-,322).
- Outsourcing maintenance was significantly correlated with total development (-,241) and support (,271).
- Outsourcing operation was significantly correlated with operations (-.356).
- Outsourcing support was significantly correlated with operations (-,290) and application portfolio (+,277 and -,278).

These numbers indicate not surprisingly that organizations who outsourced development, spent less time on this themselves. These organizations also spent more time on support.
It was expected that outsourcing of operations caused less internal work spent on operations. However, this was not the case for support. Organizations who outsourced support, had no

correlation with the in-house time spent on support. Organizations who outsourced much development, spent more time on application portfolio upkeep, and less time on evolution.

H22. The use of outsourcing is not dependent on the size of the company

			Total outsourcing	outsourcing development	Outsourcing maintenance	Outsourcing operation	Outsourcing support
Spearman's	#employees	Correlation Coefficient	-,038	,341**	,061	-,155	-,154
		Sig. (2-tailed)	,755	,004	,622	,207	,208
		N	68	68	68	68	68
	#internal end-users	Correlation Coefficient	-,041	,297*	,049	-,119	-,140
		Sig. (2-tailed)	,742	,015	,698	,343	,263
		N	66	66	66	66	66
	#external end-users	Correlation Coefficient	-,080	-,083	-,079	-,049	-,300*
		Sig. (2-tailed)	,524	,506	,530	,698	,014
		N	66	66	66	66	66

**. Correlation is significant at the 0.01 level (2-tailed).

Table 8.25: Number of employees vs outsourcing

H22 is rejected. From *Table 8.25*, under outsourcing of development, this had a significant correlation of ,341, indicating that larger organizations outsourced more of the development. When looking at numbers of end-users, larger organizations outsourced more development considering internal end-users (corr: 0,297), but development did not correlate with large numbers of external end-users. However, organizations with many external end-users outsourced less support. This may indicate that those organizations probably focus on having a business strategy where they execute the support themselves, instead of outsourcing it.

H23. There are no differences in the distribution of work in organizations that develop most of their main systems internally, through an external organization or use package solutions.

			Total maintenance	Total development	Operations	Support	Application portfolio upkeep	Application portfolio evolution
Spearman	internally by IT-department	Correlation Coefficient	,250	,400	-,119	-,425	-,286	,287
		Sig. (2-tailed)	,044	,001	,346	,000	,024	,024
		N	65	65	65	65	62	62
	external organization	Correlation Coefficient	-,006	-,230	,167	,104	,363	-,364
		Sig. (2-tailed)	,965	,066	,185	,412	,004	,004
		N	65	65	65	65	62	62
	Package solution, with major adaptations	Correlation Coefficient	-,104	-,056	,114	,033	,147	-,147
		Sig. (2-tailed)	,411	,659	,365	,793	,256	,256
		N	65	65	65	65	62	62
	Package solution, with minor adaptions	Correlation Coefficient	,144	-,073	-,039	,103	,078	-,077
		Sig. (2-tailed)	,251	,561	,758	,413	,549	,553
		N	65	65	65	65	62	62

Table 8.26: Type of system development vs distribution of work

H23 is rejected. From *Table 8.26* it is clear that organizations developing in their IT-department, spent much more time on development (correlation ,400) and on maintenance (correlation ,250). They also spent less time on support (correlation -425), and there was a correlation in application portfolio (+-286). It is normal that an organization who develops much themselves, also has the opportunity to do their own maintenance.

When development is performed externally, more time is spent on upkeep (,363)and less on evolution (-,364). There was no correlation with organizations using package solutions.

8.9 Service-oriented architecture

H24. There are no differences in the distribution of work between organizations that have deployed service-oriented architecture and organizations that have not deployed service-oriented architecture.

			Total maintenance	Total development	Operations	Support	Application portfolio upkeep	Application portfolio evolution
Spearman's	Use SOA	Correlation Coefficient	-,132	,394**	,005	-,082	-,265*	,267*
		Sig. (2-tailed)	,312	,002	,972	,531	,045	,043
		N	61	61	61	61	58	58

**. Correlation is significant at the 0.01 level (2-tailed).

*. Correlation is significant at the 0.05 level (2-tailed).

Table 8.27: Use of Service-oriented architecture vs distribution of work

H24 is rejected. *Table 8.27* indicates that organizations that used SOA, spent more time on development. They also spent less time on application portfolio upkeep and more time on evolution. It looks like organizations that used SOA had a focus on development and evolution.

H25. The use of service oriented architecture is not dependent on the size of the company.

			Use SOA	Plan SOA
Spearman's	Number of employees	Correlation Coefficient	,182	,322*
		Sig. (2-tailed)	,156	,011
		N	62	61

*. Correlation is significant at the 0.05 level (2-tailed).

**. Correlation is significant at the 0.01 level (2-tailed).

Table 8.28: Number of employees vs service-oriented architecture

H25 is partly rejected. There were no correlations between number of employees and the use of SOA, but there was a correlation between organizations that plan SOA. This may indicate that large organizations have a wish to implement SOA, but because it is harder for these organizations, not all of them succeed. However, it is easier for smaller organizations to implement it. This is why there was no correlation in use of SOA vs the number of employees, even if there may have been more larger organizations planning it.

8.10 Comparisons with previous study

H26. There are no differences between the percentage of maintenance time in our survey and what was reported in the previous survey.

Ranks

Year		N	Mean	Mean Rank	Sum of Ranks
Total maintenance	2008	61	35,14	58,93	3595,00
	2013	67	41,47	69,57	4661,00
	Total	128			

Test Statistics[a]

	Total maintenance
Mann-Whitney U	1704,000
Wilcoxon W	3595,000
Z	-1,621
Asymp. Sig. (2-tailed)	,105

a. Grouping Variable: Year

Table 8.29: Difference total maintenance between 2013 and 2008 study

H26 is not rejected. *Table 8.29* confirms that there was not a significant difference between maintenance in 2013 (mean 41,5) and in 2008 (mean 35,1).

H27. There are no differences between the breakdown of maintenance work (corrective, adaptive, enhancive and perfective) in our survey and what was reported in the last survey.

Ranks

Year		N	Mean	Mean Rank	Sum of Ranks
Corrective	2008	61	8,17	62,35	3803,50
	2013	67	9,30	66,46	4452,50
	Total	128			
Adaptive	2008	61	6,34	58,11	3544,50
	2013	67	8,90	70,32	4711,50
	Total	128			
Enhancive	2008	61	11,39	64,16	3913,50
	2013	67	12,28	64,81	4342,50
	Total	128			
Non-Functional perfective	2008	61	9,25	63,50	3873,50
	2013	67	11,19	65,41	4382,50
	Total	128			

Test Statistics[a]

	Corrective	Adaptive	Enhancive	Non-Functional perfective
Mann-Whitney U	1912,500	1653,500	2022,500	1982,500
Wilcoxon W	3803,500	3544,500	3913,500	3873,500
Z	-,628	-1,872	-,101	-,292
Asymp. Sig. (2-tailed)	,530	,061	,920	,770

a. Grouping Variable: Year

Table 8.30: Difference maintenance categories between 2013 and 2008 study

H27 is not rejected. *Table 8.30* states that there was no significant differences in types of maintenance between 2008 and 2013.

H28. There are no differences between the percentage of development time in our survey and what was reported in the last surveys.

Ranks

Year		N	Mean	Mean Rank	Ranks
Total development	2008	61	21,28	74,53	4546,50
	2013	67	13,66	55,37	3709,50
	Total	128			
Development of replacement systems	2008	61	9,73	73,80	4501,50
	2013	67	6,85	56,04	3754,50
	Total	128			
Development of new systems	2008	61	11,54	70,29	4287,50
	2013	67	6,82	59,23	3968,50
	Total	128			

Test Statistics[a]

	Total development	Development of replacement systems	Development of new systems
Mann-Whitney U	1431,500	1476,500	1690,500
Wilcoxon W	3709,500	3754,500	3968,500
Z	-2,939	-2,754	-1,716
Asymp. Sig. (2-tailed)	,003	,006	,086

a. Grouping Variable: Year

Table 8.31: Difference development between 2013 and 2008 study

H28 is rejected. *Table 8.31* shows that there was a significant difference on both total development (p: ,003) and on development of replacement systems (p: ,006). In 2008 total development had a mean of 21,3 while in 2013 this was 13,6. Development of replacement systems had in 2008 a mean of 9,7 while in 2013 this was 6,9. This shows that more time was spent on total development and development of replacement systems in 2008, compared to 2013.

H29. There are no differences between the percentage of time used on support and operation in our survey and what was reported in the last survey.

Ranks

Year		N	Mean	Mean Rank	Sum of Ranks
Operations	2008	61	23,66	67,62	4125,00
	2013	67	21,79	61,66	4131,00
	Total	128			
Support	2008	61	19,19	61,83	3771,50
	2013	67	23,07	66,93	4484,50
	Total	128			

Test Statistics[a]

	Operations	Support
Mann-Whitney U	1853,000	1880,500
Wilcoxon W	4131,000	3771,500
Z	-,910	-,779
Asymp. Sig. (2-tailed)	,363	,436

a. Grouping Variable: Year

Table 8.32: Difference support and operations between 2013 and 2008 study

H29 is not rejected. From *Table 8.32* there was no significant differences in either operations or support between 2008 and 2013.

H30. There are no differences between the distribution of work among maintenance and development in our survey and what was reported in the last surveys when disregarding other work than development and maintenance.

Ranks

Year		N	Mean	Mean Rank	Sum of Ranks
Isolated maintenance	2008	59	65,70	51,11	3015,50
	2013	64	77,70	72,04	4610,50
	Total	123			
Isolated development	2008	59	34,30	72,86	4298,50
	2013	65	22,30	53,10	3451,50
	Total	124			

Test Statistics[a]

	Isolated maintenane	Isolated development
Mann-Whitney U	1245,500	1306,500
Wilcoxon W	3015,500	3451,500
Z	-3,269	-3,072
Asymp. Sig. (2-tailed)	,001	,002

a. Grouping Variable: Year

Table 8.33: Difference isolated maintenance and development between 2013 and 2008

H30 is rejected. *Table 8.33* displays that there was a significant difference on both maintenance and development between 2008 and 2013 when only considering development and maintenance. In 2008 more time was spent on development (mean 34 vs 24) and less time on maintenance (mean 66 vs 78) than in 2013.

H31. There are no differences between the distribution of application portfolio upkeep and application portfolio evolution in our survey and what was reported in the last surveys.

Ranks

Year		N	Mean	Mean Rank	Sum of Ranks
Application portfolio upkeep	2008	59	63,00	47,74	2816,50
	2013	64	68,11	75,15	4809,50
	Total	123			
Application portfolio evolution	2008	59	37,00	68,68	4052,00
	2013	64	31,95	55,84	3574,00
	Total	123			

Test Statistics[a]

	Application portfolio upkeep	Application portfolio evolution
Mann-Whitney U	1046,500	1494,000
Wilcoxon W	2816,500	3574,000
Z	-4,262	-1,996
Asymp. Sig. (2-tailed)	,000	,046

a. Grouping Variable: Year

Table 8.34: Difference application portfolio between 2013 and 2008

H31 is rejected. From table *Table 8.34* it is possible to conduct that there was a significant difference in application portfolio between 2008 and 2013. In 2008, upkeep had a mean of 63% while in 2013 this was 68%. Evolution had a smaller difference, whereas in 2008 there was a mean of 37%, while in 2013 this was 32%. This shows that more time was spent on upkeep and less time on evolution in 2013 compared to 2008.

8.11 Replacement systems

H32. There are no differences in the share of total new systems being developed that is classified as replacement systems in our survey and what was reported in 2008 and 2003.

	N	Minimum	Maximum	Mean	Share of all systems	Std. Deviation
#replacement systems 2003	54	-	-	0,43	60 %	-
#replacement systems 2008	57	0	3	1,00	65 %	1,134
#replacement systems 2013	53	0	3	1,06	58 %	,989

Table 8.35: Number of replacement systems being developed in 2003, 2008 and 2013

H32 is partly rejected. *Table 8.35* displays that the mean of 2008 (1,00) and 2013 (1,06) was almost identical. The mean number of 2003 was 0,43, which was more than half than in the other two studies.

The percentage of replacement systems compared to total new systems have been generally stable between the studies. In 2013, 58% of systems being developed were replacement systems. In 2008 that number was 65% and in 2003 60%. This was very similar in all three investigations. H32 is therefore partly rejected.

H33. The average age of a system that is being replaced, is the same in our survey and what was reported in 2008 and 2003.

	N	Minimum	Maximum	Mean	Std. Deviation
Average age systems 2003	16	-	-	5,38	4,27
Average age systems 2008	37	-	-	6,79	3,85
Average age systems 2013	66	,50	15,00	5,92	2,92

Table 8.36: Average age of systems in the study of 2003, 2008 and 2013

H33 is rejected. As seen in *Table 8.36*, the average age of systems being replaced was getting older. From 5,4 years in 2003 to 8,3 years in 2013. The difference between 2013 and 2008 was 18% and the difference between 2013 and 2003 was 45%. The differences are in such a degree that H33 is rejected.

8.12 Public and private differences

H34. There are no differences in the amount of outsourcing between public and private organizations.

Public og private		N	Mean	Mean Rank	Sum of Ranks
Outsourcing total	Private	24	37,1	37,96	911,00
	Public	44	30,3	32,61	1435,00
	Total	68			
outsourcing development	Private	24	27,9	26,23	629,50
	Public	44	57,6	39,01	1716,50
	Total	68			
Outsourcing maintenance	Private	24	32,5	35,10	842,50
	Public	44	35,6	34,17	1503,50
	Total	68			
Outsourcing operation	Private	24	44,0	38,54	925,00
	Public	44	28,5	32,30	1421,00
	Total	68			
Outsourcing support	Private	24	12,3	32,27	774,50
	Public	44	18,2	35,72	1571,50
	Total	68			

Test Statistics[a]

	Outsourcing total	outsourcing development	Outsourcing maintenance	Outsourcing operation	Outsourcing support
Mann-Whitney U	445,000	329,500	513,500	431,000	474,500
Wilcoxon W	1435,000	629,500	1503,500	1421,000	774,500
Z	-1,070	-2,605	-,188	-1,256	-,763
Asymp. Sig. (2-tailed)	,285	,009	,851	,209	,446

a. Grouping Variable: Public og private

Table 8.37: Outsourcing divided on public and private organizations

H34 is rejected. There was a significant difference of ,009 on outsourcing when it comes to development. The mean of private outsourcing of development was 27,9, while in public this was 57,6. This is a major difference, H34 is therefore rejected.

These results are not completely similar to results from 'IT i praksis'. In that investigation private organizations outsourced more development and support, which was not the case in our study. But the distribution of maintenance and operation was very similar.

H35. There are no differences between time spent on maintenance in public and private organizations.

Ranks

Public og private		N	Mean	Mean Rank	Ranks
Total maintenance	Private	24	46,12	39,08	938,00
	Public	43	38,89	31,16	1340,00
	Total	67			
Corrective Maintenance	Private	24	9,15	35,13	843,00
	Public	43	9,33	33,37	1435,00
	Total	67			
Adaptive Maintenance	Private	24	9,01	36,25	870,00
	Public	43	8,74	32,74	1408,00
	Total	67			
Enhancive maintenance	Private	24	17,79	44,25	1062,00
	Public	43	9,14	28,28	1216,00
	Total	67			
Non-Functional perfective maintenane	Private	24	10,17	33,17	796,00
	Public	43	11,68	34,47	1482,00
	Total	67			

Test Statistics[a]

	Total maintenance	Corrective Maintenance	Adaptive Maintenance	Enhancive	Non-Functional perfective
Mann-Whitney U	394,000	489,000	462,000	270,000	496,000
Wilcoxon W	1340,000	1435,000	1408,000	1216,000	796,000
Z	-1,599	-,357	-,716	-3,249	-,263
Asymp. Sig. (2-tailed)	,110	,721	,474	,001	,792

a. Grouping Variable: Public og private

Table 8.38: Differences in time spent on maintenance between private and public sector

H35 is partly rejected. *Table 8.38* shows that there were no significant differences between public and private sector under total maintenance. However, there was a significant difference in enhancive maintenance. In that case, p was only ,001 and the difference was a mean of 17,8 in private to 9,1 in public.

This was the same results as in 'IT i praksis'. In that study there was also a significant difference (,000) in enhancive maintenance, while there was no correlation in the other categories.

H36. There are no differences between the percentage of time used on development in private and public sector.

Ranks

Public og private		N	Mean	Mean Rank	Ranks
Total development	Private	24	14,86	37,33	896,00
	Public	43	12,95	32,14	1382,00
	Total	67			
Development of replacement systems	Private	24	5,12	33,88	813,00
	Public	43	7,79	34,07	1465,00
	Total	67			
Development of new systems	Private	24	9,74	41,40	993,50
	Public	43	5,16	29,87	1284,50
	Total	67			

Test Statistics[a]

	Total development	Development of replacement systems	Development of new systems
Mann-Whitney U	436,000	513,000	338,500
Wilcoxon W	1382,000	813,000	1284,500
Z	-1,066	-,041	-2,385
Asymp. Sig. (2-tailed)	,286	,968	,017

a. Grouping Variable: Public og private

Table 8.39: Difference in time spent on development between private and public sector

H36 is partly rejected. In *Table 8.39* there were no significant differences between public and private sector when looking at total development. However, there was a significant difference between development of new systems in private (mean - 9,74) and public sector (mean - 5,16). This indicates that private organizations spend more time on development of new systems than public organizations.

This is partly similar to 'IT i praksis'. Those results had a significant difference on development of new systems (sign. 0,000), but also on total development (sign. 0,001).

H37. There are no differences between the distribution of work among maintenance and development between private and public sector when disregarding other work than development and maintenance.

Ranks

		N	Mean	Mean Rank	Sum of Ranks
Isolated maintenance	Private	23	76,74	30,11	692,50
	Public	41	78,28	33,84	1387,50
	Total	64			
Isolated development	Private	23	23,26	34,89	802,50
	Public	42	23,58	31,96	1342,50
	Total	65			

Test Statistics[a]

	Isolated maintenance	Isolated development
Mann-Whitney U	416,500	439,500
Wilcoxon W	692,500	1342,500
Z	-,781	-,605
Asymp. Sig. (2-tailed)	,435	,545

a. Grouping Variable: Public og private

Table 8.40: Difference isolated maintenance and development, private and public sector

H37 is not rejected. From *Table 8.40* there were no significant differences between isolated development and maintenance between public and private sector. Looking at the mean values, they were almost identical.

When looking at the results from 'IT i praksis', H37 is rejected. Both isolated maintenance and isolated development had a significant difference between private and public sector, both with a significance of ,018.

H38. There are no differences between the distribution of application portfolio upkeep or evolution in private and public sector.

Ranks

		N	Mean	Mean Rank	Sum of Ranks
Application portfolio upkeep	Private	23	55,73	20,13	463,00
	Public	41	74,99	39,44	1617,00
	Total	64			
Application portfolio evolution	Private	23	44,27	44,87	1032,00
	Public	41	25,01	25,56	1048,00
	Total	64			

Test Statistics[a]

	Application portfolio upkeep	Application portfolio evolution
Mann-Whitney U	187,000	187,000
Wilcoxon W	463,000	1048,000
Z	-3,989	-3,989
Asymp. Sig. (2-tailed)	,000	,000

a. Grouping Variable: Public og private

Table 8.41: Application portfolio distributed on private and public sector

H38 is rejected. *Table 8.41* supports that there was a significant difference on application portfolio upkeep and evolution between private and public sector, both having a p of zero. Private sector spent less time on upkeep than public sector, and more time on evolution. 'IT i praksis' had identical analytic results where both had a significance of zero.

H39. There are no differences between the percentage of time used for operation and support between private and public sector.

Ranks

		N	Mean	Mean Rank	Sum of Ranks
Operations	Private	24	18,13	28,31	679,50
	Public	43	23,82	37,17	1598,50
	Total	67			
Support	Private	24	20,89	28,54	685,00
	Public	43	24,34	37,05	1593,00
	Total	67			

Test Statistics[a]

	Operations	Support
Mann-Whitney U	379,500	385,000
Wilcoxon W	679,500	685,000
Z	-1,795	-1,719
Asymp. Sig. (2-tailed)	,073	,086

a. Grouping Variable: Public og private

Table 8.42: Operation and support split on public and private sector

H39 is not rejected. *Table 8.42* shows that there are no significant differences in operations or support between private and public sector.

This was not the case for the results from 'IT i praksis', where both operations and support were significantly different.

8.13 Cloud

H40. There are no differences in the distribution of work between organizations with many main systems using cloud, compared to organizations with few main systems using cloud.

			Total maintenance	Total development	Operations	Support	Application portfolio upkeep	Application portfolio evolution
Spearman's	Systems using cloud / Total systems	Correlation Coefficient	-,106	,027	,301*	,016	-,142	,143
		Sig. 2-tailed	,417	,839	,018	,905	,289	,285
		N	61	61	61	61	58	58

*. Correlation is significant at the 0.05 level (2-tailed).

**. Correlation is significant at the 0.01 level (2-tailed).

Table 8.43: Share of main systems using cloud vs distribution of work

H40 is partly rejected. *Table 8.42* presents the share of systems using cloud, correlated with distribution of work. The only significant correlation was in operations (0,301). Because there was only one category correlating, H40 is only partly rejected.

H41. There are no differences in the use of cloud between organizations having many employees and end-users, compared to organizations with few employees and end-users.

			# employees	# external end-users	# internal end-users
Spearman's	Systems using cloud / #Total systems	Correlation Coefficient	,101	,086	,159
		Sig. (2-tailed)	,435	,506	,216
		N	62	62	62

**. Correlation is significant at the 0.01 level (2-tailed).

Table 8.44: Share of main systems using cloud vs number of employees and end-users

H41 is not rejected. *Table 8.44* shows that there was no correlation between the use of cloud and number of employees or end-users. This indicates that the size of a company, and the number of users have no influence on organizations acquiring cloud systems or not.

Chapter 9

Discussion

Both in the descriptive chapter and the hypothesis-testing chapter, there has been some discussion about specific results. This chapter will contain discussion on a higher level. This involves discussing the relations between different results and statistical calculations, to find reasons for trends or significant data. Both data from this study and previous studies will be accounted for. There are many assumptions as of why the sudden significant change in application portfolio and share of maintenance and development the last five years. This could be influenced by change of IT-strategy, business strategy, population, or all of them. This chapter will first present these results, and then discuss factors and reasons for the change.

9.1 Main results

This section will present the main results for this investigation; the share of isolated maintenance, isolated development and application portfolio upkeep/evolution. In this study, the amount of maintenance, when we only look at maintenance and development, was 78%. Development was 22%. When looking at the results from *Table 9.1*, this is the highest amount of maintenance in the replication studies. There was an increase in maintenance from the first study by Lientz/Swanson in 1977. After the peak in 1998 with 73%, the trend has gone down. It was discussed that the Y2K-crisis was the reason for this shift. During the Y2K there was a lot more development being performed. However, in this study the amount of maintenance has succeeded the level of 1998 and the trend seems to have turned again.

Investigation	Isolated maintenanec	Isolated development
Lientz/Swanson (1977)*	53,0 %	47,0 %
Nosek/Palvia (1990)*	62,0 %	38,0 %
Krogstie (1993)	59,0 %	41,0 %
Holgeid (1998)	72,9 %	27,1 %
Jahr (2003)	65,9 %	34,1 %
Davidsen (2008)	65,7 %	34,3 %
Own (2013	77,8 %	22,2 %

*Not part of the internal replication study and not performed in Norway

Table 9.1: Isolated development and maintenance in all studies

If you look at 1998 as an exception from the norm, the share of isolated maintenance has always stayed around 60%. The increase has been small up to this study, and between 2003 and 2008 there was no significant difference in isolated maintenance. A conclusion about the change was therefore not suitable. But between 2008 and 2013, H30 shows that there was a significant difference in isolated maintenance.

It should be noted that the share of maintenance when also looking at other categories of work, has not changed that much (2013 - 41%; 2008 - 35%; 2003 - 36%; 1998 - 41%; 1993 - 40%). H26 showed that total maintenance was not significantly different from the 2008 investigation. But these results do not give the same clear picture of the situation as isolated maintenance.

The application portfolio was not addressed before Krogstie's study in 1993 [32]. That is why there was only data since that study. The application portfolio upkeep and evolution in the replication study are presented in *Table 9.2*. When looking at H3 and H4, there were significant differences between development and maintenance, and upkeep and evolution. These variables are not the same, but show different aspects of the distribution of work in an organization. An example is that when looking at development, it does not distinguish the effort between development of enhanced functionality and replacement systems. An organization's share of development is therefore categorized as application portfolio upkeep and evolution [32].

Investigation	Upkeep	Evolution
Krogstie (1993)	44 %	56 %
Holgeid (1998)	62 %	38 %
Jahr (2003)	61 %	39 %
Davidsen (2008)	63 %	37 %
Own (2013)	68 %	32 %

Table 9.2: Application portfolio in all studies

Considering all the different data and factors during the five studies in these 20 years, the application portfolio trend has been very consistent. 1993 was the only year evolution was higher than upkeep. Since then, upkeep has been greater than evolution in every study. Between 1998 and 2008 the trend was stable enough to not draw any conclusions of change. But between 2008 and 2013 there has been a significant increase (statistically calculated in H31).

The rest of this chapter will consist of a discussion towards why the results have turned out as they did.

9.2 Organizations and business-strategy

In all studies, the survey was sent out to IT-managers in different organizations. The reason for this was that the IT-managers have an overall knowledge of the organizations. There was also a bigger chance that they had direct access to data related to the questions. This way, they could answer based on accurate data, instead of answering with a rough estimate, or in worst case, take a guess. 92,6% of the participants were managers. This was an acceptable amount. The quality of the answers was also something that influenced the correctness of the results. The average answers where quality was asked, had an answer percentage of 72, where the answers were based on reasonably accurate, or on good data (2008 - 81%). This was also an acceptable amount, and should be sufficient for reasonably accurate answers.

The management of an organization controls the IT and business strategies. How much these strategies integrate with each other may have an impact on how the organizations work. 73,5% said that business and IT-strategy were integrated or influenced by each other. This should mean that the large amount of maintenance is happening because they think it is best for their businesses. It is not just an unfortunate event that happened because of reasons on business strategy. Some organizations may believe that upkeep pays off more than evolution. This may be the fact for a short period of time, but as presented in the background chapter, too much maintenance and extensions on a system over a long period of time, might make the code too complex to execute effectively. At some point, application portfolio evolution is necessary and will benefit the organization.

Results show that organizations with top-managers who spend most of their time fire fighting, do the least development. The organizations who had top-managers developing new business models that exploit technological opportunities performed almost five times more development than top leaders focusing on fire fighting. They also spent the least amount of time on operations.
Also, top-managers who spent more time on improvements on applications, performed the most application portfolio evolution. These results show that the manager's focus and strategy, influence the organization's distribution of work and results. The business strategy is therefore a major influence on distribution of work.

9.3 Outsourcing and consulting

There has been an increase in outsourcing between 2008 and 2013. This increase was largest in development where it went from 31% to 41% between the studies. The amount of maintenance did not have the same amount of increase, going from 31% in 2008 to 34% in 2013. Throughout the studies, there has been a gradual decrease of internal development, and vice versa externally. This may be due to the growing popularity of outsourcing. But the amount of internal development is pretty much the same between the 2008 and 2013 study, despite that outsourcing increased in the same period. Even if the amount of outsourcing has increased, and work hours on development has decreased, the fact that internal development has stayed the same may be an indication that the amount of total development has not gone that far down after all. If the outsourced work on development and internal development are combined, the gap between 2008 and 2013 is not that large. The global amount of development has therefore not necessarily decreased. This is also backed up with that throughout the investigation, there have been a similar amount of systems in production and systems under development.

Because maintenance was outsourced the same amount as in 2008, much of the internal distribution of work hours spent on development was shifted over to maintenance in this study. Organizations seek external services to develop their systems, but they want to perform the maintenance themselves. There are many advantages in not letting one part have all the responsibilities. However, the people who does the development, becomes experts on that system, and is therefore the most suited to do the maintenance. This reduces the risks of creating new errors and the maintenance work is often executed more efficiently. Outsourcing development and performance of the maintenance internally may not be the most efficient method, but since this is the practice, it may be the most economical.

Only half of the managers had top focus on development, while the rest focused on administrative tasks. This may be because of the increase of outsourcing towards development. When development is outsourced, it gives managers more time to focus on other tasks. In some cases, work on upkeep could be considered administrative tasks. This could be when the business environment changes, the portfolio needs enhancive modifications to cover all aspects of the business (adaptive maintenance).

The fact that only half of the managers have top focus on development, is another reason for why development may be at its lowest. When new projects and plans are not started, this will affect the work distribution. Employees will keep on with what they are doing, which is maintenance and application portfolio upkeep.

9.4 System portfolio

This study had organizations with the most main systems (mean of 11,6). The average age of the systems was similar to the last study, but the amount of systems older than 10 years was twice as large (12% vs 6,6%). In 1993, 51% of the systems was 0-3 years old. In 2013 this share was only 31%. This is another reason for the increase of application portfolio upkeep. The low number of new systems indicates less development, and the large number of old systems indicates that systems are kept alive and are being maintained longer. It must also be considered that decades of software experience, and the evolution of development methods and practices, may have resulted in more durable applications and system architecture. Good development results in less and easier performed maintenance, which results in longer durability. An example of this is that the reuse of specification, design and code has increased the last years.

An observation made was that application portfolio upkeep has increased since 1993, together with the number of replacement systems (1993 - 48%; 1998 - 57%; 2003 - 60%; 2008 - 64%; 2013 - 58%). However, it looks like the number of replacement systems has peaked between 2008 and 2013, which is not the case for upkeep in the same period (63% to 68%). The main reasons to replace a system were that it was hard to maintain, or integration with another system. Even so, the amount of work on maintenance was at a record high in this study. This could indicate that organizations wish to maintain a system for a longer period of time, instead of developing a new one.

9.5 Technology

In the last two decades, there have been many trends in technology. Package solutions, external development and use of Java have had an increase between 1993 and up 2008. After 2008 these numbers have halted, and some even decreased. Instead, web services and script coding have had a major increase the last ten years. This could indicate the beginning of a new trend and popularity. The IT-market has always gone in waves regarding popularity. From assembly to object oriented development, to package solutions and now maybe to web-solutions. It is a natural trend, because web-services automatically support many different devices in today's market (E.g personal computers, tablets, smart phones). Instead of making several different systems, with the use of many different technologies, it is possible to just use one.

The questions surrounding SOA were included in these studies since 2008. The use of SOA has increased from 57% in 2008 to 66% in 2013. When planning the implementation of SOA, about one third had no plan to implement SOA both in 2013 and 2008, but less had a desire to implement SOA in this study. The desire to implement SOA was reduced from 31,6% to 19,7% in 2013. This could mean that even if the use of SOA has increased the last five years, this may not increase as much in the upcoming five years until the next study. There is a big chance that it will remain the same, or even be reduced.
H24 showed a correlation that organizations using SOA, also spent a larger share of time on development. This is a positive sign for SOA, and it will be interesting to see in future investigation if change in the use of SOA, also changes the distribution of work.

In this investigation, we looked at characteristics that would be influenced with the use of cloud. Our results showed that the use of cloud did not influence these characteristics severely. One of the characteristics was that cloud reduces operation. Our survey's user opinion also stated that cloud reduced time spent on operation for them. However, our results on distribution of work, stated that organizations that used cloud, spent more time on operations. This could indicate that these organizations try to implement cloud in order to reduce time spent on operations. Another contradiction about cloud, was that many factors that cloud is supposed to influence, were "neutral". Examples of factors that were neutral were "easier to correct errors" and "easier to implement new functions". Cloud is still in an early stage, and it is exciting to see if these factors' influence has improved until the next study.

9.6 Private and public organizations

This study had a majority of organizations working in public sector (2013 - 64%; 2008 - 18%; 1993 - 13%). The reason for this was that it would give new dimensions to the investigation. Public organizations are often large (median number of employees public - 525 vs private - 82), and often cover municipal or national areas with their services. This is why both the number of employees and end-users (internal and external) in this study has increased significantly. The budget has also increased to a record high with an average of NOK 21,5 millions, but there were no differences between public and private organizations when it came to budget.

The large amount of public organizations could also influence work distribution and outsourcing. H34 showed a significant difference in outsourcing of development. Private organizations spent twice as much time on development than public organizations. In contrary, public organizations outsource as much as 58% of development, while private organizations only outsource 28%. A reason for this is that public organizations usually work in a sector that has a whole other focus than development. Another reason is that public organizations struggle with getting the specific expertise, because such employees are more attracted to private sector. Instead, public organizations have to outsource these services. A reflection of this can be seen in distribution of development. Private organizations developed a significantly larger amount of new systems, and spent more time on enhancive maintenance. Another example is that there were less system developers in this study. A main reason for this could also be the increasing amount of outsourcing and the majority of public organizations. They do not need system developers, because development is not their main priority.

As stated earlier in this section, public organizations averagely larger than private organizations. H22 displays a correlation that larger organizations outsource more than smaller organizations. This is part of influencing the results that public organizations outsource more than private.

Chapter 10

Evaluation

This chapter is a personal evaluation of the study. It will address matters toward the survey, the research method and the discussion.

10.1 Survey population

The participating organizations were selected from mailing lists provided by the university. There were two different lists, one from the members of the data society and the other one from public sector data society (OSDF). It is debatable if these lists should be filtered in advance. This to provide consistency and a more specific population. An example of this could be to only pick organizations who develop themselves and are of a certain size. This could provide more specific and in some way correct answers. It must be said that because the mailing list was provided by the Computer society, all participants had some relevance to IT. On the other side, having different types of organizations do widen the investigation field. It would also give a more general population. To use different types of organizations, was also done in all of the previous studies, making the populations and results more accurate.

There could be a larger number of participants. There were 62 fully-completed responses. This was enough to perform statistical analysis, and it was the largest amount of replies in all of the replication studies. However, the more participants, the more accurate data. When splitting replies into different categories for comparison, some of the categories had a very small number of cases. This made these results vulnerable for outliers. If there were more cases, these outliers would not influence the results as much. The collaboration with 'IT i praksis', who had 200 respondents, made the population larger. The fact that most of our results were similar to their results, indicates that the population was large enough.

10.2 Biased answers

It was important to make questions with answer alternatives sufficient for the correct results. The survey was therefore approved by researchers with a high amount of experience and knowledge toward survey investigations. After this, the survey was sent out as a pilot-test to three practitioners who were IT-managers in different organizations. After positive responses from this test-pilot, the survey was conducted and sent out to the real participants.
The questions had to be easy to understand, and there could be no room for different interpretations between the participants. An explanation of some of the terms used was put at the start of the survey, and some questions had an explanatory addition. Doing this would reduce the chances of recipients not understanding the questions, and answer something else than was asked. Participants were encouraged to make contact if they did not understand something, or had any questions.

The survey was directed to IT-managers in the IT-departments. It was feared that these managers would answer some questions too positively because of pride or denial. Examples of such questions were toward IT- and business strategy. Letting employees on a lower level answer the survey, could give more neutral answers toward these questions. However, these employees would lack experience and knowledge to answer many of the other questions correctly. To let managers take the survey was therefore considered the best choice. All organizations were interested in IT because they were subscribed to our IT-relevant mailing lists. This could have resulted in some biased questions, but it was necessary to make sure all participants were relevant to answer the questions.

When analysing the answers, it was discovered that questions used in previous surveys, which had been changed this year, often led to different results. An example is a question that earlier consisted of check boxes, this year consisted of text boxes. The participants had to write the name of methods they used in these text boxes instead of just clicking a check box. The share of methods was significantly lower in this study. It was suspected that the reason for this was if a participant did not know the name of the method, he wrote nothing at all. Previous surveys did not have this problem, because check boxes were used. It was from this learned that changing the format of a question could influence the results.

10.3 Research method

The lack of qualitative data in this study did give a lack of depth in the conclusion. Having qualitative results could make room for alternate answers and explanations. The quantitative answer alternatives to the questions could in some cases restrict the participants. Using qualitative data could give important information to improve the investigation.

To run some case studies in IT-organizations in advance of the survey, could help give advice and information of what to ask for in the survey. Examples of this could be relevant problems, or new technology that influenced the distribution of work. Instead, the survey was based on pre-study and the investigations done previous in the replication study. In the 2008 investigation, case studies were performed. The results from these studies confirmed the presumptions of the survey that year[14].

10.4 Result analysis

When comparing results with previous years, or running statistical analysis on the data, only quantitative data was used. Even if significant changes or correlations were found, the results were just numbers. The test did not say anything about what caused these findings. To draw conclusions, an assumption had to be made by looking at the rest of the results, previous results or relevant theory from other studies. In addition there were some assumptions and thoughts discussed towards the findings. A single correlation is not proof in itself. There are also occasionally "shotgun statistics issues", where doing a lot of tests, some will turn out significant by chance. This is why this study is careful with drawing too many conclusions, and instead focuses mainly on presenting the results.

Chapter 11

Conclusion and further work

The main variables in this replication study have continued their trend. Isolated maintenance had a share of 78%, which is a significant increase, and the highest share of all previous studies. This was also the case regarding application portfolio upkeep, which was 68%. Even if these numbers seemed to have halted in the last study (between 2003 to 2008), the increase continued coming in to this study. It is uncertain if upkeep will continue to rise, or if it will stay where it has been, between 60-70%.

Outsourcing of development has also increased significantly the last years. A hypothesis was therefore that development/evolution were outsourced instead of performed internally. If we add up outsourced development and internal development, the differences are not that great between 2008 and 2013. The global share of development/evolution may therefore not have decreased as much as it may seem. Maintenance is outsourced at the same amount as it was in 2008. The void in the distribution of work when an organization outsources development, is therefore filled with maintenance.

The distribution of work is reflected in the managers focus. Organizations where top-managers spend most of their time fire fighting, perform the least development. As an opposition, when top-managers spend more time on methods that exploit technological opportunities, they also spend five times more time on development. Considering the increase in the main variables, there may be a connection that only half of the managers have a top focus on development, while the rest focus on administrative tasks.

This study had, in opposition to the other studies, a majority of public organizations. This may have influenced some of the results. The study showed that there was significant differences in application portfolio evolution (mean public - 25; mean private - 45) and upkeep (mean public - 75; mean private - 56). The reason for this could be that public organizations on average outsource twice as much development than private organizations. It may look as public organizations outsource development, and perform the maintenance themselves. This is probably because public organizations usually focus on public services, and not on software development. What ever the reasons are, many of the results are different. When only considering the private organizations, the application portfolio is generally the same as in 2008.

In this study the systems were older than in previous studies. Twice as many systems were more than 10 years old. This may be a reason for the increase in maintenance and upkeep. Organizations choose to keep their systems alive with modifications for a longer period of time, instead of developing new systems.

11.1 Further work

This study contains several discoveries and assumptions that should be investigated further. An example could be to perform case studies or interviews with organizations to get a better understanding of what is going on. Gathering qualitative data, and have the opportunity to talk about discoveries from the survey, could help shed more light on the assumptions.

Further work in this investigation would be to evaluate other aspects of the data. The data collected by the survey created the source of numerous investigations. To study the differences between public and private organizations, or the size of the companies, are examples of what could be investigated further.

Examples of work beyond this study could be to follow up on the continuous trend of outsourcing. This is an interesting trend that has roots many years back, but has only been part of this replication study since 2008. How has it evolved, and how will it evolve in the future? Also, do the organizations save as much on outsourcing as expected? Does it provide the same quality of results? One of the main conclusions of this study was that outsourcing influences the distribution of work. That is why outsourcing is an important topic to investigate further.

I am also proposing to investigate more thoroughly how development models and maintenance models affect the distribution of work. The questions used in this investigation did not give the sufficient answers as expected. The evolution of different technologies that may influence the distribution of work, like the use of cloud and SOA, should also be investigated further. Cloud may still not be too integrated in organizations' processes to give too much influence on other factors, but this may change in the future. In addition to these technologies, investigation of SAP and big data could also be part of future studies.

To continue the collaboration with 'IT i praksis' is also encouraged. To compare results from another population builds support to our investigation. There are also room for a larger cooperation with them, using more of the same questions.

References

[1] April, A., Huffman Hayes, J., Abran, A., and Dumke, R. (2005). Software maintenance maturity model (SMmm) the software maintenance process model. *Journal of Software Maintenance and Evolution: Research and Practice*, 17(3):197–223.

[2] Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., and Zaharia, M. (2010). A view of cloud computing. *Commun. ACM*, 53(4):50–58.

[3] Basili, V. (1990). Viewing maintenance as reuse-oriented software development. *Software, IEEE*, 7(1):19–25.

[4] Belady, L. A. and Lehman, M. M. (1976). A model of large program development. *IBM Syst. J.*, 15(3):225–252.

[5] Bennett, K. H. and Rajlich, V. T. (2000). Software maintenance and evolution: A roadmap. ., pages 73–87.

[6] Bhatt, P., Shroff, G., and Misra, A. K. (2004). Dynamics of software maintenance. *SIGSOFT Softw. Eng. Notes*, 29(5).

[7] Boehm, B. W. (1988). A spiral model of software development and enhancement. *Computer*, 21(5):61–72.

[8] Brooks, A., Roper, M., Wood, M., Daly, J., and Miller, J. (2008). Replication's role in software engineering. In Shull, F., Singer, J., and Sjøberg, D., editors, *Guide to Advanced Empirical Software Engineering*, pages 365–379. Springer London.

[9] Capers, J. (2006). The economics of software maintenance in the twenty first century. http://www.compaid.com/caiinternet/ezine/capersjones-maintenance.pdf. [Online; accessed 23-October-2013].

[10] Chapin, N. (2000a). Do we know what preventive maintenance is? In *Proceedings of the International Conference on Software Maintenance (ICSM'00)*, ICSM '00, pages 15–17. IEEE Computer Society.

[11] Chapin, N. (2000b). Software maintenance types - a fresh view. In *ICSM*, page 247.

[12] Chou, D. C. and Chou, A. Y. (2011). Innovation outsourcing: Risks and quality issues. *Computer Standards and Interfaces*, 33(3):350 – 356.

[13] Consulting, R. M. and Dataforeningen (2012). IT i praksis. In ., pages 1–132. Ramboll Management Consulting AS.

[14] Davidsen, M. (2009). *A Study of Development and Maintenance in Norway*. Master thesis, Norwegian University of Science and Technology.

[15] Davidsen, M. K. and Krogstie, J. (2010). A longitudinal study of development and maintenance. *Inf. Softw. Technol.*, 52(7):707–719.

[16] E. Soloway, B. Adelson, K. E. (1988). Knowledge and process in the comprehension of computer programs. *The nature of expertice*, pages 129–152.

[17] Ericsson, K. A., Krampe, R. T., and Tesch-romer, C. (1993). The role of deliberate practice in the acquisition of expert performance. *Psychological Review*, pages 363–406.

[18] Fjeldstad, R. and Hamlen, W. (1983). Application program maintenance study - report to our respondents. *MD: Computer Society Press.*

[19] Foegen, T. (2010). What is scrum. https://www.wibas.com/media/filer_public/2013/09/30/wibas_whatisscrum_en.pdf. [Online; accessed 15-January-2014].

[20] Gefen, D. and Schneberger, S. (1996). The non-homogeneous maintenance periods: a case study of software modifications. In *Software Maintenance 1996, Proceedings., International Conference on*, pages 134–141. IEEE Computer Society.

[21] Glass, R. L. (1992). *Building Quality Software.* Prentice-Hall, Inc.

[22] Heeks, R. (2006). *Implementing and Managing eGovernment.* Sage Publ.

[23] Holgeid, K. K., Krogstie, J., and Sjøberg, D. I. K. (2000). A study of development and maintenance in norway: assessing the efficiency of information systems support using functional maintenance. *Information & Software Technology*, 42(10):687–700.

[24] Humphrey, W. S. and Kellner, M. I. (1989). Software process modeling: Principles of entity process models. In *Proceedings of the 11th International Conference on Software Engineering*, ICSE '89, pages 331–342, New York, NY, USA. ACM.

[25] IEEE (1999). *Standard Glossary of Software Engineering Terminology 610.12-1990*, volume 1, pages 1–84. IEEE Press.

[26] IEEE (2006). International standard - iso-iec 14764 ieee std 14764-2006 software engineering # 2013; software life cycle processes # 2013; maintenance. *ISO-IEC 14764:2006 (E) IEEE Std 14764-2006 Revision of IEEE Std 1219-1998)*, pages 01–46.

[27] Jørgensen, M. and Sjøberg, D. (2000). Learning from experience in a software maintenance environment. Submitted to IEEE Transactions on Software Engineering.

[28] Kakabadse, A. and Kakabadse, N. (2001). Outsourcing in the public services - a comparative analysis of practice, capability and impact. *Public Administration and Development*, 21(5):401–413.

[29] Kemerer, C. and Slaughter, S. (1999). An empirical approach to studying software evolution. *Software Engineering, IEEE Transactions on*, 25(4):493–509.

[30] Krogstie, J. (1995). On the distinction between functional development and functional maintenance. *Journal of Software Maintenance*, 7(6):383–403.

[31] Krogstie, J., Jahr, A., and Sjoberg, D. I. (2006). A longitudinal study of development and maintenance in norway: Report from the 2003 investigation. *Information and Software Technology*, 48(11):993 – 1005.

[32] Krogstie, J. and Sølvberg, A. (1994). Software maintenance in norway, a survey investigation. In *Proceedings of the International Conference on Software Maintenance*, pages 304–313. IEEE Computer Society.

[33] La Sorte, M. (1972). Replication as a verification technique in survey research: A paradigm. *The sociological Quarterly*, 13(2):218–227.

[34] Lewis, G. and Smith, D. (2008). Service-oriented architecture and its implications for software maintenance and evolution. In *Frontiers of Software Maintenance, 2008. FoSM 2008.*, pages 1–10.

[35] Lientz, B. P. and Swanson, E. B. (1980). *Software Maintenance Management*. Addison-Wesley Longman Publishing Co., Inc.

[36] McKeithen, K. B., Retiman, J. S., Rueter, H. H., and Hirtle, S. C. (1981). Knowledge organization and skill differences in computer programmers. *Cognitive Psychology*, 13(2):307–325.

[37] Mell, P. and Grance, T. (2011). The NIST definition of cloud computing. Technical Report 800-145, National Institute of Standards and Technology (NIST).

[38] Moløkken-Østvold, K. J. (2004). *Effort and schedule estimation of software development projects*. thesis, University of Oslo.

[39] Nosek, J. T. and Palvia, P. (1990). Software maintenance management: Changes in the last decade. *Journal of Software Maintenance*, 2(3):157–174.

[40] Oates, B. J. (2006). *Researching Information Systems and Computing*. Sage Publications Ltd.

[41] Pigoski, T. M. (1996). *Practical Software Maintenance - Best Practices for Managing Your Software Investment*. John Wiley and Sons Inc.

[42] Royce, P. and Takang, A. (1996). *Software maintenance: Concepts and Practice*. World Scientific Publishing Company.

[43] Silva, F., Suassuna, M., Franca, A., Grubb, A., Gouveia, T., Monteiro, C., and Santos, I. (2012). Replication of empirical studies in software engineering research: a systematic mapping study. *Empirical Software Engineering*, pages 1–57.

[44] Sølvberg, A. and Kung, D. C. (1993). *Information Systems Engineering: An Introduction*. Springer-Verlag New York, Inc., New York, NY, USA.

[45] Sommerville, I. (1996). Software process models. *ACM Comput. Surv.*, 28(1):269–271.

[46] Swanson, E. B. (1976). The dimensions of maintenance. In *Proceedings of the 2Nd International Conference on Software Engineering*, ICSE '76, pages 492–497. IEEE Computer Society Press.

[47] Welke, R., Hirschheim, R., and Schwarz, A. (2011). Service-oriented architecture maturity. *Computer*, 44(2nd):61–67.

[48] Wonnacott, T. H. and Wonnacott, R. J. (1990). *Introductory Statistics, Fifth Edition*. Wiley.

[49] Zelkowitz, M. V. (1978). Perspectives in software engineering. *ACM Comput. Surv.*, 10(2):197–216.

Appendix A

The questionnaire

This appendix contains the questionnaire used in the survey executed in 2013. It is in Norwegian, because all the recipients were Norwegian.

1. Veiledning for utfylling

***1. Virksomhetens navn:**

***2. Ditt navn:**

3. E-post (for kontakt i forhold til eventuell etterfølgende betaling)

Informasjon og veiledning

Spørreskjemaet vil enklest kunne besvares av virksomhetens øverste IT-ansvarlige eller en som innehar tilsvarende stilling i virksomheten. Svarene skal være basert på de rutiner og den praksis som virkomheten har i dag.

Spørsmål merket med * er obligatorisk. Relevansen til noen av spørsmålene vil være avhengig av svar på tidligere spørsmål. Hvis enkelte spørsmål ikke er relevante, fyll ut med antall -1 eller en blank. Alle svar blir anonymisert, og det vil ikke offenliggjøres resultater fra undersøkelsen som kan spores tilbake til den enkelte virksomhet

De som besvarer undersøkelsen vil motta 500 kroner skattefritt (skattefritt gitt at man ikke har annen inntekt i 2013 fra NTNU). For å følge opp dette vil vi i etterkant av undersøkelsen benytte e-post adressen du har lagt inn over.

Det er mulig å avbryte besvarelsen av spørreskjemaet. Når man returnerer til skjemaet fra samme maskin, vil man kunne fortsette utfyllingen

Ordforklaringer

Outsourcing
Med outsourcing menes all aktivitet relatert til IT som utføres av en tredjepart, enten i Norge eller i utlandet. Dette kan for eksempel være oppgaver knyttet til utvikling, vedlikehold eller drift av virksomhetens IT-systemer.

SOA - Tjensteorientert arkitektur
Med tjenesteorientert arkitektur, menes en progamvarearkitektur hvor informasjonsressurser og funksjonalitet tilbys gjennom et standardisert grensesnitt (for ekspemel via Webservices)

2. Informasjon om deg

***4. Din stilling:**

- IT-Leder
- IT-arkitekt
- Forretningsansvarlig leder
- Prosjektleder
- Systemutvikler

***5. Formell utdannelse:**

***6. Antall års IT-erfaring:**

3. Informasjon om virksomheten

*7. Type virksomhet:

- ○ Telekommunikasjon og data
- ○ Bank og forsikring
- ○ Offentlig forvaltning (stat/kommune)
- ○ Helsevesen
- ○ Reiseliv og transport
- ○ Varehandel
- ○ Industri
- ○ Tjenesteyting/konsulentvirksomhet
- ○ Bygg og anlegg
- ○ Annet (Spesifiser)

8. Hvilken av følgende påstander beskriver best øverste IT-ansvarliges rolle i virksomheten i dag? (et kryss)

	Nå
Øverste IT-ansvarlig bruker mest tid på å være proaktiv overfor virksomhetsledelsen med nye ideer og initiativer for å endre prosesser og applikasjoner	○
Øverste IT-ansvarlig bruker mest tid på samarbeid med virksomhetsledelsen om forbedringer av applikasjoner	○
Øverste IT-ansvarlig bruker mest tid på å utvikle nye virksomhetsmodeller som utnytter teknologiske muligheter	○
Øverste IT-ansvarlig bruker mest tid på å sikre at nye prosjekter leveres til avtalt tid, budsjett og kvalitet	○
Øverste IT-ansvarlig bruker mest tid på å sikre kostnadseffektiv leveranse av basis IT-tjenester (f.eks. kontorstøtte, e-post, IKT-infrastruktur som PC'er, nettverk, print osv.)	○
Øverste IT-ansvarlig bruker mest tid på brannslukking og daglig drift	○
Vet ikke	○

*9. Hvor tett er virksomhetsstrategi og IT-strategi integrert ?

- ○ Virksomhets- og IT-strategien påvirker ikke hverandre
- ○ Virksomhetsstrategien utvikles først, og brukes til å gi retningslinjer for IT-strategien
- ○ Virksomhets- og IT-strategien er tett integrert og påvirker hverandre
- ○ Vi har ingen egen IT-strategi - IT er fullstendig integrert i virksomhetsstrategien
- ○ Vet Ikke

*10. Hvor mange ansatte har virksomheten totalt ?

***11. Hva er det årlige budsjettet for IT-avdelingen inklusive maskinvare, programvare, personell og outsourcing? (oppgitt i millioner kroner, og uten avskrivninger)**

- ◯ mer en 50 millioner
- ◯ mellom 40 og 50 millioner
- ◯ mellom 30 og 40 millioner
- ◯ mellom 20 og 30 millioner
- ◯ mellom 10 og 20 millioner
- ◯ mellom 1 og 10 millioner
- ◯ mindre enn 1 million

4. IT-aktiviteter i virksomheten

***12. Hvor stor andel av følgende aktiviteter gjøres av andre virksomheter, ved outsourcing av aktiviteten?**

Av den totale IT-aktiviteten(i %)

Utvikling av nye IT-systemer (i %)

Vedlikehold/forvaltning av eksisterende IT-systemer (i %)

Drift (i %)

Brukerstøtte (i %)

***13. Svaret på spørsmålet om andel outsourcing ovenfor er:**

- ◯ Rimelig nøyaktig, basert på gode data
- ◯ Et grovt estimat, basert på minimale date
- ◯ En best mulig gjetning, ikke basert på noen data

***14. På bakgrunn av de totale utførte timeverk internt i IT-avdelingen i løpet av et år, hvor mye (i prosent, totalt 100%) brukes til:**

Rette feil i IT-systemer som er i drift

Tilpasse IT-systemer i drift til endret teknisk arkitektur

Utvikle ny funksjonalitet i IT-systemer som er i drift

Forbedre ikke-funksjonelle egenskaper (f.eks. ytelse og sikkerhet) i IT-systemer som er i drift

Utvikle nye IT-system som overlapper/erstatter eksisterende IT-systemer funksjonelt sett

Utvikle nye IT-system for å dekke nye funksjonsområder

Drift

Brukerstøtte

Annet

15. Spesifiser "Annet" i forrige spørsmål:

***16. Svaret på spørsmål om fordeling av timeverk ovenfor er:**

- ◯ Rimelig nøyaktig, basert på gode data
- ◯ Et grovt estimat, basert på minimale date
- ◯ En best mulig gjetning, ikke basert på noen data

Begrepet **vedlikehold** omfatter oppgaver under de fire første alternativene i spørsmål 14.

5. Spørsmål om IT-avdelingen

***17. Hvor mange personer er ansatt i IT-avdelingen (omregnet til fulltidsansatte)?**

***18. Hvor mange av de ansatte i IT-avdelingen er systemutviklere (omregnet til fulltidsansatte)?**

***19. Hva er fordelingen av systemutviklerne med hensyn til hvor lenge de har arbeidet i avdelingen? (antall personer, summert til svaret i spørsmål 18)**

0-1 år

1-3 år

3-6 år

6-10 år

Mer enn 10 år

***20. Hvor mange innleide konsulenter innen systemutvikling og vedlikehold har IT-avdelingen i gjennomsnitt over et år (omregnet til fulltidsansatte)?**

6. Virksomhetens applikasjonsportefølje

***21. Hvor mange større IT-systemer (hovedsystemer) er i drift i virksomheten?**

***22. Hvor mange sluttbrukere innenfor virksomheten har hovedsystemene?**

***23. Svaret ovenfor er:**

- ○ Rimelig nøyaktig, basert på gode data
- ○ Et grovt estimat, basert på minimale data
- ○ En best mulig gjetning, ikke basert på noen data

***24. Hvor mange sluttbrukere utenfor virksomheten har hovedsystemene?**

***25. Svaret ovenfor er:**

- ○ Rimelig nøyaktig, basert på gode data
- ○ Et grovt estimat, basert på minimale data
- ○ En best mulig gjetning, ikke basert på noen data

***26. Hva er aldersfordelingen til eksisterende hovedsystemer regnet i år etter første installasjon? (Antall systemer skal summere til svar i spørsmål 21)**

0-1 år

1-3 år

3-6 år

6-10 år

Mer enn 10 år

***27. Hvordan er de ulike hovedsystemene utviklet? (antall systemer, skal summere til svar i spørsmål 21)**

Utviklet av IT-avdelingen internt

Utviklet i brukeravdelingen i virksomheten

Utviklet av et eksternt selskap

Pakkeløsning, med store interne tilpasninger (f.eks. ERP)

Pakkeløsning, med små interne tilpasninger

Løsning som bruker Web services/komponenter utviklet eksternt

7. Teknologibruk

***28. Hvilke programmeringsspråk er i bruk? (Angi antall hovedsystemer pr språk. Flere enn ett språk kan være i bruk per hovedsystem, ta med alle)**

COBOL	
C	
C++	
C#	
Java	
Scriptspråk (PHP, Perl osv.)	
4 GL språk	
Andre	

29. Spesifiser "Andre" i forrige spørsmål:

***30. I hvor stor grad brukes en tjeneste-orientert arkitektur (SOA) for dagens hovedsystemer?**

	Ingen grad				Stor grad	Vet ikke
I hvilken grad?	○	○	○	○	○	○

***31. Baserer noen av hovedsystemene seg på cloud-teknologi?**

- ○ Ja
- ○ Nei, men er i gang med innføring av dette
- ○ Nei, har heller ingen konkret plan om å innføre dette

***32. Hvor mange av hovedsystemene baserer seg på cloud-teknologi ?**

Antall systemer

***33. Dersom virksomheten har implementert hovedsystem basert på cloud, hvordan har det påvirket disse systemene? (Du kan velge "vet ikke" dersom virksomheten ikke har cloud i hovedsystem)**

	I negativ grad	I liten grad	Nøytralt	I noe grad	I stor grad	Vet ikke
Reduksjon av hardware-utgifter	○	○	○	○	○	○
Reduksjon av software-utgifter	○	○	○	○	○	○
Enklere å legge til nye funksjoner	○	○	○	○	○	○
Enklere å rette mindre systemfeil	○	○	○	○	○	○
Enklere å rette større systemfeil	○	○	○	○	○	○
Mer fornøyde brukere	○	○	○	○	○	○
Enklere med generell vedlikehold	○	○	○	○	○	○
Reduksjon av brukerstøtte	○	○	○	○	○	○
Reduksjon av utgifter til drift	○	○	○	○	○	○

8. Utvikling av nye systemer

***34. Har virksomheten en plan for bruk av SOA i fremtidige IT-systemer?**

- ◯ Ingen plan om å innføre SOA
- ◯ Et ønske om å innføre SOA
- ◯ En klar plan om å innføre SOA
- ◯ Allerede i gang med bruk av SOA

***35. Hvor mange nye IT-systemer er for tiden under utvikling?**

***36. Av totalt antall nye IT-systemer under utvikling, hvor mange av disse er "erstatningssystemer"? (Systemer som hovedsaklig dekker funksjonalitet som alt er dekket i eksisterende systemer) (dersom ingen nye system er under utvikling,skriv -1)**

***37. Hva er aldersfordelingen på de IT-systemene som eventuelt erstattes? (Antall systemer skal summere til svar på spørsmål 36)**

0-1 år	
1-3 år	
3-6 år	
6-10 år	
Mer enn 10 år	

38. Ved utvikling av erstatningssystemer, hva er de viktigste grunnene for at de eksisterende systemene blir erstattet (gi score fra 1-5 på alle punktene nedenfor)

	1 (ikke viktig/relevant)	2 (lite viktig)	3 (noe viktig)	4 (viktig)	5 (svært viktig)
Svært vanskelig å vedlikeholde eksisterende system	◯	◯	◯	◯	◯
Svært vanskelig å drifte eksisterende system	◯	◯	◯	◯	◯
Svært vanskelig å bruke eksisterende system	◯	◯	◯	◯	◯
Finnes alternativ pakkeløsning	◯	◯	◯	◯	◯
Finnes alternativ applikasjonsgenerator	◯	◯	◯	◯	◯
Overgang til service orientert arkitektur (SOA)	◯	◯	◯	◯	◯
Overgang til ny teknisk arkitektur (ikke SOA)	◯	◯	◯	◯	◯
Standardisering med resten av organisasjonen	◯	◯	◯	◯	◯
Integrering med andre nye eller eksisterende systemer	◯	◯	◯	◯	◯
Annet	◯	◯	◯	◯	◯

***39. Ved utvikling av erstatningssystemer i hvor stor grad er man i stand til å gjenbruke kode, spesifikasjoner og design fra IT-systemene som blir erstattet?**

	Svært lite				Svært mye	Utvikler ikke selv
Spesifikasjoner	◯	◯	◯	◯	◯	◯
Design	◯	◯	◯	◯	◯	◯
Kode	◯	◯	◯	◯	◯	◯

9. Metoder og verktøy ved utvikling og vedlikehold

40. I hvilke deler av livssyklusen til IT-systemene anvendes en på forhånd definert metode? (Skriv inn metoden(e) og verktøy som støtter denne i tekstboksene. Dersom ingen metode, la stå tom)

Planlegging	
Analyse	
Kravspesifikasjon	
Design	
Implementasjon	
Testing	
Utrulling	
Drift	
Vedlikehold	
Prosjektledelse	
Programmledelse	
Gevinstrealisering	

41. Hvilke rutiner er etablert for forvaltning og vedlikehold av IT-systemer?

- [ ] Alle brukerkrav som kommer inn blir dokumentert
- [ ] Endringsforslag blir klassifisert etter type og viktighet
- [ ] Alle endringsforslag gjennomgår en konsekvensanalyse og kostnadsestimering
- [ ] Alle endringer av programvaren blir dokumentert
- [ ] Alle endringer av IT-systemet blir testet før systemet settes i produksjon
- [ ] Med unntak av driftstruende feil blir alle endringer samlet opp for periodisk implementasjon
- [ ] Ved akseptansetest av endringer, sjekkes også at den tilliggende dokumentasjon er oppdatert
- [ ] Brukere som etterspør endringer får beskjed både hvis endringsforslaget gjennomføres eller underkjennes
- [ ] Man bruker samme rutiner for endringsforslag som kommer fra IT-avdelingen som for endringsforslag som kommer fra brukergrupper
- [ ] Det gjennomføres en formell gjennomgang av systemet periodisk
- [ ] Økonomiske utstyrskostnader som er forbundet med drift og vedlikehold av IT-systemet belastes brukergruppene
- [ ] Personellkostnader forbundet med drift og vedlikehold av IT-systemet belastes brukergruppene

Kommentar:

Appendix B

Survey mail

This appendix contain the mail sent out to the recipients, inviting them to answer the survey.

Til IT-ansvarlig i Pinta as

Ved NTNU gjennomfører vi en undersøkelse blant private og offentlige norske virksomheter rundt utvikling og vedlikehold av virksomhetens egne IT-systemer.

Undersøkelsen følger opp tilsvarende undersøkelser foretatt i 1993,1998, 2003 og 2008. De som besvarer undersøkelsen vil få en utfyllende rapport med resultater både fra årets undersøkelse og langsiktige trender innen området

Da spørreskjemaet er relativt omfattende (anslått tid for utfylling 30-45 minutter) vil deltakelse i spørreundersøkelsen kompenseres med 500 kr (skattefritt gitt at man ikke har annen inntekt fra NTNU samme år).

Spørreskjema kan besvares på nettside:
https://www.surveymonkey.com/s.aspx?sm=VE1WWsubUWoxgzWTdF3l9w_3d_3d

Svarfrist for undersøkelsen er 01/12-2013.

En papirutgave av spørreskjemaet kan lastes ned fra denne siden, og kan brukes for å skaffe seg oversikt over det det spørres etter før man fyller inn skjemaet på nett.

Tor Kristian Veld, hovedfagsstudent ved Institutt for datateknikk og informasjonsvitenskap ved NTNU i Trondheim følger opp den praktiske gjennomføringen av undersøkelsen. Spørsmål angående selve spørreskjemaet, eller generelt om undersøkelsen kan rettes til Tor Kristian på e-post: torkristianveld@gmail.com

Med vennlig hilsen
John Krogstie
krogstie@idi.ntnu.no
Professor NTNU
Leder av Offentlig Sektors Dataforum

Dersom du ikke ønsker flere e-poster angående spørreundersøkelsen, klikk her: https://www.surveymonkey.com/optout.aspx?sm=VE1WWsubUWoxgzWTdF3l9w_3d_3d

Figure B.1: An example of a mail invitation to the survey