Software Organizations and Test Process Development

Jussi Kasurinen
Software Engineering Laboratory
Lappeenranta University of Technology
Lappeenranta, Finland
jussi.kasurinen@lut.fi

ABSTRACT
In this study, the components important for testing work and organisational test process are identified and analysed. This work focuses on the testing activities in real-life software organisations, identifying the important test process components, observing testing work in practice, and analysing how the organisational test process could be developed.

Software professionals from 14 different software organisations were interviewed in several phases to collect data on organisational test process and testing-related factors. Moreover, additional data on organisational aspects was collected with a survey conducted on 31 organisations. This data was further analysed with the Grounded Theory method to identify the important test process components, and to observe how real-life test organisations develop their testing activities.

The test management at the project level is an important factor; the organisations do have sufficient test resources available, but they are not necessarily applied efficiently. In addition, organisations in general are reactive; they develop their process mainly to correct problems, not to enhance their efficiency or output quality. The results of this study allows organisations to have a better understanding of the test processes, and develop towards better practices and a culture of preventing problems, not reacting to them.

Keywords: organisational test process, test process components, test process improvement, test strategy

1. INTRODUCTION
The software testing process is one of the core processes in software development, as every successful software product is tested in one way or another. However, the testing process often has to operate on limited resources in terms of time, personnel or money (Slaughter et al. 1998). To compensate for lack of resources, the test process can be adjusted to cater to the limitations set by the operating ecosystem; in fact, there are studies which conclude that adequate testing can be achieved with low amount of resources, even as low as 15 percent of the requested resources (Petschenik 1985, Huang and Boehm 2006). On the other hand, it is also plausible to say that software testing can become expensive and wasteful if it is done without any preceding planning. A comprehensive set of the test cases including all possible scenarios and outcomes simply cannot be done when software complexity starts rising (Myers 2004). Finally, there is room for developing test process, if only to steer the testing practices towards better efficiency and effectiveness (Bertolino 2007). Observing the software testing from the viewpoint of loss of investment, it is easy to understand why organisations should pay attention to testing activities. In United States alone, the lack of
resources and poor infrastructure in testing has been estimated to cause 21.2 billion dollars worth of losses to the software developers. Combined with the losses caused to the clients and customers, this estimate rises to 59.5 billion dollars, from which 22.2 could be saved by making reasonable investments on software testing (Tassey 2002).

The incentive to develop software testing and software quality has been addressed in the development of software industry standards. The new standards, ISO/IEC 29119 (ISO/IEC 2010) for software testing and ISO/IEC 25010 (ISO/IEC 2009) for quality define the testing processes and software quality characteristics. The ISO/IEC 29119 introduces three levels of testing activities; organisational process, divided to test policy and test strategy, test management process and testing work itself, consisting static and dynamic test processes. In this study, our research is focused on testing from the organisational viewpoint. This study aims to answer to a research problem ―what components affect the software testing strategy and how should they be addressed in the development of test process‖. This problem is approached from several viewpoints; how do different testing-related components affect the company test process, how can the components defined in the test strategy be used in the development of test process and finally, what concepts should the company address in process development. Additionally, this work also discusses the state of testing in software-producing organisations and possible application of ISO/IEC 29119 testing standard to the benefit of actual testing processes in different types of organizations.

To study such a large topic, both quantitative and qualitative methods were applied, allowing the empirical results to be triangulated for improved validity of the results. Our selection of observed level in organisations was in organisational units (OUs) as described in ISO/IEC 15504 (ISO/IEC 2002) to enable us to compare different sizes and types of software companies and make observations on their test processes as a whole. Overall, the high abstraction level constructs were used because using detailed level constructs might have led to too complicated description of the software development process and testing strategies. According to the results of the preliminary studies and existing models such as TMMi2 (TMMi 2010) or ISTQ-B (ISTQ-B 2007), the affecting factors and their relationships were analysed from the viewpoint of test process improvement and testing strategy development. Describing the practice of software testing at a high abstraction level was important because, for example, comparing methods, tools and techniques of software testing has a high contextual relevance, and direct comparison between different types of organisations is not feasible approach for scientific, unbiased and universal observation and measurement.

This chapter includes results from several of our previous research publications, which have discussed the test process and test organization from several viewpoints, such as problems and enhancement proposals (Kasurinen et al. 2009a), test resources and test automation (Kasurinen et al. 2009b), the role of development method (Kettunen et al. 2010), test case development (Kasurinen et al. 2010), test process development (Kasurinen et al. 2011b), end-product quality (Kasurinen et al. 2011c) and enabling organizations to assess their test process independently (Kasurinen et al. 2011a). In this chapter we aim to introduce the main results of these works and discuss the combined observations and implications on the development of test organizations.

This study contains six sections. Section 2 introduces software testing, viewpoints of the study and the applied testing-related standards. Section 3 describes the research problem and subject, the selection of the research methods, and the research process. In Section 4, the previous results and studies discussing the sub-problems are summarised. Section 5 combines the results of the different sub-problems and Section 6 discusses the limitations of the applicability of the results. Finally, Section 7 concludes this chapter with summary over the main contributions.
2. SOFTWARE TESTING IN GENERAL

The definition of software test process was adopted from the draft of the international standard ISO/IEC 29119 Software Testing Standard (ISO/IEC 2010). According to the standard, software testing consists of three different layers, all of which contribute to the software test process. By researching test processes, the answer was sought to three questions: Which components affect the software testing in practice, what are the important factors from the viewpoint of the test strategy, and how should they be addressed in the development of the test process? In general, what affects the strategy and what concerns should the strategy address.

2.1 CONCEPTS OF SOFTWARE TESTING

Test process can be evaluated from several different perspectives, as the process is a compilation of different components and factors, combining technical infrastructure and human interactions to a larger sociotechnical (Geels 2004) phenomenon. Test process improvement and testing strategy development were selected as the viewpoints according to the results of the preliminary studies and literature review. This selection was made so as to observe the existing testing process practices from the point of view of software designers, project managers and software testers. This selection enabled us to concentrate research resources on the issues that respondents evaluated as important, and observe the entire testing process, rather than focus on individual mechanisms or process phase activities.

The literature contains many definitions of software testing. In the joint ISO/IEC and IEEE standard, a glossary of software engineering terminology, ISO/IEC/IEEE 24765-2010 (ISO/IEC/IEEE 2010), testing is defined as:

- activity in which a system or component is executed under specified conditions, the results are observed or recorded, and an evaluation is made of some aspect of the system or component. IEEE Std 829-2008 IEEE Standard for Software and System Test Documentation.3.1.46 (IEEE 2008).

The preparation actions, actual testing work and test reporting done in a software project formulates a test process. For example, in ISTQB Glossary (ISTQ-B 2007) of used terms used in software engineering, the software process is defined as follows:

- **test process**: The fundamental test process comprises test planning and control, test analysis and design, test implementation and execution, evaluating exit criteria and reporting, and test closure activities.

Further, the working draft of the ISO/IEC 29119 standard (ISO/IEC 2010) specifies three layers of testing process, dividing the process of conducting testing to following components:

1. **Organisational test process**, including test policy and test strategy
2. **Testing management processes**, including test planning, test monitoring and control and test completion.
3. **Fundamental test processes**, are further divided into static test processes, which constitute universal activities done with all test cases such as test reporting or case design, and dynamic test processes, which constitute changing activities, such as configuring of different tools or executing a test case.
Related to these layers are the four different concepts of test process, which are defined in the ISO/IEC 29119 glossary as follows:

- **test policy**: A high level document describing the principles, approach and major objectives of the organisation regarding testing.
- **test strategy**: A high-level description of the test levels to be performed and the testing within those levels for an organisation or programme (one or more projects).
- **test management**: The planning, estimating, monitoring and control of test activities, typically carried out by a test manager.
- **test execution**:
  1. The process of running a test on the component or system under test, producing actual result(s).
  2. Processing of a test case suite by the software under test, producing an outcome <BS 7925> (BSI 1998).

Finally, in everyday terms the software testing is defined in the “Art of Software Testing, 2nd edition” by Myers (2004) as follows:

*Testing is the process of executing a program with the intent of finding errors.*

Basically, software testing should be defined in this way because it offers a broad viewpoint on software development. By defining the testing work this way, different approaches to testing tools, development models, resource availabilities and organisation models could be accounted for. However, there is an argument that this definition does not take into account the design-based shortcomings, where the product is working correctly, but the product itself is not correct.

In a traditional sense of testing, the product definition and design are architectural decisions made prior to the software test process. However, in the ISO/IEC 29119 model the verification and validation are both, parts of the test process. Validation - confirming that the software is able to fulfil the intended use (ISO/IEC/IEEE 2010) - and verification - confirming that the software complies with the given requirements (ISO/IEC/IEEE 2010) - are both related to the objectives of the test process as defined in the test policy, and exit criteria as defined in the test strategy. Based on the ISO/IEC 29119 standard, the test process should not be understood solely as the roadmap for the traditional phase of finding errors, but in a larger organisational context, including all of the development activities needed to verify and validate the item in testing.

Another important concept is the test component. In this study, the test process is observed and analysed from the perspective of the organisational process. One of the main themes of the study is to understand which test process components have influence on the practical testing work. In this work, the test process component is defined based on the principles of ISO/IEC 24765, in which one of the definitions of a component is as follows:

- 2. one of the parts that make up a system. IEEE Std 829-2008 IEEE Standard (IEEE 2008)

In the same standard, a process is defined as follows:

NOTE [ISO 9000:2005] The term "activities" covers use of resources. A process may have multiple starting points and multiple end points. The prescribed manner may be a partially ordered sequence. A process specification can be a workflow specification. An enterprise specification may define types of processes and may define process templates.

The test process components are understood as a group of concepts, which constitute all of the items of the test process, such as test personnel, test tools, test methods, test management, or other. As one of the themes of this study is to identify the important test process components, these concepts are not limited to categories such as technical or social aspects, but used as an umbrella term for every concept, item or activity that has influence on the test organisation or testing work in practice.

2.2 TESTING RESEARCH IN GENERAL

In software development, the basic objectives of software process are to produce software, which fulfils the required functionalities, has acceptable quality, is completed within budget, and released in time (Kaner et al. 1999). These attributes are all important to the software end-product, as if any of these four - functionality, quality, money and timing - is handled poorly the software is more likely to fail economically. However, in the real world the software development is usually a tradeoff between these four project attributes (Kaner et al. 1999). From this standpoint, it is not very surprising that the testing research is used to develop practices towards better coverage of testing to find more errors, or make the testing work cheaper and quicker while maintaining the pre-existing quality.

Bertolino (2007) lists four desired objectives for the software testing research to pursue: efficiency-maximized test engineering, 100% automatic testing, test-based modelling, and universal test theory. The efficiency-maximised test engineering would mean that the test process could be run on maximum efficiency and effectiveness with the help of smart tools and efficiency-optimised testing methods to ensure good quality (Harrold 2000). The second desired objective, the fully automated testing, aims to build an advanced test automation system, which would be able to do complete autonomous testing work. However, this objective is unlikely to be achieved as even with high degree of automation the system would still need human interaction to confirm results or at least configure and maintain the system (Bach 1997). The third vision of test-based modelling aims at developing software systems towards modelling practices which allow easier and more comprehensive support for testability. The difference between test-based modelling and model-based testing (for example Utting and Legeard 2007) is in the premises; model-based testing tests the software using the model, test-based modelling builds the models based on testability. The last requisite of the universal test theory aims at developing a comprehensive, coherent, and rigorous framework for assessing and comparing the strengths and weaknesses of different testing approaches. The desired objectives of Bertolino may not be very realistic to achieve in the short term, but they all aim at one objective, making testing easier.

The impact of software engineering research in software configuration management is discussed in an article by Estublier et al. (2005). In this discipline of software engineering, the impact of academic studies has been studied in relation to the software industry. Based on the results, it seems that software engineering research and software industry have close relationship; the fundamental systems and new concepts stem from the academia, while industry affects to the development of new technologies. However, as observed by the Estublier et al., the industry may sometimes take several years, even decades to adopt and fully implement the studied concepts. Against this background the current state-of-the-art software engineering and testing research may be still a completely new concept for a real-life software organisation. In fact, in a study by Juristo et al. (2004) it was concluded that even though testing techniques have been studied for over 25 years, there are still several areas that should be examined in more details. Their conclusion is that the testing
technique knowledge is still limited, and that over half of the existing studies are based on impressions and perceptions, not on formal foundations, that would allow replicable results. Bertolino (2007) concludes that one way to create the foundation for building test theory is to produce an empirical body of knowledge to understand which factors can explain where the problems arise.

Juristo and Moreno (2001) discuss the empirical software engineering. The application of knowledge in software engineering discipline is not as straightforward as in other fields. For example, by applying one method to one type of project the results may vary greatly. In software engineering, the basis of acquiring knowledge is iterative; the hypothesis are founded on existing knowledge, but during the observations and data collection in the real world the original hypothesis changes. This process is defined in three steps by Pfleeger (1999):

- Reach an initial understanding, including identifying likely variables, capturing magnitudes of problems and variables, documenting behaviours and generating theories to explain perceived behaviours.
- Test theories by matching theory with practice, eliminate variables and identify new ones, determine the relative importance of variables, and identify the range of variable values and probabilities.
- Reduce uncertainty, but not always by determining cause and effect.

The important part is to continually question and improve the theory until it explains the observed phenomenon. Some of the measurements may be fuzzy or inaccurate, and some theories may only explain the phenomenon partially. However, it is better to have a partial understanding that can serve as a basis for future theory than to discard a result simply because it does not explain or cover everything (Pfleeger 1999).

2.3 Testing as Defined in the ISO/IEC 29119 Test Standard

In the working draft of the upcoming ISO/IEC 29119 Test standards, the test process and the testing work is considered to include both the activities to validate and verify the item being tested. In the standard, the test process also encompasses the entire organisation, beginning from the upper management, policies, and quality requirements. The organisational policies and strategy steer the testing work at the project level, where the project level management creates test plans, and monitors and controls the testing activities at the fundamental level. Based on the fundamental level results, a test completion report is created and used along with the feedback to develop testing at both the project and organisational level. The process model is illustrated in more detail in Figure 1.
The ISO/IEC 29119 testing standard defines the testing process in four documents, which define how the organisation and individual projects should perform testing. These documents are test policy, test strategy, test plan and test completion report. Test policy and test strategy are organisational level documents; the organisational management defines these documents to steer the test process in the project level. At the project level, project management is responsible for defining the test plans for the project based on the organisational level policies and strategies. Project-level management is also responsible for reporting feedback from the project to the organisational management by compiling test completion reports, which then are assessed and form a basis for the organisational test process improvement. The process produces four documents; test policy, test strategy, test plan and test completion reports.

Test policy is a short, two-paged high-abstract level document, which defines the scope, principles and rules for testing to which the organisation should adhere. The main concept is that the test policy defines what is accomplished with testing, leaving the actual implementation for the other documents. The test policy defines and includes items such as objectives of testing, test process, test organization structure, required tester education, test ethics, applied standards and test asset reuse strategies. In addition to the test policy, the organisational management also defines test strategy, which is a more specific and detailed document describing how test activities should be done in the projects. The test strategy addresses items such as generic risk management, entry and exit criteria for testing, test documentation strategy, applied test phases, test techniques, test types and test selection and prioritization methods amongst other organization-level guidelines. Overall, the level of details in the test strategy is more refined than in test policy, and may include clear indicators and thresholds to steer test process at the project-level.
At the project level, the test policy and strategy are applied as a foundation, when a new test process is being defined. At the project-level, the management defines a third test process definition, a test plan, based on the principles and criterion set by the organisational documents. The test plan further elaborates on the same topics as strategy, and includes items such as test items which are to be tested, test scope, staffing, schedule for test work and cross-references to identified project risks. In addition to designing test plan, the project-level management is also responsible for providing feedback to the organisation from the completed test processes. In the standard model, this requirement is fulfilled by compiling a test completion report. This report summarises the testing work done during the project, and lists the deviations, collected metrics, lessons learned, reusable assets and change recommendations based on the test process.

Besides documentation, the standard process model is layered into three levels, which are (1) organisational test processes, (2) test management processes in project-level and (3) fundamental level, which constitutes (a) static and (b) dynamic test processes. In these layers, the testing activities are further divided into sub processes, which define the different activities happening in the layers. These processes are as follows (ISO/IEC 2010):

**Organisational test process (OTP)** is used to develop and manage organisational test specifications, such as test policy and test strategy. It is also responsible for monitoring and controlling that testing activities use the organisational level specification.

**Test management processes (TMP)** are the project-level management activities in the test process. TMP defines the test planning, test monitoring and control and test completion. They are also responsible for updating the test plan at the project-level.

**Test planning process (TPP)** is the process which is responsible for developing the test plan. Depending on the project phase, this may be a project test plan, or a test plan for a specific phase such as system testing or acceptance testing.

**Test monitoring and control process (TMCP)** ensures that the testing is performed in line with the test plan and organisational test documents. It is also responsible for identifying updates necessary for the test plan.

**Test completion process (TCP)** is a process that includes activities, which are done when testing is completed. It ensures that useful test assets are made available for later use.

**Static test processes (STP)** describes how static testing activities, such as test preparation, test result reviews and analysis and test follow-up are done. These activities are the “general” activities, which are done to all test cases in all test phases of the project, such as reserving test resources, reviewing the test results and seeing through that necessary follow-up actions are done based on results.

**Dynamic test processes (DTP)** describe how dynamic test activities such as test implementation, test execution, test environment set-up, and test incident reporting are done in the organisation. These activities are the “practical” activities, which vary between different types of testing, including configuring test tools, deciding test conditions based on test documents and practical tasks of preparing test cases and test sets.

In the ISO/IEC 29119 standard, some of these processes, such as STP or DTP, are also divided into smaller sub-categories within these definitions. This does not affect the overall meaning of the process, but rather further illustrates and explains the purposes of the activities they represent. Some processes, such as TMP, are also the owners of the other processes of the standard. The relationships between the model processes are illustrated in the Figure 2.
2.4.Viewpoints into the testing work
The effects of different test process components and test process development were selected to be the view points of this study in order to understand how test strategy can be defined and where the organisations should focus their process improvement effort. The objective was to study how different test process components affect the practical testing work, and how the test organisations could be developed based on the principles and practices presented in the ISO/IEC 29119 test process standard. In the following segment, some of the most interesting test process components with possible influence on the test process activities are discussed, followed by a segment briefly introducing the test process improvement and its concepts in practice.

2.4.1. Test process components
In software testing, the test strategy encompasses several components, which have a direct effect on the testing activities. The test strategy should define following answers in all testing-related software production phases: “What should be tested in different phases, by whom, by which method, by what tools, in what kind of environment”. The test strategy is the core of the test process; it defines the test process concepts by setting an overall framework for testing: the objectives and defining methods and resources available to the test work in the lower layers of the model. The strategy is a high-level document, which has a large influence on several test process components, as illustrated in the Figure 3. In Figure 3, the different components which are identified by different test certificates (TMMi2 2010, ISTQ-B 2010) and the upcoming ISO/IEC 29119 standard are collected and loosely categorised into five categories. The sixth category “Possible areas of interest” is then taken from the concepts suggested by the other sources, such as existing research literature and research results from our earlier research project ANTI. The figure also divides the components into the main viewpoints; on the right hand side, the components of interest, which define the organisational test strategy are listed, while the left hand side constitutes the different levels of test process activities which constitutes the organisational test process.
In the test strategy, the organisation defines several components for the test process, which all affect the testing work, such as testing tools, available time and testing personnel. It has been established, that the lack of investment in the testing infrastructure causes losses worth several billion dollars (Tassey 2002), but the studies also indicate that improving the testing infrastructure is not cheap or easy to implement. In a study by Ng et al. (2004), the most common barrier on adoption of new testing tools were considered to be the costs associated with the adoption process, the time consumption the adoption process takes and the difficulty of adopting new tools. Similarly, on adoption of testing methodologies, lack of expertise was considered the most important reason preventing the adoption of new test methodologies, and providing training was seen as too costly and time-consuming to allow investment in a real software-producing organisation.

In the traditional software development models such as the waterfall model (for example Pfleeger and Atlee 2006), the testing work usually follows the main development phase of the software. In this approach, the testing phase should not include changes to the design or requirements, but in reality, the software may still undergo changes, especially if the customer has influence on the development (Highsmith and Cockburn 2001). To address this issue, a new trend regarding the software development approach, agile development, has been defined (Abrahamsson et al. 2002). In a publication by Abrahamsson et al. (2002), agile methods are described as an attempt to answer to the business community asking for lighter-weight and more adaptable software development process. The agile models differs from traditional, plan-driven, development models by promoting communication between stakeholders and production of working releases instead of excess documentation and design before implementation (Fowler and Highsmith 2001). In comparison between plan-driven approaches and agile methods, the main difference can be characterised for example by a model for agility in the development (Boehm and Turner 2003). This model is illustrated as Figure 4.
Software testing aims to improve the quality of a software product, and in fact is a major component on deciding if the software project is profitable (Huang and Boehm 2006). However, in the measurement of quality, the definition of quality can be troublesome, as the concept of quality is closely related to a number of subjective observations. For example, Garvin (1984) has discussed the definitions of quality and made extensive definition work for establishing what the quality actually is and how it affects product concepts such as profitability or market situation. Garvin defines five different definitions for quality; transcendent, product-based, user-based, manufacturing-based and value-based definition. Even though they define the same phenomena, product quality, they vary greatly. For example, transcendent quality is “innate excellence”, which is absolute and uncompromising standard for high achievement, certainly identified if present. On the other hand, user-based quality is the more common “satisfies user needs”-definition, whereas manufacturing-based definition promotes conformance to the product requirements. Garvin also discusses the different definitions by mentioning that it also explains why different people seem to have differing opinions as to what constitutes quality; they tend to apply the definition they are most familiar with.

The different aspects and definitions of quality also mean that the measurement of software quality has some considerations. A paper by Jørgensen (1999) introduces three assumptions for establishing measurement for software quality: There are no universal quality measurements, but meaningful measures for particular environments. Secondly, widely accepted quality measurements require maturity in research and thirdly, quality indicators predict or indirectly measure quality. In short, Jørgensen establishes that there are no universal measurements, but the approaches using quality indicators – characteristics and attributes – can be used to approximate or predict software quality.
Jørgensen also discusses the different aspects of software quality. In addition to a set of quality factors, there also exist other definitions for quality; quality as user satisfaction and quality as the degrees of errors in software. However, both of these other models have serious flaws. In quality as user satisfaction, the most obvious flaw lies in the question as to why the measurement of user satisfaction is called software quality? There exist several groups of users for software, such as administrators and basic users, so how can the total satisfaction be calculated? Furthermore, how can the user group A’s “very satisfied” be related to group B’s “very satisfied”? They may not even mean the same concept, or at least may not be based on the same features. In the quality as the degrees of errors, the problem lies within the classification; how many flaws in the user interface relate to a critical system error? Therefore, by Jørgensen’s definition, the most sensible model for estimating quality seems to be based on the characteristics, observing different aspects of the software. However, criticism also exists towards this approach, for example by Salvaneschi and Piazzalunga (2008).

In the standard ISO/IEC 25010-3 Software product Quality Requirements and Evaluation standard (2009), the definition of software quality is similar to the interpretation presented by Jørgensen. In the standard, the software quality is defined in generally applicable and measurable terms. The quality is presented as a composite of eight quality characteristics, such as operability, security, or compatibility. These characteristics are further divided into sub-characteristics such as fault tolerance, accuracy, or compliance, which aim to be measurable either by internal or external measurements (ISO/IEC 2009). The product quality is understood to be an amalgam of all of the quality characteristics, with a prioritization and weight distribution based on the quality objectives. The quality model is illustrated in further detail in Figure 5.

![Software product quality model](image)

Figure 5: Software product quality model as presented in ISO/IEC 25010

In addition to the software quality characteristics, another indicator for software quality requirements is the criticality (adapted from Boehm and Turner 2003, Huang and Boehm 2006). Software criticality is an approximate indicator, indicating the worst possible outcome for the software failure. The criticality is represented as a scale from one to five, with the following descriptions for each level of criticality:
1. None or at most user irritation; for example “user has to reboot the game system”
2. Small economic losses; “the ticket fails to print and money is lost”, “no record of sale is made”
3. Significant economic losses; “Store has to be closed for a couple of days”, “product stock has to be scrapped”.
4. Bodily harm or great economic losses; “Operator loses hand”, “production line has to be closed for repairs.”
5. Loss of human life; Operator or people depending on the software system are killed.

The criticality of the software product may affect the quality objectives of a software organisation, and possibly correspond with the amount of resources allocated to the test process.

2.5.2 Test process development

Other main objective of this study was to observe and identify important test components to understand how they should be addressed from the viewpoint of test process development. Identification of the important components could offer external assistance to the organisations in the adoption of practices and operating models such as the model illustrated in the ISO/IEC 29119 standard. This objective required the study to observe the test process and test strategy from a viewpoint that consisted of all test process components, and observations regarding how the real-life organisation developed their testing practices.

The first objective was to assess whether the ISO/IEC 29119 model itself was feasible enough to implement in a real-life organisation. To assess the test process model feasibility, an understanding was required of the software process improvement (SPI) in real-life organisations. SPI literature includes studies about the effect of different factors on software process improvement. For example, a study by Abrahamsson (2001) discusses the requirements for successful process improvements. The most important factor according to the Abrahamsson study is the commitment to change from all organisational levels. If some of the levels disagree with the process improvement, the process improvement process tends to fail. In addition, the process improvement has to be executed in a controlled, well-planned, and organised way to ensure the possibility of permanent, positive improvements. In a more specific example, Pino, Garcia and Piattino (2009) discuss process improvement in small-sized companies. They conclude that process improvement should define management-level commitments immediately after the improvement process is established, and that improvement proposals are sometimes hard to align with the strategic planning in the organisation. They also consider that organisations should have advisers to initially trial the first few improvement iterations. Similar findings are also reported in the articles by Sulayman and Mendes (2010) and Hardgrave and Armstrong (2005). In addition, the article by Wong and Hasan (2008) also includes cultural influences in process improvement considerations. Culture, whether it is organisational culture or national culture, affects the requirements for effective process improvement. For this reasoning, as process improvement would assume cultural changes, it is important to also study the aspects of the social science in SPI (Conradi and Fugetta 2002).

In studies applying certain process models in organisations, the Hardgrave and Armstrong study (2005) observed that their case organisation had trouble reflecting their existing process in the theoretical models. In their paper, the organisation estimated the time needed for process improvements to achieve CMMi (CMMi 2010) level 2 as 10 months, when in fact the entire process took four years. In their reported case, the organisation decided to employ external advisors after 16 months of internal process improvement. Hardgrave and Armstrong also conclude that organisations tend to lose the initial drive for process improvement because the drive for an improvement process, in many cases, is not the internal need to develop, but rather to reach out for certain external rewards, such as certifications. Kautz, Hansen and Thaysen (2000) describe a case, where a simplified iterative development model was introduced into an
organisation and applied in the practice. Their main finding was that organisations can adjust to given models, provided that the model itself is sound and is not too strict with the process requirements.

Dybå (2003) conducted a study on SPI activities in different types of organisations. They concluded that the company size does not hinder or restrict the process improvement activities. Small organisations are at least as effective as large ones in implementing process improvement. Small organisations tend to be less formal in organisational hierarchy and in turbulent business environments they use explorative methods more willingly. Another interesting observation was also that organisations have a tendency to define their own best practice methods, as regards what is working, while failure in process improvement is considered an unacceptable possibility. As process improvement projects often fail, companies tend to support the status quo if corrective actions are not absolutely necessary. Dybå also discusses the explicit process definitions, which should also be understood as a guideline; informal practices are used to supplement the formal way of working, and collect experiences for the subsequent improvement efforts. Overall, the literature indicates that organisations can adopt different models if the model is sound and reasonably adjustable (Kautz, Hansen and Thaysen 2000), and that the size of organisation does not restrict their ability to make process improvements (Dybå 2003). It is also indicated, that process development is dependent on several stakeholders and other contributing factors (Abrahamsson 2003, Wong and Hasan 2008), and that organisations tend to have difficulties in observing and changing their own processes without external assistance.

In the development of the process improvement framework, the general requirements for any relevant construct should include, at least, that it is acceptable to the software development community and that it is based on agreed software engineering principles and practices (Burnstein et al. 1996). For example, the validity issues for developing frameworks have been addressed in prior studies (Jung 2009, Karlström et al. 2005). Jung (2009) developed a test process maturity model based on internal needs, and validated the results via a case study and a survey. Similarly, with the minimal test process framework (MTPF) developed by Karlström et al. (2011), the initial model was designed based on observations in real-life organisations, and further elaborated and validated with surveys and an empirical case study.

In larger scale test process development, there are some testing-related process frameworks such as Test maturity model (TMM) (Burnstein et al. 1996) and Test improvement model (TIM) (Ericsson et al. 1997). The TMM framework was developed from the principles of CMM and a group of other pre-existing practices (Burnstein et al. 1996) to allow organisations to develop their test process towards better principles and practices. TMM was developed with three main objectives. The first objective was to create a set of levels that define testing maturity hierarchy, where each level represented a stage of evolution towards mature testing capacity. The second objective was to create a set of maturity goals for each level, which gives the organisation a concrete example for development. The third objective was to create an assessment model, which would allow the organisation to obtain a clear understanding of their situation (Burnstein et al. 1996). Currently, the TMMi reference model covers 16 test process areas, divided into five different maturity levels from managed process at level 2 to the self-optimizing process at maturity level 5 (TMMi 2010).

Test Improvement Model (TIM) development has been based on developing the TMM and CMM (See CMMi 2010) principles further, by introducing the positive traits of the existing process models in a new context and new application method. The TIM development focused on two major components, a framework consisting of level ladders and key areas, and the assessment procedure itself. The important innovation of the TIM model was the ability to assess the current state of the practice in the key areas of testing independently, and put the assessed organisation “on the map” with their current test process.
The viewpoint of this study regarding the test process development and process improvement is not as straightforward as it may seem, as there are some considerations in the development of a framework for adopting the existing process model. The development of a framework for self-assessment and adoption is necessary, as the test process models (such as the TMM and subsequently the ISO/IEC 29119 model) are rather difficult to adopt in a real-life organisation, as they lack the guidelines in adoption of the process activities, and organisations tend to try to preserve the status quo (Dybå 2003). Organisations also tend to favour only the process improvement proposals, which they can relate to (Hardgrave and Armstrong 2005). Even if the adoption model exists, the adoption process is not easy to implement; for example the TMM process adoption model TMMi (TMMi 2010) has been criticised for being counter-intuitive (Jung 2009) and unrealistic to implement (Oh et al. 2008) even if the model itself is fundamentally based on the best practices and related standards (Burnstein 1996).

By identifying the most influential test process components and their relationship to the whole software process, a framework for test process improvement can be defined to steer the test work in real-life organisations towards better practices such as increased cost-effectiveness or risk-avoiding techniques. Overall, the objective is to enable the organisations to assess their test process needs more accurately, and be able to develop their testing work towards better practices and better quality. The development of self-assessment framework or adoption model is important; if no such adoption model is available, it seems that the organizations gain only limited benefits from the developed process models.

3. RESEARCH METHODOLOGY

To address the research problem, i.e. what components contribute to the software testing process and how should they be addressed in the development of a test process, the problem was decomposed into a group of sub-problems which were discussed separately in several smaller studies. The objective of the first sub-problem was to identify the contributing factors from the prior research into the topic. The objective of the other sub-problems was to study the effect and relevance of the identified testing factors and derive process improvement hypotheses by analysing the research subjects from selected viewpoints with quantitative and qualitative methods.

Software testing and related software development in organisations formed the research subject. To initially describe the study subjects, international standards were used to define a software organisation and the activities and sub-processes which happened within the organisation. The standards ISO/IEC 12207, Software Life Cycle Processes (2008), ISO/IEC 15504-1, Concepts and Vocabulary (ISO/IEC 2002), ISO/IEC 25010-3 Software Engineering – Software product Quality Requirements and Evaluation (SQuaRE) - Quality Model (ISO/IEC 2009) and the working draft for ISO/IEC 29119 Software and Systems Engineering— Software Testing (ISO/IEC 2010), define a software organisation, which was used as an a priori framework for the research subjects. The ISO/IEC 29119 test process model was applied in this study even though the international standard was still only a draft as it defined a size and maturity-independent definition of the test process in software organisations. The concept of the standard model is to be applicable in any organisation testing software, regardless of the size, business domain or product type, and simultaneously be applicable in cooperation with other established international standards such as ISO/IEC 12207 and ISO/IEC 15504. In this study, the research subject was initially understood to have processes in both software development and testing; conducting one coherent software process similarly as defined in the standards ISO/IEC 12207 and ISO/IEC 15504 in development and ISO/IEC 29119 in testing.

The research process consisted of three phases: preliminary study (viewpoints for the study), main data collection and analysis phase (identification of important factors) and validation phase (studies in test process improvement). In the selection of the research methods, the objective was to find the best method to
approach the research subject. For the preliminary phase, the Grounded Theory method (Strauss and Corbin 1990) was selected for the analysis of the prior data, a decision which was based on the type of the existing data, and considered feasible approach for extended application in the latter qualitative research phases. The survey method (Fink & Kosecoff 1985) was used to collect the quantitative data.

3.1 THE RESEARCH PROBLEM

According to the literature, more than 50% of development effort is frequently focused on testing (Kit 1995). On the other hand, testing can also be effective with only a small portion of the “optimal” resources (Petschenik 1985, Huang and Boehm 2006) and in many cases, the test processes have to adjust to resource limitations (Slaughter et al. 1998), so as an organisation the test organisation has to be adaptive and to some degree, even creative. However, the studies on organisational decisions and activities concerning the test strategy composition and test process components are less common. There are some organisation level studies which introduce organisational level test process components (Ng et al. 2004) and metrics (Chen et al. 2004, Afzal and Torkar 2008), but the studies on test process from the viewpoint of the organisation and the test strategy were limited. The identification of the important test process components should be done to ensure that at the organisational level all the important factors of testing are addressed. When the important factors are known, the test organisation can be developed towards better practices by removing hindrances and introducing changes, which are not in conflict with these components. Moreover, by understanding which components of the test process are important, the different test process models such as the ISO/IEC 29119 can be assessed for feasibility in a practical organisation.

As for the research approach to the organisational test strategy, the impact of different factors such as tools, methods, personnel, test case design and quality criteria required further investigation. One feasible approach was to analyse the practical impact of different components to the test process, and determine how the test strategy differs in different types of organisations. The identification of major contributing factors to the test process efficiency and perceived end-product quality would be especially helpful in allowing organisations to achieve better practices. If the identification was successful, it could also be worthwhile investigating whether there are certain test strategies for certain types of organisations which can be generalized into different template models for test strategy. Based on literature review, this approach was plausible, as the concept of “Generic Test Strategy” (De Grood 2008) already exists, and is used to define the general approach for the test process. In addition to generalization, developing the concept of preferred test strategies and identifying important test process components for test organisations was also considered beneficial outcome. In this way, the theoretical ISO/IEC 29119 model and practical testing done in organisations could have been adjoined.

The research problem, i.e. which components affect the software testing strategy and how they should be addressed in the development of test process, was decomposed into sub-problems. The sub-problems approached the overall topic by first identifying the central concepts and viewpoints (sub-problem 1). Sub-problems 2, 3, 4 and 6 were used in the qualitative analysis of the current software test processes, concerning the emergent special questions of how different test strategy components affect the test process and the identification of the important test components from the viewpoint of the test process. Finally, sub-problems 5 and 7 (applying qualitative analysis) focused on assessing the test process and test process improvement as an organisational activity. Sub-problems, objectives of the studies, and the respective publications are listed in Table 1.
Table 1. Decomposition of the research problem

<table>
<thead>
<tr>
<th>Sub-problem</th>
<th>Objective of the study</th>
<th>Publication</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Which are the current problems and enhancement possibilities for software</td>
<td>Specification of the concepts and additional viewpoints for research problem.</td>
<td>Kasurinen et al. 2009a</td>
</tr>
<tr>
<td>testing process?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Which methods and tools are applied on real-world software testing?</td>
<td>Identification and decomposition of common testing practices which are applied in real</td>
<td>Kasurinen et al. 2009b</td>
</tr>
<tr>
<td></td>
<td>world testing.</td>
<td></td>
</tr>
<tr>
<td>3. How organisations develop software and does the selected approach affect</td>
<td>Study the effect of development methods and agile development on the testing practices.</td>
<td>Kettunen et al. 2010</td>
</tr>
<tr>
<td>the testing practices?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. How does the organisation decide on what has to be tested?</td>
<td>Identify the different selection methods and prioritization process of test cases in</td>
<td>Kasurinen et al. 2010</td>
</tr>
<tr>
<td></td>
<td>projects.</td>
<td></td>
</tr>
<tr>
<td>5. How and when do the organisations develop their test processes?</td>
<td>Analysis of the requirements and approaches applied when organisations decide to improve</td>
<td>Kasurinen et al. 2011b</td>
</tr>
<tr>
<td></td>
<td>their existing test process.</td>
<td></td>
</tr>
<tr>
<td>6. How do the software quality-related aspects reflect to the test process?</td>
<td>Analysis of the effect of quality-related aspects from the viewpoint of test process.</td>
<td>Kasurinen et al. 2011c</td>
</tr>
<tr>
<td>7. How applicable is the test standard process from the viewpoint of the real</td>
<td>Development and analysis of a process improvement framework, which applies the ISO/IEC</td>
<td>Kasurinen et al. 2011a</td>
</tr>
<tr>
<td>world organisations?</td>
<td>29119 test process model.</td>
<td></td>
</tr>
</tbody>
</table>

3.2 RESEARCH SUBJECTS AND THE SELECTION OF THE RESEARCH METHODS

The ISO/IEC 12207 Software life cycle processes (ISO/IEC 2008) standard was initially used to describe the research subjects, software-producing organisations and their product outputs. In ISO/IEC 12207 the organisation and products are described to compose a set of processes. This definition was expanded in the testing-related processes with the test process model defined in the ISO/IEC 29119 Test Standard (2010), which defines the organisation and process activities from the viewpoint of testing. In addition of organisation model, ISO/IEC 15504-1 (ISO/IEC 2002) was applied to define the fundamental concepts for process improvement, as this standard offers an assessment model for organisational processes defined in the ISO/IEC 12207. In addition to the process models for development and testing, the definition of software quality was taken from the standard ISO/IEC 25010 Software product quality requirements and evaluation quality model (ISO/IEC 2009) to enable the study to assess the output of the research subjects. In this model, the software quality is defined as an amalgam of eight quality characteristics, which each have a number of objectively measurable or evaluable sub-characteristics, which describe the software specific activities and the system context on which the software is developed and maintained.

From ISO/IEC 15504, a concept of an organisational unit (OU) was also derived to define the organisational subsets studied in this research. As defined in ISO/IEC 15504-1 Concepts and Vocabulary (2002), an organisation unit is a part of an organisation, which deploys one or more processes with coherent processes context and operates within a coherent set of business goals. An organisational unit can consist of one specific project or a specific group responsible for one product within a larger corporation, but especially in micro and small-sized (EU 2003) companies, one organisation unit can consist of the entire company. In larger organisations, an OU operates mostly independently, but receives some amounts of organisational level steering from the upper management. In smaller organisations, the organisational management operates within the OU or is directly above it. As the large companies may have different business goals than the small companies, it was unfeasible to compare them directly; similarly different projects may have different purposes and goals. The reason to apply OUs as an assessment unit instead of entire corporations or projects was to normalise the differences between the organisations, and minimise the effect of different objectives and business goals, and to enable direct comparison between research subjects.
In Grounded Theory (Glaser and Strauss 1967), the objective of the research is to present an accurate description of what is being studied, and by methods of reduction and generalisations to build a believable descriptive narrative and chain of evidence from observations to a descriptive model with little or no interpretation on the studied phenomenon (Strauss and Corbin 1990). The Grounded Theory method allows the research question freedom to explore the phenomenon in depth, and allows a broader viewpoint on the topic than quantitative approaches. The Grounded Theory method was selected as an analysis method in the preliminary phase of the study, as the nature of the research topic and the existing data was considered too broad and unstructured for quantitative analysis. This method was considered appropriate, as the Grounded Theory method is in general considered suitable to uncover and understand complex phenomena founded on large ecosystems and gain novel and fresh viewpoints on areas, which are otherwise generally well-known (Strauss and Corbin 1990).

In the main data collection and analysis phase, the Grounded Theory method was applied as it suited the purposes of the study as the research topic, test processes in organisations, was considered a large and broad topic. The concept of conducting the study by using some form of action research (for example, Susman and Evered 1978) was rejected as the possibility of affecting the organisations and studying the effect of the changes, which forms the core of the action research approach, was limited.

On selection of the Grounded Theory, the second decision was then between the disciplines of Glaserian (outlined in Glaser 2002, van Niekerk and Roode 2009) and Strauss-Corbin (1990) approaches. The Strauss-Corbin-approach focuses on coding paradigms and in systematic categorisation and analysis of the collected data to uncover the relevant factors behind observed phenomena, whereas the Glaserian approach focuses on passive observation and emergence of strong codes from the data which then can be used to identify the relevant factors. In the preliminary phase, Strauss-Corbin was applied on the analysis of the existing data because of its codification method, which allowed detailed and structured analysis on the collected qualitative data set. In the latter phases, the Strauss-Corbin-method was applied as the number of organisations participating in the study was relatively high for a qualitative study, and the possibilities of passively and comprehensively observing the twelve organisations to the degree required by the Glaserian approach was considered unfeasible. Although the Glaserian approach is also a merited and appropriate method, the practical limitations and applicability in our research context made the Strauss-Corbin more suitable for the study purposes, and therefore it was applied throughout the research process.

In addition to the qualitative study using the Grounded Theory approach, quantitative data was collected from a survey (Fink and Kosecoff 1985). The survey method is an appropriate method to collect data from a standardised group of personnel, such as software development professionals such as software project leaders and test managers. The survey was also selected as an additional research method for the study to enable triangulation of research data (Denzin 1978). The triangulation of data in research means application and comparison of several types and sources of data to further validate the results. According to the literature (Seaman 1999, Paré and Elam 1997), the combination of quantitative and qualitative methods is usually more beneficial than applying either approach separately: statistical relationships found between the quantitative variables can be verified against qualitative data and vice versa. In this study, the qualitative data collected with the interviews and quantitative data collected with survey enabled the comparison between the data sources and was applied to further validate the results, as demonstrated in (Kasurinen et al. 2009b and 2011c).

3.3 Research process
The research process was divided into three phases. In the preliminary phase, the Grounded Theory method was applied on the previously collected interview data, along with a literature review on the relevant topics to establish basic understanding of the research area. Additional research topics were collected from the
expert group on the software testing, consisting of software engineering researchers and industry representatives. In the second phase, the main data collection and analysis, the research methods were a qualitative analysis using the Grounded Theory method on collected interview data, supplemented with a quantitative survey. In the third phase, the validation phase, the observations from the earlier phases were studied with additional interviews and subsequent Grounded Theory analysis. The research process, along with the different research phases, is illustrated in Figure 6.

**Figure 6. Research process and phases**

### 3.3.1 Preliminary phase
During the preliminary phase of the study, an extensive literature review was done to better understand the test processes and search for categories of interest and specify the viewpoints for further study. In addition to the literature review, the existing data from previous research project ANTI, reported in (Karhu et al. 2009, Taipale and Smolander 2006, Taipale et al. 2006a, 2006b, 2006c), were examined to establish basic understanding over real-life-testing and find appropriate seed categories (Miles and Huberman 1994). The previous research data was collected from five organisational units (OUs, see Table 2) which participated in the previous research project ANTI. This data which consisted of interview recordings, transcriptions, earlier codifications and interview memos, was codified according to the Strauss & Corbin Grounded Theory principles to identify strong categories in the interview themes of test process problems and enhancement proposals.
Table 2. Analysed organisations from the preliminary phase

<table>
<thead>
<tr>
<th>Business</th>
<th>Company size</th>
<th>Interviewed personnel</th>
</tr>
</thead>
<tbody>
<tr>
<td>A MES producer and integrator</td>
<td>Large/international</td>
<td>Testing manager, tester, systems analyst</td>
</tr>
<tr>
<td>Software producer and testing service provider</td>
<td>Small/national</td>
<td>Testing manager, tester, systems analyst</td>
</tr>
<tr>
<td>A process automation and information management provider</td>
<td>Large/international</td>
<td>Testing manager, tester, systems analyst</td>
</tr>
<tr>
<td>Electronics manufacturer</td>
<td>Large/international</td>
<td>Testing manager, 2 testers, systems analyst</td>
</tr>
<tr>
<td>Testing service provider</td>
<td>Small/national</td>
<td>Testing manager, tester, systems analyst</td>
</tr>
</tbody>
</table>

3.3.2 Main data collection and analysis phase

In the main data collection and analysis phase, the focus of the research was on collecting data on a large, heterogeneous group of real-life software organisations to understand how software testing in real life works. The areas of interest were to test whether the a priori constructs such as literature review and preliminary results were still valid, and in collecting data on testing-related aspects in both software development and in the testing itself. The data collection was done with two main approaches intended to complement each other. Qualitative data was collected for the Grounded Theory analysis in twelve “focus group” organisations based on theoretical sampling, and quantitative data was collected with a survey from 31 organisations, which were selected by supplementing the “focus group” with probability sampling.

The case selection criteria was set to include only organisation units, which as their main type of business activity develop software or provide software process-related services in a professional manner. Furthermore, on order to limit a possible company bias, the number of participating organisation units was limited to one OU per company, even if some larger companies could have participated with several different OUs. By applying this criteria, twelve OUs were selected as the “focus group” (see Table 3) based on the previous results and identified domain types. The sampling was theoretical (Paré and Elam 1997) and the cases were chosen to provide examples of polar types (Eisenhardt 1989), which meant that the cases represented different types of OUs, with differences in the business area, size of the company and market size. Theoretical sampling (Glaser and Strauss 1967) describes the process of choosing research cases to compare with other cases. The goal of theoretical sampling is not the same as with probabilistic sampling: the goal is not to collect representative sample of the entire population, but to gain a deeper understanding of the analysed cases and identify concepts and their relationships. In practice, the organisations were selected from a group of research partners and collaborators, and supplemented with additional organisations to represent organisation types not present. The actual data collection instruments were theme-based questionnaires and a survey, all available at address http://www2.it.lut.fi/project/MASTO/.
Table 3. Analysed organisations from the main data collection and analysis phase

<table>
<thead>
<tr>
<th>OU</th>
<th>Business</th>
<th>Company size / Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case A</td>
<td>MES producer and electronics manufacturer</td>
<td>Small / National</td>
</tr>
<tr>
<td>Case B</td>
<td>Logistics software developer</td>
<td>Large / National</td>
</tr>
<tr>
<td>Case C</td>
<td>ICT consultant</td>
<td>Small / National</td>
</tr>
<tr>
<td>Case D</td>
<td>Internet service developer and consultant</td>
<td>Small / National</td>
</tr>
<tr>
<td>Case E</td>
<td>Naval software system developer</td>
<td>Medium / International</td>
</tr>
<tr>
<td>Case F</td>
<td>Safety and logistics system developer</td>
<td>Medium / National</td>
</tr>
<tr>
<td>Case G</td>
<td>Financial software developer</td>
<td>Large / National</td>
</tr>
<tr>
<td>Case H</td>
<td>ICT developer and consultant</td>
<td>Large / International</td>
</tr>
<tr>
<td>Case I</td>
<td>Financial software developer</td>
<td>Large / International</td>
</tr>
<tr>
<td>Case J</td>
<td>SME business and agriculture ICT service provider</td>
<td>Small / National</td>
</tr>
<tr>
<td>Case K</td>
<td>MES producer and logistics service systems provider</td>
<td>Medium / International</td>
</tr>
<tr>
<td>Case L</td>
<td>Modeling software developer</td>
<td>Large / International</td>
</tr>
<tr>
<td>19 survey-only cases</td>
<td>Varies; from software consultancies to software product developers and hardware manufacturers</td>
<td>Varies</td>
</tr>
</tbody>
</table>

The data collection phase included three theme-based interview rounds, of which the second combined both qualitative and quantitative aspects. The companies were visited personally and 36 recorded interviews were carried out for the case OUs of the qualitative research, and an additional 19 interviews for the quantitative analysis to achieve the requirements of statistical relevance. The interviews were conducted by the project researchers to ensure that the interviewees understood the questions correctly. The duration of the interviews varied between one and one and a half hours and they were all tape-recorded and transcribed. A memo containing the issues emphasised was also written during the interviews.

The first interview round that was completed during the qualitative analysis served also as the review for the quantitative interview themes. The first interview round contained only semi-structured (open) questions, and the objective was to understand the basic practice of testing, identify the central themes for the next round, and in general, identify central concepts and factors of the test process in the real-life organisations. The interviewees were software or architecture developers or test designers. In some interviews, there was more than one interviewee present, for example a software developer and architecture developer. Such interviews usually lasted more than one hour. The questions on the first round were themed around the basics of the OU testing process, testing resources, software development processes and testing environment.

The interviewees in the second round were test managers or project leaders responsible for software projects. As earlier, the duration of the interviews varied between one and one and half hours and consisted of a survey and a supplemental set of semi-structured interviews, conducted by researchers working on the project. The objective of the second interview round was to achieve deeper understanding of the software testing practice and gain formal information on company testing framework and practices. The interviewees were selected to be managers and leaders because it was considered that they were more capable of assessing the test process from the viewpoint of the entire organisation.

The questions were theme-based and concerned problems in testing, the utilisation of software components, the influence of the business orientation, communication and interaction, schedules, organisation and knowledge, product quality aspects, testing automation, and economy. The structure of the questions varied from structured survey questions to supplemental, semi-structured, open questions. From the 19 interviews with the organisations only participating in the survey, the semi-structured interview answers were not included in the qualitative data analysis.

In the third interview round the interviewees were testers or programmers who had extensive testing responsibilities in the same OUs that were interviewed during the first and second round. Once again, in the
In two of the first round interviews, the organisation elected two people for the interview, as they considered that they do not have any individual worker, whose responsibilities match with the desired interviewee role. Additionally, on one occasion, the organisation was allowed to supplement their earlier answers in a later interview as the interviewee thought that the original answers lacked some crucial details.

**Data analysis with the Grounded Theory**

The grounded analysis was used to provide insight into the software organisations, their software processes and testing activities. By interviewing people in different positions from the software organisation, the analysis could gain additional information on testing-related concepts, such as different testing phases, test strategies, testing tools and case selection methods. Later this information was compared between organisations, allowing hypotheses on the test process components from several viewpoints and from the test process itself as a whole.

The Grounded Theory method contains three data analysis steps: open coding, axial coding and selective coding. The objective for open coding is to extract the categories from the data, whereas axial coding identifies the connections between the categories. In the third phase, selective coding, the core category is identified and described (Strauss and Corbin 1990). In practice, these steps overlap and merge because the theory development process proceeds iteratively. Additionally, Strauss and Corbin state that sometimes the core category is one of the existing categories, and at other times no single category is broad enough to cover the central phenomenon.

The objective of open coding is to classify the data into categories and identify leads in the data, as shown in the Table 5. The interview data was classified into categories based on the main issue, with any observation or phenomenon related to it being the codified part. In general, the process of grouping concepts that seem to pertain to the same phenomena is called categorising, and it is done to reduce the number of units to work with. In this study, this was done using ATLAS.ti software (ATLAS.ti 2011) which specialises on the analysis of qualitative data. The open coding process started with “seed categories” (Miles and Huberman 1994) that were formed from the research sub-question the publication was studying and prior observations from the earlier publications. Overall, the analysis process followed the approach introduced by Seaman.
(1999), which notes that the initial set of codes (seed categories) come from the goals of the study, the research questions, and predefined variables of interest. In the open coding, we added new categories and merged existing categories into others, if they seemed unfeasible or if we found a better generalisation.

<table>
<thead>
<tr>
<th>Table 5: Example of codification process</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Interview transcript</strong></td>
</tr>
<tr>
<td>“Well, I would hope for stricter control or management for implementing our testing strategy, as I am not sure if our testing covers everything and is it sophisticated enough. On the other hand, we do have strictly limited resources, so it can be enhanced only to some degree, we cannot test everything. And perhaps, recently we have had, in the newest versions, some regression testing, going through all features, seeing if nothing is broken, but in several occasions this has been left unfinished because time has run out. So there, on that issue we should focus.”</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

After collecting the individual observations into categories and codes, the categorised codes were linked together based on the relationships observed in the interviews. For example, the codes “Software process: Acquiring 3rd party modules”, “Testing strategy: Testing 3rd party modules”, and “Problem: Knowledge management with 3rd party modules” were clearly related and therefore could be connected together in the axial coding. The objective of axial coding is to further develop categories, their properties and dimensions, and find causal, or any other kinds of connections between the categories and codes. For some categories, the axial coding can also include actual dimension for the phenomenon, for example “Personification-Codification” for “Knowledge management strategy”, or “Amount of Designed Test Cases vs. Applied” with dimension of 0-100%, where every property could be defined as a point along the continuum defined by the two polar opposites or numeric values. Obviously, for some categories, which were used to summarise different observations such as enhancement proposals, opinions on certain topics or process problems, defining dimensions was unfeasible.

Our approach to analysis of the categories included Within-Case Analysis and Cross-Case-Analysis, as specified by Eisenhardt (1989). Basically, this is a tactic of selecting dimensions and properties with within-group similarities coupled with inter-group differences based on the comparisons between different research subjects. In this strategy, one phenomenon that clearly divided the organisations into different groups was isolated, and looked into for more details explaining differences and similarities within these groups. As for one central result, the appropriateness of OU as a comparison unit was confirmed based on our size difference-related observations on the data; the within-group- and inter-group comparisons did yield results in which the company size or company policies did not have strong influence, whereas the local, within-unit policies did. In addition, the internal activities observed in OUs were similar regardless of the originating company size, meaning that in this study the OU comparison was indeed a feasible approach.

Each chain of evidence was established and confirmed in this interpretation method by discovering sufficient citations or finding conceptually similar OU activities from the case transcriptions. Finally, in the last phase of the analysis, in selective coding, the objective was to identify the core category – a central phenomenon – and systematically relate it to other categories and generate the hypothesis and the theory. Overall, in theory building the process followed the case study research described by Eisenhardt (1989) and its implementation examples (Klein and Myers 1999, Paré and Elam 1997).
The general rule in Grounded Theory is to sample until theoretical saturation is reached. This means, until (1) no new or relevant data seem to emerge regarding a category, (2) the category development is dense, insofar as all of the paradigm elements are accounted for, along with variation and process, and (3) the relationships between categories are well established and validated (Strauss and Corbin 1990). In this study, saturation was reached during the third round, where no new categories were created, merged or removed from the coding. Similarly, the attribute values were also stable, i.e. the already discovered phenomena began to repeat themselves in the collected data.

Data analysis with the survey instrument

In the quantitative parts of the study, the survey method described by Fink and Kosecoff (1985) was used as the research method. According to Fink (2003), a sample is a portion or subset of a larger group called a population, which includes all organisations which are potential survey respondents. The sample in the survey should aim to be a miniature version of the population, having the same consistency and representatives for all relevant domain types, only smaller in size. In this study, the population consisted of organisational units as defined in ISO/IEC 15504-1. The sample was constructed by taking the focus group collected for the qualitative analysis, and supplementing it with probability sampling (Fink & Kosecoff 1985) to have sufficient statistical relevance, following principles presented by Iivari (1996). In practice, the sample was supplemented with 19 additional organisations, collected from the university and research group company contacts by random selection and confirming by a phone call that the organisation fitted the sample criteria. Out of a total of 40 organisations that were contacted, 11 were rejected based on this contact, as they either did not fit the sample criteria or decided not to participate on the study.

For the selected approach, the actual methods of data analysis were partially derived from Iivari (1996). He surveyed computer-aided software engineering tool adoption. The sample was 109 persons from 35 organisations. He derived the constructs from the innovation diffusion/adoption theory. Iivari estimated the reliabilities of the constructs using Cronbach coefficient alpha (Cronbach 1951). In factor analysis, he used principal component analysis (PCA) and in data analysis regression analysis. We used also used Cronbach alpha for measuring the reliabilities of the constructs consisting of multiple items and in comparisons of the correlations between different constructs with Kendall’s tau_b correlation. In these calculations, a specialised statistical analysis software SPSS (SPSS 2011) was used. A validated instrument increases the reliability of the measurements, but such an instrument was not available in the literature, so we designed our own interview instrument based on the questionnaire derived from Dybå (2004). This questionnaire was an instrument for measuring the key factors of success in software process improvement, which we in our study adapted to study the perceived end-product quality and the effect of different quality-related factors in software testing.

Related surveys can be categorised into two types: Kitchenham et al. (2002) divide comparable survey studies into exploratory studies, from which only weak conclusions can be drawn, and confirmatory studies, from which strong conclusions can be drawn. This survey belongs to the category of exploratory, observational, and cross-sectional studies as our intention was to study the different identified factors and observe their effect on the test process and end-product quality.

The survey was conducted at the second interview round during the face-to-face interviews. A few open-ended questions were located at the end of the questionnaire to collect data for the qualitative study. The questionnaire was planned to be answered during the interview to avoid missing answers because they make the data analysis complicated, for example, for the calculation of correlation. For these reasons, a self-assisted, mailed questionnaire was rejected and personal interviews were selected. The questionnaire was
also piloted with three organisations and four private individuals before the actual data collection round to test the form and the questions for clarity and understandability.

### 3.3.3 Validation phase

In the validation phase of the study, the focus shifted from the identification of testing work-effecting process components to the entire process organisation. In this phase, the test process of the organisation, and subsequently the concepts of test process improvement were studied. The objective was to understand how the identified test process components should be addressed at an organisational level. Additional concern was to test the feasibility of the ISO/IEC 29119 test process model and develop a framework for organisations to develop their test process towards better practices and conformance with the principles presented at the standard-defined test process model.

**Data collection**

The validation phase had a new set of data collection interviews with a partially new group of participating organisations. Otherwise the interviews were organised similarly, as in the main data collection and analysis phase interview rounds one and three. The fourth round interviewees were test managers, as their viewpoint was considered, from the project-level organisation, the most suitable to assess and discuss the observations from earlier rounds and to assess the applicability of the standard process model within the organisations. The interviews were theme-based, including questions from themes such as test strategy, test policy, test planning, testing work in general, software architecture, and crowd sourcing. A list of interviewed organisations is available as Table 6.

**Table 6. Analysed organisations from the validation phase**

<table>
<thead>
<tr>
<th>OU</th>
<th>Business domain, product type</th>
<th>Company size / Operation domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case M*</td>
<td>ICT developer and consultant, service producer</td>
<td>Small / National</td>
</tr>
<tr>
<td>Case N*</td>
<td>Safety and logistics systems developer, software products</td>
<td>Medium / National</td>
</tr>
<tr>
<td>Case O</td>
<td>Financial and logistics software developer, software products</td>
<td>Medium / National</td>
</tr>
<tr>
<td>Case P*</td>
<td>MES producer and logistics system provider, embedded software for hardware products</td>
<td>Medium / International</td>
</tr>
<tr>
<td>Case Q*</td>
<td>MES producer and electronics manufacturer, embedded software for hardware products</td>
<td>Small / National</td>
</tr>
<tr>
<td>Case R*</td>
<td>Maritime software systems developer, software products</td>
<td>Medium / International</td>
</tr>
<tr>
<td>Case S</td>
<td>ICT consultant specializing in testing, test consulting services</td>
<td>Medium / National</td>
</tr>
<tr>
<td>Case T*</td>
<td>Modeling software developer, software products</td>
<td>Large / International</td>
</tr>
<tr>
<td>Case U*</td>
<td>ICT developer and consultant, software production consulting</td>
<td>Large / International</td>
</tr>
<tr>
<td>Case V</td>
<td>ICT consultant specializing in testing, test consulting services</td>
<td>Small / National</td>
</tr>
</tbody>
</table>

* This organisation also participated in interview rounds 1-3

In addition to the fourth round of interviews, a validation step for (Kasurinen et al. 2011a) also included a study on four organisations based on the prior interview data. To confirm the findings of this study, three of the organisations were interviewed to review and collect feedback on the study results. A fourth organisation was offered the opportunity, but due to the changes in their organisation, they declined to participate in this part. Additionally, one interviewee from the fourth round interviews cancelled the interview for personal reasons, but provided written answers by email. Both of the interview sets, 4th interview round and validation interviews for the sub-problem VII (Kasurinen et al. 2011a) were analysed with the Strauss-Corbin Grounded Theory -approach, similarly to the previous research phase.
4. RESULTS

In this section an overview and the most important results of the introduced study sub-problems are shortly discussed. In the following, the sub-problems are summarised based on the objectives, results and impact as regards the whole study on software test process development.

4.1 SUB-PROBLEM I: OVERVIEW OF THE REAL-LIFE CONCERNS AND DIFFICULTIES ASSOCIATED WITH THE SOFTWARE TEST PROCESS

The objective of this Grounded Theory study (Strauss & Corbin 1990, Glaser & Strauss 1967) was to reveal important testing process issues and generate insights into how the testing processes could be enhanced from the viewpoint of the organisations, and what factors in testing seem to be the most usual problematic areas.

4.1.1 Results

The results indicated that the main components associated with testing process difficulties are most likely caused by the testing tools, knowledge transfer, product design, test planning, or test resource issues. According to the results, standardisation and automation levels in test process are not very high, and all cases the OUs had several enhancement proposals for immediate improvements in test processes. Similarly, it reinforced assumption that OU level comparisons between different sizes and types of organisations are feasible, as the results indicated similar issues regardless of the company of origin. Based on these results our study was able to pinpoint several key issues that were incorporated into the categories of interest in the following phase, and also gave insight on the testing infrastructure and operational framework of a real-life test organisation.

4.1.2 Relation to the whole

The results of this preliminary study was to examine the existing data on software organisations, to identify the test process components, and collect possible lead-in seed categories (Miles and Huberman 1994) for the main data collection and validation phase. Additionally, this preliminary publication was used to assess the feasibility of applying the Grounded Theory approach to the data analysis, even though the existing theory (Strauss and Corbin 1990) along with the studies by Sjoberg et al. (2007) and Briand and Lapiche (2004) supported the empirical observations on the test process research.

The results indicated several possible weaknesses in the test processes, such as, resource availability and allocation, weak testability of the software product, and testing tool limitations. The study also confirmed that in the qualitative studies, the different types of organisations could be studied and compared against each other by conducting the study on the organisation units (OU). Additionally, the study results indicated that an organisational study on software test process could be fruitful; most of the identified issues could have been handled by designing a better organisational approach, for example, by introducing test and resourcing plans. Overall, the generated hypothesis and results of the literature review in this publication were applied later in the development of the data collection questionnaires.

4.2 SUB-PROBLEM II: OVERVIEW OF THE TESTING RESOURCES AND TESTING METHODS APPLIED IN REAL-LIFE TEST ORGANISATIONS

The objective of this mixed method study combining both the Grounded Theory method (Strauss and Corbin 1990, Glaser and Strauss 1967) and statistical analysis was to examine and identify the current state of testing tools and test automation in the software industry. Another objective was to examine what types of
Software testing are performed in the professional software projects, and what percentage of total development resources are dedicated to software testing.

**Results**

The results presented further evidence on the practical test work, indicating that the test processes in organisations are defined but in many cases, not in a very formal way. Based on the results, it was established that majority of the organisations did have an established procedures which could be understood as a formal test process but in several cases these processes were only generally agreed principles or otherwise very open to interpretation. The organisations on average dedicated one fourth of their resources to the testing tasks, although variance between individual organisations was considerable. In a few organisations the test process was considered to be fully or almost fully resourced, whereas other organisations reported that as low as 10 percent of the optimal resource needs were available. The test resource results are indicated in a Figure 7.

![Figure 7. Testing resources available in software organisations](image)

As for the test tools and test automation, it was evident that automation is a costly investment, which can be done correctly but requires dedication and continuous commitment from the organisation in order to succeed. It was also established that most of the organisations do have testing-dedicated tools, the most common groups being test management tools, manual unit testing tools, test automation tools and performance testing tools. Similarly, as shown in Kasurinen et al. 2009a, the testing tools yielded results which indicated that the tools need configurability and extendibility, as several organisations also reported conducting test tool development themselves, not relying on the existing options.
Relation to the whole

Overall, these results give an insight into the test infrastructure and current state of software testing in the industry. The focus areas were on the applied tools and the purposes they are used for, discussing the automation tools in more detail. Other important observations concerned the test resources other than test tools, namely time restraints and human resources, and the types of testing methods applied in the test process.

The results of this study gave an insight into the amount of available resources in real-life organisations. The survey results indicated that the organisations do have access to a relatively high amount of test resources, as the average amount of resources was 70%\(^1\), and that on average 27% of the project effort is spent on testing. These values are somewhat different than those which could be expected based on literature and preliminary phase results. On a larger scale, the results of this study also meant that the test tools and test resourcing was generally at an acceptable level, and that the organisational management issues were more prominent than prior studies indicated. Furthermore, the average amount of effort allocated mainly to testing was less than expected, based on the software engineering literature (for example Kit 1995, Behforooz and Hudson 1996, Pfleeger and Atlee 2006).

4.3 Sub-problem III: Analysis of the effects the applied development method has on the test process

The objective for this Grounded Theory study was to establish the relationship between the development process and the test process, and assess how the development method affects the practical implementation of testing.

4.3.1 Results

The results from this study established several observations from test organisations. First and foremost was the observation that the development method itself is not a large influence on the way the testing is done, and that none of the development methods applied in the case organisations are inherently better or worse from the viewpoint of testing. In highly agile development, the approach allows more time for testing, as testing tasks can be started earlier than in traditional waterfall-approach, although there are some difficulties in deployment of testing in the early iterations. By applying agile methods the resource requirements for testing were also more predictable. This can be considered an obvious advantage in organisations, where testing resources are limited and distributed competitively between different projects. In agile development, the customer participation or at least cooperation with the clients is one of the key aspects. Overall, the agile practices when compared against the traditional waterfall-development style changes the testing only in a few ways. The customer needs to understand the requirements and differences of the applied development method, the test strategy focuses on testing the new features and functionalities and the organisation resource allocation and needs change. As for problems in testing, the agile development may expose the organisation to problems with making and following the test plans.

In general, the organisations which applied agile methods were also more flexible in terms of implementing and testing changes in the product. However, the agile approach also causes the development and testing to run in parallel, which is difficult to execute in practice and requires more coordination than traditional approach. From the viewpoint of strictly testing, agile methods offer some benefits such as early involvement

---

\(^1\) for example, if organisation had 3 testers and they considered that they would need 4, this would translate to 75% of resources.
or predictable resource needs, but also hinders testing in some areas, such as in availability and quality of documentation needed in the testing work, while making the test management more laborious.

4.3.2 Relation to the whole
This study observed the effect the development process has on the test process, and concluded that the effect of development style is not very relevant for the test process itself even though it changes some process dynamics such as resource needs in different phases and customer participation. Even still, the amount of agile development is relatively low in the software industry. However, the study results indicated that even if the real-life software organisations do not apply the entire agile development process, most of them have adopted some agile practices, such as code reviews, daily meetings or daily builds. For the whole study, the results of this publication indicate that the test process is feasible to study separately as an independent activity, as the way the software has been developed has only a small influence on how the testing work is organized.

4.4 SUB-PROBLEM IV: ANALYSIS OF THE TEST CASE SELECTION AND TEST PLAN DEFINITION IN TEST ORGANISATIONS
The objective of this Grounded Theory study was to observe and study the project level decision making in testing, and assess how the organisations decide on which test cases are included and which excluded from the test plan. The study also studied the prioritisation process of test cases, to establish if there were detectable patterns, which could explain the motivation behind the decisions.

4.4.1 Results
This study identified several components, which affect the decision making process and resulted to two stereotypical approaches on test case selection and prioritization method, named risk-based and design-based selection methods. The risk-based selection method was favoured in organisations, in which the test resources were limited or competed, and the decisions on test cases were made by testers themselves or designers in the lower levels of organisation. In design-based approach, the selection and prioritization process was done by the project-level management or dedicated expert. In the risk-based approach, the focus of testing was on verification, “what should be tested to minimise possible losses from faulty product”, whereas the design-based approach focused on validation, “what should be done to ensure that the product does what it is supposed to do”. More details are available in Table 7.

Overall, the study observed several testing-related components, which were tied to the test plan development by the decision makers, the role of the customer, the resource availability, and the development approach. In addition, it was established that the explorative testing, i.e. testing without a case plan, is also somewhat connected to the test case selection approach: in many organisations where the test plan was design-based, doing test work without clear cases - “just using the system” - was considered an unproductive ad hoc approach.
Table 7. Two stereotypical approaches for test case selection

<table>
<thead>
<tr>
<th>Category</th>
<th>Risk-based selection</th>
<th>Design-based selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test designers</td>
<td>Developers: programmers and testers</td>
<td>Managers: test and project managers</td>
</tr>
<tr>
<td>Development approach</td>
<td>Leans towards agile methods</td>
<td>Leans towards plan-driven methods</td>
</tr>
<tr>
<td>Testing resources</td>
<td>Limited</td>
<td>Sufficient</td>
</tr>
<tr>
<td>Explorative testing</td>
<td>Applied commonly</td>
<td>Applied rarely</td>
</tr>
<tr>
<td>Effect of policies in</td>
<td>Small; most decisions done in project level.</td>
<td>Large; most decisions are based on company policies or</td>
</tr>
<tr>
<td>decisions on testing.</td>
<td></td>
<td>customer requirements.</td>
</tr>
<tr>
<td>Customer influence</td>
<td>In the testing process</td>
<td>In the design process</td>
</tr>
<tr>
<td>Limitations of the model</td>
<td>Test case coverage may become limited.</td>
<td>Test process may become laborous to manage</td>
</tr>
<tr>
<td>Design concept</td>
<td>“What should be tested to ensure smallest losses if the</td>
<td>“What should be tested to ensure that the product does</td>
</tr>
<tr>
<td></td>
<td>product is faulty?”</td>
<td>what it is intended to do?”</td>
</tr>
</tbody>
</table>

However, test automation was observed to be rather independent from the test case selection approach. It seemed that the decision to apply test automation in testing work is based on other process factors, and test case selection or development of a test plan has only a little influence on it.

4.4.2 Relation to the whole

This study focused on the project-level management process activities. The theme of the work, the development of the test plan, and the prioritisation method of test cases studied, not only defined the process for the tested features, but also the low-level management of the testing work. The results indicated that all organisations have some systematic approach to deciding what should tested and that in all observed organisations, some form of formal test management existed, even though in some cases the role of the test manager was not defined. In addition, the observations propose that the selection method for test cases and the fundamentals behind a test plan tend to steer towards two identified strategies; risk-based or design-based selections.

4.5 SUB-PROBLEM V: ANALYSIS OF THE REQUIREMENTS FOR DEVELOPING TEST PROCESS OR ADOPTING NEW TESTING METHODS IN SOFTWARE ORGANISATIONS

In this qualitative Grounded Theory study, the focus was on establishing the requirements for organisation to start the test improvement process, and study how they adopt new testing techniques. An additional area of interest was to study how closely the ISO/IEC 29119 test process model (ISO/IEC 2010) fits the existing industry organisations.

4.5.1 Results

The main results of the study focus on the test process improvement in the software industry. The main observation of the study was that the organisations try to preserve the status quo, meaning that they do not develop their test process or try out new testing techniques unless the process is in dire need of changes. Even in organisations which continuously collect performance data and feedback from the test process, the results may be completely ignored if the existing process is “good enough”. As process development exposes the organisation to a possibility of failure and unnecessary costs, the threshold for conducting process development is high, even if the applied change would be positive and sensible. Based on the collected data, a model was defined for this process, along with an explanation of the limitations of adopting new test techniques and the development of test processes in organisations. The model is illustrated in Figure 8.
The second topic of interest in this work was the applicability and usability of the ISO/IEC 29119 test process in the real-life software organisation. The organisation representatives were asked to analyse how the proposed test process model differs from the approach the organisation is currently using and based on their experience and opinion, whether the model looks applicable or is it in need of changes. Based on the study results, the most common difference between the standard-defined test process model and the practical test process is in the number of organisational management and feedback process. The interviewed organisations considered the model to have too many details in the upper management, and that the model itself lacked support for actually adopting the process model. However, the overall concept was considered feasible, not omitting any major components or important concepts.

4.5.2 Relation to the whole
This was the first part of the study that observed the entire software organisation, and instead of one aspect of testing work, such as, project management or test infrastructure, studied the organisation’s behaviour. This study also explored the state of the test standard process model, finding some areas such as adaptability and the amount of details, which were considered difficult for the organisations. This study confirmed that the process model itself was feasible and did not omit anything obvious from the viewpoint of real-life software developers. On the development of the test processes, the study confirmed findings similar to those presented by Dybå (2008). Organisations prefer a status quo, and only conduct process development if the existing state becomes unbearable, even discarding the collected feedback data in a case where the process is at least in acceptable state. The organisations need a strong positive incentive to try out new techniques, even if the new method or proposed change in the way testing is done would seem sensible.

4.6 SUB-PROBLEM VI: ANALYSIS OF ASSOCIATIONS BETWEEN PERCEIVED SOFTWARE QUALITY CONCEPTS AND TEST PROCESS ACTIVITIES
The objective for this mixed method study, combining both the quantitative survey and qualitative Grounded Theory analysis, was to study the effect the quality-related aspects in software development and in software testing. In this study, the different quality characteristics as based on the upcoming ISO/IEC 25010 (ISO/IEC 2009) were tested in organisations, while different testing-related aspects such as outsourcing, open-source software in product and customer participation were studied from the viewpoint of perceived quality. A study by Garvin (1984) has identified the different types of software quality, and together with Jørgensen (1999) expressed a method of measuring the quality of a software product. In this study, these concepts were tested to see what types of quality are important to the software organisations.

Figure 8: Adopting new practices in test organisation.
4.6.1 Results

The central theme of this publication was in the different quality characteristics as defined in the ISO/IEC 25010 quality model, and studying how the perceived quality and different testing activities are related to each other. One of the most important observations of this publication was that almost all of the organisations do consider all of the different quality characteristics at least somewhat valuable to their product. The organisations were asked to evaluate how well the quality characteristic was taken into account in their product on a scale of 1-5; the averages only differed from 3,3 to 4,2 between the answers. Moreover, organisations were allowed to give a score 0, “this characteristic is irrelevant to us”, but this option was used only in 9 cases out of 248 assessments (3,6%), out of the 31 surveyed organisations. Results for each quality characteristic are listed in Figure 9.

Figure 9: The importance of different ISO/IEC 25010 quality characteristics.

The most important phase of a software process as a source of product quality was considered to be the development (average 4,3 on scale 1-5), whereas the test process was considered less important (2,9). The results also indicated that within the organisations, the level in which the organisations already follow the concepts of the test process was somewhat low (3,3 on a scale of 1-5) in organisational activities, 3,4 on project level management and 3,5 on fundamental level activities. Overall, the most important factors in testing, which positively affected the perceived end-product quality were identified to be the trust between the software organisation and the clients, as well as existing process conformance with the concepts presented in the ISO/IEC 29119 test process standard, and finally the identification and communication of the desired quality characteristics throughout the software organisation. In addition, some concepts such as customer participation in product design and general control over the development project were identified to be somewhat important.

---

2 1 = “this characteristic in our software is taken into account very badly”, 5 = “this characteristic in our software is taken into account very well”

3 1 = “fully disagree, or this level is very bad in our organisation”, 5 = “fully agree, or this level is very good in our organisation”
Besides the identification of testing aspects that affected the perceived end-product quality, the other important results were the aspects that were considered not to be very effective. Based on the survey and the qualitative analysis, such concepts as software criticality, product/service-orientation or outsourcing did not have a strong effect on the perceived end-product quality. Software criticality is obviously an important factor when deciding on how the product is produced, but changes in criticality do not alter the testing priorities or objectives of the testing work. Based on the results, it seems that the product domain is the most important factor affecting the selection of the tested components; software intended for Internet banking is generally tested for similar faults whether the target customer for the software is an individual users or a large corporations. Similarly, outsourcing was not considered a very important aspect affecting the perceived quality, in fact large organisations tended to think that outsourcing was helping the organisation to focus on the development of important features.

The last important observation was that the designers and testers rarely had similar ideas on the most important quality characteristics. In only two organisations were the same quality characteristics named and prioritised in the same order between designers and testers. Overall, this confirms some of the concepts presented by Garvin (1984). The organisations do not have one clear image on the preferred quality, and attention should be paid to identifying and communicating the desired quality characteristics to all stakeholders.

4.6.2 Relation to the whole

This publication applied two data sources, survey data as the main research method, and qualitative interviews providing additional viewpoints and validation for results which were considered important. In this publication, the test process was observed from the viewpoint of quality. Based on the survey results, it seems that development is considered a more important quality factor than testing. However, this can be somewhat expected, as the main objective for test process is in validating and verifying that the end-product is what was designed and works appropriately (Behforooz and Hudson 1996, Pfleeger and Atlee 2006), and quality can be understood in several contexts (Garvin 1984), one being that it “satisfies the user needs”. If the user or customer satisfaction is not met in a design or development, the testing work cannot fulfil that type of quality. If the users do not think the product is of a high quality, it is difficult to argue that the product is actually of a high quality, for example, because of its technical excellence. In this context, it is plausible to say that the source of perceived quality is not in the test process, but in the design and development. However, the test process does have an influence on the end-product outcome and profitability (Huang and Boehm 2006, Tassey 2002). Therefore, it can be argued that the test process is used to realise the potential quality in the developed software.

4.7 SUB-PROBLEM VII: SELF-ASSESSMENT FRAMEWORK FOR FINDING IMPROVEMENT OBJECTIVES WITH THE ISO/IEC 29119 TEST STANDARD

The objective of this study was to construct a maturity level-based framework to assess the existing test processes against the ISO/IEC 29119 standard process model and do preliminary testing on the validity and applicability of the framework.
4.7.1 Results

The concept was to combine the elements from a well-known and accepted software process evaluation model TIM (Ericson et al. 1997) to the draft of the ISO/IEC 29119 standard model to create a concept for a self-assessment framework. The objective was that the self-assessment framework could be applied to discover enhancement objectives in the organisational test process and alleviate the process adoption difficulties.

In the publication, a concept for combining the maturity levels from the Test Improvement Model and test processes of ISO/IEC 29119 was introduced. The different processes of the ISO/IEC 29119 standard model were assessed based on the maturity levels of TIM, which were customised to fit to the context of the processes:

- Level 0, Initial: The organisation does not have defined methods for this process activity.
- Level 1, Baseline: The organisation does have documented or at least generally agreed guidelines for these process activities, the process is systematically done to enable the finding and correcting of errors in the software.
- Level 2, Cost-effectiveness: The organisation tries to systematically promote cost-effectiveness or increase the efficiency of the process activities.
- Level 3, Risk-lowering: The organisation has metrics or other methods to enable the organisation to conduct risk-lowering and preventative actions in process activities.
- Level 4, Optimizing: The organisation has activities that aim to optimise the process; activities are done in a manner that is conceptually the same as in the standard.

The TIM model was applied as it was conceptually very similar to the standard; the key areas of TIM are assessed separately from each other, so that the organisation has a better understanding of what test process areas need most improvement. The evaluation work is easier to do as the number of simultaneously interacting concepts is kept reasonably small for an organisational assessment. Furthermore, the key areas of the TIM maturity model are similar to ISO/IEC 29119 processes; the organisation is conceptually close to organisational management process (OTP), planning and tracking to test management process (TMP) and TMCP, test cases to test plan process (TPP), testware to STP and DTP, and reviews to TCP. Overall, four organisations from the fourth round interview organisations were selected for the pilot study, and assessed based on the interviews held during the research project. The resulting profiles are illustrated in Figure 10.

![Figure 10: Assessment results using the experimental maturity levels](image-url)
Besides the developed profiles, a number of practical enhancement proposals were derived based on the observations. The profiles were also tested out with three out of the four profiled organisations to assess the accuracy and development needs for the framework. The fourth case organisation had recently changed their process, so they declined to participate in this assessment. The main points of this feedback is presented in the Table 8, where “++” denotes very very positive attitude towards the assessed attribute, and “−−” very negative.

### Table 8. Feedback from the case organisations

<table>
<thead>
<tr>
<th></th>
<th>MobileSoft</th>
<th>DesignSoft</th>
<th>SoftPortals</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Suitability of the framework</strong></td>
<td>+: The applied approach is generally feasible.</td>
<td>++: Practical approach on quick and easy assessment of the level of different testing tasks.</td>
<td>+: Levels are too universal, but model itself seems to cover everything needed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+: Levels in general are OK but the definitions should be less ambiguous.</td>
<td></td>
</tr>
<tr>
<td><strong>Suitability of the assessment levels</strong></td>
<td>− −: In large organisation, the levels overlap, unnecessary processes for some organisations.</td>
<td>+: Usable, although some processes do not need to be better than cost-effective.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>−: Levels in general are OK but the definitions should be less ambiguous.</td>
<td></td>
</tr>
<tr>
<td><strong>Accuracy of the profile</strong></td>
<td>−: The profile should be more detailed.</td>
<td>+: The profile was accurate enough, although with some differences.</td>
<td>++: The profile represents the organisation quite well.</td>
</tr>
<tr>
<td><strong>Accuracy of the results</strong></td>
<td>+: This type of feedback is always good for bringing out new ideas.</td>
<td>+: Results seemed usable.</td>
<td>++: Results same or similar to the internal discussions.</td>
</tr>
<tr>
<td><strong>Framework development proposals</strong></td>
<td>The assessment unit type and size should be clearly defined.</td>
<td>More definite descriptions for each framework level to reduce the overlap.</td>
<td>The assessment needs practical examples and more measurements.</td>
</tr>
<tr>
<td><strong>Best profiler</strong></td>
<td>An outsider from a third party, internal review is not accurate.</td>
<td>At least two manager-level employees; can be used internally.</td>
<td>A quality manager with a handpicked group of people, usable internally.</td>
</tr>
</tbody>
</table>

The overall attitude towards the developed framework was somewhat positive, although a few problems and enhancement proposals were identified. For example, the organisations considered that the framework profiling levels overlapped and were not very definite. Moreover, additional practical examples of actions denoting certain level of maturity were requested. However, the framework was applauded for the applicability as a tool for easy and fast assessment, and the accuracy of the results for being usable and similar to the issues discussed internally.

### 4.7.2 Relation to the whole

This publication discusses the test process from the viewpoint of process improvement. All of the results and ideas derived from the earlier studies, including the identified important test process components and applicability of the ISO/IEC 29119 standard, are applied in this publication in order to present a concept for an assessment tool to derive process improvement objectives. Obviously, the framework presented in this work is not complete, and it needs more studies before it can be established as a serious process improvement tool. However, the results and collected feedback from the proof-of-concept framework, so far
at least, suggest that this type of self-assessment method could be feasible to implement based on the ISO/IEC 29119 test process model.

5. IMPLICATIONS OF THE RESULTS

The objective of the overall research project was to study the organisational test process, identify important factors affecting the test process, and define guidelines for the organisations in order for them to pursue better testing practices and develop their process towards operating methods presented in the ISO/IEC 29119 testing standard (ISO/IEC 2010). This objective was pursued by applying both qualitative and quantitative research methods when observing the organisations, and trying to understand how test processes work in real-life software development.

The selected viewpoints - organisational test processes and development of test processes - were based on the literature suggestions and existing knowledge over testing in software organisations. In the preliminary phase of the study, a literature review on the topics and discussions with an expert group was used to understand the important factors of the study. Further concepts were derived from the earlier research project (for example Karhu et al. 2009, Taipale and Smolander 2006, Taipale et al. 2006a, 2006b, 2006c), from which the interviews regarding test process problems and enhancement proposals were used as a foundation for the data collection and analysis phase. The background work and analysis on test process problems based on the existing knowledge were reported in (Kasurinen et al. 2009a).

The assessment of different test strategy components was conducted in the second phase of the study, in the main data collection and analysis. In this phase, the components constituting the test strategy were divided to conceptual categories (see Figure 3), which were analysed over several smaller studies focusing on subproblems of the overall study. In addition to these categories, an additional category of “Other” was also used based on the literature review suggestions and earlier phase results in order to study other possible areas of interest.

The first categories analysed were the testing tools and the testing personnel in (Kasurinen et al. 2009b). This publication studied the test resources in the organisations, focusing on identification of different types of test tools available in the organisations, the amount and types of test automation and human resources. Based on the results we were able to understand the situation of the testing work in the organisations, and identify what kind of approaches the different organisations use for testing software. The situation in industry was better than what could be expected based on the literature review; there were some organisations in which there still were problems with quality and availability of tools or testing resources in general. However, the average amount of 70 percent of the test resources, when compared with the organisation’s self-defined optimum, was more than expected based on the prior knowledge. This resourcing level also indicated that the issues of testing are more related to the organising and managing of the process itself, not on the availability of resources. It was also apparent that the most important knowledge for testers was the domain knowledge, which was mostly attained by working in the field. Additionally, even though the organisations had positive attitudes towards different certifications and standardisation programs, they were not very common in every-day application. Based on these results, it seems that the testing work in industry is in a better condition than could be expected based on the literature. It also implies that the management aspects in testing are more important than originally thought; in many organisations the resourcing was not an issue, but the test process still experienced problems, mostly in the early test phases such as integration or unit testing.

Sub-problem III (Kettunen et al. 2010) focused on organisational aspects and on the effect of the development method. The study results indicate that the production method has only limited effect on the test
The end-product may be of high quality regardless of the applied production method, but based on the results it can be argued that successful application of the agile methods allows testing more time to work with the application in development and allows the organisation to be better prepared for test resource needs.

The application level of agile practices and principles was generally low, even though one organisation applied SCRUM principles in their development process. However, several organisations did apply some principles or practices which can be considered agile-oriented, such as daily builds, test automation, pair programming, code reviews or daily meetings, even if the amount of purely agile developers was limited. It was also apparent that agile development was favoured in patching and feature addition projects, whereas “traditional approaches” were favoured in main version development. This discussion was elaborated upon in (Kasurinen et al. 2011c), in which the effect of open source resources was discussed. The conclusion was that the open source resources are useful when applicable in projects, but they do not offer significant benefits over the “closed” - bought - third party modules, mainly because the open source material has to be reviewed and tested before being accepted into the product. From the viewpoint of the developers, it was also apparent that the source of the code did not matter, as everything went through more or less the same procedures. Overall, the results from Kettunen et al. (2010) indicate that the development method does not affect the test process very much. The test work and development are separate entities, and how the development work is implemented has only a minor actual effect on how the testing work is organised. Based on these results, it could be argued that in studies focusing on the test process, the development process itself is not a major concern, provided that the development follows at least some credible approach.

Kasurinen et al. (2010) continued with the test process implementation, and observed the test process from the viewpoint of developing the test plan and selection of the test cases. The most interesting result in this publication was the strong division of the test plan development into two approaches, design-based and risk-based approaches. Based on the observations the organisations divided into two main design approaches, “What should be tested to ensure smallest losses if the product is faulty” and “What should be tested to ensure that the product does what it is intended to do”. Stereotypically the risk-based approach was favoured when the amount of resources was limited and mainly the developers made the decisions, whereas design-based approach was used mainly when the amount of resources was not a limiting factor and decisions were affected by the customers and management. However, one important observation was that the project-level management does exist; in every organisation there was a person who was responsible for project-level test management.

Other observations include the application of test automation, which did not seem to follow the test plan pattern otherwise observed. Based on these results, it seems that the decision to apply test automation is not related to the applied approach on developing test plan. Another interesting finding was that the explorative testing was considered unprofessional and unproductive in several organisations. One possible explanation could be that the explorative testing is difficult to document, the results are difficult to predict and the effectiveness is dependent on the experience and professionalism of the tester doing the explorative testing. By applying these results in practice, the selection and prioritisation of applied test cases can be improved. Organisations should define what the test plan aims for, and based on that elaborate on the test plan development and selection of applied test cases. These results also confirm the existence of the project-level test management, indicating that the improvement activities focusing on test management can also improve the overall testing in projects.

The results of (Kasurinen et al. 2010) can be used in organisations to develop the process of creating a test plan, and understand the weaknesses and needs of the different approaches. Basically, it seems that the objective of test process in project level is either to minimize the possible losses or make sure that the
required features are acceptable. The results also indicate that at the project level, the test process activities are not always very formal, in many organisations, the designers and developers had a major influence on the testing decisions and even in large, well-resourced organisation some of the important test cases may be discarded if they are considered too resource-intensive. Furthermore, the role of the customer in development is not very active; usually the customer only approves the end-product in some form, not actively participating in the testing work.

The resulting end-product quality was observed in (Kasurinen et al. 2011c). Based on the earlier observations discussions of test resources and test tools (Kasurinen et al. 2009b) and the test plans in practice (Kasurinen et al. 2010), this publication assessed the outcome by means of quality model as presented in the quality standard ISO/IEC 25010 (ISO/IEC 2009). The most important observation of this publication was the uniformity in the quality characteristics. The prior indication that different types of organisations would strongly focus on certain types of quality did not hold true in practice. In fact, most organisations did have at least some over all concern regarding the different quality characteristics, and even when assessing the practical implementation of the said characteristics in their products, the differences did not focus on any particular characteristic. An additional interesting result was that the software criticality and desired quality characteristics did not have a strong correlation; the desired quality comes from the product domain, and has only a weak relationship with the possible repercussions of the product. From the other quality-related concepts, customer participation, product/service-orientation and outsourcing also had only a weak correlation. The customer is an enabler for quality, but the customer has to either provide substantial amounts of resources or a commitment to have any effect on quality, and in large organisations, outsourcing was not considered to have any meaningful effect on the perceived end-product quality.

A further interesting finding was that organisations, which considered themselves to closely follow the concepts presented in the ISO/IEC 29119 test standard, also considered themselves to produce good quality. This result indicates that if the organisation has organised their testing work in a manner that has a systematic approach, including the different documents and feedback system, they are more confident about their work. Organisations that have a systematic or at least a codified approach on testing, also have objectives for their testing work, and tend to know the general level of quality they are pursuing. This would also imply that by introducing the ISO/IEC 29119 concepts into an organisation the perceived end-product quality would improve, and that communicating the preferred quality helps the test organisation to focus on the important characteristics. Even if the test process does not have a large influence on the origin of the quality, identifying and communicating the preferred quality characteristics in test organisation improves the perceived quality.

With sub-problem V (Kasurinen et al. 2010b), the focus of the studied topics shifted from the different influential test components to the test process itself. The main result of this study was that the organisations do not actively pursue new techniques or ideas. In fact, organisations even discard the collected process feedback, if the process is “good enough”. This status quo mentality can be explained by several factors. The process improvement process and introduction of new ideas costs money, and there are no guarantees that the improvements always justify the expenses. Additionally, the change resistance causes conflicts. Other important result established in this publication was the feasibility assessment of the standard process model. Based on the results, the model was feasible but had some criticism over limitations in adoptability and excess details. Overall, the observation that the organisations tend to resist process changes would indicate that the organisations are reactive in nature, they do process improvement but mostly to fix problems, not to improve outcomes. In practice, this would indicate that the organisations should identify the process problems earlier, and in order to enhance output they should try to implement process changes before absolutely necessary.
In (Kasurinen et al. 2011a), the framework for assessing the test process against the ISO/IEC 29119 standard was introduced. Based on the prior results, this approach was considered appropriate, as it was established that the test processes are usually at least moderately resourced (Kasurinen et al. 2009b), the development process does not excessively interfere with testing (Kettunen et al. 2010), the project-level management exists in practice (Kasurinen et al. 2010), there are aspects of quality which are affected by testing (Kasurinen et al. 2011c), and the overall model is feasible enough for application in a practical environment (Kasurinen et al. 2011b). The framework was developed based on the concepts presented in the Test Improvement Model (TIM) (Ericson et al. 1997), by combining the TIM levels with the individual processes of the ISO/IEC 29119 model. Based on the feedback from organisations, the developed concept-level framework was a step towards a helpful tool, implying that there is a use for such a tool. The proof-of-concept framework can be seen as one of the concepts from this study, which shows considerable potential for future research.

Overall, the major implications for the test process development in practice can be summarised into a number of major observations:

- The test process can be assessed and developed separately from the development process. The development method does not affect the test process activities to a large degree, as the development process creates a product, and the test process validates and verifies this product.
- Besides resourcing, the test process hindrances and critical areas for development are also closely related to the organisational and project-level management, an observation which was established in several of the publications.
- The concepts presented in the ISO/IEC 29119 test process model seem to enable better end-product quality, as the organisations, which had implemented test processes which followed the principles similar to the standard, were also more confident regarding their end-product quality.
- Even though the test process itself is not a major source of the perceived product quality, the best way for the test process to enhance the perceived end-product quality is to identify and communicate the preferred quality characteristics to all test organisation participants.
- Organisations are reactive, they perform process improvements in order to fix problems, not to improve outcome. Organisations should identify the process problems earlier, and in order to avoid larger problems try to implement process changes before they are absolutely necessary.

6. LIMITATIONS

All research projects have shortcomings, threats to their validity and limitation on the scope of their results, and this work is no exception. Overall, the first limitation of this study is the applied scope of the study. The scope of organisational test process restricts the study to the process concepts coming from the software organisation, and thereby not taking into account the possible process hindrances caused by the external stakeholders, such as upper management, public relations, marketing or sales. However, this limitation has to be accepted to allow the comparison between organisations of different sizes and operating domains, as the concept of an organisation unit (OU) is used to normalise the differences between observed units and allow meaningful comparison between organisation types.

Another limitation in the qualitative study is the sample organisation limitations. The observed organisations are of a high technical ability and professional software developers, meaning that the results may not reflect the problems of starting organisations or organisations, which rely on volunteers, as is common in open source communities. It is also possible to formulate more dimensions for defining new types of software development organisations, and by applying those new definitions to finding types of organisations, which
were not covered in the theoretical sampling of this study. However, as with qualitative studies in general, outside the scope of the study the results should only be applied as guidelines and recommendations. When applied in a new ecosystem, as in the context of non-professional organisations, these results can be applied if the collected evidence suggests that there are enough similarities within the results of this study and the observations from the new ecosystem.

Overall, it can be argued that the number of organisations in this study is also rather low to allow a study of the effects of different concepts in the test process. Our objective was to establish the general effect of the different test process components and find the most important factors. It can be expected that by adding more organisations to the sample, the list of affecting factors would be more detailed, but the objective was not in compiling a comprehensive list of all possible variables, but to establish an understanding of the most important factors.

As for the quantitative survey sample, a sample of 31 organisations may also seem somewhat limited, but this limitation can also be avoided by designing the study to cater to these limitations. In our study, similarly to Ivivari (1996), the sample size is small but sufficient if analysed correctly. In our study, the threat of overfitting the data was addressed by selecting the organisations to represent different software domains and types of organisations, and triangulating the data with different approaches. This approach was also used to defer a non-response bias in the results; by maintaining heterogeneity in the sample, the results do not favour certain types or sizes of software producers. Additionally, in a paper by Sackett (2001) there is discussion regarding conceptualisation of signal-to-noise-ratio in statistical research. Their approach is to define confidence as being based in the practicality of observations: confidence = (signal / noise) * square root of sample size. In practice, this indicates that the confidence for the result being non-random weakens if the amount of noise increases while the signal decreases. In the Sackett model, the attributes are abstracted, meaning that the noise can be considered to be the uncertainty of the answers or any source of variation. Even if this Sackett model is not mathematical but more probably a logically defined conceptualisation, the concept is that the confidence is strongly related to the noise in the survey data. In our study, the response rate was 74 % for the organisations originally considered for the survey, and the data collection in all sample organisations was conducted by the same researchers who also participated in the survey design to avoid misunderstandings of the questions and to obtain a larger response rate from the sample (Baruch 1999). In this sense, it can be argued that the noise-ratio in the survey data is low, allowing more confidence as to the appropriateness of the survey answers and the data in general. This confidence is important, as the study by Armstrong (2007) argues, in studies covering aspects from social sciences there may be problems with statistical approaches and validation of answers with a statistical approach.

As for the validity of the qualitative research, there are threats that should be addressed to assert the validity of the results (for example Denzin 1978, Robson 2002, Onwuegbuzie and Leech 2007). Golafshani (2003) discusses the validity and reliability of qualitative research, and makes some notations on the reliability and validity issues. The reliability and validity in a qualitative study are not the same as the traditional mathematically proved concepts. In the quantitative study, the reliability and validity are rather a conceptualisation of trustworthiness, rigor, and the quality of the study. To increase the validity in a qualitative study, the research must eliminate bias and remain truthful to the collected data and observed phenomena. Similar observations are also discussed by Moret et al. (2007). Moret et al. points out that qualitative and quantitative studies should not exclude, but rather complement each other. Each approach has their method of validating the data, and if the research question is quantitative by nature, it is appropriate to apply a quantitative approach to collect that part of the data even if the overall subject requires a qualitative analysis.
To guarantee the validity of the study we used probability sampling when selecting the OUs for the survey, and theoretical sampling when selecting the in-depth cases in the qualitative study. Robson (2002) lists three threats to validity in this kind of research: reactivity (the interference of the researcher’s presence), researcher bias, and respondent bias and suggests strategies that reduce these threats. To avoid the researcher bias, the interviews were conducted by the researchers and for the data analysis, new researchers were brought in to participate in the data analysis to enable observer triangulation (Denzin 1978). During the data analysis for new publications, the new data (data triangulation) and the results were also compared with earlier quantitative and qualitative results (method triangulation) to further validate the study. Overall, the mixed-method approach allows the study to validate the overall results with comparisons and cross-referencing between different sources of data from both qualitative and quantitative sources. In this sense, the threats to the validity of the results from this study are low, as the results can be traced through different analysis methods and are based on overlapping, but ultimately different data sets.

7. CONCLUSIONS
This work makes three main contributions. The first contribution is based on the results of the main data collection and analysis phase, in which the effect of different test process-related aspects to the actual test process were identified and studied from the viewpoints of both qualitative and quantitative data. The second contribution is the assessment of the ISO/IEC 29119 test process model (ISO/IEC 2010) in practical organisations. This work studied the model concepts and the applicability of the model from the organisational viewpoint, assessing the feasibility of the model in practical organisations, and highlighting improvement needs for the model. The third contribution is the analysis of the test processes as a whole, studying the process improvement process of test organisations, and identifying the process difficulties.

Based on the presented results, the test process is an entity, which can be assessed and improved separately from the overall development process. The observations and analysis on the test processes yielded the following hypotheses for application in both research and industrial contexts:

1. The test strategy establishes a framework for testing work at the project level. The following hypotheses promote the development of a test strategy to address the factors important for the testing work:

- The development of a test plan can be characterised as applying two stereotypical approaches. The first approach promotes a design-based approach, in which the testing work focuses on validating the object under testing. The second approach promotes the risk-based approach, where the testing work focuses on minimising the potential losses caused by the object under testing.

- There is only a loose association between development methods and test processes. The applied development method does not restrict the practical testing work to any large degree, or require compromises in the test process definitions.

- The most important aspects in the test process which have positive association with the perceived end-product quality are trust between the customer and producer, a test process which conforms to the self-optimising processes as defined in the ISO/IEC 29119 standard and the communication of the preferred quality characteristics to all of the process stakeholders.

- In the test process resourcing, the organisations have an average of 70% of their self-defined “optimal” amount of resources and dedicate on average 27% of the total project effort to testing. Based on the study results presented in the literature and the survey data, the test process hindrances are also based on the efficiency factors and test management, in addition to simple resourcing issues.

2. The ISO/IEC 29119 test standard is a feasible process model for a practical organisation with the following limitations as regards for applicability:
The standard model can be characterised as being overtly detailed in the definition of roles and activities. In the practical test organisations, the boundaries of different levels, processes and roles are less organised than the model presents.

The process model is top heavy and places a considerable emphasis on the organisational aspects of the test process. Based on the qualitative analysis, the model defines several responsibilities for the upper management, many of which are performed, in real-life organisations, at the project-level management or at least not as systematically as defined in the model.

The current standard model does not include a roadmap or phased process for adopting the model. This hinders the applicability of the model in organisations, as the organisations had difficulties in creating an approach which their existing process could adopt for the concepts presented in the standard.

3. The organisations do not actively try out new test methods and prefer a status quo in their test process. The following hypotheses relate to the test process development at the organisational level:

- The organisations do not test out new testing tools or apply new testing methods unless they have a significant external incentive to do so. Based on the qualitative analysis, these incentives are things like the current state of the existing process or business needs in the operating domain.

- The organisational test process may have feedback processes in place to allow continuous development, but in practice, the organisations tend to disregard the evidence of process enhancement needs if the existing process still performs at least at an acceptable efficiency.

- In test process development, the organisations need a way of relating their existing process to the proposed changes to understand the objectives, and more pragmatically, the requirements the process improvement needs to succeed.

Overall, the test process can be seen as an autonomous part of the development process, which can be assessed and developed separately from the actual development. In software organisations, the test process is related to several components, and by developing the test process, it is possible to enhance the perceived end-product quality and achieve better cost/efficiency ratios. However, in practice, the organisations tend to avoid process improvement, allowing the test process to exist in a state where it could be developed, but there is not a great enough incentive to start the process improvement process. To lower the threshold for process improvement, the organisations need practical information to understand and relate to the requirements and objectives. One approach to achieve this is to focus on the concepts highlighted in this study and compare the existing process with the ISO/IEC 29119 standard model by applying the conceptual framework introduced in this study. By continuing the qualitative research on the test processes, this framework could be extended to allow more details and better support for the organisations to develop their testing practices.

REFERENCES


