Institute of Software Technology
Reliable Software Systems Research Group
University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Prozessanalyse Nr. 2

Analysis of Performance Testing Processes

Huy Viet Le Patrick Strobel Kevin Wenz

Course of Study: Software Engineering

Examiner: Prof. Dr. Lars Grunske

Supervisor: Dipl.-Inform. André van Hoorn

Commenced: November 01, 2013

Completed: April 29, 2014

CR-Classification: D.2.5, D.2.9

Kurzfassung

Trotz des Einflusses von Performance auf die Kundenzufriedenheit, werden funktionale Anforderungen oft bevorzugt. Aus diesem Grund untersucht diese Arbeit den Performance Test-Prozess im Rahmen einer Prozessanalyse bei der adesso AG. Hierzu wurden Mitarbeiter nach ihren Erfahrungen in Performance-kritischen Projekten befragt. Die Ergebnisse dieser Befragung werden ausgewertet und vorgestellt.

Durch die Auswertung der durchgeführten Interviews und des erstellten Fragebogens konnten einige Problemfaktoren ermittelt werden. Um diese festgestellten Schwächen der Prozess auszubessern, wird ein einheitlicher generischer Test-Prozess vorgestellt, welcher den Test-Prozess der adesso AG unterstützen soll. Es werden Verbesserungsvorschläge präsentiert, die die bestehenden Prozesse dem generischen Prozess annähern.

Abstract

Despite its huge impact on the customer satisfaction, performance is often preferred over functional requirements. For this reason, this work analyses the performance testing process at adesso AG in the context of a process analysis. Therefore, employees have been asked for their experiences in performance-critical projects. The results of this survey are analyzed and presented.

By analyzing the performed interviews and the built questionnaire, we were able to determine several issues. To mend these found weaknesses in the process, a uniform generic test process is presented, which is meant to support the test process done at adesso AG. Recommendations for improvements are presented, that bring the existing projects closer to the generic process.

Contents

Chapter 1

Introduction

1.1 Motivation

The perceived quality of an application is highly influenced by its performance. Although there are more quality attributes beside performance, performance is the one that is directly visible to the end-users. Research shows that people will switch to another product if the performance is insufficient, which causes loss in revenue. For example, users will leave a website if loading a page takes too much time, as mentioned by Liddle [Liddle 2008] and Stefanov [Stefanov 2008].

Despite its great importance, performance, as part of the non-functional requirements, is often neglected during the development of applications. Development is focused on functional requirements while little attention is paid on non-functional requirements [Molyneaux 2009]. Due to the fact that performance requirements are insufficiently specified and information about the application's real workload is barely available, problems concerning the performance are not detected until the application is deployed and put into production.

However, fixing bugs and performance issues in late stages during an applications lifecycle requires high effort and causes heavy costs [Ludewig and Lichter 2010, page 64]. To reduce these costs, insufficient performance has to be detect as early as possible and before the application is deployed. To do so, performance management has to be included into the whole development process: From detailed performance requirements in the specification phase, though performance tests when the application is tested, to performance monitoring when the application is deployed and running under real workload.

1.2 Goals

The aim of this process analysis is to examine the performance testing process at adesso AG, a medium-sized company focusing mainly on the development of software for the financial sector. In this process analysis, we will concentrate on performance testing. Performance testing is one important building block in the performance management, as it ensures that the application meets specified performance requirements.

This process analysis covers three main goals:

1. **Analyzing and documenting the current state.** We analyze how performance is currently tested at adesso AG. This includes tools, people and roles involved in the testing process.
2. **Identifying weaknesses and developing improvements.** Based on the results of our first goal, we search for weaknesses and present improvements to overcome the identified weaknesses in the overall performance testing process.
3. **Presenting the results.** The found weaknesses and improvements are presented to the employees at adesso AG. This allows them to improve their current testing process based on our results.

1.3 Document Structure

The remainder of this document is structured as follows:

- ▻ Chapter 2 gives an overview over the underlying foundations of our analysis. The general structure of process analysis, survey and testing techniques are presented.
- ▻ Chapter 3 describes the process analysis setting at adesso AG, the situation there and what people there expect from our analysis. This chapter relates to the first goal of our analysis.
- ▻ Chapter 4 describes the process we used to gather information about the current state at adesso AG, the performance testing process used so far at adesso AG, and roles relevant for performance testing. This chapter aims as a description of our analysis process rather than the processes used at adesso AG and therefore partly relates to our second goal.
- ▻ Chapter 5 highlights the results of our analysis. We present the current state at adesso AG and the proposed improvements. This chapter relates mainly to our second and third goal.
- ▻ Chapter 6 describes the threats to validity that influence the results of our analysis and gives an outlook for future work.
- ▻ The conclusion in Chapter 7 sums up our process analysis.

Chapter 2

Foundations

This chapter contains important basic concepts that are necessary to perform this process analysis. It starts by introducing the term *Process Analysis* (Section 2.1), than takes a deeper look into *Survey Techniques* (Section 2.2) and concludes with basics about *Performance Testing* (Section 2.3).

2.1 Process Analysis

In a process analysis, an existing business process is analyzed. Furthermore weaknesses are detected and improvement suggestions are developed. The whole activity of process analysis is described by Behr and Tyll [Behr and Tyll 2003] and depicted in Figure 2.1. It describes three steps including becoming acquainted with the companies structure, learning about their business process, and analyzing this business process:

- First the basic structure of the company has to be analyzed. Therefore the company's structure, its organization and its hierarchies need to be acquired. This step provides the basis for the next step. It is possible to find first weaknesses, for example a too deep hierarchy.

- Second the current business process, called *current state*, should be acquired. In this step a model of the important and relevant processes is built. It is only allowed to consider real processes; wishful thinking must be ignored. Information about these processes are gained by questionnaires, observation, and the analysis of existing documents. Questionnaires are discussed in Section 2.2. To document these models, UML's behavior or interaction diagrams[1] or BPMN[2] can be used. These notations include information about relevant actors, artifacts and their interactions.

- The third and last step is to analyze the business processes by analyzing the previously built model. Therefore the model is checked for weaknesses and possible saving should be found. To do this, the following methods can be used:

[1] UML: `http://www.uml.org`
[2] BPMN: `http://www.bpmn.org`

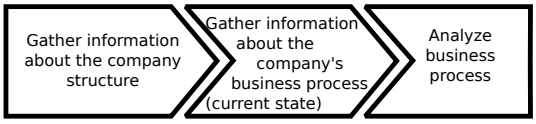

Figure 2.1. The structure of a process analysis. Adopted from Behr and Tyll [Behr and Tyll 2003]

- *Activity based costing* Break down costs to each sub-process and calculates its creation of value.
- *Weakness analysis* Find problems and weaknesses in organizational parts.
- *Benchmarking* Compare the company's processes with "best-practice-processes".
- *Time simulation* Analyse processing time.

2.2 Survey Techniques

Questionnaires and interviews are tools used for retrieving data in an empirical study. In order to avoid falsifying those data through unwanted influences, we familiarized ourselves with techniques needed for designing an empirical survey. In the following we will present some basic knowledge about the survey techniques, which are described in detail by Brosius and Koschel [Brosius and Koschel 2005]. This includes the type of questions, the process of creating a questionnaire or an interview and concluding with noises and falsifications.

2.2.1 Types of Questions

A questionnaire or an interview consists of many types of question, which all have different functionalities and purposes. These types are amongst others the factual issues, knowledge questions, opinion questions, behavior questions, and conditional questions, which can be further separated in subtypes. A *factual issue*, or a question about the matter itself, can be answered by a participant without long thinking and is independent of his physical state, knowledge, or his attitude towards a topic. Those kind of questions can be used to retrieve factual data about a person or his environment. An example would be like "*Do you use Eclipse as an IDE?*"or "*How long are you working on project xyz?*". A *knowledge question* can be used to determine the information both of the participant in order to weight his further answers. A similar purpose is fulfilled by the *opinion questions*, which can also be used to adjust the influence of his further answers based on his opinion towards a topic. For both aforementioned question types, it is recommended to use pre-defined answer possibilities, to ease the evaluation. Another type of questions are the *behavior questions*. Using those kind of questions, we can actually determine how the participants behave in particular situations. An example would be to determine how participants are testing their software

in a software project. An important difference between *behavioral questions* and *factual issues* is that *behavioral questions* do *not* ask directly, how a participant does something, but instead tries to get other information in order to assemble the answer. Hence, we can measure what is actually happening and not what the participants thinks happens in his point of view. The last kind of questions are the *functional questions*. Those kind of question routes the participant through the questionnaire or the interview. Examples are *conditional questions*, which can be used to filter participants out for a block of questions, if he is not qualified to answer it. Another example are *transitional questions*, which can be used to transition from one topic to another. Last but not least, there are the *demographic questions*, which retrieve personal information like gender, age, or the level of experiences towards a topic.

2.2.2 Creating a Questionnaire or an Interview

As described in the previous sections, there are many things to consider in order to conduct an empirical survey. Brosius and Koschel [Brosius and Koschel 2005] suggest a step-for-step guide in their book on how to conduct an empirical survey. The first step consists of formulating the scientific question that we want to research. Often, it is also necessary to do a literature research in order to be aware of particular things in this topic. In the second step, we have to refine our original scientific question into multiple *subject areas* containing so-called *question for programs*. These questions are mostly written using terms that only scientists have to understand, because we are still in the phase where we develop the first concept of our final questionnaire or interview script. The task of the next step is to transform the *questions for programs* into *test questions* that can be printed on the questionnaire or asked in an interview. To do this, we have to ensure that there are no foreign words, that participants might not understand. Further, we have to consider noises and falsifications which will be shown in the following subsection. When we finally have a release candidate of our questionnaire or interview script, we have to do a pre-test in order to find possible errors and avoid major problems in the survey.

2.2.3 Noises and Falsifications in Stating Questions and Retrieving Answers

There are many different noises to consider in the process of stating a question to a participant. Many of those can be avoided by trying to state the question as clear and unambiguous as possible, but there are also a bunch of cases which cannot be completely avoided. Those are amongst others the social desirability, primacy- and recency-effects or the Non-Opinions, which will be explained in the following. The *social desirability* describes a tendency, where the participant try to adjust the answer so that the society, his family or his principal agree in opinion. One possible solution to this problem is to ask what the participant thinks his co-workers or his circle of acquaintances would do in this case. Here we assume, that he represents a social group or he even belong to it. Another example are the *primacy- and recency-effects*, where participants perceive response options on the

first or last spots better than options, which are in the middle of the response options list. To avoid this, we should limit the answer options to seven, plus or minus two [Miller 1956] or shuffle the answer possibilities for each participants, so that all options are equally distributed over the different spots. We should also consider the so called *Non-Opinions*. Participants often have no opinions to a question. If we force them to answer a question by not explicitly providing a possibility for abstinence, they tend to answer with random opinions, which influences the evaluation in a bad way. There are many further noises to consider. A summary of them can be looked up in [Brosius and Koschel 2005, p. 92ff].

2.3 Performance Testing

When studying and testing an application's performance, different aspects have to be taken into account. This includes the software components, but also the used hardware, the network used for communication between the application components and the users and the environment. This sections introduces the most important aspects and terms needed to understand the performance testing process. The information given in this section is mainly based on the books by Molyneaux [Molyneaux 2009] and Meier et al. [Meier et al. 2007].

2.3.1 Performance Requirements

Performance requirements can be divided into service-oriented or efficiency-oriented requirements. The service-oriented requirements describe how well the application suits the needs of its users, whereas the efficiency-oriented requirements focus on the way the application uses the resources provided by its environment.

People use software for a large variety of tasks; from tasks that require less attention, like browsing an online shop, to tasks that require much attention, like filling in a insurance form. Since the required performance depends highly on the task that should be supported by the application and the level of required attention, there exists no industry standard on what is good or bad performance. Molyneaux gives an rough overview over different types of tasks and the maximum response time for each of these categories [Molyneaux 2009]. For example, if the user has to remember information between different steps, the transition from on step to the next should take less than two seconds.

Fixing performance bugs late in the lifecycle of an application causes high effort and costs [Ludewig and Lichter 2010, page 64]. Therefore, performance testing should be performed in early stages.

Before doing performance tests, different metrics have to be clarified. These are:

- The number of end-users
- The number end-users concurrently accessing the application

- The type of usages
- The expected increase in usage over time
- The number and location of servers
- The influence of network capacity

Consequently, a lot of time and work has to be spent to get all required information about the expected performance. Though this work is important, we will not dig deeper into this part of performance management, but concentrate on performance testing.

2.3.2 Goals

After each performance requirement and the expected workload of the application has been specified in collaboration with the stakeholders, the application can be analyzed and tested for different performance characteristics. Afterwards, the test results can be used to check if the applications fulfills set performance goals and for planning the later infrastructure.

- **Availability:** The portion of the time the application is usable (uptime) is measured. Shows if enough capacity and resources are available.
- **Concurrency:** Different users access the application at the same time. Gives a hint on the maximum number of users that can use the system concurrently, which in turn can be used to for capacity and resource planning.
- **Response time:** The required maximum response time differs between the specified functions. If existing, data from older versions of the applications can be used as a basis. Can be used the model the relation between workload and response time.
- **Network resources:** The influence of the network used to connect different application components or the network between the application and the user on the overall application performance is determined. Shows how slow network links effect the application.
- **Hardware resources:** Different hardware-metrics, like resource utilization, are monitored. Can be used to detect abnormal application behavior or hardware exceeding its limits.

2.3.3 Tests

Each performance test is scripted in a transaction, which describes the steps that must be performed for a single test case. Different information is encoded for each step. This includes the request that is sent to the application, the required results and the virtual user

that performs the request. For example, the steps for a transaction "Buy article" performed by user "John Doe", could be: Log in, browse the store, add to cart, send order, log out.

User profiles are built to executed transactions in a realistic manner. This profiles describe the virtual user's behavior between the steps, like delays because of thinking times.

Performance tests can be categorized in types listed below:

- **Baseline test.** Each transaction is executed with a single user to record the best-case performance of the application. The recorded data is used to compare the results of further tests.
- **Load test.** The application is put under the expected workload. Different virtual users execute the scripted transactions and simulate the behavior of the real users when the application is deployed. The test aims to check whether the applications is capable to handle the expected load and if it meets the specified performance requirements.
- **Stress test.** The applications workload is increased steadily. More and more virtual users are sending requests until the application crashes or is unable to response to user requests. This tests gives the upper limit of load the application can handle without failing.
- **Stability test.** The application is run under the target workload for a longer period of time and performance metrics are recorded. This test is used to identify bugs that will cause the application's performance to decrease (like memory leaks).
- **Smoke test.** Changes in the application are tested. Only transactions are executed that are affected by code changes. These are transactions whose performance might have changed between the revisions.
- **Isolation test.** Identified performance problems are tested. That is, only transactions are executed that have shown performance problems in tests run before.
- **Component test.** A single component of the application (like module or architectural component) is performance tested. For examples, module tests can be enhanced with performance requirements that shall be fulfilled by the tested module.

2.3.4 Tools

Before starting to test the performance, suitable tool have to be selected. Performance tests should be reliable and repeatable with low effort, as are functional tests. To support this, tools that automate the performance testing process need to be installed and integrated into the development process.

The performance testing tools provide a scripting interface through which test engineers specify transactions that shall be tested. After the transactions are scripted, the tool executes

the transactions through virtual users. Each virtual user generates the load as it is scripted and the metrics (like response time) are recorded automatically by the tool. Afterwards, the recorded metrics can be analyzed and compared to specified requirements or to prior test runs.

For testing web-applications over HTTP or HTTPS, many commercial and open-source tools are available, whereas outside of the scope of web-applications there are less tools. Rick Hower provides an extensive list of performance testing tool on his homepage [Hower 2014]. Some popular tools are listed in the following sub-sections.

Load generators

The tools listed below support scripting, virtual users and different network protocols to generate load on the application:

- **JMeter.** Open source, written in Java and developed by the Apache Software Foundation[3].
- **LoadRunner.** Commercial, developed by Hewlett-Packard available for Linux and Windows[4].

Load generators with cloud integration

The tools listed below are similarly to the tools above but can be integrated into to cloud to generate the load from there:

- **Silk Performer.** Commercial, developed by Borland[5].
- **WebLOAD.** Open source, marketed by RadView[6].

Component level

The tools listed below enhance component testing tools with performance measurements and therefore allow component tests only:

- **ContiPerf.** Open source, enhancement for JUnit[7].
- **JUnitPerf.** Open source, enhancement for JUnit[8].

[3] JMeter: `http://jmeter.apache.org`
[4] LoadRunner: `http://www8.hp.com/us/en/software-solutions/loadrunner-load-testing/index.html`
[5] Silk Performer: `http://www.borland.com/products/silkperformer`
[6] WebLOAD: `http://sourceforge.net/projects/webload`
[7] ContiPerf: `http://databene.org/contiperf`
[8] JUnitPerf: `http://www.clarkware.com/software/JUnitPerf.html`

2.3.5 Environment and Planning

Performance tests should be executed on an environment that is at close to the application's target environment as possible. However, running the tests on a live environment, that is serving user requests, is not an option because of its negative impact on the requests. Also the applications performance might differ between physical and virtual server systems. Additionally, realistic data is required to run the tests properly. These are, among others, the required input data (login credentials, queries, etc.) and target data to populate the application's database.

To build the test environment, enough resources have to be reserved. To run performance tests, at least three server have to be set up.

One or more server hosts the target application that shall be tested. On this server, a stable snapshot of the target application is installed that has already passed the functional tests. The application environment (database-, application-server etc.) on the server system has to be the same as on the target environment for the tests to be meaningful.

A second server hosts the performance testing tool that creates the workload. The tool executes the scripted transactions that simulate different virtual users interacting with the application. If the load for running the tests is too high for a single server, three or more servers can be used to run the tests. Thereby, one server is used to manage the tests executed on the two or more servers.

The applications is monitored on a third server. Different metrics are recorded to gain insight onto the applications performance and to see whether the performance goals set before are met. There might be uncontrollable influences on the goals—like network bandwidth between the end-users and the target environment—the might have to be taken into account.

Chapter 3

Process Analysis Settings

This chapter starts with an introduction about the in this process analysis participating company adesso AG. The second part contains their expectations.

3.1 Company Background

Adesso AG develops customer-specific software. Customers of adesso AG belong to different sectors of economy. Primarily these firms are medium- and large-scale enterprises from sectors of insurance, counter insurance, bank, financial service provider, health care and lottery [Adesso AG 2014].
As a full service provider adesso AG provides the spectrum from the development of a vision for a company to the operation of software. In most cases adesso AG appears as an IT-consulter. Therefore, adesso AG supports the customer in the development of its company vision and on this basis its business processes. If there is already an existing system that further should be used, this system will be integrated. Software systems are developed based on the company's business processes. Adesso AG provides operation responsibilities for the system.
In the developed solutions the focus lies on operation optimization. For example by cost reduction. Software is developed on marketable standards.
Adesso AG is also involved in research to be part of future trends and technologies [Adesso AG 2014].

Projects at adesso AG differ in their size and besides employees from adesso AG people from other firms are involved. These projects are of heterogeneous nature. Each project has a project leader, but there are no standard processes. The process is individual and depends on the type of a project. Adesso AG is located in different cities. A project can be distributed, but mostly customers are served locally. For testing, adesso AG has a group called the *Test Factory*, which takes care of testing. Not all projects have performance critical aspects. When a bigger project is performance-critical two to three percent (on a 200 persons project these are five to six people) of its participants are performance engineers. On smaller projects this job is done by standard developers.

3.2 Expectations

With this project adesso AG aims to check their performance testing methods. As a guideline for future performance-critical software development at adesso AG, we shall provide a preferably generic process that can be adopted on most projects. Therefore we need to evaluate their current state of performance testing. Since at adesso AG there is no standard way of performance testing we have to derive the generic process from their current processes. Through the generic process adesso AG should be enabled to build their performance testing processes like with a toolbox. It should also be investigated, whether such a generic process is reasonable.

Chapter 4

Procedure

This chapter first describes the project progression and explains steps that were taken, milestones, and other kinds of action that were necessary to achieve the projects objectives. In the second Section the evolution of the questionnaire, that was used to gain knowledge about the business processes, is explained.

4.1 Steps

In this section we describe the individual steps of the process analysis. The process started with the familiarization of several topics, followed by an initial gathering of information about the company and its employees and structure. We then created a questionnaire in multiple iterations and further improved it by rehearsing it in an interview with few employees. Afterwards, we transferred the questionnaire into an online survey in order to interrogate more employees of the adesso AG. Then, a generic performance test process was defined. The gathered information were analyzed and some improvements were recommended to transform the current processes to the generic process. Figure 4.1 describes the project progression. We refer to Eduard Tudenhöfner, who supervised this project at adesso AG, as our supervisor.

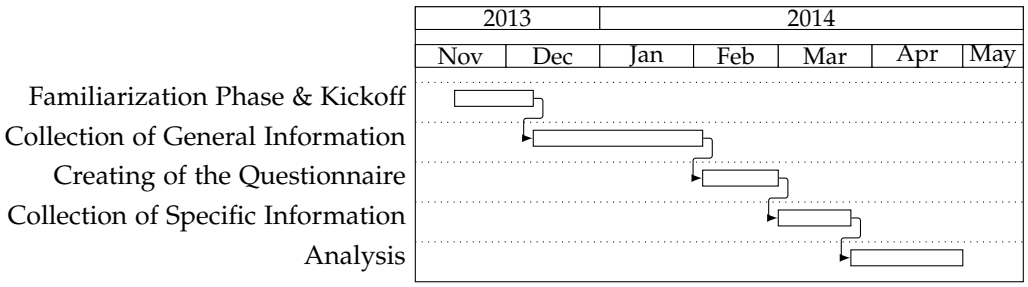

Figure 4.1. The Gantt project progression chart of the process analysis. It presents steps token and its chronological sequences.

4.1.1 Collection of Information and First Meeting

One of the first steps of the process analysis is to collect general information about the company itself and was performed from mid-December to the beginning of February. Besides gathering information by using the company's website, we arranged a meeting with our supervisor, a senior software engineer at the adesso AG, in order to conduct an interview. The result of the interview consists of information about the company and its structure, its employees and possible participants for our surveys. Further, our supervisor provided us his master thesis about a generic performance management model, which we considered in the following steps.

4.1.2 Iterations of Creating the Questionnaire

We designed the first draft of the questionnaire in February by using the aforementioned information. A detailed explanation of the process is described in section 4.2. After multiple iterations of proofreading and correction together with our supervisor, we sent the revised version to our supervisor for further improvements. We arranged another meeting, in which we received both improvements and additional information about the performance testing processes in the company. Using these information, we finally constructed a version of the questionnaire, of which we think it is ready to be used in an interview.

We asked five employees of the adesso AG for an interview over telephone or Skype. In those interviews, we both collected the answers and the reactions of the participants to our questions. We used these information in order to either improve, remove, or add new questions to the questionnaire.

4.1.3 Collecting Information Using an Online Survey

In March, we transferred the final draft of the questionnaire to an online survey service[1]. Using the URL, we distributed it over the mailing list for developers at the adesso AG. The survey phase took the whole February.

The mailing list is targeted to over 200 adesso AG employees at different locations in Germany who are part of a software development team. Potential participants had 20 days to fill out our survey. In the end we had 17 participants with different working focus.

4.1.4 Analysis of the Information

We spent the April to analyze the completed survey and figured out some weaknesses in the performance testing habits and found potential for improvement. Therefor, we analyzed each question separately and correlated different aspects, like the criticality of a project, to others.

[1] Google Forms. http://www.google.com/google-d-s/createforms.html

Further, we built a generic process that can be used as a best-practice example for performance-testing. The process consists of four phases, including *Plan & Design*, *Preparation*, *Execution*, and *Analysis*, and is presented in Section 5.2. This process provides the basis for our improvement recommendations.

Finally, we presented our findings in front of employees of adesso AG by discussing the survey results, the generic process, and our recommendations.

4.2 Questionnaire

The design of the questionnaire toke several revisions to get to the final version. Two of these revision are considered to be the important ones, because they were used to gain actual information from persons consulted. There also were two steps in which information was retrieved. The first version of the questionnaire was used to get general information about adesso AG's structure and their working methods (Section 4.1.1 and Section 4.1.2). It was targeted on questions regarding the organizational and hierarchical structure, general development processes, testing guidelines required by adesso AG and general roles. Findings of this questionnaire were used to improve the questionnaire itself. The second version of the questionnaire contains improvements derived from the first one (Section 4.1.3). In the second version the focus laid on the actual performance-testing process. Therefore the questions were targeted to concrete development and testing processes of specific projects, including questions about working parts that should provide good development processes and result in great products. The questionnaires can be found in the appendix. The first questionnaire in Appendix A and the second in Appendix B.

Both questionnaires were divided into sub-parts to structure the questions used. In the first version the parts *Person & Company*, *Projects & Processes*, and *Testing* were included. In the second version, *Person & Company* was excluded because general information about adesso AG were collected during the interviews with the first questionnaire and those information are similar for each person working at adesso AG. In the following those three sub parts are described.

- **Person & Company** This section contains questions regarding general information about adesso AG. It is divided in three subparts. There are questions about *Positions and Roles* at adesso AG and in their development processes. Followed by a section about *employee training* and *employee motivation* incentives.

- **Projects & Processes** The second part asks questions about projects and their general processes at adesso AG. The answers to this questions should refer to a specific project which the interviewee could choose by himself. This part start with questions about the *Type of the Project*, including size, involved employees, and main focus. After that the *Structure of the Project* is evaluated, that means roles and participants. Followed by questions about the *Development Process* and the *Interviewee's Role* in the the project.

- **Testing** The last section touches the concrete testing habits in that project from *Projects & Processes*. The first few question focused on the *Importance of (Performance-)Testing* in the project. After that the interviewee's *Knowledge* about (performance-)testing is evaluated. The next sub-part treats with *Tools* used and support the testing process. *Documentation* is not only used to evaluate the creation of documentation artifacts but also how (performance-)requirements are collected and handled. The questionnaire ends with questions about the *Test Processes* and how test *Results are Communicated* and how they influence the project.

The first questionnaire was used on five adesso AG employees conducted on phone calls. While the second version of the questionnaire was completed in an online survey by 17 employees.

Chapter 5

Results

This chapter describes the results of our process analysis. Most of the information and data given here is based on the answers we got in the interviews and through our questionnaire. Section 5.1 describes and analyzes the situation we found at adesso AG. This results are used to develop a general process for performance testing. This process is described in Section 5.2.1. In the last Section, 5.2.2, we show improvements that we have identified and developed.

5.1 Current State

This section evaluates the findings of the performed survey. First Section 5.1.1 gives an overview of the general state at adesso AG. Afterwards Section 5.1.2 evaluates the answers of the survey in a structured way.

The performed survey consisted of interviews with five participants and an online questionnaire with 17 participants from adesso AG and their partners. All participants were somehow involved in performance-testing, but on different development phases and levels.

5.1.1 General State

The statements in this section are based on the answers of our interview and questionnaire participants. We condensed information from the answers that give a short overview over the situation at adesso AG. Information that is too detailed and is not helpful to describe the general state is left for the following analysis section.

As stated in Section 3.1, adesso AG supports projects developed at other companies. Between 10 and 50 employees are working in total on the projects described by our participants, although some projects have more than 100 employees. From these employees, 5 to 30 are developers and 5 to 10 are employees of adesso AG. More than 70% of the projects reported by the participants are large projects with more than 2,000,000 lines of code.

More than 80% of the participants describe their project as performance critical, yet this result are likely to be influenced by our analysis' setting (since we mainly addressed participants taking part in performance-critical projects). In many projects, safety and secu-

rity are in favor concerning performance, but the other non-functional requirements (e. g., usability or reliability) are weighted about the same as performance by the questionnaire participants. However, many of the interviewees stated that other non-functional requirements, especially maintainability and availability, are more important than performance. This contrasts with the criticality rating, since one would expect that in performance-critical projects performance is at least equally important than other non-functional requirements.

The participants mentioned different problems concerning the performance management in their project. However, we got a consensus that nearly each project has unclear, imprecise, or unfulfillable performance requirements. This raises the question, why experiences made in older projects do not effect the planning and decisions in newly started projects. The reason for this could be twofold: Maybe, there are only little exchanges of experience or experiences are not recorded properly and are lost between projects.

In nearly all described projects, an agile development project is used. Here, Scrum was mentioned the most. Nearly 90% said, that they have pre-defined roles in their project. In the roles enumerated by the participants, we found that only about half of them mentioned testing-roles. One participant even stated that there is no quality assurance in the project at all. Due to the fact that only a few roles where added to the projects later on, the projects seem to be well-planed.

In about 59% of the described projects, the application is typically performance-tested right before it is shipped to the customer. The developer tests the application for performance after major changes in 35% of the projects and in 29% after new components are finished. In only two projects (ca. 12%) performance is tested during regression tests. The degree of recording performance test results is quite mixed: 31% never to seldom record results and 44% record results mostly or always. The performance tests tend to be manually as 58% of the participants said that test are done completely or mostly manually whereas 35% said that their tests are nearly automatized.

The participants' knowledge regarding performance testing tends to the average. However, more than 80% said that they have enhanced their knowledge already at least moderately to greatly. Knowledge is mainly shared during meetings or personal contacts, while technical platforms are seldom used. Experts that concentrate on performance tests are rarely found in the participants' projects, so conclusions in relation to experts are vague.

Most participants mentioned, that performance problems have arisen after the application has been taken into operation at about half of all projects with tendency to less.

5.1.2 Questionnaire Analysis

In this section the results of the questionnaire phase are described and findings are summarized. The conducted data is structured like the questionnaire.

Exchange The first part deals with information exchange between employees and their supervisors. The results of the survey show that most employees exchange project-related information with other project-involved persons daily. Also, employees communicate with their supervisors at least two times a week. With colleagues of other projects, which also could provide valuable advices, just for 63% there is an exchange at least once a week. This seems to be a good basis to communicate common problems and good solutions within adesso AG. The result of the survey also provides some information about how helpful this exchange with colleagues is. While the communication with colleagues of the same project seems to be very helpful, the exchange with members of other projects were rated as helpful. In contrast, the exchange with supervisors is not that good. Only 47% of the survey's participants classify the exchange with their project supervisors as helpful or better, while the communication with their hierarchical supervisor is more likely graded as unhelpful (only 26% checked helpful or better). This leads to the conclusion that while small and direct information between colleagues is exchanged well, general and sometimes problematic information cannot be communicated in an appropriate way. While it is obvious that the communication has potential to be improved, this improvement is rather not desired by the participants of the survey. 65% prefer the communication to be scheduled infrequently. That implies that the communication is too complex (e.g. the communication channel is too long).

Performance-Testing Even though 88% of the projects are rated from three or higher on a criticality in relation of performance scale from one to five only 12% have experts for performance-testing. Also most participants grade their knowledge as average. But for many participants it was possible to improve their knowledge by working on projects.

Development Process The way the participants analyze performance-related aspects (question 17-19 in the second questionnaire revision) indicates, that many projects did not consider performance-related requirements and aspects in earlier phases (e.g. analysis phase). Instead, the consideration of performance-related aspects starts primarily in the design phase of the software. Since this may already be too late, the consequences show up in the already mentioned problems.

Further, performance-related tests were done exclusively before releasing. In order to have more control in fulfilling performance-related requirements, one should try to conduct performance tests more often. For example, performance tests or validations could be done automatically after each revision.

To simplify or at least enable automatic testing, one needs an appropriate test environment. In most cases, an efficient way to set up a test environment, is to adapt an already available template or the test environment of an past project. Evaluating the answers to question 18 (in the second questionnaire revision) shows us, that this is not the case at the moment.

Individuals in the Project Since the comprehensibility of the artifacts are mostly not rated that great (31%-38% in a 100% scale) by our participants, there is potential for improvement. One possible way to improve the comprehensibility is to stick to standards and use given notations that everybody in the project uses. Hence, we can avoid misunderstandings and reduce the time for familiarization, since every individual already has some kind of mental model of the artifact.

Tools for Performance Testing Analyzing the questions about tools for performance testing, we found out, that more than half of the participants were not provided with tools. We think that providing and standardizing specific performance testing tools should ease gaining Know-how and exchanging experiences. This would not be possible, if each individual uses his preferred or even own tools. Further, one can ensure that all kind of tests were covered by providing appropriate tools.

In addition, test results are not documented and stored, which is also a consequence of not providing tools that provide this possibility. Participants stated, that in over 50% of their past projects only a mediocre or even less documentation of test results were made.

Documentation in Projects Typically, as stated by about 50% of the participants, performance requirements are gathered through interviews. However, 21% select none of the available option. Hence we assume there is no general procedure for gathering performance requirements. If we separate the gathering methods in method that do include the customer and those that do not, the customer participates in the performance requirement gathering process in 64% of the participant's projects. Though in the remaining 36% the requirements are deduced by the developers themselves.

In 75% of the answers, at least performance-critical use cases are enhanced with performance requirements. When comparing whether performance requirements are prioritized with the way the performance management is perceived by the participants, we find no correlation in about 50 %. Thus, we guess there is no relation between prioritization and bad results.

Financial aspects seem to have a high impact on the projects, as performance requirements are often checked for their costs and feasibility. In most cases, the participants inspect the performance requirements through reviews (88%), yet other additional checks are also performed.

If projects have missing performance requirements, the customer is asked in 77% of the cases. In 46%, experiences from earlier projects are used to add missing requirements. This shows that contacting the customer is preferred in most cases and experiences or standards are used as a fallback.

Electronic platforms for communicating performance topics are rarely used or do not exist at all. 50% and 75% mention that there is no wiki and forum, respectively, installed. On the other hand, personal communication is used more often: About 75% change performance knowledge at least seldom during meetings and more than 80%

change knowledge in personal discussions. Although internal trainings and conferences are provided by adesso AG, 50% of the participants say that there is no conference where knowledge regarding performance is exchanged.

In 29% of the projects, our participants describe, response time requirements are not taken into account. If they are, response times are categorized in 14% of the described projects and value ranges or exact values are used in 57%. Workload requirements are taken into account less often (in about 54%). In case they are taken into account, value ranges or exact values are preferred. For concurrency requirements, there is a gap between not taken into account and value ranges: Either the requirements are not taken into account or value ranges or exact values are used, but no categorization is done. The way availability requirements are take into account is distributed nearly equally. Not taken into account, categorization and exact value are selected by about 23% each, whereas value ranges are used in about 31% of the projects. Hardware and network requirements are not taken into account in more than 50%. If they are, value ranges are in favor (more than 35%). Thus we can conclude that, among all mentioned performance requirements, response time and availability is relevant, whereas workload is less relevant. All other requirements are only relevant for a small subset of the described projects.

Testing behavior and types of tests Our participants stated, that the average effort to conduct performance testing is mostly at about 0-10% of the whole project effort. Some minority of the participants stated, that the effort for performance testing is at about 20-30%.

Analyzing the performance testing behavior, we found out, that the majority (58%) of the participants conduct manual tests. The remaining participants conduct tests partially automatically. Since automatic testing could be conducted with lower effort, one should try to set up a test environment, which allows automatic testing of performance attributes.

We asked our participants about the different kind of tests they conducted in their projects. The result is shown in figure 5.1. As we can see, baseline tests and load tests are conducted the most with 59% and about 47%, respectively. The reason may be, that those two kind of tests are the basic ones and covers most of the usage scenarios, while the remaining tests (e.g. stability test, smoke test or isolation test) are more specific. The remaining tests are conducted by far less participants, mostly at about 12-29%. The analysis of the types (accordingly number) of tests conducted and the performance criticality of the project in the form of a correlation, shows that the more critical the project is, the more types of tests are conducted.

Feedback The questionnaire points out that in more than 50% of the projects bad testing results do not have influence on the testing schedule or employee assignment. Likewise, bad testing results have less influence on the project budget (influence in just 40%) and the development toolchain (influence in just 13%). 27% of the participants state that nothing is influenced by insufficient test results. However, Figure 5.2 shows a correlation

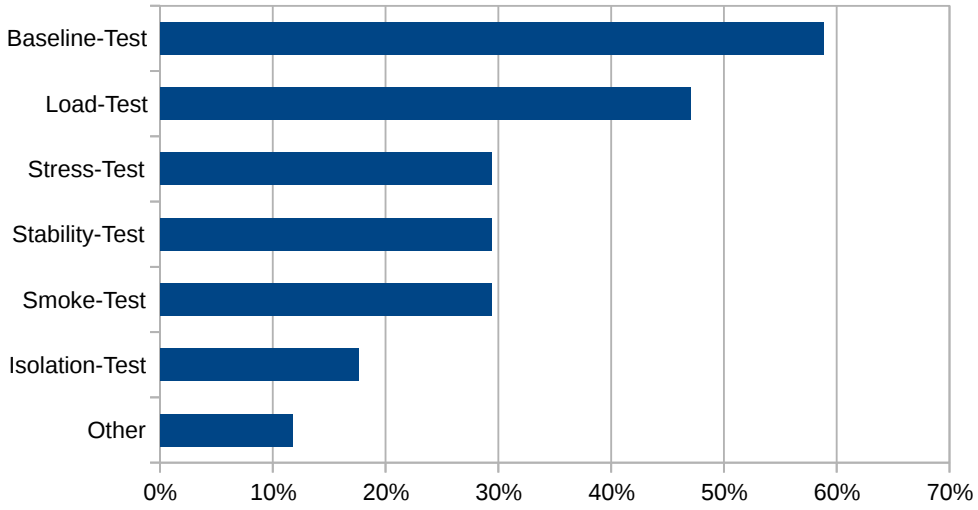

Figure 5.1. Test strategies used by the participants. The y-axis shows the name of the test strategies. The x-axis shows the percentage of the participants, that used the correspondent test strategy.

with a coefficient of 0.4735 between the criticality of a project and the effort used to plan performance tests. When there are bad test results, in 80% of all cases the tested artifact will be improved. To summarize, there is too less organizational influence of bad performance test results on projects.

There is also a slight correlation between the performance-criticality of a project and its influence on further projects. This correlation coefficient is 0.3698 and is pictured in Figure 5.3. In general there hardly is an influence on further projects (86% consider the influence as nonexistent to moderate). So, experiences are not transferred to new projects and there is no learning effect.

Results gathered by performance-tests are presented to the customer in 79% of the cases. After that, the customer can react on and has influences regarding to these results in 69% of the projects. The developers get information about affected components, requirements and working packages of test results. This implies that developers get enough information to further work on the tested artifacts. The question asking for performance problems after commissioning of the developed software results in a normal distribution (pictured in Figure 5.4).

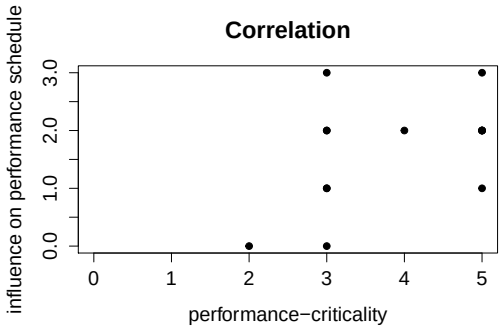

Figure 5.2. Presents the correlation between the criticality of a project and the planning of performance tests. Correlation coefficient: 0.4735. Criticality: 1 = not critical, 5 = very critical; Influence: 0 = no influence; 5 = much influence

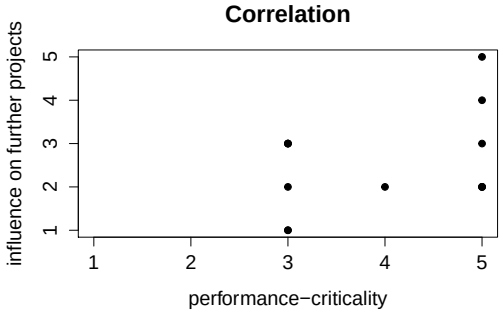

Figure 5.3. Presents the correlation between the criticality of a project and its influence on further projects. Correlation coefficient: 0.3698. Criticality: 1 = not critical, 5 = very critical; Influence: 1 = no influence; 5 = much influence

5.2 Generic Process

The second result of this analysis is a generic process for performance testing. This section starts with the definition and the presentation of this the generic process (Section 5.2.1) and ends with recommendations (Section 5.2.2) for adesso AG to apply performance testing based on the generic process.

5.2.1 Definition of the Generic Process

The presented generic process is based on the work of Eduard Tudenhöfner [Tudenhöfner 2011] and influenced by Ian Molyneaux [Molyneaux 2009]. Tudenhöfner presents a general

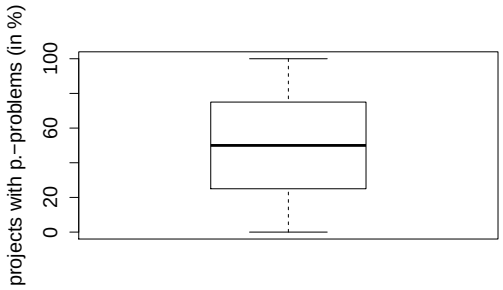

Figure 5.4. Presents the frequency by which projects have performance problems after commissioning.

process for performance management that can be adapted to the development process used in a project. In contrast Molyneaux concentrates on the performance testing itself and how these tests should be performed. There are four phases that have to be performed to use this process. In the first phase planning and designing is performed. Then some preparation were made in second phase. Phase three handles the execution of the tests and finally phase four includes the analysis of the results recorded in phase 3.

The remainder of this section describes the four phases of the process in detail.

Phase 1—Plan and Design The generic performance testing process starts with planning and designing the test. In this phase, critical use cases and scenarios were identified and the user behavior has to be understood and specified. For those, we want to define the acceptance criteria. For example, we want to define the conditions for the performance tests and several performance requirements, which has to be fulfilled by the software. Further, we want to select the test strategies, which will be used to evaluate the acceptance criteria.

In order to fulfill the acceptance criteria, a test plan has to be scheduled. The test plan contains amongst others the selected test strategies and the phases of the project, in which performance tests have to be conducted. The phases of the project, in which performance tests should be conducted, are for example the integration phase or after every revision.

Phase 2—Preparation The second phase focuses on preparing the test execution. Therefore, this phase is composed by three main task: To be able to run the performance tests properly and satisfactory, a test environment has to be built, tools that will be used to run the performance tests and are compatible with the test environment have to be selected, and each test case has to be scripted in a test transaction.

While building the test environment, several issues have to be considered. For the performance tests to be representable, the test environment has to be as close to the production environment as possible and based on the same components. Beside the same

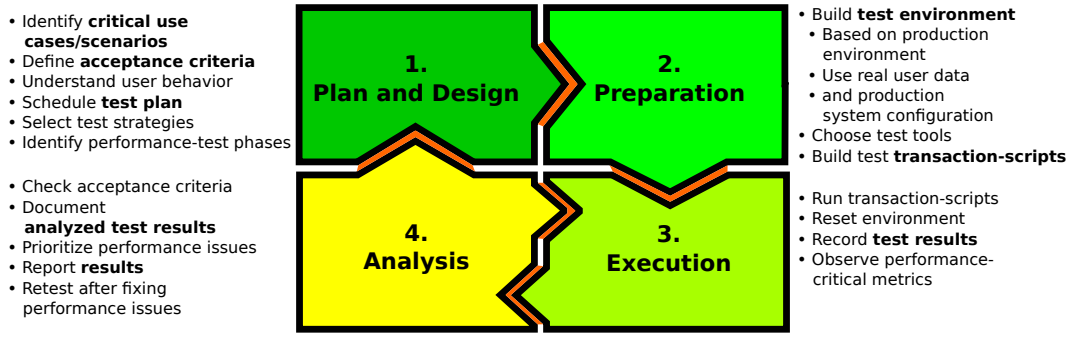

Figure 5.5. The Generic process consists of four phases. Each phase includes some actions and (bold marked) results. These results are the basis for the following phase. After the fourth phase tests can be rerun and experiences can be passed to the first phase.

execution environment, like database systems or application containers, this also includes the infrastructure as a whole (e. g., number of servers and their hardware). Also, the test environment's configuration has to be similar to the target environment. To be able to design test scripts properly and to get representative results, representative user data and behavior models are needed as well to populate databases with realistic data and to script the transaction realistically (e. g., viable input data and execution steps).

Phase 3—Execution In the third phase we execute our prepared test. This includes running the transaction scripts from phase two in a reset environment in order to be reproducible and to prevent the result of the tests to be biased. It is also important to record all results because they are the basis for the fourth phase and enable us to compare the results of further executions with the current. To achieve this comparability it is important to use the same environment on each test execution.

It is good to observe performance-critical metrics on the test system to identify high demanded hardware components. For example, the load of the CPU to discover differences in load distribution between different kernels.

Phase 4—Analysis The last phase deals with the analysis of the results recorded in phase three. The first thing to do is to check if the acceptance criteria is met and in case it is not, mark the test case as failed. These test results need be documented to ensure comparability to retests or tests of further software versions. In cases where there are many performance issues, these issues need to be prioritized to ensure that more critical ones are dealt with first. Then, these results need to be reported.

After performance issues were fixed we need to step back to the third phase and run the tests again, to make sure all criteria are met now. It is a good practice to give feedback to the first phase to improve the planning and designing of performance tests.

5.2.2 Recommendations Regarding the Generic Process

Phase 1 There are some problems regarding performance requirements in the current state. Many participants stated, that performance requirements are unclear or even unfulfillable. Based on this, we recommend to carry experiences of past projects over to the current project and use them to refine and negotiate performance requirements. Further, in order to refine and revise performance requirements, they need to be involved in the project as soon as possible. At the moment, performance requirements are not involved until the design phase of the project. According to the generic process, performance requirements should be involved as early as possible (e.g. in the analysis phase).

Further, our participants stated, that performance-testing is conducted mostly right before it is shipped to the customer in about 56% of the described projects. In this case, we recommend to automate the performance-testing process and conduct it after each phase. Additionally, the project team should always try to involve the customer in every phase of the software development project. Thus, it is possible to get feedback from the customer and consider it as soon as possible.

Regarding the exchange of information and experiences, we recommend to improve the communication between the employees and their supervisors and try to motivate the communication between them. The reason for this is, that most of the participants stated, that a communication between colleagues is done more often than the communication with the supervisors. Further, the analysis of the questionnaire results shows, that employees perceive the communication with colleagues as more helpful than with the supervisors. However, since the supervisor has a better overview of the project and therefore is in a better position to spot erroneous activities or even the lack of important activities, it is definitely worth to improve this communication path.

By improving the communication with the supervisors, it is also easier for the supervisors to bring in experiences of past projects and therefore avoid repeating errors.

Phase 2 Section 5.1.1 reveals that two-thirds of the tests are executed manually. To achieve a easier, more standardized, and better reproducible way of performing tests, each test should be executed completely automatic. This is provided by the use of transaction-scripts.

The Paragraph 5.1.2 states that mostly standard types of tests, like baseline or load test, are performed. In particular on performance critical projects some further types (see 2.3.3) of test should be used.

Phase 4 General improvement and experiences figured out in Phase 4 should be passed on to Phase 1. This would provide a continuous improvement of the whole test process.

General As described in paragraph *Exchange* in Section 5.1.2, experienced made in one project are not sufficiently transferred to other projects. To improve the amount of experience exchanged between projects, the knowledge exchange between project members

and leaders should be eased and members should be motivated to communicate their experiences and knowledge.

Section 5.1.1 and paragraph *Performance-Testing* in Section 5.1.2 shows that the project participants have only an average knowledge regarding performance testing. Thus, this knowledge should be improved by training courses focusing on performance testing. Experts having a high knowledge level of performance testing should take part in performance-critical projects. Such experts support other members in all project phase that handle with performance, like requirement analysis, testing, and monitoring.

Paragraph *Individuals in the Project* mentions a lack of comprehensibility in artifacts, thus the overall quality of artifacts should be improved. All project member shall understand the artifacts, even though they require different information from the artifacts, or have different levels of knowledge or experience. Therefor, it might be appropriate to take this consideration into account when artifacts are quality assured.

Tools used for performance testing differ between the projects as stated in paragraph *Tools for Performance Testing*, even if projects are similar. Hence, we recommend to provide a set of test tools sufficient for most projects. This should be possible, since most performance testing tools, such as load generators, are by nature independent of the language used to develop the target application or the platform it runs on. When such a tool-set is provided, project participants should be encouraged to use these recommended tools. Tools that are selected and added to the set should support a complete, and ideally automatically, documentation of test results.

In paragraph *Documentation in Projects* we see that electronic documentation platforms are rarely used. However, personal communication is not sufficient to document knowledge centralized and durable. Therefore, a project-wide electronic platform should be established and used to document knowledge. This will ease the exchange of experiences and give members from different projects a source of information and knowledge they can reference.

As stated in paragraph *Feedback*, further project planning is often not influenced by bad performance test results. To be sure performance requirements are met, it should be possible to adjust a project's planning and the focus when insufficient performance is detected. Similarly, the project's criticality should influence its planning, like time reserved for performance testing and association of project members to roles.

Chapter 6

Threats to Validity

The results of this work, which includes the analysis results and recommendations for improvement, are based on several surveys. Therefore, the validity is amongst others dependent on the correct usage of the measurement tools, which in this case are telephone interviews and an online questionnaire. Further, a significant analysis is also dependent on the number of participants, which should be as high as possible, and on the participants to answer as objective as possible. In addition, a significant analysis should always consider noises and other sources of falsification.

In our survey, we had 17 participants in the questionnaire and four participants in the interviews, which results in a relatively low representativeness of the analysis result. As the questionnaire addressed performance testing, it was answered mainly by participants taking part in performance-critical projects. As our supervisor mentioned, some potential participants did not answer the questionnaire since they took part in a less performance-critical project and felt their experiences are not relevant for our analysis. Hence, our results can only be applied to a small subset of performance critical projects at adesso AG.

Further, participants who answered the questionnaire may have different backgrounds. Some of them might be developers, others might be project leaders. Accordingly, the expertise as well as the degree of overview about the project might differ amongst the participants. Together with a relatively low number of participants, this may aggregate to results, that are not significant at all.

We ourselves have no practical experiences in performance testing. For us, it was therefore hard to draw the right conclusions from all answers and to predict whether our results, especially the general process and the improvements, will help to reduce the identified problems in practice. Additionally, there was sparse number of literature available that focus on performance testing processes. These were the master thesis written by Eduard Tudenhöfner [Tudenhöfner 2011] and two books [Meier et al. 2007; Molyneaux 2009].

Regarding the given answers, not all questions of the questionnaire were answered completely. Some of our participants tended not to answer to the last parts of the questionnaire. Our speculation is, that the questionnaire is too long, which may result in abortions or in fatigue, where some questions were simply ignored by the participant. Since the first question blocks receive more answers than the last ones, the answers to the last question blocks may have a lower significance than the first ones. Accordingly, this influences our analysis of the correlational relationships between those answers.

In addition, we cannot guarantee, that our participants are always answering objectively and honestly. At this point, we refer to Section 2.2, which explains effects like the social desirability or non-opinions.

Summing all up, we come to the conclusion, that using solely surveys may not be the most effective way to get an comprehensive overview of the performance-testing process. However, the possibilities for information retrieval was constrained by time and scope of our work. A possible way to continue this work in the future is therefore to conduct a field survey and even be a part of a project team in order to get an objective and comprehensive overview of the process.

Chapter 7

Conclusion

In this process analysis, we analyzed the performance testing process performed at adesso AG. We therefor did interviews and built a questionnaire where we asked adesso AG employees taking part in performance critical project. Based on the answers we got an overview over the strength and weaknesses in development projects at adesso AG.

Most projects described by the questionnaire participants were large project with more than two million lines of code. Although projects typically have between 10 to 50 members, some participants said that their project has more than 100 members. The participants often stated that missing or unclear requirements are often the cause for problems in the performance management of projects. This led us to the question, why these experiences do not influence decisions and planning in future projects.

Project members stated that they often exchange knowledge with other colleagues in the same or other projects through personal discussions. However, communication with leaders is less often performed and rated less helpful. Beside these personal communications, project members often do not use electronic platforms to exchange knowledge between projects. Hence, we suggested to motivate the project members to communicate their experiences more often with project leader and to use electronic platforms to keep knowledge in a durable and project wide manner.

Experts for performance testing were often not mentioned by the questionnaire participants. For this reason, experts should be included into the project to support other project members in performance critical phase, like (performance) requirement elicitation. This experts might also help to reduce the risk of performance requirements to be considered to late.

By analyzing the way performance tests are performed, we found out that in most projects enough effort is spent for performance testing. However, the tests' quality is not as good as it could be. Since some participants mentioned performance testing to be done at least partially manually, we suggested to automatize more tests.

In summary, we found strengths in the processes used at adesso AG, but also some weaknesses. To reduce these and other weaknesses, we developed a general process based on Tudenhöfner [Tudenhöfner 2011] and Molyneaux [Molyneaux 2009] for performance testing. This process covers four phases and can be adapted to the different development processes used at adesso AG. Each phase is described in detail, including its required tasks and artifacts.

Based on general process, we also presented several recommendations. These recommendations address the weaknesses identified throughout the questionnaire analysis. Some of the recommendations are of a general nature, while others are bound to one of the four phases of the general process.

Appendix A

Questionnaire - Version 1

0.1 Was verstehen Sie unter Performance?

1 Person & Firma

Positionen und Rollen

1.1 Wie sieht Ihre Geschäftshierarchie ungefähr aus? Bitte beschreiben Sie diese Hierarchie oder skizzieren sie diese in der folgenden Box:

1.2 Haben Mitarbeiter mehrere Rollen in der Softwareentwicklung?

☐ Ja ☐ Nein

1.3 Gibt es vordefinierte Rollen?

☐ Ja ☐ Nein

1.4 Falls ja: Welche wären das?

Weiterbildung der Mitarbeiter

1.5 Werden bei Ihnen Schulungen durchgeführt?

☐ Ja ☐ Nein

Bitte beantworten Sie die folgenden Fragen 1.6, 1.7 und 1.8 nur, wenn Sie die letzte Frage mit *Ja* beantwortet haben.

1.6 Wie oft findet sowas statt?

☐ vierteljährlich ☐ halbjährlich ☐ jährlich ☐ _____

1.7 Sind diese Schulungen Pflicht?

☐ Ja ☐ Nein

1.8 Wer trifft Entscheidungen bezüglich der Schulungen?

☐ Teamleiter ☐ Geschäftsführer ☐ _____

1.9 Erfahrungsaustausch. Wie werden bei Ihnen Erfahrungen ausgetauscht? (Mehrere Antworten möglich)

☐ Plattform ☐ Meetings oder Präsentationen
☐ Dokumentationen ☐ Es werden keine Erfahrungen ausgetauscht
☐ _____

1.10 Falls Erfahrungen ausgetauscht werden: Wie häufig werden diese Optionen genutzt?

☐ Sehr selten ☐ selten ☐ Häufig ☐ Sehr häufig

Motivation allgemein und im Speziellen

1.11 Werden Mitarbeiter für überdurchschnittliche Leistungen belohnt, um diese zu motivieren?

☐ Ja ☐ Nein

1.12 Falls ja: Wie?

☐ Finanziellen Anreize ☐ Karriereaufstieg ☐ Auszeichnungen

1.13 Gibt es Guidelines und Standards für Projekte, die eingehalten werden müssen? (bzgl. Dokumentieren, Qualitätssicherung)

☐ Ja ☐ Nein

2 Projekte & Prozesse

Art des Projekts
Die folgenden Fragen beziehen sich auf Ihr aktuelles Projekt.

2.1 Wie viele Mitarbeiter sind im Gesamten am Projekt beteiligt?

2.2 Wie viele Mitarbeiter der Adesso AG sind am Projekt beteiligt?

2.3 Wie viele KLOC umfasst das Projekt?

☐ 100 ☐ 500 ☐ 1000 ☐ 2000 ☐ _____

2.4 Bei der entwickelten Software wird Performance gegenüber folgenden Eigenschaften bevorzugt?

	vollkommen bevorzugt	bevorzugt	gleichwertig	vernachlässigt	stark vernachlässigt
Security	☐	☐	☐	☐	☐
Safety	☐	☐	☐	☐	☐
Usabilty	☐	☐	☐	☐	☐
Reliability	☐	☐	☐	☐	☐
Functionality	☐	☐	☐	☐	☐
Maintainability	☐	☐	☐	☐	☐
Availability	☐	☐	☐	☐	☐

Struktur im Projekt

2.5 Wie sieht die personelle Struktur in Ihrem Projekt aus (Rollen, Hierarchie, Zuständigkeiten, Richtlinien)? Ergänzend können Sie die Struktur skizzieren.

2.6 Wie häufig tauschen Sie sich mit Ihren Arbeitskollegen aus?

	täglich	zweitäglich	wöchentlich	monatlich	seltener
Mitglieder aus gleichem Projekt	☐	☐	☐	☐	☐
Kollegen mit gleicher Aufgabe	☐	☐	☐	☐	☐
Vorgesetzten	☐	☐	☐	☐	☐
Allgemein Kollegen aus anderen Projekten	☐	☐	☐	☐	☐
Mitarbeitern anderer Firmen	☐	☐	☐	☐	☐

Huy Viet Le, Patrick Strobel, Kevin Wenz
Prozessanalyse: Performance-Testing

2.7 Wie hilfreich empfinden Sie den Austausch zu den genannten Personen?

	nicht hilfreich	kaum hilfreich	ok	hilfreich	sehr hilfreich
Mitglieder aus gleichem Projekt	☐	☐	☐	☐	☐
Kollegen mit gleicher Aufgabe	☐	☐	☐	☐	☐
Vorgesetzten	☐	☐	☐	☐	☐
Allgemein Kollegen aus anderen Projekten	☐	☐	☐	☐	☐
Mitarbeitern anderer Firmen	☐	☐	☐	☐	☐

2.8 Würden Sie mehr Möglichkeiten für die Kommunikation befürworten?

☐ nein ☐ eher nicht ☐ kein Bedarf ☐ ja ☐ auf jeden Fall

Entwicklungsprozess

2.9 In welchen Projektaufgaben beschäftigen Sie sich mit performance-relevanten Aufgaben?

☐ gar nicht ☐ wenig ☐ normal ☐ viel ☐ sehr viel

2.10 Richten Sie für jedes Projekt eine eigene Testumgebung ein?

☐ Ja
☐ Eine bestehende Testumgebung aus alten Projekten wird angepasst
☐ Die Testumgebung wird aus einer Vorlage angepasst
☐ Nein

2.11 Wann testen Sie Ihre Software auf Performance?

☐ Bei jeder neuen Version
☐ Bei Fertigstellung einer neuen Komponente
☐ Während dem Einsatz
☐ Vor der Auslieferung

Person im Projekt

2.12 Welche Tätigkeit führen Sie im Projekt durch?

☐ Anforderungsanalyse
☐ Anforderungsspezifikation
☐ Entwurf & Architektur
☐ Implementieren
☐ Integrieren
☐ Testen (nicht performance-spezifisch)
☐ Testen (performance-spezifisch)
☐ Andere: _____

2.13 *Eigene Schätzung:* Als wie wichtig empfinden Sie Ihre Tätigkeit für den Erfolg des Projektes?

☐ sehr unwichtig ☐ unwichtig ☐ neutral ☐ wichtig ☐ sehr wichtig

2.14 Welche Artifakte benötigen Sie für die Durchführung Ihrer Arbeit?

2.15 Sind Informationen, die Sie erhalten, klar und verständlich?

Verständlichkeit	unverständlich	☐	☐	☐	☐	☐	verständlich
Klarheit	unklar	☐	☐	☐	☐	☐	klar

2.16 Welche Artifakte erstellen Sie?

3 Testen

Stellenwert

3.1 Gibt es in Ihrem Projekt Experten, die sich auf das Performance-Testen konzentrieren?

☐ Ja ☐ Nein

3.2 Falls ja: An welchen Arbeiten sind diese Experten beteiligt? (Mehrere Antworten möglich)

☐ Anforderungsanalyse ☐ Testplanung ☐ Entwurf & Architektur
☐ Implementierung ☐ Test ☐ Monitoring & Wartung

3.3 Falls ja: Können diese Experten Performance-Tests anordnen?

☐ Ja ☐ Nein

3.4 Falls ja: Haben Testingenieure Einfluss auf die Zeitplanung (Meilensteine, Aufgabenpakete, Beteiligte verschieben)

☐ Ja ☐ Nein

3.5 Falls ja: Wie kann ein Testingenieur die Zeitplanung beeinflussen?

☐ Entscheidet selbst ☐ Hat ein Stimmrecht bei Entscheidungen ☐ Kann einen Antrag stellen

Kenntnisse

3.6 Wie schätzen Sie Ihre Wissen in Bezug auf Performance-Testen ein?

☐ Keines vorhanden ☐ Rudimentär ☐ Durchschnittlich
☐ Überdurchschnittlich ☐ Umfangreich

3.7 In wie Weit konnten Sie Ihr Wissen in Bezug auf Performance-Testen bereits durch praktische Erfahrungen ausbauen?

gar nicht ☐ ☐ ☐ ☐ ☐ sehr stark

Werkzeuge

3.8 Werden Werkzeuge für das Performance-Testen bereitgestellt?

☐ Ja ☐ Nein

3.9 Falls bereitgestellt: Welche Arten von Performancetests lassen sich mit diesen Werkzeugen durchführen? (Mehrere Antworten möglich)

☐ Last-Tests ☐ Stress-Tests ☐ Kapazitäts-Tests
☐ Andere: _____

3.10 Falls bereitgestellt: Ist der Einsatz der bereitgestellten Werkzeuge verpflichtend?

☐ Ja ☐ Nein

3.11 Dürfen Sie eigene Werkzeuge für das Performance-Testen verwenden?

☐ Ja ☐ Nein

3.12 In welchem Umfang werden Testergebnisse aufgezeichnet, so dass die Ergebnisse später jederzeit abgerufen werden können?

☐ Gar nicht ☐ Minimal ☐ Meistens ☐ Umfangreich ☐ Vollständig

Dokumentation

3.13 Wie werden Performance-Anforderungen erhoben? (Mehrere Antworten möglich)

☐ Auswertung von Dokumenten ☐ Interviews ☐ Fragebögen
☐ Use-Case Bestimmung ☐ Beobachtung ☐ Experimente
☐ Prototyping ☐ Andere: _____

3.14 Wenn Use-Cases bestimmt werden: Werden Use-Cases priorisiert?

☐ Ja ☐ Nein

3.15 Wie werden Use-Cases priorisiert? (Auswahl)

☐ Voting durch relevante Mitarbeiter ☐ Voting durch relevante Mitarbeiter und Kunden
☐ Expertenmeinung ☐ Andere: _____

3.16 Wenn Use-Cases bestimmt werden: Enthalten Use-Cases Angaben zur Performance? (Auswahl)

☐ Keine Angaben zur Performance ☐ Angaben bei Performancekritischen Use-Cases
☐ Jeder Use-Case enthält Angaben ☐ Jedes Szenario eines Use-Cases enthält Angaben
☐ Jedes Teilszenario enthält Angaben

3.17 Werden Performance-Anforderungen priorisiert?

☐ Ja ☐ Nein

3.18 Wie werden Performance-Anforderungen priorisiert? (Auswahl)

☐ Voting durch relevante Mitarbeiter ☐ Voting durch relevante Mitarbeiter und Kunden
☐ Expertenmeinung ☐ Andere: _____

3.19 Auf welche der nachfolgenden Aspekte werden Performance-Anforderungen geprüft? (Mehrere Antworten möglich)

☐ Vollständigkeit ☐ Konsistenz
☐ Machbarkeit ☐ Performanceeigenschaften
☐ Andere: _____

3.20 Wie werden Anforderungen geprüft? (Mehrere Antworten möglich)

☐ Review ☐ Walk-Through ☐ Andere: _____

3.21 Wird das Pflichtenheft (die endgültigen Anforderungen) vom Kunden unterschrieben?

☐ Ja ☐ Nein

3.22 Wer entscheidet, ob eine Anforderung performancekritisch ist?

☐ Kunde ☐ Entwickler
☐ Testingenieur ☐ Teamleiter
☐ Andere: _____

3.23 Wie wird mit fehlenden Performanceanforderungen umgegangen?

☐ Nachfrage beim Kunden ☐ Standardwerte werden genommen (Industrie, Best-Practice)
☐ Erfahrungswerte aus früheren Projekten ☐ Andere: _____

3.24 Gibt es eine Plattform, auf der sich Entwickler austauschen können? (Mehrere Antworten möglich)

☐ Wiki ☐ Forum
☐ Meetings, bei dem aktuelle Projekte vorgestellt werden ☐ Persönlich
☐ Andere: _____

3.25 Wenn Performanceanforderungen erhoben werden: Werden folgende Eigenschaften berücksichtigt?

	nicht berücksichtigt	Einordnung (z.B. schnell, sofort)	Bereichsangabe	exakter Wert
Antwortzeit	☐	☐	☐	☐
Workload	☐	☐	☐	☐
Zahl gleichzeitiger Zugriffe	☐	☐	☐	☐
Verfügbarkeit	☐	☐	☐	☐
Hardware-Ressourcen	☐	☐	☐	☐
Netzwerk-Ressourcen	☐	☐	☐	☐

Testabläufe

3.26 Wie hoch ist bei Ihnen der durchschnittliche Aufwand für das Performance-Testen (in Stunden pro Projekt)?

3.27 Werden Tests automatisch oder manuell Durchgeführt (Automatisch bedeutet hier, dass das Ausführen, Messen und eventuelles Finden von Ursachen von einer Test-Suite übernommen wird.)?

☐ Alles automatisch ☐ Größtenteils automatisch
☐ Sowohl automatisch als auch manuell ☐ Größtenteils manuell
☐ Alles manuell

3.28 Welche Arten von Performancetests werden bei Ihren Projekten durchgeführt?

☐ Last-Tests ☐ Stress-Tests ☐ Kapazitäts-Tests
☐ Andere: _____

3.29 Wann im Projektverlauf testen Sie entwickelte Systeme auf ihre Performance?

Rückkopplung

3.30 Welche Punkte werden im laufenden Projekt durch Performance-Testergebnisse beeinflusst?

	durch schlechte Ergebnisse	durch gute Ergebnisse
Zeitplanung	☐	☐
Zuordnung der Mitarbeit zu Tätigkeiten	☐	☐
Projektbudget	☐	☐
Werkzeugeinsatz	☐	☐
Andere: _____	☐	☐

3.31 Wie stark wird die weitere Planung im Projekt durch Performance-Testergebnisse beeinflusst?

	Kein Einfluss	Kleine Anpassungen	Teilweise Überarbeitung	Vollständige Überarbeitung
Durch schlechte Ergebnisse	☐	☐	☐	☐
Durch gute Ergebnisse	☐	☐	☐	☐

3.32 Wie stark wird die Planung zukünftiger Projekt durch Erfahrungen in Bezug auf Performance beeinflusst?

Durch schlechte Ergebnisse: kein Einfluss ☐ ☐ ☐ ☐ ☐ starker Einfluss
Durch gute Ergebnisse: kein Einfluss ☐ ☐ ☐ ☐ ☐ starker Einfluss

3.33 Werden Kunden über Performance-Testergebnisse informiert?

☐ Ja ☐ Nein

3.34 Falls ja: Können Kunden anschließend Einfluss auf die Performance-Anforderungen nehmen?

☐ Ja ☐ Nein

3.35 Welche Informationen erhalten Entwickler über die Performance-Testergebnisse?

☐ Betroffener Komponenten ☐ Betroffene Anforderungen ☐ Betroffene Arbeitspakete
☐ Grad der Verbesserung ☐ Andere: _____

3.36 Bei wievielen Projekten zeigen sich nach Inbetriebnahme Performance-Probleme?

bei keinen ☐ ☐ ☐ ☐ ☐ bei allen

Appendix B

Questionnaire - Version 2

Prozessanalyse: Performance-Testing

Das Testen einer Software gehört zu den wichtigsten Aktivitäten in der Softwareentwicklung. Tests, bei denen die Funktionalität einer Software überprüft wird, können dabei anhand der funktionalen Anforderungen, die an eine Software gestellt werden, relativ einfach abgeleitet werden. Nicht-funktionale Anforderungen hingegen sind oft nicht "greifbar" und werden deshalb häufig nicht ausreichend spezifiziert. Daher kann es passieren, dass diese vernachlässigt und insbesondere beim Testen ignoriert werden.
Während fehlende Funktionen beim Testen oder Vorführen schnell entdeckt werden können, treten nicht-funktionale Missstände – wie zu lange Antwortzeiten bei hoher Nutzlast -- erst im realen System beim Kunden auf.
Daher sollte die Spezifikation von nicht-funktionalen Anforderungen klar und testbar sein, was jedoch nur funktioniert, wenn zum Beispiel ein geeigneter Entwicklungsprozess und das nötige Budget vorliegen.

Um herauszufinden in welcher Form Performance-Tests bei der adesso AG und bei beteiligten Firmen gehandhabt werden und wo noch Verbesserungspotential liegt, führen wir im Rahmen eines studentischen Projekts eine Umfrage durch. Wir freuen uns, wenn Sie an dieser Umfrage teilnehmen und danken Ihnen recht herzlich.

Für die Bearbeitung des Fragebogens sollten sie 15 - 20 Minuten einrechnen.

Alle von Ihnen eingegebenen Daten werden anonymisiert und vertraulich behandelt.

Alle Fragen (wenn nicht anders formuliert) beziehen sich auf ein von Ihnen durchgeführtes Projekt, das Sie als Beispiel für gelungenes oder schlechtes Performance-Management sehen. Wählen Sie daher ein abgeschlossenes oder zur Zeit laufendes Projekt aus, das diesem Kriterium entspricht.

Wenn Sie nicht bei der adesso AG angestellt sind, ersetzen Sie "adesso AG" durch den Namen ihres Arbeitgebers.

Falls Sie Interesse an der Auswertung und dem daraus resultierenden Bericht über die Prozessanalyse des Performance-Testprozesses haben, hinterlassen Sie uns einfach bei der letzten Frage Ihre E-Mail-Adresse.

Projekte und Prozesse

1. **Ist das gewählte Projekt bereits abgeschlossen?**
   Markieren Sie nur ein Oval.
   - ◯ Ja
   - ◯ Nein

2. **Sofern Sie das Performance-Management als nicht komplett gelungen ansehen: Was sind Ihrer Ansicht nach die Problemfaktoren? (Stichpunkte)**

   .....
   .....
   .....
   .....
   .....

3. **Als wie performance-kritisch sehen Sie das von Ihnen gewählte Projekt an?**
   Markieren Sie nur ein Oval.

	1	2	3	4	5	
unkritisch	◯	◯	◯	◯	◯	sehr kritisch

4. **Wie viele Mitarbeiter sind im Gesamten (Adesso AG + weitere Firmen) am Projekt beteiligt?**

   .....

5. **Wie viele Personen sind davon Entwickler?**

   .....

6. **Wie viele Mitarbeiter der Adesso AG sind am Projekt beteiligt?**

   .....

7. **Wie viele KLOC umfasst das Projekt?**
   1 KLOC = Tausend Codezeilen
   Markieren Sie nur ein Oval.
   - ◯ < 100
   - ◯ 100
   - ◯ 500
   - ◯ 1000
   - ◯ 2000
   - ◯ > 2000

8. **Bei der entwickelten Software wird Performance gegenüber folgenden Eigenschaften bevorzugt?**
Security bedeutet, dass Angreifer der Software und ihren Daten keinen "Schaden" zufügen können. Safety bezieht sich darauf, dass die Software ihrer Umwelt keinen Schaden zufügen kann.
Markieren Sie nur ein Oval pro Zeile.

	stark vernachlässigt	vernachlässigt	gleichwertig	bevorzugt	vollkommen bevorzugt
Security	◯	◯	◯	◯	◯
Safety	◯	◯	◯	◯	◯
Usability	◯	◯	◯	◯	◯
Reliability	◯	◯	◯	◯	◯
Functionality	◯	◯	◯	◯	◯
Maintainability	◯	◯	◯	◯	◯
Availability	◯	◯	◯	◯	◯

Struktur im Projekt

9. **Welches Prozessmodell wurde in Ihrem Projekt verwendet?**

.....

.....

.....

.....

.....

10. **Gibt es in Ihrem Projekt vordefinierte Rollen?**
Markieren Sie nur ein Oval.

◯ Ja
◯ Nein

11. **Sofern es vordefinierte Rollen gab, zählen Sie diese bitte auf.**
Geben Sie die Rollen bitte Komma getrennt ein.

.....

.....

.....

.....

.....

12. **Welche Rollen haben sich im Projektverlauf gebildet?**
Geben Sie die Rollen bitte Komma getrennt ein.

.....

.....

.....

.....

.....

13. **Wie häufig tauschen Sie sich fachlich mit Ihren Arbeitskollegen aus?**
Markieren Sie nur ein Oval pro Zeile.

	täglich	zweitäglich	wöchentlich	zweiwöchentlich	monatlich	seltener
Mitgliedern aus dem gleichen Projekt	◯	◯	◯	◯	◯	◯
Kollegen mit gleicher Rolle	◯	◯	◯	◯	◯	◯
Linienvorgesetzten	◯	◯	◯	◯	◯	◯
Projektvorgesetzten	◯	◯	◯	◯	◯	◯
Allgemein mit Kollegen aus anderen Projketen	◯	◯	◯	◯	◯	◯
Mitarbeitern anderer Firmen	◯	◯	◯	◯	◯	◯

14. **Wie hilfreich empfinden Sie den fachlichen Austausch mit den genannten Personen?**
Markieren Sie nur ein Oval pro Zeile.

	nicht hilfreich	kaum hilfreich	ok	hilfreich	sehr hilfreich
Mitgliedern aus dem gleichen Projekt	◯	◯	◯	◯	◯
Kollegen mit gleicher Rolle	◯	◯	◯	◯	◯
Linienvorgesetzten	◯	◯	◯	◯	◯
Projektvorgesetzten	◯	◯	◯	◯	◯
Allgemein mit Kollegen aus anderen Projketen	◯	◯	◯	◯	◯
Mitarbeitern anderer Firmen	◯	◯	◯	◯	◯

15. **Würden Sie mehr Möglichkeiten für die Kommunikation befürworten?**
Markieren Sie nur ein Oval.

- ◯ nein
- ◯ eher nicht
- ◯ manchmal
- ◯ ja
- ◯ auf jeden Fall

16. **Falls Erfahrungen ausgetauscht werden: Wie häufig wird diese Option in Ihrem Team genutzt?**
Markieren Sie nur ein Oval.

- ◯ sehr selten
- ◯ selten
- ◯ manchmal
- ◯ häufig
- ◯ sehr häufig

Entwicklungsprozess

17. **In welchen Aufgaben beschäftigen Sie und Ihr Team sich mit performance-relevanten Aufgaben?**
Wählen Sie alle zutreffenden Antworten aus.

- ☐ Anforderungsanalyse
- ☐ Anforderungsspezifikation
- ☐ Entwurf & Architektur
- ☐ Implementieren
- ☐ Integrieren
- ☐ Testen (nicht performance-spezifisch)
- ☐ Testen (performance-spezifisch)
- ☐ Sonstiges: .....

18. **Richten Sie und Ihr Team für dieses Projekt eine eigene Testumgebung ein?**
Markieren Sie nur ein Oval.

- ◯ ja
- ◯ Eine bestehende Testumgebung wird aus alten Projekten übernommen
- ◯ Die Testumgebung wird aus einer Vorlage angepasst

19. **Wann testen Sie und Ihr Team ihr Softwareprojekt auf Performance?**
Wählen Sie alle zutreffenden Antworten aus.

- ☐ Bei jeder größeren Änderung
- ☐ Bei jeder Version
- ☐ Beim Regressionstest
- ☐ Bei der Fertigstellung einer neuen Komponente
- ☐ Vor der Auslieferung
- ☐ Während des Einsatzes

Person im Projekt

20. **Welche Tätigkeit führen Sie im Projekt durch?**
Wählen Sie alle zutreffenden Antworten aus.

- ☐ Anforderungsanalyse
- ☐ Performance Anforderungen erheben
- ☐ Anforderungsspezifikation
- ☐ Entwurf & Architektur
- ☐ Implementieren
- ☐ Integrieren
- ☐ Testen (nicht-performance-spezifisch)
- ☐ Testen (performance-spezifisch)
- ☐ Testkoordination
- ☐ Sonstiges: .....

21. **Eigene Schätzung: Als wie wichtig empfinden Sie die Tätigkeit Ihres Teams für den Erfolg des Projektes?**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
sehr unwichtig	◯	◯	◯	◯	◯	sehr wichtig

22. **Welche Artefakte benötigen Sie für die Durchführung Ihrer Arbeit?**
Artefakte = Dokumente und Daten die für eine Softwareentwicklung beötigt werden oder dabei entstehen. Geben Sie die Artefakte bitte Komma getrennt ein.

.....

.....

.....

.....

.....

23. **Sind die Artefakte, die Sie erhalten, inhaltlich klar?**
Klar bedeutet: alle benötigten Informationen sind enthalten
Markieren Sie nur ein Oval.

	1	2	3	4	5	
unklar	◯	◯	◯	◯	◯	klar

24. **Sind die Artefakte, die Sie erhalten, verständlich?**
Verständlich bedeutet: widerspruchsfrei, sprachlich verständlich
Markieren Sie nur ein Oval.

	1	2	3	4	5	
unverständlich	◯	◯	◯	◯	◯	verständlich

25. **Welche Artefakte erstellen Sie?**
Artefakte = Dokumente und Daten die für eine Softwareentwicklung beötigt werden oder dabei entstehen. Geben Sie die Artefakte bitte Komma getrennt ein.

.....

.....

.....

.....

.....

Testen: Stellenwert

26. **Gibt es in Ihrem Projekt Experten, die sich auf das Performance-Testen konzentrieren?**
Markieren Sie nur ein Oval.

◯ ja *Weiter mit Frage 27*
◯ nein *Weiter mit Frage 31*

Testen: Stellenwert - Experten

27. **An welchen Arbeiten sind diese Experten beteiligt?**
Wählen Sie alle zutreffenden Antworten aus.

☐ Anforderungsanalyse
☐ Testplanung
☐ Entwurf & Architektur
☐ Implementierung
☐ Testen
☐ Monitoring (Betrieb)
☐ Monitoring (Test)
☐ Wartung

28. **Können diese Experten Performance-Tests initiieren?**
Markieren Sie nur ein Oval.

◯ ja
◯ nein

29. **Haben Experten Einfluss auf die Zeitplanung (Meilensteine, Aufgabenpakete, Beteiligte verschieben)?**
Markieren Sie nur ein Oval.

◯ ja
◯ nein

30. **Wie kann ein Experte die Zeitplanung beeinflussen?**
Wählen Sie alle zutreffenden Antworten aus.

☐ Experte entscheidet selbst
☐ Experte hat ein Stimmrecht bei der Entscheidung
☐ Experte kann eine Empfehlung aussprechen
☐ Experte kann einen Antrag stellen

Testen: Kenntnisse

31. **Wie schätzen Sie Ihr Wissen in Bezug auf Performance-Testen ein?**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
keines vorhanden	◯	◯	◯	◯	◯	umfangreich

32. **In wie weit konnten Sie Ihr Wissen in Bezug auf Performance-Testen bereits durch praktische Erfahrungen ausbauen?**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
gar nicht	◯	◯	◯	◯	◯	sehr stark

Werkzeuge

33. **Werden Werkzeuge für das Performance-Testen bereitgestellt?**
Markieren Sie nur ein Oval.

- ◯ ja
- ◯ nein

34. **Sofern bereitgestellt, welche Arten von Performance-Tests lassen sich mit diesen Werkzeugen durchführen?**
Wählen Sie alle zutreffenden Antworten aus.

- ☐ Baseline-Test (Einzelnutzertest)
- ☐ Last-Tests (Testet auf bestimmte Nutzerzahl)
- ☐ Stress-Test (Provoziert einen Ausfall)
- ☐ Stabilität-Test (Langzeittest)
- ☐ Smoke-Test (Tests von Änderungen)
- ☐ Isolation-Test (Fokusierung auf fehlerbehaftete Testfälle)
- ☐ Komponenten-Test (Ausführungsdauer einzelner Funktionen)
- ☐ Sonstiges: .....

35. **Sofern bereitgestellt, werden Sie zum Einsatz der bereitgestellten Werkzeuge ermutigt?**
Markieren Sie nur ein Oval.

- ◯ ja
- ◯ nein

36. **Dürfen Sie eigene Werkzeuge für das Performance-Testen verwenden?**
Markieren Sie nur ein Oval.

- ◯ ja
- ◯ nein

37. **Welche Werkzeuge werden bereitgestellt oder von Ihnen für das Performance-Testen eingesetzt?**
Geben Sie die Namen der verwendeten Werkzeuge bitte Komma getrennt ein.

.....
.....
.....
.....
.....

38. **In welchem Umfang werden Testergebnisse aufgezeichnet, so dass die Ergebnisse später jederzeit abgerufen werden können?**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
gar nicht	◯	◯	◯	◯	◯	vollständig

Dokumentation

39. **Wie werden Performance-Anforderungen erhoben?**
Wählen Sie alle zutreffenden Antworten aus.

- ☐ Auswertung von Dokumenten
- ☐ Interviews
- ☐ Fragebögen
- ☐ Use-Case Bestimmung
- ☐ Beobachtung
- ☐ Experimente
- ☐ Prototyping
- ☐ Sonstiges: .....

40. **Sofern Use-Cases bestimmt werden: Werden Use-Cases priorisiert?**
Markieren Sie nur ein Oval.

- ◯ ja
- ◯ nein

41. **Sofern Use-Cases priorisiert werden: Wie werden diese priorisiert?**
Wählen Sie alle zutreffenden Antworten aus.

- [ ] Voting durch relevante Mitarbeiter
- [ ] Voting durch Kunden
- [ ] Expertenmeinung
- [ ] Sonstiges: .....

42. **Sofern Use-Cases bestimmt werden: Werden Use-Cases um Performance-Anforderungen erweitert?**
Wählen Sie alle zutreffenden Antworten aus.

- [ ] Keine Angaben zur Performance
- [ ] Angaben nur bei performance-kritischen Use-Cases
- [ ] Jeder Use-Case enthält Angaben
- [ ] Jedes Szenario eines Use-Cases enthält Angaben
- [ ] Jedes Teilszenario enthält Angaben
- [ ] Sonstiges: .....

43. **Werden Performance-Anforderungen priorisiert?**
Markieren Sie nur ein Oval.

- ◯ ja
- ◯ nein

44. **Sofern eine Priorisierung statt findet: Wie werden Performance-Anforderungen priorisiert?**
Wählen Sie alle zutreffenden Antworten aus.

- [ ] Voting durch relevante Mitarbeiter
- [ ] Voting durch Kunde
- [ ] Expertenmeinung
- [ ] Sonstiges: .....

45. **Auf welche der nachfolgenden Aspekte werden performance-kritische Anforderungen geprüft?**
Wählen Sie alle zutreffenden Antworten aus.

- [ ] Vollständigkeit
- [ ] Konsistenz
- [ ] Machbarkeit
- [ ] Performanceeigenschaften (z.B.: Sollwerte, Schwellwerte oder Mengengerüst)
- [ ] Sonstiges: .....

46. **Wie werden Performance-Anforderungen geprüft?**
Wählen Sie alle zutreffenden Antworten aus.

- [ ] Review
- [ ] Walk-Through
- [ ] Sonstiges: .....

47. **Wer entscheidet, ob eine Anforderung performance-kritisch ist?**
Wählen Sie alle zutreffenden Antworten aus.

- [ ] Kunde
- [ ] Entwickler
- [ ] Testingenieur
- [ ] Teamleiter
- [ ] Gemeinsam im Gespräch
- [ ] Sonstiges: .....

48. **Wie wird mit fehlenden Performance-Anforderungen umgegangen?**
Wählen Sie alle zutreffenden Antworten aus.

- [ ] Nachfragen beim Kunden
- [ ] Standardwerte werden genommen (Industrie, Best-Practices)
- [ ] Erfahrungswerte aus früheren Projekten
- [ ] Sonstiges: .....

49. **Gibt es eine Plattform, auf der sich Entwickler über Performance-Themen austauschen können, und wie rege wird diese Plattform verwendet?**
Wählen Sie "gibt es nicht", wenn die Plattform nicht bereitgestellt wird. Wählen Sie andernfalls, wie rege die Plattform verwendet wird.
Markieren Sie nur ein Oval pro Zeile.

	gibt es nicht	selten genutzt	manchmal genutzt	häufig genutzt
Wiki	◯	◯	◯	◯
Forum	◯	◯	◯	◯
Meetings (Vorstellung aktueller Projekte)	◯	◯	◯	◯
persönlich	◯	◯	◯	◯
Konferenz	◯	◯	◯	◯

50. **Wenn Performanceanforderungen erhoben werden: Werden folgende Eigenschaften berücksichtigt?**
Markieren Sie nur ein Oval pro Zeile.

	nicht berücksichtigt	Kategorisierung	Bereichsangabe	exakter Wert mit Toleranz
Antwortzeit	◯	◯	◯	◯
Workload	◯	◯	◯	◯
Zahl gleichzeitiger Zugriffe	◯	◯	◯	◯
Verfügbarkeit	◯	◯	◯	◯
Hardware-Ressourcen	◯	◯	◯	◯
Netzwerk-Ressourcen	◯	◯	◯	◯

Testabläufe

51. **Wie hoch ist Ihrem Projekt der durchschnittliche Aufwand für das Performance-Testen (im Vergleich zum Gesamten Projektaufwand)?**
*in Prozent
.....

52. **Werden Tests automatisch oder manuell durchgeführt?**
Automatisch bedeutet hier, dass das Ausführen, Messen und eventuelles Finden von Ursachen von einer Test-Suite übernommen wird
Markieren Sie nur ein Oval.

	1	2	3	4	5	
alles manuell	◯	◯	◯	◯	◯	alles automatisch

53. **Welche Arten von Performance-Tests werden bei Ihrem Projekt durchgeführt?**
Wählen Sie alle zutreffenden Antworten aus.

- ☐ Baseline-Test (Einzelnutzertest)
- ☐ Last-Tests (Testet auf bestimmte Nutzerzahl)
- ☐ Stress-Test (Provoziert einen Ausfall)
- ☐ Stabilität-Test (Langzeittest)
- ☐ Smoke-Test (Tests von Änderungen)
- ☐ Isolation-Test (Fokusierung auf fehlerbehaftete Testfälle)
- ☐ Sonstiges: .....

Rückkopplung

54. **Welche Punkte werden in Ihrem Projekt durch Performance-Testergebnisse beeinflusst?**
Markieren Sie nur ein Oval pro Zeile.

	gar nicht	durch schlechte Ergebnisse
Zeitplanung	◯	◯
Zuordnung der Mitarbeit zu Tätigkeiten	◯	◯
Projektbudget	◯	◯
Werkzeugeinsatz	◯	◯

55. **Wie stark wird die weitere Planung im Projekt durch Performance-Testergebnisse beeinflusst?**
Markieren Sie nur ein Oval pro Zeile.

	kein Einfluss	kleine Anpassungen	teilweise Überarbeitung	vollständige Überarbeitung
Durch schlechte Ergebnisse	◯	◯	◯	◯
Durch gute Ergebnisse	◯	◯	◯	◯

56. **Wie stark wird die Planung zukünftiger Projekte durch schlechte Ergebnisse in Bezug auf Performance beeinflusst?**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
kein Einfluss	◯	◯	◯	◯	◯	starker Einfluss

57. **Wie stark wird die Planung zukünftiger Projekte durch gute Ergebnisse in Bezug auf Performance beeinflusst?**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
kein Einfluss	◯	◯	◯	◯	◯	starker Einfluss

58. **Werden die Kunden Ihres Projekts über Performance-Testergebnisse informiert?**
Markieren Sie nur ein Oval.

- ◯ ja
- ◯ nein

59. **Wenn Kunden über Performance-Testergebnisse informiert werden, können sie anschließend Einfluss auf die Performance-Anforderungen nehmen?**
Markieren Sie nur ein Oval.

- ◯ ja
- ◯ nein

60. **Welche Informationen erhalten Entwickler über die Performance-Testergebnisse?**
Wählen Sie alle zutreffenden Antworten aus.

- ☐ Betroffene Komponenten
- ☐ Betroffene Anforderungen
- ☐ Betroffene Arbeitspakete
- ☐ Grad der Verbesserung
- ☐ Sonstiges: .....

61. **Bei wie vielen Projekten zeigen sich nach Inbetriebnahme Performance-Probleme?**
Markieren Sie nur ein Oval.

	1	2	3	4	5	
bei keinen	◯	◯	◯	◯	◯	bei allen

62. **Wenn Sie Interesse an der Auswertung haben, hinterlassen Sie hier bitte Ihre E-Mail-Adresse.**

.....

Bibliography

[Adesso AG 2014] Adesso AG. Leistungen der Adesso AG. 2014. URL: http://www.adesso.de/de/leistungen/. (Cited on page 11)

[Behr and Tyll 2003] T. Behr and T. Tyll. Online-Lehrbuch. In: Universität Erlangen, Prof. Beyer, 2003. Chapter 3: Geschäftsprozesse - Ablaufoptimierung, pages 1–8. (Cited on pages 3, 4)

[Brosius and Koschel 2005] H.-B. Brosius and F. Koschel. Methoden der empirischen Kommunikationsforschung. Eine Einführung. Wiesbaden: VS Verlag für Sozialwissenschaft, 2005. (Cited on pages 4–6)

[Hower 2014] R. Hower. Web Site Test Tools and Site Management Tools. Online. 2014. URL: http://softwareqatest.com/qatweb1.html. (Cited on page 9)

[Liddle 2008] J. Liddle. Amazon Found Every 100ms of Latency Cost Them 1% in Sales. Online. 2008. URL: http://blog.gigaspaces.com/amazon-found-every-100ms-of-latency-cost-them-1-in-sales. (Cited on page 1)

[Ludewig and Lichter 2010] J. Ludewig and H. Lichter. Software Engineering. 2. Heidelberg: dpunkt.verlag GmbH, 2010. (Cited on pages 1 and 6)

[Meier et al. 2007] J. D. Meier, C. Farre, P. Bansode, S. Barber, and D. Rea. Performance Testing Guidance for Web Applications. Microsoft Corporation, 2007. URL: http://msdn.microsoft.com/en-us/library/bb924375.aspx. (Cited on pages 6 and 29)

[Miller 1956] G. A. Miller. The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. *The Psychological Review* 2 (1956), pages 81–97. (Cited on page 6)

[Molyneaux 2009] I. Molyneaux. The Art of Application Performance Testing: Help for Programmers and Quality Assurance. 1st. O'Reilly Media, Inc., 2009. (Cited on pages 1, 6, 23, 29, and 31)

[Stefanov 2008] S. Stefanov. Exceptional Website Performance with YSlow 2.0. In: *China Software Developers Network*. 2008. URL: http://de.slideshare.net/stoyan/yslow-20-presentation. (Cited on page 1)

[Tudenhöfner 2011] E. Tudenhöfner. Management into the Application Lifecycle. Master Thesis. Stuttgart University of Applied Science, 2011. (Cited on pages 23, 29, and 31)

Declaration

We declare that this thesis is the solely effort of the authors. We did not use any other sources and references than the listed ones. We have marked all contained direct or indirect statements from other sources as such. Neither this work nor significant parts of it were part of another review process. We did not publish this work partially or completely yet. The electronic copy is consistent with all submitted copies.

Place, Date, Signature