

# Project plan for degree project

Version 1.3 – October 12, 2016

PA2534: MASTER THESIS IN SOFTWARE ENGINEERING

August 29, 2019

Thesis	Tentative title	A Systematic approach to analyze the performance of the real-time software systems
	Classification	Software, Performance, Factors, Real-time, Analyze*
Student 1	Name	Sravani Bejawada
	e-Mail	besr17@student.bth.se
	Social security nr	9705259563
	Visa expiration date	
Supervisor	Name and title	Deepika Badampudi, Associate senior lecturer
	e-Mail	deepika.badampudi@bth.se
	Department	Software Engineering
External	Name and title	Iwe Lingström, Operative System Manager
	e-Mail	iwe.lingstrom@ericsson.com
	Company/HEI	Ericsson

\*2012 ACM Computing Classification System: [www.acm.org/about/class/2012](http://www.acm.org/about/class/2012).

\*\*Co-advisor from industry or a higher education institution (HEI).

## 1 Introduction

“Nowadays, nonfunctional testing has a great impact on the real-time environment” [1]. Performance testing is a nonfunctional black box testing. It is a type of software testing in which it ensures whether the software performs well under the expected load. Not only the features and functionality, even response time, reliability, resource usage, and scalability matter. Performance testing goal is not only finding bugs but also to eliminate the performance bottlenecks [2]. Performance testing is done to improve some common issues like long load time, poor response time, poor scalability, bottlenecks, poor usability and inconsistency in different operating systems. Some of the common performance bottlenecks are CPU utilization, network utilization, memory utilization, operating systems limitations and disk usage [2]. Performance testing also determines whether the software under expected workloads meets speed, scalability and stability requirements. The software which is released with poor performance testing or poor performance metrics may lead to a bad reputation and affects the sales goals by decreasing [2].

The basic Performance testing metrics which are monitored during testing are: *Processor usage, Memory use, Disk time, Bandwidth, private bytes, committed memory, memory pages/second, page faults/second, CPU interrupts per second, disk queue length, network bytes total per second, response time, throughput, amount of connection pooling, maximum active sessions, hit ratios, hits per second, rollback segment, database locks, top waits, thread counts,*

*garbage collection. For identifying the performance acceptance criteria, performance testing includes goals and constraints for throughput, response time, and resource allocation [2]. Only the client-server-based systems require performance testing. This means that any system which isn't client-server-based does not require performance testing. For example, the Microsoft calculator can be run by multiple users, but it isn't client-server-based. Hence it does not require performance testing [2].*

In many cases, the software projects which lack in performance lead to severe problems and may also lead to project failure [3]. Therefore, in the early stages conducting performance testing and performance measurement are essential to know for the better performance of the software [4]. While testing a web application, its performance is considered as an essential factor. The performance testing of large industrial client transaction processing software is evaluated using some measures such as resource usage, throughput, stimulus-response time, services (which have been waiting for the maximum number of tasks) and typical resources (such as network bandwidth requirements, CPU cycles, disk space, disk access operations, memory usage and switch usage ) [5].

The performance testing in traditional applications focuses on performance metrics only on a particular workload with fixed configuration, whereas the cloud test measures the performance metrics related to workloads, which run on multiple virtual and real machines. For a better understanding of performance in real-time, the web applications must be tested for different types of hardware, multiple operating systems and multiple browser platforms [6]. Testing performance in cloud computing with multiple operating systems, updates, multiple browser platforms, versions to understand performance in real-time different types of hardware and a large number of users are involved [7].

Software performance testing is used in every software vertical, including space programs, financials, digital communication, mail servers, cars, geo-positioning systems, nuclear power plants, farming applications, weather forecast games etc. When combining a high volume of data for instant processing, with low response latency and extensive logging: the need for efficient performance testing and performance modelling is important. One vertical which includes high volume, low latency and logging are usages and financial tracking for digital communication which can be phrased as part of complex revenue and customer management systems using software to large *Communication service providers* (CSPs) in all geographies [8]. CSP companies use *Business Support Systems* (BSS) to run their business operations towards the customers [9]. Customers are accessed remotely in a software platform where the provider runs the web application in the cloud, where the provider is responsible for managing and maintaining the software with *Service level agreements* (SLAs) [10].

## **2 Research Problem**

Cloud-based BSS which has recently released to the market, accounting for existing hardware limitations and generally imposed by the customer operator and it is necessary to understand how a BSS performs (e.g., CPU and RAM usage) and what factors impact its performance. Here BSS is selected as an example because this is one of the most prominent software which has more complexity, latency, and load. The efficient understanding of how the system behaves helps for planning an early dimensioning of needed resources, and the efficient understanding of the factors which has significant and minor impact helps to know which test cases are to be

neglected and thereby reducing the storage of additional data.

The goal of the study is to illustrate an approach for performance analysis because there is no systematic approach to analyze the performance of the complex system like BSS more. Myriad of factors impact impacts performance; it is not so easy to identify the adequate hardware for different performance requirements. *Identification of factors is the path of the performance analysis approach, and this is one of the activities in this process.* The approach is designed to analyze not only for BSS performance but also for any of the real-time software systems. The efficient understanding of how the system behaves is essential because how the systems behave to allocate hardware and to improve the configuration of the system to comply with the existing constraints.

This case (online charging system) is more important because dealing with the systems which are generating revenue for companies, and there will be a loss if the system doesn't perform as expected. The systems which are not optimized would be more expensive to deploy the systems. Right now, the organization doesn't have a thorough understanding of the behavior of the system how it impacts the hardware dimensioning, which makes hard for the operator. For example: where operator 'x' needs to know how much hardware is required to deal with how many subscribers and to help the operators to allocate the adequate the hardware. When the organization needs more hardware resources, the company needs to invest more money. Similarly, when it is less hardware, the system doesn't perform as expected. So, we need to understand the performance of real-time systems by measuring in an systematic way to analyze the performance.

## 3 Background and Related Work

### 3.1 Background

In Software Engineering, we have six types of Performance testing types namely: *load testing, stress testing, endurance testing, spike testing, volume testing and scalability testing* [2]. Load testing and stress testing are the most commonly used types in performance testing. Load testing helps to determine the system performance and behavior under normal workload situations. Load testing is used only to simulate under the actual workload. Load testing helps in identifying the bottlenecks, and when this testing is avoided, it may lead to a commercial loss. Stress testing also used in identifying the system behavior and helps to know under which extreme conditions the system fails to perform. So, both load testing and stress testing is focused in this research [2], [1].

BSS is a collection of systems and functions which are interconnected to give access to the operator. The operator collects the money from the end users in cash management systems like bank/ATM (*ATM- Automated teller machine*) in time according to the rules of the agreement [11]. The primary usage of the BSS, which helps to stores all the information related to the customer. Example: A BSS system holds all the customer's information like business processes like postpaid/prepaid customers, the aggregated traffic/usage and pricing. The network checks each subscriber when there is an action like data traffic, instant changes, call, message, etc., whether there is a sufficient amount in the account to charge or applicable credit. The four primary functions of BSS are customer management, order management, product management, and revenue management [11].

The framework used in BSS is business process framework, which is also known as the *Enhanced Telecom Operations Map(eTOM)* framework. This is a reference framework where the business activities are classified, i.e., the business activities like "strategy, infrastructure and product", "operations" and "enterprise management". Activities with similar features are grouped into one category based on their definitions [12]. For example, the activities are differentiated as organizational, functional, non-functional and others. Within a model, all these categories are explained with the help of process flows. This framework even serves as a blueprint, which will help as a start point by setting directions for the development and integration of BSS [12]. In BSS more specifically focus on the charging system, Charging (CHA) executes business rules to determine charges and updates account balances accordingly. To be able to rate and charge different scenarios, CHA executes rating rules and applies prices defined in the product offering. CHA provides the single rating function that handles diameter interfaces and calculates all charges in the system, regardless of source or type, like online, offline, recurring, usage and late arrivals for retail customers.

### 3.2 Related Work

Earlier, many approaches are proposed for the performance analysis and testing resource allocation problems of the software. The following approaches which are considered related to this research discussed below.

*Software performance engineering* (SPE) approach [3] which directly fulfills the requirements of the performance in a step by step process and the different tasks that are performed in this approach are: Identifying the qualitative data (factors affecting the performance) of the software, analyzing the performance, Performance Prediction, Performance testing, Maintenance of the software and analysis of the overall system. In SPE approach, the factors affecting the performance are identified. But in SPE approach, there is a need of better performance measurement tools and modelling for better results and to improve the performance of the system [3].

Saaty [13] introduced the *Analytic hierarchy process* (AHP) mathematically based, a decision-making and multi-objective tool. In many applications like an evaluation of business performance, allocation of resources, selection of projects and other areas AHP approach is adopted. In this process, the major step is comparisons are judged through highly sophisticated solutions from experiments or judgment [14][15]. The judgments are made by using a nine-point ratio scale when ranges for two factors are equally important. However, one factor would be more important than other factor so, the decision is collected from experts to prioritize the factors. Then calculations are made by using Eigen principal vectors [16], priorities are made in the final stage of the analysis. From different perspectives for development, AHP is an important tool for the systematic analysis of the practitioner's opinions [14][15].

To overcome the problems in the performance measurement: a model was developed by researchers named as *Quantitative models for performance measurement system* (QMPMS) by using AHP [4]. QMPMS is used to measure the impact of factors on performance. *The three most important steps in QMPMS are: Factors influencing the performance and the relationship between them are identified, structuring the factors progressively and evaluating the impact of the factors on performance* [4]. In AHP Rank reversal phenomenon is applied when there is an

addition or deletion of the factors in the model [14] [15].

Heiko Koziol [17], focused on proposing new approaches for performance prediction and performance measurement of component-based software systems. *Factors influencing the component performance of software are implementation of the component, required services, deployment platform, usage profile and resource contention.* To analyze the running system, all the factors which influence the performance are measured by using tools [17]. The need for improving the performance testing and its automation is explained in detail by an experiment [18]. Instead of focusing on the high-level *Quality of services* (QoS) like latency and throughput, the author focused on low-level network cloud infrastructure. Introduced an optical multi-layer architecture, the resource status is monitored, and network bandwidth is guaranteed at run time [19]. Worked on dynamic resource optimization for self-organized clouds and task resource allocation problem is solved under budget instead of performance [20].

Sun, Yu et al. [21] reported QoS-oriented framework called the *resource optimization, allocation and recommendation system* (ROAR) used for web applications which automate the testing and the derivation of optimized cloud resource allocation. It even automates end to end orchestration for application deployment in the cloud platform, collection of performance metrics and performance model test can be analyzed. This system uses a domain specific language called *generic resource optimization for web application language* (GROWL) [21], which generates high level and customized resource allocation configuration with optimization to meet the web application QoS goals. But with this ROAR system, the answers for best practices in load testing and resource optimization are missing. Analyzing the behavior of complex systems with model-driven input parameters is missing. To analyze the behavior of complex systems, there are a large number of input factors which impact the performance. So, there is a need for systematic optimization of the factors which impact the performance [21].

Compared optimal testing resource allocation with Ukimoto et al.'s model [22] which resulted showing that Ukimoto et al.'s includes all the risks released in software fails with low reliability, as the complex systems require high reliability, which is achieved by architecture-based software model. This model did not determine parameters for testing resource allocation, so; there is a need of approach for calculating the parameters [22]. There is a need of approach to allow people to analyze and plan the resources to achieve optimum resources utilization the performance of real-time systems, more specifically, for the complex systems like BSS.

From the existing literature observed that there are a large number of papers proposing new frameworks and metrics for performance [23]. Even though factors have a direct impact on the performance of the software, but there is no systematic approach with input factors to analyze the performance of the real-time software system [5][21]. As the technology is developing, there is a need of better performance measurement to measure all the factors [3]. The main reason is many organizations don't have time, resources or experience to derive myriad of factors impacting performance. Instead, developers use the trial and error approach to analyze the performance. Trial and error approach is difficult and time taking process while working with complex systems [21]. By illustrating this approach, performance can be improved by analyzing the behavior of the software system. Systems behavior is very important to allocate the hardware and to improve the system configuration with the existing constraints.

## 4 Aim and Objectives

**Aim:** The main aim of this research is to illustrate a systematic approach for performance analysis of real-time software systems with factors as input.

The following objectives which help in achieving the overall goal of the thesis:

**Objective1:** To identify the most important factors that impact the performance of the real-time software system.

**Objective2:** Create an approach to improve the performance of the real-time software system.

## 5 Research Questions

**RQ1: What are the most important factors that impact the performance of the real-time software system?**

In an organization, the online charging system of BSS is taken as a case to answer this RQ. Initial factors had a direct impact on the performance of the software in distributed telecommunication applications [5]. This RQ helps to know the important factors that impact the performance of the real-time software system. This is the primary step to analyze the performance of the software system.

**RQ2: How to improve the performance of the real-time software system?**

Many organizations don't have time, resources or experience to derive myriad of factors impacting performance. Instead, developers use the trial and error approach to analyze the performance. Even there is no systematic optimization approach to analyze the behavior of the real-time software system with input factors [21]. By answering this RQ, performance and the behavior of the system can be better understood.

## 6 Research Methodology

### 6.1 Literature review

This method helps in answering the RQ1. This method is preferred to study the important factors that impact the performance of a real-time software system from the existing literature. The primary purpose of choosing this method to gain knowledge about the different performance metrics, performance issues and performance approaches from the literature. A literature review is a step by step approach where the relevant papers are selected based on research aim [24]. The search string is formulated to study the efficient literature which is related to this research, i.e., journals and conferences from different databases like Google Scholar, IEEE Xplore, and Scopus. The results obtained from the literature review are compared with survey results. To know the important factors which impact the performance of real-time software systems.

### 6.2 Survey

This method helps to answer RQ1. The survey is conducted to know the important factors and performance metrics that impact the performance of the software from the experienced people

working in BSS. The survey will be conducted with experts in BSS like data scientist, software developer, system managers, and practitioners. For empirical studies, a well-known strategy is performing a survey for asking people about their opinions. There are different types of surveys like, street surveys, mail surveys or skype surveys [25]. Surveys have lower internal validity and higher external validity when compared to the case study and experiment. The steps evolved during the survey are: (a) First, the goal of the thesis is explained, (b) Then the questionnaire is formulated according to the goal and selection of respondents, (c) Check whether the survey is executable, (d) Collection and processing the data, (e) Data is interpreted, (f) The survey results are reported [25].

The survey is conducted in Ericsson, and the participants are selected using sampling. Where sampling is a process to select the respondents in the survey. Highly experienced 12 respondents are selected, who have knowledge in this domain and working on cloud-based BSS in Ericsson [26]. The quantitative data is collected from the Ericsson experts working with BSS in different teams. From the literature knowledge, the questionnaire is prepared. For example, the important questions that will be asked in the survey are: what are the most important factors that impact the performance of BSS? What are the factors that impact the performance of the online charging ? And what are the most important three factors that impact the performance of the online charging system? The experts answer a fixed questionnaire, and the results are prioritized based on their experience.

## 6.3 Empirical methods

In the context of software engineering, the widely used empirical methods are Experiment [27], Surveys[25], Action research [28] and Case study[24]. From empirical methods, Action research and survey is selected as suitable methods for this study.

***Reasons for rejecting the experiment and case study are:***

An ***Experiment*** is avoided because all the variables should be identified in the earlier stages, which is difficult to perform, and we cannot have full control over the system [27].

A ***Case study*** is avoided because it is used primarily for descriptive and exploratory purposes for pre and post-event studies [24]. But the study aim is intervened in the real world and observing the effects and improvements are made so, to reach this goal case study is not suitable [29].

### 6.3.1 Action Research

Action research is a suitable research methodology for answering RQ2. This methodology will be used for analyzing the behavior of the software by a systematic approach for making improvements, changes in process, or to analyze the impact of the obtained outcome [24][28]. This research method is important as it primarily focuses on real-time problems, especially while working with real-time systems like BSS, where the changes take place very quickly [28],[30]. This methodology is used to improve the situation by observing its effects and take an action [30]. Both qualitative and quantitative data is collected. The approach will be improved in different phases:

**In phase 1:** Initially, a systematic approach will be designed for performance analysis of the online charging system with input factors. Based on the outcome, the changes will be made in the approach until the desired result is obtained.

**In phase 2:** Further approach is developed in such a way that the approach can analyze the

performance of any of the real-time software system.

**In phase 3:** Based on the outcome, the changes will be made in the approach until the expected result is obtained. Further implementation is carried out by CHA unit in Ericsson.

## 6.4 Data Collection

Interviews help to know in detail which factors had affected in earlier projects in what way with their experience. The interviewees will be selected from all the teams which are involved in BSS. The main objective of the interview, the researcher typically focuses on finding the relation by combining the answers from the interviewees. The closed questionnaire is formulated based on the collected data from a survey from the interviewees, and this may increase to come up with better results. An interview is conducted to obtain qualitative data from the interviewee, and the data is analyzed [24]. For example, the typical Questionnaire for the interview is prepared based on the survey results answered by the interviewee: why do you think only these factors have an impact? Why not other factors which are answered by the other respondents? Initially, there would be an introduction of ourselves and the thesis. Externally or internally in the organization, only the information which is anonymously collected from the interviewees is published. All the interviews are audio recorded with permission from the interviewees.

## 6.5 Data Analysis

Thematic analysis is one of the widely used analysis methods for qualitative data [24] [31]. “Thematic analysis is a method used for identifying, analyzing and reporting themes within data”. This analysis, has flexibility, gives a rich and detailed description of the data [32] [33] and [34]. Thematic analysis is suitable when the questionnaire is related to people’s opinions and experiences[31]. The results are gathered, and some important data is interpreted in related to the research questions and patterns are presented. A generic thematic approach is selected as the study focuses on collecting the opinions of the practitioners [34]. This analysis helps to know the most important factors that impact the performance of the online charging system. The quantitative data is analyzed by using some statistical methods like mean, median and mode.

## 7 Expected Outcomes

The following are the expected outcomes after performing this research:  
A systematic approach to analyze the behavior of the real-time software system.

## 8 Time and Activity Plan

Tasks	Start date	End date
Initial meetings with supervisors and team to understand the research scope	15-01-2019	03-02-2019
Background analysis	17-01-2019	15-02-2019
Writing a thesis proposal	16-02-2019	13-06-2019
First draft submission of thesis proposal	09-02-2019	10-02-2019
Second draft submission of thesis proposal	22-03-2019	23-03-2019
Conducting a literature review	20-03-2019	23-04-2019



Discussion of survey and interview questionnaire with supervisor	24-04-2019	30-04-2019
Conducting survey and interviews at Ericsson	01-05-2019	19-05-2019
Analyzing the survey and interviews results and the results are discussed with the supervisor	20-05-2019	30-05-2019
Action research implementation	31-05-2019	15-07-2019
Final submission of the thesis proposal	14-06-2019	16-06-2019
Analyzing the action research results and the results are discussed with the supervisor	18-08-2019	03-08-2019
Writing thesis documentation	15-05-2019	12-09-2019
First draft thesis document to the supervisor	13-09-2019	14-09-2019
Final thesis presentation	30-09-2019	04-10-2019
Final thesis document to the examiner	11-10-2019	13-10-2019

Table 1: Time and Activity plan.

## 9 Risk Management

1. **Lack of complete access to data.** Because of the non-disclosure agreement, the company might not provide access explicitly. In that aspect, we would gather the qualitative information needed for estimating the necessary metrics from the developers so that it would be useful for our study.
2. **Project effort underestimation:** It is often the case that people underestimate the effort of tasks, mainly if they will do something new, and this may lead to bad planning. To avoid this, our supervisors and we will have discussed the plan to make it as accurate as possible.
3. **Difficulty in obtaining the data:** We will need data to be able to analyze the impact of different factors. However, the data may not be in an adequate format or enough. To avoid this, we will involve the related stakeholders to ensure we get data or produce if necessary.

## References

- [1] Shiva Bhavani Reddy Charla. Examining various input patterns effecting software application performance: A quasi-experiment on performance testing, 2016.
- [2] G Claudio Demartini and Ebrahim Kargar Nasrabadi. Quality assurance of software products. 2017.
- [3] Murray Woodside, Greg Franks, and Dorina C Petriu. The future of software performance engineering. In *2007 Future of Software Engineering*, pages 171–187. IEEE Computer Society, 2007.
- [4] PTBITITCI Suwignjo, Umit S Bititci, and Allan S Carrie. Quantitative models for performance measurement system. *International journal of production economics*, 64(1-3):231–241, 2000.

- [5] Filippos I Vokolos and Elaine J Weyuker. Performance testing of software systems. In *Proceedings of the 1st International Workshop on Software and Performance*, pages 80–87. ACM, 1998.
- [6] Eljona Proko and Ilia Ninka. Analysis and strategy for the performance testing in cloud computing. *Global Journal of Computer Science and Technology*, 2012.
- [7] Mohd Ehmer Khan and Farmeena Khan. Importance of software testing in software development life cycle. *International Journal of Computer Science Issues (IJCSI)*, 11(2):120, 2014.
- [8] Magic quadrant for integrated revenue and customer management for csps. <https://www.gartner.com/en/documents/3901476/magic-quadrant-for-integrated-revenue-and-customer-manag>.
- [9] Gb991 core frameworks concepts and principles r18.0.1 - tm forum. <https://www.tmforum.org/resources/standard/gb991-core-frameworks-concepts-and-principles-r18-0-0/>.
- [10] Shu Zhang and Meina Song. An architecture design of life cycle based sla management. In *2010 The 12th International Conference on Advanced Communication Technology (ICACT)*, volume 2, pages 1351–1355. IEEE, 2010.
- [11] Bss archives - telecom oss bss. <http://www.telecomossbss.com/category/bss/>.
- [12] Gb921 business process framework getting started r14.5.1 (pdf) - tm forum. <https://www.tmforum.org/resources/best-practice-reference/gb921-business-process-framework-getting-started-r14-5-1-pdf/>.
- [13] Thomas L Saaty. How to make a decision: the analytic hierarchy process. *European journal of operational research*, 48(1):9–26, 1990.
- [14] Omkarprasad S Vaidya and Sushil Kumar. Analytic hierarchy process: An overview of applications. *European Journal of operational research*, 169(1):1–29, 2006.
- [15] Godwin G Udo. Using analytic hierarchy process to analyze the information technology outsourcing decision. *Industrial Management & Data Systems*, 100(9):421–429, 2000.
- [16] Lindsay I Smith. A tutorial on principal components analysis. Technical report, 2002.
- [17] Heiko Koziolk. Performance evaluation of component-based software systems: A survey. *Performance Evaluation*, 67(8):634–658, 2010.
- [18] Priyanudeep Eada. Experiment to evaluate an innovative test framework: Automation of non-functional testing, 2015.
- [19] Pan Yi, Hui Ding, and Byrav Ramamurthy. Cost-optimized joint resource allocation in grids/clouds with multilayer optical network architecture. *IEEE/OSA Journal of Optical Communications and Networking*, 6(10):911–924, 2014.
- [20] Sheng Di and Cho-Li Wang. Error-tolerant resource allocation and payment minimization for cloud system. *IEEE Transactions on Parallel and Distributed Systems*, 24(6):1097–1106, 2012.

- [21] Yu Sun, Jules White, Sean Eade, and Douglas C Schmidt. Roar: A qos-oriented modeling framework for automated cloud resource allocation and optimization. *Journal of Systems and Software*, 116:146–161, 2016.
- [22] Hiroyuki Okamura and Tadashi Dohi. Optimizing testing-resource allocation using architecture-based software reliability model. *Journal of Optimization*, 2018, 2018.
- [23] Xiao-yang Guo, Ying-hui Chen, Xue-song Qiu, and Fan Tang. Design and implementation of performance testing model for web services. In *2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR 2010)*, volume 1, pages 353–356. IEEE, 2010.
- [24] Per Runeson and Martin Höst. Guidelines for conducting and reporting case study research in software engineering. *Empirical software engineering*, 14(2):131, 2009.
- [25] Teade Punter, Marcus Ciolkowski, Bernd Freimut, and Isabel John. Conducting on-line surveys in software engineering. In *2003 International Symposium on Empirical Software Engineering, 2003. ISESE 2003. Proceedings.*, pages 80–88. IEEE, 2003.
- [26] Martin N Marshall. Sampling for qualitative research. *Family practice*, 13(6):522–526, 1996.
- [27] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. *Experimentation in software engineering*. Springer Science & Business Media, 2012.
- [28] Rory O’Brien. An overview of the methodological approach of action research. 1998.
- [29] Steve Easterbrook, Janice Singer, Margaret-Anne Storey, and Daniela Damian. Selecting empirical methods for software engineering research. In *Guide to advanced empirical software engineering*, pages 285–311. Springer, 2008.
- [30] Colin Robson and Kieran McCartan. *Real world research*. John Wiley & Sons, 2016.
- [31] Mojtaba Vaismoradi, Hannele Turunen, and Terese Bondas. Content analysis and thematic analysis: Implications for conducting a qualitative descriptive study. *Nursing & health sciences*, 15(3):398–405, 2013.
- [32] Richard E Boyatzis. *Transforming qualitative information: Thematic analysis and code development*. sage, 1998.
- [33] Mojtaba Vaismoradi, Jacqueline Jones, Hannele Turunen, and Sherrill Snelgrove. Theme development in qualitative content analysis and thematic analysis. 2016.
- [34] Virginia Braun and Victoria Clarke. Using thematic analysis in psychology. *Qualitative research in psychology*, 3(2):77–101, 2006.