Creating a ITIL-based Software Incident Categorization Model for Measurement: A Case Study

Sanna Heikkinen, Antti Suhonen, Mika Kurenniemi, and Marko Jäntti
University of Eastern Finland
School of Computing
Email: firsname.lastname@uef.fi

***Abstract*—Many IT organizations have recognized incident categorization as a problematic subject because there are no general policies or guidelines for incident categorization. This leads to incident categorization usually being seen as an optional task for the specialists who handle incidents. This article presents the results of a case study that was carried out in an energy business unit of a Nordic IT service company. The research problem of this study is as follows: what type of software incident categorization model would be efficient and would also support ITIL-based continual service improvement? The results of this study consist of two parts: First, the software incident categorization (SIC) model which helps an IT organization to categorize incidents effectively and recognize the weak points of the software development process, and second, the provision of the lessons learned for improving incident categorization and measurement practices.**

Keywords*—IT service management; ITIL; continual service improvement; incident management; software incident categorization model

I. INTRODUCTION

Managing incidents effectively is an essential operation for an IT organization and it usually affects several of the activities of the organization e.g., software development needs to change or fix an application or software in order to resolve an incident. IT organizations use different types of terms to define an incident (e.g., error, fix, bug, problem, programming error, user error, and hardware error), which may complicate understanding the meaning of the term, especially when the organization and its stakeholders are communicating about incidents. According to ITIL version 3 (Information Technology Infrastructure Library), an incident is an unplanned interruption to an IT service or reduction in the quality of an IT service [1]. In practice, an incident can be e.g., a software error, which prevents normal use of software, a malfunction in the printer, or a crashed database server. In this paper, the researchers use the description of ITIL v3 for the term ”incident”.

The ITIL is a set of good practices for directing and managing IT services and it can be tailored to any IT organization [2]. This study will focus on the Service Operation [1] and Continual Service Improvement (CSI) [3] lifecycle phases. One of the key processes of the Service Operation is incident management, which is responsible for managing the lifecycle of all incidents. According to the CSI ideology, an organization needs to measure the incident management process so that the organization can be sure that the process works effectively. The measurement data should be used to identify ideas for improvement to IT services or processes.

During the incident management process, incidents are arranged into categories. This is usually done by the service desk employees who are responsible for handling incident tickets through IT service management system. Incident categorization enables similar incidents to be tracked, which helps to recognize the weak points of services and processes. Although incident categorization is an important phase in incident management, there are no common incident categorization models, guides, or other best practices. This leads to the fact that organizations may create ineffective and unclear models for incident categorization and might mean that employees do not always understand the reasons and benefits which suggest why incident categorization should be performed in the first place. In practice, incident categorization should be user-friendly and explicit, and it should not slow down IT service management activities conducted by employees, such as diagnosing, escalating, and resolving incidents.

Incident categories are an important source of information when it comes to measuring and analyzing. The data that software incident categorization produces help IT organizations to identify the challenges and quality gaps in services and processes from the software lifecycle management point of view. Appropriate software incident categories allow the comparison of incident categorization data without country- or product- specific limitations. The organization’s future process improvement plans can also benefit from the data that software incident categorization produces. Ultimately, effective software incident categorization leads to increased customer satisfaction by improving product and service quality.

A. Related Work

Incident management is a central process for IT organizations and therefore many articles have been written about the subject from the software engineering and IT service management (ITSM) points of views. However, there have only been a few studies that have concentrated on incident categorization from the ITSM perspective. The present researchers exploited the following scientific articles while creating the software incident categorization model. In their paper Vipindeep and Pankaj [4] describe some of the common programming errors and poor programming practices that are often the cause of different types of bugs. Collofello and Balcom [5] intro-

 ISBN: 978-1-61208-304-9

duce a causative software error classification scheme which emphasizes the entire software lifecycle and the causative attributes of the errors. In their paper Nakajo and Kume [6] researched the cause-and-effect relationship of software errors and human errors, which offers an appropriate framework for classifying the software errors. Lutz [7] used this framework when analyzing software requirement errors in safety-critical embedded systems. In their paper Leszak, Perry, and Stoll [8] describe a four-dimensional root cause classification. These four dimensions are human, review, project, and lifecycle. Owens, Womack, and Gonzalez [9] researched software error classification using a defect detection tool. Software errors were categorized into five classes: uninitialized memory read, array bounds write, array bounds read, free memory read, and free memory write errors. IEEE standard 1044-2009 [10] provides a uniform approach to classifying software anomalies, regardless of whether they occur within the project, product, or system lifecycle. Classification data can be used for a variety of purposes, including defect causal analysis, project management, and software process improvement.

B. Our Contribution

The main contributions of this paper are: 1) the software incident categorization (SIC) model which helps an IT organization to categorize incidents effectively and recognize the weak points of its software development process; 2) the provision of lessons learned for improving incident categorization and measurement practices.

The goal of this study was to design an appropriate and consistent incident categorization model which an IT organization could configure into its ITSM system. The purpose of the SIC model is to help IT organization to allocate incidents to a specific part of the software development process. In other words, the SIC model makes it easier to detect sections where customers have found incidents and which are not detected by the IT organization. The results of this study are mainly meant to be of benefit to the persons who are responsible for managing, measuring, and reporting IT services and IT service management processes (e.g., service owners, service managers, process owners, and process managers). This research does not address how the SIC model should be integrated into different ITSM systems. However, this integration should not be problematic with the systems that support ITIL v3 best practices because the SIC model was built on the basis of ITIL.

The rest of the paper is organized as follows. The research problem and methods are described in Section 2. The creation and validation of the software incident categorization model is covered by Section 3. The analysis of the findings, with lessons learned, is covered in Section 4. The conclusion in Section 5 summarizes the case study.

II. Research Methods

The research problem of this study is this: what type of software incident categorization model would be efficient and would also support ITIL-based continual service improvement? This study was a qualitative research study which was built using the case study research and action research methods. The research problem was divided into the following research questions:

- RQ1: What type of information can be used as a guide in creating an effective software incident categorization model?
- RQ2: How should the software incident categorization model be structured so that software-related incidents can be arranged effectively?
- RQ3: How should the software incident categorization model be validated?
- RQ4: How can incident categorization be used to support key CSI activities, such as measurement, reporting, and identifying the ideas for improvements?

During the case study, a researcher is an outsider, who observes and analyses an environment and makes notes by combining different data collection methods [11]. According to Baskerville [12], the action research method produces highly relevant research results because it is grounded in practical action, and it solves an immediate problem case while carefully informing theory. These selected methods support a situation where the researchers work together on a research project and their objective is to identify and solve problems in the IT organization's environment. The researchers used ITIL [2], and the ISO/IEC 20 000 standard [13] as theoretical frameworks in this study.

A. Case Organization and Data Collection Methods

The case subject of this study was an energy business unit which is part of a Nordic IT service company that provides solutions and services for Scandinavian energy companies. In 2012, the Nordic IT service company had around 17 000 employees operating in over 20 countries. The company's energy business unit is one of the research project's cooperation partners. This energy business unit will be referred to by the name Alpha for the rest of the paper.

The research was conducted in January 2013, using the KISMET (Keys to IT Service Management Excellence Technique) model as a roadmap to improve incident management practices. The KISMET model is presented in more detail in Suhonen's et al. research paper [14]. Multiple data collection methods proposed by Yin [11] were used during the study and the following data sources were used:

- **Documents**: meeting memos and process charts.
- **Archival records**: articles, incident categorization sets, and incident records.
- **Participatory observation**: meetings and discussions with managers (e.g., product, portfolio, development, release, and test managers).
- **Physical artifacts**: access to the intranet and to the IT service management system.
- **Semi-structured themed interviews**: interviews with five of the IT organization's staff members (senior software engineer, service desk specialists, and continuous service manager).

 ISBN: 978-1-61208-304-9

B. Data Analysis Method

This study was performed by using within-case analysis for a single organization. According to Eisenhardt [15], the within-case method typically involves detailed case study write-ups for each site and becoming familiar with the case as a stand-alone entity. The data analysis was performed collectively with the research group. The idea behind this collective analysis is to provide "seeds for development" and to use their expertise in the analysis, as they know their specific fields best [16]. The triangulation used in this study allowed the researchers to be more confident about their results. Denzin [17] extended the idea of triangulation beyond its conventional association with research methods and designs. During the study the researchers used three forms of triangulation [17]: 1) data triangulation, which includes collecting data through several sampling strategies; 2) investigator triangulation, which refers to the use of more than one researcher in the field to gather and interpret data, and 3) methodological triangulation, which refers to the use of more than one method for gathering data. The research work was organized into chronological order by the phases of the KISMET model. The research work was validated during weekly meetings with Alpha's representatives.

III. RESULTS

In this section, the researchers will introduce the way in which the software incident categorization model was created in cooperation with the case organization and the research team. The research work consisted of five main phases: A) investigating the current state of incident management and planning improvement actions; B) designing a software incident categorization model based on ITSM practices; C) presenting the main categories and subcategories of the software incident categorization model; D) validating the SIC model, and E) presenting continual service improvement actions. These phases are described in the following subsections.

A. Investigating the current state of incident management and planning improvement actions

The kickoff meeting between the research team and the business unit Alpha was held in January 2013. At that meeting, the representatives of Alpha reported that they would like to improve and unify their unit's internal measurement practices by designing a software incident categorization model.

The researchers analyzed the current state of Alpha's incident management. During the analysis, the research team recognized a few challenges which implied to the team that appropriate improvement actions were needed. After that the researchers defined the improvement actions for Alpha and explained to them why executing these actions systematically is important (business benefit).

The recognized challenges: the researchers recognized that Alpha uses different incident categorization sets (sets of values for categorizing incidents). The lack of a consistent incident categorization set means that incidents are not categorized similarly inside Alpha. For this reason the same types of incidents may be arranged into different categories. This complicates the consistent measuring and reporting of different types of incidents. **Improvement actions:** Alpha requires an appropriate and consistent software incident categorization model in order to categorize incidents in a systematic way throughout the business unit. This model will help to analyze and compare different types of incidents and their frequencies inside Alpha (and between other business units if they implement the same software incident categorization model). **Business benefits:** by using an appropriate software incident categorization model, Alpha is able to design clear and measurable objectives for incident management. The measurement results can be used to identify areas or activities which cause delays in incident management. For instance, these results can show that the resolution times in network-related incidents are much longer than the resolution times for other types of incidents or lots of incidents were initiated during a testing phase (which may indicate that the testing is not executed properly). Regularly reviewing effectively categorized incidents on the basis of priorities and underlying causes could help to identify opportunities for continual service improvement, increase the quality of IT services, and improve customer satisfaction. A systematic model for managing improvement actions concerning IT services and IT service management processes have been presented in Heikkinen's and Jäntti's paper [18].

B. Designing a software incident categorization model based on ITSM practices

The researchers designed the software incident categorization model by using the ITIL technique [1], which can be applied to creating a complete set of incident categories. This technique contained the following steps:

1) Organize brainstorming sessions. Appropriate stakeholders should be invited to the sessions (e.g., service desk managers, incident managers, and problem managers).
2) Create the main categories for incidents by using the information collected during Step 1. Additionally, add an "Other" incident category.
3) Test the main categories which were created in Step 2. Testing should last a sufficiently long period of time for the appropriate amount of data to be collected.
4) Analyze the data which were collected during the Step 3. The successfulness of the main category is determined by the number of incidents that have fallen into it. Additionally, analyze incidents which have been categorized as "Other" incident. If the "Other" incident category contains a large number of incidents, form new main categories for these incidents on the basis of similarities found.
5) Execute a breakdown analysis of the incident categories that have been created. The purpose of this analysis is to review the main categories and design appropriate subcategories for them.
6) Repeat Steps 2 to 5 for an appropriate period of time (approximately, from one to three months). Review the categories and subcategories regularly to ensure that they remain relevant.

The data sources that the researchers collected, analyzed, and used while executing these six steps are presented in Section II.

 ISBN: 978-1-61208-304-9

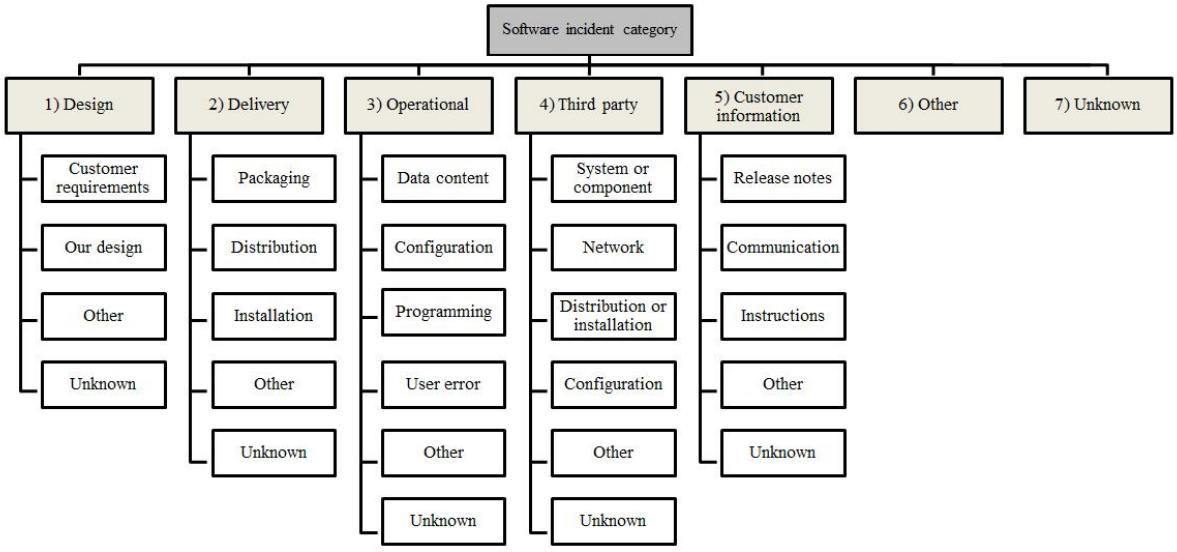

Fig. 1. The software incident categorization (SIC) model

C. Presenting the main categories and subcategories of the software incident categorization model

The software incident categorization model that was created offers a consistent and practical means of incident categorization. The SIC model is hierarchical and it consists of seven main categories and twenty-six subcategories. The model is not bound to any specific software or business unit. Figure 1 presents the structure of the software incident categorization model.

The model includes the main categories ”Other” and ”Unknown” (categories six and seven). Additionally, the main categories from one to five contain subcategories ”Other” and ”Unknown”. In practice, the ”Other” and ”Unknown” main categories and subcategories are meant to be used in the following way: the ”Other” category contains incidents that cannot be classified into the other categories and the ”Unknown” category will be used when the right classification category for the incident is not (yet) known. The list below presents the software incident categorization model's main categories and subcategories in more detail:

1) **Design**: this main category contains incidents caused by customer requirements, improper translation of requirements into design, or the poor definition or inadequate specification of software.
   - **Customer requirements**: this subcategory covers incidents caused by inconsistent, incomplete, or incorrect customer requirements.
   - **Our design**: this subcategory covers software incidents caused by the improper translation of requirements into design. Incidents caused by the poor definition or inadequate specification of software also fall into this subcategory.
2) **Delivery**: this main category contains incidents that occur during software delivery or installation procedures.
   - **Packaging**: this subcategory covers incidents caused by software packaging.
   - **Distribution**: this subcategory covers incidents caused by software distribution.
   - **Installation**: this subcategory covers incidents caused by software installation.
3) **Operational**: this main category contains incidents that occur during the normal use of software (e.g., the software behaves incorrectly or it does not work with all inputs).
   - **Data content**: this subcategory covers incidents related to data management (e.g., database incidents, file handling incidents, and incidents related to measurement data).
   - **Configuration**: this subcategory covers incidents related to configuring the software.
   - **Programming**: this subcategory covers incidents related to programming errors. A programming error produces an incorrect or unexpected result, or causes software to behave in unintended ways (code may compile and run without error, but the outcome of an operation may produce an unexpected result).
   - **User error**: this subcategory covers incidents related to errors made by users. A user error results from a mistake made by a user.
4) **Third party**: this main category contains incidents that occur with the use of a third party's software and hardware.
   - **System or component**: this subcategory covers incidents related to third party systems or components which do not behave as they were supposed to.
   - **Network**: this subcategory covers incidents related to the network.
   - **Distribution or installation**: this subcategory covers distribution and installation incidents caused by a third party.
   - **Configuration**: this subcategory covers incidents related to the configuring of the software caused by a third party.

 ISBN: 978-1-61208-304-9

5) **Customer information**: this main category contains incidents that are caused by incorrect or misleading information between the customer and the organization.
   - **Release notes**: this subcategory covers incidents related to release notes (e.g., customers feel that they have not been informed properly about the changes to hardware, software, or other components).
   - **Communication**: this subcategory covers incidents related to communication between a customer and the organization's employees (e.g., the service desk or support specialists).
   - **Instructions**: this subcategory covers incidents caused by written instructions, manuals, or training materials.
6) **Other**: this category contains incidents that cannot be categorized into the previous categories (categories 1 - 5). This category should exist because it helps to understand whether the SIC model works correctly and the categories that have been created are easy to use. This category also indicates whether other categories need expanding.
7) **Unknown**: this category will be used when the right category (1 - 6) for an incident is not yet known.

D. Validating the SIC model

The software incident categorization model that was created was validated by collecting data from Alpha's personnel (e.g., product area managers, problem managers, and service desk employees) using interviews and surveys. The following questions were used to validate the model:

- How does incident logging or managing appear to you in your job? Could you describe a typical incident situation?
- Are the software incident categorization model's main categories and subcategories appropriate and consistent, in your opinion?
- Is there a lack of any categories of the SIC model (e.g., are there any missing main categories or subcategories that you can think of)?
- In your opinion, is the software incident categorization model easy to use? Do you find it easy to discover the proper category for an incident?
- Are the descriptions of the main categories and subcategories appropriate and easy to understand?
- Have you found categorizing incidents challenging? If that is the case, please describe.
- What benefits can be achieved by using incident categorizing?
- Do you have any other ideas on how to improve the incident categorization?

The judge from the validations, Alpha's representatives were pleased with the model and its categories. The personnel were also keen to know when the model would be implemented and ready for use. The following comments were collected during validation meetings:

- The SIC model will help us see the most critical incident sources in software development. We will be able to identify the areas that cause most of the incidents and we can take appropriate countermeasures once these areas have been identified.
- Work was done earlier in small groups when our working practices were not a concern. Today, when work is done in cooperation with several groups, working practices need to be consistent if we want to measure and compare work e.g., from the quality point of view.
- Change and service request types of tickets need to have their own categorization models.
- Using the model (choosing the right main category and subcategory) may be challenging at first if appropriate documentation about the model is not available.
- The "Other" and the "Unknown" categories are useful in situations when it is hard to know the right subcategory for the incident, e.g., when an incident is sent to the service desk, which cannot know for sure what the exact incident subcategory is without the help of support specialists.
- What type of reports can be created by using the categorization data and how can these reports be exploited?

E. Presenting continual service improvement actions

Continual Service Improvement (CSI) aims to continually improve the effectiveness and efficiency of IT processes and services. Measuring the current performance of services and processes is an important factor when identifying improvement opportunities. The SIC model is closely linked to CSI by supporting the measurement of ITSM services and processes. With clear and measurable objectives (e.g., increase number of incidents related to software installation) organization is able to direct its ITSM improvement actions by using incident categorization data of the SIC model. The measurement data can be also used to identify flaws in e.g., incident, problem, and release management processes.

Before the implementation of the SIC model, Alpha should document and validate all the necessary instructions and training materials (e.g., example cases for every category). Alpha should also organize training for its employees to make sure that the SIC model is used properly. It would be wise to arrange regular checks on the SIC model after the implementation to ensure that the model works as expected. In practice, Alpha needs to review how well employees can use the categories and start appropriate improvement actions in case there arises any shortages during the SIC model implementation phase. All the identified opportunities for improvement should be logged in the CSI register, where they are evaluated, approved, prioritized, measured, and reported in a systematic way.

 ISBN: 978-1-61208-304-9

IV. Analysis

In this section, the researchers analyze the research findings in the form of the lessons learned. A source for each lesson is presented using the following abbreviations: DR = documents and archival records; PO = participatory observation, and PA = physical artifacts.

Lesson I: having and understanding consistent IT service management terminology is vital (DR, PO, PA). The researchers discovered that Alpha's personnel do not fully comprehend the actual meaning of an incident and how an incident differs from other support ticket types, e.g., service request. The issue was confirmed in January 2013, when the researchers noticed several dozen different definitions of incidents in the IT service management system. For this reason, Alpha has created several incident categorization sets. Using consistent ITSM terminology makes it easy to recognize what types of support tickets are incidents by nature.

Lesson II: there should be an appropriate and maintainable amount of incident categories (DR, PO). The incident categorization is more useful when it is kept simple. Adding new categories always has to be reasoned. This means that the categorization should help support groups to assign incidents to different categories. The categories should also support incident management analysis and reporting. Help desk personnel may find it difficult to decide which category is the right one if there are too many categories. Besides, if the number of categories grows too large, it is more likely that some of the categories would never be used. An unused category is useless and it has no value in reporting.

Lesson III: the category of the incident should be checked and updated if necessary during the lifecycle of the incident (DR, PO). The details available at the time of the incident categorization may be incomplete, misleading, or incorrect (the "Other" and "Unknown" categories in the SIC model are meant to be used in situations where the incident category is unclear). It is therefore important that the incident categorization is checked and updated if necessary, e.g., during the closure of the incident. The capability to track changes in incident category throughout the lifecycle of an incident may prove useful when looking for potential improvements (e.g., analyzing why the underlying cause of the incident was difficult to identify).

Lesson IV: automation is the key to logging incident information successfully (PO). The work of support group employees should not be slowed down by incident categorization. In practice, support group employees may need to complete several tasks to log an incident (e.g., fill mandatory input fields and choose the right values for drop-down lists). To save time and to make the incident logging process easier, employees may be unwilling to use the SIC model, which is why the incident logging process should be automated as much as possible so that employees' workload does not increase substantially. In addition, customer input for incident logging should be exploited whenever it is possible and convenient.

Lesson V: incident categorization supports continual service improvement (DR, PO). The organization should use reactive and proactive actions during the continual service improvement. From the reactive point of view, incident categorization makes it possible to recognize challenges and shortages in services. Proactively, acting in advance by executing appropriate procedures can be used to guide an organization in the desired direction. Managing and fixing recurring incidents is not effective. The organization should learn from previous incidents and take proper counter actions to ensure that the same incidents will not recur in the future. For example, incidents related to releases need to be monitored and analyzed for a sufficient period of time. The results and conclusions drawn from the analysis have to be recorded and reviewed to identify opportunities for improvement.

V. Conclusion and Future Work

The research subject of this study was an energy business unit, Alpha, which is part of a Nordic IT service company. The research problem of this study was this: what type of software incident categorization model would both be efficient and support ITIL-based continual service improvement? The research work consisted of five main phases: A) investigating the current state of incident management and planning improvement actions; B) designing a software incident categorization model based on ITSM practices; C) presenting the main categories and subcategories of the software incident categorization model; D) validating the SIC model, and E) presenting continual service improvement actions. The result of this study consisted of two parts: one, the software incident categorization (SIC) model which helps an IT organization to categorize incidents effectively and recognize weak points of the software development process, and two, the provision of the lessons learned for improving incident categorization and measurement practices.

The use of a case study and action research methods includes certain limitations. First, the research was performed with one organization, which means that the research work needs to be repeated in other organizations so that the results can be generalized. Second, the study was executed within a short period of time. A longer research period would have provided more detailed analysis of the SIC model and its work in practice. Third, the researchers could have conducted more validation meetings with Alpha's other business units to get a better understanding of whether the SIC model works as expected. Fourth, the purpose of this paper was not to research how the SIC model should be integrated into different ITSM systems. Since SIC model is built on the basis of ITIL v3 practices, it should be easily integrated to the systems which support ITIL. The management (e.g., adding, removing, and editing categories) of the SIC model should be also straightforward in organizations that are already familiar with ITIL best practices.

More studies are needed to investigate how the SIC model categories work and how the SIC model could be expanded to cover e.g., hardware-related incidents. Additionally, future research could concentrate on designing new models to support other ticket types (service requests and problems) by using the SIC model as a starting point.

Acknowledgments

This paper is based on research in the KISMET project, funded by the National Technology Agency, TEKES (no. 70035/10), the European Regional Development Fund (ERDF), and industrial partners.

 ISBN: 978-1-61208-304-9

REFERENCES

[1] Cabinet Office, *ITIL Service Operation*. The Stationery Office (TSO), United Kingdom, 2011.

[2] OGC, *Introduction to ITIL*. The Stationery Office, London, 2007.

[3] Cabinet Office, *ITIL Continual Service Improvement*. The Stationery Office (TSO), United Kingdom, 2011.

[4] V. Vipindeep and P. Jalote, "List of common bugs and programming practices to avoid them," 2005.

[5] J. S. Collofello and L. B. Balcom, "A proposed causative software error classification scheme," 1985, pp. 537–546.

[6] T. Nakajo and H. Kume, "A case history analysis of software error cause-effect relationships," 1991, pp. 830–838.

[7] R. R. Lutz, "Analyzing software requirements errors in safety-critical, embedded systems," in *Proceedings of IEEE International Symposium on Requirements Engineering*, 1993, pp. 126–133.

[8] M. Leszak, D. E. Perry, and D. Stoll, "A case study in root cause defect analysis," in *Proceedings of the 2000 International Conference on Software Engineering*, 2000, pp. 428–437.

[9] H. D. Owens, B. F. Womack, and M. J. Gonzalez, "Software error classification using purify," in *Proceedings of International Conference on Software Maintenance*, 1996, pp. 104–113.

[10] IEEE Computer Society, "IEEE standard classification for software anomalies," 2009.

[11] R. K. Yin, *Case Study Research: Design and Methods*. SAGE Publications ltd, 2003.

[12] R. L. Baskerville, "Investigating information systems with action research," *Commun. AIS*, vol. 2, no. 3es, Nov. 1999.

[13] ISO / IEC, *ISO/IEC 20000-1:2011, IT Service management, Part 1: Service management system requirements*. ISO/IEC JTC 1/SC 7, 2011.

[14] A. Suhonen, S. Heikkinen, M. Kurenniemi, and M. Jäntti, "Implementation of the ITIL-based service level management process to improve an organizations efficiency: A case study," in *The Eighth International Conference on Software Engineering Advances (ICSEA), Paper accepted*, 2013.

[15] K. M. Eisenhardt, "Building theories from case study research," in *Academy of Management Review*, 1989, pp. 532–550.

[16] P. Eriksson and A. Kovalainen, *Qualitative Methods in Business Research*. SAGE Publications ltd, 2008.

[17] N. Denzin, *The Research Act in Sociology*, 1970.

[18] S. Heikkinen and M. Jäntti, "Establishing a continual service improvement model: A case study," in *Proceedings of the 19th European Conference: Systems, Software and Service Process Improvement (EuroSPI)*, 2012, pp. 61 – 72.

 ISBN: 978-1-61208-304-9