Professionals' Overview On Software Change Impact Analysis For Sensitive Applications- A Primary Analysis

Ruchika Malhotra, Sumit Bansal

Abstract: All the security standards suggest change impact analysis (CIA) in the development process of sensitive applications and software. Even though the CIA can be considered as a very basic activity, practical studies about its execution during the development of a sensitive application are still missing. In this study, our objective is the presentation of an automated system in the context of the CIA for sensitive applications based on professionals' overviews. Our analysis proposes that computer and software engineers spend 60 – 80 hrs. on the CIA per year. However, this time duration differs significantly from the phases of development. Moreover, the respondents described different implications to the CIA and supposed the significance of the CIA differently. We present important CIA challenges and several research ideas for the CIA. Additionally, this research work only presents primary analysis, and our work also draws attention to practical importance to the CIA.

Index Terms: Change Impact Analysis, Sensitive Applications, Case Study, Safety-Critical, Interviews, questionnaire

1. INTRODUCTION

Sensitive software and applications change during their life cycle with more effective security techniques and standards. These changes also have an impact on the systems in which these applications and software run; therefore, change impact analysis is required, which highlights the results of any change in the system. The CIA also predicts the modification in the system for compliance with the change [1]. The primary objective of the CIA is the security assurance, and also suggested by the safety standards like IEC 61508 describes that "at any stage of a software lifecycle, if any change is required then CIA will detect the software modules that are going to be affected by this change, and earlier security practices for lifecycle will be revised." More similar standards that the designer needs to stick to while developing a sensitive application are ISO 26262 that is used for the automotive industry, IEC 62304, that is used in the medical industry, etc. Sensitive software and applications are those systems that are life-critical, i.e., they aim at creating safe systems. Any failure in such systems can be detrimental and can result in loss of life and property. A few examples of such software systems are spaceflights, railway signalling systems, airbag systems, braking systems, air traffic control systems, etc. Some other traditional examples of safety-involved software or sensitive applications are compilers, requirements traceability tools, simulators, and test rings. The software that is used for generating the data used by sensitive applications is also considered to be a sensitive application. Moreover, some technologies are required to go beyond the avoidance of collapse, and they are often used in the medical industry, such as dialysis machines, heart-lung machines, medical imaging devices, etc. As the system complexity is increasing at a very fast pace, safety mitigation measures become an essential requirement for safety-involved sensitive applications; such critical systems should be designed such that they fail safely, i.e., the system should not turn off immediately when the power goes off, for instance, in case of a heart bypass machine. Moreover, such sensitive software is constantly changed during their lifetime; hence the analysis of the result of such changes on the overall system performance becomes necessary. In order to identify the modifications required to be made, to accomplish the change, the use of Change impact analysis becomes essential. Sometimes, the CIA is considered as a difficult task to perform due to the sensitive systems [1] [10]. Insufficient CIA activities have been the reasons for vulnerability exploitation in the past [11]. Now, there is a time to get benefits from new CIA techniques, knowledge, and practices to more economic security assurance, to avoid risks and mitigation of attacks. Despite the importance of the CIA for security-critical applications and system, there is a limited research study for the CIA techniques and practices. The available research studies mostly focused on non-sensitive applications and systems and analysed data from the previous work. For instance, Rovegard et al. [2] took views from software engineers to analyse the significance of the CIA. Though Borg et al. analysed the reports based on an industrial control system [6], some implications have been presented in [7] regarding verification and validation of requirements and on traceability in [5]. Nair et al. [8] presented with the safety evidence management methods, highlighting some points about change management. We performed a survey on sensitive applications and systems for the CIA and asked questions from software engineers to get their views on security practices to develop a sensitive application. We interviewed 14 software engineers in two teams of assessment from two different organizations working on sensitive software projects. Our objective is to help technical decisions while developing cyber-security applications or physical systems, in which the CIA has its importance. To conduct these interviews for the sake of research, we mainly asked the following three questions- (Question-1) how difficult is the CIA work activity? (Question-2) what is the software engineers' behaviour towards the CIA? And (Question-3) how to support the CIA for the completion of the engineer's tasks? The remaining of the

- *Ruchika Malhotra is Associate Head and Associate Professor at the Department of Software Engineering, Delhi Technological University (formerly Delhi College of Engineering), Delhi, India. E-mail: ruchikamalhotra@dtu.ac.in*
- *Sumit Bansal is currently pursuing master's degree program in software engineering in Delhi Technological University (formerly Delhi College of Engineering), E-mail: sumitbansal6816@gmail.com*

paper is organized as follows. In Section II, we describe the research methodology used. In Section III, we provide the case description. In Section IV, present the results of the study and discuss results, and finally, section 5 concludes this study.

2. RESEARCH METHODOLOGY

We formed a questionnaire that can be used to survey multiple industrial sites, and it should not deviate from its context. The questionnaire is constructed such that the three questions mentioned in Section I of the paper, are addressed in a detailed manner. We took two development teams to establish assessment units, referred to as Team 1 and Team 2. Figure 1 describes the research process, where smiley presents the number of researchers involved in each process.

The following steps were involved in the research methodology;

(1) Three researchers repeatedly designed the questionnaire in a case-study protocol format. And all the phases in the design were keenly reviewed by experienced researchers. We used an interview instruction set (available online [14]) that assists to ask open and close-ended questions. We started with open questions and ended interviews with the time glass interview model [3].

(2) The process of interviews from engineers is processed in the English language. For the reason of secrecy single researcher from the team did interviews.

(3) The same researcher did the assessment on conducted interviews and forwarded them back to the interviewees for acceptance.

(4) We interviewed fourteen professionals in team-1 of whom nine are software developers that develop sensitive applications and write their documentation. We also interviewed an R&D manager specifically, two security engineers, and two senior developers. On the other hand, in team-2, we interviewed eight professionals, four software developers, two senior developers, one technical manager, and one product manager.

(5) For the primary analysis step, the acceptance report is processed further, and long answers are divided into the bullets. Unnecessary answers were removed, and the analyzed report was filtered further for the Coding and debugging context. The acceptance report was based on the following scheme explained as follows; A 2-D coding scheme that used the intent of CIA and its importance along the two axes, taking both negative and positive range of values to show the result was followed in Pre2.a; while Pre2.b used the frequency of conducting CIA activity as a measure; a time axis was used in Pre2.c; Pre2.d coding method deals with some of the more difficult challenges of CIA; the time and the difficulty level after modifications addressed in Pre3; and Pre4 open coding is based on the support system provided the change impact analysis detection. Each of these the coding scheme explained above are represented by their respective IDs (used for the convenience of understanding) as listed in the Table-1. Here "Pre" notation is used to emphasize that it is the pre-reporting process.

(6) Finally, in the reporting process, we finalized the report, for the matter of secrecy and confidentiality; we hide traceability from the answers.

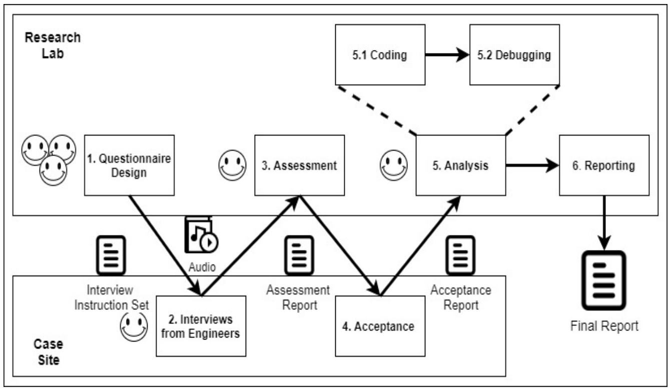

***Fig. 1**. The layout of the Research Process*

This paper consists of a part of the questions inquired. Table 2 presents the subset of the questions from the interview instruction set.

***Table 1**. Categorization of questions to specific parts of the interview instruction set*

Questions	ID	Interview Instruction Set
How difficult is the CIA work activity?	Pre2.b	Do you do CIA activity daily/weekly/monthly?
	Pre2.c	What is the duration of time spent on the CIA?
	Pre2.d	What are the CIA problems?
What is the software engineers' behaviour towards the CIA?	Pre2.a	What are your views on CIA activity?
How much the CIA is supported in the completion of engineers' tasks?	Pre4	What type of assistance do you need in the CIA conduction?
	Pre3.a	Would you like to comment on the parameters, we have gathered from previous CIA conducted activities?
	Pre3.b	How much it would take to complete the CIA process?

3. CASE DESCRIPTION

Sensitive applications are developed to run on sensitive or highly secured systems. It is necessary to properly examine the source code before implementing it to the system in order to develop good quality software. Additionally, detailed documentation should be formed for future reference and abstraction. A process of project completion needs collaboration from engineers because every project consists of over one million lines compiled in different computer languages. After the completion of the development process, the organization requires to set a security case for an external audit, to achieve a secure and safe system while operating in any given environment. The CIA analysis is a critical step for the security case. However, security engineers developed a CIA report template, as presented in Table 2. The software developers follow the template and conducted and documented their CIA before the changes to the source code are made. This template is derived from Klevin [4].

***Table 2**. CIA template used in the company, derived from Klevin [4].*

(Q-1) Is the problems security critical?
(Q-2) In which version of the software/application problem exists?
(Q-3) How sensitive systems will be affected by this change?
(Q-4) Enlist changed code modules and files
(Q-5) Which firmware is affected by the change?
(Q-6)After committing a change, which documents should be modified? (e.g,specs, architecture)
(Q-7) Which test to ensure security should be executed? (e.g., design/functional/sequence tests)
(Q-8) How long it will take to resolve the problem?
(Q-9) What is the major reason for the problem?
(Q-10) What all requirements and specifications of functions are required to be retested?

There are limited tools available in the organization to conduct the CIA. This conduction is completely linked with the issue management process. And all of the finished CIA reports are secured in an issue repository in form of attachments for issue reports. Moreover, in the scope of this research, there is a web interface available within the organization to access the issue repository.

4.RESULTS AND DISCUSSION

We investigate and present results based on the survey, as will be discussed in the following subsections.

Question 1: How Difficult is the Change Impact Analysis activity work?

One of the key questions that require a well-thought answer is the difficulty face by the experts while performing the CIA. To understand the difficulties that professionals experience during the CIA work activity, we inquired about the occurrence of CIA activity by the professionals, the time required to complete the CIA activity, and the challenges experienced by the professionals during the CIA activity. Each of these will be discussed below. The occurrence of Change Impact Analysis- All the professionals, have stressful experience with the occurrence of the CIA activity that varies from daily to weekly to monthly, depending upon the development phases. Several professionals reported that they conducted the CIA activity daily, while some other professionals reported that they conduct it on a weekly basis during the most extensive time of the development. A senior developer in team-1 estimated that an ordinary software developer may conduct 23 CIAs related to debugging every year. A security engineer in team-2 estimated a higher number: “in the high time of the project, a junior developer conducts CIA activity daily.” The reason behind this variation is a phase-wise development process. In the initial phases of the project, the source code of underdeveloped software is churn, and the individual problems are not managed with change. Once a project completes some milestones and debugging, and testing phases are initiated, the objective is to ensure the quality of the product, and all changes after this phase are related to debugging. Effort in Change Impact Analysis - When asked about the average time spend on the CIA activity, five professionals responded with an average CIA effort. Figure 2 depicts a general view of the data collected. Similar to the occurrence results, CIA efforts varies significantly with the time required. A senior professional expressed his view that the effort may depend upon the complexity of the project phase, while others expressed that it depends on the documentation involved in the development process.

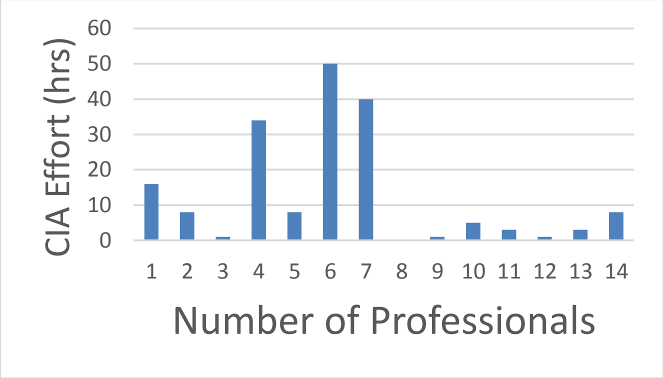

***Fig.** 2: Professionals reported CIA effort.*

Three professionals reported that the average CIA effort is an hour or two, four professionals expressed 5-6 effort hours, and three professionals expressed a day or two. On the other hand, three professionals reported that the complete CIA activity is the matter of not more than an hour, more specifically, 40-50 minutes. Nine of the professionals reported that at least 35 minutes or less are required to finish a CIA, of which two reported 10 minutes or less. Two senior developers in team 2 expressed that swift CIA necessarily requires an hour or two. In the context of the maximum time, five professionals expressed 2-3 days, three reported 7 days, and a junior developer reported up to many months. We filtered three main reasons for this variation. First, it is difficult to delink the CIA from the general issue resolution; therefore, the professionals reported their answers differently. The explanation of this work involved in regenerating and understanding a problem connected with the CIA. To resolve a problem completely, professionals often need to set up a specific test environment, debugging, and testing source code. Challenges to CIA- There are 30 significant general and specific challenges reported by professionals presented in Table 1. It contains some questions that deal with management issues like Q-9 and Q-10. They mostly occurred challenge is related to motivation, like understanding the need of the CIA in the process of sensitive application development. A senior developer reported that “my primary task is to describe and motivate why we do CIA; however, we regularly remind ourselves why we do it. Furthermore, if the developer starts to know the importance of the CIA, it helps me in my work”. The second most critical challenge is the information overloading; the senior developer reported that it is hard to understand the system due to its complexity. There is a massive number of documents presenting the system except for the source code. A software developer stated that: “looking for the right information is a major challenge and we need to find the key people and query for dependencies and other documentation, after all this, we do not get the answer.” There are several other challenges worthwhile to mention, sometimes, software developers and engineers require previous CIAs documents that were processed long time ago. Resolving old issues is cumbersome and needs time and effort. Another important challenge is to establish trust in conducted CIAs to answer the questions in Table 1. Some professionals reported that they are not sure whether their answers are right or not, how could we evaluate it. Lastly, some interviewees said that the major challenge is CIA guidelines are not followed by the developers.

Question 2: What is the software engineer's behaviour towards Change Impact Analysis?
We highlighted the professional implications to "Change Impact Analysis", i.e., the behavioural connectivity towards the CIA. Moreover, we connect the implication to the impression of how significant the CIA is for the professional, depicted in Fig. 2.

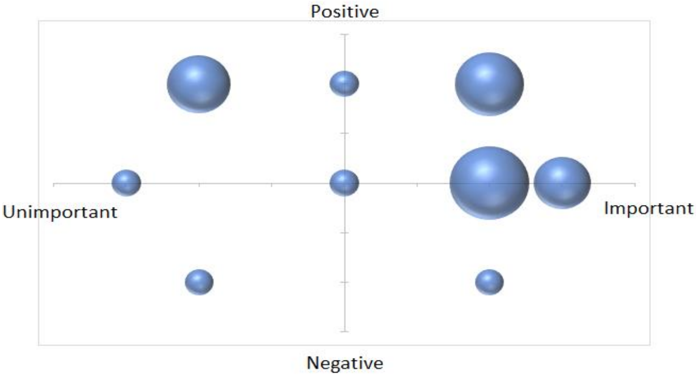

Fig. *3: Professional's behaviour towards the CIA*

The professional's behaviour to the CIA described in all quadrants in Fig. 3. There are not many strong implications to change impact analysis. It was observed that a large number of engineers regard their CIAs essential. Several professionals reported positive associations, e.g., "a positive sign" and "shows that we do complex software engineering." Also, several professionals had neutral implications: "no values on a personal level" and "just part of the job." Two professionals reported negative implications. They consider the CIA is a too heavy or rigid activity, and need a better way to perform it. Considering the professionals' supposed significance of the CIA activity to their work, all levels are covered. There is a range of responses from "for some issues, it is just worthless stuff, done for the process" to "professionally, it is very fundamental" and "side effects are extremely important in our complex product". We observed no indications of any behaviour differences between the two groups. It is identified that the senior professionals have a slight leaning towards the importance of the CIA when compared with the juniors. This comes from the fact that the seniors had seen more cases of side effects being caused as a result of source code changes, from earlier development work when the CIA was less formal, as explained by a senior engineer "we have seen many cases when fixes introduced bugs".

Question 3: How to support the CIA for the completion of the engineer's tasks?
CIA Support proposed by some professionals - When asked for potential methods to support the CIA, the tool support was the most frequent answer by the software engineers. Three professionals proposed that the level of automation in the CIA can be increased by the introduction of some tool support. A possible explanation is that professionals are already well-aware of the efforts being put on developing the traceability in the company, and they believe it is not being utilised up to its potential. One proficient clarified how a tool may be traceable through a framework, from an input source code file to a design depiction, and then the resulting tests, at that point, proceed up the abstraction layers to utilitarian details and up to the requirements. This permits to have the identification of the test straightforwardly from the requisites, as a tool-based arrangement that can happen near the regularlyThe challenging loophole between the requisites and the test cases. Another scheme, provided by some lately appointed senior experts, is that it's unusual to link a free-text string of a CIA report with some third party device, rather this can be done by using a tool developed inside the company to have a way better command over the input. Measuring CIA Support-The CIA history stored inside the issue tracking system was explored, and the two possible measures were developed: TIME and MODS. These measures were estimated for suitable interviewees, and observations were presented during the interviews. The first measure, TIME, is defined as "the time between a developer being assigned an issue and the first CIA report being submitted". Hence TIME focuses the effort needed to perform CIA, but none of the professionals considered TIME to be related to the amount of time taken to conduct a Change Impact Analysis. Six experts straightforwardly refuted this measure, while two of the professionals could not understand how to decipher it. The foremost doubtful views clarified: "it's definitely a question of priorities" "I work on several parallel products, and that measure can be anything", and "you have to measure when I begin making related changes". It can be concluded that TIME is a very disconcerted measure to be utilized in assessing and analysing the solutions that point at diminishing the time required to perform Change Impact Analysis. Another measure, called MODS, can be expressed as the total number of modification on a CIA report after it has been first submitted. We recommended utilizing it as a proxy measure for the trouble faced while finishing a specific CIA; hence it focuses on CIA exactness. Three of the experts were somewhat positive to the measure, one of them explained that: "more modification implies a difficult CIA. And it implies that you couldn't promptly capture everything. While, opposite to this, two experts toppled the measure completely by communicating that various changes involve only the typos and copy-paste blunders. Conclusively, TIME shows up to not at all be connected with the time required to perform CIA. While MODS received some support, many also expressed that it is bounded by trifling changes to the extent that it can't be entrusted. While the two proposed measures can be considered simple to gather from the logging timestamps as well as the revision history of most of the systems, some enhancements are needed in both of them. While TIME must arise as a result of actual changes concerned with an issue and not based on when that issue is relegated, MODS should be modified to expel trivial changes like spelling and grammatical adjustments.

5. CONCLUSION AND FUTURE WORK

This study focuses on the professional's overview of Change Impact Analysis. We interviewed professionals in two teams of analysis in the context of sensitive software and applications run on security-critical systems. We concluded that efforts and time require to complete CIA activity vary, that dependent on the phase of the underdeveloped software with the complexity of required change. According to our results, software developers and engineers working in a sensitive organization spend around 60 – 80 hrs on CIA activity. Some senior engineers also shared their experiences that the CIA can take 10% of the overall project's time to solve a normal problem. We also mentioned some CIA challenges that include communication with the team working on a project and documentation of developed source code. We also concluded that the CIA is a significant but expensive activity in the sensitive software development environment. We also observed the professional's behaviour towards the importance of the CIA. We also proposed CIA improvements that include the usage of additional tools, optimized traceability, individuals, and training programs. It is a little difficult to measure the value of CIA quantitatively because it is a secluded activity; however, it is well connected with development and management problems in a security-critical project. There is a need to analysed CIA activity for quantitative measures in the future. Our primary analysis also has some limitations; the professionals are obliged to follow the Non-Disclosure Agreement (NDA) due to the security reasons and policies of the organization. The interview and assessment activities were done by the first author (see Fig. 1). The acceptance step is done by another researcher. Additionally, we are focusing on the acceptance of our analysed report. Another limitation to the experiment is that the results are concluded on the views of a limited number of practitioners; therefore, we plan to add more number of participants to the study to further increase the validation of our results. Moreover, the survey experiment is conducted only in one language i.e., English, this may act as a language barrier to some of the potential interviewees, thus we further plan on taking the interviews in more than one language, involving people from different countries, to overcome this limitation. While evolving a sensitive software or application based on architectural or design decisions, the CIA is a valuable activity for the decision-makers. Our survey report highlights that professionals put an extensive time to the CIA's activities, and generally give importance to their content.

REFERENCES

[1] Bohner, Shawn Anthony. "A graph traceability approach for software change impact analysis." (1996).

[2] Rovegård, Per, Lefteris Angelis, and Claes Wohlin. "An empirical study on views of the importance of change impact analysis issues." IEEE Transactions on Software Engineering 34.4 (2008): 516-530.

[3] Runeson, Per, Martin Host, Austen Rainer, and Bjorn Regnell. Case study research in software engineering: Guidelines and examples. John Wiley & Sons, 2012.

[4] Klevin, Artour. People, process and tools: A Study of Impact Analysis in a Change Process. Department of Computer Science, Faculty of Engineering, LTH, Lund University, 2012.

[5] Regan, Gilbert, Fergal McCaffery, Kevin McDaid, and Derek Flood. "Investigation of traceability within a medical device organization." In International Conference on Software Process Improvement and Capability Determination, pp. 211-222. Springer, Berlin, Heidelberg, 2013.

[6] Borg, Markus, Orlena CZ Gotel, and Krzysztof Wnuk. "Enabling traceability reuse for impact analyses: A feasibility study in a safety context." 2013 7th International Workshop on Traceability in Emerging Forms of Software Engineering (TEFSE). IEEE, 2013.

[7] Bjarnason, Elizabeth, Per Runeson, Markus Borg, Michael Unterkalmsteiner, Emelie Engström, Björn Regnell, Giedre Sabaliauskaite, Annabella Loconsole, Tony Gorschek, and Robert Feldt. "Challenges and practices in aligning requirements with verification and validation: a case study of six companies." Empirical software engineering 19, no. 6 (2014): 1809-1855.

[8] Nair, Sunil, Jose Luis de la Vara, Mehrdad Sabetzadeh, and Davide Falessi. "Evidence management for compliance of critical systems with safety standards: A survey on the state of practice." Information and Software Technology 60 (2015): 1-15.

[9] Abdulkhaleq, Asim, and Stefan Wagner. "A controlled experiment for the empirical evaluation of safety analysis techniques for safety-critical software." Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering. 2015.

[10] de La Vara, Jose Luis, Markus Borg, Krzysztof Wnuk, and Leon Moonen. "An industrial survey of safety evidence change impact analysis practice." IEEE Transactions on Software Engineering 42, no. 12 (2016): 1095-1117.

[11] Leveson, Nancy G. Engineering a safer world: systems thinking applied to safety. The MIT Press, 2016.

[12] Musco, Vincenzo, Antonin Carette, Martin Monperrus, and Philippe Preux. "A learning algorithm for change impact prediction." In 2016 IEEE/ACM 5th International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering (RAISE), pp. 8-14. IEEE, 2016.

[13] Angerer, Florian, Andreas Grimmer, Herbert Prähofer, and Paul Grünbacher. "Change impact analysis for maintenance and evolution of variable software systems." Automated Software Engineering 26, no. 2 (2019): 417-461.

[14] Online]: http://serg.cs.lth.se/fileadmin/serg/ImpRec_EvalStudy/ImpRec_InterviewGuides.pdf.