ISSN- 2394-5125 VOL 7, ISSUE 15, 2020

STUDY ON IMPACT ANALYSIS FOR REQUIREMENT CHANGE OVER SOFTWARE SYSTEM

SATHYARAJASEKARAN K[1], GANESAN R[2]

[1,2]School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India.

E-mail: sathyarajasekaran.k@vit.ac.in

Received: 28.04.2020 Revised: 30.05.2020 Accepted: 27.06.2020

Abstract— Change Impact Analysis (CIA) is becoming a significant feature in all areas of software development particularly. When a software project is designed, so many issues occur in the development process from the analysis of the problem to the implementation of the program, where such issues contribute to project failure because the project cannot be implemented at the right time. The explanation behind this is that evolving software requirements constantly affects, and new user requirements that change more frequently. The key factors concealed in this are the specifications which are recorded manually, and the issues of reliance on the specifications are also done manually.

Keywords— Change Impact Analysis, Design Phase, Impact Analysis, Requirements Phase, Software Engineering, Software Systems.

1 INTRODUCTION

Nowadays, the software systems generally function in an active business environment where goals of business frequently vary. Consequently, the needs of software systems vary constantly and novel requirements appear repeatedly. Thus, repeated changes have been clearly stated as important for software systems, which were employed in an ever-changing perspective. Owing to the significance of changes following the primary improvement of software, software maintenance has turn out to be gradually more appropriate for companies, researchers and entire developers. Several analyzation point out that companies nowadays pay out the majority of their software costs, up to 50% and 70%, on maintaining their software systems. The combination of the employed software systems and the new requirement adaptations are considered as time consuming and costly. Therefore, even a slight variation in a particular requirement might have a considerable impact on the entire system. Portraying such an effect is generally known as the Change Impact Analysis.

2 RELATED WORKS

In 2018, Heinrich et al. [1] have adopted a new KAMP4aPS method for architecture-oriented CIA in production mechanization. Accordingly, several meta-models were introduced for specifying a variety of system artifacts and in addition, the modifications made to rectify them were introduced depending on the models along with the rules and techniques for change propagation analysis. Further, the established model was estimated for three varied change scenario depending on the developed “xPPU community” case study on production mechanization. Here, miscellaneous compositions of meta-models and change propagation methods were investigated thoroughly. Finally, the examination outcomes point out the accurateness of the proposed change propagation methodology by deploying the “KAMP4aPS model” with respect to the implemented rules in a precise and efficient manner.

In 2018, Hajri et al. [2] have offered a new approach, which has applied and assessed with a CIA methodology for developing decisions regarding the configuration in PL use case approaches. Accordingly, the presented scheme has comprised of two statements, initial one was the programmed support to recognize the influence of decision variations on previous and following decisions in PL diagrams; and subsequent one was the programmed incremental restoration of PS approaches and developing decisions regarding the configurations. The supported tool was incorporated with “IBM Doors” and the offered model was computed in an engineering case analysis, which in turns confirmed that it was beneficial and practical to examine the changes impact decisions and to redevelop PS use case approaches incrementally in developed settings.

In 2018, Ks, Rg,[3] have offered a systematic study on literature in the field of Change Impact Analysis for the software development process, the work concentrates on requirement phase and design phase impact analysis study by covering the three types of impact process namely dependency, Trace and Experiential. Evaluating all the procedures from finite state machine to machine learning.

In 2014, Arda et al. [4] have modelled a scheme that has intended to enhance the CIA in requirements by deploying requirements based on changing formal relations types and requirements semantics. In the preceding work, a requirement meta-model was presented with frequently deployed formal relations types and requirements semantics in 1st-order logic. Here, the categorization of requirement variations depending on the modelling of a textual requirement was offered along with the proper semantics. The formalization of changes and relation’s requirements was deployed for proliferating the adopted changes and the consistency of implemented changes were examined in requirements models. Here, the support for implementing the CIA methodology was found to be an extension of the TRIC tool.

In 2014, Sun et al. [5] have presented a scheme to address the issues in CIA, so that the modifications to be done in software could be determined. CIA could be deployed for taking accurate decisions on the change applications, i.e. unpredictability evaluation, and to execute effectual modifications for a change proposal. Here, in this work, an experimental analysis was carried out on three publically available Java systems for demonstrating how CIA could be exploited for modifying the software. The outcomes have pointed out three facts, initial one was the evaluation of the unpredictability of a change proposal depending on the impact of CIA outcomes that was not correct on concerning the precision; subsequent one was that the implemented impact measure that indicated the effectual changeability evaluation for the changes made; and third one was that the CIA could formulated the change operation easier and efficient.

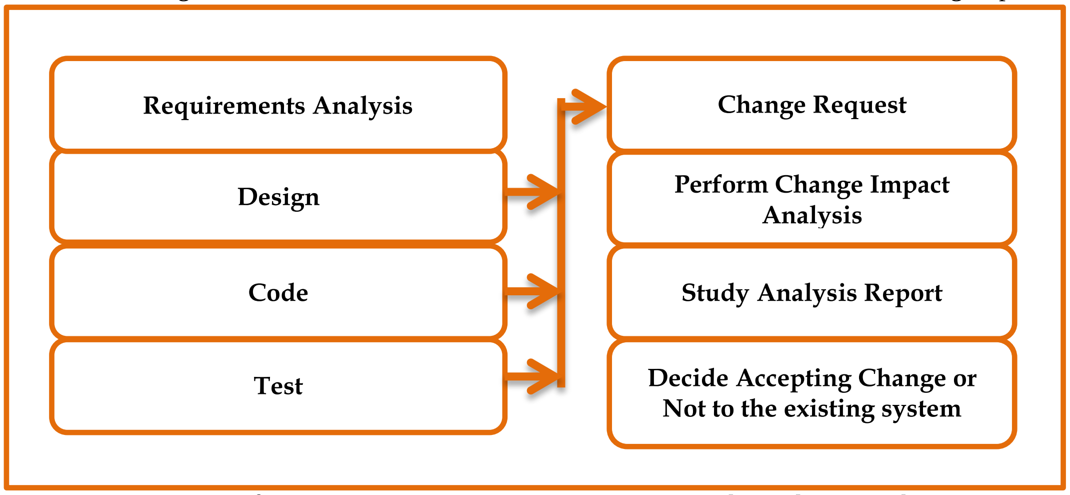
Requirements Analysis
Design
Code
Test
Change Request
Perform Change Impact Analysis
Study Analysis Report
Decide Accepting Change or Not to the existing system

Fig 1 Managing of new requirement to an existing system Traditional Approach

In 2013, Dam and Aditya [6] have suggested a scheme regarding the CIA using two different methodologies for the renowned and extensively developed BDIA systems. Accordingly, the presented static method has evaluated the change impact by examining the source code and recognizing a variety of dependence within the system. In addition, the presented dynamic method has constructed a design based on the agent's performance by examining its implementation traces that has comprised of plans and goals, and it has deployed this design for estimating the impacts. Further, the presented approaches were implemented and analysed over the investigational results, which has thus evaluated their efficiency in practice.

In 2016, Jihen et al. [7] have introduced an automated technique, which examined the feature model development of change and their impact traces on the SPL model, and it further provides numerous suggestions for making clear about the reliability of both schemes. The suggested technique portrays various novel measures regarding the SPL development for recognizing the attempts that were required to continue the SPL schemes constantly with a better quality. The techniques and its implementations were demonstrated via a case point of SPL in the domain of text editing. Furthermore, they were computed experimentally with respect to both the precision and quality of the concerned SPL models of the CIA.

In 2015, Cai and Raul [8] have systematically examined the analytical accuracy of dynamic CIA model. Here, the accuracy was assessed with huge amounts of repository and artificial changes and it was identified that dynamic CIA could go through both low recall and low precision. The accurateness for distinctive repository changes has to be minimal over the arbitrary changes. Reduced time consumption for implementing the programs can be related with an even minimal recall. The accuracy of software got exaggerated by its unvarying changes and as a result, the analysts make use of CIA to be aware of the possible outcomes of changing their software at an early stage. Dynamic CIA was a realistic method, which recognized the probable change impacts for delegate implementations. However, it was unidentified how consistent its outcomes were, owing to their improper analysis on accuracy. This research work offers the initial wide-ranging analysis regarding the analytical accuracy of dynamic CIA in two corresponding methods. Initially, enormous counts of random changes in several Java applications were used to cover up the entire feasible locations of changes. Subsequently, above 100 changes were analysed from software repositories that were considered as the representatives of developer trainings. The introduced investigational scheme deployed an analysis over the execution differentiation and sensitivity analysis for measuring the metrics like recall and precision of dynamic CIA systematically with regard to the corresponding impacts that were noticed for these changes. The outcomes for both classifications of change demonstrated that the majority of the cost-effective dynamic CIA methodologies were imprecise with a standard accuracy of 38-50% and recall of 50-56% in many cases. Furthermore, this wide-ranging analysis have provided a clear description on the efficiency of the implemented dynamic CIA model and it further offered the research challenges that have to be look forwarded in the upcoming years for more precise impact analyses.

In 2013, Li et al. [9] have established a novel call graph-oriented CIA method that considered the interventions found between the numerous adopted changes for enhancing the accuracy of the impact outcomes. The introduced CIA model was motivated from the observable facts of "water wave propagation". Further, this process was compared to the procedure of “water wave propagation”. Initially, the "core" (a particular types of techniques) produced by the introduced changes was identified. Furthermore, the ripple effects were computed through propagation study on this core. Moreover, experimental assessments on two real-world datasets illustrated the efficiency of the established CIA scheme. The outcomes demonstrated that the presented method could predict more impact changes, under the condition, when the majority of changes were identified by the system. In addition, the presented model was also found to offer numerous ways for realistic applications. Furthermore, it could successfully eliminated certain negative metrics such as, false-negative measures and false-positive measures when distinguished over the conventional CIA (call graph oriented) methodologies.

In 2017, Haoues et al. [10] have offered a new scheme based on the FC impact in use case diagrams for designing the FUR. It made a distinction among external and internal FC to a use case. It intended to measure the impact of FC with respect to the CFP units. Moreover, it computed the rank of FC based on the size of functions. In addition, an algorithm was proposed for ranking the changes, when there was more than a single functional change. The established model was based on various heuristics, in which the main intention was to reduce the attempts necessary to respond the changes. It considered the majority of significant constraints during the change prioritization process (size of FC functions, and the first choice of the change campaigner). These two constraints were recognized depending on a study with the COSMIC society.

In 2018, Ks, Rg,[11] have mapped an architectural pattern for Impact Analysis study, the proposed work has used Model View Controller pattern in performing the impact analysis study on each individual categories over classes, the automation of class building is been invoked so that the classes sent for Impact Analysis study is minimal which enhances the efficiency of Impact study.

In 2015, Chen et al. [12] have developed a novel object- based, attribute- oriented technique for comprehensively and effectively carrying out the tasks of CIA in variant product model. This technique designed the incorporated content of a product by differentiating its related requirements and components with linkages and attributes. In addition, so as to manage with the recursive and dynamic CIA loops, it featured a change propagation design based on object. This scheme was also capable to demonstrate the joint impact when numerous changes were attempted. A programmed model, "EPCII-EC", was executed for realizing the scheme, and a descriptive product case with a relative assessment was offered so as to confirm the work. Furthermore, the limitations and issues existing in this analysis were also examined and further proposals were offered for upcoming studies.

In 2013, Li et al. [13] have developed a *FCA–CIA* model, which deployed *FCA* for generating an intermediary demonstration of program depending on the static analysis. This demonstration was known as the *LoCMD* approach. *Here, the presented model* considered the changed classes as a whole, and it discovered the available set based on the changes on *LoCMD*. Depending on the hierarchical nature of *LoCMD*, the presented techniques were ordered in terms of the impact measure. In addition, the experimental assessments on four real-world datasets have revealed the effectiveness of the presented *FCA–CIA* system.

In 2016, Cai et al. [14] have implemented a new dynamic-CIA method known as SensA that combined execution differentiation and sensitivity analysis. The developed model offered fine-grained impact sets and it moreover ranks the possible impacts by means of "semantic-dependence quantification" for program slices. In addition, the advantages of impact prioritization was computed by deploying SensA on concerning the dynamic and static forward slicing through an wide-ranging experimental analysis on three publically available Java case studies and applications. Here, the outcomes demonstrated that SensA could present much improved cost efficiency over slicing model in helping the analysts with impact assessment.

In 2006, Ren et al. [15] have suggested a scheme based on semantic CIA mode that chiefly concern on removing the errors and bugs. Throughout the program maintenance process, a programmer might formulate changes, which improved the functionality of program or remove the errors in code. Further, the programmer generally performed regression or unit tests to avoid the cancellation of previously examined functions. If a test becomes unsuccessful suddenly, the programmer has to look at the edits to discover the causes of failure for that test. Here, a tool named as Chianti (a tool, which carried out semantic CIA) was adopted, to permit the programmer to scrutinize the edited parts that have an effect on the failing test.

In 2008, Rovegård et al. [16] have established a novel method regarding the role of CIA in change management process. Here, issues of impact analysis were ranked based on the significance assigned by software specialists from the self viewpoint and organisational viewpoint. The software specialists usually belong to three organisational stages: (i) operative, (ii) tactical and (iii) strategic. Statistical and qualitative analyses were also offered in terms of varied differentiations among levels as well as perspectives. The outcomes demonstrated that significant issues for a specific level were closely associated with how the level was described. Likewise, the problems significancy from an organisational viewpoint was more holistic than those significant from a self- viewpoint. In addition, the presented model has pointed out that the self- viewpoints colours the organisational viewpoint, indicating that personal attitudes and opinions could not be ignored easily. On evaluating the levels and the perspectives of the adopted model, the differences were visualised such that it has allowed us to describe the two classes of issues such as medium-priority and high-priority. The main significant issue from this viewpoint was taken as the fundamental aspects of CIA and its implementation.

In 2019, Ks, Rg,[17] has derived an Impact Analysis study on Software Readiness that helps in performance of adapting patches and version building which derives the concept of providing support in maintenance phase of the system.

In 2015, Teufl and Georg [18] have introduced a scheme for recognizing the change impacts in a requirements specification in an effective manner. This research work implemented a model-oriented scheme, which categorized the components of requirements specification based on their contents and also based on the candidate elements, which were probably exaggerated by the change. This work demonstrated the automation application of the approach through a PPU instructor. The technological confirmation illustrates that in the case of PPU, the CIA was more proficient and the challenges on impact analysis were further minimized while incorporating the content model in the domain of mechanized automation systems.

In 2019, Ks, Rg,[19] has proposed a novel process for requirement phase in software development using Impact Analysis by proposing an Impact Transition Model by analyzing the Software Lifecycle Objectives which derive Estimated Impact Set and derive the actual Impact Set over the requirements.

3 OBSERVATIONS THROUGH LITERATURE ON TRADITIONAL METHODOLOGY OF PROCESSING CHANGE IMPACT ANALYSIS

The major findings through the literature study over the methodologies have been listed below
1 Changes can be monitored by analyzing different source of code dependency. If a component is dependent on any modified component then there is a scope of impact.
2 Requirement changes are prioritized but through fuzzy compatibility.
3 Relationships can be established between the challenges and the improvement areas.
4 Impact values are generated to state the amount of risk involved in implementing the requirement change.
5 Using the process outside the environment it lacks scalability due to the absence of modularity
6 An impacted element provides information on how they were impacted.

4 PROPOSED METHODOLOGY OF PROCESSING CHANGE IMPACT ANALYSIS

Over the observation on existing approach the following are areas to be focused to enhance the efficiency of Change Impact Analysis on Requirements
1 Relationship between the challenges encountered and the system has to be made through a model.
2 Establishing a model to deliver the traceability effect of a change incurred.
3 Design pattern for change impact analysis is a key aspect to be explored
4 Need to analyze the performance of context aware requirement patterns by different methods.

4.1 The following model is proposed to enhance the procedure of CIA

The building of a software system works in the form of
Step 1: perform requirement analysis
Step 2: Identify the Functional Requirements
Step 3: Determine the dependency (Trace/ Dependency) between the functional requirements
Step 4: Design, Code and Test are the traditional form of working.

4.2 Change Request to the existing model

(The system is either in the code phase or test phase or in the market)
Step 1: Arrival of a change request
Step 2: Identify the Functional Requirements
Step 3: Perform the CIA based upon the dependency chart which was built while the project was been built from the beginning
Step 4: Follow the traditional form of design, code and test.

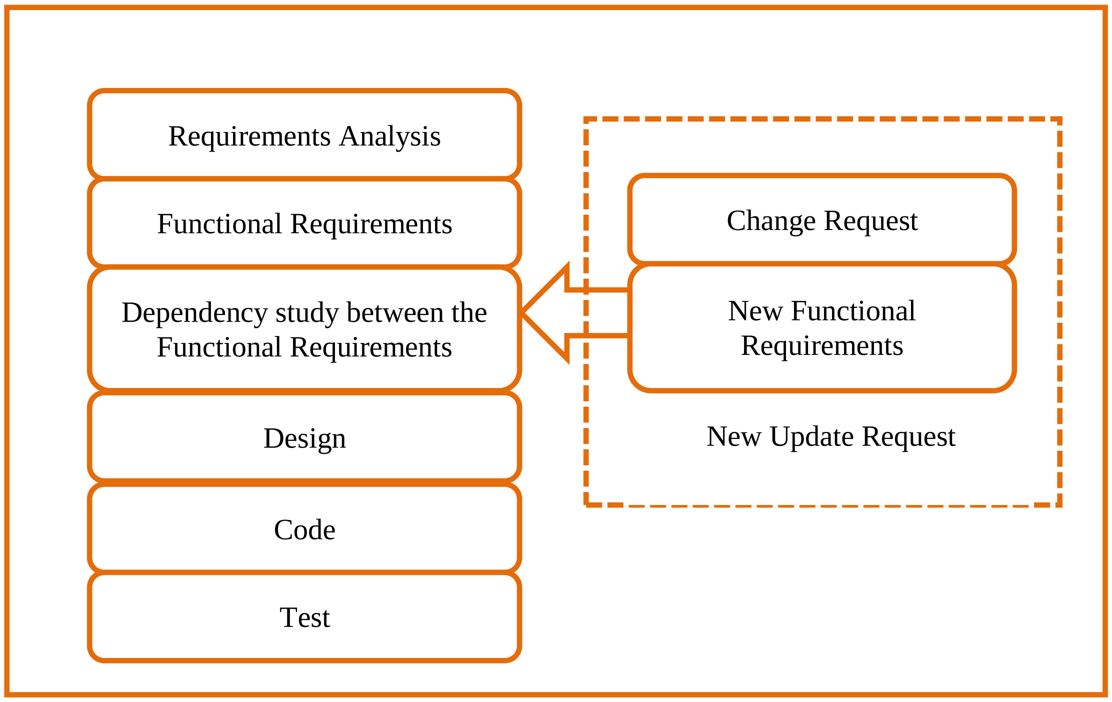

Fig 2 Managing of new requirement to an existing system Proposed Approach

In the proposed model the major advantage is that the performance of CIA is not getting in all the functional requirements of the existing system, it works on the reference of the dependent functional requirements identified through dependency method. This increases the efficiency of the CIA by having reduced workload in the analysis.

5 CONCLUSION

Through this study we found that more methodologies and techniques are available to perform Change Impact Analysis. All the methods and techniques illustrates in carrying the overall functionality of the system in the study of impact analysis of the system. Here we have proposed a methodology to consider only the relevant functionalities for the Change Impact Analysis study. So that the unnecessary functions and all not to be considered for the Impact Analysis study, this will improve the efficiency of the process and produce accurate result. Further the same procedure can be extended to the design phase change and code phase changes also at the same time the experiential technique of Impact Analysis. In the proposed method of dependency tracking of functional Requirement the cost and time spent will be very less when compared to the traditional approach.

6 References

1. Robert Heinrich, Sandro Koch, Suhyun Cha, Kiana Busch, Birgit Vogel-Heuser, “Architecture-based change impact analysis in cross-disciplinary automated production systems”, Journal of Systems and Software, vol. 146, pp. 167-185, December 2018.
2. Ines Hajri, Arda Goknil, Lionel C. Briand, Thierry Stephany, “Change impact analysis for evolving configuration decisions in product line use case models”, Journal of Systems and Software, vol. 139, pp. 211-237, May 2018.
3. Sathyarajasekaran K, Ganesan R., “Change impact analysis (CIA) and its role in analysis and design of software development”, International Journal of Engineering and Advance Technology (IJEAT), Volume-8, Issue-2S, PP 275-283 December 2018.
4. Arda Goknil, Ivan Kurtev, Klaas van den Berg, Wietze Spijkerman, “Change impact analysis for requirements: A metamodeling approach”, Information and Software Technology, vol. 56, no. , pp. 950-9728, August 2014.
5. Xiaobing Sun, Hareton Leung, Bin Li, Bixin Li, “Change impact analysis and changeability assessment for a change proposal: An empirical study “, Journal of Systems and Software, vol. 96, pp. 51-60, October 2014.
6. Hoa Khanh Dam, Aditya Ghose, “Supporting change impact analysis for intelligent agent systems”, Science of Computer Programming, vol. 78, no. 9, pp. 1728-1750, 1 September 2013.
7. Jihen Maâzoun, Nadia Bouassida, Hanêne Ben-Abdallah, “Change impact analysis for software product lines”, Journal of King Saud University - Computer and Information Sciences, vol. 28, no. 4, pp. 364-380, October 2016.
8. Haipeng Cai, Raul Santelices, “A comprehensive study of the predictive accuracy of dynamic change-impact analysis”, Journal of Systems and Software, vol. 103, pp. 248-265, May 2015.
9. Bixin Li, Qiandong Zhang, Xiaobing Sun, Hareton Leung, “Using water wave propagation phenomenon to study software change impact analysis”, Advances in Engineering Software, vol. 58, pp. 45-53, April 2013.
10. 9.Mariem Haoues, Asma Sellami, Hanêne Ben-Abdallah, “Functional change impact analysis in use cases: An approach based on COSMIC functional size measurement”, Science of Computer Programming, vol. 135, pp. 88-104, 15 February 2017.
11. Sathyarajasekaran K, Ganesan R., “Effect of Code Generation in Change Impact Analysis using MVC Design Pattern”, Jour of Adv Research in Dynamical & Control Systems, Vol. 10, 10-Special Issue, PP 2417-2429, November 2018.
12. Chung-Yang Chen, Gen-Yih Liao, Ku-Shen Lin, “An attribute-based and object-oriented approach with system implementation for change impact analysis in variant product design”, Computer-Aided Design, vol. 62, pp. 203-217, May 2015.
13. Bixin Li, Xiaobing Sun, Jacky Keung, “FCA–CIA: An approach of using FCA to support cross-level change impact analysis for object oriented Java programs”, Information and Software Technology, vol. 55, no. 8, pp. 1437-1449, August 2013.
14. H. Cai, R. Santelices and S. Jiang, "Prioritizing Change-Impact Analysis via Semantic Program-Dependence Quantification," IEEE Transactions on Reliability, vol. 65, no. 3, pp. 1114-1132, Sept. 2016.
15. Xiaoxia Ren, O. C. Chesley and B. G. Ryder, "Identifying Failure Causes in Java Programs: An Application of Change Impact Analysis," IEEE Transactions on Software Engineering, vol. 32, no. 9, pp. 718-732, Sept. 2006.
16. P. Rovegård, L. Angelis and C. Wohlin, "An Empirical Study on Views of Importance of Change Impact Analysis Issues," IEEE Transactions on Software Engineering, vol. 34, no. 4, pp. 516-530, July-Aug. 2008.
17. Sathyarajasekaran K, Ganesan R.,“ An Approach on Software Readiness Workflow with Change Impact Analysis (SRW-CIA)”, International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-7 PP 284-288, May, 2019.
18. Sabine Teufl, Georg Hackenberg, “Efficient Impact Analysis of Changes in the Requirements of Manufacturing Automation Systems”, IFAC-PapersOnLine, vol. 48, no. 3, pp. 1482-1489, 2015.
19. Sathyarajasekaran K, Ganesan R.,“ A Novel Process using Impact Analysis over Requirement Phase with Impact Transition Model”, Test Engineering and Management, Volume-81 PP 4435-4445, November-December 2019.