Impact of Recovery Blocks in industrial practices

Research Report

Siddhartha Srinadhuni
Registration number: 9508286854
E-mail: sisr16@student.bth.se

Sairam Yepuru
Registration number: 9412175375
E-mails: saye16@student.bth.se

I. Group Members' Participation

The group members participated in idea creation and in report writing with the following amount of involvement.

Group Member	Idea Creation	Report Writing
Siddhartha Srinadhuni	50 %	50%
Sairam Yepuru	50 %	50%

***Abstract*—Software fault tolerance (SFT) is one of the prime attributes associated to the principles of dependability. Through this research report, we present the impacts of Recovery Blocks (RB) in industrial practice. A questionnaire based survey was conducted followed by Grounded Theory Analysis. The results conclude elicit enhancements to the limitations of Recovery Blocks. This document also reflects on future work that is possible in this field.**

II. Introduction

A. *Background:*

The threshold of the search for a promising research started with a possible gap that existed in the current literature. After presenting the Systematic Literature Review on Software Fault tolerance techniques, we wanted to narrow our search and study on impacts of recovery blocks in industrial practices.

The basic concepts of tolerating faults in computing have been there since a couple of decades. Initially, hardware structures were used to tolerate faults in order to make a system dependable. However, hardware structures were prone to component failures and thereby planking the dependability at high stakes [1].

Dependability has always been a key aspect in computing. A dependable system has few attributes that are to be met to present its dependability. The attributes of dependability are Availability, Reliability, Safety and Security. The same applies to software as well. A software demands to be dependable and dependable software systems are the procurement of fault avoidance and fault tolerance [2] [3].

B. *Problem:*

We, as a team, have identified the importance of Software fault tolerance and that SFT plays a key role in software engineering from our Systematic literature that we have presented. SFT techniques have consistently paved a way in designing dependable systems. The literature that we found have structured on evaluations as to which techniques in SFT are efficient. N-Version Programming and Recovery Blocks are two efficient software design methods. [4]. Literature such as [5] has shown that Recovery blocks has a wide spread scope of being implemented in industries. There is very limited study addressing Recovery Blocks in industrial practices. We have written this report to address the limitations and possible enhancements that could be a part of recovery blocks in industrial practices. Hence, this motivated us to conduct a survey based research in this topic.

C. *Objectives:*

Our research has aimed towards portraying the challenges identified for implementation of recovery blocks in industries to achieve software fault tolerance. This document also provides with the data that can be used for future reference in this particular field.

Since our aim of this research has been investigating and exploring different challenges in recovery blocks in industrial practices and suggest enhancements for the same, the scope of this research has been limited to finding the challenges, functionally. Findings have justified the limitations and suggested enhancements equally.

D. *Methods:*

Survey suits best for the kind of aim and objectives that we have considered. Apprehending knowledge by conducting a survey is the best suit for this research since our pivot is to address recovery blocks in practice. The questions have been structured in such a way that it answers the questions that are posed unambiguously and triangulates the data simultaneously. The questionnaire for the survey has been provided with a list questions, which could be answered in the space provided in the form [refer Appendix-A**]**. The collected data is then analyzed and conclusions have been inferred using grounded theory analysis.

Furthermore, we have presented our article in the following way. The section deals with the background and motivation is next. It summarizes the existing knowledge, the importance of the challenges and the problem faced by the state of art. Section IV presents the definition of research and plan. Section V talks as to how the research was carried out. Analysis and the interpretation of the outcomes is articulated in section VI. Section VII includes the results and threats to validity. Conclusions and the summaries are documented in the last section of the report.

III. Background and Motivation

The Systematic Literature Review imparted an overall outlook on different techniques of Software Fault Tolerance. Upon referring multiple journal articles and conference papers, a precise inference was drawn on impact of Recovery Blocks through industrial practices.

Dependability is closely associated to the concept of Software Faults. Martin in [2] explained that Traditional fault-tolerance techniques often deploy some kind of backwards recovery which may not be feasible in real-time systems because they may take too much time or have properties which a real-time system cannot tolerate. A single task computation will produce unacceptable results when either primary, secondary software module fail or experience transient, faults, or if the decider fails [6]. Hence, it requires adaptable techniques to maintain software that also ushers faster computations. As per former researches, improving on drawbacks in traditional techniques could outplay in performance. According to Rendell [1] the progress of Recovery Blocks can be encapsulated in separate but interacting computational processes. Recovery Blocks is a data diverse approach that uses Acceptance Testing and backward recovery method [7]. The flow and working algorithm of RB is depicted by Rendell and is illustrated in APPENDIX- B. By mitigating the possible limitations in RB, it could be implemented in industries, thus by tolerating the software faults and fulfilling the dependability aspects.

IV. Research Definition and Plan

A. Research Objective:

Our research has aimed to evaluate the challenges for implementing recovery blocks in industries to achieve software fault tolerance. Our research aim has the objectives which are as follows:

- To understand the scope of the recovery blocks technique in practicality.
- Survey targeted to practitioners so as to identify challenges in implementing recovery blocks technique from a technical perspective.
- Analysing and synthesizing of the qualitative data collected.
- Suggest possible enhancements, basing on the outcomes in the context of recovery blocks.

B. Research Questoins:

The Research questions were defined after a detailed study on the Research gap. An overview on the Systematic Literature Review and the Background study helped in formulating the Research questions.

RQ1) What is the state-of-practice of Recovery Blocks to tolerate software faults?
Motivation: Software is highly liable to frequent upgradations. Real-time implementation and technological know-how on Recovery Blocks helps in maintaining the software and assurance to Information Technology industries.

RQ2) What are the limitations of Recovery Blocks in practice?
Motivation: There exists a possibility that Recovery Blocks might not be adopted by industrial firms due to business policies, compact time constraints or lack of acquisitions. By carrying out research on this factor provides a scope to improve Recovery Blocks and simulate its usage.

RQ3) What are the possible enhancements in Recovery Blocks to improve Fault Tolerance in Software?
Motivation: A systematic study on improving Recovery Blocks forefronts the aspects like efficiency, throughput and software maintenance. The derived enhancements help in facilitating the clients and users with quality driven services.

C. Research Method:

Since, results obtained through this survey provide technician's perspective and we wanted to address implementations, limitations and enhancements of RB, it is considered better and reliable than a generic case study. Also, browsing through this survey is relatively easy, it is the best available option [8]**.** Additionally, survey suits best for the timeline we have chosen. Hence, this has been a major motivation behind choosing survey as our research methodology. This survey is addressed to practitioners as this is a technical view that we are putting forth. The methodology for our research is a practitioner's survey. The survey, in the field of recovery blocks, explores different open ended and closed ended questions whose responses are provided as a detailed description.

D. Units of Analysis:

The target respondents, for our research article, are practitioners. Also, the survey had to be answered in the given time frame. Out of probabilistic and non-probabilistic sampling, the latter serves the best for our survey [9]. The rationale behind our sampling technique is that non probabilistic sampling is less expensive in the context of time. The respondents are carefully selected individuals who are experienced in this particular domain or at the very least, have an idea about the domain. The scale of experience is also considered in our survey so as to triangulate the data that is obtained. To enhance the reliability of the survey, researchers and post-doctoral students have also been the participants of our survey.

E. Data Collection Method:

A google survey form was created in order to send out the questionnaire to the respective participants that we have shortlisted. After our SLR, we framed few survey questions that were relevant to our Research questions. We carried out a pilot for usability and inferred that it was relatively easy for answering the questions in the forms. The participants were chosen from various IT firms located in India and few post-doctoral students through LinkedIn. Email IDs have been listed out and survey forms have been sent. The survey form comprises a set of questionnaire for collecting the responses by the participants. The survey form can be seen in the Appendix-A below. The responses of this survey are further reflected on a spread sheet as it is convenient and helps storing the data with ease. Further, analyzing and accessing the responses also becomes handy when usage of spreadsheets is implemented.

F. Data Analysis Method:

The subsequent process after conducting the survey and recording the responses is the Data Analysis Method. Ground Theory Analysis was chosen to evaluate the data. The major motivation behind choosing the Grounded Theory Analysis is the number of responses for open ended questions is high and can only be interpreted by Grounded Theory Analysis. In “Principles of Survey Research Part 6: Data Analysis”, Kitchenham states that, data analysis is assumed to undertake probability sampling [10]. If we do not have a probability sample, we can calculate various statistics associated with the data that we have collected. But, we cannot estimate population statistics. As the responses recorded are homogenous on Recovery Blocks, the partitioning is carried out on the basis of demographic information. Based on these prospects, Ground Theory Analysis was preferred over other data analysis methods.

V. Research Operation

A. Research Procedure:

The research was scheduled on a time-scale of 65 days. A group of two members was formed and a primary study on the research domain was emphasized in “Software Fault Tolerance”. A Systematic Literature Review was conducted in duration of 20 days. The research questions were then defined based on the research gap observed from the SLR. In order to obtain an inference from our research questions, we decided to conduct a Survey.

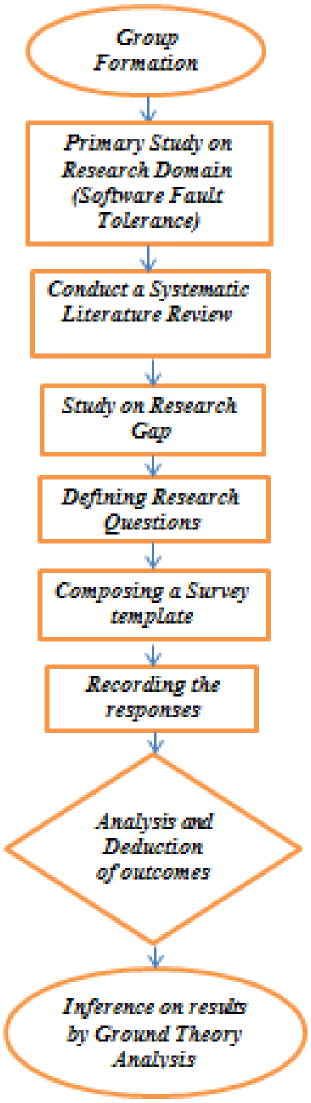

Group Formation
Primary Study on Research Domain (Software Fault Tolerance)
Conduct a Systematic Literature Review
Study on Research Gap
Defining Research Questions
Composing a Survey template
Recording the responses
Analysis and Deduction of outcomes
Inference on results by Ground Theory Analysis

FIGURE_1: Research Procedure

Upon formulating the questionnaire for the survey, it was forwarded to post-graduates, doctoral and practitioners active in the field of SFT. The responses were recoded and analyzed in a span of 10 days. A total of 16 responses were recorded after excluding 3 incomplete and 1 inappropriate responses. The final results were then depicted using Ground Theory Analysis.

B. *Quality Assurance:*

Primarily, to assure quality, anonymous filling of the questionnaire was a step. This step of making sure to not reveal the respondent's identity would lead to fair responses. Careful sampling has been done when it came to the survey. We made it sure that our sample included practitioners and post-doctoral students in the field of software maintenance and design engineering. The importance of the research that we are carrying out has been explained to practitioners who belong to different firms and organizations.

VI. Data Analysis and Interpretation

A. *Results and Analysis:*

We have listed out our criterion as the participant's experience in the given field in terms of years and job title. Given the time frame, we had to explore different domains of designations of the participants to gain reliable and qualitative data. That being said, we also listed out another criterion on know-how rating. This criterion explains us as to how familiar are the participants are with the area that we are dealing with. The respondents for this survey were 13, out of which post-doctoral students were approximately 15% and practitioners, as a whole constituted the remaining 85%. Figure 2 describes the percentages of different participants who were part of our survey. Figure 3 illustrates a bar graph of the number of responses. The detailed list of the respondents and the job titles are shown in the table below.

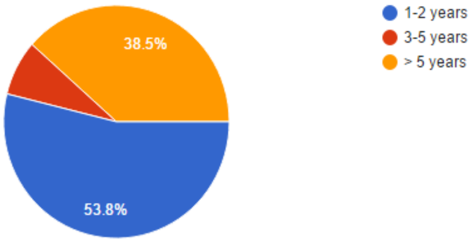


FIGURE_2: Experience of Survey Respondents

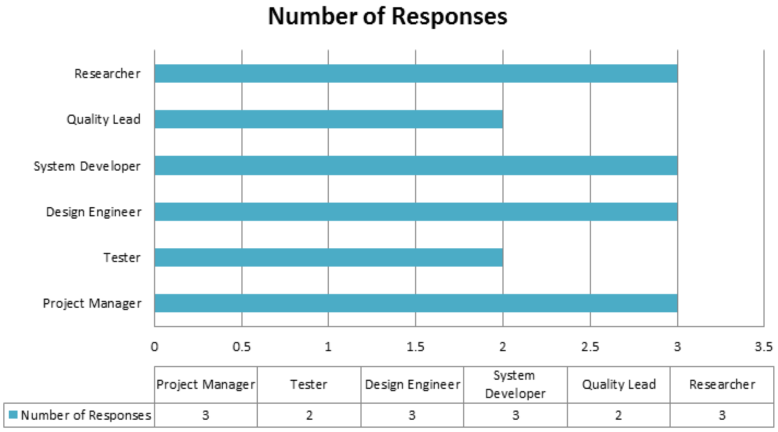


FIGURE_3: Number of Responses

S.no	Job Title	Number of Responses
1	Quality Lead	2
2	System Developer	3
3	Tester	2
4	Design Engineer	3
5	Project Manager	3
6	Researcher	3

Table_1: Job Title with Number of Responses

Bogdan in [11] says that there are factors that set qualitative research apart. Factors such as working with a lot of data, breaking it down, investigating it and provide with findings after inferring the patterns in it. The similar kind of process has been a part of our research. Integrating the results with our research questions after obtaining the data has been proven as a qualitative factor.

For analyzing the data that is provided, grounded theory analysis has to be implemented. The major motivation behind this is that we are new to this field of research and grounded theory provides the needful patterning whilst conducting a qualitative research.

Primarily, grounded theory analysis has three types of coding which are, open coding, coaxial coding and selective coding [9]**.** To understand the responses, we carried out open coding. Open coding is one of the three types of processes that exist in grounded theory. The other two being selective and axial coding. We required a process where the responses had to be understood before further analyzing them. Hence, [9] describes that open coding is required to understand and compare the responses for similarities and differences without any bias. This has been the rationale for carrying out open coding for grounded theory. By carrying out the same, we have diagnosed 10 codes. The summary of responses on implementation of recovery blocks is shown in the Table_2 and the Table_4 shows the plausible enhancements that can be a part of recovery blocks.

S. no	Summary	Number of Responses
1	Economic Feasibility	6
2	Client Specifications	4
3	Efficient Performance	3
4	Graceful Degradation	3

Table_2: The summary for RB Implementations

After narrowing the final open code sets, we have now moved to implement axial coding. Finding the connections between the obtained data set of open codes in the major goal of axial coding. 4 axial codes were derived after relating them to the open codes. Axial coding provides a void to place the open codes into axial categories. Moving towards selective coding, where we analyzed to investigate the crux of the categories. This is where recovery blocks, our field of research, is established in terms of the enhancements that could be a part of and also the limitations that are present. Consistency has been ensured with respect to the final category and thus the theory is reliable.

Therefore, on implementing analysis on the axial categories, ‘*Inefficient design of working mechanism*’ was concluded as the core that all our data circumscribe. Thus we have answered the first research question, limitations of recovery blocks. Therefore, the challenge for the recovery blocks in implementing on an industrial scale lies in ‘*Inefficient design of working mechanism’*. Therefore, our first question has been answered based on the survey based date. Further, enhancements were also a part of our research and it defined our second research question.

S.no	Summary	Number of Responses
1	Domino Effect	3
2	System Overhead	2
3	Mechanism for the recovery cache	3
4	Designing the acceptance test	3
5	Design of the primary and alternate modules	5

Table_3: The summary of Limitations in RB

After presenting the answer to the first research question, we applied a similar kind of approach for answering the second question as well. Identification of 13 codes has been identified as a part of open coding and final selective coding dawned down to one central core. We analyzed this and inferred that enhancement in ‘*’check pointing and transactional nature*” would define the enhancements that we were looking for thereby answering the question of possible enhancements in recovery blocks to achieve software fault tolerance.

S. no.	Summary	Number of Responses
1	Support acceptance test at user’s end	3
2	Metrics associated with the flow control of the code	4
3	Parallel degradation of recovery blocks	5
4	Byzantine faults	2
5	Manual Intervention	2

Table_4: The summary for Enhancements in RB

For Grounded Theory Analysis, in the context of research we have done, should list out limitations for the existing implementations and enhance for the current limitations. After analyzing, we listed out the implementations, its limitations and enhancements that should be a part of Recovery Blocks in state of practice.

S.NO	Implementations	Limitations	Elicited Enhancements
1	Economic Feasibility	System Overhead	Parallel degradation of recovery blocks
2	Client Specifications	Domino Effect, System Overhead	Manual Intervention, Byzantine faults
3	Efficient Performance	Domino Effect, Designing the acceptance test	Support acceptance test at user's end
4	Graceful Degradation	Design of the primary and alternate modules	Metrics associated with the flow control of the code

Table_5: Summary of Grounded Theory Analysis

The Table_5 presents the possible elicited enhancements to the limitations that have been derived from the survey. For the implementations of Recovery Blocks the possible limitations have been listed out.

B. Threats to Validity:

From the results in Figure_2 and Table_1**,** approximately 39% of the participants have more than 5 years of experience in our research field. Furthermore, the job designations such as quality lead, researcher, tester, design engineer, project manager ensure the quality of the responses thereby providing high reliability and validity quotient.

Yet another potential threat is the implementation of the grounded theory analysis that we have performed due to its liberty to perform the coding.

VII. Discussion

A. Contribution:

The results were presented after a detailed scrutiny of the recorded responses. A contrast was observed with the perspectives of researchers next to practitioners. The sequence of similarities and differences were marked through their given inputs and contributions.

Similarities:

- A common inference was made that Recovery Blocks was implemented due to its economic feasibility.
- Both researchers and practitioners agreed that Recovery Blocks degrade parallel to the performance of the software. Thereby, this aspect primarily requires troubleshooting.
- The attribute of limitations, 'Inefficient design of working mechanism' and possible enhancement from software degradation were commonly answered by both the set of respondents.

Differences:

- The practitioners preferred Recovery Blocks due to client specifications, while researchers supported it because of its ability to induce graceful degradation.
- Some researchers mentioned the drawbacks for Recovery Blocks like Domino-effect, system-overhead etc., whereas the practitioners mentioned few alternative techniques.
- The responses of the researches and practitioners can be clearly differentiated on a technical and business perspective.

B. Threats to Validity:

From the literature [12] referred, the possible threats to validity were identified and noted to our research.

i. **Researcher bias/Respondent bias:** The approach to our research domain was systematically surmised by narrowing our research gap. The results were obtained and the conclusions were claimed, after thoroughly analysing the survey and reducing the results by Ground Theory. The personal credentials of the respondents were anonymized and responses obtained were open-ended perspectives.

ii. **External Validity Threats:** This was one of the prime factors affecting our research. In order to triangulate the data, we have forwarded our survey to a limited population, i.e., two countries and few companies. This resulted in a restricted outreach in terms of population sampling.

iii. **Construct Validity Threats:** After duly analysing and verifying the results few irrelevant, incomplete and inappropriate responses were eliminated. This mitigated the potential for Construct Validity Threats.

iv. **Internal Validity Threats:** There was no scope for any external party to view or modify the results. There forth, the scope for Internal Validity Threat was completely nullified.

v. **Conclusion Validity threats:** The conclusions and future work were derived with respect to the given guidelines, mitigating the possibility for this validity threat.

VIII. Summary and Conclusions

A. *Summary and Conclusions:*

Recovery blocks has been a major mechanism to tolerate faults in a software system. When efforts have put into this field of research, findings have proven that this mechanism is not being implemented widely. This spurred us to investigate the reasons behind not using it extensively. Our realm of research has been focused to challenges and thereby suggesting few possible changes in the functionality of the mechanism through a technical perspective. Therefore the purpose of this paper is to address the challenges that recovery blocks are facing in practice.

Through the survey that we have conducted, it was diagnosed that design has been inefficient with respect to the working mechanism of recovery blocks. Additionally, flaws in the acceptance test, system overhead, flaws in primary and alternative modules have also been critical and comprised as highly stated challenges in our results. We have also investigated to provide few suggestions for possible enhancements as a part of this research article. Upgradation of checkpointing and transactional nature of recovery blocks proved to be enhancements that would optimize the performance of RB. Manual Intervention and Parallel degradation have also been a part of the enhancements through our findings.

B. *Future Work:*

By addressing the challenges, this particular mechanism could be improved in terms of practical usage. Now that, constrains such as scale and scope were in our way, future research could outreach to a larger number of target participants. This can improve in finding more challenges that recovery blocks are facing and accommodate further improvements. This can also include experimentation as to verify if the enhancements can suffice the limitations in the context of Recovery Blocks.

References

[1] B. Randell, “System structure for software fault tolerance,” *IEEE Trans. Softw. Eng.*, vol. SE-1, no. 2, pp. 220–232, Jun. 1975.

[2] M. Hiller, “Software fault-tolerance techniques from a real-time systems point of view,” *Dep. Comput. Eng. Chalmers Univ. Technol. SE-412*, vol. 96, pp. 98–16, 1998.

[3] J. M. Smith, “A survey of software fault tolerance techniques,” 1988.

[4] G. Levitin, “Optimal structure of fault-tolerant software systems,” *Reliab. Eng. Syst. Saf.*, vol. 89, no. 3, pp. 286–295, 2005.

[5] Z. Xie, H. Sun, and K. Saluja, “A survey of software fault tolerance techniques,” *Univ. Wis.-MadisonDepartment Electr. Comput. Eng.*, vol. 1415, 2006.

[6] J. B. Dugan, S. A. Doyle, and F. Patterson-Hine, “Simple models of hardware and software fault tolerance,” in *Reliability and Maintainability Symposium, 1994. Proceedings., Annual*, 1994, pp. 124–129.

[7] Z. Xie, H. Sun, and K. Saluja, “A survey of software fault tolerance techniques,” *Univ. Wis.-MadisonDepartment Electr. Comput. Eng.*, vol. 1415, 2006.

[8] T. Punter, M. Ciolkowski, B. Freimut, and I. John, “Conducting on-line surveys in software engineering,” in *2003 International Symposium on Empirical Software Engineering, 2003. ISESE 2003. Proceedings*, 2003, pp. 80–88.

[9] Juliet Corbin, Anselm Strauss, “Grounded Theory Research: Procedures, Canons and Evaluative Criteria.” .

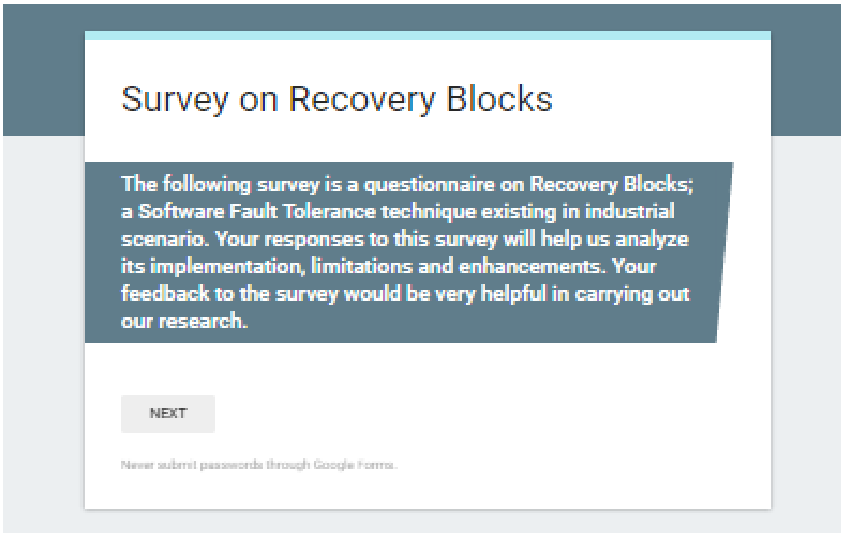
[10] B. Kitchenham and S. L. Pfleeger, “Principles of survey research: part 5: populations and samples,” *ACM SIGSOFT Softw. Eng. Notes*, vol. 27, no. 5, pp. 17–20, 2002.

[11] Robert C. Bogdan, “Qualitative Research for enducation: An Introduction to theory and methods.” 2003.

[12] Robert Feldt, Ana Magazinius, “Validity Threats in Empirical Software Engineering Research - An Initial Survey.” [Online]. Available: https://www.researchgate.net/profile/Ana_Magazinius/publication/221390199_Validity_Threats_in_Empirical_Software_Engineering_Research_-_An_Initial_Survey/links/53ecc7f80cf2981ada10e3d6.pdf. [Accessed: 04-May-2016].

APPENDIX

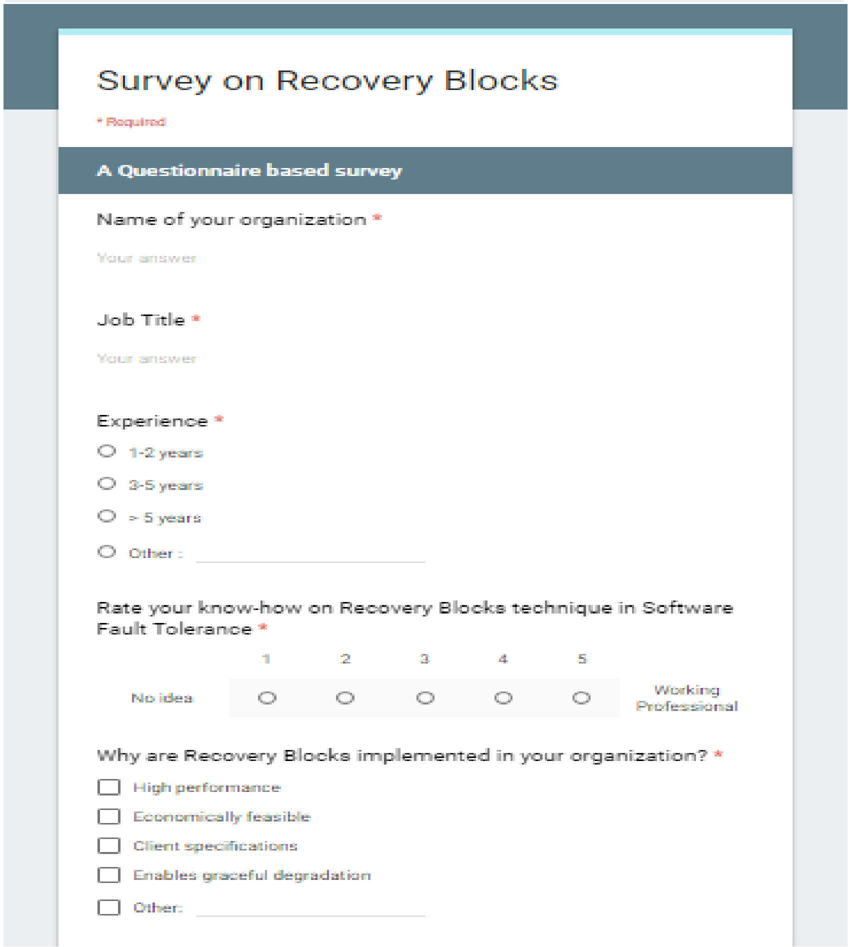
APPENDIX- A: Questionnaire survey form



Survey on Recovery Blocks

The following survey is a questionnaire on Recovery Blocks; a Software Fault Tolerance technique existing in industrial scenario. Your responses to this survey will help us analyze its implementation, limitations and enhancements. Your feedback to the survey would be very helpful in carrying out our research.

Never submit passwords through Google Forms.



Survey on Recovery Blocks

* Required

A Questionnaire based survey

Name of your organization *

Your answer

Job Title *

Your answer

Experience *

○ 1-2 years

○ 3-5 years

○ > 5 years

○ Other :

Rate your know-how on Recovery Blocks technique in Software Fault Tolerance *

	1	2	3	4	5	
No idea	○	○	○	○	○	Working Professional

Why are Recovery Blocks implemented in your organization? *

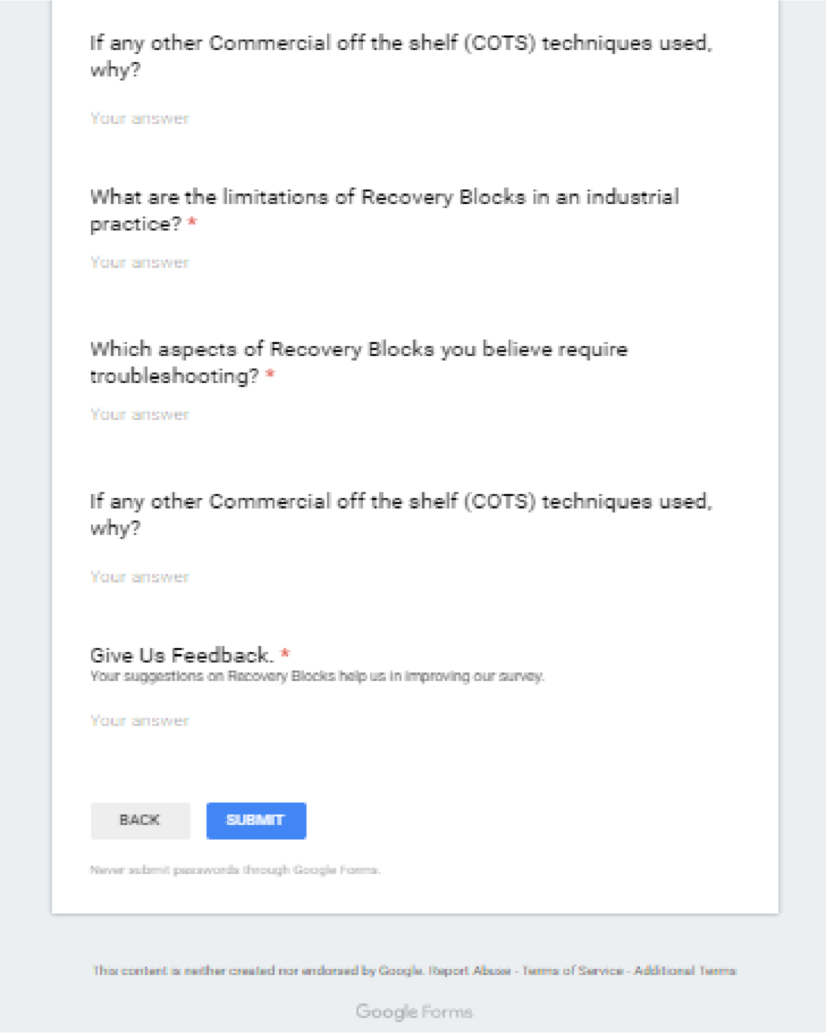
□ High performance

□ Economically feasible

□ Client specifications

□ Enables graceful degradation

□ Other:



If any other Commercial off the shelf (COTS) techniques used, why?

Your answer

What are the limitations of Recovery Blocks in an industrial practice? *

Your answer

Which aspects of Recovery Blocks you believe require troubleshooting? *

Your answer

If any other Commercial off the shelf (COTS) techniques used, why?

Your answer

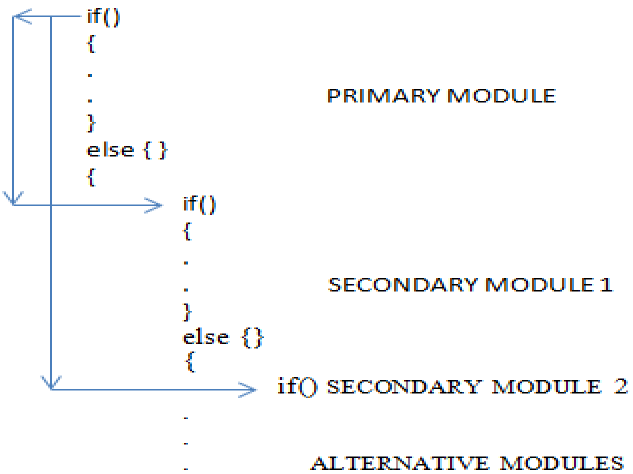
Give Us Feedback. *
Your suggestions on Recovery Blocks help us in improving our survey.

Your answer

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. Report Abuse - Terms of Service - Additional Terms

APPENDIX- B: Basic algorithm for Recovery Blocks by Rendell


if()
{
.
.
}
else { }
{
PRIMARY MODULE
if()
{
.
.
}
else {}
{
SECONDARY MODULE 1
if() SECONDARY MODULE 2
.
.
.
ALTERNATIVE MODULES