Situational Modern Code Review Framework to Support Individual Sustainability of Software Engineers

Sumaira Nazir[1], Nargis Fatima[2], Suriayati Chuprat[3]
Razak Faculty of Technology and Informatics[1, 2, 3]
University Technology Malaysia (UTM)
Kuala Lumpur, Malaysia
Faculty of Engineering and Computer Science[1, 2]
National University of Modern Languages (NUML)
Islamabad, Pakistan

***Abstract*—Modern Code Review (MCR) is a socio-technical practice to improve source code quality and ensure successful software development. It involves the interaction of software engineers from different cultures and backgrounds. As a result, a variety of unknown situational factors arise that impact the individual sustainability of MCR team members and affect their productivity by causing mental distress, fear of unknown and varying situations. Therefore, the MCR team needs to be aware of the accurate situational factors, however, they are confronted with the issue of lack of competency in the identification of situational factors. This study aims to conduct the Delphi survey to investigate the optimal and well-balanced MCR-related situational factors. The conducted survey also aimed to recognize and prioritize the most influencing situational factors for MCR activities. The study findings reported 21 situational factors, 147 sub-factors, and 5 Categories. Based on the results of the Delphi survey the identified situational factors are transformed into the situational MCR framework. This study might be helpful to support the individual sustainability of the MCR team by making them aware of the situations that can occur and vary during the execution of the MCR process. This research might also help the MCR team to improve their productivity and sustain in the industry for longer. It can also support software researchers who want to contribute to situational software engineering from varying software engineering contexts.**

***Keywords*—Situations; situational factors; Modern Code Review (MCR); sustainable software engineer; situational software engineering**

I. Introduction

Software development paradigm shift from conventional to more, complex, and sustainable software [1] demands better software engineering techniques and practices. Software methods that can effectively manage software engineering activities, operations, and maintenance are needed [2]. In the modern age of software development situation-aware software development is highly desirable. Situation-aware software engineering also known as situational software engineering [3] allows software engineers to deal with known situations instead of being unproductive with unknown situations. These situations have a strong influence on the individual sustainability of software engineers as well as on successful software development. In the literature, these situations are termed as *"situational factors"*.

Individual sustainability is one of the core dimensions of sustainable software engineering [4] that refers to psychological well-being, self-respect, and the sustainability of software engineers [5], [6] in the industry. It is argued that individual sustainability of software engineers can be ensured by situational software engineering [5] that can help the software engineers to deal with varying and unknown situations [7]. Situational software engineering by identifying the situational factors [1] boosts the confidence, competence, and productivity of the software engineers.

It is contended that effort has been contributed regarding situational factor identification for software development generally [8] and specifically software requirement engineering [9], [10], [11], however, other software engineering activities need further insight. Presently, the software engineers engaged in Modern Code Review (MCR) are facing the issue of lack of competency in the identification of situational factors [12], [13], [14], [15].

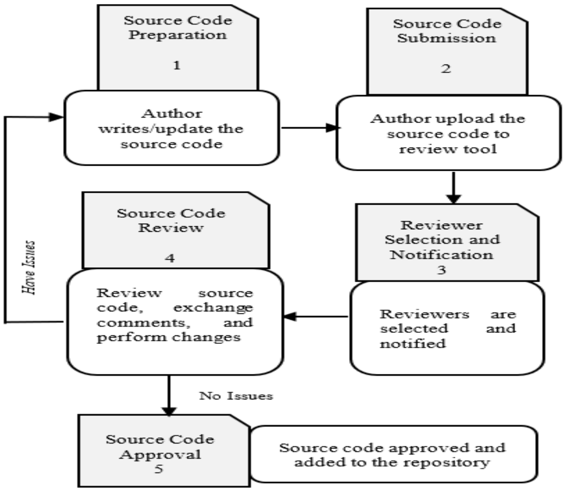

Source Code Preparation
1
Author writes/update the source code
Source Code Submission
2
Author upload the source code to review tool
Reviewer Selection and Notification
3
Reviewers are selected and notified
Source Code Review
4
Review source code, exchange comments, and perform changes
Have Issues
No Issues
Source Code Approval
5
Source code approved and added to the repository

Fig 1. MCR Overview [19].

MCR is a significant light weight [16], [17] software engineering activity. In MCR [18], [19] the reviewer reviews the source code submitted by the author, identifies the issues and provide feedback to the author for source code improvement. The author then performs changes according to the feedback of the reviewer and again submit the source code for review. The reviewed cycle continues till the source code is approved by the reviewer. After the approval, the source code is added to the repository. MCR process is executed with the support of review tools [16], [18], [20] such as Code Flow and Gerrit. The MCR process overview is given in Fig. 1.

Though the existing research [8], [9], [10], [11], [21] has provided attention to situational software engineering by identifying situational factors, however, the existing research has discussed the situational factors' identification for software development generally and specifically for requirement engineering. There are fewer considerations regarding the exploration of situational context for MCR activities to help software engineers' engaged in MCR activities to sustain in the industry for longer [12], [13], [22] lack of such research can cause mental distress of software engineers and ultimately failure of software [6], [12].

No, systematized investigations are available that comprehensively report the situational factors in the context of MCR. Hence, there is a need to recognize situational factors [14], [22] and to develop a comprehensive situational MCR framework to address the issue of lack of competency in the identification of situational factors, to support the individual sustainability of software engineers, and for successful software development.

Therefore, this study aims to conduct the Delphi survey to finalize the list of situational MCR factors, sub-factors, and categories for their practicality, naming, grouping, and sub-grouping. Delphi survey also aimed to identify and prioritize the most influential situational factors for MCR activities. The results obtained after the Delphi survey are used to develop the situational MCR framework to support the individual sustainability of software engineers engaged in MCR activities. This study is an extension of our previous research that is Systematic Literature Review (SLR), to identify the situational factors from literature and expert review to evaluate by the experts the list of situational factors obtained as a result of SLR. The results of SLR and expert review are presented in [13], [23], [24].

The paper is organized as Section II delivers the literature review. Section III highlights the details regarding the methodology employed to conduct the research. In Section IV Delphi survey results are presented. The research discussion is provided in Section V. Sections VI and VII provides the conclusion and work opportunities in the future. The research contributions are highlighted in Section VIII.

II. Literature Review

Over the past 10 years, the trend of situational software engineering has been observed to successfully develop software projects. The situational context in software engineering was explored by numerous researchers over the last decade. Researchers have highlighted the importance of situational factor identification in the context of requirement management. It is conveyed that situational factors such as varying business needs, communication channels, project stakeholders, and technologies influence the requirement management [25].

Moreover, a comprehensive situational framework for the software development process has been introduced by [8]. They followed SLR and data coding procedures of grounded theory to identify the situational factors. According to their investigations team size, culture, task complexity, organization structure, and customer satisfaction are the situational factors that affect the software development and sustainability of engineers. Likewise, situational factors for the requirement engineering process have been explored by [21]. They followed SLR, data coding based on grounded theory to identify situational factors. They argued that management awareness, project characteristics, requirement estimation, and requirement understanding are the situational factors that create an impediment for the requirement engineer to perform requirement engineering activities. It is also argued by [26] that situational factors such as organization size, management, and people impact project success.

TABLE I. Comparison of State-of-the-Art of Situational Software Engineering Research

Reference	Domain of Contribution	Contribution regarding situational Awareness
Ghosh et al., (2011)	Requirement management	Reported situational factors such as varying business needs, communication channels, project stakeholders, and technologies
Clarke and Connor., (2012)	Software development process	Introduced situational framework.
Huma et al., (2014)	Requirement engineering process	Introduced situational framework.
Bakht et al., (2015)	Requirement Engineering	Reported situational factors organization size, management, and people.
Clarke et al., (2016)	Software Development Process	Assessed the impact of the Situational factors such as task, complexity, management commitment, customer satisfaction, and organization structure.
Mark et al., (2017)	Software Development Process	Analyzed the impact of situational factors such as team size, culture, productivity, commitment, skills, turnover, experience, and developer profile
Gulzar et al., (2018)	Requirement Engineering	Reported situational factors such as organization, distance, knowledge management, trust, stakeholder, project, tools and technology, national culture, and physical factors.
Knononenko et al., (2018)	Modern code Review	Reported 'merge type' situational factor
Sadowski et al., (2018)	Modern code Review	Reported situation factors such as source code change size, number of the reviewer, number of comments on the change, distance, social interactions, and review subject
Ebert et al., (2019)	Modern code Review	Reported 'Confusion' situational factor

Additionally, it is reported that situational factors make it difficult to harmonize software development process activities, therefore the researchers should explore the situational factors for software engineering activities to avoid software failures [3]. Similarly, the impact of situational factors such as team size, culture, productivity, commitment, skills, turnover, experience, developer profile on the software development process has been analyzed by [1]. It is contended that software development is highly dependent on situational factors. Likewise, situational context concerning requirement engineering has been explored by [11]. They have identified situational factors such as organization, distance, knowledge management, trust, stakeholder, project, tools and technology, national culture, and physical factors.

Little effort has also been provided concerning situational factors identification in MCR. The recent effort provided by [27], they highlighted the situational factor 'Confusion' in code review. They argued that confusion is an integral part of human problem-solving, which normally arises from the information. They stated that this situational factor causes a decrease in confidence, abandonment of the project, and negative emotions. Likewise, [28] explored situational factors such as source code change size, number of the reviewer, number of comments on the change, distance, social interactions, and review subject can impact the review process outcome. Similarly, [29] argued that the merge type situation has a noteworthy effect on peer review merge time. Table I shows the comparison of the state-of-the-art of situational software engineering research.

From the literature, it is observed that effort has been provided concerning the situational context in software development. It is also analyzed from the literature that three frameworks on situational factors are presented by researchers. One of them focused on the software development process [8], though this framework is the foundation for the research concerning situational factors identification, however it discussed the situational factors' identification for software development generally. Two of them focused on requirement engineering [11], [21].

There is a gap in the literature regarding situational factors' identification for MCR. As a result, there is an unavailability of situational guidelines and frameworks for MCR to aid software engineers to enhance their competency in the recognition of situational factors for MCR activities. To fill this gap this study focused on situational factors' identification in the context of MCR as the MCR team members are facing the individual sustainability challenge of situational factors identification [12], [13], [14], [15]. Besides this, the research aimed to prioritize the most influencing situational factors for MCR activities. The study also aimed to provide a comprehensive Situational MCR framework to support the sustainability of software engineers engaged in MCR activities.

III. Research Methodology

This study conducted a Delphi survey to come up with a unique and validated list of situational factors for MCR, as well as the prioritized list of most influencing situational factors for MCR activities. This study is the extension of our previous research work that involves SLR, [23], [24] and expert review [13]. The research methodologies are discussed in the following sub-sections. The research procedure is summarized in Fig. 2.

A. Systematic Litearture Review (SLR)

SLR has been performed to identify the initial list of situational factors from the literature. The guideline given by [30] has been followed to conduct the SLR. The conducted SLR involved 158 studies. As a result of SLR 23 situational factors, 167 sub-factors and 5 categories have been recognized. The primary results of SLR are presented in [23], [24].

B. Expert Review

After conducting the SLR, the obtained list of situational factors was evaluated through expert review. The guideline given by [31] were followed to conduct the expert review. As a result of expert review, 23 situational factors, and 5 categories were obtained, however, the list of sub-factors was reduced to 152 according to the recommendations of the experts. The detailed results of the expert review are presented in [13].

C. Delphi Survey

To evaluate the list of situational factors from industry experts, the Delphi survey was conducted. The Delphi technique is a reliable research method with the potential for use in problem-solving, decision making, and group consensus in a wide variety of areas [32]. In the Delphi method the questionnaires are provided to the experts in such a way that the anonymity of their responses is preserved. Feedback against questionnaires continues till convergence of expert's opinion, or consensus is attained. The work product of the Delphi method is consensus among the experts and their comments, on questionnaire items.

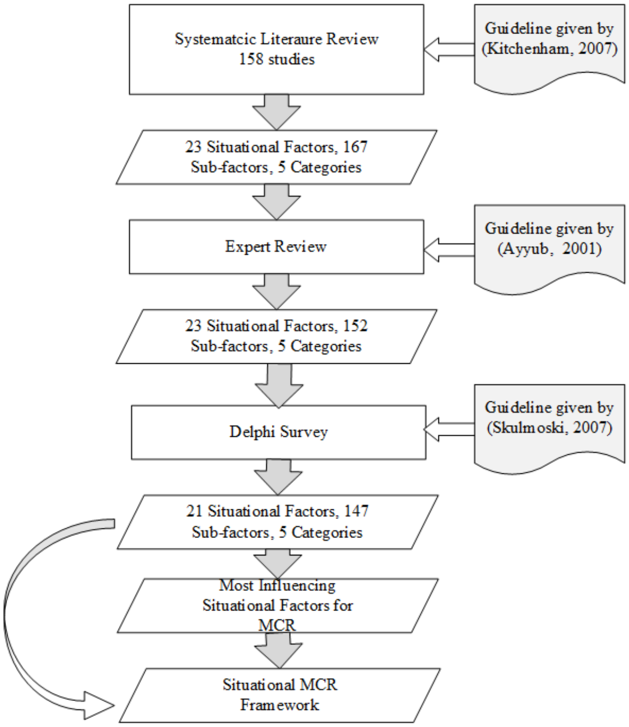


Fig 2. Research Procedure.

D. Delphi Survey Objectives

The Delphi survey has been performed 1) to assess the practicality of the identified situational factors, sub-factors, and categories regarding MCR concerning industry 2) to recognize and prioritized the most influential situational MCR factors for MCR activities 3) to get the recommendation about naming conventions, grouping, and sub-grouping of provided situational MCR factors, sub-factors, and categories 4) to distinguish new industry-based situational MCR factors, with their connected sub-factors, and categories for MCR. The guidelines given by [33] were used to conduct the Delphi Survey.

E. Delphi Experts' Selection and Panel Size

The selection of the experts for the Delphi study is a very important aspect as the Delphi results are intensely based on the views of selected experts [34]. It is claimed that there are four key criteria that must be met while experts' selection: (1) knowledge and experience with the problems under investigation; (2) willingness to participate (3) satisfactory time to give in survey (4) good communication skills [33]. Based on these criteria for this study the experts are selected having experience of 8 or more than 8 years in the industry, must have knowledge of MCR, situational software engineering, and sustainable software engineering particularly individual sustainability. Besides this their capacity and willingness to participate in the study as well as time commitment were also considered for their selection.

Expert panel size deals with the number of experts to participate in the study. It is conveyed that for Delphi study the expert panel size is flexible [33] and a similar group of people, ten to fifteen experts might be enough" [33]. Therefore, we contacted fifteen experts, however, ten experts agreed to participate.

F. No. Rounds in Delphi Study

The performed Delphi survey involved two rounds. The Delphi experts' input was collected through questionnaires. It is claimed that in Delphi study most convergence of panel responses occurs between one to two rounds [35].

G. Delphi Questionnaire Design

The questionnaire is the core aspect of the Delphi study. The Round 1 questionnaire was distributed into four segments. Segment I intended to gather the background of the experts. Segment II was composed of a list of identified situational factors, connected sub-factors, and categories produced as a result of our earlier study grounded on SLR [13], [23], [24] and expert review [13] and expert review. In Segment II the experts were requested to grade the situational factors for their practicality and their level of influence for each MCR activity. Segment III intended to collect new industry based situational factors, connected sub-factors, and categories that must be included in the list. In the same segment the experts were also asked to indicate any recommendation regarding naming conventions, grouping, and sub-grouping of the provided list of factors, sub-factors, and categories from the industry perspective. In segment IV, the experts were requested to provide current real project examples for which they performed MCR activities with the set of situational factors that they experienced. This section was particularly planned for creating the scenario for experimental evaluation of developed situational MCR framework. The Round 2 questionnaire design was similar to Round 1 excluding segment I that aimed to collect the experts' demographic information.

H. Pilot Study

Prior to giving the questionnaire to the experts, it was evaluated in a pilot study by five software engineering researchers for clarity and understanding. As it is conveyed that though a pilot study is an optional aspect, it helps in the identification of obscurities in the questionnaires that might affect the outcome [36]. The feedback received was encouraging. No changes were suggested in the pilot study.

I. Procedure for Data Analysis

This study involved descriptive statistics as it is a basic analytical approach. It gives a basic quantitative approach for examination and produces an overall view of the results [37]. To grade the practicality of situational factors a five-point Likert scale i.e. from 1 to 5 (Very High- 5, High - 4, Moderate - 3, Low- 2, Very Low – 1) was utilized. Likewise, to grade the level of influence of situational MCR factors for each MCR activity, a five-point Likert scale i.e. 1 to 5 (5-Most Influential, 4-Influential, 3-Moderate, 2-Weakly Influential, 1-Not Influential) was utilized. For calculating the practicality of situational MCR factors and to recognize and prioritize the most influential situational MCR factors, the mean values were gathered into the discrete categories as shown in Table II. Table II also shows the grouping of the mean values to measure the practicality of MCR factors and the level of influence of situational MCR factors.

The mean practicality values along with mean influential values of sub-factors were computed primarily and were further converted into a single composite mean value showing composite mean practicality and composite mean influence value for the connected situational MCR factors. To obtain the consensus on the situational MCR factors' practicality and their influence level the standard deviation has been utilized as presented in Table II. Primarily the standard deviation of the sub-factors was computed than was further converted into a single composite standard deviation for the related situational MCR factor. Grounded on the attained composite standard deviation of situational MCR factors the consensus level among the experts was obtained. Equation (1) has been formulated based on the guidelines given by [38] to compute the composite standard deviation of situational factors.

$$SD(SitF)=\sqrt{\left((SD(SitSbF_1)) +\ldots+(SD(SitSbF_k))\right)/k} \quad (1)$$

TABLE II. GROUPING OF MEAN VALUES TO MEASURE PRACTICALITY AND INFLUENCE LEVEL OF SITUATIONAL MCR FACTORS

Mean Score =X	**Level of Practicality**	**Level of Influence**
4.0≤X≤ 5.0	Very High	Most Influential
3.0≤X< 4.0	High	Influential
2.0≤X< 3.0	Moderate	Moderate
1.0≤X< 2.0	Low	Weakly Influential
0≤X< 1.0	Very Low	Not Influential

TABLE III. DECISION PRINCIPLES FOR THE LEVEL OF CONSENSUS

Standard Deviation (SD=X)	Level of Consensus
0 ≤ X <1	High
1 ≤ X <1.5	Fair Level
1.5 ≤ X <2	Low Level
2 < X	No Consensus

Where ‘*SD*’ refers to the standard deviation, ‘*SitF*’ denotes to situational factor. ‘*SitSbF*’ denotes the sub-factor of the associated situational factor and its value ranges from 1 to *k*, and ‘*k*’ denotes to the total number of sub-factors for associated situational factors.

To achieve the consensus among the experts the standard deviation was measured based on [39]. Table III denotes the consensus levels used in this study. A standard deviation near to ‘0’ indicated that the experts’ gradings tended to be very close to each other, standard deviation far from ‘0’ indicated that the gradings were spread out over a large range.

J. Data Collection and Analysis

This section detailed the data collected from the Delphi experts along with the analysis of collected data relying on the procedure of data analysis mentioned in sub-section ‘I’. The conducted study involved two rounds. The details about data collection in Delphi rounds are discussed in sub-sections.

K. Delphi Survey Round 1

In the Delphi Round 1 the questionnaire was provided to the experts to collect their input. They were given one week to provide the feedback. The experts were contacted on phone calls to make sure about their mindfulness concerning the feedback submission date for Round 1. It takes two weeks to complete the Delphi survey Round 1. The Round I intended to collect the background information from the experts. It also intended to assess the list of situational MCR factors, connected sub-factors, and categories for their naming, grouping, and sub-grouping, which was produced as a result of our previous study based on SLR [23], [24] and expert review [2]. Round 1 includes the assessment of the situational MCR factors for their practicality and their level of influence for each MCR activity. In Round 1 the experts were also asked to state new industry-oriented situational MCR factors, connected sub-factors, and categories that need to be present in the provided list. The scale used to grade the practicality and level of influence is presented in sub-section ‘I’. The particulars regarding the Round 1 questionnaire are given in sub-section ‘G’. In Delphi Round 1 the expert provided some recommendations, therefore Delphi Round 2 has been conducted to get the consensus on the recommended changes among the experts.

L. Delphi Round 2

In Round 2, the experts were provided with the summary of Round 1 results along with the updated list of situational MCR factors, sub-factors, and categories based on the suggestions of the experts given in Round 1. The details about the Round 2 questionnaire are given in sub-section ‘G’. Round 2 was completed in 2 weeks. In Round 2 the consensus among the experts on the provided list of all situational MCR factors was achieved, and no changes were suggested therefore we decided to stop at Delphi Round 2.

IV. RESULTS

This section highlights the Delphi survey results. The practicality level of situational MCR factors with the standard deviation for Delphi Round 1 and 2 are presented in Fig. 3 to Fig. 6. Fig. 3 and Fig. 4 shows the comparison of the Delphi survey results of mean perceived practicality values of the situational MCR factors. It also shows the changes performed based on the suggestions of experts. For instance, the factors name ‘Team’ was changed as ‘Team Dynamics’. This comparison also indicated that the mean perceived practicality value for the situational factors have been increased in Delphi Round 2. The comparison of Fig 5 and Fig. 6 shows that the consensus level for most of the situational factors was increased in Round 2 for their practicality among the Delphi experts as standard deviation moves near to ‘0’ in Round 2. For instance, for the factor ‘Tool,’ the standard deviation was ‘0.3855011’ in Round 1 whereas in Round 2 the standard deviation was ‘0.3312434’.

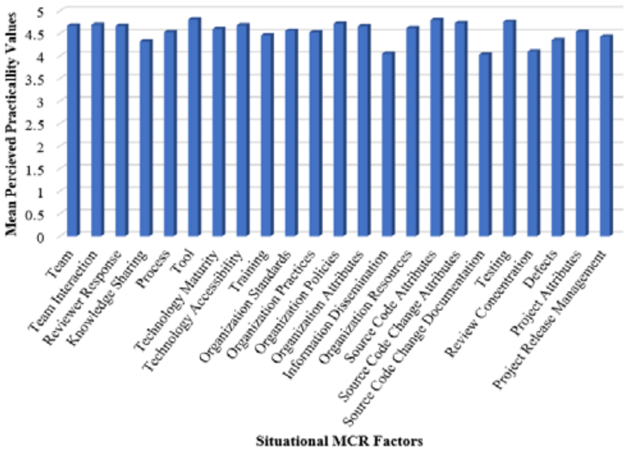


Fig 3. Composite Mean Perceived Practicality Value of Situational MCR Factors -Round 1.

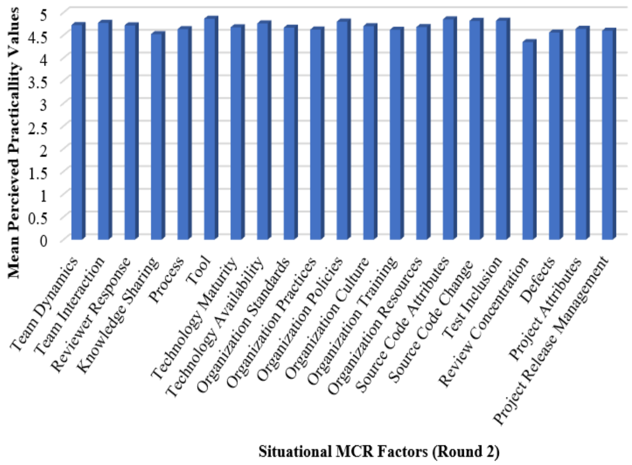


Fig 4. Composite Mean Perceived Practicality Value of Situational MCR Factors -Round 2.

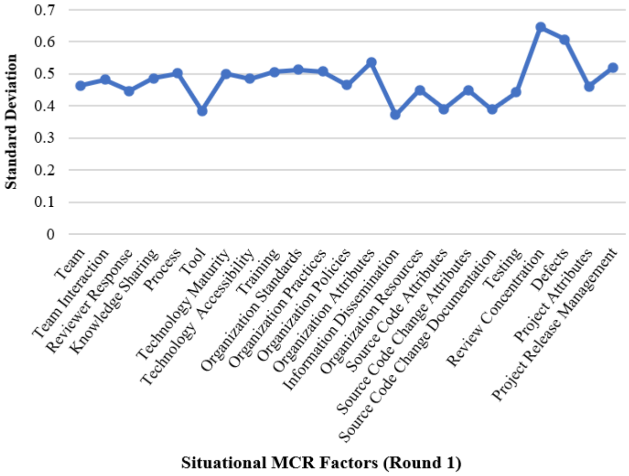


Fig 5. Consensus Level among the Delphi experts for Mean Perceived Practicality of situational MCR factors - Round 1.

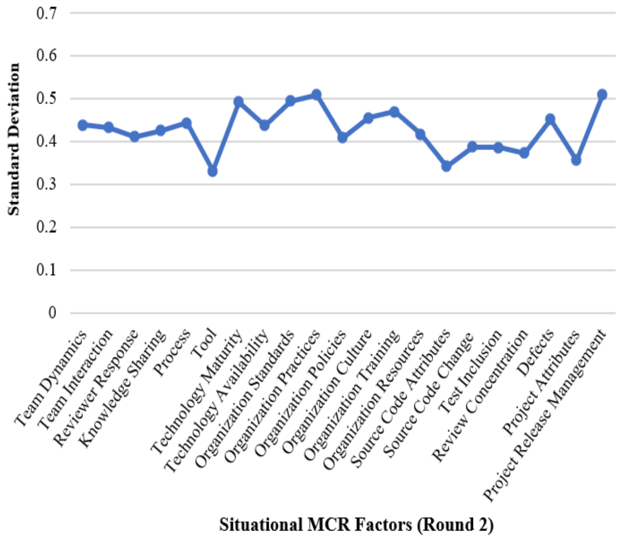


Fig 6. Consensus Level among the Delphi experts for Mean Perceived Practicality of situational MCR factors - Round 2.

Table IV displays the ranking of situational MCR factors for their level of practicality.

Concerning the most influential situational MCR factors, the mean influential values of sub-factors of each situational MCR factor in final Round for; Source Code Preparation ranges from 2.5 to 5.0, Source Code Submission ranged from 1.2 to 5.0, Reviewer Selection and Notification ranges from 1.8 to 5.0, Source Code Review ranges from 3.0 to 5.0, Source Code Approval ranges from 2.0 to 5.0. To find the most influential factors, the composite mean influential values of their related sub-factors were computed. The factors having composite mean values equal to or above 4.00 were considered as the most influential factors for that specific MCR activity. For each MCR activity, the most influential factors were identified based on their composite mean values after the final Delphi Round and are presented in Tables V to IX with the standard deviation.

TABLE IV. RANKING OF SITUATIONAL FACTORS FOR PERCEIVED LEVEL OF PRACTICALITY

Situational MCR Factors	Composite Mean Practicality Values	Standard Deviation	Rank
Tool	4.8625	0.331243449	1
Source Code Attributes	4.85	0.341565026	2
Test Inclusion	4.82	0.38586123	3
Source Code Change	4.816	0.387298335	4
Organization Policies	4.8	0.40824829	5
Team Interaction	4.772	0.43228311	6
Technology Availability	4.76	0.437162568	7
Team Dynamics	4.724	0.438684903	8
Reviewer Response	4.71667	0.410510071	9
Organization Culture	4.7	0.455420034	10
Organization Resources	4.68	0.4163332	11
Technology Maturity	4.675	0.49159604	12
Organization Standards	4.667	0.494413232	13
Project Attributes	4.641	0.356162676	14
Process	4.633	0.443053379	15
Organization Practices	4.625	0.508265023	16
Organization Training	4.62	0.469041576	17
Project Release Management	4.6	0.509175077	18
Defects	4.56	0.451335467	19
Knowledge Sharing	4.525	0.424918293	20
Review Concentration	4.35	0.372677996	21

TABLE V. RANKING AND INFLUENTIAL LEVEL OF SITUATIONAL MCR FACTORS FOR SOURCE CODE PREPARATION

Most influential Situational Factors	Composite Mean Influential Value	Standard Deviation	Rank
Source Code Attributes	4.8625	0.343592135	1
Source Code Change	4.725	0.361324723	2
Tool	4.7125	0.445658065	3
Team Dynamics	4.704	0.436144726	4
Test Inclusion	4.6	0.503322296	5
Organization Culture	4.45	0.45338235	6
Organization Policies	4.43333	0.389681731	7
Project Attributes	4.4166	0.449279258	8
Process	4.11666	0.307318149	9
Technology Availability	4.08	0.53748385	10
Project Release Management	4.066666667	0.384900179	11
Defects	4.0333	0.36004115	12
Organization Practices	4	0.307318149	13

TABLE VI. RANKING AND INFLUENTIAL LEVEL OF SITUATIONAL MCR FACTORS FOR SOURCE CODE SUBMISSION

Most influential Situational Factors	Composite Mean Influential Value	Standard Deviation	Rank
Tool	4.8375	0.170782513	1
Technology Availability	4.8	0.405517502	2
Process	4.75	0.396746024	3
Team Dynamics	4.676	0.450678501	4
Organization Practices	4.6	0.48876261	5
Project Attributes	4.3	0.344265186	6
Organization Culture	4.2	0.298142397	7
Project Release Management	4.166	0.349602949	8
Source Code Attributes	4.05	0.333333333	9
Source Code Change	4.025	0.358752984	10
Organization Policies	4.016	0.129099445	11
Test Inclusion	4	0.529150262	12

TABLE VII. RANKING AND INFLUENTIAL LEVEL OF SITUATIONAL MCR FACTORS FOR REVIEWER SELECTION AND NOTIFICATION

Most influential Situational Factors	Composite Mean Influential Value	Standard Deviation	Rank
Team Dynamics	4.888	0.27325202	1
Team Interaction	4.82727	0.325669474	2
Reviewer Response	4.71666	0.440958552	3
Tool	4.625	0.329140294	4
Source Code Attributes	4.5375	0.385501117	5
Project Attributes	4.533	0.394405319	6
Source Code Change	4.5083	0.311804782	7
Process	4.45	0.387298335	8
Technology Availability	4.32	0.333333333	9
Organization Practices	4.25	0.414996653	10
Organization Culture	4.2333	0.344265186	11
Organization Resources	4.18	0.294392029	12
Organization Policies	4.05	0.396746024	13
Project Release Management	4.03	0.182574186	14

Based upon the practicality and influence level of situational MCR factors, the situational MCR framework has been developed to support individual sustainability of software engineers engaged in MCR activities. The developed framework guides the MCR team members about the situational factors as well as most influential situational MCR factors for each MCR activity. The developed framework might also help the software engineers to be aware of upcoming situations and to improve their competence and productivity. The situational MCR framework is attached in the Appendix I (Fig. 7a and Fig. 7b).

TABLE VIII. RANKING AND INFLUENTIAL LEVEL OF SITUATIONAL MCR FACTORS FOR SOURCE CODE REVIEW

Most influential Situational Factors	Composite Mean Influential Value	Standard Deviation	Rank
Source Code Attributes	4.925	0.252762515	1
Source Code Change	4.866	0.288675135	2
Test Inclusion	4.84	0.359010987	3
Team Dynamics	4.792	0.354024481	4
Team Interaction	4.781818	0.386645767	5
Reviewer Response	4.683	0.341565026	6
Tool	4.5625	0.305050087	7
Organization Culture	4.45	0.36767538	8
Process	4.4	0.272165527	9
Organization Policies	4.383333	0.36767538	10
Technology Availability	4.38	0.380058475	11
Project Attributes	4.25	0.423827358	12
Defects	4.15	0.319142369	13
Organization Practices	4.1	0.25819889	14
Organization Training	4.02	0.391578004	15

TABLE IX. RANKING AND INFLUENTIAL LEVEL OF SITUATIONAL MCR FACTORS FOR SOURCE CODE APPROVAL

Most influential Situational Factors	Composite Mean Influential Value	Standard Deviation	Rank
Source Code Attributes	4.8125	0.35551215	1
Source Code Change	4.6583	0.267013663	2
Defects	4.6	0.370185139	3
Project Attributes	4.5	0.430331483	4
Team Dynamics	4.464	0.388730126	5
Tool	4.35	0.263523138	6
Organization Culture	4.23	0.316227766	7
Process	4.15	0.324893145	8
Project Release Management	4.1	0.278886676	9
Technology Availability	4.06	0.374165739	10
Organization Policies	4.016	0.36767538	11

V. DISCUSSION

Software engineers engaged in MCR activities belongs to various cultures and backgrounds, as a result, various unknown and varying situational factors arise that impact their sustainability and productivity due to mental distress. Therefore, this study has presented a situational MCR framework based on Delphi survey. The developed framework guides the software engineers about the situational factors as well as the most influential situational MCR factors for each MCR activity. The developed framework might also support the sustainability of software engineers, help them to be aware of upcoming situations, and to improve their productivity. The situational MCR framework is attached in Appendix I (Fig. 7a and Fig. 7b).

VI. Conclusion

This study presented a situational MCR framework to support the sustainability of software engineers engaged in MCR activities. The study findings reported 21 situational factors, 147 sub-factors, and 5 Categories. In this paper, the Delphi survey results along with the development of the situational MCR framework are presented. The developed framework might guide the software engineers engaged in MCR activities to consider the reported situations, identify situations according to their context, and improve their productivity while ensuring their individual sustainability.

VII. Future Work

This study can be further elaborated for other software engineering contexts and activities for instance software design, software testing, etc. The ongoing research plans are to validate the developed situational MCR framework through experiment and to develop a web-oriented situational MCR tool to have an electronic situational guideline for software engineers engaged in MCR activities.

VIII. Contribution

The investigation contributed to the software engineering body of knowledge (SWEBOK) specifically situational software engineering and sustainable software engineering, particularly the individual sustainability of software engineers. The study reported the situational factors for MCR from literature, academic, and industry. The study also recognizes and prioritizes the most influential situational MCR factors and present the situational MCR framework to support the individual sustainability of software engineers engaged in MCR activities.

References

[1] G. Marks, R. V. O'Connor, and P. M. Clarke, "The impact of situational context on the software development process – A case study of a highly innovative start-up organization," Commun. Comput. Inf. Sci., vol. 770, pp. 455–466, 2017.

[2] C. K. Chang, "Situation Analytics: A Foundation for a New Software Engineering Paradigm," Computer (Long. Beach. Calif)., vol. 49, no. 1, pp. 24–33, 2016.

[3] R. V. O. Connor, P. Elger, and P. M. Clarke, "Exploring the impact of situational context – A case study of a software development process for a microservices architecture," pp. 6–10, 2016.

[4] B. Penzenstadler et al., "Software Engineering for Sustainability: Find the Leverage Points!," IEEE Softw., vol. 35, no. 4, pp. 22–33, 2018.

[5] S. Nazir, N. Fatima, S. Chuprat, H. Sarkan, N.F. Nilam, N.A. Sjarif "Sustainable Software Engineering: A Perspective of Individual Sustainability Challenges," Int. J. Adv. Sci. Eng. Inf. Technol., vol. 10, no. 2, pp. 676–683, 2020.

[6] S. Nazir, N. Fatima, and S. Chuprat, "Individual Sustainability Barriers and Mitigation Strategies: Systematic Literature Review Protocol," in 2019 IEEE Conference on Open System, ICOS 2019, 2019, pp. 1–5.

[7] B. Penzenstadler, A. Raturi, D. Richardson, C. Calero, H. Femmer, and X. Franch, "Systematic Mapping Study on Software Engineering for Sustainability (SE4S)," in Proc. 18th International Conference on Evaluation and Assessment in Software Engineering, 2014, pp. 1–14.

[8] P. Clarke and R. V. O'Connor, "The situational factors that affect the software development process: Towards a comprehensive reference framework," Inf. Softw. Technol., vol. 54, no. 5, pp. 433–447, 2012.

[9] M. Salam and S. U. Khan, "Challenges in the development of green and sustainable software for software multisourcing vendors: Findings from a systematic literature review and industrial survey," J. Softw. Evol. Process, vol. 30, no. 8, pp. 1–21, 2018.

[10] D. Mishra, S. Aydin, A. Mishra, and S. Ostrovska, "Knowledge management in requirement elicitation: Situational methods view," Comput. Stand. Interfaces, vol. 56, no. September, pp. 49–61, 2018.

[11] K. Gulzar, J. Sang, A. A. Memon, and M. Ramzan, "A Practical Approach for Evaluating and Prioritizing Situational Factors in Global Software Project Development," vol. 9, no. 7, 2018.

[12] R. Chitchyan, I. Groher, and J. Noppen, "Uncovering sustainability concerns in software product lines," J. Softw. Evol. Process, vol. 29, no. 2, pp. 1–20, 2017.

[13] S. Nazir, N. Fatima, and S. Chuprat, "Situational factors for modern code review to support software engineers' sustainability," Int. J. Adv. Comput. Sci. Appl., vol. 11, no. 1, pp. 498–504, 2020.

[14] R. Chitchyan, L. Duboc, C. Becker, S. Betz, B. Penzenstadler, and C. C. Venters, "Sustainability Design in Requirements Engineering : State of Practice," in IEEE/ACM 38th IEEE International Conference on Software Engineering, 2016, pp. 533–542.

[15] N. Fatima, S. Chuprat, and S. Nazir, "Challenges and Benefits of Modern Code Review-Systematic Literature Review Protocol," in Proc. International Conference on Smart Computing and Electronic Enterprise, 2018, pp. 1–5.

[16] A. Bacchelli and C. Bird, "Expectations, outcomes, and challenges of modern code review," in Proc. International Conference on Software Engineering, 2013, pp. 712–721.

[17] N. Fatima, S. Nazir, and S. Chuprat, "Understanding the Impact of Feedback on Knowledge Sharing in Modern Code Review," in 6th IEEE International Conference on Engineering Technologies and Applied Sciences (ICETAS) , 2019.

[18] L. MacLeod, M. Greiler, M. A. Storey, C. Bird, and J. Czerwonka, "Code Reviewing in the Trenches: Challenges and Best Practices," IEEE Softw., vol. 35, no. 4, pp. 34–42, 2018.

[19] A. Bosu, J. C. Carver, C. Bird, J. Orbeck, and C. Chockley, "Process Aspects and Social Dynamics of Contemporary Code Review: Insights from Open Source Development and Industrial Practice at Microsoft," IEEE Trans. Softw. Eng., vol. 43, no. 1, pp. 56–75, 2017.

[20] N. Fatima, S. Nazir, and S. Chuprat, "Knowledge sharing, a key sustainable practice is on risk: An insight from Modern Code Review," in 2019 6th IEEE International Conference on Engineering Technologies and Applied Sciences (ICETAS), 2019.

[21] H. H. Khan, M. N. Mahrin, and S. Chuprat, "Situational requirement engineering framework for Global Software Development," 2014 Int. Conf. Comput. Commun. Control Technol., no. I4ct, pp. 224–229, 2014.

[22] P. Clarke, R. V. O. Connor, R. V. O. Connor, and B. Leavy, "A complexity theory viewpoint on the software development process and situational context," no. May, 2016.

[23] S. Nazir, N. Fatima, and S. Chuprat, "Does Project Associated Situational Factors have Impact on Sustainability of Modern Code Review Workforce?," in 6th IEEE International Conference on Engineering Technologies and Applied Sciences (ICETAS) , 2019

[24] S. Nazir, N. Fatima, and S. Chuprat, "Situational factors affecting Software Engineers Sustainability: A Vision of Modern Code Review," in 6th IEEE International Conference on Engineering Technologies and Applied Sciences (ICETAS), 2019.

[25] S. Ghosh, A. Dubey, and S. Ramaswamy, "C-FaRM: A collaborative and context aware framework for requirements management," 2011 4th Int. Work. Manag. Requir. Knowledge, MaRK'11 - Part 19th IEEE Int. Requir. Eng. Conf. RE'11, pp. 29–30, 2011.

[26] K. Bakhat, A. Sarwar, and Y. H. Motla, "A Situational Requirement Engineering Model for an Agile Process," Bahria Univ. J. Inf. Commun. Technol. ISSN – 1999-4974, vol. 8, no. 1, pp. 21–26, 2015.

[27] F. Ebert, F. Castor, N. Novielli, and A. Serebrenik, "Confusion in Code Reviews: Reasons, Impacts, and Coping Strategies," SANER 2019 - Proc. 2019 IEEE 26th Int. Conf. Softw. Anal. Evol. Reengineering, pp. 49–60, 2019.

[28] C. Sadowski, E. Söderberg, L. Church, M. Sipko, and A. Bacchelli, "Modern code review: : A Case Study at Google," in Proc. ACM/IEEE 40th International Conference on Software Engineering: Software Engineering in Practice, 2018, pp. 181–190.

[29] O. Kononenko, T. Rose, O. Baysal, M. Godfrey, D. Theisen, and B. De Water, “Studying Pull Request Merges : A Case Study of Shopify ’ s Active Merchant,” in Proc. 40th International Conference on Software Engineering: Software Engineering in Practice, 2018, pp. 124–133.

[30] B. Kitchenham and S. Charters, “Source: " Guidelines for performing Systematic Literature Reviews in SE " , Kitchenham et al Guidelines for performing Systematic Literature Reviews in Software Engineering Source: " Guidelines for performing Systematic Literature Reviews i,” pp. 1–44, 2007.

[31] B. Ayyub, “A practical guide on conducting expert-opinion elicitation of probabilities and consequences for corps facilities,” Inst. Water Resour. Alexandria, VA, USA, no. January, 2001.

[32] H. A. von der Gracht, “Consensus measurement in Delphi studies. Review and implications for future quality assurance,” Technol. Forecast. Soc. Change, vol. 79, no. 8, pp. 1525–1536, 2012.

[33] G. J. Skulmoski, F. T. Hartman, and Jennifer Krahn, “The Delphi Method for Graduate Research,” J. Inf. Technol. Educ., vol. 6, 2007.

[34] M. Adler and Z. Erio, Gazing Into the Oracle: The Delphi Method and Its Application to Social Policy and Public Health. Jessica Kingsley, 1996.

[35] H. W. Lanford, Technological forecasting methodologies; a synthesis. New York: American Management Association, 1972.

[36] N. J. Shariff, “Utilizing the Delphi Survey Approach: A Review,” J. Nurs. Care, vol. 04, no. 03, 2015.

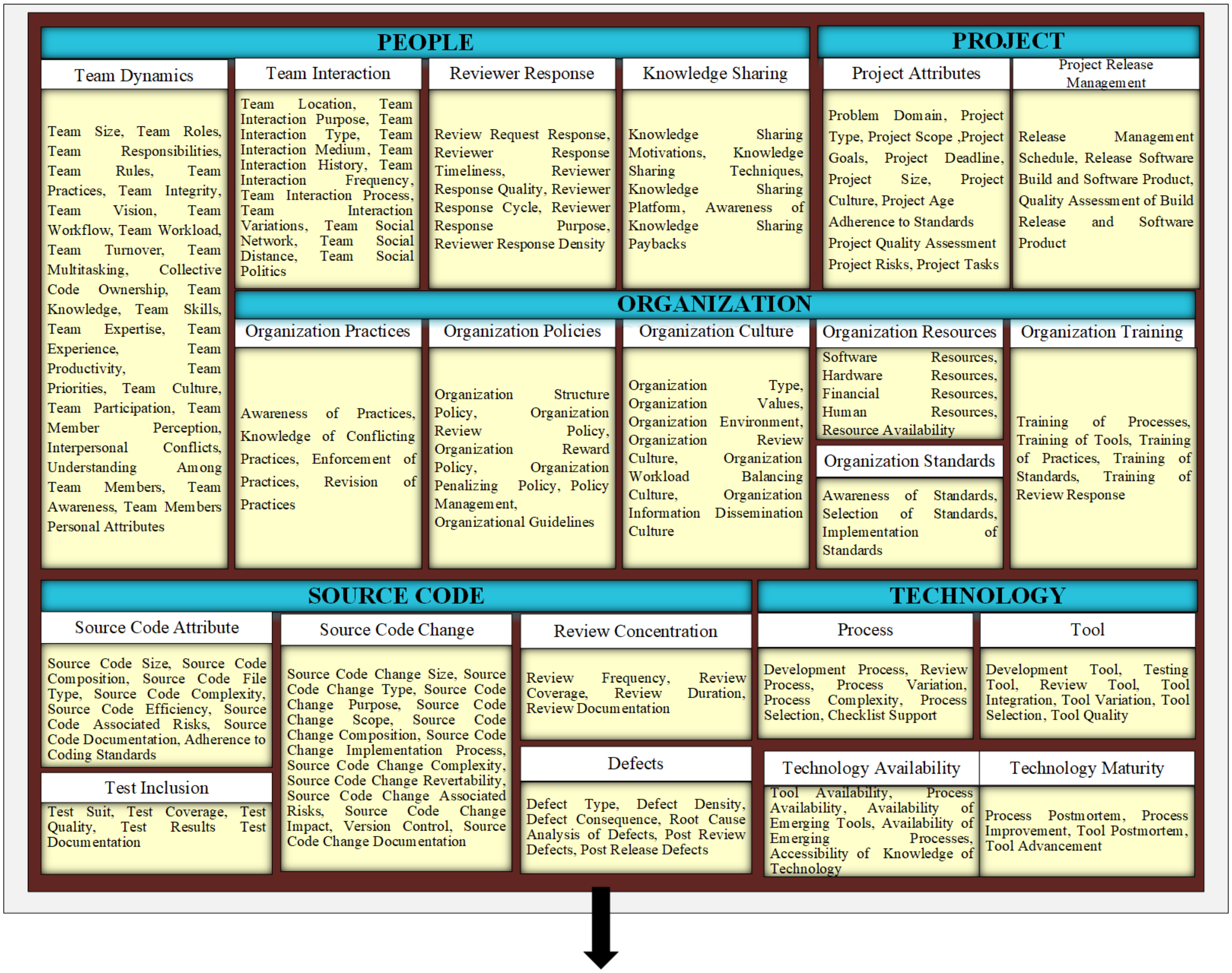
[37] S. G. Naoum, Dissertation research and writing for construction students, Second Edition, 2nd ed. Oxford : Butterworth-Heinemann, 2012.

[38] J. Cohen, Statistical power analysis for the behavioral science, 2nd ed. New: Lawrence Erlbaum Associates, 1988.

[39] M. R. Rogers and E. C. Lopez, “Identifying critical cross-cultural school psychology competencies,” J. Sch. Psychol., vol. 40, no. 2, pp. 115–141, 2002.

APPENDIX I

Situational MCR Framework



PEOPLE

Team Dynamics

Team Size, Team Roles, Team Responsibilities, Team Rules, Team Practices, Team Integrity, Team Vision, Team Workflow, Team Workload, Team Turnover, Team Multitasking, Collective Code Ownership, Team Knowledge, Team Skills, Team Expertise, Team Experience, Team Productivity, Team Priorities, Team Culture, Team Participation, Team Member Perception, Interpersonal Conflicts, Understanding Among Team Members, Team Awareness, Team Members Personal Attributes

Team Interaction

Team Location, Team Interaction Purpose, Team Interaction Type, Team Interaction Medium, Team Interaction History, Team Interaction Frequency, Team Interaction Process, Team Interaction Variations, Team Social Network, Team Social Distance, Team Social Politics

Reviewer Response

Review Request Response, Reviewer Response Timeliness, Reviewer Response Quality, Reviewer Response Cycle, Reviewer Response Purpose, Reviewer Response Density

Knowledge Sharing

Knowledge Sharing Motivations, Knowledge Sharing Techniques, Knowledge Sharing Platform, Awareness of Knowledge Sharing Paybacks

PROJECT

Project Attributes

Problem Domain, Project Type, Project Scope ,Project Goals, Project Deadline, Project Size, Project Culture, Project Age Adherence to Standards Project Quality Assessment Project Risks, Project Tasks

Project Release Management

Release Management Schedule, Release Software Build and Software Product, Quality Assessment of Build Release and Software Product

ORGANIZATION

Organization Practices

Awareness of Practices, Knowledge of Conflicting Practices, Enforcement of Practices, Revision of Practices

Organization Policies

Organization Structure Policy, Organization Review Policy, Organization Reward Policy, Organization Penalizing Policy, Policy Management, Organizational Guidelines

Organization Culture

Organization Type, Organization Values, Organization Environment, Organization Review Culture, Organization Workload Balancing Culture, Organization Information Dissemination Culture

Organization Resources

Software Resources, Hardware Resources, Financial Resources, Human Resources, Resource Availability

Organization Standards

Awareness of Standards, Selection of Standards, Implementation of Standards

Organization Training

Training of Processes, Training of Tools, Training of Practices, Training of Standards, Training of Review Response

SOURCE CODE

Source Code Attribute

Source Code Size, Source Code Composition, Source Code File Type, Source Code Complexity, Source Code Efficiency, Source Code Associated Risks, Source Code Documentation, Adherence to Coding Standards

Test Inclusion

Test Suit, Test Coverage, Test Quality, Test Results Test Documentation

Source Code Change

Source Code Change Size, Source Code Change Type, Source Code Change Purpose, Source Code Change Scope, Source Code Change Composition, Source Code Change Implementation Process, Source Code Change Complexity, Source Code Change Revertability, Source Code Change Associated Risks, Source Code Change Impact, Version Control, Source Code Change Documentation

Review Concentration

Review Frequency, Review Coverage, Review Duration, Review Documentation

Defects

Defect Type, Defect Density, Defect Consequence, Root Cause Analysis of Defects, Post Review Defects, Post Release Defects

TECHNOLOGY

Process

Development Process, Review Process, Process Variation, Process Complexity, Process Selection, Checklist Support

Tool

Development Tool, Testing Tool, Review Tool, Tool Integration, Tool Variation, Tool Selection, Tool Quality

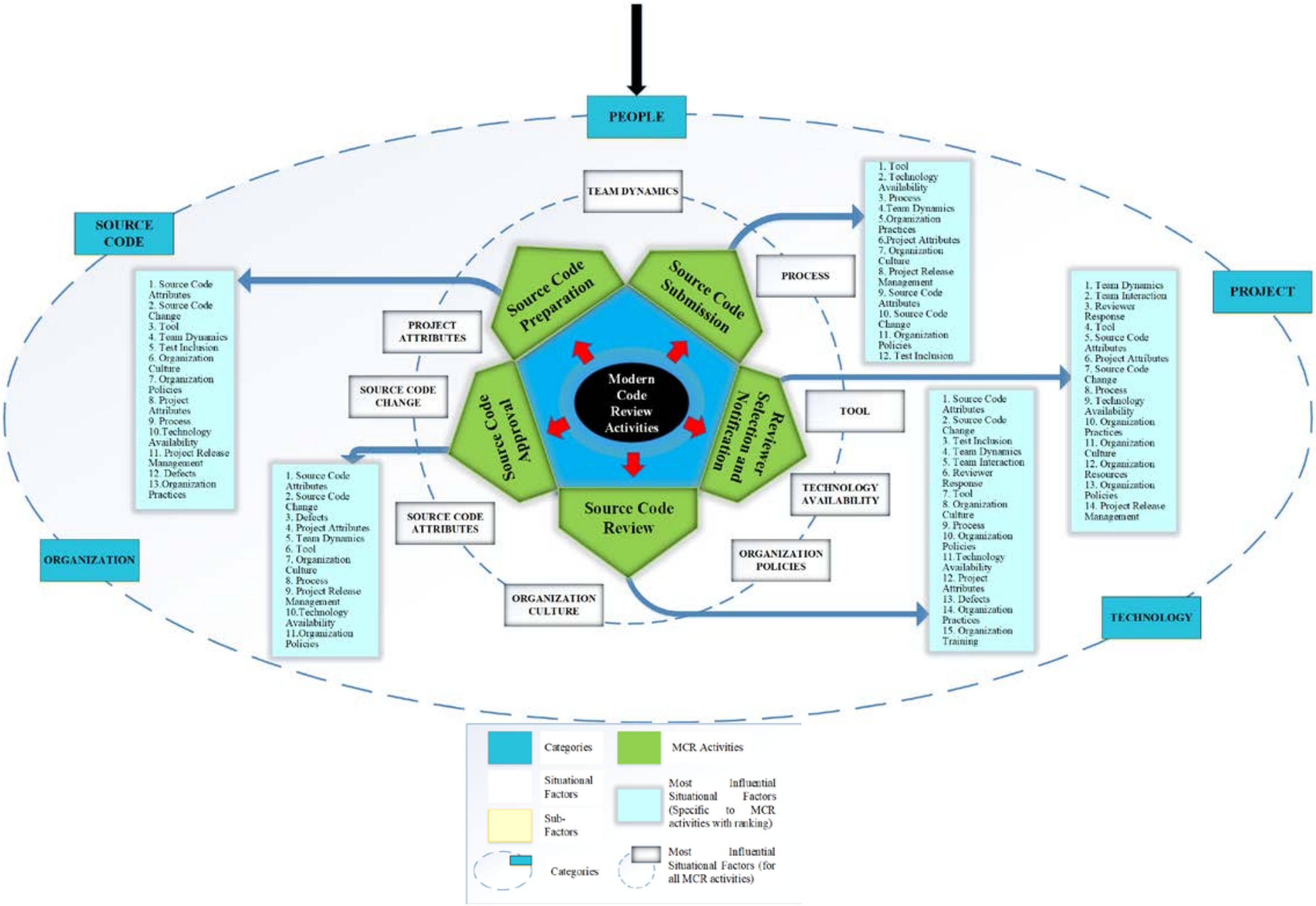
Technology Availability

Tool Availability, Process Availability, Availability of Emerging Tools, Availability of Emerging Processes, Accessibility of Knowledge of Technology

Technology Maturity

Process Postmortem, Process Improvement, Tool Postmortem, Tool Advancement

(a)



(b)

Fig 7. (a) Situational MCR Framewrok for Sustainable Software Engineers- Part I, (b) Situational MCR Framewrok for Sustainable Software Engineers- Part II.