Pak. j. life soc. Sci. (2024), 22(2):14781-14812 E-ISSN: 2221-7630;P-ISSN: 1727-4915

Pakistan Journal of Life and Social Sciences
www.pjlss.edu.pk

https://doi.org/10.57239/PJLSS-2024-22.2.001063

RESEARCH ARTICLE

Development of Taxonomy for Addressing Decision-Related Challenges of Bug Prioritization and its Use for Scrum Retrospective

Sohaib Altaf Raja[1*], Madihah Sheikh Abdul Aziz[2], Idrees Alsolbi[3], Abid Ghaffar[4]

[1] Department of Computer Science KICT, International Islamic University Malaysia, Kuala Lumpur, Malaysia
[2] Department of Information Systems KICT, International Islamic University Malaysia, Kuala Lumpur, Malaysia
[3] Department of Data Science College of Computing, Umm Al- Qura University, Saudi Arabia
[4] Department of Software Engineering, College of Computing, Umm Al-Qura University, Saudi Arabia

ARTICLE INFO

Received: Sep 2, 2024

Accepted: Oct 16, 2024

Keywords

Bug Prioritization, Bug Triage

Taxonomy

Datasets of Bug Reports

Past Data

Role of the Bug Triagers

Decision-Making

Scrum Retrospective

***Corresponding Author:**

suhaibraja@hotmail.com

ABSTRACT

Empirical studies have shown that the software maintenance phase contributes approximately two-thirds of the project development costs. Prioritizing bugs is an essential decision-related task that influences maintenance activities. The empirical studies on bug prioritization (BP) have shown a gap in addressing the decision-related challenges in prioritization bug reports. The literature illustrates the significance of open-access bug reports in deriving valuable insights for bug triagers from past data containing decision-related knowledge for BP. This paper assumes that learning by employing a retrospective approach using bug reports of previous sprints is useful for addressing the decision-related challenges for BP. The literature review reveals that studies do not prescribe which substantial BP decision-related knowledge is documented in bug reports that may be significant for retrospective meetings. Therefore, the objective of this paper is to qualitatively explore the components of BP decision-related data contained in past bug reports, classify them, and evaluate their significance for scrum retrospectives. In the findings of this paper, a taxonomy is created that effectively structures components of BP decision-related knowledge. The findings reveal that the past bug reports contain significant BP decision-related knowledge that may be used to observe correlated factors and obtain insights for the bug triagers that aid them in addressing the decision-related challenges of BP. The illustrative approach and focus group method are employed for validation in which the significance of the findings is evaluated.

INTRODUCTION

It is critical to resolve software bugs appropriately to successfully deploy software solutions (Jahanshahi et al., 2022). An abundant backlog of bugs is a serious constraint on the performance of software projects because it increases the maintenance cost (Kanwal, 2010; Uddin et al., 2017). Empirical studies have shown that the software maintenance phase contributes approximately two-thirds of the project development costs (Uddin et al., 2017). Bug triaging is an important task of the software maintenance phase that should deal with the bugs when they are reported and handle them for accurate and timely prioritization so that important bug reports can be addressed promptly (Almhana & Kessentini, 2021; Li et al., 2024; Noei et al., 2019; Uddin et al., 2017). The process of BP is of the utmost importance in dealing with quality and timely resolution of bug reports. Accurate and timely prioritization and resolution of bug reports not only improve the quality of software

maintenance tasks but also provide the basis to keep particular software systems alive (Uddin et al., 2017).

The BP process is an essential decision-related task. Abundant studies have been conducted on the importance of effective DM (Cunha et al., 2016; Hesse et al., 2016). Information is a key factor for decision-makers to facilitate the DM process effectively while unavailability of enough information can delay the decision (Citroen, 2011; Da Cunha et al., 2016). According to (PMI, 2015), 47% of unsuccessful projects are impacted by poor DM, therefore, factors impacting the decisions e.g., such as sufficient information, require attention (Cunha et al., 2016). Literature reports various decision-related challenges in prioritizing bugs (Akbarinasaji et al., 2020; Jahanshahi et al., 2022; Raja et al., 2023).

Software companies are coping with various decision-related challenges in handling BP tasks such as a handling large backlog of bug reports, changeovers in priority decisions, and so on (Almhana et al., 2020b; Gökçeoğlu & Sözer, 2021; Li et al., 2024; Uddin et al., 2017). According to the literature, a decision problem occurs in the BP process due to many factors including lack of automation support in DM, limitations in the cognitive ability of decision makers, weak insight of decision makers due to not enough information, and so on (Akbarinasaji et al., 2020; Jahanshahi et al., 2022; Kaushik et al., 2013; Noei et al., 2019; Raja et al., 2023; Uddin et al., 2017; Zhang et al., 2015a). Therefore, the studies highlight the importance of the role of decision-makers and the availability of information that impacts DM tasks for prioritizing bug reports.

The literature reports a gap in addressing the decision-related problems in prioritization tasks and highlights challenges in addressing them (Akbarinasaji et al., 2020; Jahanshahi et al., 2022; Kaushik et al., 2013). In order to provide insight into the bug triagers handle BP tasks and address decision-related challenges, this paper explains the significance of the Scrum retrospective as a DM approach that uses historical data as a source of information from prior bug reports. The study examines the DM phenomena of BP using past data from previous bug reports. Besides, to describe the phenomena of BP from the perspective of DM, it needs to explore the factors associated with sources of information that influence DM**.** Therefore, this research studies the phenomena of DM tasks for BP and explores various factors associated with sources of information from past bug reports. The data is collected from empirical studies on BP, contents of past bug reports, and online resources belonging to software corporations such as Atlassian, Apache, and so on.

Bug reports are primary artifacts containing essential information about reported bugs. Bug reports contain past information, which can be utilized to derive other bug statistics, evaluate various facets of their quality, and study their behavior (Bettenburg & Just, 2008; Gökçeoğlu & Sözer, 2021; Hu et al., 2014; Laiq et al., 2023; Noyori et al., 2019). While it helps to handle the bugs and to evaluate many other factors for DM (Hesse et al., 2016). The knowledge contained in bug reports is therefore referred to by Hesse et al., (2016) as "decision knowledge" and it is explained in terms of different decision components.

According to the literature bug reports record the actions of people involved in DM. This implies the importance of previous bug reports as a useful source of information that reflects past events, and it demonstrates the usefulness of past data for DM to handle bugs. The importance of bug reports as digital project artifacts for the retrospective meeting is highlighted (Matthies, 2019). Thus, bug reports as a prominent source of information can be employed for addressing decision-related challenges in handling BP decisions.

The Role of Bug Triager in Handling BP Tasks

The selection of appropriate priority decisions is of utmost importance for the efficient resolution of bugs (Uddin et al., 2017). However, the changeovers in priority decisions are noticed in many bug

reports, where priority needs to be reassigned 2(Almhana & Kessentini, 2021; Li et al., 2024). The lifecycle of the Eclipse bug report shows several aspects of the bug-handling process. The role of the bug triagers is key in the effective handling of BP tasks (Raja et al., 2023; Xie et al., 2013). The workflow explains the BP tasks and the role of the people involved in handling bugs (Zhang et al., 2015a). In Eclipse, Red Hat, Apache, Atlassian, and Mozilla projects, the bug reporter has to perform various tasks in which he validates the bug, examines the necessary information for creating a bug report, checks its duplicate existence, reproduces the bug, diagnoses it, communicates with other stakeholders using various labels, and prepares it for prioritization (Akbarinasaji et al., 2020; Jahanshahi et al., 2022; Zhang et al., 2015a).

A decision is a cognitive process of human behavior that involves gaining knowledge by understanding through a thought process, learning from experience, and choosing a course of action from among several choices to arrive at that conclusion (Cohen J, 2013; Dean & Sharfman, 1996; Marler & Arora, 2004; White, 2018). The human aspect is essential in DMP because DMP is not only a data-driven task but also a people-driven task (Mendes et al., 2021). Because of the importance of the human aspect, it is important to explore his roles in addressing the decision problem. A decision-maker can learn from his experience by assessing the effectiveness of past decisions in order to predict the likeliness of future consequences. This gives him insight to judge the available information and investigate necessary purposeful information (Tsoukias et al., 1994). It highlights that the role of the bug triagers as a decision maker as well the role of different people connected with bug handling workflow needs to be elaborated whereas it is emphasized that bug triagers need better insights and precision to make effective decisions in handling bugs for prioritization tasks.

Scrum Retrospecting

Learning lessons from historical data on software projects is one of the significant practices for organizations to understand the factors behind past events and actions (PMI, 2023; The Standish Group, 2023). DM is among the core tasks for Scrum meetings (Drury-Grogan et al., 2017; Matthies, 2019; Scrum Alliance, 2018; Verwijs & Russo, 2023). It can facilitate quick DM by providing useful data, frequent feedback, and collaboration among teams, and customers to make decisions (Cunha et al., 2016; Ghozali et al., 2019; Svensson et al., 2019).

A survey conducted by the Scrum Alliance, (2018) reports that 81% of the participants hold scrum retrospective meetings to identify opportunities for future improvements. The retrospective meeting is an event that reflects the team performance of the past sprints by gaining experience from past actions of team members, provides the opportunity to identify the areas for improvement, and carries over the action items into the next sprint (Drury-Grogan et al., 2017; Loeffler M, 2017; Matthies, 2019; Scrum Alliance, 2023). It uses past data as a significant source to generate insights for DM tasks and provide indicators to the bug-handling team for conducting daily scrum meetings. On the other hand, the availability of digital project artifacts, such as bug reports, makes it possible to collect the development data for retrospective meetings with less overhead (Matthies, 2019). Hence, scrum retrospective is a useful practice for improving performance in the overall organization by improving their DM capacity (Matthies, 2019; Verwijs & Russo, 2023).

Plenty of research has been conducted on bug reports over more than a decade examining the phenomena of handling bug reports, reveals that they contain valuable information for DM and thoroughly describe the behavior and their characteristics (Abou Khalil et al., 2019; Hesse et al., 2016; Jahanshahi et al., 2022; Keung, 2017). This motivates looking into the significance of the contents of previous bug reports for scrum retrospectives. It is noticed that studies have given more focus to the development of AI and ML-based techniques and less focus is given to other aspects such as the quality of data (Svensson et al., 2019). Hence, there is a need for more research that directly addresses the challenges and opportunities in scrum retrospectives.

Research Gap

For appropriate handling of BP tasks, insights from past data can be significant and helpful for bug triagers to accomplish correct and timely prioritizing of bug reports. BP experiences considerable changeovers in priority decisions; consequently, bug triagers require deeper insight to make suitable prioritization decisions, which motivates us to look back to notice this changeability in BP decisions (Almhana et al., 2020b; Jahanshahi et al., 2022; Raja et al., 2023). Many research studies have been conducted on scrum methodologies and industrial surveys show that applying the retrospective approach benefits industry professionals. This emphasizes the necessity to investigate various aspects of scrum methodologies, their significance to the bug triagers, and their impact on BP (Matthies, 2020; Verwijs & Russo, 2023). Therefore, introducing a retrospective approach to handling BP tasks is a good prospect. A literature review reveals that learning by retrospective approach is not introduced for the bug triagers to handle BP tasks. Furthermore, the empirical studies on BP do not prescribe substantial components of decision knowledge related to BP contained in previous bug reports that may be significant for retrospective meetings. Further, they need to be organized under some taxonomy since it gives researchers and practitioners a common terminology that helps them comprehend relationships between classified categories and sub-categories (Usman et al., 2017).

According to our preliminary study, bug reports of previous sprints contain qualitative and quantitative information about BP tasks such as comments indicating the number of events that occurred, and actions of bug triagers were recorded in bug reports, which are associated with triaging and prioritization tasks; history of bug reports also reveals the number of tasks conducted for information gathering, adding and updating information in the description, and attaching relevant information with a bug report. In past bug reports, decision-related knowledge is documented such as that may be significant for learning lessons in prioritizing bug reports, such as DM tasks performed, events that occurred, actions taken, and so on whereas tracing rationales of documented decisions can be useful subjects for retrospective study.

Furthermore, our prior work shows the gap in utilizing the past data for BP, uses the past events from earlier bug reports to characterize the descriptive workflow of BP, and exhibits the use of the workflow for a retrospective DM approach (Raja et al., 2023). This workflow provides various insights into the bug triagers on handling BP tasks. Based on the literature review, the preliminary study on previous bug reports, and our prior work, we assume develop a hypothesis in this study that different components of decision knowledge related to BP tasks are documented in previous bug reports which can be useful for retrospective meetings and provide insights to the bug triagers into future BP tasks.

The objective of this paper is to explore the components of decision knowledge related to triaging and prioritization tasks contained in historical bug reports and identify insights that can aid the bug triagers in handling BP tasks retrospectively. Therefore, this study introduces the use of a retrospective approach to address the problem of BP. Thus, this research has a twofold contribution; it describes the components of decision knowledge related to BP contained in bug reports, and secondly, it presents the learning by retrospective approach from bug reports of previous sprints. This gap presents an opportunity for future research to evaluate and develop new approaches and tools for handling BP tasks using the past data of bug reports created in previous sprints. This paper addresses the following research question.

Research Question: Which decision-related past triaging and prioritization data may be gleaned using previous bug reports that would be significant for retrospective meetings?

In the first stage, this question qualitatively explores the significant decision-related triaging and prioritization data contained in bug reports using the prior bug reports of large software

corporations. In this stage, a taxonomy is created which classifies the components of decision knowledge into various categories. This serves as a guideline for comprehending valuable content from a bug report related to BP tasks and provides valuable insights to the bug triagers for Scrum retrospective in prioritizing bug reports.

This taxonomy is further evaluated in two phases. In the first phase, an illustrative approach is used for the evaluation of the utility of the proposed taxonomy in which qualitative data is gleaned from the selected bug report. The focus group method is employed in the second phase to assess the significance of the proposed taxonomy, and insights provided by it for the Scrum retrospective meetings to aid bug triagers in handling BP tasks.

The order of the remaining sections is as follows. The literature review is presented in Section 2. Details of the study design are provided in Section 3. Section 4 contains the findings. Section 5 describes the validation. Section 6 includes the discussion. The conclusions and future directions are presented in Section 7.

LITERATURE REVIEW

This section presents the literature review studies on handling BP tasks.

Uddin et al., (2017) surveys BP processes and highlights the importance of accurate and timely prioritization for efficiently handling bug-triaging tasks. Problems of DM in prioritizing bugs are highlighted in the literature (Jahanshahi et al., 2022; Li et al., 2024; Uddin et al., 2017) which cause inaccuracy in priority decisions (Gökçeoğlu & Sözer, 2021). Abundant empirical studies are published on bug reports, analyzed their contents, and demonstrated for different bug-handling perspectives (Hesse et al., 2016; Jahanshahi et al., 2022; Matthies et al., 2020; Saha et al., 2015; Uddin et al., 2017; Zhang et al., 2015a).

Plenty of studies have been specifically conducted on handling BP tasks using past data from bug reports. Several empirical studies proposed automated techniques for prioritizing bugs to improve bug resolution and emphasize their significance. Some studies used qualitative features using past data from previous bug reports for their empirical work (Feng et al., 2016; Kaushik et al., 2013; Kumari & Singh, 2020; Raja et al., 2023; Tian et al., 2015; Xu et al., 2016). Others presented quantitative data to show the statistics on BP decisions (Almhana et al., 2020a; Gökçeoğlu & Sözer, 2021; Jahanshahi et al., 2022; Li et al., 2024). This illustrates the significance of bug reports in deriving past data for BP.

Many studies proposed ML and DL-based techniques for predicting priority and evaluated their techniques based on their performance (Bani-Salameh et al., 2021; Uddin et al., 2017). Kanwal, (2010) proposes a classification-based approach for improving BP and finds that SVM performs better for text features while Naïve Bayes performs better for categorical features. Bani-Salameh et al., (2021), apply machine learning techniques to predict priority, showing encouraging results in reducing the time spent on BP. However, there is a gap in the literature regarding evaluating them for industry use as well as putting them into practice (Jahanshahi et al., 2022; Uddin et al., 2017; Zhang et al., 2015a).

Hesse et al., (2016) analyze the contents of bug reports qualitatively to observe the phenomena of DM. Zhang et al., (2015b) provide a review on bug handling and classify the research work into bug triaging, prioritization, and developer selection processes. Raja et al., (2023) characterized the workflow of BP tasks and illustrated the states, statuses, and transitions between them, however, BP tasks that are performed during the lifecycle of bug reports need to be identified. According to the literature, the bug reports are a discussion forum and contain an evolutionary nature of data that is not being employed for designing empirical studies for BP and is not explored in-depth (Jahanshahi et al., 2022; Raja et al., 2023). To effectively triage and prioritize the bug reports, a retrospective

approach can provide valuable insights to the bug triagers using past data. Besides, it is important to move towards data-informed retrospectives for handling BP tasks rather than rely on the perceptions and opinions of the team.

METHODOLOGY

Empirical studies reveal that bug reports are the platform that records the actions of software teams when they communicate (Hesse et al., 2016; Matthies, 2020; Noyori et al., 2019). Plenty of research studies use bug reports as a subject for different statistical inferences and suppositions. The structure of this study is based on ex-post facto design. The ex-post facto is "after-the-fact" or "retrospective" research in which research starts after the fact has occurred without any interference from the researcher (Höfer & Tichy, 2007). This research design is useful for retrospective study because it looks into past factors using the facts and events that have occurred in the past (Politowski et al., 2018).

By employing the ex post facto design, the outcome of bug priority decisions and the history of the associated events and actions taken during the DM tasks for handling BP tasks can be traced back, so that decisions-related issues in BP could be investigated. Fig 1 illustrates the research design process. This study investigates the retrospective data in the bug report that correlates with BP and describes its significance for the DM for handling BP tasks.

The research process used in this study is qualitative while the research reasoning employed included both deductive and inductive reasoning (Wohlin & Aurum, 2015). Categorical, temporal, and people labels are identified as pre-determined categories from the literature for the categorization of BP-related DM knowledge available in the bug reports (Jahanshahi et al., 2022; Kumari & Singh, 2020; Tian et al., 2015), while other categories are identified inductively. Using the pre-conceived categories, in the second step, BP-related categories are explored inductively from DM knowledge contained in the bug report.

Collection of Data and its Analysis

The bug reports are taken as a primary unit of observation in this study for data collection whereas the nature of the data is qualitative. The BP-related data available in the repository of bug reports is taken as a unit of analysis for qualitative study. The qualitative data is analyzed by thematic analysis. The sampling technique for selecting bug reports is purposive because not all bug reports are rich in documenting bug triaging and prioritization-related data. This paper collects data from open-source bug reports of various software corporations[i], literature on DM, empirical studies on BP tasks, and online resources belonging to Atlassian, Mozilla, and other software corporations. Atlassian is a large commercial organization that serves over 200000 customers with over 8000 employees. Online sources of software corporations prescribe guidelines for triaging and prioritizing bug reports.

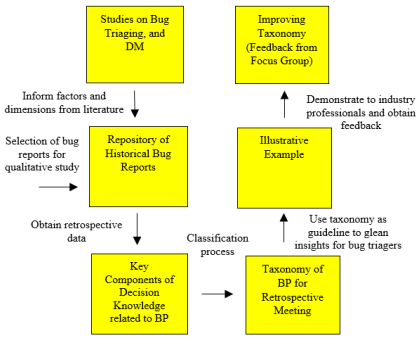

Figure 1: Illustrates the Methodology

The methodology for the collection and analysis of information is explained following steps:

In the first step, data is collected from the literature on DM, triaging and prioritizing tasks, and online sources. This step informs about important factors and dimensions from the literature impacting DM, and triaging and prioritizing tasks that are used for the collection, investigation, and analysis of BP-related data from the bug reports.

In the second step, bug reports are selected for the collection of BP-related features based on data collected in the previous step.

In this step, the decision-related key features interrelated to bug triaging and prioritization tasks logged in the bug reports are explored, using the data collected in the first step.

Data collection and analysis are parallel tasks. Open-source bug reports contain rich information which are structured in various sections. The fourth step involves the classification process in which taxonomy is developed to classify components of decision knowledge contained in open-source bug reports related to BP. This study mainly follows the guidelines for the development of a taxonomy proposed by Usman et al., (2017), while a partial approach is adopted from Kaplan et al., (2022); Ralph, (2019). In this step, a thorough understanding of the subject matter within the field of BP is obtained. The approach used in this study for classification is a faceted analysis because there are several perspectives to view and structure this BP domain (Usman et al., 2017). During the development of the taxonomy, it is ensured that the process should comply with the criteria prescribed by Ralph, (2019) so that it can be aligned with domain knowledge and capture relevant concepts.

Validation Phase

The validity of taxonomies increases their reliability and utility. Usman et al., (2017) highlight three quality criteria to evaluate the taxonomies: orthogonality demonstration, benchmarking, and utility demonstration. Kaplan et al., (2022) define nine quality criteria to evaluate the taxonomies: generality, appropriateness, orthogonality, reliability, correctness, ease of use, novelty, significance, and relevance and structures them in a three-step process for evaluation: evaluating the structure's suitability, evaluating its applicability, and evaluating its purpose. In this paper, the validation is conducted in two phases.

The illustration is the most used approach in demonstrating the utility of software engineering taxonomies; its goal is to make taxonomies more accessible and understandable (Usman et al., 2017). It makes generalization more specific. An illustrative example can be used to either clarify a thought or to support a stance. In the first phase, an illustrative approach using a real-world case is employed to demonstrate the utility of the proposed taxonomy for BP, its ease of use, and its applicability as well as to evaluate its significance for Scrum retrospective meetings.

This phase applies the taxonomy to historical bug reports to validate it with real-world data. These aspects are evaluated in the validation phase. In this phase, a Jira bug report with a rich history of triaging and prioritization tasks is selected and the proposed taxonomy is applied as a guideline to identify the BP-related components of decision knowledge contained in the bug report. Therefore, the contents of the selected bug report are analyzed for how they are related to BP tasks, and qualitative data is gleaned from the selected bug report. This step also functions as a pilot test because the proposed taxonomy is applied to a small dataset in order to test its structure, utility, use, and applicability before being expanded upon in subsequent research studies.

In the second phase focus group method is employed in which, the illustrated example is used to demonstrate the significance of the proposed taxonomy to the industry practitioners and to seek their feedback. Focus group methods are used for validation in this research study which is a valuable

and powerful method in software engineering research for gathering qualitative data, generating new ideas, validating the research findings, and ensuring that the research is grounded in practical and real-world experiences (Kontio et al., 2004). In this phase, participants are selected based on their experiences with bug prioritization tasks, and familiarity with Scrum retrospectives.

The role of the moderator is performed by the principal researcher. It is ensured in this phase that the idea of the research is well communicated to experts so that they should well understand it and propose solutions before participating in the discussion. The research findings are evaluated by experts and professionals using a set of pre-defined questions. The obtained feedback helped in refining the research and ensuring its applicability and significance in real-world scenarios. At this stage, it is crucial to ensure that the taxonomy aligns with the domain knowledge and captures the relevant concepts. Therefore, categories and sub-categories should be refined in adhering to the quality criteria (Kaplan et al., 2022; Usman et al., 2017).

FINDINGS

Open-source bug reports contain rich information which are structured in various sections. The section involves the classification process in which taxonomy is developed to classify components of decision knowledge related to BP contained in bug reports. The guideline used for development is taxonomy from Kaplan et al., (2022); Ralph, (2019); Usman et al., (2017). Taxonomy is a tool to analyze and understand the domain of interest in the field of science and practice. It plays a major role in the structuring knowledge, of a vast body of knowledge in science, classifying approaches and processes as well as fosters understanding of complex domains of software engineering, and provides guidance and insight to the development team use that knowledge. Taxonomy is described as a hierarchy of classes that provides unified terminology to researchers and practitioners and assists them in understanding relationships between classified categories and sub-categories (Usman et al., 2017).

The literature reveals a strong interest in creating new taxonomies in the domain of software engineering. The purpose of developing a new taxonomy in this study is to classify components of decision knowledge in bug reports related to triaging and prioritizing tasks. It can help the bug triagers prioritize bugs during the Scrum retrospective. Because the existing taxonomies don't work for this purpose, the proposed taxonomy needs to be developed. In addition, there are a few taxonomies that deal with people-related subject matters while the proposed taxonomy also deals with people-related subject matters, for instance, the class of people categorizes various roles logged in the bug reports who are involved directly or indirectly with BP tasks.

This study follows the revised taxonomy development method consisting of four phases which is a systematic approach to developing a taxonomy (Usman et al., 2017). The relevant data is collected from various sources including research studies, and industrial empirical data which include past open-source bug reports and online sources of large corporations, and then analyzed and interpreted to develop the new taxonomy. The contents of past open-source bug reports are used as a subject in this study to explore the relationship between BP and other factors. It is noticed that these bug reports are created and updated in prior sprints and are rich in decision knowledge. It has been observed that data related to bug triaging and prioritization tasks, and various components of decision knowledge, are logged in different sections of bug reports. Hence, the findings of this study demonstrate the usefulness of bug reports of prior sprints for retrospective meetings because the data gleaned from these bug reports includes inputs given by the bug triagers in the form of their opinions for DM or actions taken by them as a prioritization decision during the life cycle of the bug

Many empirical studies and online resources on BP are available. Empirical studies use various features related to BP to observe the correlation of different factors with BP and their impact on BP tasks (Kumari & Singh, 2020; Tian et al., 2015; Uddin et al., 2017; Xu et al., 2016). Besides, online sources prescribe policies and guidelines about BP tasks (Apache Software Foundation, 2023;

Atlassian, 2023; Atlassian Bug Fix Policy, 2023; Atlassian Priority Policy, 2023; Atlassian Workflow, 2021; Mozilla Priority and Bug Policy, 2023; Mozilla Triage and Priority Guide, 2023; Mozilla Triage for Bugzilla, 2023). Various factors impacting DM are discussed in the literature (Anvik, 2011; Cunha et al., 2016; Ghozali et al., 2019; Hesse et al., 2016; Klein, 2015; Mendes et al., 2021; White, 2018).

In the first step, different bug-triaging and prioritization-relevant key features and dimensions are identified from the literature and online sources which draw attention to important factors related to handling bug-triaging and prioritization tasks. As observed in our previous study, all bug reports do not contain rich bug triaging and prioritization-related past data. Therefore, in the second step, several bug reports belonging to different software corporations including (Apache Software Foundation, 2023; Atlassian Bug Reports, 2023; Eclipse Foundation, 2023; Red Hat, 2023), were accessed and observed for the collection of data related to BP-related features. Thus, the bug reports are filtered on the basis that they possess bug triaging and prioritization-related features using the data collected in the previous step.

Using the data collected in the first step, the contents of selected bug reports are examined in the third step to explore the decision-related knowledge interrelated to BP logged in the bug reports. In the fourth step, it is required to structure the knowledge of bug reports so that it can be used for triaging and BP during retrospective meetings which emphasize the purpose of creating the taxonomy. Therefore, this step involves the categorization task in which taxonomy is developed which classifies the components of decision knowledge related to BP contained in bug reports. In this step, initial categories and sub-categories are developed based on extracted features while they are further refined through iterative cycles using feedback from academic reviewers and industry practitioners.

Some broad categories are taken from pre-conceived categories while others have evolved inductively. In this step, common patterns of information are identified, and various themes are generated. These patterns are identified and analyzed in parallel with the collected data resulting in the creation of multiple themes. Some categories emerged as new ones during the categorization process, while others were merged into existing ones, for example, the "temporal" and “comments” categories emerged as new ones, while "description" and "workaround" were merged as sub-categories into the "diagnosis” category. Hence, broad categories are broken down into more specific and concise sub-categories. As a result, the contents of bug reports are organized into categories and sub-categories considering their conformation with quality criteria (Kaplan et al., 2022; Usman et al., 2017).

Clear and concise labels are assigned to each category and sub-category. The categories represent the components of decision knowledge including people, categorical, temporal, comments, traceability, history, keywords and labels, and diagnostic information, which is illustrated in Table 1. These components possess qualitative and quantitative characteristics. Therefore, BP-related features are required to be analyzed qualitatively and quantitatively. This study only analyzes the BP-related feature qualitatively; however, it is being expanded upon in subsequent research studies. extended for further development in subsequent studies.

The first column of Table 1 illustrates the broad categories as components of decision knowledge in bug reports whereas the second column illustrates the sub-categories that describe the decision-related triaging and prioritization data. The following section explains the decision-related BP data logged in bug reports. This section deals with the categorization process in which the contents of bug reports are categorized into eight dimensions in relevance to triaging and prioritization tasks, each representing a component of decision knowledge which is explained below.

Diagnostic Information: Diagnostic data that is represented textually is logged in different sections of bug reports such as descriptions, summaries, steps to reproduce, and workaround (Almhana &

Kessentini, 2021; Hesse et al., 2016; Ko & Chilana, 2011). These components are most helpful for bug triagers to assess the importance of the bug, for example, the summary captures the important keywords to explain the diagnostics that can help the bug triagers in locating important bugs; the description provides detailed diagnostic report; workaround describes the proposed solution to a bug; steps to reproduce display important information to revisit the bug. Several research studies employ textual features, such as the length of description and summary, the nature of contents (e.g. emotional words in the vocabulary), choice of specific words, etc., to design automated techniques for BP (Kumari & Singh, 2020; Tian et al., 2015; Umer et al., 2020).

It is analyzed from bug reports that diagnostic information is useful for the bug triagers in analyzing the rationale behind various tasks that are performed for triaging and prioritizing bugs in different timestamps which provides insight into various factors for conducting the retrospective study. This implies that the diagnostic data is a vital source of information to investigate the rationale behind BP decisions as well as observe the factors that affect them.

When a bug is observed, the bug triagers initially triage the bug and diagnose its nature, sensitivity, and importance. Bug details are provided as a description of a bug which is textual content that appears in the "Bug Description" field of the bug report. A bug report's description should include the causes of bugs, their diagnosis, and information about severity and impact. These details are important sources of information for analyzing BP tasks (Feng et al., 2016). The description should contain enough information for the bug triagers so that they can use it for triaging, diagnosing, and assigning accurate priority to complex bugs.

Table 1: Displays the categorization of the contents of bug reports into various components of decision knowledge

Components of Decision Knowledge	Triaging and Prioritization Tasks and Decision-Related Data
Diagnostic Information	Diagnostic data refers to textual data that the bug report narratively records in the following sections of bug reports: descriptions, summaries, steps to reproduce, and workaround. It provides the bug triagers with details about the diagnosis and other key information about the bugs that are useful in observing the rationale behind the priority-related decisions and comments of different people involved in handling prioritization tasks.
Keywords, Tags, and Labels	Keywords, tags, labels, and flags are used to structure and classify textual information in bug reports for better use, while they draw attention to specific characteristics of bugs to differentiate them from others e.g., urgent, important bugs, and hotfixes
Categorical Information	Categorical components reflect the decisions and other actions that are made, they include priority, severity, release/version ID, lifecycle, and resolution status, component, and product, e.g. rationale behind the priority decision taken is logged as diagnostic information and opinion of the decision-makers for assigning a priority is logged as comments whereas actions such as

	priority decision taken are assigned as categorical information.
Traceability Information	Traceability information refers to sources of information that are directly not a part of the bug report instead it needs to be attached as an external source in the form of a traceable link which includes different types of bug reports attached as external sources which include blockers, blocked, related or dependent bug reports; various format of images such as jpg; different nature of documents such as google docs, MS office docs; video clips, different hyperlinks to access information at external source and so on.
People Information	This category refers to people who are involved in various DM tasks to triage and prioritize bug reports. They are either making the decisions or involved in the process of DM and providing their opinions. They include watcher, voter, customer, owner, assignee, and reporter. These roles are defined in a section labeled "People" in a Jira bug report. Some roles provide comments related to diagnosis, triaging, and prioritization that include members of the engineering team which are not defined under the "People" section.
History Information	History contains an evolutionary nature of information that grows as bug reports remain open and can be traced back from the history section. Its categories various components which include a description of past information added or modified, triaging and prioritization tasks performed, events that occurred, and actions taken, type of operation performed which includes either information is added or modified, description of existing and new information, name of the role involved in different operations, and timestamp, e.g. An existing description is modified 8 days ago by Ahmed, an existing priority decision changed from high to medium today by Ali, a new traceability link added 2 days ago by Adam.
Temporal Information	Temporal information categorizes various types of time-based information that is associated with different components of bug reports. This includes information created and modified, people assigned to the bug report, the past events that occurred, and past actions that are taken on different timestamps by various

	software team members who are involved in handling bug reports.
Comments	Comments contain opinions of the customers and the team members about assigning priority and severity, diagnosis given by experts, and other decisions of the team related to prioritization and triaging of bug reports, see Figure 2.

Labels, Keywords, Flags, and Tags: Description, summaries, and comments are unstructured data whereas labels, tags, flags, and keywords are structured. The lack of constructs for structuring the unstructured data poses a challenge for the bug triagers to locate and access the required information. Bugs have a variety of characteristics. Labels, tags, flags, and keywords draw attention to specific characteristics of bugs to differentiate them from others. Thus, the Mozilla, Eclipse, Jira, and Apache bug reports are studied to analyze the use of keywords, tags, labels, and flags in triaging and prioritizing bug reports, which reveals their significance.

Keywords, tags, labels, and flags are used to structure and classify textual information in bug reports for better use, e.g., urgent, important bugs, and hotfixes. These features are helpful for the bug triagers, developers, and other team members in handling BP because they can easily search, and locate critical, urgent, hotfixes, and important bugs when they are tagged with such characteristics in the bug reports. Furthermore, these features may assist the bug triagers in searching and locating valuable past data for the retrospective meeting.

It is examined from bug reports that keywords and label attributes vary amongst software corporations, for instance, some bug reports have keywords and some have labels. Various types of bug reports are analyzed that reveal that the concepts of labels, tags, flags, and keywords are not similar, however, still they are frequently used interchangeably. The use of these features is dependent on the bug-tracking system. For adding keywords and labels, Bugzilla has a “Keyword” field, whereas Jira only has “Labels” fields.

Bugzilla also has a “Whiteboard” attribute, which is a free-form single-line text entry box for adding tags and other status information. Besides, JIRA provides a feature to define specific labels and keywords, so it can be used for handling BP tasks. Keywords are used in a short phrase that contains a combination of two or three words. Keywords are helpful for bug triagers to easily search critical and urgent bugs when they are tagged with such characteristics in the bug reports. On the other hand, large corporations have created a pre-defined list of labels that are helpful for bug triagers to choose from.

- Keywords are used to draw attention to specific characteristics of bugs to differentiate them from others. They are extracted from textual information as a piece of key information or representing key characteristics of a reported bug, for instance, frequent-crash, UAT-bug, alpha-bug, need-urgent-fix, security-bug, and need-immediate-attention.

- Labels are available as a pre-defined list of words used to classify bug reports which can defined in the drop-down field and need to be selected for a reporting bug, e.g. critical, urgent, and hotfixes.

- Tags are more customizable, and details can be created on the spot. They are used when certain details cannot fit into pre-defined labels

- Flags are markers used to indicate specific conditions in a bug report, requiring specific actions or attention.

Figure 2: Illustrates the Opinions of the Bug Triagers About DM, Decision Taken, Change of Decisions, and the Correlated Factors

Keywords and labels can also be classified under categorical information because some large corporations already use pre-defined a list of keywords and labels for the bug triagers to choose from, however, they are identified as a broad category in this study because information classified in category information belongs to a property of a bug, phase, project, product or an artifact which represents as a permanent attribute of a bug. The labels and keywords are used for structuring and classifying the contents of a bug report so that it can be easily located.

Furthermore, they are used for communication purposes. However, they may use to associate or represent the property or attribute of triaging and prioritization tasks temporarily, for example, bugs can be less urgent, or of low importance, but later they can become urgent or important bugs. The definition of keywords, flags, labels, and tags is subjective to organizational context because organizations have different prioritization policies as well as subjective to individual use.

Categorical Information: Several empirical studies used textual and categorical features compositely to design their research for BP (Feng et al., 2016; Kumari & Singh, 2020; Noei et al., 2019; Tian et al., 2015; Umer et al., 2020). Categorical information comprises product, component, hardware, display screen size, operating systems, roles of people, resolution, current and target release number, affects versions, fix versions target milestone, severity, lifecycle status, and priority tasks. The DM tasks are associated with these categorical attributes, few are described below which are illustrated in Table 2.

Table 2: Displays the sub-categories of categorical information

Severity	The severity field communicates decisions related to the assessment of the bug impact.
Target Milestone	Target milestones communicate decisions made for assigning either a current or future version to a bug report.
Priority	Priority communicates a decision related to evaluating the importance and urgency of fixing a bug.
Affects Versions	This field is useful for tracking the version in which a bug has been fixed. Some reported bugs are found in multiple versions which increases their importance in terms of deciding priority.

Fix Versions	This field indicates the version where the bugfix is planned to be released for customers. The bug triagers should know which sprints are planned to be integrated with the fix version so that they can decide on bugs for the highest and high priority.

These categorical fields describe the characteristics and statuses of bugs as well as communicate the decision that is made, e.g., the severity field communicates decisions related to the assessment of the bug impact, target milestones communicate decisions made for assigning either a current or future version to a bug report and priority communicates decisions related to evaluating the importance of bug. It is highlighted in the previous section that the opinions of the bug triagers are logged in textual fields, however, the actions taken by him for handling bug reports are reflected in categorical fields, e.g., a decision taken to assign priority or severity.

Fig 2 shows the opinions of the bug triagers and the changeover that occurred in prioritization and severity decisions. In this figure, the textual information "As a result, I am reducing the priority of this ticket…" shows the correlation of bug triagers or their opinions with the changeover that occurred in both priority and severity decisions, whereas the severity decisions occurred in the same timestamps also illustrate its correlation with priority decision. Therefore, the rationale of the BP decisions that were taken in the past can be analyzed from textual information. This information is significant to the scrum team during their retrospective meetings. This also highlights the role of people as an essential factor in DM.

People Information: People have an integral role in DM. The bug report provides a digital platform where these people coordinate. Plenty of studies inform that bug reports record the actions of people shown in Fig 2. It illustrates opinions and decisions regarding triaging and BP tasks. Comments and history display the triaging and prioritization tasks and associated roles. Therefore, the actions of people involved in DM for triaging and BP tasks can be tracked from previous bug reports. This study focuses on learning retrospective data from bug reports of previous sprints. In order to better understand how different stakeholders collaborate, this study looks into the role of those engaged in DM for BP tasks by employing historical bug reports of different software corporations. Table 3 illustrates these roles.

Table 3: Displays the sub-categories of people's information

Reporter	A reporter is responsible for creating a bug report with quality information.
Bug Triager	This role can be performed by the team lead or manager, or it can be a dedicated role that conducts DM sessions where various decisions related to triaging and prioritization of bugs are made.
Assignee	The primary role of the assignee is to diagnose and resolve the reported bug. He provides the necessary information about the reported bug.
Customer	The Jira bug report shows the number of affected customers. This is a quantitative field that shows the increasing interest of customers in handling the bug.
Watcher	Watchers do not directly participate in any DM session. They only want to be kept informed about various decisions to be taken. This is

	a quantitative field that shows the increasing interest of different people in handling the bug.

The information contained in the bug reports reveals that people actively participate in the tasks associated with triaging and prioritization of the bug reports, including reporting bugs, participating in DM, voicing concerns, exchanging ideas, providing their opinions, and voting to facilitate DM. Fig 3 illustrates the categories of people who directly or indirectly participate in handling BP tasks. The opinions given by the bug triagers regarding the prioritizing of bugs, which have previously been covered in earlier sections, are shown in Fig 2.

To analyze and comprehend how correctly decision-makers including the diverse stakeholders, are participating in bug triaging and prioritization DM, the bug triagers can get relevant insight from different components of a bug report, e.g., they can access the categorical information of priority which shows that high-priority is assigned to a bug report, then he can access rationale of priority decision by examining the comments; then he can access the temporal information by tracing back the history of earlier priority decisions; further he can investigate the other correlated factors using the temporal data having similar timestamps. Thus, the repository of bug reports would be helpful for retrospective study.

Traceability Information: The bug reports as sources of information can be categorized as internal and external sources. These sources of information are inputs to bug triaging and prioritization tasks. Textual and categorical are contained in bug reports and therefore are internal sources of information. In contrast, traceability files are external sources of information that are accessible to the bug triagers through traceability links that connect them to other files comprised of attached bug reports, hyperlinks, wiki pages, voice files, images, video clips, doc files, pdf files, and xls files. In Jira bug reports, traceability files are attached to various sections of a bug report: the “Issue Links” section, description section, history, and comment section. Table 4 illustrates the sub-categories.

Table 4: Displays the sub-categories of people's information

Attached Bug Reports	Different types of bug reports are attached as external sources to the reported bug: bug reports attached as a blocker; bug reports attached as blocked; related bug reports refer to either duplicate; bugs of a similar nature that are also related to other modules; bug reports that are similar in some other contexts; dependent bug reports refer to bug reports that will be resolved once reported bug is resolved.
Images and video files	Bug reporters attach a number of images and video files with a bug report that assists the bug triagers in its reproducibility, understandability, and evidence. It contains various formats: PNG, GIF, or JPG.
Hyperlinks	Hyperlinks are provided in the bug report that point to the bug fixation policy, workflow of the reported bug, wiki pages, and other relevant information. Wiki pages include Confluence pages and Git pages.
Attached Documents	Various documents are attached in the form of Google Docs, Google Sheets, MS Office Docs, and pdf as information sources.

The traceability information provides useful resources to triage and prioritize bug reports. Decision makers collect information for DM tasks for triaging and prioritizing bug reports throughout the bug life cycle through traceability (Akbarinasaji et al., 2020; Almhana & Kessentini, 2021; Feng et al., 2016; Jahanshahi et al., 2022). In the selected bug report twelve files have been added as an external source that could help the bug triagers in handling DM tasks. Therefore, the availability of enough information is an essential factor for DM that can help the bug triagers in handling DM tasks for BP effectively. Therefore, sources of information are a useful factor for the retrospective study to examine the effectiveness of triaging and prioritization decisions.

Historical Information: Information in bug reports evolves during the lifecycle of the bug because the bug handling team keeps adding and modifying it at different timestamps. These changes can be tracked from the history section of bug reports. The format and content of the history section vary among software corporations. The history section of Apache bug reports illustrates the description of events performed (e.g., update of description, change of priority decision, a new traceability link added), information modified (e.g., existing priority was modified from low to high), timestamp (e.g., 8 days ago), and role involved is (e.g., Ahmed).

It is useful to assess the importance of bugs for determining bug priority (Kumari & Singh, 2020; Tian et al., 2015). Most of the study prescribes the prospective approach for DM. A study from past bug reports reveals multiple changeovers in BP decisions. In a retrospective approach, the historical information will be helpful for the bug triagers to diagnose the rationale behind the BP decisions taken in the previous sprints and analyze their changeovers. Hence, the retrospective information will be input for the bug triagers to learn lessons that can be used for the prospective DM approach. Hence, both the retrospective and prospective DM approach can assist them in addressing the decision-related challenges of BP.

Table 5: Displays the sub-categories of people's information

Events Performed	Events include an update of description, a change of priority decision, a new traceability link added, and so on.
Existing or Original Information (Change from)	Existing or Original Information (Change from): Existing priority was low, new version was assigned.
New Information (Change to)	New Information (Change to): Information was modified to high, new traceability link was added.
Timestamp	Priority 8 days ago, new version was assigned 2 days ago.
Role Involved	Ahmed changed the priority; Brian added the new traceability link and so on.

Jahanshahi et al., (2022) highlight the usefulness of evolutionary information for conducting triaging and prioritization tasks. The selected bug report was edited 20 times, whereas the bug priority decision was modified 6 times. The evolutionary information can be extracted and traced back using the components of history by connecting various other components of a bug report, for example, the prioritization decision was discussed in comments 7 days ago by the team lead, Adam as the owner of a bug report changed the existing priority from low to high 8 days ago. Table 5 displays the components of history.

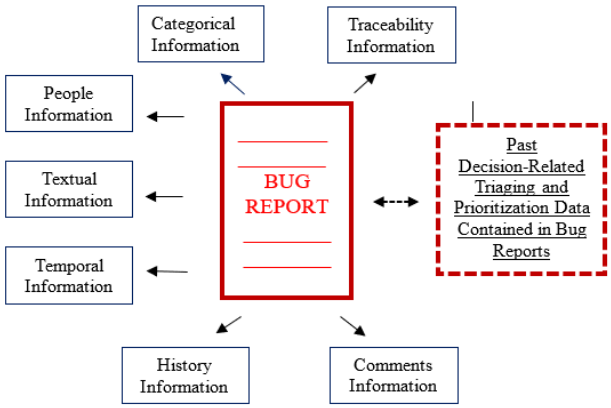

Figure 3: Illustrates the categorization of the contents of the bug reports into different components of decision knowledge based on their relationship with BP

Temporal Information: Temporal information is time-based information about various events that are performed and actions that are taken on different timestamps by various software team members who are involved in handling bug reports. These events include meetings sessions or events conducted for DM to triage and prioritize bug reports. Jahanshahi et al., (2022 and Tian et al., (2015) examined the relationship of temporal information with BP for the design of their research study.

In Fig 2, temporal information is used to observe the correlation between BP decisions and other factors, for instance, the figure shows the same time stamp for opinions given by the bug triagers in comments, the changeover of priority, and severity decisions given in history. The temporal information includes the following components

- the date of bug report creation.
- the date of the bug report was modified.
- the date of the bug report was resolved.
- the date of comments shared.
- and the date of operation performed, e.g. tasks performed, an event occurred and action taken.

Comments Information: It is observed from the comments section that the software team records different types of information for handling bugs. In Fig 2, textual contents appearing in the comments of the bug report highlight the opinions of the bug triagers about BP. These opinions are useful for the bug triagers to understand the rationale behind the priority decisions taken and diagnosing their changeovers during scrum retrospective (Almhana & Kessentini, 2021).

The comments section contains development data showing the collaborative tasks of team members which evolves till a bug report remains open for resolution. This section provides a common place for the bug triagers and other members of the bug-handling team to share suggestions for handling bugs. It includes different opinions shared by team members, DM tasks performed and meetings conducted, diagnostics, solutions for fixing code, decisions made, and relevant resources. This helps to track the bug report for triaging and prioritization tasks.

Many bug reports may take longer time to resolve due to many reasons such as pending triaging and prioritization tasks, and so on. Therefore, the comments section provides a traceable path for other team members to understand what is being done and what is being discussed so far for the resolution of the bug. Therefore, the rationale of various actions performed and reflected in a categorical section can be traced back from the comments sections. Comments comprise various components of decision knowledge which are illustrated in table 6.

Table 6: Displays the sub-categories of components of comments

Event Occurred	DM sessions or backlog sessions are conducted to take priority decisions, opinions of team members, and the rationale behind various actions performed.
Tasks Performed for DM	Tasks performed for DM include obtaining information, reproducing bugs, finding the workaround, changes in description, and adding new sources of information.
Sources of Information	Sources of information employed in DM: traceability links, diagnostic solutions, and workaround, risk of delay in fixation.
Constraints	Constraints are the complexity of bugs, consequences of fixation, and so on.

Validation

There is a growing interest in software engineering taxonomies, but they are rarely assessed. A taxonomy that conforms to the quality criteria can help researchers and practitioners communicate more effectively and develop a shared vocabulary and knowledge (Kaplan et al., 2022; Usman et al., 2017). Therefore, taxonomies must be examined and validated against quality criteria to verify their accuracy, completeness, and usability in real-world situations. This step structures the evaluation process into three steps: assessing the structure's suitability, applicability, and purpose (Kaplan et al., 2022). It refines the categories and sub-categories through iterative cycles, using inputs from reviewers, illustration approaches, and industry practitioners.

Validation Steps

The validation phase involves collaborating with subject experts and industry practitioners to refine categories. In this phase, the structure's suitability of taxonomy is assessed by examining the organization, hierarchy, and relationships among categories. The proposed taxonomy accurately depicts the underlying concepts and their relationship, ensuring that the knowledge components in bug reports related to BP are classified suitably under the broad categories and specific where no sub-categories are redundant or need to be split. For example, the priority, status, severity, and sub-categories of other classes are found in specific classes. As a result, during the evaluation process, the generality of the taxonomy is determined suitable.

In the validation phase, the taxonomy's appropriateness can be assessed by ensuring that it covers a wide range of components of knowledge relevant to BP. This step ensures that the identified broad categories and sub-categories are sufficiently sound and that the proposed taxonomy does not include any unnecessary classes. The sub-categories are further assessed by an independent evaluator; a few conflicts developed but were addressed. This step addresses the threats to internal validity. However, ongoing research and refining processes may reveal new BP-related components of knowledge in bug reports that should be grouped by adding more broad categories. Hence, the completeness of broad classes in the proposed taxonomy may be low. To improve the taxonomy, it is required to consider expanding the dataset with open-source bug reports from other large corporations as well as closed-source bug reports and new components of knowledge that need to be explored in future iterations.

Finally, to assess orthogonality, the proposed taxonomy is checked for overlap by analyzing components of knowledge relevant to BP to confirm that they fit cleanly into one category with no ambiguity. It is noticed that comments and diagnostic categories overlap because both contain textual data, however, the purpose of both categories is different while a few components have

the same scope that shows improved orthogonality. This step addresses the threats to construct validity.

The validation takes place in two phases. This phase assesses the suitability of the structure followed by assessing the applicability by an illustrative example and a focus group to explore its purpose.

Illustrative Approach

Taxonomies play a crucial role in structuring knowledge and classifying processes. Illustration is the most used method to demonstrate the applicability of software engineering taxonomies. Its objective is to make taxonomies more accessible and clearer by providing actual examples. It helps researchers and decision-makers understand the taxonomy's practical implications so that they can improve its usability and adoptions and how it can be applied in various settings (Usman et al., 2017).

In the first phase, the proposed taxonomy is applied to a small dataset that also serves as a pilot test before being expanded upon in subsequent research studies. In order to illustrate how the proposed taxonomy can be used, this phase applies the taxonomy to a selected bug report as a real-world case, demonstrating the utility of the proposed taxonomy for BP, its ease of use, and its applicability to the industry practitioners as well as to evaluate its significance for Scrum retrospective meetings. Therefore, an illustrative approach is employed in this phase.

It is observed in our previous study (Raja et al., 2023) that the structure of bug reports varies in different software corporations and all bug reports do not contain rich information correlated with BP. An illustrative example makes it easier to understand concepts or thoughts. Therefore, to illustrate the concepts clearly, it was necessary to choose a bug report as an example for qualitative data collection that contains rich BP-related features. Consequently, a number of bug reports from various software corporations are accessed and filtered according to the rich BP-related data they possess.

Thus, a Jira bug report[ii] possessing a rich history of bug triaging and prioritization tasks is selected based on prescribed criteria, for example, more than one priority decisions were taken, bug reports contain rich comments and history, discuss the rationale behind the priority decisions taken, contains other priority-related features, and so on. The Jira bug report structures the information in various sections: detail, people, date, description, issue links, forms, workflow, activity, and export. The activity further contains sub-sections: comments, history, work log, and activity section.

To extract the different types of data from the selected bug reports, the proposed taxonomy is used as a guideline in which contents of the selected bug report are analyzed to see what BP-related tasks are relevant, and therefore, BP-related qualitative data is gleaned from the selected bug report.

Real scenario from the illustrated example: Various components of the selected bug report are examined, and it is observed that the description of the bug also keeps on changing, and Theodora Boudale who created the bug report changed the existing priority from low to medium on 16/10/2018. It is analyzed from the comments, history, and temporal data, that Sean McLucas, who is head of an engineering team had led the backlog session for prioritization DM and decided to change the priority decision. The temporal information shows that this discussion was initiated on 19/08/2020, where historical data shows its correlation with various other factors. It is analyzed from the comments that team members kept on sharing their opinions about prioritization decisions. Thus, bug reports provide various types of retrospective data learned from the previous sprints. Table 7 and Fig 4 display the illustrative approach. In Fig 4, labels are given to the components of bug reports that display how various components of bug reports relate to priority decisions in the selected bug report providing insights to the bug triagers to use the retrospective knowledge for BP in the present sprints.

Various insights are analyzed from the selected bug report. An illustrative example shows that bug report contains actions of people that tell us about triaging and prioritization (a) tasks initiated for bug reports (b) events are performed e.g., the meeting initiated for DM where the bug triagers could have shared his opinion (c) actions are taken, e.g., priority change from medium and high priority was assigned (d) essential information are involved, e.g., 66 customers are affected by bug reported (e) decisions being taken (f) severity decision shown as a correlated factor to BP. The collected data can also be used to trace the rationale behind the bug triaging and priority decision being taken, e.g., Fig 2 shows the correlation of opinions shared by the assignee or the bug triagers, and a priority decision being taken in similar timestamps.

It can also be used to trace the number of decisions being taken for prioritization and the reasons behind it, e.g., 7 times priority decisions are taken which means decisions are being changed 6 times and reasons can be investigated from 5 threads of comments that are about prioritization decisions whereas other 22 threads of comments are about other triaging and bug fixation related decisions. Thus, the collected data in the bug reports shows its significance for DM for triaging and prioritization of bug reports and provides a valuable source of information for conducting a retrospective study.

Insights Interpreted qualitatively from Categorical Information:

- Priority Decision: The "Low" priority was assigned to the bug report when it was created because at this stage information was not complete. Hence, the priority decision was based on partially triaged information. Later, priority was changed six times which is displayed in the categorical field of priority which can be examined from the temporal and historical data while the rationale behind these decisions can be analyzed from comments. The present priority of the selected bug report was "High" which was observed on 12/01/2022.

- Severity Decision: It was observed from the comments and history of the selected bug report as well as examined from the literature review that the severity decision was correlated and interdependent with the priority decision, see Fig 2.

- Workflow: When the medium priority was assigned, the status of the bug was changed from "gathering impact" to "short-term backlog" within the prioritization phase. This is an interdependent action that informs the workflow of BP tasks.

Table 7: An illustration example of the proposed taxonomy

Dimensions	Decision-related contents of the selected bug report
Temporal	The selected bug report was created on 16/10/2019, however, it was modified at various timestamps.
	In the selected bug report, the description was added when the bug report was created and then updated on Oct 16, 2019, and further updated at different timestamps implying that information was not stable because it was being evolved.
	The timestamps inform that more than one developer is assigned to the bug report.
History	It is noticed from history that priority was changed several times in the selected bug report which shows that the decisions taken for prioritization were not stable on various occasions.

	It is observed from the history section that low priority was assigned to the bug report when it was created on 21/04/2019. Then a changeover occurred in BP decisions on Oct 16, 2019, and priority was changed from low to medium. Further, it was changed six times in the selected bug report. The current priority of the selected bug report is finally assigned as low.
Comments	The thread of comments between the assignment of priority decisions shows that people provide various opinions about diagnostic and priority decisions at various timestamps. On the other hand, high volume threads of comments are observed and plenty of opinions of team members about diagnostic information and BP are shared between assignments of medium and highest priority.
	When the supplementary resolution of the bug was performed then an increasing number of comments were not observed, and bug priority was changed to low for stable resolution which was postponed.
Categorical Information	Priority: The present priority of the selected bug report is "Low", however, the categorical field of priority displayed various priorities at different timestamps which can observed from history.
	Severity: The present severity of the selected bug report is "Major", however, the categorical field of priority displayed various priority labels at different timestamps which can observed from history.
	Status: When the bug report was created, no status was assigned to it, however on 29/10/2019, the "need triage" status was assigned to the bug report. It is noticed from the history section that the bug is moving back and forth between different phases and to other statuses.
Traceability Information	It is examined from the selected bug report that "13 files" were attached as external sources of information whereas the bug ID is hyperlink information to access the reported bug report from other bug reports or the attached reports.

	In the "Detail" section of the selected bug report, a hyperlink to the bug fixation policy of Atlassian Corporation is provided referring to an external source of information that describes the workflow of BP and fixation, the correlated factors, and the policy of how bugs are triaged, prioritized, and resolved within the workflow.
People Information	Comments show that Theodora Boudale is assigned as a reporter who is involved in adding and updating the contents of the bug reports. Christopher Csomme was allocated as an assignee after the medium priority was assigned. It is a correlated and interdependent factor in the prioritization decision.
	Sean McLucas, head of the engineering team working for Bitbucket Cloud, led the backlog refinement meeting for BP as role as decision-maker.
	The increasing trend in a number of watchers and customers is noticed in the selected bug reports in different prioritization situations.
Diagnostic Information	It is examined from the description that the bug was not reproducible at that instant of time, therefore, steps to reproduce the bug were not described. Hence, more information was required to reproduce the bug.
	The description of the selected bug report shows that the diagnostic was not completed, and a workaround was not suggested at this stage.

Insights Interpreted qualitatively from Comments and Historical Information:

- History Data: History is a valuable source of information for exploring factors correlated with priority decisions, severity, and assignees; see Figure 2. History provides retrospective data to analyze insights for future actions. It is observed from the selected bug report that bug report priority was changed six times, whereas severity was changed two times. On the other hand, the bug report was edited several times. In this context, the bug reports of previous sprints containing evolutionary data need to be observed during retrospective meetings.

- Comments: Comments provided several types of insights that can help the bug triagers in addressing the decision-related form of BP. The increasing number of comments from the development team and affected customers demonstrate people's interest in the bug. Various actions are taken in the bug reports which are shown in the categorical section while the rationale behind them can be diagnosed from the comments. It is noticed from the comments that the resolution of the bug had a certain complexity due to which the triaging team was not assigning this bug a higher priority, it could be either the limitation of the expert team members to resolve the bug or the impact of resolution of the bug on the entire branch.

Insights from Diagnostic Information:

- Workaround: Workaround was a key factor for BP which is also mentioned in the bug fix policy of Atlassian. The description of the selected bug report shows that the diagnostic was not completed, and a workaround was not suggested at this stage. Therefore, still, information is required to reproduce the bug.

- Summary: The summary was logged after the bug was reported, however, the bug was not reproducible, therefore, steps to reproduce the bug were not described.

Insights Interpreted qualitatively from People Information:

- Reporters: Reporters are responsible for providing updated information to the bug report as well as maintaining the quality of information. Therefore, they are information experts, and the decision-makers can obtain necessary information from them directly for handling BP.

- Watchers and Affected Customers: The number of watchers and affected customers are factors that show interest in people for the bug. They were not prominent until medium priority was not assigned. However, the number of affected customers increased after the assignment of medium priority instead of the high priority which moved the bug towards negative consequences.

- Team Lead/ Decision Maker: Sean McLucas, head of the engineering team, Bitbucket Cloud, led the backlog refinement meeting for BP, on Aug 19, 2020, and performed the role of decision-maker for BP.

Insights Interpreted qualitatively from Traceability Information:

- The bug triagers for prioritizing bug reports during the prospective DM approach can visit the policy of BP and fixation using the traceable link so that everyone should know about the policy of handling BP and thus remains on the same page. For the retrospective approach, the team member can visit the newly reported or reopened bug or previous sprint and access the information through traceable links provided in the bug report.

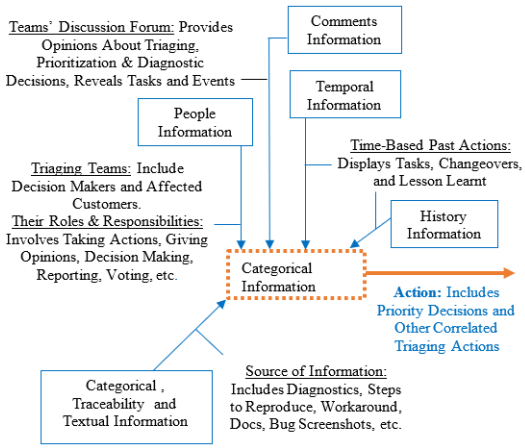

Figure 4: Illustrates the components of bug reports and their relationship for providing insights about handling BP

Insights Interpreted qualitatively from Temporal Information

Temporal data is useful for looking back at time-based historical information associated with triaging and priority decisions made, for instance, it was analyzed that initially developer was assigned on Oct 29, 2019, and then it was reassigned when the priority of the bug was changed from low to high.

Focus Group

In the second phase focus group method is employed in which, the illustrated example is used to demonstrate the significance of the proposed taxonomy to the industry practitioners and to seek their feedback. In this phase, participants are selected based on their interaction with bug reports, experiences with BP tasks, and familiarity with Scrum retrospectives. The participants include a product owner, quality control engineer, solution architect, and project lead. The role of the moderator is performed by the principal researcher. It is ensured in this phase that the idea of the research is well communicated to experts so that they can understand it and propose solutions before participating in the discussion. The material needed for conducting the focus group includes a set of pre-defined questions, a presentation, Table 1, and an illustrative example which is given in Table 7. The obtained feedback helped in refining the research and ensuring its applicability in real-world scenarios.

The interactive discussions facilitated by the focus group allowed for a comprehensive evaluation of the proposed taxonomy and its practical implications for Scrum retrospectives. Therefore, experts and professionals provided critical feedback on the relevance and applicability of the research findings. The participants' insights and suggestions helped improve the taxonomy categories and suggest their applicability while ensuring that the findings are significant in the real world in bug triaging and prioritizing practices. The participants acknowledged the challenges of archiving the development data for the Scrum retrospective. This collaborative validation process not only strengthened the credibility of the research but also highlighted the practical utility of the retrospective DM approach in addressing decision-related challenges faced by bug triagers.

The validation process with a focus group revealed a few good insights for using retrospective data in BP in the future. The focus group positively reacted to a proposed taxonomy derived from past bug reports and provided several key pieces of feedback. They found the taxonomy relevant, and useful, streamlining the prioritization process and offering insights that are often overlooked in traditional bug triaging and prioritizing tasks.

However, the focus group also highlighted limitations and provided constructive feedback for improvement: suggestions include adding additional sub-categories, adding detailed descriptions for each sub-category, incorporating more bug reports for illustrative examples and real-world scenarios, creating an interactive visualization of the taxonomical model, and integrating the taxonomy with existing AI tools for adoption. Therefore, a visualized model illustrating the relationship of different components of knowledge in bug reports for providing insight into BP tasks is added, see Fig 4.

The focus group participants suggested that taxonomy could be integrated with other Agile practices, particularly with other ceremonies of SCRUM such as sprint planning and review meetings, to identify recurring issues and improve processes. The group emphasized the flexibility of the proposed taxonomy so that it could be customizable for specific project sizes and nature i.e. medium-sized projects, and the nature of projects such as close-source projects.

They further highlighted the limitations of adopting the taxonomy in their organization and three reasons are highlighted: the absence of an AI tool to put taxonomy into practice, lack of proper guidance in taxonomy adoption, and the absence of rich bug reports due to a closed-source environment. The taxonomy was developed from open-source bug reports whereas the organization selected for the focus group is a proprietor organization in which bug reports are not created with rich development data. This limitation was already known and couldn't be addressed due to some constraints. However, it will be considered for future work. Hence, feedback is crucial in refining the research.

The focus group also emphasized the importance of continuous refinement, suggesting periodic reviews and updates based on new data and evolving bug-triaging practices. The research was refined and enhanced to address some suggestions obtained from the focus group. In this regard, the taxonomy was updated by adding more details in sub-categories while accommodating additional components of knowledge.

Hence, the feedback is helpful for the study to refine the findings for practical and real-world BP tasks presently and in the future. In the future, more bug reports from Jira and other large organizations should be utilized to demonstrate the applicability and significance of the proposed taxonomy. This will address threats to external validity, by including important decision knowledge components that may not be covered in this study.

Threats to Validity

Threats to the research findings' validity are identified and dealt with, with a particular emphasis on the taxonomy's building process. The validation section discusses the mitigation of validity threats in detail.

Interval Validity Threats: The researchers' biases may affect the taxonomy-building process. In order to address the threats of internal validity, the taxonomy development process already follows the quality criteria outlined by Kaplan et al., (2022); Usman et al., (2017) and it is made sure that multiple researchers participate in the process, and that the taxonomy undergoes independent evaluation. Therefore, the proposed taxonomy is sound, but ongoing research may reveal new knowledge components, requiring expansion with open-source and closed-source bug reports.

The selected population of bug reports used to develop the taxonomy belongs to a large open-source corporation. However, all open-source bug reports do not possess rich content that cannot be useful in our context, which can be a threat to internal validity. Good-quality bug reports are useful digital artifacts and are the essence of effective bug prioritization. Therefore, creating bug reports that comply with quality attributes is essential to mitigate the threat of internal validity.

External Validity Threats: The applicability and significance of the study findings are ensured by employing one bug report having rich contents for an illustrative approach and one Scrum team is engaged in a focus group, and the taxonomy is refined by the critical feedback acquired from the experts, so mitigating the threats to external validity in a limited context. However, since this is ongoing research, some suggestions will be considered for further studies. On the other hand, more bug reports should be utilized for illustrative examples and results should be applied to multiple Scrum teams in future research. Real data should therefore be gathered from multiple Scrum teams and retrospective meetings, including those with various degrees of experience and from various industries.

The taxonomy was primarily developed with bug reports from Atlassian; but, in order to address generalizability, it also utilizes bug reports from other open-source software companies. Nevertheless, a significant number of bug reports from many software companies are not considered in the construction of the taxonomy. This might limit how widely the study's findings can be applied. Therefore, to overcome threats to external validity, future research should draw from a wide range of sources. This will demonstrate the importance and broad applicability of the proposed taxonomy.

Threats to Construct Validity: To address the threats to construct validity, the taxonomy development process adheres to the orthogonality with the criteria cited in (Kaplan et al., 2022). Further, the structure's suitability of the proposed taxonomy is evaluated by looking at its organization, hierarchy, and orthogonality; as a result, the proposed taxonomy is refined. For example, the textual category is initially identified as a broad category rather than a diagnostic one, but during evaluation, it is noticed that comments and history categories also contain textual

components of information; therefore, the textual label as a broad category is replaced by a diagnostic label.

DISCUSSION

The research question is addressed and findings are validated

Interpretation of Findings: The proposed taxonomy classifies significant decision-related triaging and prioritizing that can be utilized for Scrum retrospectives which are further found suitable based on the evaluations conducted. The feedback highlights the taxonomy's clarity, comprehensiveness, and intuitiveness and describes its utility and relevance. Some feedback is included in existing work while some advice will be incorporated in the future.

Comparison with Existing Work: The findings of the research align with previous research that refers to bug reports as "components of decision knowledge" that can be used for handling various aspects of bugs (Gökçeoğlu & Sözer, 2021; Hesse et al., 2016; Jahanshahi et al., 2022; Li et al., 2024). But, unlike existing taxonomies, the proposed taxonomy classifies the components of decision knowledge related to BP. Thus, the proposed taxonomy can complement existing guidelines used for handling bugs, potentially leading to more effectiveness in providing valuable insight to the bug triagers for prioritizing bugs and introducing the use of historical bug reports in retrospective meetings for this purpose.

Implications for Practice: The proposed taxonomy serves as a valuable mechanism for industry practitioners as it offers guidelines for identifying significant components of decision knowledge related to BP from bug reports created in previous sprints and how to use the retrospective meetings for BP. To enable software organizations to adopt this taxonomy in their bug-handling practices, an AI tool using the taxonomy should be developed in the future.

CONCLUSION AND FUTURE DIRECTION

Bug triagers require deeper insights to make appropriate BP decisions. Decision problems occur in the BP process due to many factors among which availability of information is a prominent factor. Therefore, bug triagers have to face decision-related challenges in handling BP tasks in case enough information is not available. On the other hand, bug reports contain components of decision knowledge related to BP tasks.

Summary of contributions: This study emphasizes the significance of Scrum retrospectives in providing insights using bug reports created in previous sprints for handling BP tasks. Therefore, in this study, components of decision knowledge related to BP contained in historical bug reports are explored, and a taxonomy is developed that effectively structures knowledge components that can provide guidelines to the bug triagers in handling BP tasks in retrospective meetings. Thus, this paper will be a motivation for conducting a retrospective study of BP using the past data.

The research findings are validated in two phases. In the first phase, the suitability of the structure of the proposed taxonomy and its applicability is assessed using an illustrative example, and a focus group methodology is employed to evaluate its purpose. The taxonomy is refined after the feedback received from the validation phases. However, some suggestions are considered for future research.

Limitations and Future Directions: There are a few limitations in this study. For example, the findings of the study cannot be customizable to closed-source projects, taxonomy cannot be used in practice due to the absence of an AI tool that limits the availability of real data. Furthermore, the study lacks proper guidance for adoption. One bug report is used for illustration which is not sufficient and is a potential threat in addressing external validity.

Future research should consider a wider range of sources (i.e. include bug reports from closed-source and other open-source corporations and multiple retrospective meetings) to ensure the

generalizability of the findings and demonstration of the proposed taxonomy's applicability. To get feedback on real data, the suggested taxonomy should be applied in an industrial setting. Retrospective meetings face few challenges due to limited past data, therefore suggested taxonomy should be mapped with these challenges. In the future, focus groups must be expanded to include participants from various organizations.

REFERENCES

Abou Khalil, Z., Constantinou, E., Mens, T., Duchien, L., & Quinton, C. (2019, September). A Longitudinal Analysis of Bug Handling Across Eclipse Releases. *2019 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. https://doi.org/10.1109/ICSME.2019.00010

Akbarinasaji, S., Kavaklioglu, C., Başar, A., & Neal, A. (2020). Partially observable Markov decision process to generate policies in software defect management. *Journal of Systems and Software*, *163*. https://doi.org/10.1016/j.jss.2020.110518

Almhana, R., Ferreira, T., Kessentini, M., & Sharma, T. (2020a). Understanding and Characterizing Changes in Bugs Priority: The Practitioners' Perceptive. *Proceedings - 20th IEEE International Working Conference on Source Code Analysis and Manipulation, SCAM 2020*, 87–97. https://doi.org/10.1109/SCAM51674.2020.00015

Almhana, R., Ferreira, T., Kessentini, M., & Sharma, T. (2020b, September). Understanding and Characterizing Changes in Bugs Priority: The Practitioners' Perceptive. *2020 IEEE 20th International Working Conference on Source Code Analysis and Manipulation (SCAM)*. https://doi.org/10.1109/SCAM51674.2020.00015

Almhana, R., & Kessentini, M. (2021). Considering dependencies between bug reports to improve bug's triage. *Automated Software Engineering*, *28*(1). https://doi.org/10.1007/s10515-020-00279-2

Anvik, J. (2011). *Reducing the Effort of Bug Report Triage : Recommenders for Development-Oriented Decisions*. *20*(3). https://doi.org/10.1145/2000791.2000794

Apache Software Foundation. (2023). *Apache Software Foundation, "Bug Priority Guidelines and Release Schedule," Release Schedule - NetBeans - Apache Software Foundation. [Online]. Available: https://cwiki.apache.org/confluence/display/NETBEANS/Release+Schedule. [Accessed: 21-Feb-2023].*

Atlassian. (2023). *"Issue Tracker System, and Application Lifecycle Development and Collaboration Tools for Software, IT and Business Teams," Atlassian. [Online]. Available: https://www.atlassian.com/. [Accessed: 01-Feb-2023].*

Atlassian Bug Fix Policy. (2023). *"Atlassian Cloud Bug Fix Policy," Atlassian Documentation. [Online]. Available: https://confluence.atlassian.com/support/atlassian-cloud-bug-fixing-policy-206865884.html. [Accessed: 01-Feb-2023].*

Atlassian Bug Reports. (2023). *"Browse projects - Jira Bug Tracking System for Closed-Source Projects of Atlassian". [Online]. Available: https://jira.atlassian.com/projects. [Accessed: 01-Feb-2023].*

Atlassian Priority Policy. (2023). *"What are priority levels in Jira Service Management?: Jira Service Management Cloud," Atlassian Support, 01-Feb-2023. [Online]. Available: https://support.atlassian.com/jira-service-management-cloud/docs/what-are-priority-levels-in-jira-service-management/. [Accessed: 03-Feb-2023].*

Atlassian Workflow. (2021). *"Work with issue workflows," Atlassian Support, 19-Nov-2021. Atlassian. [Online]. Available: https://support.atlassian.com/jira-cloud-administration/docs/work-with-issue-workflows/. [Accessed: 21-Feb-2023].*

Bani-Salameh, H., Sallam, M., & Al shboul, B. (2021). A Deep-Learning-Based Bug Priority Prediction Using RNN-LSTM Neural. *E-Informatica Software Engineering Journal*, *15*(1). https://doi.org/10.37190/e-Inf210102

Bettenburg, N., & Just, S. (2008). *What Makes a Good Bug Report ?*

Citroen, C. L. (2011). The role of information in strategic decision-making. *International Journal of Information Management*, *31*(6). https://doi.org/10.1016/j.ijinfomgt.2011.02.005

Cohen J, F. R. H. W. (2013). *A Defect Prioritization Method Based on the Risk Priority Number. White Paper. Carnegie Mellon Universtiy, Pittsburgh, SEI [Online]. Available: https://insights.sei.cmu.edu/documents/367/2013_019_001_70276.pdf [Accessed: 02-June-2023].*

Cunha, J. A. O. G., Moura, H. P., & Vasconcellos, F. J. S. (2016). Decision-making in Software Project Management: A Systematic Literature Review. *Procedia Computer Science*, *100*. https://doi.org/10.1016/j.procs.2016.09.255

Da Cunha, J. A. O. G., Da Silva, F. Q. B., De Moura, H. P., & Vasconcellos, F. J. S. (2016). Decision-making in software project management: A qualitative case study of a private organization. *Proceedings - 9th International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE 2016*. https://doi.org/10.1145/2897586.2897598

Dean, J. W., & Sharfman, M. P. (1996). Does decision process matter? A study of strategic decision-making effectiveness. *Academy of Management Journal*, *39*(2). https://doi.org/10.5465/256784

Drury-Grogan, M. L., Conboy, K., & Acton, T. (2017). The Journal of Systems and Software Examining decision characteristics & challenges for agile software development. *The Journal of Systems & Software*, *131*, 248–265. https://doi.org/10.1016/j.jss.2017.06.003

Eclipse Foundation. (2023). *"Browse projects - Bugzilla Bug Tracking System for Eclipse Open-Source Projects, [Online]. Available: https://bugs.eclipse.org/bugs/query.cgi. [Accessed: 21-Feb-2023].*

Feng, Y., Jones, J. A., Chen, Z., & Fang, C. (2016). Multi-objective test report prioritization using image understanding. *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering*, 202–213. https://doi.org/10.1145/2970276.2970367

Ghozali, R. P., Saputra, H., Nuriawan, M. A., Suharjito, Utama, D. N., & Nugroho, A. (2019). Systematic Literature Review on Decision-Making of Requirement Engineering from Agile Software Development. *Procedia Computer Science*, *157*. https://doi.org/10.1016/j.procs.2019.08.167

Gökçeoğlu, M., & Sözer, H. (2021). Automated defect prioritization based on defects resolved at various project periods. *Journal of Systems and Software*, *179*. https://doi.org/10.1016/j.jss.2021.110993

Hesse, T. M., Lerche, V., Seiler, M., Knoess, K., & Paech, B. (2016). Documented decision-making strategies and decision knowledge in open source projects: An empirical study on Firefox issue reports. *Information and Software Technology*, *79*, 36–51. https://doi.org/10.1016/j.infsof.2016.06.003

Höfer, A., & Tichy, W. F. (2007). Status of empirical research in software engineering. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *4336 LNCS*. https://doi.org/10.1007/978-3-540-71301-2_3

Hu, H., Zhang, H., Xuan, J., Sun, W., & Bug, E. (2014). *Effective Bug Triage based on Historical Bug-Fix Information To cite this version : Effective Bug Triage based on Historical Bug-Fix Information*.

Jahanshahi, H., Cevik, M., Navas-Sú, J., Başar, A., & González-Torres, A. (2022). Wayback Machine: A tool to capture the evolutionary behavior of the bug reports and their triage process in open-source software systems. *Journal of Systems and Software*, *189*. https://doi.org/10.1016/j.jss.2022.111308

Kanwal, J. (2010). *Managing Open Bug Repositories through Bug Report Prioritization Using SVMs*. www.mozilla.com

Kaplan, A., Kühn, T., Hahner, S., Benkler, N., Keim, J., Fuchß, D., Corallo, S., & Heinrich, R. (2022). Introducing an Evaluation Method for Taxonomies. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3530019.3535305

Kaushik, N., Amoui, M., Tahvildari, L., Liu, W., & Li, S. (2013, March). Defect Prioritization in the Software Industry: Challenges and Opportunities. *2013 IEEE Sixth International Conference on Software Testing, Verification and Validation*. https://doi.org/10.1109/ICST.2013.40

Keung, J. (2017). *An Empirical Analysis of Reopened Bugs Based on Open Source Projects*. *October*. https://doi.org/10.1145/2915970.2915986

Klein, G. (2015). A naturalistic decision making perspective on studying intuitive decision making. *Journal of Applied Research in Memory and Cognition*, *4*(3). https://doi.org/10.1016/j.jarmac.2015.07.001

Ko, A. J., & Chilana, P. K. (2011). Design, discussion, and dissent in open bug reports. *Proceedings of the 2011 IConference*, 106–113. https://doi.org/10.1145/1940761.1940776

Kontio, J., Lehtola, L., & Bragge, J. (2004). Using the focus group method in software engineering: Obtaining practitioner and user experiences. *Proceedings - 2004 International Symposium on Empirical Software Engineering, ISESE 2004*. https://doi.org/10.1109/ISESE.2004.1334914

Kumari, M., & Singh, V. B. (2020). *An Improved Classifier Based on Entropy and Deep Learning for Bug Priority Prediction*. https://doi.org/10.1007/978-3-030-16657-1_53

Laiq, M., Ali, N. bin, Börstler, J., & Engström, E. (2023). A data-driven approach for understanding invalid bug reports: An industrial case study. *Information and Software Technology*, *164*, 107305. https://doi.org/10.1016/j.infsof.2023.107305

Li, Z., Cai, G., Yu, Q., Liang, P., Mo, R., & Liu, H. (2024). Bug priority change: An empirical study on Apache projects. *Journal of Systems and Software*, *212*. https://doi.org/10.1016/j.jss.2024.112019

Loeffler M. (2017). *Improving Agile Retrospectives: Helping Teams Become More Efficient, 1st edition* (M. Loeffler, Ed.; 1st Edition). Addison-Wesley.

Marler, R. T., & Arora, J. S. (2004). Survey of multi-objective optimization methods for engineering. In *Structural and Multidisciplinary Optimization* (Vol. 26, Issue 6). https://doi.org/10.1007/s00158-003-0368-6

Matthies, C. (2019). Feedback in scrum: Data-informed retrospectives. *Proceedings - 2019 IEEE/ACM 41st International Conference on Software Engineering: Companion, ICSE-Companion 2019*. https://doi.org/10.1109/ICSE-Companion.2019.00081

Matthies, C. (2020). Playing with your project data in scrum retrospectives. *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Companion Proceedings*, 113–115. https://doi.org/10.1145/3377812.3382164

Matthies, C., Dobrigkeit, F., & Hesse, G. (2020). Mining for Process Improvements. *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*, 189–190. https://doi.org/10.1145/3387940.3392168

Mendes, F., Mendes, E., Salleh, N., & Oivo, M. (2021). Insights on the relationship between decision-making style and personality in software engineering. *Information and Software Technology*, *136*. https://doi.org/10.1016/j.infsof.2021.106586

Mozilla Priority and Bug Policy. (2023). *"Priority definitions, Bug Handling Policy," Firefox Source Docs documentation. [Online]. Available: https://firefox-source-docs.mozilla.org/bug-mgmt/guides/priority.html. [Accessed: 01-Feb-2023].*

Mozilla Triage and Priority Guide. (2023). *"Triage and Priority Guide," Docs/bug-MGMT - mozsearch. [Online]. Available: https://searchfox.org/mozilla-central/source/docs/bug-mgmt/. [Accessed: 01-Feb-2023].*

Mozilla Triage for bugzilla. (2023). *"Triage for bugzilla," Triage for Bugzilla - Firefox Source Docs documentation. [Online]. Available: https://firefox-source-docs.mozilla.org/bug-mgmt/policies/triage-bugzilla.html. [Accessed: 01-Feb-2023].*

Noei, E., Zhang, F., Wang, S., & Zou, Y. (2019). Towards prioritizing user-related issue reports of mobile applications. *Empirical Software Engineering*, *24*(4), 1964–1996. https://doi.org/10.1007/s10664-019-09684-y

Noyori, Y., Washizaki, H., Fukazawa, Y., Ooshima, K., Kanuka, H., Nojiri, S., & Tsuchiya, R. (2019). What are Good Discussions Within Bug Report Comments for Shortening Bug Fixing Time? *2019 IEEE 19th International Conference on Software Quality, Reliability and Security (QRS)*, 280–287. https://doi.org/10.1109/QRS.2019.00044

PMI. (2015). *"Capturing the Value of Project Management Through Decision Making". Pulse of Profession. Report. 2015. [Online]. Available: https://www.pmi.org/learning/thought-leadership/pulse/capture-value-decision-making/ [Accessed: 01-Jun-2023].*

PMI. (2023). *A guide to the Project Management Body of Knowledge (PMBOK guide) (6th ed.). Project Management Institute*.

Politowski, C., Fontoura, L. M., Petrillo, F., & Guéhéneuc, Y. G. (2018). Learning from the past: A process recommendation system for video game projects using postmortems experiences. *Information and Software Technology*, *100*. https://doi.org/10.1016/j.infsof.2018.04.003

Raja, S. A., Aziz, M. S. A. , & Shah, A. (2023). Modeling the Workflow of Bug Prioritization Tasks Descriptively Using the Past Events. *International Journal on Perceptive and Cognitive Computing (IJPCC)*, *9*(2), 14–24.

Ralph, P. (2019). Toward Methodological Guidelines for Process Theories and Taxonomies in Software Engineering. *IEEE Transactions on Software Engineering*, *45*(7). https://doi.org/10.1109/TSE.2018.2796554

Red Hat. (2023). *"Browse projects - Jira Bug Tracking System for Red Hat Open-Source Project. [Online]. Available: https://issues.redhat.com/projects/. [Accessed: 01-Feb-2023].*

Saha, R. K., Khurshid, S., & Perry, D. E. (2015). Understanding the triaging and fixing processes of long lived bugs. *Information and Software Technology*, *65*. https://doi.org/10.1016/j.infsof.2015.03.002

Scrum Alliance. (2018). *The State of Scrum Report 2017-2018*. Survey. https://www.scrumalliance.org/

Scrum Alliance. (2023, September 15). *Scrum - an Art of Decision Making*. https://www.scrum.org/resources/blog/scrum-art-decision-making

Svensson, R. B., Feldt, R., & Torkar, R. (2019). The unfulfilled potential of data-driven decision making in agile software development. *Lecture Notes in Business Information Processing*, *355*. https://doi.org/10.1007/978-3-030-19034-7_5

The Standish Group. (2023). *The Standish Group. [Online]. Available: https://www.standishgroup.com/. [Accessed: 02-Feb-2023].*

Tian, Y., Lo, D., Xia, X., & Sun, C. (2015). Automated prediction of bug report priority using multi-factor analysis. *Empirical Software Engineering*, *20*(5), 1354–1383. https://doi.org/10.1007/s10664-014-9331-y

Tsoukias, A., Keeney, R. L., & Raiffa, H. (1994). Decisions with Multiple Objectives: Preferences and Value Tradeoffs. *The Journal of the Operational Research Society*, *45*(9). https://doi.org/10.2307/2584151

Uddin, J., Ghazali, R., Deris, M. M., Naseem, R., & Shah, H. (2017). A survey on bug prioritization. *Artificial Intelligence Review*, *47*(2), 145–180. https://doi.org/10.1007/s10462-016-9478-6

Umer, Q., Liu, H., & Illahi, I. (2020). CNN-Based Automatic Prioritization of Bug Reports. *IEEE Transactions on Reliability*, *69*(4), 1341–1354. https://doi.org/10.1109/TR.2019.2959624

Usman, M., Britto, R., Börstler, J., & Mendes, E. (2017). Taxonomies in software engineering: A Systematic mapping study and a revised taxonomy development method. In *Information and Software Technology* (Vol. 85). https://doi.org/10.1016/j.infsof.2017.01.006

Verwijs, C., & Russo, D. (2023). A Theory of Scrum Team Effectiveness. *ACM Transactions on Software Engineering and Methodology*, *32*(3). https://doi.org/10.1145/3571849

White, D. J. (2018). *Decision Theory*. Routledge. https://doi.org/10.4324/9780203793695

Wohlin, C., & Aurum, A. (2015). Towards a decision-making structure for selecting a research design in empirical software engineering. *Empirical Software Engineering*, *20*(6). https://doi.org/10.1007/s10664-014-9319-7

Xie, J., Zhou, M., & Mockus, A. (2013, October). Impact of Triage: A Study of Mozilla and Gnome. *2013 ACM / IEEE International Symposium on Empirical Software Engineering and Measurement*. https://doi.org/10.1109/ESEM.2013.62

Xu, Z., He, T., Zhang, W., Wang, Y., Liu, J., & Chen, Z. (2016). Exploring the influence of time factor in bug report prioritization. *Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE*, *2016-January*, 243–248. https://doi.org/10.18293/SEKE2016-162

Zhang, J., Wang, X. Y., Hao, D., Xie, B., Zhang, L., & Mei, H. (2015a). A survey on bug-report analysis. *Science China Information Sciences*, *58*(2). https://doi.org/10.1007/s11432-014-5241-2

Zhang, J., Wang, X. Y., Hao, D., Xie, B., Zhang, L., & Mei, H. (2015b). A survey on bug-report analysis. *Science China Information Sciences*, *58*(2), 2–3. https://doi.org/10.1007/s11432-014-5241-2

ENDNOTES

70 + bugs were qualitatively analyzed using repositories of past bug reports. Some sample bugs are cited here.

[i] Datasets of Bug Reports Belonging to Software Corporations for Development of Taxonomy

JIRA issue tracker. https://jira.atlassian.com/browse/JRACLOUD-72947

JRACLOUD-72947, JSWCLOUD-20625, JRACLOUD-72655, JSDCLOUD8330, JSWCLOUD-20625, JSWCLOUD-18715, CONFSERVER-66567, CONFSERVER-66567, JRASERVER-72836, CONFSERVER-73384, CONFCLOUD-72861, CONFSERVER-66547, JWMCLOUD-105, SRCTREEWIN-13863, BCLOUD-19548; CONFCLOUD-73781; CONFSERVER-66547: JRACLOUD-77460: BAM-21778; JRASERVER-73435; CONFSERVER-79118; SRCTREEWIN-13863; OPSGENIE-396

Bugzilla Bug Tracking System.

https://bugs.eclipse.org/bugs/Bug 121995

Bug 178923, Bug 532097, Bug 121995, Bug 575890, Bug 551483

Red Hat Bugzilla Bug Tracker System. https://bugzilla.redhat.com/show_bug.cgi?id=1972274

1972274, 1761088, 72861

Arch Linux Bug Tracker. https://bugs.archlinux.org/task/67858

FS#67858, FS#64450

Apache's JIRA Issue Tracker. https://issues.apache.org/jira/projects/MESOS/issues/MESOS-10196

MESOS-10196, MESOS-10194, MESOS-10192, NetBeans: 141198

Eclipse's Bugzilla Issue Tracker. https://bugs.eclipse.org/bugs/show_bug.cgi?id=270754,

270754, 542090, 478785, 515808,

Moodle Tracker. https://tracker.moodle.org/projects/MDLSITE/issues/MDLSITE-5494

MDLSITE-5494, MDLSITE-4617, MDLSITE-3820,

[ii] Selected Jira Bug Report for Illustrative Approach

Bitbucket Cloud: BCLOUD-19548, JIRA Bug Tracking System. [Online]. Available: https://jira.atlassian.com/browse/BCLOUD-19548 [Accessed: 02-Dec-2022].