Thesis no: MSSE-2016-10

Quality metrics in Continuous Delivery

A Mixed approach

Aman Jain
Raghu ram Aduri

Faculty of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona Sweden

This thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Software Engineering. The thesis is equivalent to 20 weeks of full time studies.

Contact Information:
Author(s):

1. Aman Jain
   amja15@student.bth.se

2. Raghu ram Aduri
   raad15@student.bth.se

University advisor:

Indira Nurdiani Jabangwe
Doctoral Student
Department of Software Engineering

Faculty of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona, Sweden

Internet : www.bth.se
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57

ABSTRACT

Context. Continuous delivery deals with concept of deploying the user stories as soon as they are finished rather than waiting for the sprint to end. This concept increases the chances of early improvement to the software and provides the customer with a clear view of the final product that is expected from the software organization, but little research has been done on the quality of product developed and the ways to measure it. This research is conducted in the context of presenting a checklist of quality metrics that can be used by the practitioners to ensure good quality product delivery.

Objectives. In this study, the authors strive towards the accomplishment of the following objectives: the first objective is to identify the quality metrics being used in agile approaches and continuous delivery by the organizations. The second objective is to evaluate the usefulness of the identified metrics, limitations of the metrics and identify new metrics. The final objective is to is to present and evaluate a solution i.e., checklist of metrics that can be used by practitioners to ensure quality of product developed using continuous delivery.

Methods. To accomplish the objectives, the authors used mixture of approaches. First literature review was performed to identify the quality metrics being used in continuous delivery. Based on the data obtained from the literature review, the authors performed an online survey using a questionnaire posted over an online questionnaire hosting website. The online questionnaire was intended to find the usefulness of identified metrics, limitations of using metrics and also to identify new metrics based on the responses obtained for the online questionnaire. The authors conducted interviews and the interviews comprised of few close-ended questions and few open-ended questions which helped the authors to validate the usage of the metrics checklist.

Results. Based on the LR performed at the start of the study, the authors obtained data regarding the background of continuous delivery, research performed over continuous delivery by various practitioners as well as a list of quality metrics used in continuous delivery. Later, the authors conducted an online survey using questionnaire that resulted in ranking the usefulness of quality metrics and identification of new metrics used in continuous delivery. Based on the data obtained from the online questionnaire, a checklist of quality metrics involved in continuous delivery was generated.

Conclusions. Based on the interviews conducted to validate the checklist of metrics (generated as a result of the online questionnaire), the authors conclude that the checklist of metrics is fit for use in industry, but with some necessary changes made to the checklist based on the project requirements. The checklist will act as a reminder to the practitioners regarding the quality aspects that need to be measured during product development and maybe as a starting point while planning metrics that need to be measured during the project.

Keywords: Agile approaches, DevOps, quality metrics, continuous delivery

ACKNOWLEDGEMENT

Firstly, we thank all the software practitioners who participated in our survey by sharing their experiences, giving inputs and suggestions throughout our thesis. Secondly we would like to thank our supervisor Indira Nurdiani Jabangwe, Doktorand, Department of Software Engineering, for trusting in us. We were able to complete the thesis in speculated time frame with the support, encouragement and guidance from our supervisor.

We are very lucky and grateful to our parents for believing in us and for supporting us. We are very glad for their love and affection. We are lucky to have some friends (who are more than a family) who taught us life and made our life quite easier. Lastly we thank our project manager Mr. Gurudutt Velpula, without whose encouragement our master's degree would not have been possible. Finally, we extend our sincere thanks to all those who supported us either directly or indirectly and who stood by our side in all phases of our study in Sweden.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 Introduction to research context

"Our highest priority is to satisfy the customer through early and continuous delivery of valuable software" -Agile manifesto [1]

Continuous delivery deals with concept of deploying the user stories as soon as they are finished rather than waiting for the sprint to end [4]-[8]. This concept increases the chances of early improvement to the software and provides the customer with a clear view of the final product that is expected from the organization. Little research has been done on the quality metrics used in continuous delivery. The objective of this research is to present a metrics checklist that can act as a starting point while planning the metrics to be measured during the project.

Most of the research present in literature states that continuous delivery in agile environment has a positive impact on the success of the project [6]. The success of the project is based on the customer satisfaction which is influenced by the quality of product delivered by the organization. The quality of product is measured using metrics and little research has been done related to quality metrics used in continuous delivery. The only thing that differentiates this research from the existing literature is the metrics checklist that is generated which can act as an aid for software practitioners to plan the metrics to be measure during the project.

Organizations involve the use of metrics to measure product quality based on certain benefits perceived by the top-management [13][14][15]. Some of the perceived benefits by the top-level management for using metrics are as follows: metrics facilitate better business related decisions, allow scope for improvement in case of deficiency, to determine success of the project and provide challenging work to the team members.

Quality metrics in field of software development have been investigated in number of empirical studies [29]. The knowledge of these studies have not yet been organized at one place. The authors are not aware of any studies that identify the state of practice on quality metrics in continuous delivery and aggregate it with knowledge from literature. The lack of aggregated knowledge may result in the misinterpretation by software practitioners regarding the usefulness of the metrics involved in continuous delivery. This study aims to support research and practice on quality metrics involved in continuous delivery.

The problem is addressed by identification and aggregation of the knowledge, from the literature and in practice on metrics used in continuous delivery. The aggregation of the knowledge from literature and in practice would help to create a metrics checklist which would help to characterize the metric usage activity during various phases of the development cycle.

The Latter part of this chapter is structured as follows: section 1.2 presents a description of the aim and objectives of this thesis. Section 1.3 presents the research questions that are investigated in this thesis. A description of the expected outcomes of the research is presented in section 1.4. A description of the various phases involved during the research are presented in section 1.5. Finally, a description of the thesis structure is presented in section 1.6.

1.2 Aim and objectives:

1.2.1 Aim of research

The Aim of this research is to present a checklist of metrics involved in continuous delivery that can be used by organizations as a part of planning which metrics to use during the project.

1.2.2 Objectives of research

Three objectives were formulated as an aid for achieving the aim of thesis. The objectives are as follows:

- The main objective of the project is to identify the quality metrics involved with continuous delivery by performing a Literature Review (LR). As specified in introduction, little research has been done regarding metrics used in continuous delivery. So the authors decided on identifying the existing knowledge presented in scientific articles related to quality metrics used in continuous delivery.
- The secondary objective is to evaluate the usefulness of the identified metrics, limitations of the metrics and identify new metrics. This objective is fulfilled by employing one of research methods i.e., survey performed using online questionnaire.
- The third objective is to present and evaluate the checklist of metrics generated based on the responses obtained from the questionnaire. To fulfill this objective, the authors decided to perform interviews with practitioners that have prior working experience with continuous delivery.

1.3 Research questions

The research performed by the authors is based on two research questions. One question provides foundation to authors regarding the background and use of continuous delivery. Other question deals with the main objective of the research i.e., to find the metrics that can help practitioners ensure quality of product being developed using continuous delivery. The figure 1.1 below represents the research instrument that helped in achieving the objectives of the research. The research questions that are included in our studies are as below:

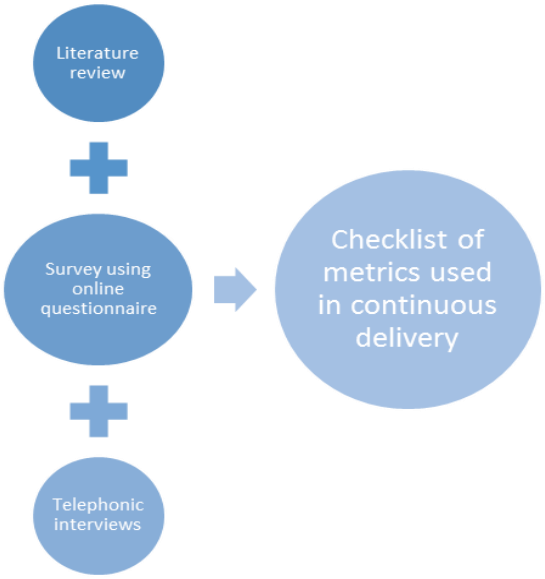

Figure 1.1: Instrument used for the research

RQ 1. What quality metrics are being used to measure product quality reported in continuous delivery literature?

RQ 2. Which metrics are highly useful as perceived by practitioners in continuous delivery approaches?

RQ 2.a. What quality metrics are being used in practice to measure software quality in continuous delivery?

The rationale behind selecting this question is the void of knowledge in the field of metrics to measure product quality in continuous delivery. One may question how metrics used in agile methods can fit to continuous delivery, but based on a research performed by Virmani [5], agile method is the base for continuous delivery. So the metrics that are used in agile can be applied to continuous delivery as well.

RQ 2.b. How do managers and practitioners perceive the usefulness of metrics used in continuous delivery to ensure software quality?

RQ 2.c. What are the limitations of quality metrics as perceived by managers and practitioners?

The rationale behind this research question is the aim of the research to evaluate the usability of the metrics to measure product quality in continuous delivery.

RQ 3. Is the quality metrics checklist fit for use (valid) in software organizations involved in continuous delivery?

1.4 Expected outcomes

The expected outcome at the end of the research is a checklist of potential metrics that can be used by software practitioners to plan the metrics to be used while developing software product using continuous delivery. The checklist is generated based on the results obtained from Literature review and the online survey performed using a questionnaire. The checklist is evaluated using telephonic interviews with practitioners involved with continuous delivery.

1.5 Phases involved in research

The research is conducted in three phases. During the first phase, the authors conducted literature review to identify the existing research related to metrics involved in continuous delivery. Based on the output obtained from the literature review, the authors performed empirical survey using online questionnaire to investigate the usefulness of the metrics identified and also identify new metrics. Based on the responses obtained from the survey, the authors presented a checklist of quality metrics used in continuous delivery. This checklist was evaluated by conducting interviews with practitioners involved in continuous delivery to check for its usability in software organizations. The phases are represented in figure 1.2 below.

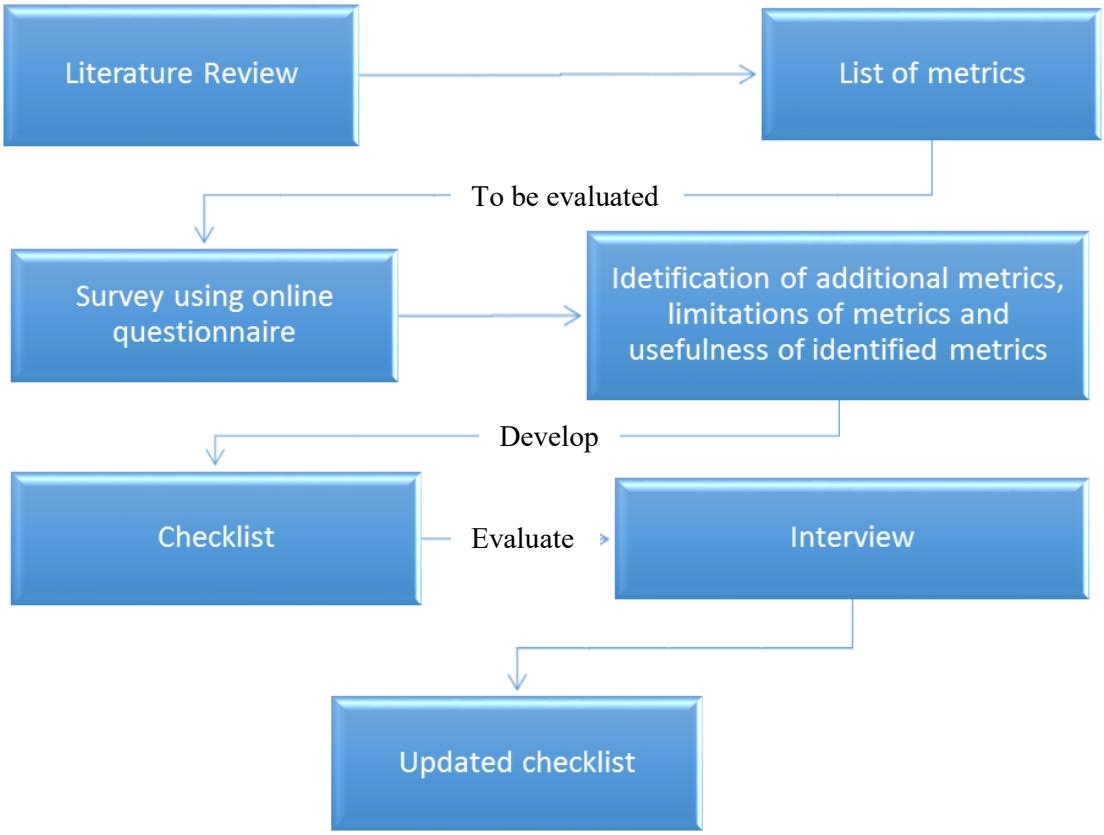

Figure 1.2: Phases involved in research

1.6 Thesis structure

The thesis document is composed of 6 chapters i.e., introduction, related work, methodology, results, analysis and discussion and finally conclusion and future work. The thesis structure is represented in figure 1.3 below.

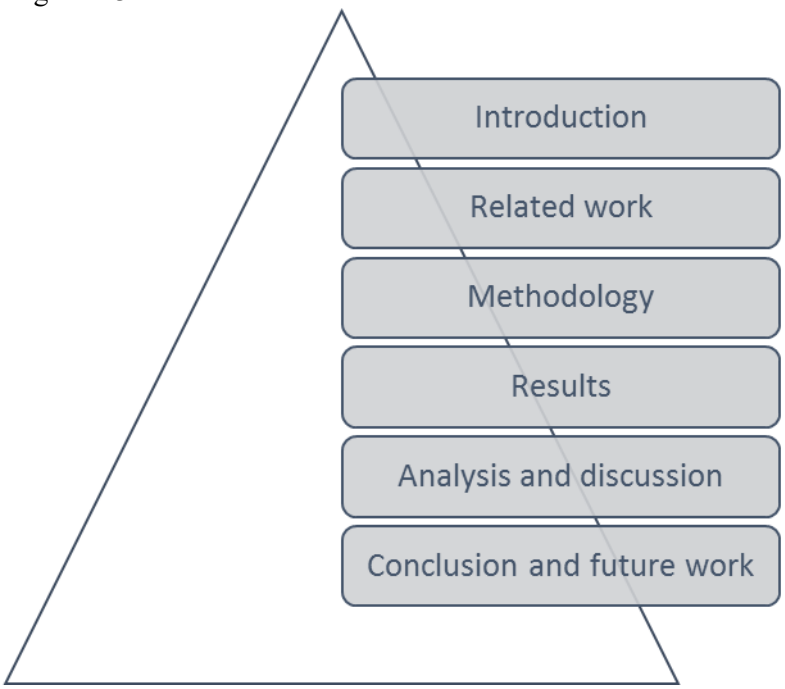

Figure 1.3: Thesis structure

The first chapter of the thesis is introduction. The subsections involved in first chapter are introduction to research context, aim and objectives, research questions, expected outcomes and finally phases involved in the research. The second chapter is background and related work. This chapter represents the existing body of knowledge regarding the quality metrics involved in agile methodology as well as continuous delivery. The subsections involved in this chapter are introduction to software metrics, introduction to rapid releases followed by agile development methodology and finally continuous delivery. Each of these sub-sections is further divided into smaller sections for ease of understanding. The third chapter is methodology which presents a detailed description of the empirical research methods used in this research to fulfill the objectives of research. The fourth chapter presents a description of the results obtained based on the data collected from the empirical survey conducted using online questionnaire and telephonic interviews. The fifth chapter represents the findings of the research based on the results obtained using empirical survey. The sixth chapter presents a description of the final conclusions made based on the data analysis performed over the collected data. A description of the future work that can be performed in the field of quality metrics in continuous delivery is also presented in chapter 6.

2 RELATED WORK

The study reveals the existing knowledge regarding the metrics involved in continuous delivery. A description of the background related to continuous delivery has also been presented in this chapter. Before digging deep into the literature regarding metrics in different development methodologies, a description of the concepts involved with software metrics is presented. A description regarding the importance of software metrics during product development and various attributes involved to ensure definition of good metrics is presented.

2.1 Introduction to software metrics

Software development nowadays has turned into a field where the organizations aim at earning profits by transforming human personnel effort into working software[13]. The organizations primary aim is to provide good quality software to the customers that satisfies all the requirements and solves the customer's problems. Now the question arises on how we interpret that a product possesses good quality. The answer to this question is presented by the use of quantitative methods such as cost and effort estimation, productivity measures, data collection, quality assessment, reliability models and management by metrics [14]. The authors of this document are addressing the concept of quality measurement using metrics based on the research gap present regarding the usability of metrics in continuous delivery. The measurement is performed by the use of metrics. The implementation of metrics involves the use of measures, metrics and indicators.

> ***"Measure is a way to appraise or determine by comparing to a standard or unit of measurement, such as the extent, dimensions, and capacity, as data points [14]"***
>
> ***"Metric is a quantitative measure of the degree to which a component, system, or process possess a given characteristic or an attribute [14]".***

2.1.1 Why to use metrics?

Metrics are extensively being used by HP, Motorola, NASA, AT&T, Boeing and many others [13]-[21]. One may question the need for involving metrics in development process. It is answered by Sanjay Mishra et.al regarding the reasons for involving metrics to measure attributes in development process[15]. The need to involve metrics are presented below:

- Facilitate better business related decisions by the organizations.
- Providing challenging work to the team members
- To determine the success of the team
- To allow the team members to improve in case of any deficiency in performance
- Increases the satisfaction levels of the team members.
- Increased cohesiveness and trust levels among the team which has a positive influence on the organization.

2.1.2 Attributes of good metric definition

There are various metrics that can be used for measurement of the process, product, resources and project. Due to lack of proper metric planning by the managers, most of the project resources are consumed in measuring various aspects of project that are of irrelevant to the project [16]. It is advised to define metrics only for the process or resources that need

to be improved to enhance the quality of product. Definition of metrics irrelevant to project results in resistance from team which serves as one of the challenges most organizations face. To ensure that these challenges are mitigated, Hartmann and Dymond presented about the various attributes involved with good metric definition. A good metric is defined by the following properties[16].

- Affirms and reinforces development methodology principles
- Measures outcomes instead of output
- Follows trends
- Answers a particular question for a human personnel
- Belongs to a small set of metrics
- Easy to collect
- Transparent context
- Provides means for useful discussion
- Frequent feedback cycles
- Ensures "good enough" quality

Based on the above described properties of metric definition, the metric planning and evaluation is performed. To ensure the intention of the metric planned, Hartmann and Dymond recommended the use of checklist to avoid ambiguity related to metric [16]. The checklist is presented in figure 2.1 below.

2.2 Introduction to rapid releases

The rapidly changing software environment has transformed the ways of working and software development methodologies to enhance customer satisfaction. Most organizations aim at providing software faster to solve user's problem. The faster software delivery drew the attention of most organizations which resulted in the evolution of concept of rapid releases. The concept of rapid releases is based on iterative model where in the phases of traditional waterfall model are transformed into a single iteration which results in delivery of product in short cycles. Each iteration lasts from 8-12 weeks. At the end of every iteration, new features or improvements are released for the product. Large organizations like Google, Facebook, Ericsson and Mozilla involve the use of rapid releases based on the benefits are also addressed in this paper. Though there are perceived benefits for rapid releases, there are also challenges which if addressed by the organizations can lead to fruitful results. The challenges are also addressed in this paper below.

2.2.1 Benefits of Rapid releases:

The benefits of rapid releases[17] over traditional waterfall model are as follows:

- Higher quality, as new fixes and improvements are released more quickly which has a positive influence over the usability
- Better service as the project team is more concerned at solving the issues of the users
- Decreased cost is one of the perceived benefits as the new versions are believed to be developed as lower costs compared to older versions
- Increased competitiveness due to the rapidly changing customer needs and an increase in the number of organizations providing similar product and services
- Increased marketing based on the services presented by the new versions. Positive response from the users over blogs and technical workshops works as a source of marketing

Metric planning checklist

- **Name:** this should be well chosen to avoid ambiguity, confusion, oversimplification.
- **Question:** it should answer a specific, clear question for a particular role or group. If there are multiple questions, design other metrics.
- **Basis of Measurement:** clearly state what is being measured, including units. Labeling of graph axes must be clear rather than brief.
- **Assumptions:** should be identified to ensure clear understanding of data represented.
- **Level and Usage:** indicate intended usages at various levels of the organization. Indicate limits on usage, if any.
- **Expected Trend:** the designers of the metric should have some idea of what they expect to see happen. Once the metric is proven, document common trends.
- **When to Use It:** what prompted creation or use of this metric? How has it historically been used?
- **When to Stop Using It:** when will it outlive its usefulness, become misleading or extra baggage? Design this in from the start.
- **How to Game It:** think through the natural ways people will warp behavior or information to yield more 'favorable' outcomes.
- **Warnings:** recommend balancing metrics, limits on use, and dangers of improper use.

Figure 2.1: Metric planning checklist as presented in [16]

2.2.2 Challenges of Rapid Releases:

The challenges [17] involved with rapid release cycle are as follows:

- Difficulties faced by users while using the product or service as the user might feel unsecure due to frequent release of new versions of the products which in turn leaves the user confused regarding the reliability on the product or service.
- Difficulties faced by developers to avoid vulnerabilities in the older versions of the product
- Difficulties faced by enterprises as the development team involves delivering new versions more often. The investment required for maintaining the new software and older versions is quite challenging without proper hardware and infrastructure support.
- Cost factors also have a significant influence over the product development using rapid release. The costs involved with maintaining the older versions increase with the release of new versions.

- E-waste also poses a challenge to rapid releases. The e-waste increases due to the advent of new products as well as availability of similar products in the market.

Once the above challenges are mitigated, rapid release turn out to be profitable to organizations. Kushwaha et.al [17] has presented remedies in order to minimize the impact of challenges over the product and project. Kerzazi and khomh [18] has presented three factors that impact the rapid releases in any organization. The three factors presented are as technical factors, organizational factors and interactional factors. The various sub-attributes in each factor that have an impact on the product release are also presented by kerzazi and khomh[18].

2.2.3 Evolution of rapid release approach:

The evolution of rapid release approach during the past ten years is presented in figure 2.2 below.

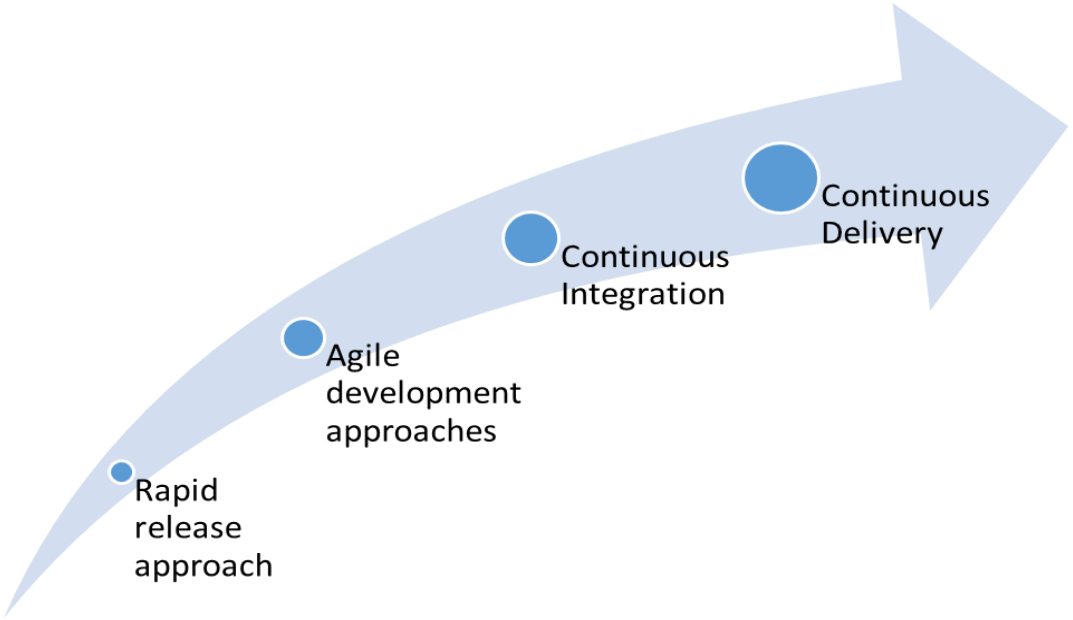

Figure 2.2: Evolution of rapid release approach

2.3 Agile development methodology

Agile development methodology is based on the concept of rapid releases. It is an iterative and team-based development approach. As discussed earlier, rapid releases make use of short delivery cycles that last for 8-12 weeks. At the end of the cycle, set of new features or improvements are presented to customers. Two of the most popular agile approaches in practice are scrum and Extreme Programming (XP). The short delivery cycles are termed as iterations that involve all the development phases involved in traditional waterfall model[19]. The main motive of agile approaches is to increase customer satisfaction by ensuring faster product delivery. The agile methodology is based on the principles specified by agile manifesto. The four key principles presented in the agile manifesto are presented below.

- Value individual and interaction
- Delivering quality working software
- Customer collaboration and satisfaction
- Responding to change over documentation, processes and plans

Agile development involves the use of short delivery cycles, continuous learning, frequent face-to-face communication and frequent deliveries.

2.3.1 Scrum:

Scrum is one of the popular agile approaches that involve the use of short iterations called sprints. Scrum also involves the use of frequent interaction among the team by ensuring scrum meeting such as daily scrum and weekly scrum. In daily scrum, the team is notified regarding the progress of the task allocated to each member. Scrum makes use of user stories that are developed based on customer requirements. These user stories are stored in the product backlog and for each sprint, the user stories are moved into sprint backlog which are then developed into features by the team. The user stories are moved into the sprint backlog based on the priority given to user stories. If any user story is not completed as planned then it is moved back to product backlog. Learning in scrum is ensured by conducting a sprint retrospective meeting.

2.3.2 Extreme Programming:

Extreme Programming is more concerned with customer satisfaction and strong emphasis over agile principles and practices. Some of the practices involved with XP are pair programming, automated testing and continuous refactoring. Refactoring refers to changing source code of a product without actually changing the functionality of the product[11]. Intense collaboration and cohesiveness among the team is facilitated by pair programming and effective communication among the members of the team. Continuous interaction with customer and responsiveness of the team to changes and crisis form the base for extreme programming.

2.3.3 Quality metrics used in Agile:

This study as explained earlier is focused on identifying metrics used in agile approaches to measure the process, project or product. It is truly said that to improve a process or product, it should be measured. Based on survey performed by Buse and Zimmermann over the needs of managers and developers at Microsoft, interpreted that most information collected during the project life cycle is related to quality [11]. The need for using metric is presented in section 1.1.

Based on the literature studied in context of metrics used in agile approaches, the authors of this document categorized metrics into functional metrics and non-functional metrics. These categories are further divided into sub-categories. The categorization of metrics in context of agile approaches is presented in figure 2.3 below. The same sub-category may co-exist in both functional and non-functional categories, but the context in which they are used and how they are applied also differs.

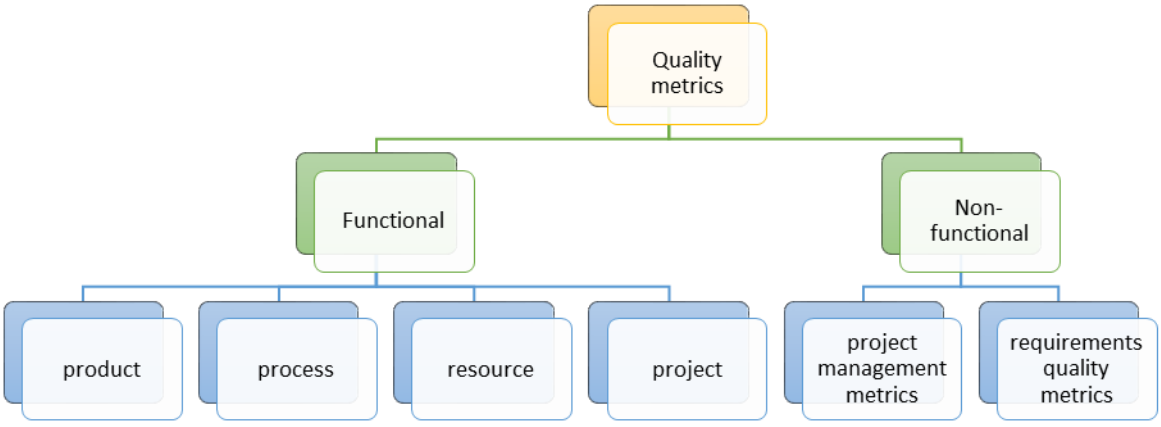

Figure 2.3 : Metric categorization based on [15], [19], [11], [20], [21]

Metric selection has been a challenge for most of the software managers and organizations. Selection of metrics and collecting information that is practically of no significant influence will only consume project resources and have a negative impact over the team's performance in future projects [11]. Improper metric definition and unnecessary use of measurement may have a negative influence on the morale of the team members [20]. The information gathered during the project may be used to improve the process or the project, but the team is mostly not convinced regarding the intention of measuring attributes of projects. It is aptly said that measurement is driven by human personnel behavior. A quote prevailing in software engineering in context of metrics is

"Tell me how you will measure me and I will tell you how I will behave"
- Goldratt [22]

The metrics are selected based on the organizational needs as well as the type of project. It is not mandatory to use metrics defined in the literature, the team can define and design their own metrics based on the attribute they want to measure and improve.

2.3.3.1 Functional metrics:

Functional metrics are termed as metrics that deals with the measurement of attributes related to project, product, process and resources [15], [16], [20]. Table 2.1 presents a detailed list of metrics specified in each sub-category as per the literature.

- **Product Metrics** deals with size metrics, quality metrics, architecture metrics, structure metrics and complexity metrics [15], [16], [20]
- **Project Metrics** deals with cost, time, quality and other attributes related to project [15], [16], [20]
- **Process Metrics** deals with development life cycle metrics, development metrics and maturity level metrics [15], [16], [20]
- **Resource Metrics** deals with hardware metrics, effort estimation metrics and performance metrics [15], [16], [20].

Category	Sub-Category	Metrics
Product metrics	Size and quality metrics	Lines of code, user story, function points, test points, number of defects, defect density
	Requirements and design metrics	Number of requirements per phase, change request per requirement, implemented versus wasted requirements, inventory of requirements over time, requirements cost types.
Project metrics		Burn down, effort estimate, lead time, processing time, progress as working code, remaining task effort, sprint burn down, schedule performance index, story effort, task estimate, cost per performance index.
Process metrics		Cycle time, build status, average velocity, queue time, processing time, planned velocity, story points

		complete,
Resource metrics		Task effort, task complete, variance in handovers, over time, check ins per day, common tempo time, business value delivered

Table 2.1: Functional metrics in agile

2.3.3.2 Non-Functional Metrics:

Non-functional metrics are a part of NORMAP (Non-functional Requirements Modeling for Agile processes) that involve the use of newly proposed project management metrics and requirements quality metrics [19]. Table 2.2 presents a detailed list of metrics specified in each sub-category as per the literature.

- **Project Management Metrics** are used for prioritizing and measuring effort estimation for team and technology processes. Project management metrics are further divided into four categories i.e., general, team, technology and process metrics [19].
- **Requirements quality metrics** deals with measuring attributes related to requirements and design aspects of agile planning [19].

Category	**Sub-category**	**Metrics involved in category**
Project management metrics [19]	General	Sprint duration, relative size estimate, estimate confidence factor, priority scheme, estimate scheme
	Team	Team size, team development factor, team technical competency, team domain competency, team PO & collaborative index, team velocity
	Technology	Technology novelty factor, team and technology novel factor
	Process	Agile process maturity
Requirements quality metrics [19]		Total number of requirements, agile requirements ambiguity, agile requirements completeness, agile use-case density, agile density per sprint, agile density per requirement, volatility threshold, ambiguity of requirements impact, validation of requirements impact

Table 2.2: Non-functional metrics in agile

2.4 Continuous Delivery

2.4.1 Introduction to DevOps

As discussed earlier, time to market is one constraint which decides the success or failure of a product. The increase in the software development organizations around the globe has increased competition among the organizations to deliver good quality software in least possible time. So to survive the competition and mark its existence, the organizations tend to adopt new approaches of faster software delivery to market. One such approach that was defined to address the problem of faster delivery is Continuous Integration (CI).

Continuous integration has presented the software industry with faster delivery approach where the developers are allowed to commit more frequently to shared repository. Automated unit-tests are triggered after every commit to detect problems with any bad commit which results in reduced necessity of backtracking [23]. The time required for issue detection is decreased which results in increased productivity and faster software development [23]. The increased demand for approaches for faster software delivery allowed the evolution of Continuous Integration (CI) into continuous delivery (CD).

The concept of "DevOps" was introduced to support continuous integration and continuous delivery as the development and operation team need to work in close collaboration to ensure faster software delivery. DevOps is a new role that was termed by merging the roles of developer and operator. Development team is mostly responsible for development phase of life cycle whereas operations team is responsible for maintenance phase of development life cycle. After the completion of each phase, the work is passed to next phase which requires manual steps and approval which serves as one of reason for slower software delivery. DevOps has no formal definition. The meaning of the concept of DevOps is still a mystery for practitioners as it is presented in various ways in scientific articles and technology blogs. According to Manish, "DevOps is set of practices that is trying to bridge developer-operations gap at the core of things" [24]. Manish also presented guidelines for applying concept of DevOps to various phases of software delivery such as continuous planning, continuous integration, continuous deployment, continuous testing and continuous monitoring. DevOps ensures that the time to market is reduced, balance between cost and quality is achieved and organizational efficiency is increased.

2.4.2 Introduction to continuous delivery

Continuous delivery is based on the agile principles of faster delivery and customer satisfaction. Continuous delivery is termed as an approach where the project team has the ability to release software at their own will and wish [6]. Chen [24], [25] termed continuous delivery as a software engineering approach where development team produces software in short cycles and the software developed during the cycle can be potentially released at any time as the software undergoes all the quality checks before being releasable [24], [25]. The deployments are done weekly or daily depending on the complexity of the product. The characteristics of continuous delivery may help for better understanding regarding the approach are presented below.

- Releasable at any time/ Frequent releases to ensure fast feedback from user which facilitates continuous improvement of the product [25].
- Reliable/ Automated release with the help of engineering tools and software such as Jenkins which are designed to support continuous delivery [23].
- Delivering valuable software as the product passes through various tests and quality checks before being ready for deploying [25].
- Small size of the user story enables the product development must faster and easier [25].

Continuous delivery is automated so the systems involved must be engineered prior to start of development, but now the question arises about the reason to consider continuous

delivery to ensure faster software delivery. The answer to this question is obtained by considering the benefits of continuous delivery.

2.4.2.1 Benefits involved with Continuous Delivery:

The benefits involved with continuous delivery as per literature [24], [25] are presented below.

- Accelerated time to market
- Building the right product
- Improved productivity and efficiency
- Reliable releases
- Improved product quality
- Improved customer satisfaction

2.4.2.2 Challenges involved with Continuous Delivery:

Though there are huge benefits involved with continuous delivery, it has its own challenges [24], [25]. The challenges presented in the literature are presented below.

- Organizational challenges as the organizations involve various departments and each department has their own style of working. So collaboration among these departments may pose challenge to the software product delivery.
- Technical challenges involve the use of infrastructure and setup required for effective and free flow of the system.
- Process challenges involved with continuous delivery is lack of familiarity with the development approach. The resistance of the human personnel to adapt new processes may also turn challenging for the organizations.

2.4.3 Research work in the field of continuous delivery

The recent work related to continuous delivery in the context of architecture of CD, networking, developing mobile applications and testing are presented below.

Architecting Continuous Delivery:

Chen [25] presented about his personal experience while moving applications from traditional approach into continuous delivery in an organization namely Paddy power. He also presented the organizational context of paddy power while transforming into continuous delivery architecture. A description of the characteristics and perceived benefits of continuous delivery are also presented. A description of the need for architecting the continuous delivery approach is also presented. Before architecting continuous delivery suitable for the application, the constraints of deployability, modifiability, security, loggability, monitoring and testability need to be fulfilled.

Continuously delivering your network:

Gebert et.al [26] presented a description on the adaptation of continuous delivery in the networking world. A description of continuous delivery and the various stages involved in continuous delivery pipeline are also presented. The need for applying continuous delivery for network functions and also recommended deployment pipeline for network functions.

Continuous Delivery of Mobile Apps:

Klepper et.al [27] presented a case study conducted in the context of applicability of continuous delivery in industries. The authors measure the applicability of continuous delivery in capgemini, where development was performed using rugby, an agile process model with workflows for continuous delivery. A description of the issues encountered during the applicability of rugby was also presented. They also suggested splitting up rugby workflow into four activities i.e., configuration management, integration, delivery and feedback to ensure successful delivery of mobile applications in industry. The tailored approach was applied to eight projects at capgemini and based on the analysis interpreted that application of workflow reduces time to build and deliver mobile applications.

Automated testing in Continuous Delivery Pipeline:

Gmeiner et.al [28] presented a case study conducted in an online company in context of how automated testing is used in continuous delivery pipeline. A description of the company context and technical and organizational challenges while applying automated testing in continuous delivery pipeline are presented. The authors also presented the lessons learned while working with continuous delivery pipeline which can act as recommendations for successful establishment and operation of continuous delivery pipeline.

2.4.4 Continuous delivery pipeline

Continuous Delivery as mentioned above mostly automated where in the developer commits the work done into a central server or repository. Automated unit tests are performed over every commit and the bad commits are reported back to the developers. If the code passes the unit tests it is moved to the production department which can deploy the code to customer with click of a button. The continuous delivery is based on the continuous delivery pipeline. The pipeline comprises of 6 phases which are subject to vary based on the type of application being developed. The pipeline involves the use of both automatic promotion and manual promotion of work from one phase to next phase. Figure 2.4 depicts a sample pipeline.

Code commit:

Code commit stage is the initial stage of the continuous delivery pipeline. The developers check-in the code developed into the central repository then this phase is triggered automatically. The code that is checked in is compiled and unit tests are triggered. If there is a problem with code, the pipeline stops and developer is notified of the problem. Based on the problem, the developers make changes in the code and updated code is checked back into the code commit stage which triggers the code commit stage again [24]. If there are no problems with code commit, it moves to next stage.

Build:

The build stage performs unit testing again and various static analysis to generate artifacts required for release. The artifacts are then uploaded into central repository which are later used for deployment [24]. The pipeline stops if there are any errors during testing and notifies the developer. If there are no errors then the pipeline automatically moves to the next stage.

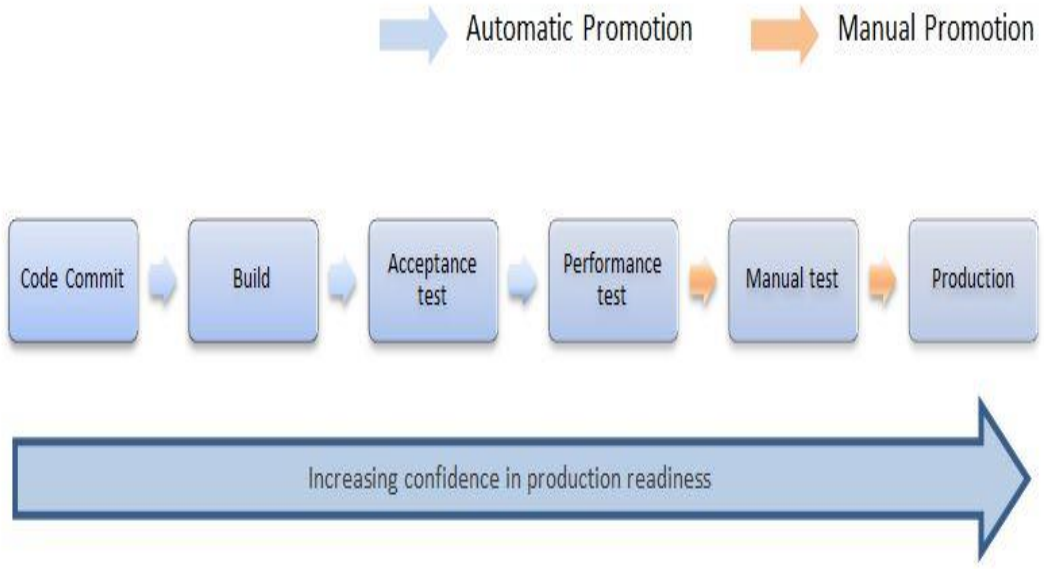

Figure 2.4: Continuous delivery pipeline[24]

Acceptance test:

This stage ensures that the product developed satisfies all the customer requirements. For acceptance testing in traditional models, the test setup was built manually, but in continuous delivery, the test setup for acceptance testing is created automatically where the product can be deployed and tested [24]. If there are any errors, the pipeline stops and notifies the developers about the error. If there are no errors while running, the pipeline automatically moves to next stage.

Performance test:

Performance test is performed to ensure that the developed product is reliable and performs well [24]. The environment required for performance test is automatically generated by pipeline and the performance test is conducted over the code. The performance test also helps to ensure the quality of the software [24]. If there are any errors, the pipeline stops and notifies the developers about the error. If there are no errors while running, the pipeline automatically moves to next stage.

Manual test:

Manual test is performed by the human personnel involved in project. Though automated testing is beneficial, manual testing is performed to ensure that there are no bugs in the software [24]. For performing manual testing in traditional development approaches, the test environment is setup by the testers, but in continuous delivery, the test environment is setup by the pipeline and the testers can test the product in that environment [24]. Once the manual testing is performed, the product is ready for release and can be moved to the production stage.

Production:

The product is deployed by the production manually by just clicking a button as the scripts have been tested and checked for quality in the previous stages of the pipeline [24].

2.4.5 Metrics involved in continuous delivery:

It is aptly said that if an attribute cannot be measure, it cannot be improved. As Continuous delivery aims at increasing customer satisfaction by providing good quality software product in short time span, it is important that it is continuously improved. So to measure the attributes of continuous delivery, metrics need to be defined. The need to involve metrics are presented in section 2.1. Metrics can help to detect issues with pipeline and allow chances of process improvement by eliminating waste.

Lehtonen et.al [29] has divided metrics used in continuous delivery into two categories. One is metrics on implementation level and another one is metrics on pipeline level.

Metrics on Implementation level [29]: These are dependent on the toolset and practices used to implement the pipeline. The data flow and throughput are dependent on the implementation of pipeline. The actual metrics involved in this implementation level are development time, deployment time, activation time and oldest done feature.

Metrics on Pipeline Level [29]: These metrics are isolated from the implementation of pipeline. The actual metrics in this category are features per month, releases per month and fastest possible feature lead time.

Continuous delivery is evolving topic in field of software development approaches. In continuous delivery literature, quality metrics are divided into two categories i.e., metrics at implementation level and metrics at pipeline level. There is little research performed over identifying the metrics involved in continuous delivery and aggregating the metrics with the knowledge present in literature. Our research aims at addressing the research gap by presenting a checklist of metrics involved in continuous delivery.

3 METHODOLOGY

The authors involved a mixed method approach to achieve the objectives of the research. The mixed method approach involves the use of qualitative research method as well as quantitative research methods to arrive at a consensus. The quantitative research method employed in this research is one of the survey methods i.e., online questionnaire. The qualitative research method employed in this research is literature review and telephonic interviews.

In this research, Literature review is performed in order to present the existing body of knowledge related to continuous delivery and quality metrics involved in continuous delivery. Based on the information obtained from the literature review, the authors designed an online questionnaire to evaluate the usability of the metrics involved in continuous delivery and also to identify any new metrics used in continuous delivery which helps the organizations to ensure good quality product delivery. Based on the responses obtained for the online questionnaire, the authors generated a checklist of quality metrics involved in continuous delivery. This checklist was evaluated using qualitative research method i.e., telephonic interviews.

A description of the research methodology followed by the authors to achieve the objectives by investigating the research questions is presented in this section. A description of the rationale behind selecting the research method is also presented along with the research design used by each method to investigate the research questions.

The remainder part of this chapter is structured as follows: section 3.1 presents a description of the rationale behind selecting literature review and also presents a description of the steps followed while performing literature review. Section 3.2 presents a description of sequence of steps followed by the authors while performing survey using online questionnaire. Section 3.3 presents a description of sequence of steps followed by the authors to validate the checklist using telephonic interviews. A mapping of the research method used to answer the research question in this study is presented in figure 3.1 below.

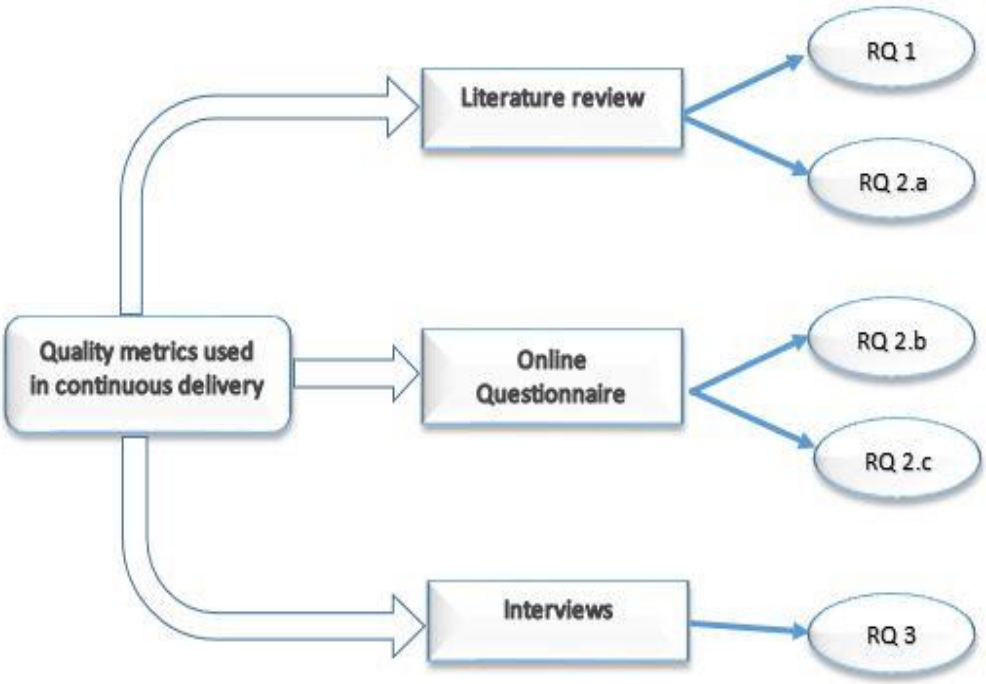

Figure 3.1: Mapping of research method with research question

3.1 Implementing Literature review

3.1.1 Rationale to select Literature review

The motivation behind selecting literature review is that it helps the authors to understand about the existing knowledge in the particular field i.e., regarding the metrics being used to ensure quality of product in agile methodology [9]. One of the readers may question that why we chose literature review rather than Systematic Literature Review (SLR). The reason why we chose literature review over SLR is that literature review deals with gathering all the evidences using a predefined search strategy that fits the research hypothesis. Whereas SLR on the other hand is intended to collect evidence of a treatment [12]. Meanwhile in this study, using literature review we are trying to identify and aggregate metrics used in continuous delivery. So our intention does not fit the aim of performing an SLR as specified by Kitchenhamn and Charters [12], however fits the aim of performing literature review as specified by Rowley and Slack [30].

3.1.2 Research questions:

The literature review is performed to help the authors investigate about two research questions. The research questions are presented below:

- **RQ 1:** what quality metrics are being used to measure product quality reported in continuous delivery literature?
- **RQ 2.a.:** what quality metrics are being used in practice to measure software quality in continuous delivery?

3.1.3 Study type, design and Execution

3.1.3.1 Study Type:

The objective of this research was to identify the quality metrics involved in agile and continuous delivery approaches. To fulfill this objective, the authors adopted literature review as its research method. The authors involved the use of online databases to search for literature related to quality metrics in agile and continuous delivery. The sequence of steps followed to perform LR are presented below.

1. Decide upon the database to be used to search for articles to perform LR.
2. Decide on the keywords to be used to perform search in the database.
3. Once the keywords are decided, search for database using the keywords.
4. Obtain an initial set of articles based on the inclusion criteria set by the authors.
5. Using backward search over the initial set of articles, select the articles most relevant to the research context.
6. Study the selected articles and document the useful information i.e., regarding continuous delivery and metrics used in agile development as well as continuous delivery in this case.

3.1.3.2 Design:

The design of the literature review involves the selection of database, deciding on the keywords to be used, selecting the inclusion criteria.

- *List of databases used to search for literature:* The authors decided to use online article databases such as IEEE Explore, ACM digital library and Science Direct.
- *Selected keywords:* metrics, measurement, software quality, software development, agile, continuous delivery
- *Search string:* (((((((metrics) OR measurement) AND software quality) AND ((software development) OR software design)) AND agile) AND continuous delivery). The search string mentioned above is updated based on the database being

searched by the authors to obtain scientific articles related to background and quality metrics involved in continuous delivery and agile approaches.

Inclusion criteria: The authors reviewed only scientific articles rather than complementing it with metrics specified in blogs and forums. The reason behind considering only scientific articles is that they are peer reviewed and the scope of the research is limited due to time constraints for the research. The inclusion criteria is used to by the authors to select or reject articles to be included in the research. The criteria set by the authors is presented below:

- The articles must be peer-reviewed
- The articles must be relevant to quality metrics in context of agile approaches
- The articles must be relevant to quality metrics in context of continuous delivery
- The full text of the articles must be available online
- The articles must be in English
- The articles must be published within 1995-2016.

3.1.3.3 Execution:

Once the design for the literature review is decided, the authors followed the 5 steps specified by Rowley and Slack in their article "conducting a literature review" [30] to guide master's students. The five steps stated in the document are "*scanning documents, making notes, structuring the literature review, writing the literature review and building the bibliography*" [30].

- *Scanning documents* refers to selecting articles relevant to the context of research using search string. The online databases selected for this task are IEEE explore, ACM digital library and Science direct. The online databases are searched using the search string specified in above sub-section and relevant articles are selected using the inclusion criteria specified above. A total of 20 scientific articles were selected at the end of this step for further analysis.
- *Making notes* is the next step once the literature is collected related to quality metrics involved in agile and continuous delivery approaches. In this step, the authors skimmed through the literature gathered and any information that helps to investigate the research questions is written down and preserved. In this case, information regarding quality metrics in agile approaches and continuous delivery are noted down. Information from 11 out of 20 scientific articles related to quality metrics was noted down.
- *Structuring the literature review* helps in understanding the information noted in the previous step. In this research, the authors intend to find any correlation among the metrics involved in agile approaches and continuous delivery based on the information noted in the previous step.
- *Writing the literature review* helps the authors to provide a summary of the literature studied regarding quality metrics in agile approaches and continuous delivery. This study also helps the authors to present any related research performed by researchers in context of continuous delivery. At the end of this step, seven articles which present the research in context of continuous delivery were identified, information stated in the research was noted down and presented as results of literature review. The topics discussed by each article is presented in table 3.1 below. Out of the seven articles selected, only one article discusses about the metrics used in continuous delivery.
- *Building the bibliography* presents the authors gratitude towards the researchers whose work has been used by citing their research.

Article name	Topic discussed in the article
Continuous Delivery with Jenkins: Jenkins Solutions to Implement Continuous Delivery [23]	Concepts related to continuous delivery
Continuous delivery: Huge benefits, but challenges too [24]	Concepts related to continuous delivery
Towards Architecting for Continuous Delivery [25]	Concepts related to continuous delivery
Continuously delivering your network [26]	Concepts related to continuous delivery
Introducing Continuous Delivery of Mobile Apps in a Corporate Environment: A Case Study [27]	Concepts related to continuous delivery
Automated testing in the continuous delivery pipeline: A case study of an online company [28]	Concepts related to continuous delivery
Defining Metrics for Continuous Delivery and Deployment Pipeline [29]	Concepts related to continuous delivery and metrics used in continuous delivery

Table 3.1 Topics addressed by selected articles

3.2 Implementing survey

The authors selected survey as a research method to identify the usefulness of metrics identified in LR, limitations of identified metrics, also to identify new metrics, present and evaluate a solution (checklist of metrics). The rationale behind selection of survey over case study and experiment is that survey helps to identify the potential views of a sample of population who are actually using a particular tool or method. Case study is mostly used to validate a framework or technique or model only to a specific industry which cannot be generalized to similar industries [10]. Whereas survey on the other hand is used to get results from large sample rather than a single case or industry. The results obtained from the survey can be generalized and used by similar industry which is quite contradictory to case study.

Experiments are mostly used to assess and compare two techniques or approaches and help the researchers to interpret which approach is more effective or efficient [10]. In this case, the authors are keen on providing a checklist of metrics that could be used to measure quality rather than improving or assessing new/existing approaches. So based on the above description regarding the three research methods, the authors selected survey as a research method as the authors aim at gathering metrics being used in industry to measure quality using questionnaire and evaluate their usability using interviews.

3.2.1 Rationale behind selecting online questionnaire

The rationale behind selecting questionnaire is that it allows the respondents (software practitioners) to select most used metrics from a set of metrics which in turn helps the authors to narrow down the metrics that ensure quality in continuous delivery [8][10]. The sequence of steps to be followed to perform survey using questionnaire are presented below:

1. Based on the metrics obtained in LR, design a questionnaire such that it fulfills the second objective of the authors to decide on the metrics relevant to continuous delivery.
2. Design a section that allows the respondents to answer questions related to their background information in the field of software industry.
3. Once the section regarding the background information is designed, a questionnaire which contains questions that helps the authors to narrow down the metrics used for

continuous delivery. This questionnaire also consists of questions that need to be answered in order to investigate the limitations and usefulness of the identified metrics.

4. A final section that involved that allows the respondents to provide any suggestions and comments on the questionnaire is also designed.
5. Once the three sections are designed, a pilot study is conducted by sending the questionnaire to few practitioners to check for the ease of understanding of the questionnaire while answering the questions.
6. Once the pilot study is conducted and the recommended changes are made to the questionnaire, the questionnaire is finally sent to software practitioners working with software development and their responses are recorded.
7. Analysis is performed over the responses to investigate the limitations and usefulness of the metrics involved in continuous delivery.
8. Based on the usefulness of the metrics, the authors generated a checklist of metrics involved in continuous delivery.

3.2.2 Research questions answered by online questionnaire

The empirical survey using online questionnaires is performed to investigate about two research questions. The research questions are presented below:

- **RQ 2.b.** How do managers and practitioners perceive the usefulness of quality metrics used in Continuous Delivery?
- **RQ 2.c.** What are the limitations of quality metrics as perceived by managers and practitioners?

3.2.3 Study type, Design and Execution

3.2.3.1 Study type:

The main objective of this research was to gather data from large sample of practitioners involved in software development irrespective of their role in the organization. The authors used an online questionnaire hosted by www.survio.com to gather data from practitioners and managers. This online tool was used as it was cost effective, easy maintenance of data collected from respondents. The data was stored in online repository resulting in less overhead of storing and managing responses in an external database such as MySQL, Oracle and many more.

3.2.3.2 Design:

The questionnaire was made available to respondents from **March** until **April 2016.** The survey population selected for the survey are the practitioners and managers involved in software development. The initial set of survey population was selected based on personal contacts and by posting the survey link in online forums related to Continuous Delivery in LinkedIn. Snowball sampling was also performed by allowing the initial contacts to recommend the survey to eligible candidates. The questionnaire was designed such that biasing was avoided. The questionnaire involved the use of contingency question to ensure that the respondents who had prior working experience with continuous delivery were only allowed to rate the metrics. The questionnaire structure used for the research is presented in following sub-section and the questionnaire is presented in Appendix A.

3.2.3.3 Questionnaire structure:

The questionnaire designed for data collection comprises of three sections i.e., experience of practitioner in field of software development, questions that provide answers to the research question and finally suggestions/comments. The data regarding Experience of

practitioner in the field of software development is collected using questions specified below.

1. What is your role in the software organization?
2. What is your working experience in the field of software development?
3. Which development methodology does your organization use?
4. How does your organization ensure quality of product being developed?
5. Does your organization use metrics to measure quality aspects of the product being developed?
6. If your organization used metrics, why does your organization adopt metrics?

The questions specified below act as an aid for the authors to answer the research questions specified in section 2.3.1. These questions were included in the second section of the online questionnaire to collect data.

Survey Question(s) for RQ 2.b.

1. How often does your organization release product to market/customer? A free text field was provided to the respondents to answer the question.
2. Do you have prior experience of working with continuous delivery? Respondents were allowed to make a single choice amongst Yes or No. If the respondents answered No, the questionnaire would be redirected to the end of the survey.
3. Continuous Delivery involves metrics at implementation level and pipeline level. Which metrics would you use often to ensure quality? The respondents were presented with a numerical rating scale where they were free to assign a particular value to 2 categories between 0-100.
4. Rate the metrics at implementation and pipeline level based on their usefulness? The respondents were presented with a free text box against each metric and asked to provide a value to each metric between 0-100 based on their usability in the organization.

Survey Question(s) for RQ 2.c

1. What are the limitations of metrics being used to ensure quality? A free text field was presented to the respondents to answer this question. The respondent could answer the question up to 1500 words.
2. Can you recommend any metrics apart from implementation level and pipeline level being used in your organization? A free text field was presented to the respondents to answer this question. The respondent could answer the question up to 1500 words.

3.2.3.4 Execution:

A total of 77 respondents viewed the survey and 49 (63.6%) respondents had prior working experience with Continuous Delivery. Therefore, we only used the responses from these 49 respondents to derive at a consensus.

3.2.3.5 Data analysis:

The data obtained from the online questionnaire is tabulated and is analyzed by using techniques such as frequency distribution and graphical displays [8]. Frequency distribution refers to tabulating the responses for each option of the question presented in survey [8]. The data obtained from frequency distribution is presented in graphical form for better understanding using a bar graph and pie charts. In addition to the above methods, the usefulness of the metrics is analyzed using mean values to derive at a consensus.

3.3 Implementing survey using telephonic interviews

Once the metrics used for continuous delivery, limitations and usefulness of the metrics are obtained, a checklist of metrics is generated. The authors can conduct interviews with software practitioners to evaluate the checklist of metrics designed to help practitioners to ensure quality in continuous delivery. The rationale behind involving interviews with practitioners is to validate the information obtained from the online questionnaire and also get a clear understanding of the limitations of the metrics in practice [8][10]. The sequence of steps to be followed in order to perform interviews are presented below.

1. Select a list of practitioners from the software industry involved in continuous delivery and fix a schedule to meet them.
2. Prepare a list of questions to be asked in interview which help the author to evaluate the checklist of metrics designed based on data obtained from questionnaire.
3. Conduct the interview with the selected participants and note down the points in the interview using meeting minutes.
4. Obtain the consent of the interviewee and if possible record the interview in case of any further reference.
5. Based on the interview with the practitioners, analysis is performed over the data collected and the check list of metrics is evaluated.

3.3.1 Research questions answered by interviews:

The interview is conducted in order investigate about one research question. The research question is presented below.

- **RQ 3.** Is the quality metrics checklist fit for use (valid) in software organizations involved in continuous delivery?

3.3.2 Study type, design and execution:

3.3.2.1 Study type:

The main objective of this research was to evaluate the checklist generated from the data collected from the responses obtained from online questionnaire. The authors intended to perform telephonic interviews with practitioners involved in continuous delivery to validate the checklist and to obtain feedback on the fitness of use of generated checklist in software organization. The rationale behind selection of telephonic interviews over face-to-face interviews is because of the ease of availability of the respondents and increased sense of anonymity among the respondents. Continuous delivery being an evolving approach, few organizations use it for their projects. Lack of availability of the respondents in the same location, the authors opted to perform telephonic interviews over face-to-face interviews.

3.3.2.2 Study design:

The authors conducted 15 interviews to evaluate the usefulness of the checklist of quality metrics involved in continuous delivery. The interviews were conducted during 1 April 2016 to 20 April 2016 based on the availability of the respondents. The sample population selected for the interviews are software practitioners using continuous delivery approach in their organization. Convenience sampling approach was used for sample selection. Snowball sampling was also performed where the initial set of contacts were asked to recommend eligible practitioners who are willing to give an interview and help the authors with the research. The protocol followed while conducting the interviews is presented in the following section.

3.3.2.3 Interview protocol:

The interviews were conducted based on the protocol. The protocol is composed of three sections. The first section is regarding the pleasantries and a small introduction to the interviewee regarding the context of research and objective behind conducting the interview.

The second section is related to the demographics of the interviewee i.e., questions related to background of the interviewee in the field of software industry. The third section is regarding the perception of the interviewee regarding the usefulness of the checklist of quality metrics involved in continuous delivery generated by the authors.

The questions asked by the authors to gather data regarding the demographics of the interviewees are presented below.

a. Please specify the number of years of experience in field of software industry.
b. What is the development approach used in your organization?
c. What type of method does your organization use to ensure quality of product (quality reviews, quality standards, process improvement approaches, metrics and measurement)
d. How often does your organization release product into the market?
e. Does your organization use metrics to measure software attributes?
f. Does your organization involve in continuous delivery?

The questions asked by the authors to evaluate the usefulness of the checklist are as follows:

a) What is the level of complexity in understanding the metrics specified in the checklist? (Rating)
b) What challenges do you face?
c) Would you recommend the usage of this checklist in your organization? Specify why you would use it or not use it?
d) Please specify some recommendations that can be made to the checklist to make it more usable.

The questions asked regarding the usefulness of the checklist allowed the authors to answer the research question specified in section 4.4.1 and also evaluate whether the checklist is fit for use in software organization or not.

3.3.2.4 Execution:

A total of 15 practitioners involved in continuous delivery agreed to participate in interview based on the condition that their anonymity will be preserved. The interview was conducted over Skype as it was not possible for the interviewees to be physically present as most of them lived in The United States of America and India.

3.3.2.5 Data analysis:

The data obtained from the interviews is analyzed using a qualitative analysis method such as coding. Based on the data obtained by coding the interview transcripts, the data is analyzed using descriptive statistics methods such as frequency distribution, bar graph and pie charts to arrive at a conclusion [8].

4 RESULTS

4.1 Introduction

The success of the project or product depends on the quality of the product delivered to the customer. So to ensure a quality product delivery, the concept of metrics was introduced. It is aptly said that "what cannot be measured cannot be improved [32]". The importance of metrics in the field of software development and the characteristics of good metrics are presented as a part of Literature Review (LR) performed by authors in chapter 2. The Literature Review resulted in categorization of metrics being used in Agile and Continuous Delivery approaches based on the 14 papers reporting 10 studies. Due to high emphasis of authors on empirically-based research, the literature review did not include articles that were not peer-reviewed, technical reports and practitioners opinion presented in forums/blogs. As an extended research, the following survey presents empirical evidence of state of practice on quality metrics in continuous delivery. The survey is an extension to literature review which results in generation of a checklist of metrics which can be used to involve quality metrics in project.

There has been only one previous literature review related to quality metrics being used in continuous delivery. This literature review could only categorize the metrics used at implementation level and pipeline level. Further information regarding the description of previous literature review in context of metrics used in continuous delivery is presented in section 2.4. The authors intended to find the usability and limitations of existing metrics as well as present new metrics that are being used in continuous delivery. Based on the results obtained from the online questionnaire, the authors generated a checklist which was validated using telephonic interviews with practitioners involved in continuous delivery.

The latter part of chapter is structured as follows: Section 4.2 presents the results obtained from the literature review. Section 4.3 presents the results of the online questionnaire. Section 4.4 presents the results of the interviews conducted by the authors to evaluate the usefulness of checklist generated based on the online questionnaire responses.

4.2 Results of literature review

This section presents the results obtained from the literature review regarding the quality metrics used in continuous delivery. Literature review is conducted using backward search as little research has been done regarding continuous delivery. There has been only one empirical research performed over the quality metrics used in continuous delivery. The quality metrics involved in continuous delivery obtained at the end of the literature review is presented in table 4.1 below.

Category	Metric involved in category	Description of metric
Metric on implementation level [29]	Development time	The development time taken by the team to implement a new feature [29].
	Deployment time	The time taken to deploy new features to production use [29].
	Activation time	The time taken by first user to activate new features after deployment [29].

	Oldest Done Feature (ODF)	Time spent by a feature in development done state i.e., waiting for being moved to production state from deployment [29].
Metric on pipeline level [29]	Features per Month (FPM)	Number of features that have moved through the pipeline in a period of one month [29].
	Releases per Month (RPM)	The number of releases made during a period of one month [29].
	Fastest Possible Feature Lead Time	The time spent by a feature in the build and test stages of pipeline [29].

Table 4.1: Quality metrics involved in continuous delivery

4.3 Results of online questionnaire

This section presents the results obtained from the online questionnaire regarding the usability of quality metrics in continuous delivery. Only 49 respondents out of 79 respondents were involved in continuous delivery. The results and data analysis was performed over these 49 responses only to eliminate the bias in the results.

4.3.1 Respondent's Credentials

The authors designed the first part of questionnaire such that the results provide information regarding the background of the respondents in field of software development and also to gain understanding of the respondent's knowledge regarding quality metrics.

4.3.1.1 Job Role

Table 4.2 below represents the various job roles of the respondents and the percentage of each role amongst the total responses. The results provide information regarding the various roles involved with Continuous Delivery other than just development and operations teams. Based on the results, the roles can be divided into three categories i.e., managerial roles, non-managerial roles and other roles.

- 10 of the 49 respondents (20.5%) have **managerial roles** such as project manager, project leader, program manager, team lead, technical lead, software engineer and product owner.
- 19 of the 49 respondents (38.7%) have **non-managerial roles** such as developer, analyst, tester, designer and module lead developer.
- 20 of the 49 respondents (40.81%) have **other roles** such as quality analyst, Business analyst, Information technology analyst, technical analyst, product architect, system consultant and staff consultant.

Role of respondent	Frequency	% of responses
Quality analyst	8	16%
Business analyst	3	6%
IT analyst	1	2%

Analyst	1	2%
Technical analyst	1	2%
Designer	1	2%
Product architect	1	2%
CEO	1	2%
Project manager	2	4%
Project leader	3	6%
Program manager	1	2%
Team lead	1	2%
Technical lead	1	2%
Developer	14	29%
Software engineer	1	2%
Module lead developer	1	2%
Tester	2	4%
System consultant	5	10%
Staff consultant	1	2%
Total	49	100%

Table 4.2: Role of respondent in organization

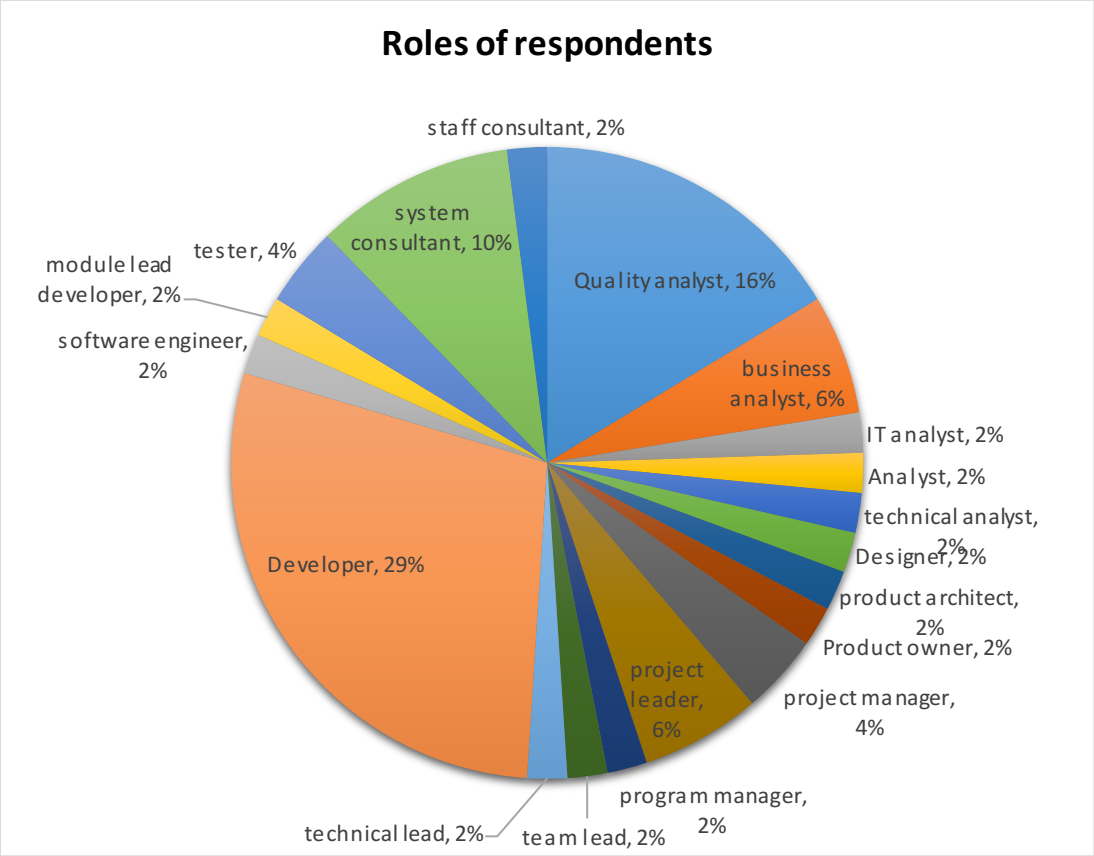

Figure 4.1: Job roles of respondents

Figure 4.1 is a graphical representation of the roles of the respondents in the software organization.

4.3.1.2 Experience in Software Organization

The respondents were presented with a text box to specify their number of years of experience in software organization. To simplify the analysis, the authors used an ordinal scale and categorized the experience level into 5 categories. The data regarding the experience of the respondents is presented in table 4.3 and the graphical representation is presented in figure 4.2 below.

Experience of respondent	Frequency of responses	% of responses
0 to 3 years	7	14%
4 to 7 years	16	33%
8 to 11 years	18	37%
12 to 15 years	7	14%
15+ years	1	2%
Total	49	100%

Table 4.3: Years of experience in software organization

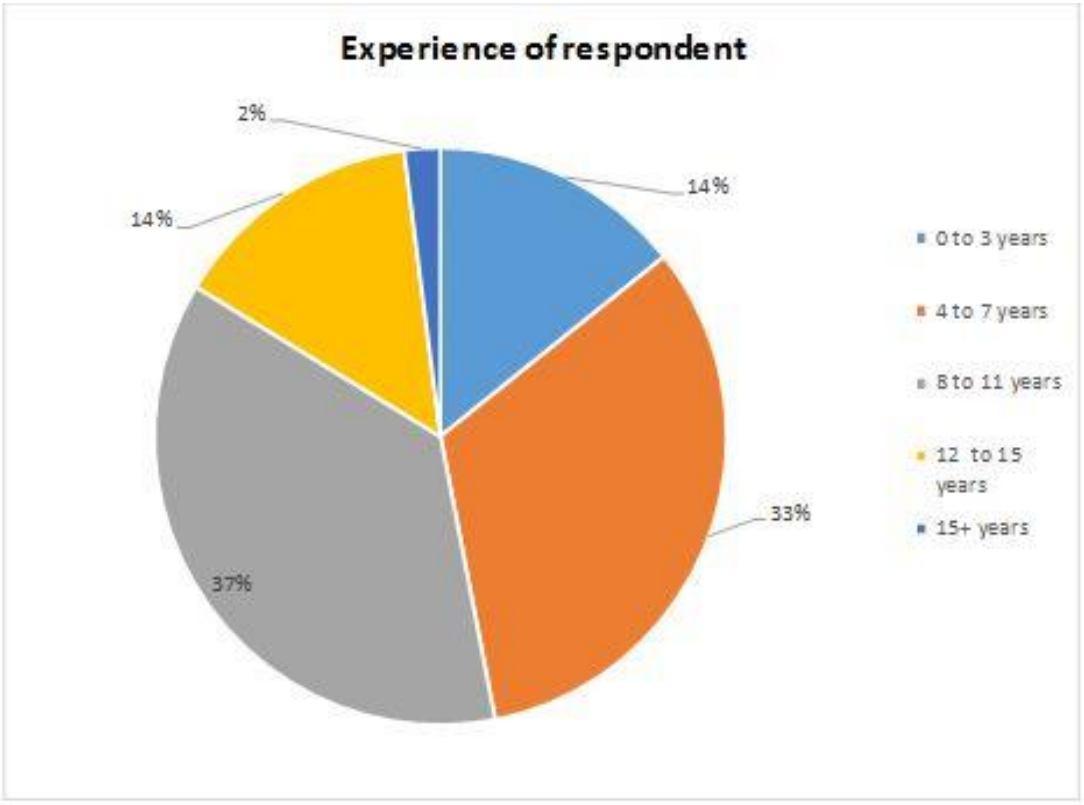

Figure 4.2: Experience level of respondents

4.3.1.3 Development methodology used in Software Organization

The respondents were asked to provide information regarding the development methodology being used in their organization. The respondents were allowed to make one choice among 5

agile approaches i.e., scrum, scrum ban, Feature Driven Development (FDD), Crystal and Extreme Programming (XP). Respondents had the option to specify the approach used in their organization other than the above specified approaches. The data regarding the development methodology used in the respondent's organization is presented in table 4.4 and the graphical representation is presented in figure 4.3 below.

Development approach	Frequency of responses	% of responses
Scrum	39	80%
Scrum-ban	2	4%
FDD	3	6%
Crystal	0	0%
XP	1	2%
Waterfall	3	6%
Hybrid agile	1	2%
Total	49	100%

Table 4.4: Development approach used in software organization

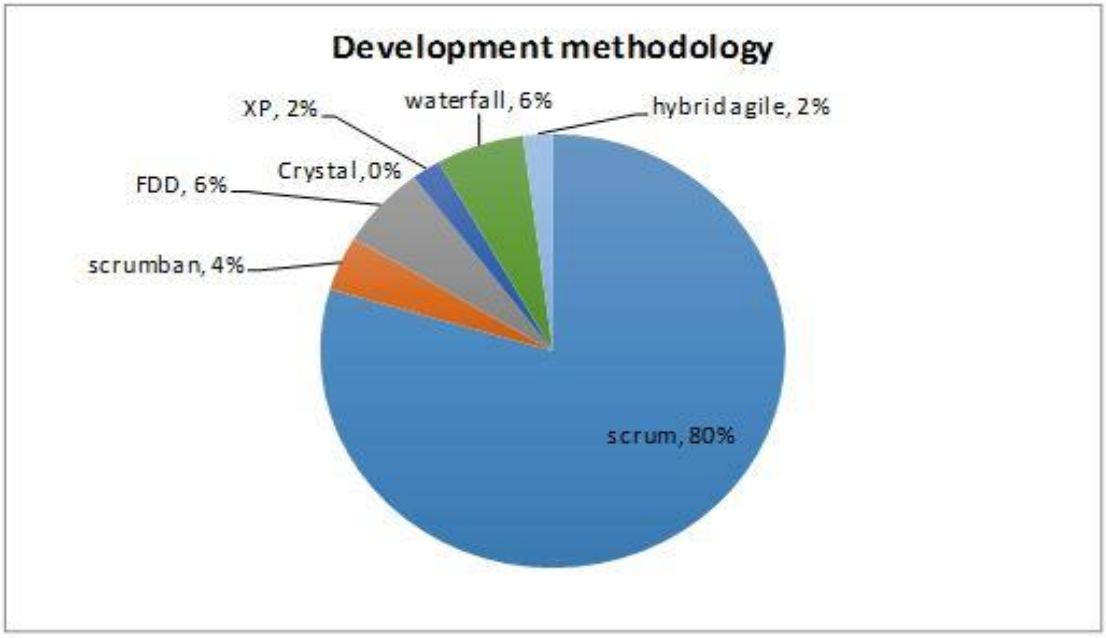

Figure 4.3: Software development methodology used in organization

4.3.1.4 Means of ensuring software quality

The respondents were asked to specify how they ensure quality of product in their organization. They were allowed to multiple choices from 4 options i.e., Quality standards, Quality reviews, Process improvement approaches and usage of measurement and metrics. Respondents had an additional text box as option where they can specify any other means of ensuring quality other than the 4 options. The data regarding the quality assurance method used by respondent's organization is presented in table 4.5 and the graphical representation is presented in figure 4.4 below.

Quality assurance method	Frequency of responses	% of responses
Quality standards	31	63%
Quality reviews	24	49%
Process improvement approaches	7	14%
Measurement and metrics	20	41%
CCB sessions	3	8%

Table 4.5: Means of ensuring software quality

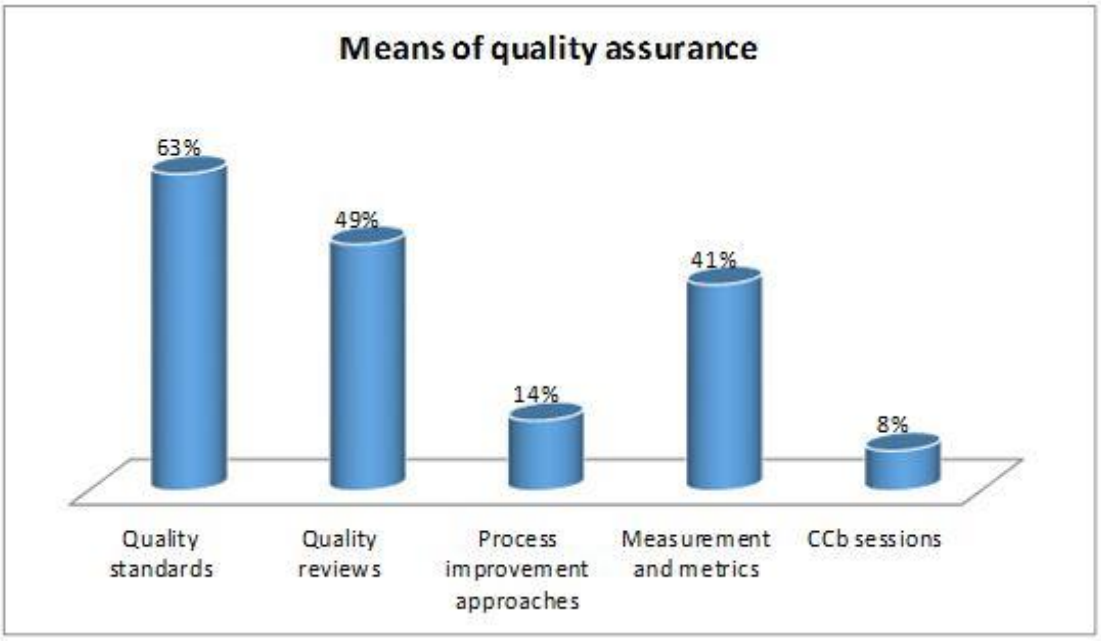

Figure 4.4: Means of Quality Assurance

4.3.1.5 Metrics usage in organization

The respondents were provided with a single choice question to answer whether metrics are used in their organization to ensure software quality. The options presented to the respondents were yes and no. Table 4.6 below presents the data of the frequency of responses who answered either yes or no. The figure 4.5 represents the frequency of respondents that use metrics to ensure quality of product in their organization.

Metrics usage in organization	Frequency of responses	% of responses
Yes	45	91.84%
No	4	8.16%
Total	49	100%

Table 4.6: Metrics usage in organization

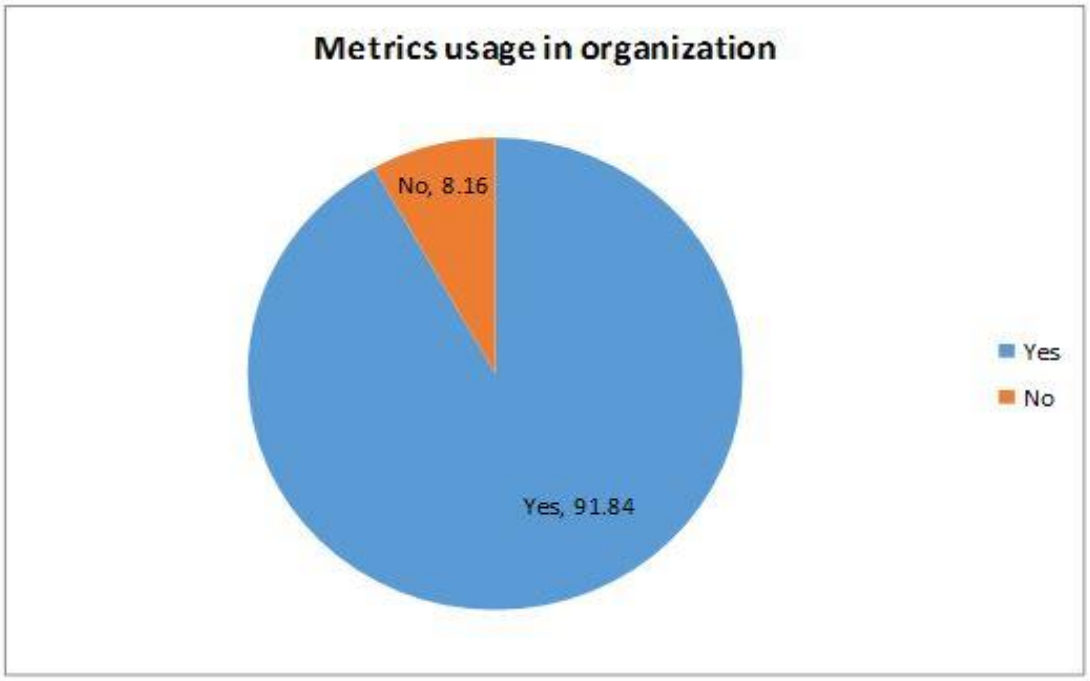

Figure 4.5: Metrics usage in organization

4.3.1.6 Reasons for using metrics in organization

Out of 49 respondents, only 9 (18.4%) respondents specified the reason for using metrics in their organization and 10 (20.4%) respondents were not sure about the reason. The remaining 30 (61.2%) respondents did not answer the question. The respondents were provided with a text box to specify the reason of using metrics in their organization. Table 4.7 presents the frequency of respondents who answered the question and Figure 4.6 is a graphical representation of the data regarding the reasons for using metrics in organization.

Reason for metric usage	Frequency	% of responses
Tracking tasks	4	8.20%
Customer satisfaction	2	4.10%
Delivery of quality product	9	18%
Identify defects and improve	4	8.20%
Not sure of reason	10	20%

Table 4.7: Reasons for metric usage in organization

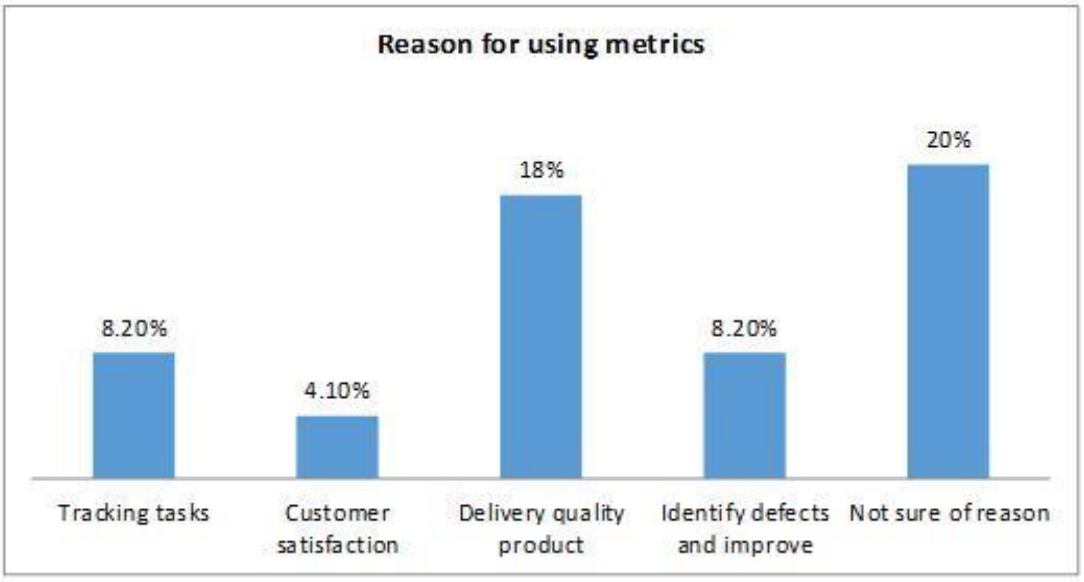

Figure 4.6: Reasons for using metrics

4.3.2 RQ 2.b : Metrics in Continuous Delivery

4.3.2.1 Release frequency of software

The respondents were asked to specify the information regarding the frequency of software release into the market. The respondents were provided with a text box to let them specify the exact release frequency. Based on the responses obtained, the authors presented them into 10 categories. The categories include the release frequency in weeks, months and years. Some of the respondents specified that the release frequency depends on the customer requirements and also on the size of the project. The number of respondents to answer the question in one of the 10 categories are presented in the table 4.8 below and graphical representation is presented in figure 4.7 below. Based on the figure, most of the respondents are involved in releasing the product for every quarter of a year i.e., 3 months.

Release frequency of software	Frequency of responses	% of responses
1 week	4	8%
2 weeks	6	12%
Monthly	5	10%
6 weeks	1	2%
2 months	7	14%
Quarterly	10	20%
3-6 months	1	2%
6 months	7	14%
Yearly	3	6%
Depends	5	10%
Total	49	100%

Table 4.8: Release frequency of software into market

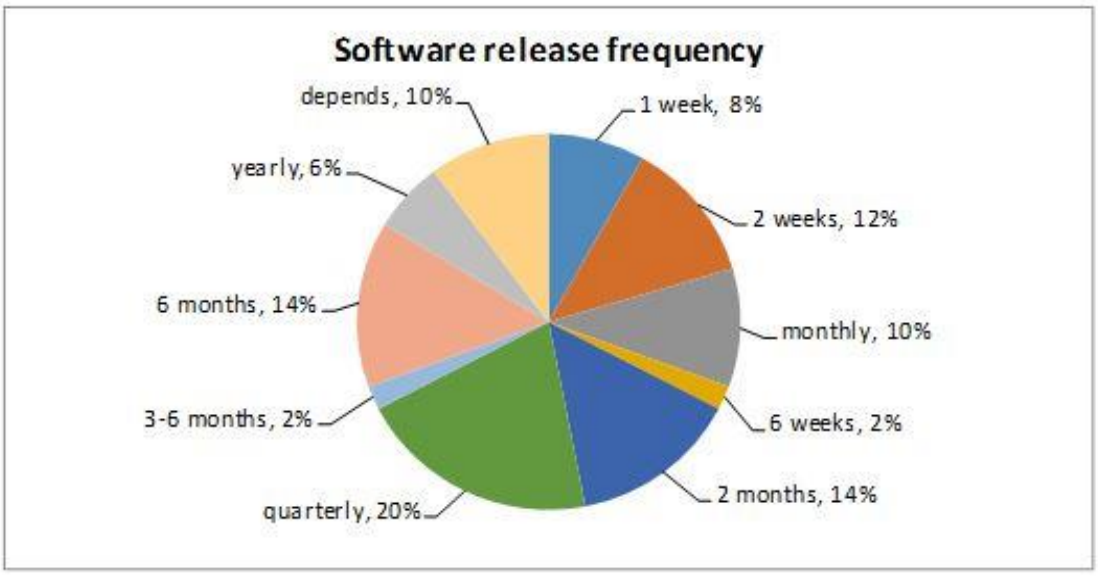

Figure 4.7: Release Frequency of software into market

4.3.2.2 Prior Working experience with Continuous Delivery

The respondents were asked to answer whether they had prior working experience with continuous delivery. The respondents were allowed to make one choice among 2 options i.e., yes/no. The data regarding the response of the 79 respondents is presented in table 4.9. Figure 4.8 below presents the frequency of respondents who answered either yes or no.

Prior working experience with continuous delivery	Frequency of responses	% of responses
Yes	49	62%
No	30	38%
Total	79	100%

Table 4.9: Prior working experience with continuous delivery

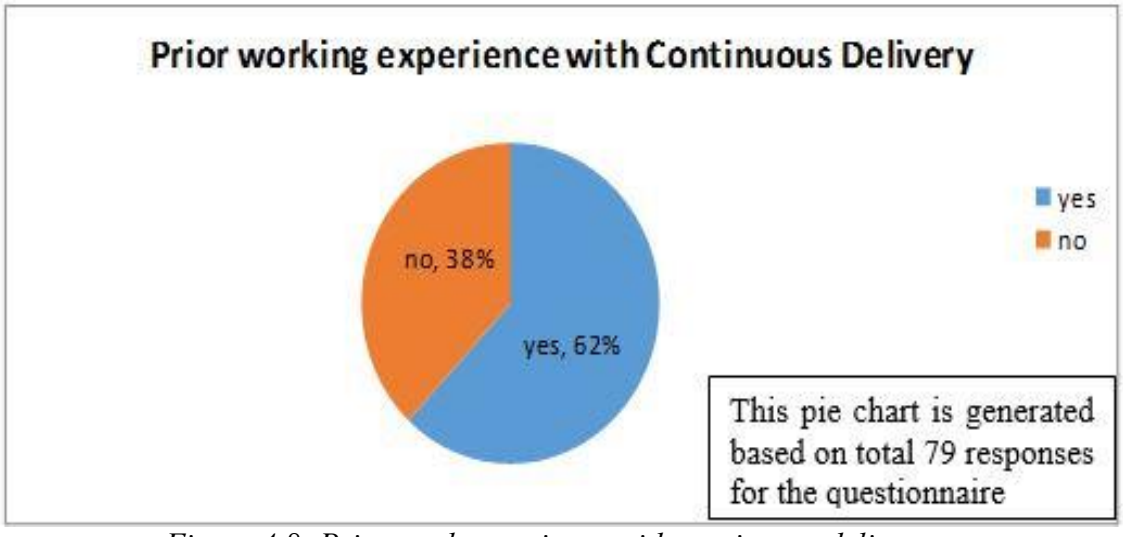

Figure 4.8: Prior work experience with continuous delivery

4.3.2.3 Usability of metrics in continuous delivery

The main aim of the online questionnaire was to collect data regarding the usability of metrics being used in continuous delivery. The usability was calculated by allowing the respondents to rate each metric within a range of 0-100, where 0 is termed least useful and 100 is termed highly useful. The usability was calculated for metrics obtained from Literature review i.e., metrics at implementation level and metrics at pipeline level. To simplify the data analysis, the rating obtained from respondents was divided into five categories having equal interval of 20. The data of the frequency of ratings from 49 respondents is presented in table 4.10 and figure 4.9 below presents the frequency of respondents to rate the metrics. The boxplots generated regarding the ratings given to each category of metrics against the role of respondents is presented in figure 4.10 and 4.11 below.

Interval of rating	**% of respondents to rate the metrics**	
	Metrics at implementation level	**Metrics at pipeline level**
0-20	3%	26%
21-40	22%	27%
41-60	33%	35%
61-80	24%	10%
81-100	18%	2%
Total	100%	100%

Table 4.10: Frequency of responses to rate the category of metrics

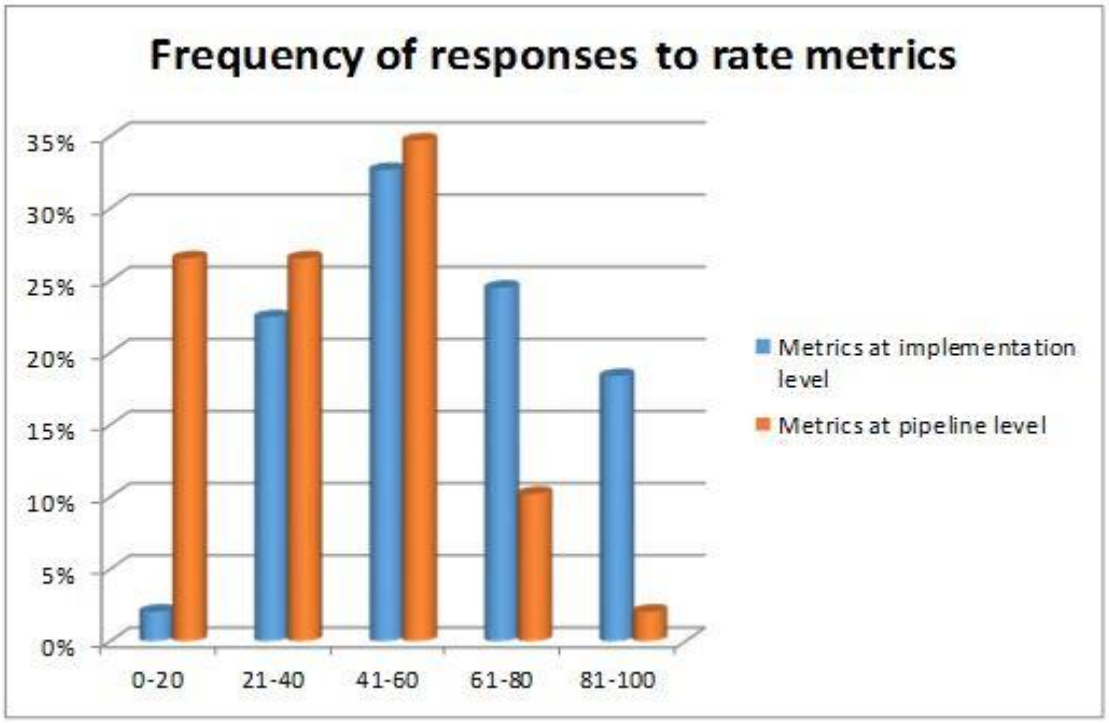

Figure 4.9: Frequency of Ratings given to categories of metrics in CD

Rating given to metrics at implementation level against role of respondents:

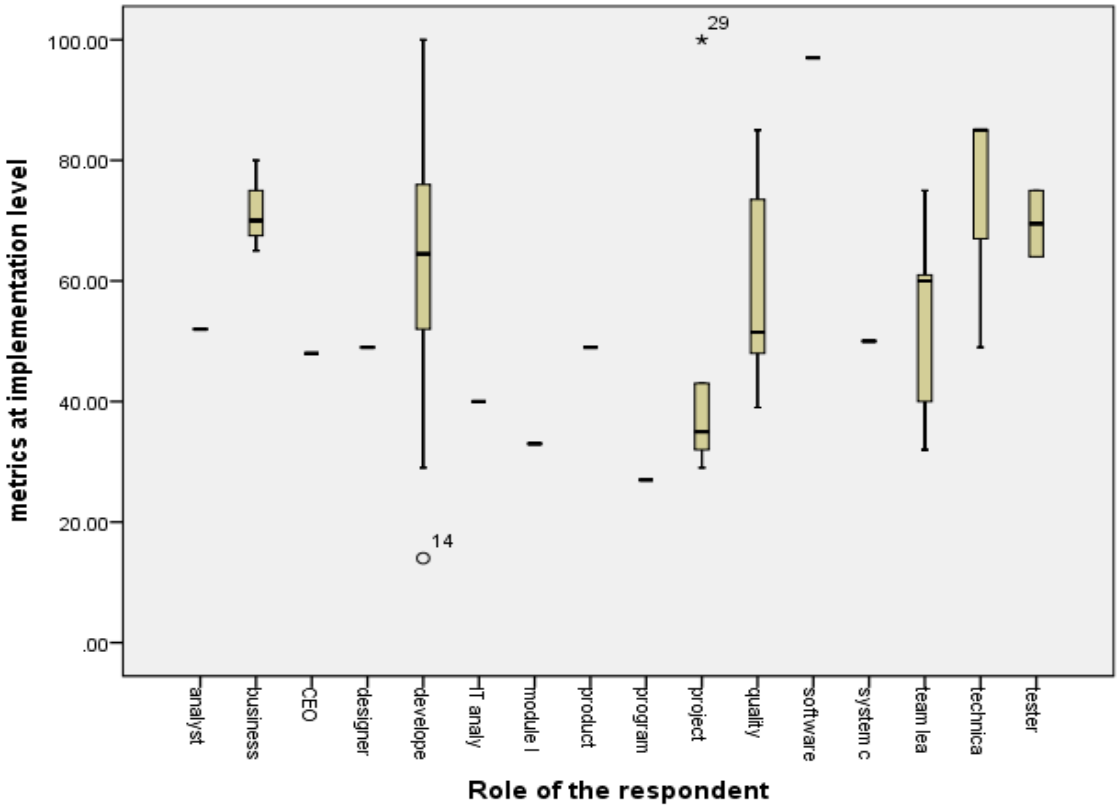

Figure 4.10: Rating given to metrics at implementation level based on role of respondent

Rating given to metrics at pipeline level against role of respondents:

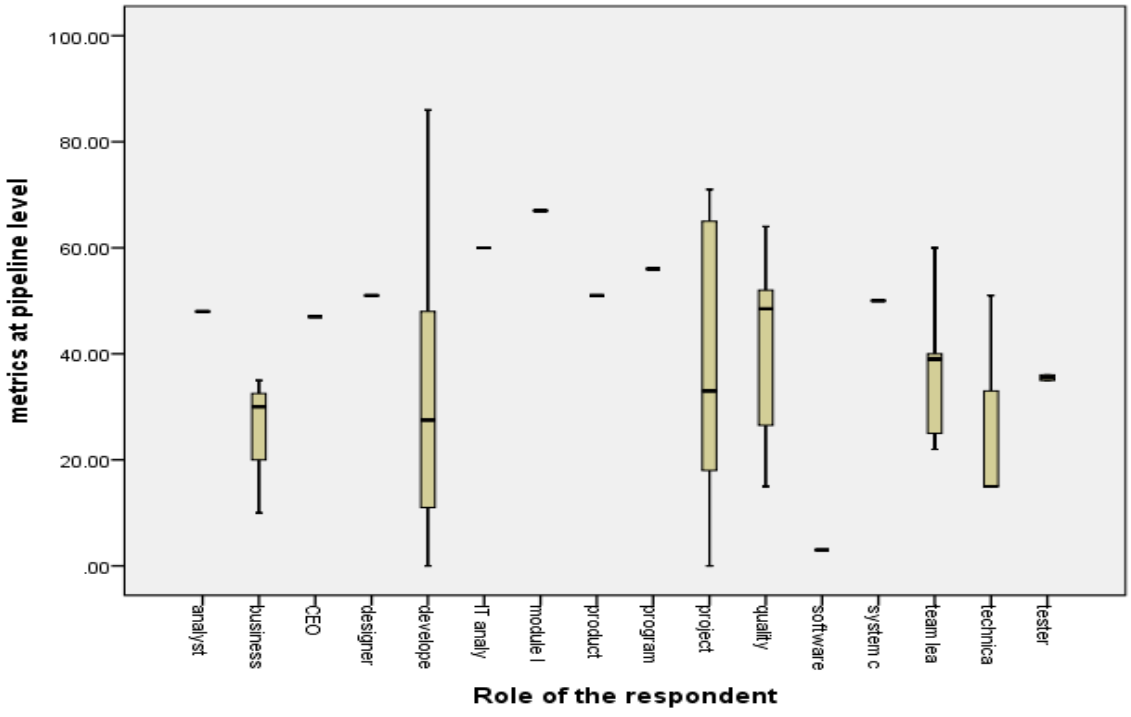

Figure 4.11: Rating given to metrics at pipeline level based on role of respondents

Once the respondents rated the two categories of metrics, they were allowed to rate the sub-category of metrics based on their perception of usefulness. The respondents were allowed to rate each metric on a scale of 0-100. The sub-categories of metrics involved with metrics at implementation level and pipeline level are presented in figure 1 in section 2.2. To simplify the data analysis, the authors categorized the rating into five categories of equal interval. Data regarding the number of respondents to rate the metrics used at implementation level within a specific category is presented in table 4.11 below. Figure 4.12 below presents the frequency of respondents given to each metric involved in category metrics at implementation level.

Interval of rating	Development time	Deployment time	Activation time	Oldest done feature
0-20	8%	12%	20%	27%
21-40	2%	6%	6%	10%
41-60	14%	34%	24%	33%
61-80	35%	24%	32%	16%
81-100	41%	24%	18%	14%

Table 4.11: Frequency of ratings given to metrics used at implementation level

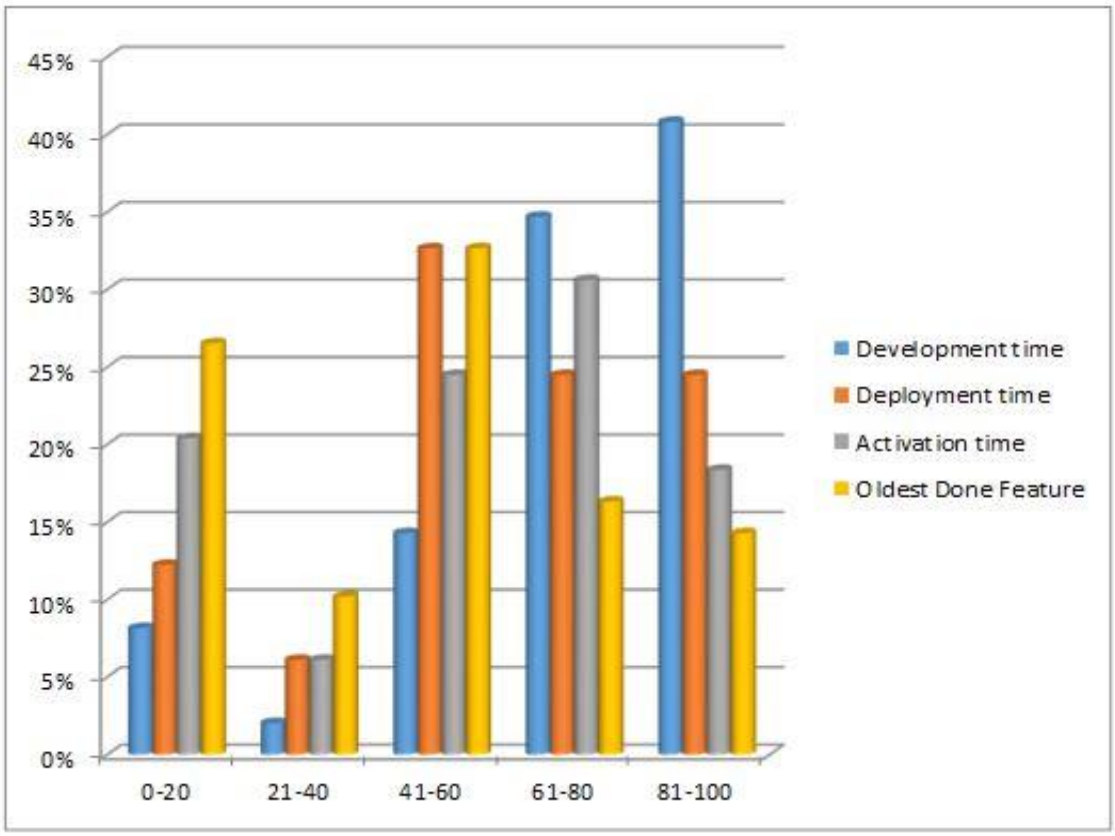

Figure 4.12: Frequency of Ratings given to metrics used at implementation level

Data regarding the number of respondents to rate the metrics used at pipeline level within a specific category is presented in table 4.12 below. Figure 4.13 below

presents the frequency of respondents given to each metric involved in category metrics at Pipeline level.

Interval of rating	Features per month	Releases per month	Fastest possible feature lead time
0-20	22%	14%	20%
21-40	2%	6%	15%
41-60	18%	23%	22%
61-80	39%	37%	29%
81-100	19%	20%	15%

Table 4.12: Frequency of ratings given to metrics used at pipeline level

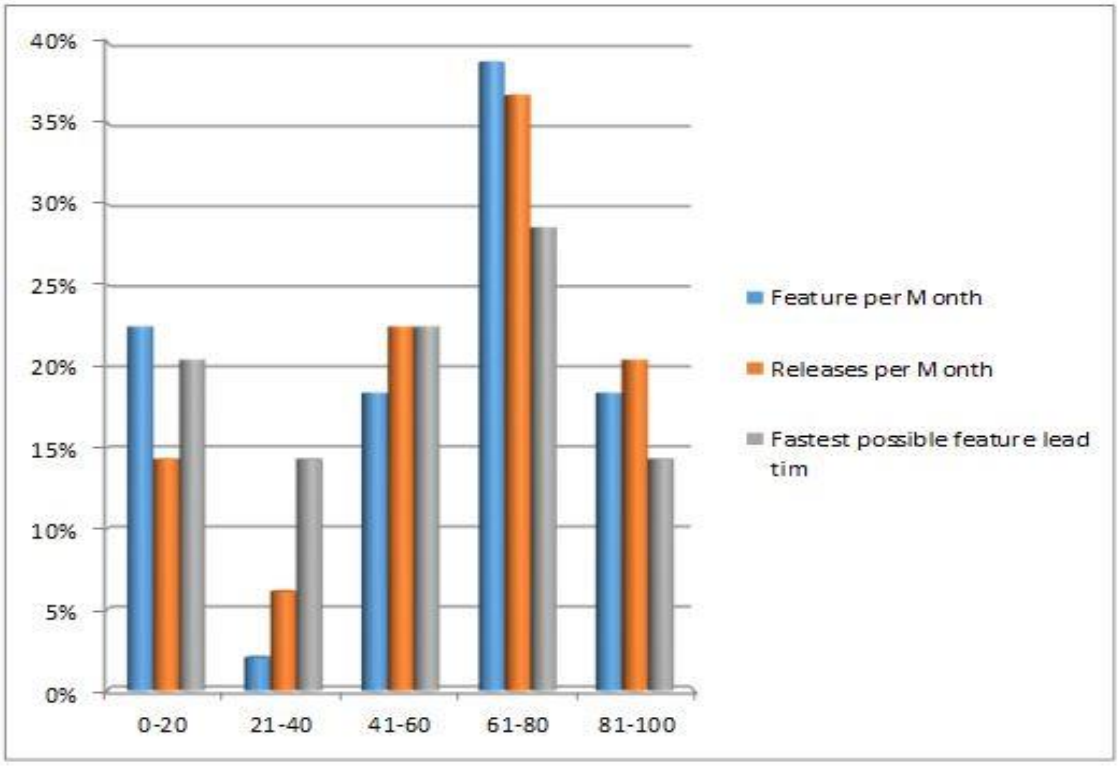

Figure 4.13: Frequency of Ratings given to metrics used at pipeline level

4.3.3 RQ 2.c : Limitations and recommended metrics in Continuous Delivery

4.3.3.1 Limitations of metrics usage in Continuous Delivery:

The authors presented a question where the respondents could specify any limitations that they observe in their organization while using metrics in their organizations. It was an open ended question as the authors intended to collect the respondent's perception. A text box was presented to the respondents to answer this question. The data of the number of respondents who answered this question is presented in table 4.13 and the graphical representation of the data is presented in figure 4.14.

Specifying limitations	Frequency of responses	% of responses
Specified limitations	20	40.80%
Not sure of limitations	2	4.1%
Did not answer the question	27	55.10%

Table 4.13: Frequency of respondents to specify limitations

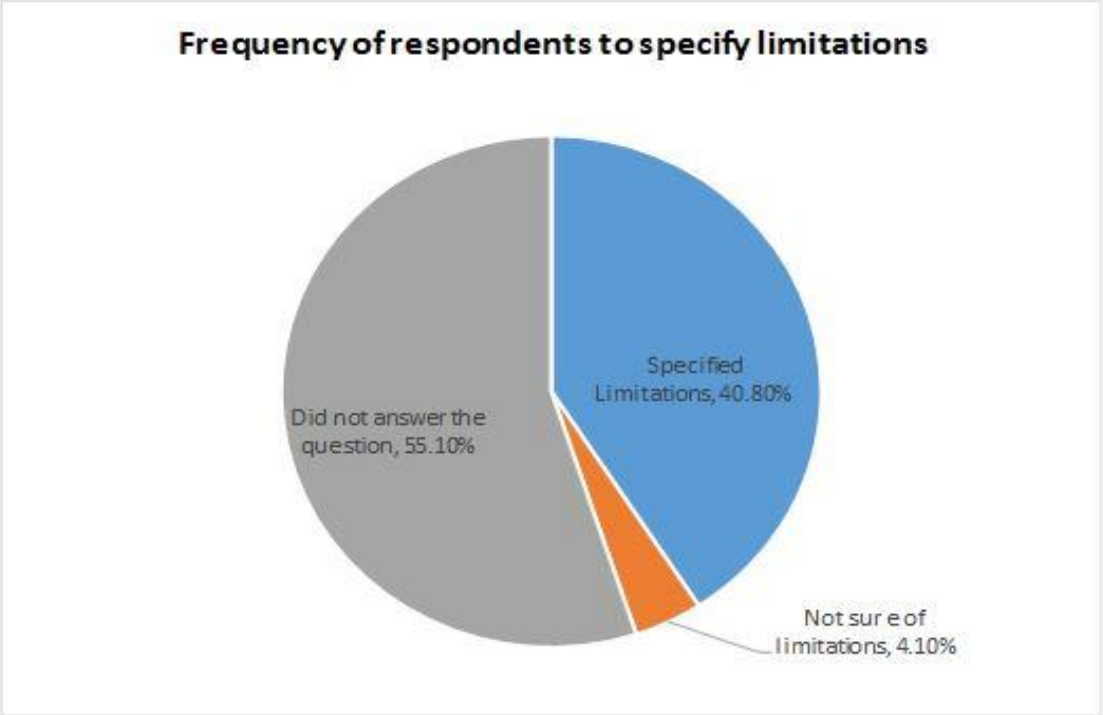

Figure 4.14: Frequency of respondents to specify limitations

Out of the 20 respondents who specified the limitations, the authors categorized the limitations into 9 categories. The data related to the number of respondents to specify limitation of each category is presented in table 4.14. The graphical representation of the data is presented in figure 4.15.

Limitations of metric usage	Frequency of responses
Depends on project	2
Deviation from actual dependency	2
Ignoring alternatives for defects	4
Numbers are of no value if they are not used	2
Natural biorhythm	1
Organization maturity	1
Inaccurate metric data collection	2
Useful under stable conditions	2
Not a generic limitation	2

Table 4.14: Number of respondents to specify the limitations

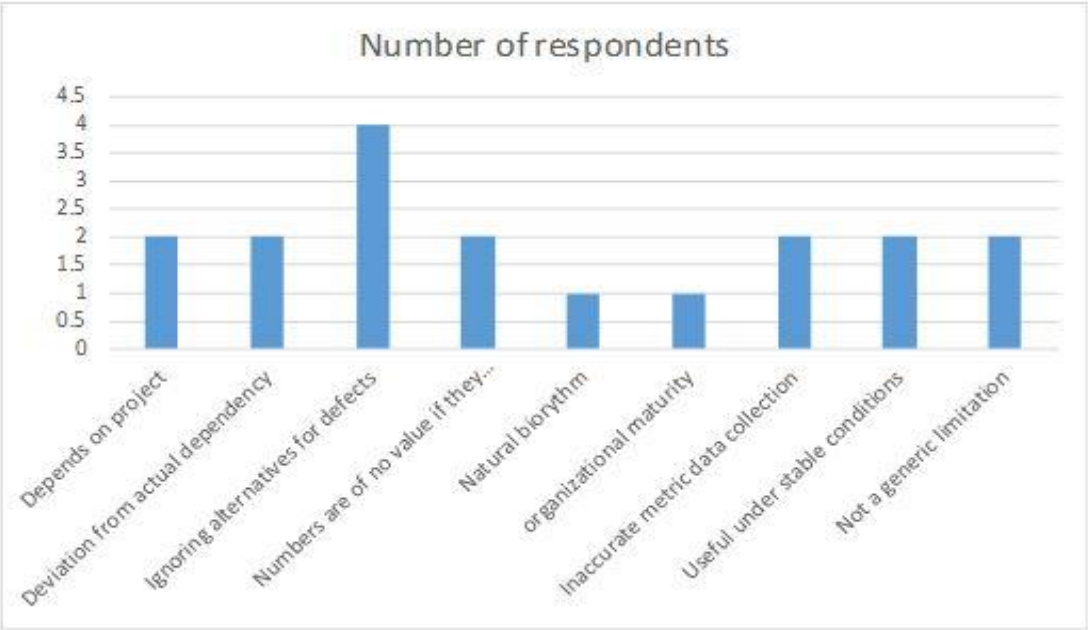

Figure 4.15: Number of respondents to specify the limitations

4.3.3.2 Recommended metrics in Continuous Delivery:

As mentioned earlier, the authors used only 49 out of 79 responses based on the constraint that the respondent had prior experience of working with continuous delivery. Out of these 49 respondents, only 10 respondents provided additional metrics that can be used to ensure quality. Out of remaining 39 respondents 7 respondents did not suggest any other metrics and remaining respondents did not answer the question. The number of respondents to recommend additional metrics are presented in figure 4.16 below. The role of respondents who recommended additional metrics are presented in figure 4.17 below.

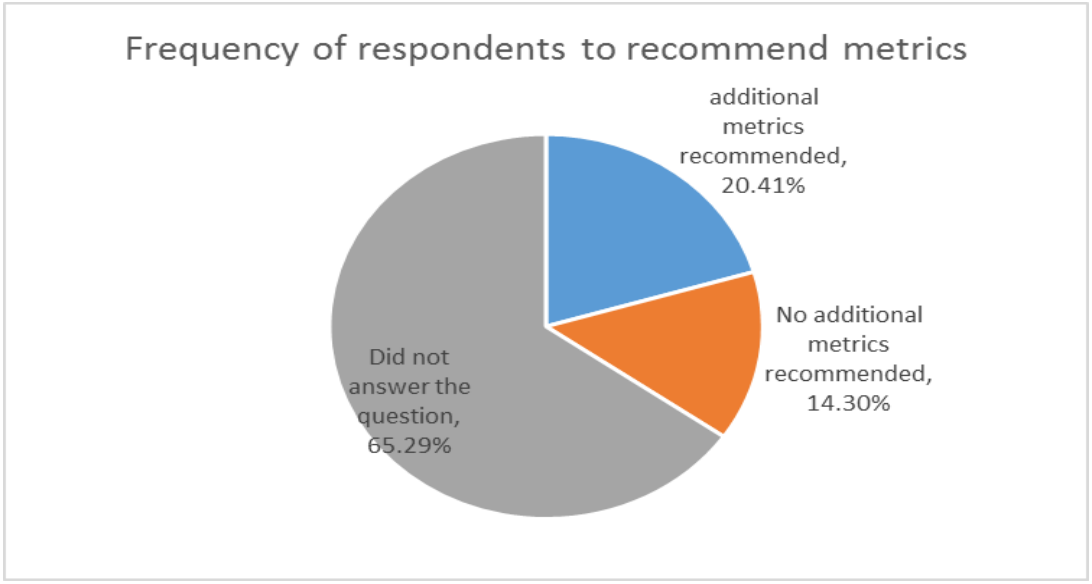

Figure 4.16: Number of respondents to recommend additional metrics

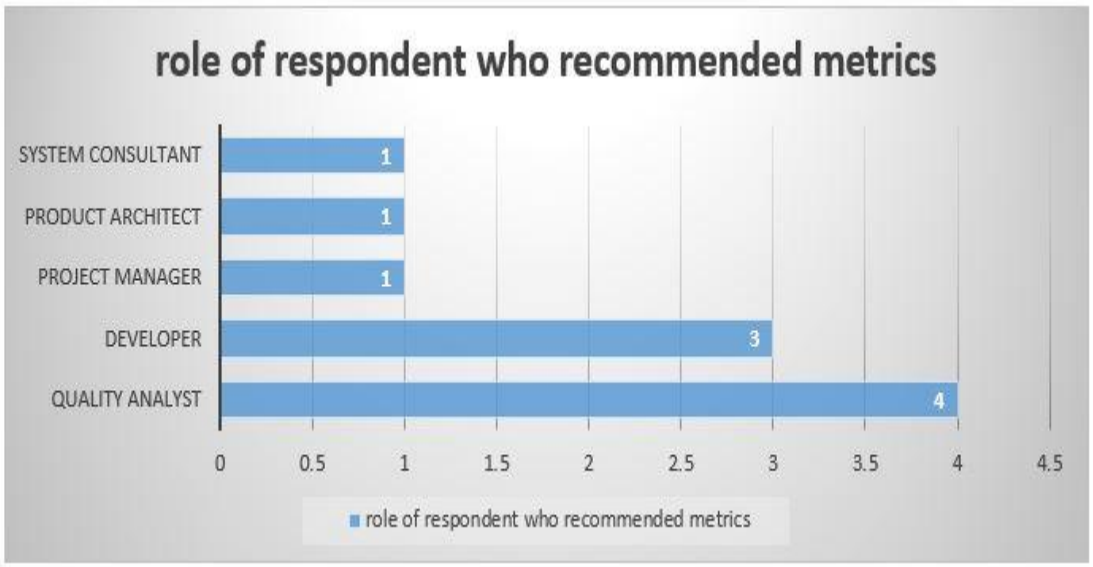

Figure 4.17: Role of respondents to recommend metrics

Metrics Recommended by Quality Analyst:

Based on Figure 4.17 above, 4 respondents were quality analysts who recommended additional metrics that can be used in continuous delivery to ensure quality. The metrics suggested are Key Performance Index (KPI), number of defects reported, and development throughput.

Metrics recommended by Developer:

Based on Figure 4.17 above, 3 respondents are developers who recommended metrics. The metrics recommended by the respondents are code scalability and defect removal efficiency.

Metrics recommended by Product architect:

Based on Figure 4.17 above, 1 respondent is product architect to recommend metrics that can be used to ensure quality in continuous delivery. The recommended metrics are mean time to repair broken builds and check-in time to go live in production.

Metrics recommended by project manager:

Based on Figure 4.17 above, 1 respondent is project manager. The metrics recommended by the project manager are communication effectiveness, customer relationship, employee satisfaction, distraction index, trust index, aggravation index, supplier/partner index, project management index and intellectual capital.

Metrics recommended by system consultant:

Based on Figure 4.17 above, 1 respondent is system consultant. The metrics recommended by the system consultant are Mean time to Defect (MTTD), Mean time to Repair (MTTR), defect removal efficiency, customer reported issues and Mean time before Failure (MTBF).

4.3.4 Checklist of quality metrics involved in continuous delivery

Based on the results obtained from the online questionnaire regarding the usability of the metrics involved in continuous delivery, the authors generated a checklist of metrics which act as remainder to the practitioners regarding the various aspects to be included in order to ensure product quality. The checklist of metrics is presented in figure 4.18, 4.19 below.

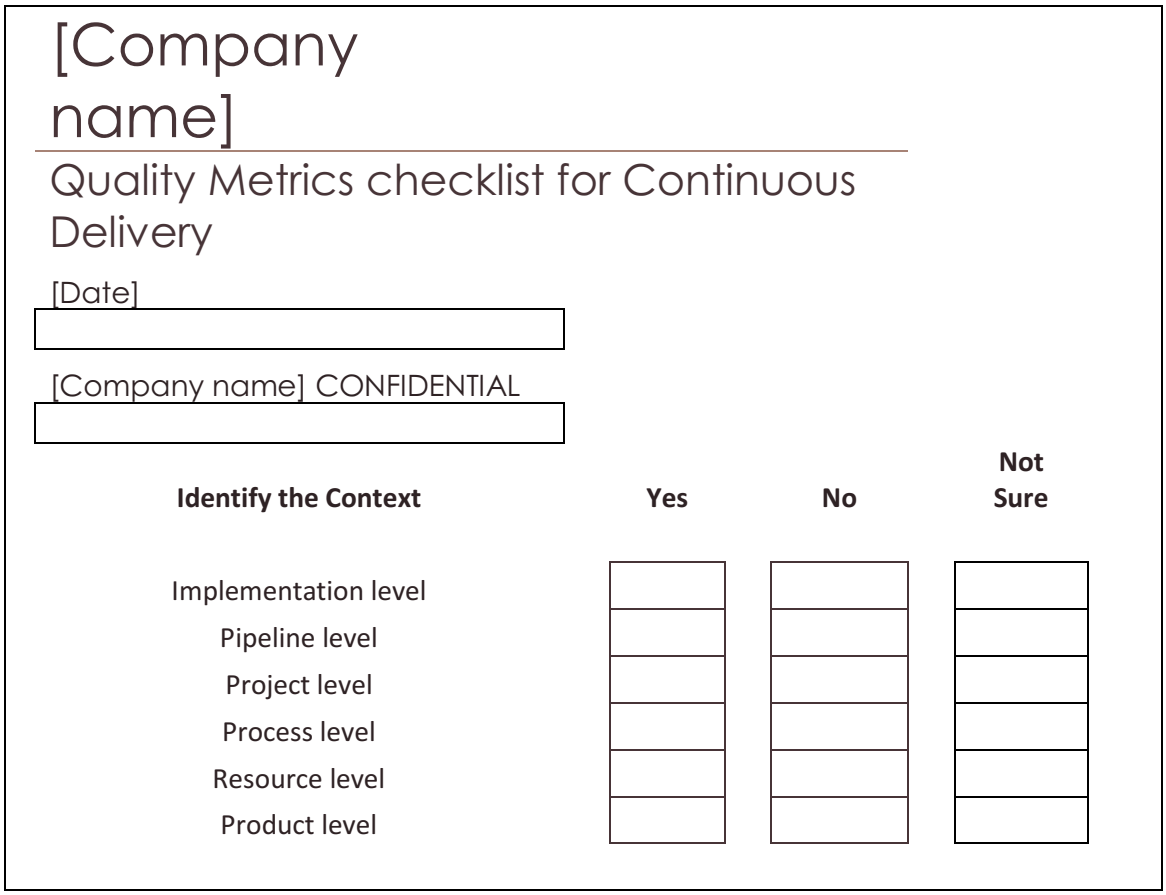

Figure 4.18: Checklist to identify the context of metric usage

[Company name]

Quality Metrics checklist for Continuous Delivery
[Date]

JOB DESCRIPTION:

Metrics checklist	Yes	No	Not sure
Metrics at implementation level			
Development time			
Deployment time			
Activation time			
Oldest done feature			
Metrics at pipeline level			
Releases per Month (RPM)			
Features per Month (FPM)			
Fastest possible feature lead time			
Product level metrics			
Number of defects			
Code scalability index			
Customer reported issues			
Resource level metrics			
Communication effectiveness			
Customer relationship			
Employee satisfaction			
Distraction index			
Trust index			
Aggravation index			
Supplier/partner index			
Process level metrics			
MTTD			
MTTR			
Defect removal efficiency			
Mean time before Failure (MTBF)			
Development throughput			
Check-in time to go live in production			
Project level metrics			
Intellectual capital			
Key Performance Index (KPI)			
Project Management index			

Figure 4.19: Checklist of metrics involved in each context

4.4 Results of telephonic interview

4.4.1 Interviewee's demographics:

The authors designed the first part of interview such that the results provide information regarding the background of the interviewee in field of software development. The results of the questions asked to regarding the interviewee's demographics are presented below.

4.4.1.1 Role of interviewee:

The authors as a warm-up for interview, started off with some light open-ended questions. The first question asked was related to the interviewee's role in the organization. Table 4.15 presents results obtained from the interviewees to this question. Figure 4.20 is a graphical representation of the role of the interviewees.

Role of respondent	Frequency of responses
project manager	4
developer	5
technical lead	3
technical manager	1
quality analyst	1
program manager	1
Total	15

Table 4.15: Role of the interviewee

Figure 4.20: Role of the interviewee

4.4.1.2 Experience of the interviewee in organization:

The respondents were asked to specify the number of years of experience in the software industry. To simplify the analysis, the authors used an ordinal scale and categorized the experience level into 5 categories. The data regarding the experience of the respondents is presented in table 4.16 and the graphical representation is presented in figure 4.21 below.

Experience of respondent	Frequency of responses
0 to 3 years	0
4 to 7 years	4
8 to 11 years	8
12 to 15 years	3
15+ years	0
Total	15

Table 4.16: Experience of interviewees in software industry

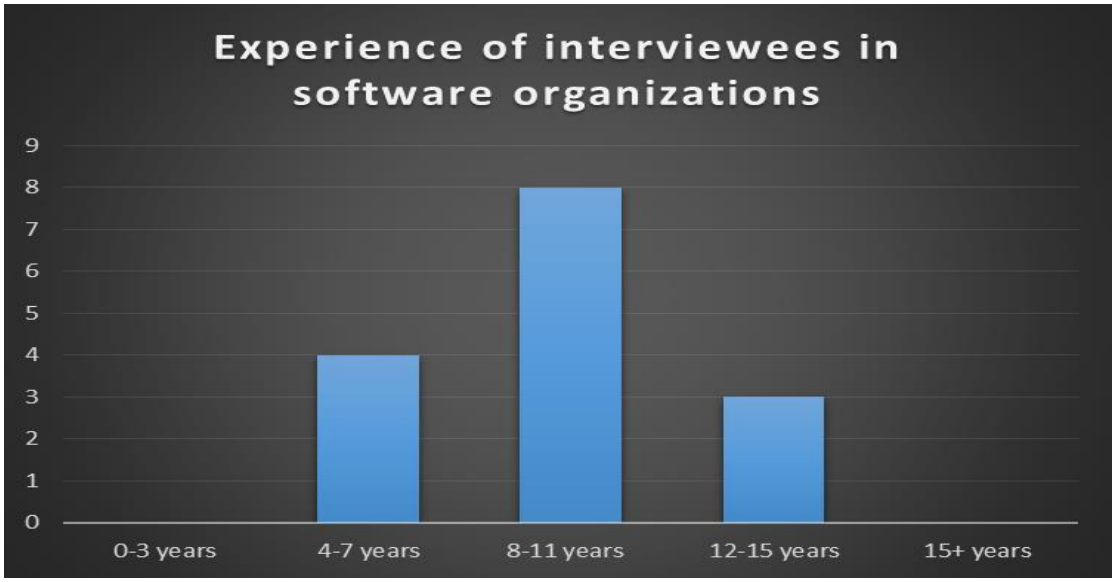

Figure 4.21: Experience of interviewee in software organization

4.4.1.3 Development approach used in organization:

The respondents were asked to provide information regarding the development methodology being used in their organization. Respondents had the option to specify the approach used in their organization as it was an open ended question. The data regarding the development methodology used in the respondent's organization is presented in table 4.17 and the graphical representation is presented in figure 4.22 below.

Development approach	Frequency
Scrum	12
Hybrid agile	2
Kanban	1

Table 4.17: Development approach used in organization

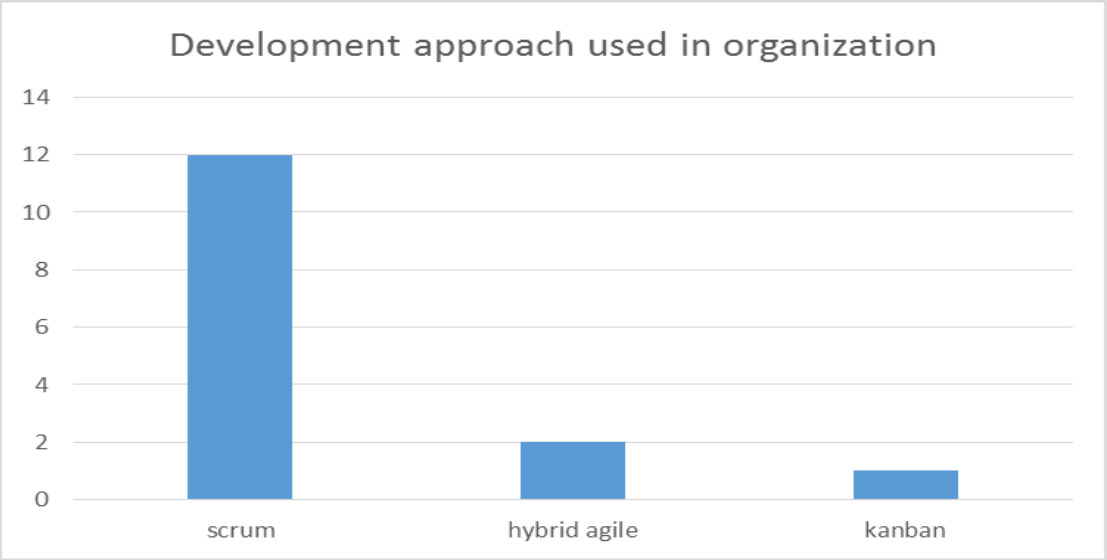

Figure 4.22: Development approach used in organization

4.4.1.4 Metric usage in organization:

The respondents were provided with a close ended question to specify whether their organization involves the use of metrics to ensure quality. The options presented to the respondents were yes and no. Table 4.18 below presents the data of the frequency of responses who answered either yes or no. The figure 4.23 represents the frequency of respondents that use metrics to ensure quality of product in their organization.

Metrics usage	Frequency of responses
Yes	12
No	3
Total	15

Table 4.18: Metric usage in organization

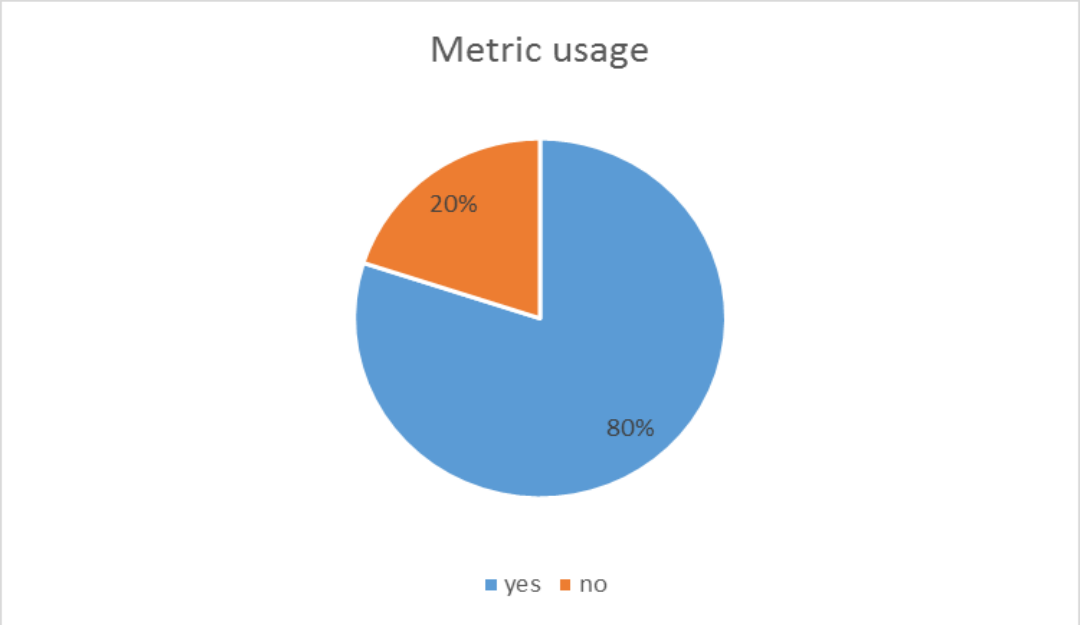

Figure 4.23: Metric usage in organization

4.4.1.5 Usage of Continuous delivery in organization

The respondents were asked to answer whether continuous delivery is used in their organization or not. The questions asked was an open ended questions. The data regarding the response of the interviewees is presented in table 4.19. Figure 4.24 below presents the frequency of respondents who answered either yes or no.

Usage of continuous delivery	Frequency
Yes	14
No	1

Table 4.19: Usage of continuous delivery in organization

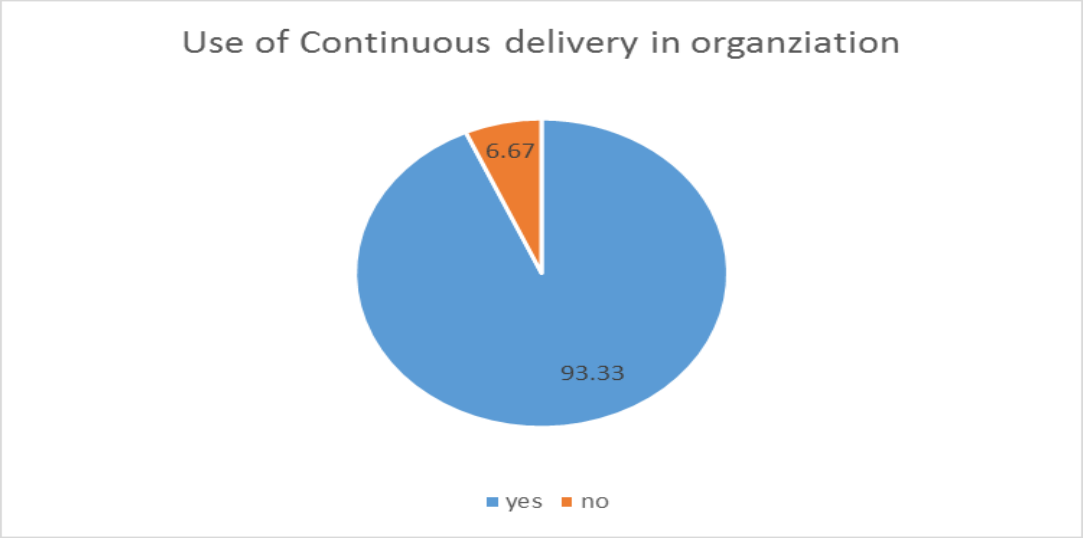

Figure 4.24:Use of continuous delivery in organization

4.4.1.6 Release frequency of product into market:

The respondents were asked to specify the information regarding the frequency of software release into the market. Based on the responses obtained, the authors presented them into 6 categories. The categories include the release frequency in multiples of weeks and months. Some of the respondents specified that the release frequency depends on the customer requirements and also on the size of the project. The number of respondents to answer the question in one of the 6 categories are presented in the table 4.20 below and graphical representation is presented in figure 4.25 below.

Release frequency	Frequency of responses
2 weeks	1
2 months	2
3 months	3
6 months	2
6-8 months	2
Depends on customer	5

Table 4.20: Release frequecy of product into market

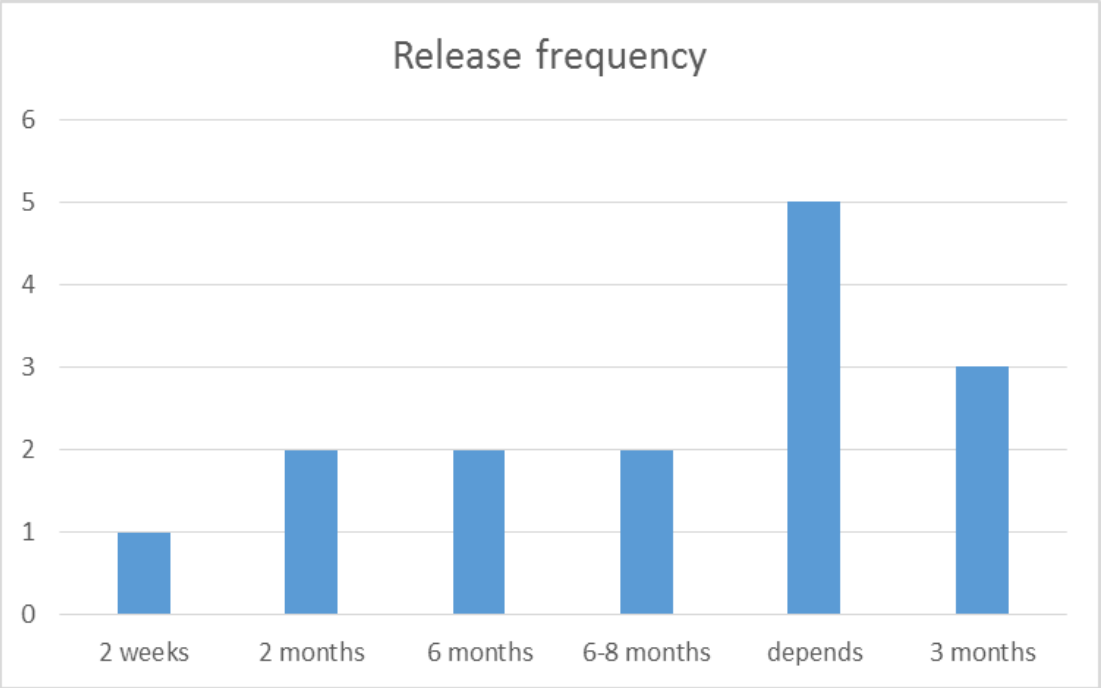

Figure 4.25: Release frequecy of product into market

4.4.2 Checklist Evaluation

The authors designed the second part of interview such that the results provide information whether the checklist generated is fit for use in organization or not. The results of the questions asked regarding the interviewee's perception of the complexity of checklist and the challenges in understanding the checklist are presented below.

4.4.2.1 Rating given to the complexity in understanding the metric checklist

As the main objective of conducting the interviews was to validate the checklist of metrics generated from online questionnaire, the authors asked the interviewees to rate the complexity in understanding the checklist on a scale of 1-10. 1 is termed as least complex and 10 is termed as highly complex. The data regarding the rating and the frequency of responses for each rating is presented in table 4.21 below. A graphical representation of data in form of bar graph is presented in figure 4.26 below.

Rating from 1-10	Frequency of responses
2	2
3	2
4	4
5	5
6	1
7	1

Table 4.21: Frequency of ratings given to level of complexity

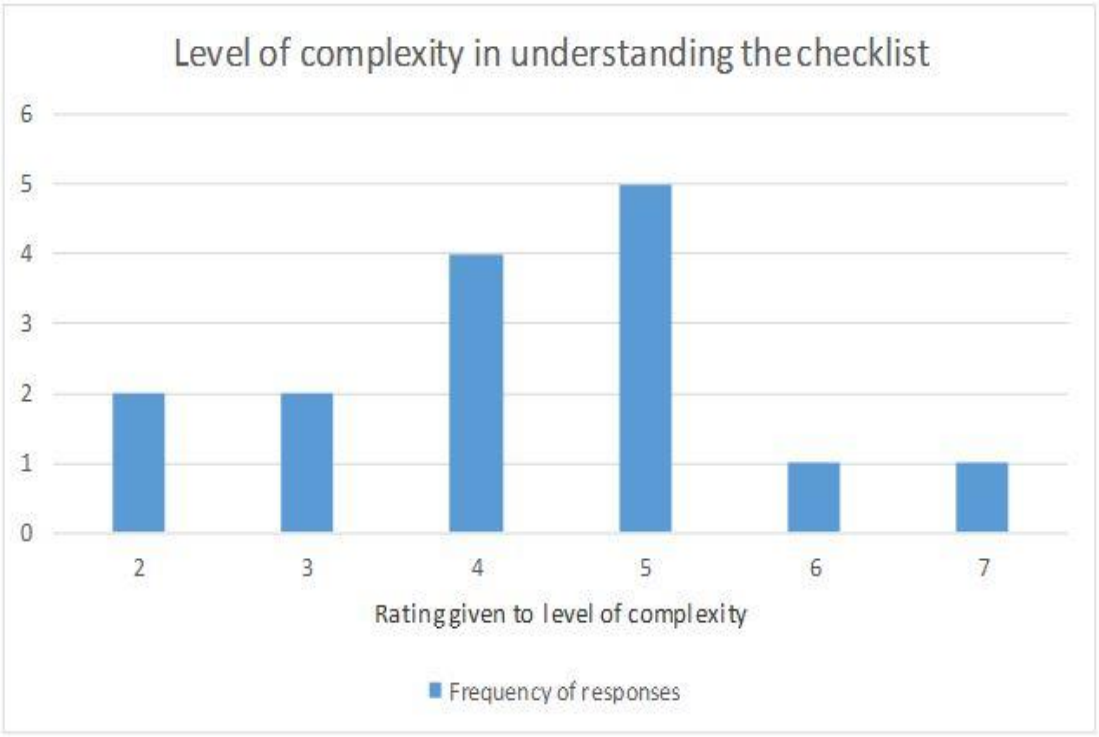

Figure 4.26: Frequency of ratings given to level of complexity

4.4.2.2 Challenges in understanding the checklist

The authors in order to make the checklist more usable, included the open ended question regarding the challenges faced by the interviewees in understanding the metrics checklist. After coding the interview transcripts, the authors categorized the challenges into 6 categories. The data presented by the interviewees falls in one of these categories. The authors distinguished the challenges based on the roles of the interviewees into either one of the categories. To simplify the analysis, the authors categorized the roles of interviewees into managerial and non-managerial roles. The responses obtained from the interviewees possessing managerial role regarding the challenges is presented in table 4.22 below. The data obtained from the interviewees possessing non-managerial roles regarding the challenges is presented in table 4.23 below. Some of the challenges faced by the interviewees is presented below.

- The checklist is slightly off the line
- The metrics specified need to be more clear and precise
- Abbreviate some metrics in checklist
- Unfamiliarity with some of the metrics in checklist
- Use of different development approach makes it difficult for checklist understandable

Table 4.22: Feedback from interviewees with managerial roles

	Slightly off the line	**Metrics need to be crisp and precise**	**Easy to understand**	**Abbreviate some metrics**	**Unfamiliarity with the metrics**	**Use of different development approach**
Managerial role	**S1:** "this is slightly off the line. You might have to improve the quality of the checklist I should say!"	**S3:** "Some of the metrics must be more precise. Then it will be good. Some things such as MTTD. We may use it, but we might not remember the acronymn at that time"	**S7:** "Most of the metrics specified are software terms and thus easy to understand" **S8:"**I don't have any problem in understanding the checklist"	**S5:**"I don't find anything challenging, but it would be good if you could abbreviate MTTD and MTTR"	**S9:**"Like I said earlier, aggregation index I am listening for the first time" **S12:** "frankly saying, some of the metrics I don't know what they mean. I will agree that aggreation index and some more terms I don't know!" **S13:** "I have no idea what MTTD and MTTR mean" **S14:** "You see, Activation time probably is the term i have heard for the first time!"	--

Table 4.23: Feedback from interviewees with non-managerial roles

	Slightly off the line	**Metrics need to be crisp and precise**	**Easy to understand**	**Abbreviate some metrics**	**Unfamiliarity with the metrics**	**Use of different development approach**
Non-managerial role	--	**S2:** "the challenges that I face are regarding the releases per month. They are straightforward and pretty challenging. In product life cycle they are actually fluctuating and is one of the challenging question"	**S4:** "No! I mean everything are software terms, kind of fine for me!" **S6:** "I don't have any problem in understanding the checklist!" **S10:** "I don't find anything difficult to understand"	**S11:** "I see some things are not abbreviated. Something like in process level metrics like MTTD. Initially, I did not get that. Maybe a short introduction will help"	--	**S15:** "Not familiar with most of the terms as they are not used in our organization. The reason maybe because our organization uses a different development approach

4.4.2.3 Checklist usage in the organization

The authors involved the question whether the interviewees would recommend the use of this checklist in their organization to ensure quality. The data obtained from the interviewees is presented in the table 4.24 below. The bar chart representing the data is presented in figure 4.27 below.

Checklist usage in organization	Frequency of responses
Yes	10
No	2
Yes, if changed slightly	3
Not sure	0

Table 4.24: Checklist usage in organization

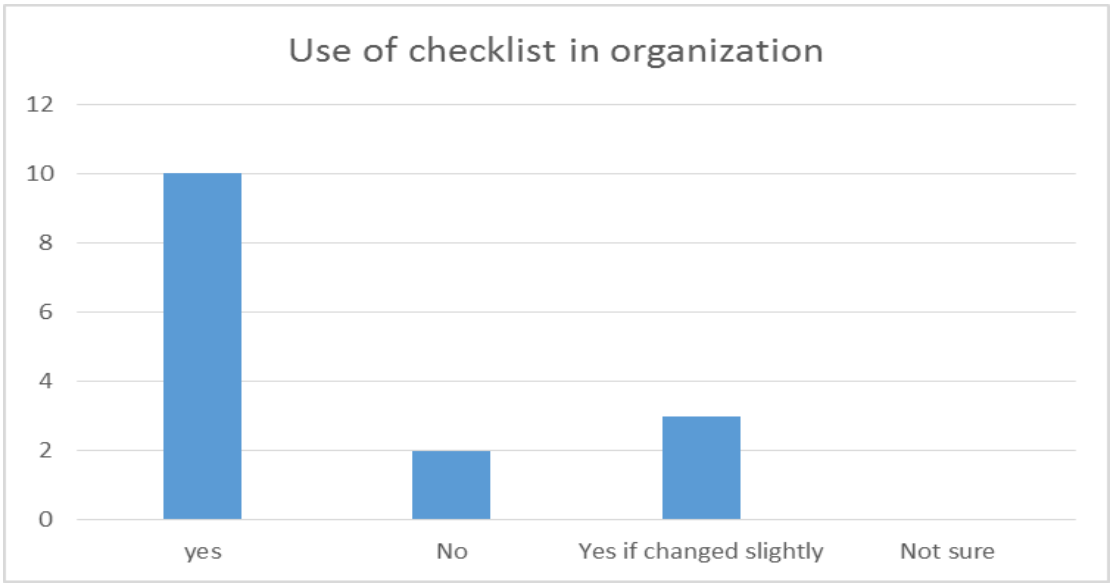

Figure 4.27: Checklist usage in organization

5 ANALYSIS AND DISCUSSION

This section presents a detailed description of the data analysis performed over the data collected from the online questionnaires and telephonic interviews.

The remainder of this chapter is structured as follows: section 5.1 presents a description of data analysis performed over data collected from online questionnaire. Section 5.2 presents a description of the data analysis performed over data collected from telephonic interviews.

5.1 Analysis of data collected from online questionnaire

5.1.1 Respondents credentials

5.1.1.1 Roles of respondents

The analysis is performed to gain a understanding of the various roles of software practitioners who are involved in continuous delivery. If you consider the table 4.2,

- 10 out of 49 respondents i.e., 20.5% of respondents playing managerial roles such as project manager, program manager, project leader, team lead, technical lead, product owner and software engineer are involved in continuous delivery.
- 19 out of 49 respondents i.e., 38.7% having non-managerial roles such as analyst, developer, tester, designer and module lead developer are involved in continuous delivery.
- 20 out of 49 respondents i.e., 40.81% respondents having other roles such as quality analyst, business analyst, information technology analyst, technical analyst, product architect, system consultant and staff consultant are involved in continuous delivery.

5.1.1.2 Experience of respondents in software industry

The analysis is performed to check if the experience of the respondents has any influence over the perception of usefulness of metrics involved in continuous delivery. For ease of analysis, the authors categorized the experience level into 5 categories. If you consider the table 4.3 and figure 4.2,

- 7 out of 49 respondents i.e., 14% of respondents possess experience in range of 0-3 years in the field of software industry.
- 16 out of 49 respondents i.e., 33% of respondents possess experience in range of 4-7 years in the field of software industry.
- 18 out of 49 respondents i.e., 37% of respondents possess experience in range of 8-11 years in the field of software industry.
- 7 out of 49 respondents i.e., 33% of respondents possess experience in range of 12-15 years in the field of software industry.
- 1 out of 49 respondents i.e., 2% of respondents possess experience in range of 15+ years in the field of software industry.

5.1.1.3 Development approach being used in organization

The analysis of the development approach being used by the respondent's organization is performed to check which of the development approaches allow the use of continuous delivery approach. Consider table 4.4 and figure 4.3, 39 out of 49 respondents (80% of respondents) use scrum as a development approach in their organization. Other approaches such as scrum-ban, Feature Driven Development, Crystal, Extreme Programming, waterfall and hybrid agile are used by 10 out of 49 respondents (20% of the respondents).

5.1.1.4 Quality assurance methods used in organization

The analysis of various means of ensuring quality in the organization is performed to allow the authors to have an understanding of how product quality is assured in organization. Consider table 4.5 and figure 4.4,

- 31 out of 49 respondents i.e., 63% of the respondent's organization involve the use of quality standards such as CMMI, ISO 9001:2013 and many other to ensure product quality.
- 24 out of 49 respondents i.e., 49% of respondent's organization involve the use of quality reviews such as quality audits to ensure quality of product.
- 20 out of 49 respondents i.e., 41% of respondent's organization involve the use of metrics and measurement to ensure product quality.
- 7 out of 49 respondents i.e., 14% of respondent's organization involve the use of process improvement approaches to ensure product quality.
- 3 out of 49 respondents i.e., 8% of the respondent's organization involve the use of CCB sessions to ensure product quality.

Based on the above analysis, it can be inferred that most of the organizations involve the use of quality standards, quality reviews, metrics and measurement to ensure quality of product.

5.1.1.5 Metrics usage in organization

The analysis of metrics usage in the respondent's organization is performed to eliminate any bias as the main objective of the research is to evaluate the usefulness of the metrics involved in continuous delivery. Consider figure 4.5 and table 4.6, 45 out of 49 respondents (approximately 92% of respondents) involve the use of metrics in their organization.

Considering the data analysis performed over the means to ensure software quality in organization i.e., section 5.1.1.4, 41% of the respondents said that their organization uses metrics as one of the quality assurance methods, but if we consider the analysis performed over the metrics usage in organization, it can be inferred that 92% of the respondent's organization use metrics to measure software attributes. The authors did not find any particular pattern or correlation among these two questions as the main objective of this study was to evaluate the usefulness of metrics involved in continuous delivery.

5.1.1.6 Reasons for using metrics in organization

The analysis of the reasons to use metrics in organization is performed to allow the authors gain an understanding of the software practitioners regarding the reasons for using metrics in organization. Out of 49 respondents, 9 respondents presented reasons for using metrics in organization. 10 respondents specified that they are not sure of the reason for using metrics in organization. The remaining 30 respondents did not answer the question. The respondents were allowed to specify any reason as it was an open ended question. Based on the responses obtained from 9 respondents, the authors categorized the response into 4 reasons that can be observed in table 4.7. Consider the figure 4.6,

- 9 out of the 49 respondents specified the reason for using metrics as delivery of quality product.
- 4 out of 49 respondents specified the reason for using metrics as a means of tracking tasks
- 4 out of 49 respondents specified the reason for using metrics as a means to identify defects and improve the processes.
- 2 out of 49 respondents specified the reason for using metrics as a means to ensure customer satisfaction.

5.1.1.7 Release frequency of software into market

The analysis of the release frequency of software into market is performed to have a better understanding of how often the software is released into the market involving

continuous delivery approach. For ease of analysis, based on the data collected from online questionnaire, the authors categorized the release frequency in order of weeks, months and years. The categories can be observed in table 4.8. Consider figure 4.7, 20% of the respondent's organization release the product into market every 3 months. Similarly, 14% of organization release product every 2 months and 6 months. 12% of organizations release the product every 2 weeks. Similarly 30% of the organization's release the product every 1 week, 6 weeks, monthly, 2-6 months and yearly. Finally, 10% of the organizations release the product based on the customer requirements.

5.1.1.8 Involvement of respondents in continuous delivery

The analysis of the involvement of the respondent's in continuous delivery is performed to eliminate any bias of the results. The response from practitioners not having prior experience of working with continuous delivery can have a significant influence on the data collected and thus negatively influence the validity of results. Consider the table 4.9, the online questionnaire was answered by 79 software practitioners, out of which 49 respondents (62% of the total responses) had prior experience of working with continuous delivery.

5.1.2 Usability of metrics involved in continuous delivery

This section presents a description of the analysis performed over the data regarding the quality metrics involved in continuous delivery obtained from the online questionnaire.

5.1.2.1 Analysis of ratings given to category of metrics

The analysis of the rating given to category of metrics is performed to evaluate the usefulness of one category of metric over the other category. The frequency of respondents to rate the metrics from 0-100 in an interval of 20 is presented in figure 4.9. The analysis is performed by comparing the mean values of ratings given to each category of metric by the respondents. The data regarding the mean values of the rating for each category of metric is presented in table 5.1 and the graphical representation is presented in figure 5.1.

Category of metric used in continuous delivery	Mean value of the ratings
Metrics at implementation level	58.91
Metrics at pipeline level	37.43

Table 5.1: Mean values of rating given to each category of metrics

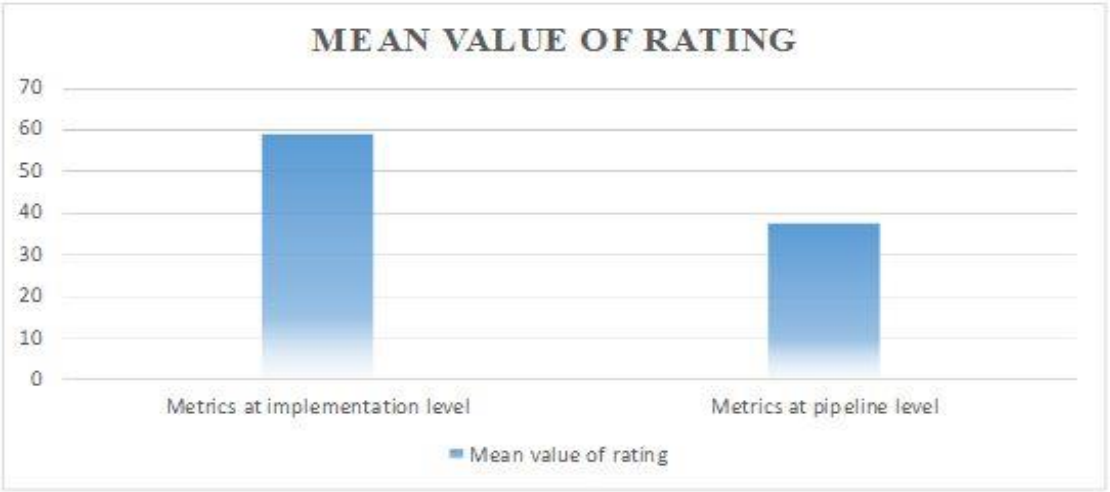

Figure 5.1: Mean value of rating given to each category of metric

Based on the figure 5.1 above, it can be inferred that metrics at implementation level are perceived to be highly useful compared to metrics at pipeline level based on the rating given by 49 respondents on a scale of 0-100.

5.1.2.2 Analysis of metrics usefulness based on the role of respondent:

Consider the figure 4.10 and 4.11, it can be inferred that most of the respondents had one of the roles i.e., business analyst, developer, project manager, quality analyst, team lead, technical lead and tester.

Business analyst:

- Based on the box plot of business analyst, 50% of the respondents (with role of business analyst) rate the metrics at implementation level more than 70 on a scale of 0-100. The remaining 50% rate it from a range of 70-65.
- Based on the box plot of business analyst, 50% of the respondents rated metrics at pipeline level from a range of 30-35. The remaining 50% rated it from a range of 10-30.

Developer:

- Based on the box plot of developer, 50% of the respondents (with role of developer) rate the metrics at implementation level more than 65 on a scale of 0-100. The remaining 50% rate it from a range of 30-65, but there is one outlier for the boxplot who rated it below 20.
- Based on the box plot of developer, 50% of the respondents rated metrics at pipeline level from a range of 30-85. The remaining 50% rated it from a range of 30-0.

Project Manager:

- Based on the box plot of project manager, 50% of the respondents (with role of developer) rate the metrics at implementation level from a range of 35-45 on a scale of 0-100. The remaining 50% rate it from a range of 25-35.
- Based on the box plot of project manager, 50% of the respondents rated metrics at pipeline level from a range of 35-70. The remaining 50% rated it from a range of 35-0.

Quality Analyst:

- Based on the box plot of quality analyst, 50% of the respondents (with role of developer) rate the metrics at implementation level from a range of 50-85 on a scale of 0-100. The remaining 50% rate it from a range of 40-50.
- Based on the box plot of quality analyst, 50% of the respondents rated metrics at pipeline level from a range of 50-65. The remaining 50% rated it from a range of 15-50.

Team Lead:

- Based on the box plot of team lead, 50% of the respondents (with role of developer) rate the metrics at implementation level from 60-75 on a scale of 0-100. The remaining 50% rate it from a range of 30-60.
- Based on the box plot of team lead, 50% of the respondents rated metrics at pipeline level from a range of 40-60. The remaining 50% rated it from a range of 20-40.

Technical Lead:

- Based on the box plot of technical lead, 50% of the respondents (with role of developer) rate the metrics at implementation level as 85 on a scale of 0-100. The remaining 50% rate it from a range of 45-85.
- Based on the box plot of technical lead, 50% of the respondents rated metrics at pipeline level from a range of 15-50. The remaining 50% rated it as 15.

Tester:

- Based on the box plot of tester, 50% of the respondents (with role of developer) rate the metrics at implementation level range of 70-75 on a scale of 0-100. The remaining 50% rate it from a range of 65-70.
- Based on the box plot of tester, 50% of the respondents rated metrics at pipeline level from as 35-38. The remaining 50% rated it from a range of 32-35.

5.1.2.3 Analysis of rating given to each metric

The main aim of the online questionnaire was to evaluate the usability of one metric over the other by allowing the respondents to rate the metrics on a scale of 0-100 based on their perception and working experience. In total 49 respondents presented rating to the metrics who had prior experience of working with continuous delivery. Out of these 49 respondents, 14 (29%) respondents were developers, 8 (16%) respondents were quality analysts and 5 (10%) respondents were system consultants. As there is diversity in the role of respondents, the perception of usability of each metrics differs from one another. So to derive at a consensus it is decided by the authors to calculate mean of the ratings given to each metric. The mean values of rating given to each metric are presented in the figure 5.2 below.

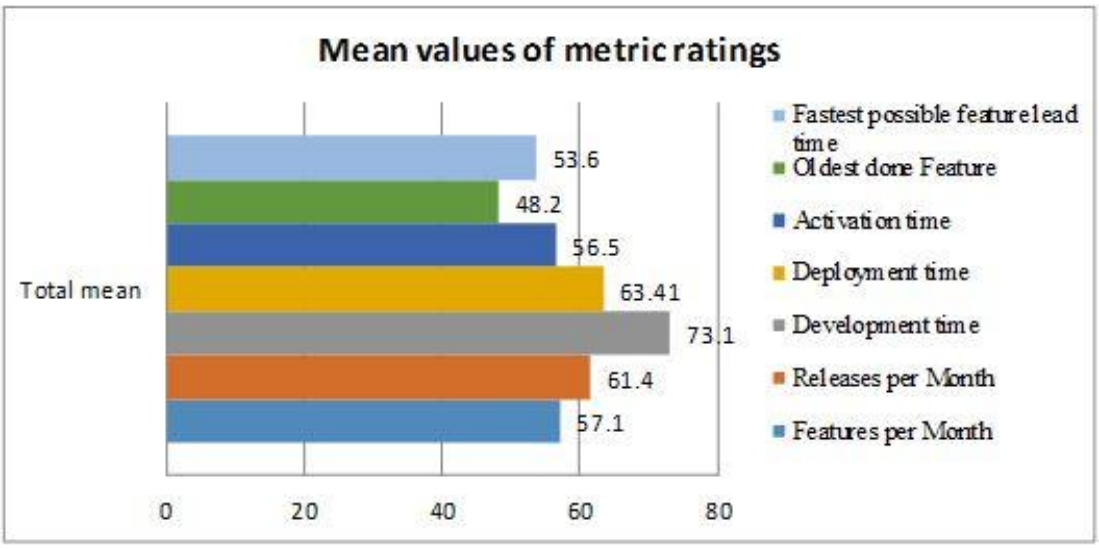

Figure 5.2: Mean values of rating given to each metric

From the figure 5.1, it is clear that the metrics at implementation level possess higher usability compared to metrics at pipeline level. It is obvious that it will be interpreted that all the sub-categories of metrics involved in implementation level are highly usable compared to sub-categories of metrics involved in pipeline level, but it is not true. It can be clearly inferred from figure 5.2 that Oldest done feature (sub-category of metrics at implementation level) is least usable metric amongst all the metrics involved in the both categories. So it can be inferred that though there is higher emphasis on a category of metrics, it is not necessarily true that the sub-category will also be highly usable. From the figure 5.2, the list of metrics ranging from higher usefulness to lower usefulness is as presented in figure 5.3.

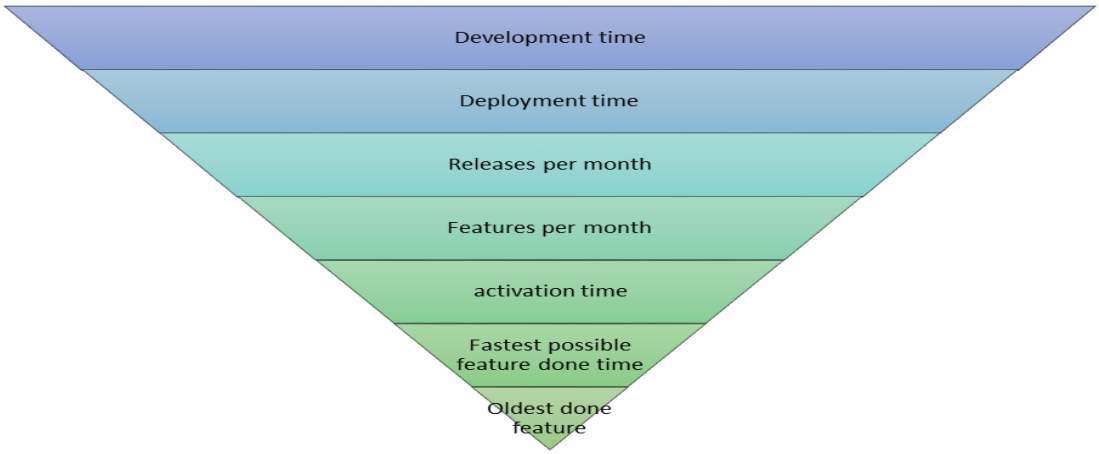

Figure 5.3: List of metrics based on usability

5.1.2.4 Limitations of metrics usage in organization:

The analysis of the limitations is performed to find the challenges that need to be addressed while using metrics. Based on table 4.13, 20 out of 49 respondents specified limitations to metrics usage. As this question was open-ended, the analysis of limitations was performed by careful reading of the 20 limitations and categorizing similar limitation into each category. Based on the 20 responses to this question, the authors categorized the limitations into 8 categories as presented in table 4.14. All category of limitations specified in table 4.14 are equally important, but the most important one that need to be addressed is alternatives for defects.

5.1.2.5 Recommended metrics:

The analysis of the metrics recommended by the respondents is performed as there are differences in usefulness of metrics and the metrics used based on the role of the respondent. Based on Figure 4.17, only 10 respondents provided additional metrics, seven respondents answered this question, but did not suggest any additional metrics that are used in continuous delivery. Remaining 32 respondents did not answer this question. The analysis of the recommended metrics by the respondents is presented in section 4.2.3.2.

5.2 Analysis of data collected from telephonic interviews

This section presents a description of the analysis of the data collected using telephonic interviews to evaluate the usefulness of the checklist in software organizations to ensure product quality. The interviews were conducted over Skype for a duration of 10-15 minutes. The interviews were recorded with prior permission of the respondent and later were transcribed. Before conducting the interview, the authors performed a survey using online questionnaire which resulted in generation of checklist of metrics which acted as a base for conducting this interview. 7 out of 15 interviewees were selected based on personal contacts of one of the authors. Remaining 8 interviewees were recommended by initial interviewees. The interviews were conducted by having a background check regarding the interviewees in the field of software industry.

The analysis was performed using manual coding by following the steps presented in “The coding manual for qualitative researchers” [31]. The authors analyzed the interview transcripts line by line and assigned codes which allowed the authors to arrive at a consensus. The code of each interview were documented using a Microsoft Excel worksheet. Once all the interviews were coded and the data was documented, the authors searched for codes that can be grouped to form a category. The analysis of the coded data, allowed the authors to evaluate the usefulness of the checklist in the organization.

5.2.1 Interviewee demographics:

5.2.1.1 Role of the interviewee in organization

The analysis of the role of interviewee in organization was performed to check whether it has any influence over the usability of the checklist presented to the interviewees for evaluation. The code assigned to role of interviewee was *role.* The analysis was performed using one of the descriptive statistics i.e., calculating the frequency of each role among the 15 interviewees. Based on data presented in table 4.16 based on 15 interviews, the authors recognized 6 different roles of the interviewees. 7 out of the 15 interviewees were developers and 3 of the interviewees were project managers. The rest of interviewees had the one of the role of technical lead, technical manager, quality analyst and program manager.

5.2.1.2 Experience of the interviewee in field of software organization

The analysis of the experience level was performed as experience level of the interviewee can have a significant influence over the results of the research. The interviewees had differences in their years of experience. So for ease of analysis, the authors categorized them into 5 categories as specified in table 4.17 above. Based on the data presented in the table 4.17, 8 of the 15 interviewees had experience in range of 8-11 years. 4 of the 15 interviewees had 4-7 years of experience and remaining 3 interviewees had experience in range of 12-15 years.

5.2.1.3 Development approach used in organization

The analysis of the development approach was performed to allow the authors to have a better understanding of the approaches that involve the use of continuous delivery in their organization. Based on table 4.18, 12 of the 15 interviewees are involved in scrum approach. 2 of 15 interviewees are involved in hybrid agile approach and remaining 1 interviewee uses kanban approach in their organization. The code assigned to analyze the data in transcripts is *Development approach*.

5.2.1.4 Metric usage in organization

The analysis of metrics usage in the respondent's organization is performed to eliminate any bias as the main objective of the research is to evaluate the usefulness of the metrics involved in continuous delivery. The code assigned to identify this category among the interview transcripts is *metric usage*. Consider the table 4.19, 12 of the 15 interviewees use metrics in their organization whereas remaining 3 interviewees does not involve the use of metrics in their organizations. The roles of these 12 interviewees who use metrics are developers, project manager, technical lead, technical manager, quality analyst and program manager. The role of the three interviewees who don't use metrics in their organization are one project manager and two developers.

5.2.1.5 Use of continuous delivery in organization

The analysis of the interviewee's organization using continuous delivery is performed to eliminate any bias as the responses obtained from interviewees not using continuous delivery can have a negative influence on the results of the study. The code used to identify this category in interview transcripts is *involvement in CD.* Based on the data presented in table 4.20, 14 of the 15 interviewees are involved in continuous delivery and remaining one interviewee's organization does not involve in continuous delivery yet the interviewee had prior working experience with continuous delivery.

5.2.1.6 Release frequency of product into the market

The analysis of the release frequency of the interviewee's organization is performed to allow the authors to gain better understanding of differences in release frequency of product into market by organizations. The code used to identify this category is

Release frequency. Based on the interviewee's responses, the authors categorized the release frequency into 6 categories. Based on data presented in table 4.21, 5 of the interviewees said that the release frequency depends on the customer requirements. 3 of the interviewees stated the release frequency of their organization to be 3 months. 2 interviewees stated the release frequency to be ranging from 6-8 months. 2 interviewees stated the release frequency to be 6 months. 2 interviewees stated the release frequency of product to be 2 months. Remaining 1 interviewee stated the release frequency to be 2 weeks.

5.2.2 Checklist validation

5.2.2.1 Analysis of rating given to level of complexity

The analysis of the rating given to the level of complexity in understanding the checklist is performed to evaluate the usability of the checklist. The code used to identify this category in the interview transcripts is *rating*. The interviewees were asked to rate the metrics on a scale of 1-10 based on the level of complexity in understanding the checklist, where 1 is least complex to understand and 10 is highly complex to understand. The data regarding the frequency of interviewees who rated the metrics is presented in table 4.22. As there are multiple values given to the level of complexity, the authors decided to calculate the mean of all the 15 values given by the interviewees. If the mean value of rating is less than 5 then the checklist can be termed as less complex to understand and fit for use in organization. If the mean value is greater than 5 then the checklist can be termed highly complex to understand and need to be improved before it is used in organization. The mean values of the ratings given by the interviewees is 4.3. Based on the mean value, it can be inferred that the checklist is less complex and can be used in organization by practitioners to ensure product quality.

5.2.2.2 Analysis of challenges presented by the interviewees

The analysis of the challenges in understanding the checklist is performed to allow the authors understand the perception of software practitioners and improve the checklist by addressing the challenges and making it more usable. The code used to identify the challenges in transcript is *challenges.* The author categorized the challenges based on the role of the interviewee. For ease of analysis, the author grouped the roles into managerial roles and non-managerial roles. The roles of project manager, technical lead, technical manager, program manager were grouped as managerial roles. The roles developer and quality analyst were grouped as non-managerial roles. The author based on the challenges specified by the 15 interviewees grouped them into 6 categories. The challenges addressed by managerial roles are quoted and presented in table 4.23. The challenges addressed by non-managerial roles are quoted and presented in table 4.24.

Consider table 4.23, with column name *slightly off the line*. The data in that column is

S1: "this is slightly off the line. You might have to improve the quality of the checklist I should say"

Here S1 refers to the response of the 1st interviewee regarding the challenge. Similarly S14 refers to the response of the 14th interviewee.

- According to the managerial roles, 4 of the interviewees addressed the unfamiliarity with the context of the metric.
- 2 interviewees with managerial roles find the checklist easy to understand.
- 1 of the interviewee with managerial role addresses that the checklist is slightly off-the line and needs to be improved.
- 1 of the interviewee with managerial role addresses that metrics needs to be precise and crisp
- 1 of the interviewee with managerial role addresses that some of the metrics such as MTTD, MTTR needs to be abbreviated.

Consider table 4.24, with column name *easy to understand*. One of the data in that column is

S6: "I don't have any problem in understanding the checklist"

Here S6 refers to the response of the 6th interviewee regarding the challenges faced in understanding the checklist. Similarly, S15 refers to response of the 15th interviewee.

- 3 interviewees with non-managerial role address that the checklist is easy to understand
- 1 of the interviewee with non-managerial role addresses that the metrics need to be crisp and precise.
- 1 of the interviewee with non-managerial role addresses that some of the metrics need to be abbreviated.
- 1 of the interviewee with non-managerial role addresses that most of the metrics are unfamiliar as they follow a different development approach.

5.2.2.3 Recommended usage of checklist in organization

The analysis of the checklist usage in the organization is performed to help the authors evaluate the usefulness of the metrics in organization. Based on table 4.25, 10 of the interviewees stated that they would definitely recommend the use of checklist in the organization. 2 of the interviewees stated that they would not recommend the use of checklist in organization as they have a better checklist. 3 interviewees stated that they would recommend the use of the checklist if the challenges that they specified are addressed and changes are made to the checklist.

5.3 Discussion

Little research has been done in the field of quality metrics used in continuous delivery. This research has discovered some potential metrics used by practitioners involved in continuous delivery approach. Our data shows that the metrics used differ based on the role of the practitioners in the organization. The limitations of metrics usage in organization as perceived by practitioners is also presented. There are differences in the practitioner's perception regarding the usefulness of the metrics involved in continuous delivery. This study has presented a metrics checklist used in continuous delivery.

Continuous delivery literature suggests that the tools used in pipeline generate lots of data which can be used to analyze the performance and present status of the pipeline [29]. The results of our analysis show similar pattern with metrics used in continuous delivery. However, our study suggests that there are more metrics used in continuous delivery approach apart from the one's specified in the literature. Further research could be performed on identifying more potential metrics used in continuous delivery apart from the ones specified in this study.

Our research suggests that the usefulness of the metrics differs from practitioner to practitioner i.e., the usability of the metrics used in continuous delivery based on various roles involved in the organization. For example, developers perceive metrics at implementation level as more useful compared to metrics at pipeline level. Similarly, project managers perceive metrics at pipeline level as more useful compared to metrics at implementation level. However, the metrics selection depends on the requirements of the project and decisions of top-level management.

Previous studies suggests the use of checklist for metrics planning to avoid any ambiguity regarding metrics being measure in the project [16]. The present study provides a metrics checklist that can be used in a project involved with continuous delivery. Further research can be done to complement the checklist presented in the literature with the checklist presented in this study.

5.3.1 Implications for Practice and Research

Our research suggests a checklist that can act as a starting point while planning metrics to be used in the project. It can also act as a remainder to the organization regarding the metrics that need to be measured before the closure of the project. This checklist was

generated based on responses from the practitioners involved in continuous delivery and has not yet been operationalized. However, the findings of our study yield implications both for researchers and practitioners.

One of the key aspects uncovered by our research is regarding the limitations of using metrics in software organization. The important limitation that need to be addressed by the practitioners is providing alternatives ways to handle defects during metrics collection. The limitations specified can help the practitioners to understand the shortcomings of metrics usage and mitigate them while trying to implement them. The limitations specified can be used by researchers as a first step towards the research in context of quality metrics involved in continuous delivery.

The key aspects uncovered by our research can be used by practitioners as a foundation for quality assurance and making improvements to the project in case of deficiency. For example, the practitioners may select the context of using the metric as Process metric (Specified in the checklist). Based on the context, the practitioners can measure certain process metrics such as Mean time to Detect (MTTD), Mean time to Repair (MTTR), Defect removal efficiency, Mean time before Failure (MTBF), development throughput and check-in time to go live in production. Once the attributes surrounding the metrics are measured, they can be used for quality assurance purposes or reference purposes by the software organization.

Our study can act as a stepping stone for future research in area of quality metrics involved in continuous delivery. To support the organization to find a threshold value based on which they can improve, a checklist is required. Identification of metrics from domains other than continuous delivery approach, like agile approaches and waterfall approach can be valuable inputs to generate checklist that can be used by organization having different software development approaches.

6 CONCLUSION AND FUTURE WORK

A description of the conclusions drawn from this study, answers to the research questions and future work for this study are presented in this chapter. The remainder of this chapter is structured as follows: section 6.1 presents the conclusions of the study and answers to the research questions. Section 6.2 presents the threats to validity for the study. Finally the future work that can be carried out is presented in section 6.3.

6.1 Conclusions

In this study, the researchers tried to explore the field of quality metrics involved in continuous delivery and present a checklist of quality metrics that can be used by practitioners to ensure software quality. The checklist is generated by gathering data regarding the usefulness of metrics using one of the survey methods i.e., online questionnaire. The data was analyzed using descriptive statistics such as mean, median, frequencies and bar graphs. The checklist generated is then evaluated for its usefulness in software organizations using telephonic interviews with software practitioners involved in continuous delivery. The data obtained from telephonic interviews is analyzed using manual coding.

Based on the analysis performed, it can be stated that the software practitioners has different views related to usefulness of the metrics. The software practitioner perception regarding the usefulness of the metrics depends on their role in the organization i.e., a developer may perceive metrics at implementation level to be more useful compared to metrics at pipeline level. Similarly, a quality analyst may perceive metrics at pipeline level to be more usable compared to metrics at implementation level. Based on the analysis performed over the metrics usefulness, it is inferred that though the software practitioner perceive one category of metrics to be more usable it is not necessary that the metrics involved in the category are perceived to be equally usable. For example, the category of metrics at implementation level is perceived to be more usable, but one of the metrics involved in this category (Oldest done feature) is perceived to be less usable, compared to metrics involved in category of metrics at pipeline level.

Based on the analysis performed over the telephonic interviews, it is inferred that the checklist of metrics is fit for use in organization to plan metrics. Most of the interviewees did not find it difficult to understand and thus addressed that they would recommend the use of this checklist in their organization. One of the interviewee addressed as follows:

> *"I just wanted to seek you permission to use this checklist on my trainees to test their understandability of concept of continuous delivery"- software practitioner*

Some of the interviewees would recommend the use of this checklist in their organization if some changes that they suggested are made to checklist. The interviewee addressed the authors question as follows:

> *"Yeah definitely I would recommend the use of this checklist in our organization, but only if the challenges that I specified are addressed and necessary changes are made to checklist"-software practitioner*

Based on the above conclusions, the organizations involving continuous delivery have various methods of ensuring software quality, but there has not been any such thing as a checklist of metrics that acts as a remainder for the software practitioners to ensure that none of the decided metrics are left out without measuring. This checklist also act as a record of the metrics decided at the start of the project or sprint and check whether the decided metrics have been measured or not.

The research questions and the answers based on our study are as follows:

RQ 1. What quality metrics are being used to measure product quality reported in continuous delivery literature?

The background regarding the evolution of continuous delivery, metrics used to measure product quality in agile and continuous delivery approaches are investigated using literature reviews. Based on the literature review, the author categorized the metrics used in agile approach into functional metrics and non-functional metrics. Functional metrics and non-functional metrics are further divided into sub-categories. Each sub-category consists of metrics that help to ensure product quality. Once the metrics in agile approaches are investigated, the background related to continuous delivery and the continuous delivery pipeline is investigated. The pipeline consists of six stages which involves code commit, build, acceptance test, performance test, manual test and finally production. Once the literature regarding the continuous delivery pipeline is studied, the literature related to metrics involved in continuous delivery is explored. Metrics involved in continuous delivery is categorized in literature as metrics at implementation level and metrics at pipeline level. Each category of metrics is further subdivided into metrics that help to ensure product quality. Metrics at implementation level is composed of activation time, development time, deployment time and oldest done feature metrics. Metrics at pipeline level is composed of feature per month, fastest possible feature lead time and releases per month metrics.

RQ 2. Which metrics are highly useful as perceived by practitioners in continuous delivery approaches?

One of the sub-questions of this research question was answered using literature review and the remaining two sub-questions were answered by conducting an empirical survey using online questionnaire. As the main aim of this research is to evaluate the usability of quality metrics involved in continuous delivery, the authors designed a questionnaire based on the metrics obtained from literature review related to continuous delivery. Based on the responses obtained from the survey, we evaluated the seven metrics used in continuous delivery and we also ranked their usefulness. Based on the usefulness of the metrics, we presented an order of metrics which can be observed in figure 5.3. The authors also investigated regarding the limitations of metrics usage in software organizations using the online questionnaire. The respondents of the questionnaire also recommended some metrics that are in practice, but not reported in literature.

RQ 3. Is the quality metrics checklist fit for use (valid) in software organizations involved in continuous delivery?

After generating the checklist of metrics used in continuous delivery, the authors decided upon evaluating the usefulness of the checklist in the organizations using telephonic interviews with software practitioners. The authors conducted 15 interviews with software practitioners and each interview lasted for 10-15 minutes. To evaluate the checklist, the authors asked the interviewees to rate the metrics based on the level of complexity in understanding the checklist on a scale of 1-10 and also specify the challenges in understanding the checklist. The average rating given to the checklist by the interviewees is 4.3 which means that the checklist is less complex and easy to understand. Later the authors also asked the interviewees whether they would recommend the use of this checklist in their organization and 13 out of 15 interviewees stated that they would recommend the use of this checklist. Some of the interviewees stated that it can act as a starting point of the development cycle and some stated that it can act as a remainder to software practitioners to ensure quality.

6.2 Limitations and Threats to validity

The authors are aware of few limitations of the study that should be taken into consideration while working with the reported findings:

- Only metrics specified in research articles were included in the study
- The study did not include the metrics that are presented in blogs/forums and other information resources.

One of the possible threats to the study is regarding the validity of the results obtained by combining metrics involved in agile with metrics involved in continuous delivery. This threat was mitigated while performing literature review where authors found that agile approach is the base for continuous delivery approach. This served as a rationale for combining metrics used in agile approaches with the metrics used in continuous delivery and generate a checklist of metrics. The threats to validity using the four types of threats as suggested by Wohlin [10] are as follows:

Internal validity: It is concerned with the issues that arise due to irrelevant responses, improper design of the survey instrument and selection of participants that might have a significant influence over the results.

- This issue regarding irrelevant responses was mitigated by asking the questionnaire respondents whether they had prior experience of work with continuous delivery or not. As the authors used an online questionnaire hosting website, they were able to use question logic. Question logic allowed the respondents to answer the remaining questions of the questionnaire if they specified that they had prior experience of working with continuous delivery. If the online questionnaire respondents specified they had no prior working experience with continuous delivery then the remaining questions of the questionnaire would be hidden. Such responses were also recorded, but were not used during data analysis.
- The survey instrument used in this research is online questionnaire. The authors possibly mitigated the issue regarding design of questionnaire by conducting a pilot survey over two practitioners in continuous delivery to check for understandability, readability and completeness of the questionnaire. Based on the feedback obtained from the practitioners, the questionnaire was updated accordingly and hosted online.
- The selection of participants was mitigated by sending the questionnaire to organizations whose personnel are involved in continuous delivery. A list of organizations that are involved in continuous delivery was gathered and the questionnaire was sent to the organizations over E-mail.

External validity: It is concerned with the issue of generalizing the results out of the context of study. The sampling method used for selecting interview participants to evaluate the checklist was convenience sampling. Since the author used convenience sampling, the results could only be generalized to those continuous delivery teams and organizations that possess similar characteristics to the interviewee's team and organizations. As the questionnaire was hosted online and over continuous delivery related online-forums and blogs, the sample of interviewees is heterogeneous with respect to work experience and interviewees role in organization.

Construct validity: It is concerned about whether the study being conducted measures the things that the authors claim. In this case, issues may arise if complexity is not the right measure for evaluating the checklist of metrics. This threat was mitigated by structuring the interview questions such that the perception of the practitioners regarding the checklist's usefulness in the organization is recorded. One such question involved in interviews with practitioners is whether they would recommend the use of the checklist in their organization based on its level of understandability.

Conclusion validity: It is concerned with the possibility of drawing incorrect conclusions about the observations due to use of improper statistical tests or measures. This issue is mitigated as the authors used only frequencies and one of descriptive statistics method i.e.,

mean to investigate the most useful metrics in continuous delivery as perceived by software practitioners.

6.3 Future work

Many studies reveal that the success of the product is based on faster delivery of a good quality software/product to the customer. So to ensure this the concept of agile was evolved which further evolved into continuous delivery. Little literature has been published related to continuous delivery and quality metrics involved in continuous delivery. This study attempted to present the field of continuous delivery with a checklist of quality metrics that can be used to ensure good software product development. The study is limited in identifying only metrics based on 49 responses for online questionnaire. There is possibility of extending this research by involving a large sample of population. The study can be extended by making the changes to the checklist as recommended by the interviewees and validating it by implementing it in an organization.

7 REFERENCES

[1] K. Beck et al, "Manifesto for Agile Software Development. Agile Alliance", http://agilemanifesto.org/ (Retrieved 14 Jan 2012).

[2] J. Humble and D.Farley, "Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation", Addison Wesley, 2010.

[3] Chang H, C-Y.Lu S, "Toward the Integration of Traditional and Agile Approaches," International Journal of Advanced Computer Science and Applications. 2013, vol. 4, no. 2, pp. 9-13.

[4] S. Krusche, L. Alperowitz, B. Bruegge, and M. O. Wagner, "Rugby: An Agile Process Model Based on Continuous Delivery," in *Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering*, New York, NY, USA, 2014, pp. 42–50.

[5] M. Virmani, "Understanding DevOps bridging the gap from continuous integration to continuous delivery," in *2015 Fifth International Conference on Innovative Computing Technology (INTECH)*, 2015, pp. 78–82.

[6] S. Neely and S. Stolt, "Continuous Delivery? Easy! Just Change Everything (Well, Maybe It Is Not That Easy)," in *Agile Conference (AGILE), 2013*, 2013, pp. 121–128.

[7] O. Akerele, M. Ramachandran, and M. Dixon, "System Dynamics Modeling of Agile Continuous Delivery Process," in *Agile Conference (AGILE), 2013*, 2013, pp. 60–63.

[8] C. Robson, *Real world research : a resource for users of social research methods in applied settings*. Chichester: Wiley, 2011.

[9] Y. Levy and T. J. Ellis, "A systems approach to conduct an effective literature review in support of information systems research," *Informing Sci. Int. J. Emerg. Transdiscipl.*, vol. 9, no. 1, pp. 181–212, 2006.

[10] C. Wohlin, *Experimentation in software engineering : an introduction*. Boston: Kluwer, 2000.

[11] E. Kupiainen, M. V. Mäntylä, and J. Itkonen, "Using metrics in Agile and Lean Software Development – A systematic literature review of industrial studies," *Inf. Softw. Technol.*, vol. 62, pp. 143–163, Jun. 2015.

[12] B. Kitchenham and S. Charters, *Guidelines for performing Systematic Literature Reviews in Software Engineering*. 2007.

[13] A. Barreto, M. de O. Barros, and C. M. L. Werner, "Staffing a software project: A constraint satisfaction and optimization-based approach," *Comput. Oper. Res.*, vol. 35, no. 10, pp. 3073–3089, Oct. 2008.

[14] M. J. Ordonez and H. M. Haddad, "The State of Metrics in Software Industry," in *Fifth International Conference on Information Technology: New Generations, 2008. ITNG 2008*, 2008, pp. 453–458.

[15] S. Misra and M. Omorodion, "Survey on Agile Metrics and Their Inter-relationship with Other Traditional Development Metrics," *SIGSOFT Softw Eng Notes*, vol. 36, no. 6, pp. 1–3, Nov. 2011.

[16] D. Hartmann and R. Dymond, “Appropriate agile measurement: using metrics and diagnostics to deliver business value,” in *Agile Conference, 2006*, 2006, p. 6 pp.–134.

[17] A. Kushwaha, S. K. Verma, and C. Sharma, “Analysis of the Concerns Associated with the Rapid Release Cycle,” *Int. J. Comput. Appl.*, vol. 52, no. 12, 2012.

[18] N. Kerzazi and F. Khomh, “Factors Impacting Rapid Releases: An Industrial Case Study,” in *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, New York, NY, USA, 2014, pp. 61:1–61:8.

[19] W. M. Farid and F. J. Mitropoulos, “NORPLAN: Non-functional Requirements Planning for agile processes,” in *2013 Proceedings of IEEE Southeastcon*, 2013, pp. 1–8.

[20] O. Aktunc, “Entropy Metrics for Agile Development Processes,” in *2012 IEEE 23rd International Symposium on Software Reliability Engineering Workshops (ISSREW)*, 2012, pp. 7–8.

[21] M. Kunz, R. R. Dumke, and N. Zenker, “Software metrics for agile software development,” in *Software Engineering, 2008. ASWEC 2008. 19th Australian Conference on*, 2008, pp. 673–678.

[22] M. Goldratt Eliyahu, “The haystack syndrome: sifting information out of the data ocean,” *EM Goldratt*, 1990.

[23] V. Armenise, “Continuous Delivery with Jenkins: Jenkins Solutions to Implement Continuous Delivery,” in *Proceedings of the Third International Workshop on Release Engineering*, Piscataway, NJ, USA, 2015, pp. 24–27.

[24] L. Chen, “Continuous delivery: Huge benefits, but challenges too,” *Softw. IEEE*, vol. 32, no. 2, pp. 50–54, 2015.

[25] L. Chen, “Towards Architecting for Continuous Delivery,” in *Software Architecture (WICSA), 2015 12th Working IEEE/IFIP Conference on*, 2015, pp. 131–134.

[26] S. Gebert, C. Schwartz, T. Zinner, and P. Tran-Gia, “Continuously delivering your network,” in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2015, pp. 766–769.

[27] S. Klepper, S. Krusche, S. Peters, B. Bruegge, and L. Alperowitz, “Introducing Continuous Delivery of Mobile Apps in a Corporate Environment: A Case Study,” in *Proceedings of the Second International Workshop on Rapid Continuous Software Engineering*, Piscataway, NJ, USA, 2015, pp. 5–11.

[28] J. Gmeiner, R. Ramler, and J. Haslinger, “Automated testing in the continuous delivery pipeline: A case study of an online company,” in *Software Testing, Verification and Validation Workshops (ICSTW), 2015 IEEE Eighth International Conference on*, 2015, pp. 1–6.

[29] T. Lehtonen, S. Suonsyrjä, T. Kilamo, and T. Mikkonen, “Defining Metrics for Continuous Delivery and Deployment Pipeline.”

[30] J. Rowley and F. Slack, “Conducting a literature review,” *Manag. Res. News*, vol. 27, no. 6, pp. 31–39, 2004.

[31] J. Saldana, “The Coding Manual for Qualitative Researchers*,”* SAGE, 2015.

[32] DeMarco, Tom. Controlling Software Projects: Management, Measurement & Estimation, Yourdon Press, New York, USA, 1982.

8 APPENDIX A: QUESTIONNAIRE

Quality Metrics in Continuous Delivery

Continuous Delivery deals with software development in small fragments and transferring the developed code into a central repository. Source code from the central repository is then moved to delivery pipeline where the code is compiled and various tests are performed automatically. Once the source code reaches the final stage of pipeline, it can be deployed by the production department just by one click of button.

Metric is a quantitative measure of the degree to which a component, system, or process possess a given characteristic or an attribute. Based on the metrics obtained from the literature review regarding Continuous Delivery, we intend to perform a survey to find the usefulness and limitations of the metrics used in Continuous Delivery. The survey is intended to fulfil the following objectives:

1. Evaluate the usefulness of metrics being used in Continuous Delivery.

2. Limitations of metrics being used in Continuous Delivery

Based on the results obtained from survey, we intend to create a check list of metrics that can be used by practitioners and managers to measure the activities involved with development process and product.

Metrics used in continuous delivery are categorized as metrics used at implementation level and metrics used at pipeline level.The metrics used at pipeline level include Features Per Month (FPM), Releases Per Month (RPM) and fastest possible feature lead time. Metrics used at implementation level involve deployment time, development time, activation time and Oldest Done Feature (ODF).

Thank you for visiting us. By filling out this 5-10 minute survey, you will:

- Help us obtain results that can be useful to fill the research gap related to usability of metrics involved in Continuous Delivery.
- Receive first hand information regarding the usefulness and limitations of Continuous Delivery Metrics.

Note: Anonymity of the Survey participant is fully preserved. The raw data would be only accessible by the two researchers. The analysis results of the survey will be presented as a master thesis at Blekinge Institute of Technology (www.bth.se).

If you have any questions about the survey please contact the researchers,

Raghu ram Aduri (E-mail: **raad15@student.bth.se**)

Aman Jain (E-mail: **amja15@student.bth.se**)

1

Job Title

Please specify your role in the organization such as developer, tester, analyst and many more

Type a sentence

250 characters remaining

2

Years of experience in field of software development

Please specify the number of years

Type one or a few words

50 characters remaining

3

Which of the below agile approach is implemented in your organization?

- Scrum
- Scrumban
- XP
- Feature Driven Development
- crystal
- Any approach other than agile? please specify

Type your answer

100 characters remaining

4

How does your organization ensure quality of product being developed?

- [ ] Use of Quality Standards (example: CMMI, ISO 9001:2015)
- [ ] Quality reviews (Reviewing the quality aspects using audits)
- [ ] Process improvement approaches (APIM, GQM+ strategies, Tick-It)
- [ ] Measurement and Metrics (example of metrics: Lines of Code, performance metrics)
- [ ] Please specify, if any other means of ensuring quality

Type your answer

100 characters remaining

5

Does your organization use metrics to measure quality aspects of product being developed?

- ( ) Yes
- ( ) No

6

If your organization uses metrics, Why does your organization adopt metrics to ensure quality?

Type a paragraph

1500 characters remaining

7

How often do you release your product to the market/customer?

The release frequency of software (number of days/weeks/months)

Type one or a few words

50 characters remaining

8

Do you have prior experience of working with Continuous Delivery?

- yes
- no

Metrics at implementation level: These are dependent on the tool set and practices used to implement the pipeline. The data flow and throughput are dependent on the implementation of pipeline. Metrics at implementation level are further divided into Development time, deployment time, activation time and oldest done feature.

Metrics at pipeline level: These metrics are isolated from the implementation of pipeline. This is further divided into metrics that measure the number of releases per month, features per month and fastest possible feature lead time.

9

Continuous Delivery involves metrics at implementation level and pipeline level. Which metrics would you often use to ensure quality?

Rate the two metrics out of 100

Assign **100** *points*

Metrics at implementation level

0 100

Metrics at pipeline level

0 100

Development time :- The time taken by the team to implement a new feature

Deployment time :- The time taken to deploy new features to production.

Activation time :- The time taken by first user to activate new features after deployment.

Oldest done feature :- Time spent by a feature in development done state i.e., waiting for being moved to production state from deployment.

Features per month :- Number of features that have moved through the pipeline in a period of one month.

Releases per month :- The number of releases made during a period of one month.

Fastest possible feature lead time :- The time spent by a feature in the build and test stages of pipeline.

10

Rate the metrics based on their usefulness to measure quality

Rate the metrics from 0 - 100. If you specify the rating as 0, it will be interpreted as your unfamiliarity with the particular metric.

	Rating
Features per Month (FPM)	
Releases per Month (RPM)	
Development time	
Deployment time	
Activation time	
Oldest done Feature (ODF)	
Fastest possible feature lead time	

11

Describe the limitations of metric usage observed in your organization.

Type a paragraph

1500 characters remaining

12

If possible, recommend any other metrics that are used in your organization to ensure quality?

Type a paragraph

1500 characters remaining

13

E-mail Id

If you are interested to discuss about the topic, please provide your contact information.

Type one or a few words

50 characters remaining

14

Comments/ Suggestions to help us improve the survey

Type a paragraph

1500 characters remaining

SUBMIT SURVEY

9 APPENDIX B: INTERVIEW PROTOCOL

Project: Quality metrics in Continuous Delivery

Date _____

Time _____

Location _____

Interviewer _____

Interviewee _____

Notes to interviewee:

Thank you for your participation. I believe your input will be valuable to this research and in helping grow all of our professional practice.

Confidentiality of responses is guaranteed

Approximate length of interview: 15 minutes, five major questions

Purpose of research:

To evaluate the metric checklist generated based on responses obtained from the questionnaire.

i. *Would you recommend the usage of this checklist in your organization? Specify why you would use it or not use it?*
ii. *What is the level of complexity in understanding the metrics specified in the checklist? (Rating)*
iii. *What challenges do you face?*
iv. *Please specify any other recommendations that can be made to the checklist to make it more usable.*

1. Can you please specify your background in field of software development?

   a. Please specify the number of years of experience in field of software industry.
   b. What is the development approach used in your organization?
   c. What type of method does your organization use to ensure quality of product (quality reviews, quality standards, process improvement approaches, metrics and measurement)
   d. How often does your organization release product into the market?
   e. Does your organization use metrics to measure software attributes?
   f. Does your organization involve in continuous delivery?

Response from Interviewee:

Reflection by Interviewer

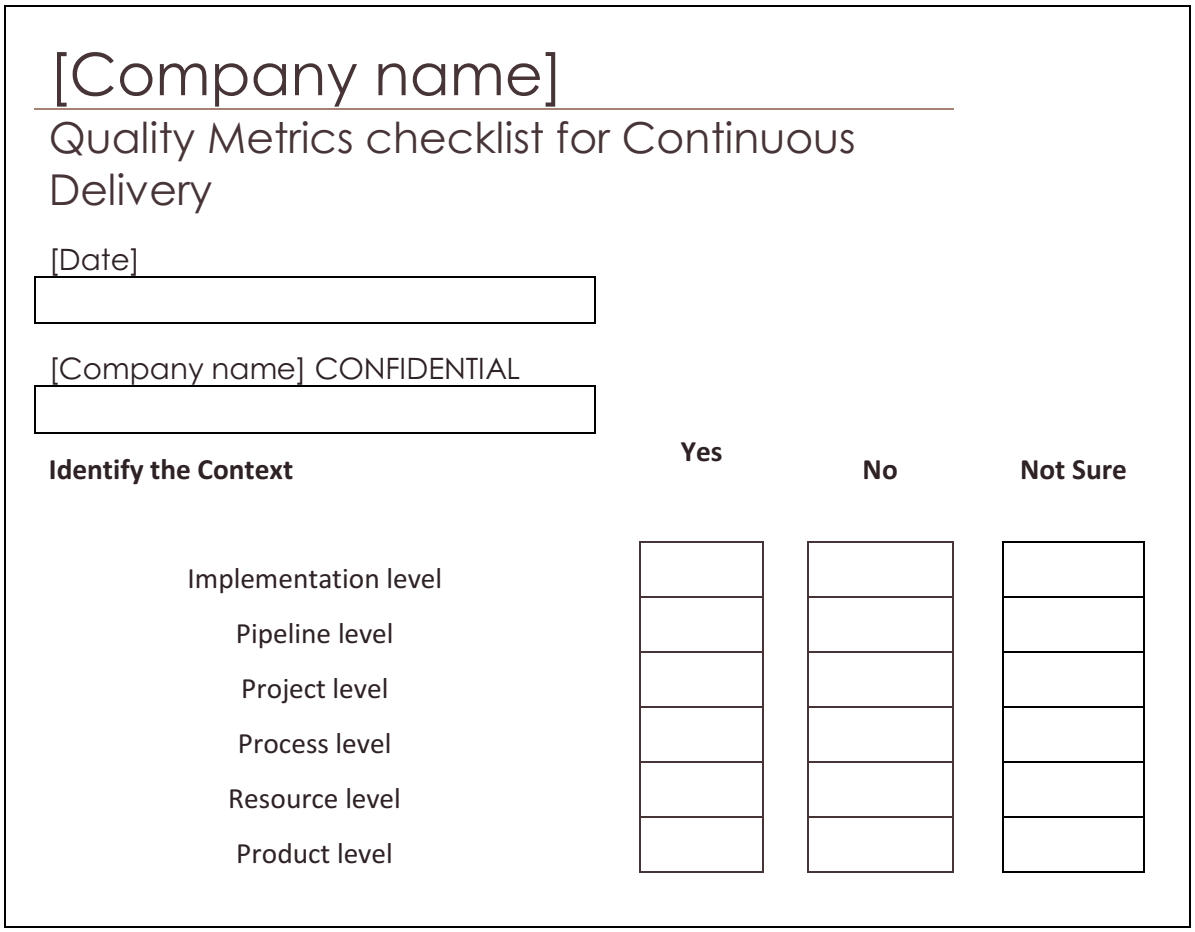

Figure 1: Checklist to identify the context where you want to ensure quality

[Company name]

Quality Metrics checklist for Continuous Delivery
[Date]

JOB DESCRIPTION:

Metrics checklist	Yes	No	Not sure
Metrics at implementation level			
Development time			
Deployment time			
Activation time			
Oldest done feature			
Metrics at pipeline level			
Releases per Month (RPM)			
Features per Month (FPM)			
Fastest possible feature lead time			
Product level metrics			
Number of defects			
Code scalability index			
Customer reported issues			
Resource level metrics			
Communication effectiveness			
Customer relationship			
Employee satisfaction			
Distraction index			
Trust index			
Aggravation index			
Supplier/partner index			
Process level metrics			
MTTD			
MTTR			
Defect removal efficiency			
Mean time before Failure (MTBF)			
Development throughput			
Check-in time to go live in production			
Project level metrics			
Intellectual capital			
Key Performance Index (KPI)			
Project Management index			

Figure 2: Checklist of metrics involved in each context

2. Please evaluate the checklist presented to you by the researchers.

    a. What is the level of complexity in understanding the metrics specified in the checklist? (Rating)
    b. What challenges do you face?
    c. Would you recommend the usage of this checklist in your organization? Specify why you would use it or not use it?
    d. Please specify some recommendations that can be made to the checklist to make it more usable.

Response from Interviewee:

Reflection by Interviewer

Closure

- Thank you to interviewee
- reassure confidentiality
- ask permission to follow-up _____