

# Quality Aspects of Continuous Delivery in Practice

Maryam Shahzeydi

Computer Engineering Department  
Dolatabad Branch, Islamic Azad University,  
Isfahan, Iran

Taghi Javdani Gandomani\*

Computer Engineering Department  
Boroujen Branch, Islamic Azad University, Boroujen, Iran  
Computer Engineering Department  
Dolatabad Branch, Islamic Azad University, Isfahan, Iran

Rasool Sadeghi

Computer Engineering Department  
Dolatabad Branch, Islamic Azad University, Isfahan, Iran

**Abstract**—Continuous Delivery is recently used in software projects to facilitate the process of product delivery in Agile software development. As an Agile practice, this practice is mainly used to achieve better quality of software development process and higher customer satisfaction. However, less attention has been paid on exploring the quality factors related to Continuous Delivery as well as quality model. The main aim of this paper is to figure out the quality aspects and factors of Continuous Delivery. Initial data analysis showed that this practice is impressed by people related factors, organizational issues, tools and process related factors as well.

**Keywords**—Continuous delivery; quality model; agile software development; agile methods; agile practice

## I. INTRODUCTION

Agile methods are widely using in software development projects since the last decade. These methods promote a different style of software development which distinguishes the development from traditional or disciplined methods in software engineering. Focusing on Agile values and principles, defined in Agile manifesto [1], these methods promote early and frequent delivery, higher quality level, better customer collaboration, embracing required changes in customer's requirements and so on [2]. It's why many software companies are looking for the best way to adopt these methods in their software product lines [3]. However, they are faced with various challenges [4].

Agile software development includes various methods such as Scrum, Extreme Programming (XP), Crystal family, Test Driven Development (TDD), Feature Driven Development (FDD), etc. [2], each defines its own particular practices, roles and artifacts. However, usually, Agile software teams use various practices that can be commonly used in all Agile methods [4], [5]. Continuous Delivery (CD) is one the popular practices which recently has gained special importance for Agile projects.

CD focuses on releasing reliable software product through software development, test and deployment [6]. This practice was introduced in 2010, as the ability to release every time [7]. However, the core concept of CD is not really continuous code development; it is the ability of release at any time [6], [8]. Indeed, the recently developed code should have the ability to

be added by new features and functionalities as easy as possible.

Since the ultimate goal of software development is achieving customer satisfaction through increasing quality of both development process and product, quality of all the development practices is important. Quality of CD also play a great role in customer satisfaction. Better conduction of this practice may lead to higher customer trust directly and satisfaction indirectly. However, the literature review shows less effort on exploring the quality related aspects of CD, proposing a clear quality model, or even providing guidelines to increase quality of this practice in real environments.

This article tries to explain the concept of CD from the lens of quality and address the most related previous studies and finally describe the outline of a required quality model dedicated for this practice. So, the rest of this article is organized as follows: Section 2 describe the underlying concept of CD briefly. Section 3 addresses the most related works followed by Section 4 which outlines a quality model associated for CD. Section 5, finally, concludes the paper.

## II. CONTINUOUS DELIVERY

Agile software development defines an underpinning concept, short cycles, which its focus is on early and frequent delivery. To establish such concept, Agile approach defines proper practices, among them CD plays a critical role. As mentioned before, CD focuses on the ability of software release whenever customer needs [6], [8]. CD is a really a practice to help the software stakeholders (i.e. business and technical parties) to collaborate in development and deployment of a software product in short cycles while focusing on the quality factors.

Technically speaking, CD is considered as an Agile practice which facilitates the process of delivery of product increments upon the customer request. However, CD focuses more on commitment to ensuring the recently developed code is able to be released at any time rather than the delivery process [9]. The promised advantages of CD temp both business and technical practices of software development to adopt it in their product line [8]. Accelerating time-to-value, quick user feedback, achieving clear and visible believable

progress, reducing the risks of delivery, providing innovations in the release process, better quality and data-driven decision making are the most addressed advantages and benefits of CD in practice [10], [11].

The above advantages have root in the concepts and goals of CD. For instance, frequent delivery and release provides the ability to get customer feedback timely and faster. Also, short cycles and frequent delivery increase the chance of risk discovery and avoid them in the next delivery. So, better quality will be expected indirectly. Recently and in the

competitive software industry, lots of the reputed companies such as Facebook, Google, IBM and Microsoft are trying to use CD as a compulsory development practice in their project [6], [11].

CD process includes a series of activities all together are known as “Continuous Delivery Pipeline”. As shown in Fig. 1, this pipeline involves some automatic and manual tasks. Although, literature review shows different steps for this pipeline, all are almost the same in tasks and activities in which Build, Staging and Production are constant [7], [11], [12].

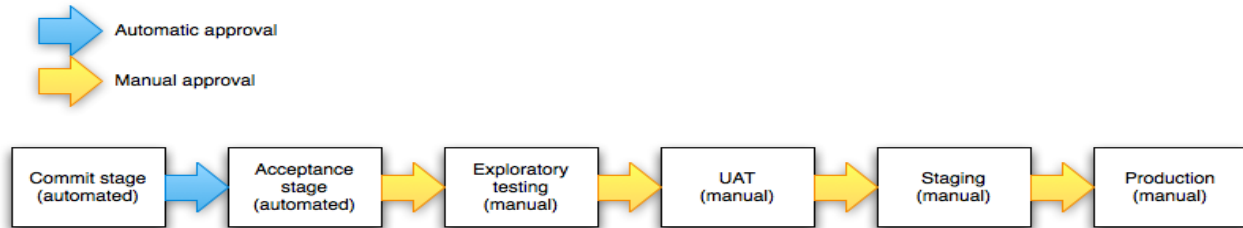


Fig. 1. Typical continuous delivery pipeline.

In the Build stage software teams use source repository as input and store an artifact in the artifact repository. The main goal of this stage is software development, software test, packaging and archiving. Unit tests are mostly used in this stage. The second stage, Staging, software teams install and deploy the recently build artifact in a staging environment and simultaneously perform regression, performance, integration and functional tests. Production stage, finally, focuses on deployment of the recently testes software into the production environment [7].

Despite of its simple concept, employing of CD in practice needs proper conditions. For instance, software development process needs to support iterative development in advance [13]. Indeed, without defining several iterations, CD cannot be considered. This would be a serious limitation for small projects where number of iterations are limited. Furthermore, extensive of positive team climate and also positive atmosphere between customer and development team is necessary [13].

### III. RESEARCH BACKGROUND

Most of the previous studies paid attention to introduction and employment of CD only. Indeed, less attempts have been made to determine and highlight the quality factors and aspects of CD in practice. However, a few studies have referred to this issue.

Some studies focused on the barriers and challenges of employing and quality of CD in real environments. “organizational challenges” was reported as a serious challenge in the CD pipeline [5]. Another study [13] technical, procedural and customer-related challenges of CD have been addressed and the details of each were explained. For instance, issues with CD downtime, problems and limitations of automatic test process and configuration related problems are listed as technical challenges.

Another study tried to create a trade-off between risk of lower release quality and time-to-market while adoption of CD

[7]. Agile practices and their impacts on employing of CD were investigated in another study. This study showed that while some Agile practices like TDD, Pair testing, and customer involvement and collaboration have a positive and significant impact on the CD, some others like Pair programming have not such impact.

In another study, a new eco-system, Rugby, was proposed to support the CD life cycle and facilitate its pipeline [12]. The main focus of this study was to indicate the impact of Agile approach on the CD pipeline. The proposed eco-system defined some new roles such as team leader, project leader, customer, and developer to support and facilitate the CD adoption in real environments. The results of this showed the increase of frequency and quality of interactions between development team and customer party.

In another study, some of the adaptable quality metrics of CD were addressed. These metrics have been categorized as project level, product level, and pipeline level. The addressed metrics are suitable to be used in evaluating quality of CD.

In sum up, literature review shows that only a few studies focused on quality aspects of CD. This indicates a research gap that can be fulfilled by conducting proper research studies in practice. Focusing on this gap, the next section provides some quality factors that may affect the process of CD in practice.

### IV. OUTLINE OF CD QUALITY

Conducting a qualitative research study led to collection of proper data related the topic under study, CD quality aspect. Data collection and analysis are ongoing at the time of this writing. However, some aspects of the results can be showed in this article. This section provides the main findings of this study. However, the details of each aspect and evaluation of the findings will be provided in another article in future.

Data analysis showed that quality of CD is impressed by four different aspects including People, Process, Organization, and Tools, as shown in Fig. 2. These aspects are the high level abstract for various quality factors. Indeed, each of them

consists of several quality factors which together impress the quality of CD in practice.

'People' category mainly indicates that people related issues are important factors that impress quality of CD. People relationship is so critical in performing CD since this practices connect both technical and business parties. Furthermore, the relationship between development team members also is important, because it seems that collaborative teams conduct CD in a better quality.

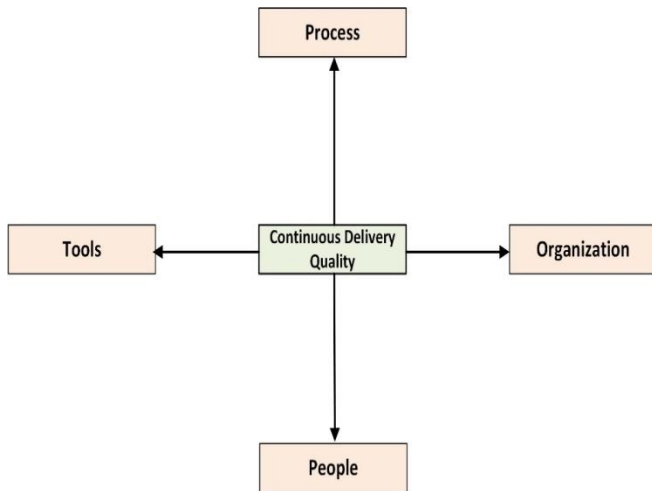


Fig. 2. The outline of quality aspects of CD.

Usually the people involved the CD process having different level of experience and so this aspect can impress the quality of CD too.

'Process' of CD has a great impact on quality of this practice. Various activities included in this process such as TDD, frequent testing, mechanisms used for requirement prioritization, and daily continuous integration seriously needed to be perform in the professional manner. Therefore, any weakness in doing such activities results in low quality of CD directly.

In 'organization' category the main focus in on the organizational culture and its related issues. Existence of culture of CD in organizational processes is compulsory to achieve the desired quality of CD. Also, providing mechanisms to manage the potential technical and human related risks greatly can lead to better quality of CD. Moreover, quality control and assurance and its process positively impress the quality of all the involved practices generally and CD particularly.

'Tools' category deals with tools related issues. For instance, automatic facilitates directly accelerate the process of CD and avoid the human related errors in this practice. Also, existence of mechanisms for version controlling leads to reduce configuration related defects.

In sum up, it seems that quality of CD depends of various technical and human related activities. However, more data analysis is necessary to explore the details of the above mentioned aspects, as noted earlier.

## V. CONCLUSION AND FUTURE WORK

CD is one the most important practices which recently is widely used in software projects. This practice focuses on the ability of software release at any time. CD defines a sequential set of activities to facilitate the release process. Quality of CD directly impresses the quality of development process. To explore the quality factors and aspects of CD, a qualitative study has been conducted. Initial data analysis showed that quality of CD is impressed by four aspects including People, Organization, Process, and Tools related factors. Each of these aspects by involving some quality factors may lead to better quality of CD in practice and real environments.

For the future work, the authors intend to employ the proposed model in two case studies to evaluate its usefulness and applicability in an empirical study.

## ACKNOWLEDGMENT

The authors would like to express their gratitude to all the participants of the research.

## REFERENCES

- [1] K. Beck et al. (2001, May 2014). Agile Manifesto. Available: [www.agilemanifesto.org](http://www.agilemanifesto.org)
- [2] D. Cohen, M. Lindvall, and P. Costa, "An introduction to Agile methods," *Advances in computers*, vol. 62, pp. 1-66, 2004.
- [3] T. J. Gandomani and M. Z. Nafchi, "An empirically-developed framework for Agile transition and adoption: A Grounded Theory approach," *Journal of Systems and Software*, vol. 107, pp. 204-219, 2015.
- [4] T. J. Gandomani and M. Z. Nafchi, "Agile transition and adoption human-related challenges and issues: A Grounded Theory approach," *Computers in Human Behavior*, vol. 62, pp. 257-266, 2016.
- [5] H. C. Esfahani, "Transitioning to Agile: A Framework for Pre-Adoption Analysis using Empirical Knowledge and Strategic Modeling," PhD, Graduate Department of Computer Science, University of Toronto, Canada, 2012.
- [6] L. Chen, "Continuous delivery: Huge benefits, but challenges too," *IEEE Software*, vol. 32, no. 2, pp. 50-54, 2015.
- [7] J. Humble and D. Farley, *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation* (Adobe Reader). Pearson Education, 2010.
- [8] G. Schermann, J. Cito, P. Leitner, and H. C. Gall, "Towards quality gates in continuous delivery and deployment," in *Program Comprehension (ICPC), 2016 IEEE 24th International Conference on*, 2016, pp. 1-4: IEEE.
- [9] Atlassian. (2016, Dec. 2017). Continuous Delivery. Available: <https://www.atlassian.com/continuous-delivery>
- [10] T. Dingsøyr and C. Lassenius, "Emerging themes in agile software development: Introduction to the special section on continuous value delivery," *Information and Software Technology*, vol. 77, pp. 56-60, 2016.
- [11] G. G. Claps, R. B. Svensson, and A. Aurum, "On the journey to continuous deployment: Technical and social challenges along the way," *Information and Software technology*, vol. 57, pp. 21-31, 2015.
- [12] 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: IEEE.
- [13] 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*, 2014, pp. 42-50: ACM.