

Agile Software Release Management



Author: William R. Cuesta
 Advisor: Dr. Héctor L. Cruzado
 Graduate School

Abstract

The Agile Software Release Management is a process standardization focused on how software is released in a manufactory using agile methods. The paper includes the development process with the data collected, the tools used, and the analysis of the results. The project's main objectives were accomplished, however continuous evaluation of software release has to be undertaken.

Introduction

The main purpose of this project is to design and implement a software release process applying Agile methods. Agile methods promote frequent testing and adaptation in an organize teamwork environment [1]. Software used in worldwide manufactories are very critical to the company and for this reason is very important to have an efficient process that can be used in any region.

This software release process will be deployed in a specific region with the aim to be deployed worldwide after validating the benefits of the new process.

- The following is the list of the main objectives of the project:
- Reduce man hours to complete a software release process.
 - Reduce bugs in software.
 - Record useful data.
 - Reduce human made errors.
 - Facilitate software availability.
 - Avoid unexpected shutdowns or functionality of the software.
 - Decentralize the software release management process.
 - Improve software development.

Background

The current software release management process involves that each member of the team compiles their code in their own machine. After all the different components are compiled, they are dropped in a common folder that can be assessed by the client. To install all the software components, the client needs to manually execute a series of scripts that will install all of the software components. To establish a baseline, some metrics were collected to help determine the success of the agile software release process implementation. The metrics that will be used will be the following:

- Bugs found in software by release.
- Delayed software releases.
- Human resources needed per release.

Problem

In any worldwide technology company, software is very crucial. Development and maintaining a software involves more than just coding in a computer. A very important part of a software life cycle is the release process, phase that tends be overlooked given that the main focus is code development. A bad or inefficient release process can lead to malfunctions and shutdowns which will impact company performance.

In the case of a worldwide company, the software release process need to be standardized. Sometimes it can be difficult due to diverse cultural aspects.

Table 1 represents the data collected to be used as baseline. The main goal is to reduce human resources, software bugs and delayed software release.

Table 1
Baseline Metrics

Metric	Quantity
Bugs found in software by release	12 (avg)
Delayed software releases	3 (total)
Human resources per release	240 (hours)

Methodology

The Agile methodology has become popular in software development because the requirement and solutions go thru a management process that is in constant inspection and adaption between the team. It needs to be very clear that Agile is not related to micro management and is related to lightweight and constant changes adaptation [2].

Some of the agile techniques and methods that can improve the software release process are the following:

- Continuous testing and integration of new software features.
- Multiple internal software releases before a public release.
- User friendly interface to facilitate client's installation and integration.
- Automatization of manually operation realized by the team members.

Automatization Methods

Automatization is a key element in this project because it will affect some of the objectives established. By doing automatization, human made mistake can be reduced and the human resources are reduced. If the resources and mistakes were reduced, the team can be focused in software development, indirectly reducing the bugs in the software.

Table 2 represents the areas of automatization with the tool used for each area.

Table 2
Automatization areas

Area of automatization	Tools used
Source code repository	GitHub
Source code compilation	Jenkins
Software installation	Wix

- Source code repository - is where all the source code of the software is stored and located. The tool selected for this area was Git, an open source version control system that is centralized. Git is designed specifically for developers and is team oriented. Multiple members of the team can work on the same source code at the same time and, when the tasks are completed, is merged to a centralized location [3].
- Source code compilation - is where source code (human readable) is translated to machine language. Compilation varies on time and it depends on code complexity and machine resources. The tool selected for this area is Jenkins, an open source system that automates the process of compiling code. To Jenkins works correctly, it needs to implement Git. Jenkins retrieves the source code form Git and then automatically compiles the source code in a centralize location. If any error or issue is found during compilation, a notification will be sent to the team members [4].
- Software installation - is where the client or user installs the software in their systems. The tool selected for this area is Wix, a software toolset that builds graphical user interface installers. Generating graphical user interface will facilitate the installation process and reduce the changes of mistakes during installation and configuration of the system.

Results

Table 3 presents the data collected after the new software releases management process. Is important to understand that only one software release has been done after the new implementation. Therefore software release will be determined after multiple releases.

Table 3
Result Metrics

Metric	Quantity
Bugs found in software by release	2
Delayed software releases.	To be determined.
Human resources per release	40 (hours)

Conclusion

To determine if the project was a success, the main objectives need to be evaluated. All the main objectives were accomplished successfully. The only part that will not be clear after multiple interactions of software releases will be the number of delayed releases. The major impact of this project was the amount of human resources reduced thanks to the automatization of multiple areas that were done manually.

The project was a learning experience for the whole team. Modern technologies were introduced, and new methodologies were also introduced. The biggest lesson learned from this project is the ability to understand and apply agile techniques to different work areas.

References

- [1] K. S, "Three Major Trends in Software Release Management You Should Adopt," CMCrossroads, 23-Oct-2013. [Online]. Available: <https://www.cmcrossroads.com/article/three-major-trends-software-release-management-you-should-adopt>. [Accessed: 15-May-2018].
- [2] G. Kim, P. Debois, J. Willis, J. Humble, and J. Allspaw, The DevOps handbook: how to create world-class agility, reliability, and security in technology organizations. Portland, OR: IT Revolution Press, LLC, 2017.
- [3] J. F. Smart, Jenkins: the definitive guide. Sebastopol, CA: OReilly Media, 2011.
- [4] . Chacon and B. Straub, Pro Git. Berkeley, CA: Apress, 2014.