Special Issue Published in International Journal of Trend in Research and Development (IJTRD), ISSN: 2394-9333, www.ijtrd.com

Impact of Software Metrics on Continuous Delivery Framework

[1]Anwin Varghese, [2]Dr.Rohni V and [3]Dr. Prabu P,
[1,2,3]Computer Science Department, Christ University, Bangalore, Karnataka, India

Abstract— Continuous Delivery (CD) is a software development discipline that imbibes fast and more frequent software releases. The deliverable is incrementally built on the implementation workflow, starting from code development and ending with software product in production for consumption. 65% of software developers, managers and executives report that their organizations have started down the path to CD. The benefits of using a CD implementation are faster time to market, better quality of product, competitive advantage, and higher customer satisfaction and reduced cost of development. CD implementations across organizations are done using CD framework pipelines. To achieve an ideal CD implementation, the delivery framework should achieve important traits as moving away from silos, monitoring as part of the framework and one that leverages tools as a service. An ideal CD framework depends on high levels of build, integration, test, and deployment automation.

CD implementations are widespread across IT organizations of all sizes, from start-ups to the large multi-national companies. These implementations use different CD framework models like agile framework, Scaled Agile Framework (SAFe), Disciplined Agile Delivery (DAD), Composable Fault Tolerance Framework (CFT), Test Orchestration Framework & Large-scale Scrum Framework (LeSS). We need to be measure these implementations to understand its benefits that it brings on board with respect to improvements in different aspects like culture & organization, design & architecture, build & deploy, test & verification and information & reporting. This requires measuring the CD framework used in the implementation. There are generic parameters that an ideal CD framework should possess – reliability, integrity, scalability, robustness and security. These parameters can be assessed by measuring the metrics that can quantify the parameters mentioned earlier.

A software metric is a standard of measure of a degree to which a software system or process possesses some property. Towards quantifying the parameters desired by an ideal CD framework, we can deduce the software metrics to be measured to a handful of them. This paper aims to understand the important software metrics that are important to measure the effectiveness of CD frameworks used as a practice across the different IT organizations.

Keywords: *Continuous Delivery, framework, parameters, software metrics.*

I. INTRODUCTION

A software product is something that we use in our daily lives. Ever thought how it is created? A software product is the result of a good number of engineers working on a requirement that was set by the product owner, with different aspects like scrum management, build management, test management, code management, quality management, artifact management, project management, time management, release management etc. We see that the overall aspects involved with creating a software product is complex.

We have software organizations that create these software products that are available to us. The software market has rapidly expanded & there are lot of these software organizations churning out software products day in & day out. Expectedly, there is a lot of competition & focus is on how fast the software product can be made available for use by the masses. There is lot of economic & business related benefits that an organization can derive by having the software product made available first in the market. This need to be fast paced drives organizations to produce the software products as faster as possible.

The complexity of software product creation & the necessity for the software products to be made available as faster as possible are two conflicting points that makes organizations to look for possible solutions. Continuous Delivery (CD) is a solution to this problem. CD helps organizations provide with a framework that can be used for the software production. CD imbibes a cultural shift in the organizations such that it helps faster pace in software releases. It also promotes more frequent software releases. Each deliverable is incrementally built on the implementation workflow, starting from code development and ending with software product in production for consumption. The focus is on quick small releases that can be consumed by customers making the product possess latest features as required by customers.

II. CD FRAMEWORKS

With evolution of CD as a discipline, we have quite a number of CD frameworks available.

A. Waterfall Framework

Waterfall framework is one in which the progress of the delivery cycle of a feature release is taken step by step. The framework follows the same principle of the waterfall software development life cycle. The decision of moving forward with the next step of the delivery for feature release depends on the result of the current step [1].

B. SAFe Framework

SAFe is an interactive knowledge base for implementing agile practices at enterprise scale. At the Team level, SAFe looks a lot like Scrum, including of extreme programming practices. Not every sprint necessarily produces a potentially shippable increment, but this should happen frequently, possibly after hardeningsprint. Lean thinking, the Principles of Product Development Flow and the extensive benefits that agile development (Agile Manifesto, Scrum, XP technical practices, Kanban) all play important roles in defining the principles and practices of SAFe framework. The framework is one in which the progress of the delivery cycle of a feature release is taken step by step [1].

C. LeSS Framework

LeSS Framework is regular Scrum applied to large-scale development in very large organizations [1].

D. DAD Framework

Disciplined Agile Delivery (DAD) is a framework which ensures scalability of agile process. The framework has various characteristics such as: people first, explicit scaling support, goal driven, enterprise awareness, risk and value driven, delivery focused, IT solution focused, agile, hybrid and learning oriented. All these characteristics if embedded properly in the product development procedure will ensure its reliability. DAD framework is the second generation of agile framework; therefore it is an amalgamation of all the excellent features of different agile methodologies. Goal-driven characteristic of DAD, make it aware of the issues that is associated with each goal. This awareness of issues leads the team to look for solution beforehand. Hence at the time of deployment, the realized goals must be associated with the underlying issues. And these issues could be addressed to overcome complications arising due to them. The delivery focused characteristic, manages the post-delivery activities, which is necessary to handle the issues occurring during deployment [1].

E. CFT Framework

The CFT framework integrates previously constructed components and also has techniques to tolerate fault. The errors in continuous delivery could be minimized through automation. CFT framework is based on workflow model composed of various execution entities. The prime aim of this framework is to uncover errors and make the system fault tolerant. The idea is increment validation, which follows failure identification, automatic fault injection, integration of fault detection/recovery with protected operation and validation of fault tolerance. After increment validation, user provided control data is verified, followed by design of automated testing environment. Having all the future failures sorted out beforehand keeps the team aware of the deployment issues [1].

F. Test Orchestration Framework

Test orchestration framework analyses the codes, selects the tests to be conducted, schedules the tests, prepare the environment, executes the tests, analyse the results and finally deploy. Primarily, its aim is to make software reliable and bug free [1].

III.PARAMETERS IN CD FRAMEWORK

With quite a number of CD frameworks available for organizations to make use for their implementation, there is a need to understand which one suits each better. In this regard, we need to measure the CD frameworks that can help us with quantifiable justifications to choose a particular CD framework. In order to measure any CD framework, we need to list down the parameters that are important for the CD framework.

A parameter, generally, is any characteristic that can help in defining or classifying a particular system. That is, a parameter is an element of a system that is useful, or critical, when identifying the system, or when evaluating its performance, status, condition, etc [2].

For any CD framework, the parameters must help with the final outcome of faster, cleaner & error free software releases. The parameters – productivity, agility, quality, value delivery, scalability, dependability, performance, predictability etc. are those that certainly help. With these parameters, we need to understand the various quality factors that help us gauge these parameters for any given CD framework.

Productivity is an average measure of the efficiency of production. It can be expressed as the ratio of output to inputs used in the production process, i.e. output per unit of input [3]. Productivity can be interchangeably used with efficiency. It can be measured by calculating the software metrics – Contribution Margin, Organizational Stability and Team Velocity versus capacity. Contribution Margin is the measure of contribution of the resources involved towards the complete success of an end to end CD implementation release. Organizational stability is the measure of resources that have stayed put in the organization throughout the release lifecycle. With attrition rate high in organizations, this metric is critical to understand the effect of the CD framework implementation on the organization stability. Team Velocity is a measure of the speed at which the complete team is able to complete the tasks involved within the release lifecycle. Capacity is the measure of the team resource in head count.

Agility is the ability to be quick and graceful. It is often related to another parameter performance [4]. It can be measured by calculating the software metrics – product ownership, release planning & tracking, planned versus actual velocity, planned versus accepted stories, percentage of stories accepted, cycle time etc.

The quality of a product or service refers to the perception of the degree to which the product or service meets the customer's expectations [5]. Quality has no specific meaning unless related to a specific function and/or object. Quality can be measured with number of support call volume, percentage of unit test coverage, number of defects, number of test cases, number of automated test cases, number of refactors etc.

Value delivery is the most important parameter as far as the end customer is concerned. The delivery to the customer has to fit the purpose the. In essence, for any service you deliver to a customer, having value delivered means that the customer can enhance the performance of their own assets. The software metrics that help us with this parameter are number of releases, number of value featured points delivered, release date percentage & architectural re-factor.

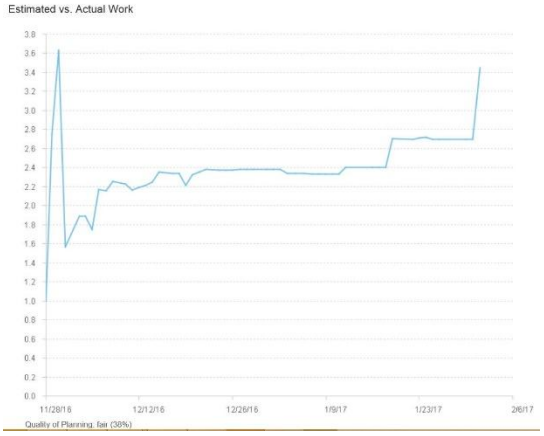
Scalability is the capability of a system to handle a growing amount of work, or its potential to be enlarged in order to accommodate that growth [6]. The software metrics that help this parameter could be percentage of automation, resource capacity availability & time taken to reproduce the system. Scalability goes hand in hand with predictability. The predictability measure helps with the scalability requirements.

With all these software metrics that we have outlined, we zero in on a few major ones to help us understand the overall measure of the CD framework. We can consider these software metrics – estimated versus actual work, open versus closed items, burn-down, Mean time to recover (MTTR), customer satisfaction survey, backlog versus requests, defect ratio, cycle time, velocity, number of releases, and percentage of automated test cases.

IV. CASE STUDY

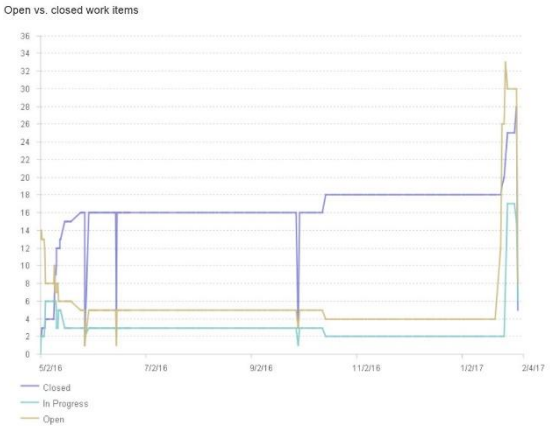
We have the following case study involving teams working in product life cycle solutions project which is involved in creating the tool chain for the product teams. The team follows agile & has the SAFe CD framework implementation. We can measure the CD framework implementation by generating the software metrics that we outlined earlier.

The productivity, quality & agility can be measured by generating the metrics estimated versus actual work, open versus closed items, burn-down. Here we make use of the RTC software tool to generate the results.



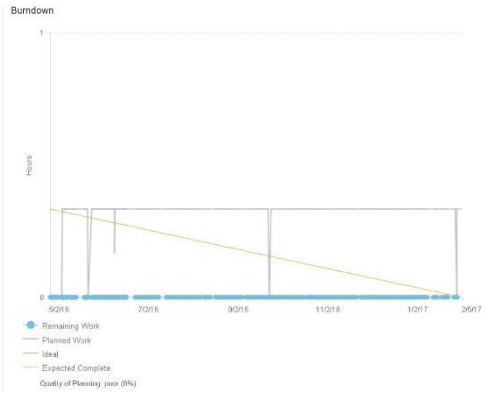
We see that the quality of planning is at 38% & he graph displays the estimated versus actual work.

Next we measure the open versus closed items. This again is a measure of the productivity & agility using this CD framework.



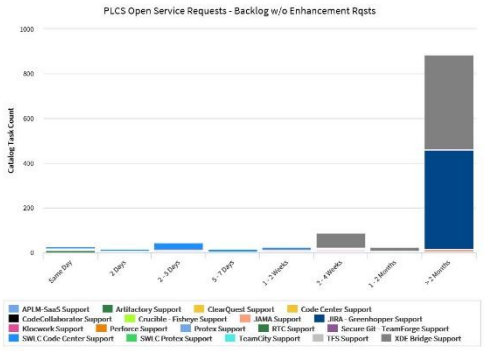
The items here refer to stories – the features that are requested. Open means that the work on the feature is ongoing. Closed means that the work is complete & ready to be released or released to customers.

Next, we can look at the individual burn-down of the work or story for a given sprint.

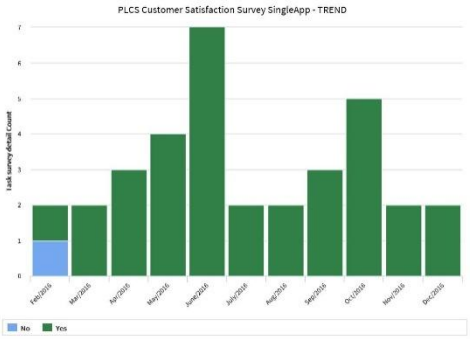


The burn-down tells us the velocity at which the task was completed over the sprint period.

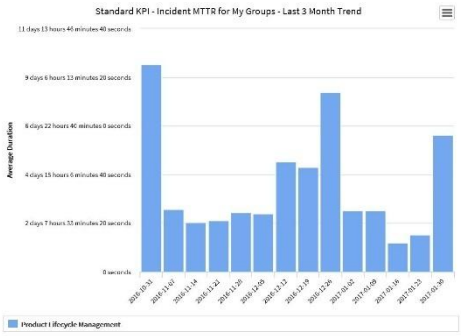
Backlog versus customer requests for new feature is a measure that helps us understand the impending work that requires to be completed & helps us with planning & predictability.



Another important metric is the customer satisfaction measure. It is a reflection of the value delivery of the CD framework.



Mean time to recover (MTTR) is an important measure to understand the integrity & agility of the CD framework.



These measures help us understand how the CD framework implementations have been able to achieve the various required must have parameters. It helps us gauge the effectiveness of the CD framework implementation.

CONCLUSIONS

CD framework is an extremely important discipline in software engineering & we are progressing to a time with almost all the organizations with software products or services implementing CD framework. However, these organizations need to measure the effectiveness of having implemented these CD frameworks. The parameters for the measurement of CD framework depend on the organizational requirement & focus. It certainly varies. However, universally, we can set few parameters that are inevitably required to be in any CD framework. These parameters are the foundations on which the CD framework can be measured. Software metrics that provide us quantifiable values, help gauge the various parameter measures for the CD framework.

References

[1] Anwin Varghese, Dr. Rohini V and Dr.Pabu P "Smart Continuous Delivery Framework for Software Releases", International Conference on Recent Trends in Engineering and TechnologyISBN 978-93-84468-80-4, Pattaya (Thailand) Dec. 14-16, 2016.

[2] Wikipedia, "Parameter," [Online]. Available: https://en.wikipedia.org/wiki/Parameter
[3] Wikipedia, "Productivity," [Online]. Available: https://en.wikipedia.org/wiki/Productivity
[4] Vocabulary, "Agility," [Online]. Available: https://www.vocabulary.com/dictionary/agility
[5] Wikipedia, "Quality(business)," [Online]. Available: https://en.wikipedia.org/wiki/Quality_(business)
[6] Wikipedia, "Scalability," [Online]. Available: https://en.wikipedia.org/wiki/Scalability