IJFANS INTERNATIONAL JOURNAL OF FOOD AND NUTRITIONAL SCIENCES

ISSN PRINT 2319 1775 Online 2320 7876

Research paper UGC CARE Listed (Group -I) Journal Volume 11, Iss 3, Mar 2022

The Survey Paper on Inevitable Metrics in Agile Software Development

[1*] T.Ravi Kumar, [2] Kalyana Kiran Kumar,, [3] Suneelgoutham Karudumpa, [4] Ch.Rajasekhara Rao,[5*] Balamurali Pydi

[1] Dept of CSE, Aditya Institute of Technology and Management, Tekkali, AP, India.
[2] Dept of EEE, Aditya Institute of Technology and Management, Tekkali, AP, India.
[3] Dept of EEE, Aditya Institute of Technology and Management, Tekkali, AP, India.
[4] Dept of ECE, Aditya Institute of Technology and Management, Tekkali, AP, India.
[5] Dept of EEE, Aditya Institute of Technology and Management, Tekkali, AP, India.
[*]Corresponding Author: Balamurali Pydi balu_p4@yahoo.com

Abstract—

Software Industry, now- a-days has been changing its magnitudes frequently. With the changing phenomena's, the scrum masters should notice the current trending developments with Agile Software and its rapid growth. With these, we should alsotake effort to consider, what are the metrics that are inevitable in agile softwaredevelopment, which they make the software efficient in various parameters like size of software, time-line attributes, cost of the project etc.

As *"deliver a working software"* is one of the key principles of Agile Manifesto, the following metrics discussed in this paper may helpful to deliver a good software product in terms of all quality metrics. In one technical report by authors Suzanne Miller, Mary Ann Lapham and Eileen Wrubel, they stated that "*Agile Metrics: Progress Monitoring of Agile Contractors*"[1]. By this we can understand the importance of metrics in the development of a project.

The metrics classified in this paper are as health metrics, product development metrics, release metrics and technical metrics. Along with these metrics, few agile terminology is also explained.

This paper trying to cover metrics that are following by industry today and those are needed to put strong effort in a softwareproject.

***Methodology*—**The survey is conducted based on qualitative and descriptive data analysis methods.

***Keywords*—** Software metrics, Preliminary metrics, Release metrics, Product development metrics, Technical metrics.

ISSN PRINT 2319 1775 Online 2320 7876

Research paper **© 2012 IJFANS. All Rights Reserved, UGC CARE Listed (Group -I) Journal Volume 11, Iss 3, Mar 2022**

Preliminary Metrics:

Predictability seems to be paramount. Clients want teams to get good at making and keeping promises, consistently delivering working, tested, remediated code at the end of each sprint. A team that is not predictable isn't "bad" – but they aren't predictable. Without stable predictable teams we can't have stable predictable programs, particularly when there are multiple dependencies between teams[2].

This post focuses on agile metrics forpredictability. To overcome these, teams should be more focused on the followingattributes or metrics:

Cumulative Flow Diagram (CFD) is an areachart that shows the current status: (howmuch work has been done, what is in progress and how much is still waiting to be done in the backlog). A Cumulative Flow chart helps to gain insight into issues, cycle time and likely completion dates [3]. It is also indispensable for identifying bottlenecks. of work items for a particular time interval. [4].

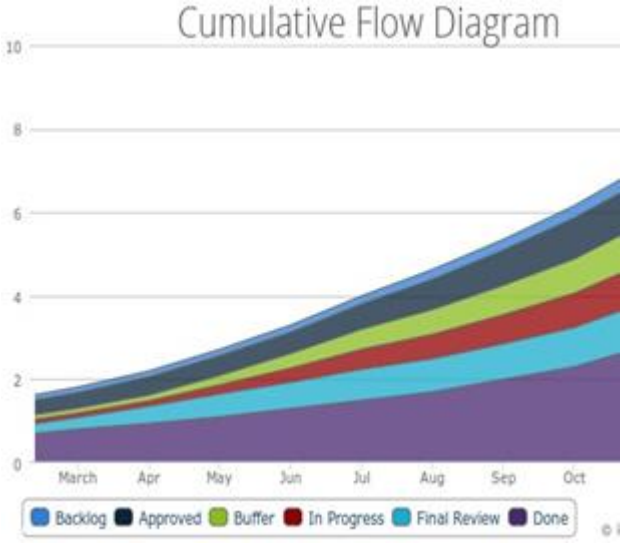

Figure 1.Example CFD

Visual Control Charts: Visual control is a management technique employed in many places where information is communicated by using visual signals instead of texts or other written instructions[5]. The design is deliberate in allowing quick recognition of the information being communicated, in order to increase efficiency and clarity. The most important information radiator in visual management is the Task Board. (In Scrum, Agilists sometimes call task boards as Scrum boards). The task board has the mission of visually representing the workthat is being done by the team. They are the most complex and versatile artifact: a physical task board is a "living" entity that has to be manually maintained.

ISSN PRINT 2319 1775 Online 2320 7876

Research paper UGC CARE Listed (Group -I) Journal Volume 11, Iss 3, Mar 2022

Flow Efficiency is one of the priliminary metrics that focuses on the work flow of the designers, developers and testing teams.

Work Item Types: Microsoft Teams use the work item types (WITs) provided with the Agile process to plan and track progress of software projects[6]. Teams define user stories to manage the backlog of work and then, using the Kanban board, track progress by updating the status of those stories.

Blocker Clustering*:* The key to performing blocker analysis is to capture the cases of impediments on the *kanban or Scrum boards*. Capturing the blocked time is important for prioritization of the most impactful blockers.

Release Metrics:

This group directs focus on identifying impediments to continuous delivery.

Escaped Defects*:* An escape is a defect that wasn't discovered by test teams. Instead, the defect was found by customers. When problems are exposed by customers, they are quite costly. If a bug is discovered after the team said the work was done then we want to track that. Prior to hitting "done", it's not really a bug – it's just unfinished work. Shipping buggy code is bad and this should be obvious[7]. Continuously delivering buggy code is worse. *"Let's get the code in good shape before we start pushing deploys out regularly".*

Release Planning: Planning and estimating in the agile world depend on a single key metric: *the development team's velocity*[8], which describes how much work the team can get done per iteration. (We describe velocity in detail separately.) Given a team's known velocity for its last project (if it is known), a release plan represents how much scope that team intends to deliver by a given deadline.

Release deadlines are often fixed, imposed externally by such things as tradeshows, accounting pressures, or contractual obligations. But since the goal is to get working software into the users' hands as quickly as possible in order to make "course corrections" as soon as possible, every effort is made to keep release software development cycles as short as possible. Agile release cycles should certainly be kept shorter than a year, and are often as short as 6 months or 3 months. A release is, in turn, made up of iterations. For a given project, iteration length will typically be fixed at a length somewhere between a week and a month. If the release is in 6 months and iterations are going to be 2 weeks each, the release will consist of 13 iterations.

Cost per Release: Cost is a product of time and people (team members). Add more time, and you add cost for employing people for longer. Add more team members, and you increase cost to deliver the same business value. The things we really want to avoid. This is why Agile principles believe in

Research paper UGC CARE Listed (Group -I) Journal Volume 11, Iss 3, Mar 2022

fixing time and team members and allowing scope to be the variable component.

Release Net Promoter Score: Net Promoter Score is a loyalty metric that quantifies how customers feel about your product and what you can do about it. The great thing about NPS compared to other quantitative engagement tools like customer satisfaction surveys, is that it's much, much simpler to rapidly gather feedback that gives a consistent and measurable result. It's only one question, and the score can be measured against other products[9].

Product Development Metrics:

These help measure alignment of productfeatures to user needs.

Customer / Business Value Delivered: Customer value increases the likelihood that customers will continue to use your product ('make it stick better'), like: *Improving usability in an application to make it easier to use (and reduce frustration).Adding a new feature that is commonly requested by users.*

Risk register and risk burn down: Fundamentally speaking, *risk* is something that may occur and cause unexpected or unanticipated outcomes. Bear in mind that the outcome may have a positive or a negative effect. A positive effect is an opportunity, while a negative effect is a threat. Traditional project management techniques would recommend a risk register to monitor for managing and controlling risks.

A simple risk register should consist of the following attributes[10]:

Description of risk: A one- or two-line overview of the risk. It should be simple and easy to comprehend.
Date identified: Date when the risk was identified.
Likelihood: Estimated probability of occurrence of the risk.
Severity: The severity of the risk is assessed based on impact of the undesired outcome.
Priority (optional): This could be either given an independent value or set as a product of likelihood and severity (above). A high-severity risk with a high likelihood should receive more importance than a high-severity risk with a low likelihood.
Owner: The person who manages, controls, and takes action in response to the risk.

Action: The response defined to manage/control the risk.
Status: Indicates whether the risk is open or closed or being monitored.

Another interesting technique, introduced byJohn Brothers (*Agile Times*, 2004), relies on using a Risk Burn-down Chart. As elaborated by Mike Cohn in his article, the risks are collated into a table similar to a riskregister. It consists of the followingelements:

Risk: Description of the risk in a few lines.
Probability: Likelihood of the risk.

ISSN PRINT 2319 1775 Online 2320 7876

Research paper **© 2012 IJFANS. All Rights Reserved, UGC CARE Listed (Group -I) Journal Volume 11, Iss 3, Mar 2022**

Size of loss: Amount of time lost should the risk occur. This could be represented in daysor story points.

Exposure: This is computed as a product of the probability and size of loss (above).
An example is shown below. The consolidated risk exposure is shown in the final row of the risk register.

Risk	Probability	Size of loss	Exposure
Failure of network connectivity with partner systems	65%	10 days	6.5
Dependent systems are incompatible during integration	80%	12 days	9.6
Prototyping has inconclusive evidence for final design	50%	10 days	5
Lack of comprehensive data from live sites	70%	5 days	3.5
User interfaces stability across all browsers	30%	10 days	3
			27.6

Figure 2.Risk Analysis Example

This risk register is reevaluated at every sprint meeting. Its values are adjusted based on the current assessment of the existing and new risks. This would define a new value for the consolidated risk exposure. The Risk Burn-down Chart can be created by plotting the consolidated risk exposure across the number of sprints run by the team. An example is shown below:

Sprint	Exposure
1	27.6
2	20
3	21
4	15
5	10
6	5
7	0

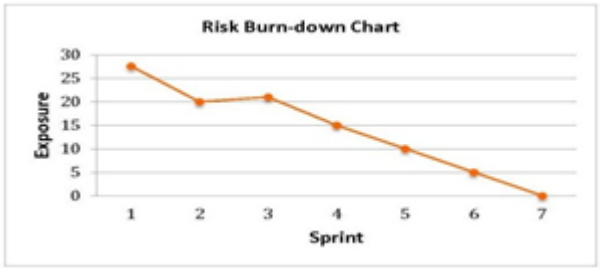

Figure 3.BurnDown Chart Example

Product forcasting is another important factor in which it analyze what the scope of the product in future and estimates what features may append to the present developing product. This helps the product development company in future to establish its sustanability.

ISSN PRINT 2319 1775 Online 2320 7876

Research paper UGC CARE Listed (Group -I) Journal Volume 11, Iss 3, Mar 2022

Code Metrics:

Caluclating defect density is the primary agenda in software testing. It defines as follows: It is the number of defects confirmed in software/module during a specific period of operation or development divided by the size of the software/module. It enables one to decide if a piece of software is ready to be released. Defect density is counted per thousand lines of code also known as KLOC.

$$\text{Defect Density} = \frac{\text{Number of Defects}}{\text{Size}}$$

Adherance to coding standards make thecode undestandable and effcient. If any case any problem or ticket received by the clients

Code churn is when an engineer rewrites theirown code in a short period of time. Softwareteams know that business value delivered is the true sign of their effectiveness. Tracking code churn helps visualize your team's code quality.

Collective code ownership[11] breaks a code base up into modules (classes, functions, files) and assigns each module to one developer. Developers are only allowed to make changes to modules they own. If they need a change made to someone else's module they need to talk to the module owner and get them to make the change[12].

Trending Software Project Management tool:

Jira[13] is a proprietary issue tracking product, developed by Atlassian company. It provides bug tracking, issue tracking, and project management functions. It has been developed since 2002. According to one ranking method, as of June 2017, Jira is the most popular issue management tool. Jira is used for issue tracking and project management by over 75,000 customers in 122 countries around the globe. Some of the organizations that have used Jira at somepoint in time for bug-tracking and project management.

Jira is offered in three packages:

- **Jira Core** is intended at generic projectmanagement
- **Jira Software** includes the basesoftware, including agile project management features (previously aseparate product: Jira Agile)
- **Jira Service Desk** is intended for use byIT or business service desks.

after some years, then it is easy to understand by the developers of the company to rewrite or modify the code.

According to Atlassian, there are five metrics that are common in any reliable software development are as follows:

ISSN PRINT 2319 1775 Online 2320 7876

Research paper UGC CARE Listed (Group -I) Journal Volume 11, Iss 3, Mar 2022

Sprint Burndown:
Scrum teams organize development into time-boxed sprints. At the outset of the sprint, the team forecasts how much work they can complete during a sprint. A sprint burndown report then tracks the completion of work throughout the sprint.

Epic and Release Burndown:
Epic and release (or version) burndown charts track the progress of development over a larger body of work than the sprint burndown, and guide development for both scrum and kanban teams. Since a sprint (for scrum teams) may contain work from several epics and versions, it's important to track both the progress of individual sprints as well as epics and versions.

Velocity:

Velocity is the average amount of work a scrum team completes during a sprint, measured in either story points or hours, and is very useful for forecasting. The product owner can use velocity to predict how quickly a team can work through the backlog, because the report tracks the forecasted and completed work over several iterations–the more iterations, the more accurate the forecast.

Control Chart:
Control charts focus on the cycle time of individual issues–the total time from "in progress" to "done". Teams with *shorter* cycle times are likely to have higher throughput, and teams with *consistent* cycle times across many issues are more predictable in delivering work. While cycle time is a primary metric for kanban teams, scrum teams can benefit from optimized cycle time as well.

Cumulative Flow Diagram:
The cumulative flow diagram is a key resource for kanban teams, helping them ensure the flow of work across the team is consistent.

Buzzwords used:

Kanban Board: A Kanban board is a work and workflow visualization tool that enables you to optimize the flow of your work. Physical Kanban boards, like the one pictured below, typically use sticky notes on a whiteboard to communicate status, progress, and issues.
Burndown Chart: A burn down chart is a graphical representation of work left to do versus time. The outstanding work (or backlog) is often on the vertical axis, with time along the horizontal. That is, it is a run chart of outstanding work. It is useful for predicting when all of the work will be completed.

Conclusion

Metrics help the scrum teams to track their progress and help them to see the status of a project. In agile software development (ASD), metrics play consistent role of management. More than 80,000 software companies using the approach of ASD. That's why agile got tremendous importance in the software industry. The leading software industry till today, the Microsoft Corp. using the agile process. By knowing the importance of this agile development, this paper focused on the important metrics that are inevitable for software development.

ISSN PRINT 2319 1775 Online 2320 7876

Research paper **© 2012 IJFANS. All Rights Reserved, UGC CARE Listed (Group -I) Journal Volume 11, Iss 3, Mar 2022**

References:

[1]. Suzanne Miller, Mary Ann Lapham, Eileen Wrubel, and Timothy A. Chick-- https://insights.sei.cmu.edu/sei_blog/201 4/09/agile-metrics-seven-categories.html
[2].ANDREW-FUQUA https://www.leadingagile.com/2013/07/agile-health-metrics-for-predictability/
[3]. KanbanTool https://kanbantool.com/kanban- library/analytics-and-metrics/explaining- cumulative-flow-diagrams
[4].Atlassain-Software https://confluence.atlassian.com/agile/jira- agile-user-s-guide/using-a-board/using- reports/viewing-the-cumulative-flow- diagram/
[5]. Agile Perals Website https://agilepearls.wordpress.com/tag/visual-controls/
[6].Microsoft Official Website https://docs.microsoft.com/en- us/vsts/work/work-items/guidance/agile-process
[7]. G. Czibula, Z. Marian and I.G.Czibula, Detecting software design defects using relational association rule mining" Knowledge Information Systems,pp. 1- 33,2012
[8]. VersionOne Inc. " Agile Development Release Planning" https://www.versionone.com/agile-101/agile-management-practices/agile- development-release-planning
[9]. Atlassain Software https://www.atlassian.com/agile/how-to- prioritize-features-using-net-promoter- scores
[10]. Scrum Alliance Org.-"Risk Management In Agile" https://www.scrumalliance.org/community/articles/2013/2013-may/risk-management-in- agile
[11]. Martin Flower-"Code Ownership" https://martinfowler.com/bliki/CodeOwners hip.html
[12]. Wikipedia Source https://en.wikipedia.org/wiki/Jira_(software)
[13]. "Metrics for agile software development teams"-Frontrow Agile https://www.frontrowagile.com/blog/posts/6 9-30-metrics-for-agile-software- development-teams