Effective collaboration across the globe through digital dash boards and machine learning

Prabhulinga ManikReddy
Siemens Technology and Services Private Limited
Bengaluru
India
Prabhulinga@siemens.com

Jaikrishna Iyer
Siemens Technology and Services Private Limited
Bengaluru
India
Jaikrishna.Iyer@siemens.com

ABSTRACT

In a large sized project with globally distributed environment, it is challenging to have a common information radiator across all global locations and it is difficult to make sense of the long list of tasks displayed on a single dashboard. Different time zones make it difficult to collaborate and cater to a customer who demands frequent releases and transparency. This paper demonstrates a unique solution driven by a novel mix of digitization and machine learning to solve this ubiquitous problem.

After value stream mapping the team developed a tool backed with machine learning, helped us to deliver our concepts faster to market. We derived unconventional digital dash boards which helped in building collaborative environment across different time zones.

This paper also highlights the critical role of machine learning in project management, proposes improvements various in areas like task estimation, team velocity and turnaround time prediction.

This paper is targeted to agile practitioners who are interested in improving workflows and building intelligent process systems using new modern technologies in globally distributed environment.

CCS CONCEPTS

• **Software and its engineering->Software creation and management->Software development process management->Agile software development**

KEYWORDS

Value Stream mapping, Information Radiators, Machine learning.

ICGSE '18, May 27–29, 2018, Gothenburg, Sweden
© 2018 Copyright is held by the owner/author(s).
ACM ISBN 978-1-4503-5717-3/18/05.
https://doi.org/10.1145/3196369.3196391

1 INTRODUCTION

Global Configuration Project (GCP) started its development nearly a decade back using a waterfall model. The team started following Agile principles in the year 2012 and improved the key factors for project success like collaboration, communication, Trust, product quality, automated testing, etc. Since two years we have been adapting to Kanban.

1.1 Background of GCP

Project Organization: Development happens majorly in India and some of it in Germany. Each development team consists of seven to eight members who are architects, developers and testers. The U.S. team consists of the product manager and the domain experts who act as product owners and they are the interface between development team and end users.

Technologies: Initially, GCP was developed with client and server based technologies, however, since last four years, GCP is targeting a mobile platform with new technologies. Most of the teams develop features and proof of concepts for client.

Project Roles: GCP had traditional software project development roles like Project Manager (PM), Team Lead (TL), Developers, Architects and Testers. We had tailored roles after moving to Scrum [1]. But in Kanban[9] we switched back to traditional roles.

1.2 Reason for moving to Kanban

Some of the reasons why we moved from Scrum to Kanban:

1. We cannot add priority task in the middle of the sprint to the sprint backlog. Even the task critical one and if the task could not fit into the sprint replacing other lower priority tasks, one has to wait till the current sprint completes.

2. Many communication channels needed as all the POs were positioned in different global locations.

3. When the features were split across the teams, it was difficult for stakeholders to make sense out of dash boards of different Scrum teams. Customer was not able to predict probable timeline for feature completion.

We believe that Kanban dash board best represents the status of any agile team, any time. It particularly works well when,

1. Team is small
2. Task Allocation is well managed

1. **Team is small:** As agile suggests if the team is small, say maximum of 8 members, then the dash board is readable. As team size increases then dashboard becomes more complicated due to lot of tasks being displayed and thus takes more time to understand current project task assignments.

2. Task allocation is well managed: WIP limit plays very vital role in managing the tasks, keeping check on lead time and cycle time and delivering the features fast[4].

1.3 Challenges in Kanban

Following are the challenges we faced with respect to dash board complexity when we moved to Kanban.

1. Since our development teams and partners are not co-located, digital dash board made more sense than physical one. And hence we zeroed in to one of the best issue tracking tool with best visual representation.
2. Domain experts, leads and architects came up with 20 states in Kanban board. Too many states in kanban board is difficult to manage.
3. We could not hold on to WIP limits because of the following reasons
    a. WIP limits created concern to our customer as he found it difficult to prioritize the tasks what he wanted to get completed. Even the smaller tasks went
    b. There were many responsibilities, for example Database Architect (DBA), who will be assigned many tasks at a time.
4. When we gave away WIP limits, dash board became more complicated due to lot of information overload. Due to this, we could not fit all the details in one view.

    On the other hand, there is tough competition in the market. We needed to be very fast in converting ideas to delivered solutions, before our competitors would grab the opportunity. We had to see all the possibilities where we could reduce the time in concept to feature realization.

1.4 Flow of tasks in GCP

Any idea or solution to business problems can be termed as concept. Once the concept is approved by the customer, the development team will start building on the concept. The feature fulfilling most of the user's key requirements will be shipped to pilot users. This is key activity where users find bugs and give their feedbacks. During the development, our team also make sure that when unwanted situation occurs appropriate logs are populated.

When a bug or issue is created by user or the support representative, we follow the traditional approach of issue assignment in software industry.

Tester → Product Owner → Team Lead → Developer

The pros of this approach are,

1. There is a clear well defined channel that ensures that the issue gets assigned to the concerned developer.
2. All the stakeholders are aware of the current status of the issue and agree on assigning the issue to an appropriate developer.

However this approach also comes with major drawbacks,

3. Time is wasted due to unnecessary handshakes amongst stakeholders.
4. Any unavailability of a stakeholder in this chain will lead to issue assignment being delayed or worse, the issue remains with a stakeholder for long time.
5. The process is redundant.

In short the two major challenges for our development were

1. Create a custom dashboard to collaborate across the regional boundaries
2. Optimize the flow that spanned between different regions in the world.

Before we get into solution for these challenges, let us understand the basics of machine learning, which is a key ingredient to one of our solutions.

1.3 Machine learning

Machine learning (ML) is a broad computer technology. The most widely quoted definition is provided by Tom M. Mitchell, "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E" (REFERENCE REQUIRED).

With ML, we have the ability to create a model that can learn, find new patterns and make accurate predictions. Machine Learning is mainly classified into,

1. Supervised Learning
2. Unsupervised Learning
3. Reinforcement Learning

1. Supervised Learning: Supervised Learning algorithms analyze the input training data, creates an intermediate function model, which can be used to predict new outputs for similar inputs. This model is generally tested against a testing model for tuning the accuracy.

2. Unsupervised Learning: In unsupervised learning, the algorithms try to find hidden patterns and structures within the input data provided to it.

3. Reinforcement Learning: In reinforcement learning, the algorithm rewards the learning model every time it takes the right step through a feedback mechanism. Here the algorithm learns to react to the subjected environment.

In this report, we discuss a model created using supervised learning, where we predict the proper developer to whom the issue has to be assigned once the issue is created by the tester.

2 CHALLEGE 1: TIME TO MARKET

The tasks created took a lot of time to reach the developers. It had to pass through various levels to reach developers, as mentioned in section 1.4. Hence, the time taken to deliver our product to market also increased.

2.1 Solution: Value stream mapping

The team traced the flow of the issue from creation state to deployment state, in the system.In figure1 , it can be seen that there is huge time wasted in moving the issue/task created from one person to other.

The team brainstormed and came with the following analysis.

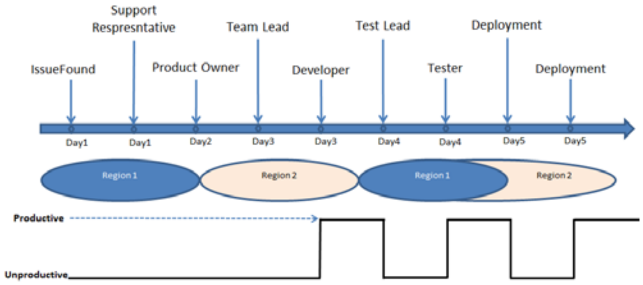


Figure1. Issue flow along with Regions and productivity graph

To eliminate this waste in the process, the team came up with the idea of Automated Issue Analysis and Assignment. We harvest the power of machine learning to identify the appropriate person responsible for the issue fix/enhancement. The type of machine learning involved here is re-enforcement learning.

Based on a set of weighted influential parameters, our model is able to identify the right person for the job. It works as follows:-

1. Issue is reported in Production/Integration Testing (IT)/ Unit Testing (UT) phases. All this data is currently fed in an issue tracking tool.
2. Appropriate parameters are read from this tracking tool by our machine learning application and the identification process begins.
3. The individual is identified and the task is assigned. If the issue assignment is improper, then the current stakeholder can change the value to the proper individual. Here the reinforcement learning ability of the tool helps it to learn from its mistakes and avoid improper future assignment.

To identify the proper individual, some main parameters in consideration are,

1. Module Strength/Experience
2. Technology Grip
3. Availability
4. Historical Turn Around Time (TAT) Index
5. Issue/Enhancement priority

All these parameters are defined with the entire stakeholder's agreement, so there is much lesser probability of disagreement. Using this approach, we have successfully brought down the time from 3.5 days to just 0.5 days, resulting in critical time savings. This is currently an automated process. When an issue is assigned, it is communicated regularly to all concerned individuals.

We rely on the historical data to enhance the accuracy of this tool. About 1.5 years of historical data has been fed as input to the model, trained and fine-tuned to create a robust automated issue assignment tool.

One can ask if the same training can be given to a support representative and if he/she can assign the task to respective developer according to the algorithm/logic. Yes, it is possible but, here are the drawbacks of using this approach.

2.1.1 *Redundant task:* It has to be repeated for multiple support representatives.

2.1.2 *Rigid:* Cannot be extended to other work flows.

2.1.3 *Close ended approach:* Possibility of data generation is minimal compared to machine learning approach.

3 CHALLEGE 2: INFORMATION RADIATORS

Our distributed development environment posed many communication and collaboration challenges. Although we had channels setup for communication across the globe when we did Scrum [1], it created confusion on what to follow after transitioning to Kanban.

Using a common dash board was challenge in itself. When we first designed our Kanban board, there were nearly 20 states encompassing all the teams. Many teams and 20 states were difficult to fit in one digital dash board the can be shared across all the regions. Although we had tools integrated tracking requirements to release, customers wanted a crisp view providing details of the moment.

The product owner's team residing in US wanted to have clear understanding of release dates, release content and responsible teams/leads for respective releases.

Continuous Delivery was also a challenge as it required high coordination between our regional teams. Handling release for over 100 modules collectively, with many test environments and test cycles was herculean task in itself. Hence, we needed a dashboard which could tell all the information at one glance.

Similarly, from the developer's perspective, team needs to be updated with features committed to customers along with the time line.

Physical dash boards are most desired, convenient and handy information radiators for collocated teams. But we needed to share information across our team in different regions.

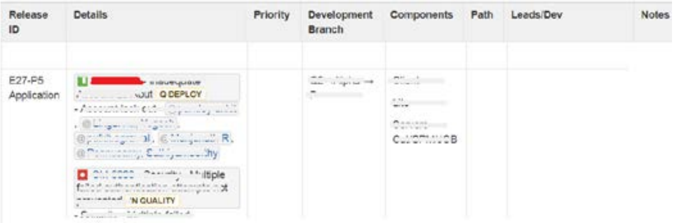
3.1 Solution

The team chose a tool which can help us in creating different views of digital dash board. More specifically, we needed an Integrated Release plan View and Cadence View.

Since of the information is inter related, we needed a views to be connected to and traversed from, each other.

3.1.1 Integrated Release Plan View

This view has information about all the planned releases. As you see in Figure [2], it shows release ID, contents of the releases, developers working on it, Team Leads who are responsible for the release, projected release date, priority of the release and all the required information about the release. This view contains one such table for each release.



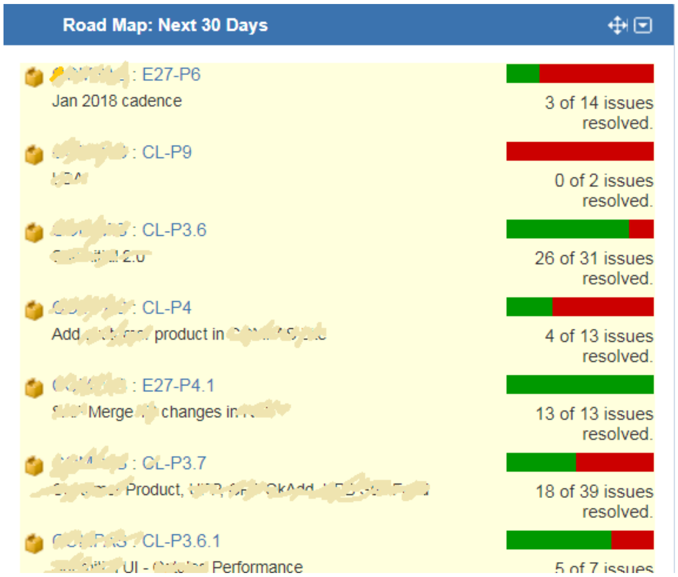
Release ID	Details	Priority	Development Branch	Components	Path	Leads/Dev	Notes
E27-P5 Application	[redacted] Q DEPLOY [redacted] N QUALITY		[redacted]	[redacted]			

Figure2. One of the tables in Release Plan View

This view also contains the details of the impediments and roads blocks which have to be addressed by the project as a whole.

This view is updated once in a week. All the stake holders meet during a fixed time, which is common working time for all time zones. During this meeting, existing release plans are updated and new ones are added as and when product owners get clarity on the releases.

There is another view created which gives pictorial information about the releases and its completion status as shown in Figure[3].



Road Map: Next 30 Days

[redacted] : E27-P6
Jan 2018 cadence
3 of 14 issues resolved.

[redacted] : CL-P9
[redacted]
0 of 2 issues resolved.

[redacted] : CL-P3.6
[redacted] 2.0
26 of 31 issues resolved.

[redacted] : CL-P4
Add [redacted] product in [redacted]
4 of 13 issues resolved.

[redacted] : E27-P4.1
[redacted] Merge [redacted] changes in [redacted]
13 of 13 issues resolved.

[redacted] : CL-P3.7
[redacted] Product, [redacted]
18 of 39 issues resolved.

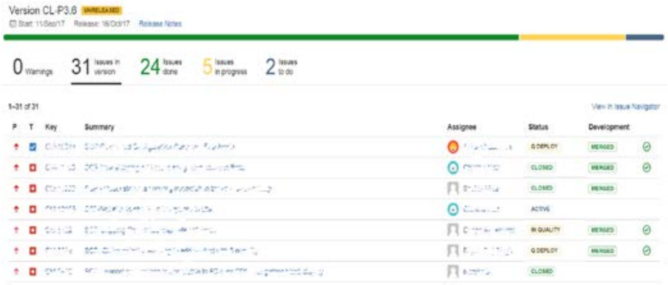
[redacted] : CL-P3.6.1
[redacted] UI - [redacted] Performance
5 of 7 issues

Figure3. Crisp Release Plan View

3.1.2 Individual Kanban/Cadence View

Cadence view, as you can see in Figure[4], gives complete details of each release. It has following details.

1. Start date and end date of the release
2. Number of tasks in each stage Done, in progress and to do.
3. Total number of tasks targeted for the release and build warnings.
4. Task Priority, Task type(Bug,Sub-Task, Epic and so on), Status of the task, merge status of the code to main branch.
5. Each task depicted here will take you to detailed page of each task where complete details of each task is shown.



Version CL-P3.6 UNRELEASED
Start: 11/Sep/17 Release: 16/Oct/17 Release Notes

0 Warnings 31 Issues in version 24 Issues done 5 Issues in progress 2 Issues to do

1-31 of 31 View in Issue Navigator

P	T	Key	Summary	Assignee	Status	Development
		[redacted]	[redacted]	[redacted]	Q DEPLOY	MERGED
		[redacted]	[redacted]	[redacted]	CLOSED	MERGED
		[redacted]	[redacted]	[redacted]	CLOSED	MERGED
		[redacted]	[redacted]	[redacted]	ACTIVE	
		[redacted]	[redacted]	[redacted]	IN QUALITY	MERGED
		[redacted]	[redacted]	[redacted]	Q DEPLOY	MERGED
		[redacted]	[redacted]	[redacted]	CLOSED	

Figure4. Individual Cadence View

We can say that we have implemented Multiban[8], but it is not completely true. We did not stick to all the practices of Kanban, especially WIP limit. We just did what is required to have a better visualization for our work status.

4 RESULTS

After months of practice these two solutions helped us in getting good customer appreciation. Following are few results of these practices.

1. Earlier Lead Time was 35 days and now it is 14 Days.
2. Release frequency increased by ~100% (24 releases in an year)
3. Concept to Production ready. (3 concepts has been accepted by customer. 1 moved to production, 2 in beta testing). Many in progress.
4. Digital dashboard helped us in collecting data like Lead time/Turnaround time from the abundance data it gets gathered, which in turn helped in improving the team performance.
5. Since all the views are interconnected, customers have broad spectrum of data from overall picture of releases to minute details of the task.
6. Abundant data available in these dash boards, smoothened collaboration between all the stakeholders, resulting in reduced meetings with product owners in other locations. Now the same time is utilized in demo and reviews.
7. Experimenting with dashboard helped all stakeholders to have their personal dashboards which will give them needed information.
8. Eventually customer rated our service the highest.

5 FURTHER POSSIBILITIES

5.1 Extension to other areas of work

The automated issue analysis and assignment can be extended to production support and feature development as well. The system can be extended to Testing and deliver states as well.

5.2 Supervised to Unsupervised

Extension to Unsupervised learning makes it more agile, flexible and extensible.

5.3 Data Analytics

Can be extended to generate data on individual and team contribution, which can be helpful for management. Inclusion of the analyzed data into dash board will help to increase value in many areas for the project. E.g. Results of this analysis can help us to find areas of issues occurring, types of issues occurring, performance of the team, and performance of the individuals. The team can choose area of focus and include the metrics accordingly.

6 LESSONS LEARNT

When physical dash boards becomes bottle neck for collaboration in a globally distributed environment, little innovation and tweaks in existing formats, digital dash boards are informative, handy and amazing collaborative tool.

We started with Kanban, as customer insisted, we kept tailoring our process, finally we let go some basics of Kanban process. It is OK to do that when customer insists and it is working for you. After all, we are doing it for customer satisfaction.

Transparency is pillar for many important agile frameworks. Dash boards giving clear picture of the projects progress bring in transparency among all the stakeholders eventually leaves customer with higher satisfaction.

Tools which combine bug tracking tool and digital dash board leave behind a dearth of data. Machine learning can help to leverage this data to add value to the system and remove waste.

With these results , what it leaves behind is ,

1. A huge scope of creating convenient, collaborative digital dash boards
2. Systems which can learn by itself to save time and reduce waste.

REFERENCES

[1] Rajeev Gupta, Prabhulinga M, “Challenges in adapting scrum in legacy global configurator project”,IEEE, Global Software Engineering (ICGSE), 2015 IEEE 10th International Conference on (ICGSE-2015), DOI: 10.1109/ICGSE.2015.19.

[2] Prabhulinga M, Rajeev Gupta, “Pragmatic approach for managing technical debt in legacy software project”,ISEC2016.

[3] Rajeev Gupta, Prabhulinga M, “Challenges in adapting Agile Testing in legacy product”, Global Software Engineering (ICGSE-2016), DOI: 10.1109/ICGSE.2016.21

[4] https://www.scrumguides.org/scrum-guide.html

[5] https://www.atlassian.com/software/jira/comparison

[6] https://www.linkedin.com/pulse/making-machine-learning-part-your-future-agile-team-ashish-gupta/

[7] https://realtimeboard.com/blog/scrum-kanban-boards-differences/#.Wpz8_mpuapo

[8] https://www.targetprocess.com/blog/2013/10/kanban-as-multiban/

[9] http://kanbanblog.com/explained/