Adapting Agile in a Globally Distributed Software Development

Rajeev Kumar Gupta, Prabhulinga Manikreddy
Siemens Technology and Services Private Limited
Bangalore, India
rajeevkumar.gupta@siemens.com, Prabhulinga@siemens.com

Abstract
This paper describes the journey of adapting Agile-Scrum in a project, referred here as Global Configurator Project (GCP), where the project members are distributed across locations in Germany, India, and U.S. Here, we will be discussing about the practices like new taxonomy on scrum roles, scaling scrum with actually using Scrum of scrums and evolving cross-functional multi-skilled and self-managed feature team from silos components teams. This paper also measures the impact of our practices on communication and collaboration at the project level. This paper will be useful for scrum masters, product owners, and project managers while adapting Scrum.

I. INTRODUCTION

The GCP project started with the traditional Waterfall approach. The product’s first version was officially released in 2008. In 2008, the product had only 10 end users and currently has almost 4,000 end users in U.S. In 2012, the GCP project team adapted the Agile-Scrum approach with an objective to have on time and stable delivery to the end users.

Background of GCP:
Project Organization:
Development happens majorly in India and some part of it in Germany. The development teams consist of seven to eight members that include architects and developers. The U.S. team consists of the product manager and the domain experts and they are the interface between the development team and end users.

Project Roles:
GCP had traditional software project development roles like Project Manager (PM), Development Lead, Developers, Architects, and Testers.

Technologies:
Initially, GCP was developed with client and server based technologies. However, since the last couple of years, GCP is targeting mobile platform with new technologies. GCP is a complex project where few million lines of code have been added in different techniques and languages since the last four years.

II. GCP AGILE PRACTICES

1. Shared Agile Roles

Before Agile:
The teams were divided based on Modules and functional flows. The teams did not have a big picture and used to defend their functional flows and modules. The Development lead was an expert in that respective area and used to lead the team, give quantitative data about the team’s productivity to the management and the Product Manager. The Development lead used to get all the required requirement clarifications from DESE’s (Domain Expert Support Engineer).

We had the Traditional waterfall roles that is, Project Manager, Development leads, architects, developers, testers, Product Manager, etc. Also, the Product Manager has developed his team of domain experts. While adapting agile,
the first challenge was to establish scrum roles as defined in the Scrum Guide [9]. Existing roles and their responsibilities are mentioned below.

**Product Manager:** Product Manager was the final authority for most of the actions/practices or requirements changes. Product Manager was also interfacing with various departments of his/her organization like Sales, Finance, etc. as well as approving the budget for the offshore team. Product Manager is also the reporting manager for domain experts and used to delegate his responsibilities of driving development to domain experts. Product Manager used to collaborate with domain experts and users for the overall status update and deciding the road map for GCP. Product managers had to micro manage offshore development team for improving team efficiency.

**Project Manager:** The Project Manager was a spokesperson for GCP and was the point of contact inside the organization for GCP. The Project Manager also owned release deliverable, updated project plan, monitored and controlled overall project progress, and acted as an interface for communicating release contents and discussing escalations with the product manager. Project Manager was also reporting manager for Development leads and architects. Project Manager used to guide the Development leads for implementing best engineering practices and ensured that they meet the entire milestone defined in GCP.

**Development lead:** The Development lead was leading one of the offshore development team and was usually co-located with the developers. As an exception, one of the Development lead was not co-located with his development team and was working remotely. His team was however very effectively delivering the assigned tasks. Development leads were technically very sound and were usually experts of various components or workflows or features. Development leads in GCP had the following assignments:

1. Direct interface to Domain experts for the status update, requirement engineering and discussing tasks for his team.
2. First point of contact to developers for any clarification on technical topics, requirements, demos or prioritization of tasks.
3. People management for developers like skill and training planning and assigning tasks to individuals.

**Domain Experts:** The Domain experts were part of the product manager’s team. As a Product Manager, domain experts work from U.S and shared their complete time mapping with users. They had three primary responsibilities:

1. Interface to users for any new requirements, any technical support or training on new features.
2. Interface to the development team at offshore location for any new requirements, clarification, reviewing and status update.
3. Assigning tasks to the offshore team

Domain experts reported to the Product Manager and was empowered to work with any one or more offshore development teams.

**Testers:** We had a separate testing team, and testers were part of the testing team. Testers were managed by the Test lead and the Test Manager. Test Manager used to develop the testing strategy and collaborated with the project manager for creating the Test plan. The Test Lead used to assign testing tasks to testers, monitor activities and update the testing status to stakeholders. Testers used to execute the test cases manually, though execution used to happen through test automation for the new feature and defect verification & validation. Testers used to wait until the software binaries and the list of changes were made available to them. Till then, the testers were mostly engaged in the verification and validation of previous software package. The testing team used to interact only when developers need steps to reproduce defects or when developers give demo to testers or before tester starts verification and validation.

Ref [9] defines three roles in any scrum team that is, dedicated and full-time Scrum Master and Product Owner, cross-functional and self-managed Development Team. These roles were not mapped one-to-one with the traditional project roles in GCP.

Due to this mismatch, the scrum team was facing difficulties in the areas of prioritization, planning, requirements, and coordination. This troubled responsibility distributions in the scrum team resulted in poor impression in the minds of the project team members.

**After Agile**

**Shared Ownership: New Scrum Taxonomy**

1) **Shared Product Owner:** From the traditional role described above, we found that the three roles who were sharing the responsibilities of a product owner were Product Manager, Domain Experts and Development lead. Also, due to the different time zones and non-availability, we shared product owner with new roles like Part Product Owner and Chief Product Owner. Refer Table 1 for Domain Experts and Product Manager Role.

Unlike a single product owner role suggested in Large-Scale Scrum[LeSS], every scrum teams has a dedicated and full-time available part product owner.

As time passed, it worked extremely well for us.
These roles are described below.

a) Part Product Owner (PPO):
Due to their domain expertise and knowledge about the complete product and its architecture, Domain Experts were the best to be the Part Product Owners. Due to different time zones, domain experts were not coexisting with the developers and were not available for team all the time, and hence we got buy-in from Chief Product owner for sharing product ownership between the Development leads and the Domain experts. However final responsibilities of Return on Investment (ROI) remains with the Domain Experts turned Part Product Owner and Chief Product Owner. And hence they played a partial role of product owners. Due to the different time zones, the Part Product Owners could dedicate only 20-25 percent of their time to the development teams. The Scrum Team used this duration very effectively in collaborating with the Part Product Owner.

b) Chief Product Owner (CPO):
The Product Manager has final authority for all product related decisions. All the essential features were discussed and prioritized by the Product Manager. The Product Manager is the person who sets goals for the team and coordinates the team of Part Product Owners. Thus, Product Manager is the Chief Product Owner. The Chief Product Owner is responsible for ensuring that the Part Product Owners are responding to the clarifications required for the requirement and communicating the same timely to the development team. This would avoid any unnecessary assumptions by the development team.

2) Shared Scrum Master:
As described above, in waterfall model, we had two roles that were sharing responsibilities similar to a scrum master, that is, Project Manager and Development lead. Refer Table1 for Role Mapping. Development leads are groomed as a Scrum Master, one for each team. However, we had three challenges with the role of a scrum master:

i. Development leads were also contributing towards product ownership
ii. One of the Development lead was working remotely with the development team. It leads to another challenge of communication and collaboration.
iii. We required coordinator among scrum teams.

These challenges lead to develop a different taxonomy for scrum master, i.e. Scrum Master cum Part Product Owner, Bi-Scrum Master and Chief Scrum Master.

Table 1: Traditional Role Mapping with Scrum

<table>
<thead>
<tr>
<th>Before Agile</th>
<th>After Agile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain Expert Support Engineer (DESE)</td>
<td>Part Product Owner (PPO)</td>
</tr>
<tr>
<td>Development Lead</td>
<td>Scrum Master-cum- Part Product Owner (SMPO)</td>
</tr>
<tr>
<td>Project Manager (PM)</td>
<td>Chief Scrum Master (CSM)</td>
</tr>
<tr>
<td>Product Manager</td>
<td>Chief Product Owner (CPO)</td>
</tr>
<tr>
<td>Developers, Architects and Testers</td>
<td>Development Team</td>
</tr>
</tbody>
</table>

These roles are described below.

a) Scrum Master-cum-Part Product Owner (SMPO):
Scrum master cum part product owner has responsibilities like guiding and coaching Scrum teams, resolving collaboration and coordination issues and clarifying requirements but not for prioritization or ranking issues. The responsibilities that include prioritization, ranking and backlog grooming remained with the Domain Experts turned Part Product Owner and Chief Product Owner.

At any given day, both the part product owners for all the scrum teams were required to be in agreement and get consent from the either side.

Scrum master cum Part Product Owner ensured that the teams were aligned on prioritization and acceptance criteria.

b) Bi-Scrum Master (BSM):
Bi-Scrum Master is co-located with the scrum team and mostly responsible for facilitating Scrum ceremonies. Bi-Scrum Master collaborates with the Scrum Master cum Part Product Owner for addressing the impediments. Bi-Scrum Master also shares the dashboard update with the Scrum Master cum Part Product Owner.

c) Chief Scrum Master (CSM):
The Chief Scrum Master ensures that the project vision and the ‘Big Picture’ are communicated to the scrum teams. The Chief scrum master usually coordinates with the Scrum team through the scrum of scrums and is responsible for impediments in the project. The Chief Scrum Master also represents the project in the organization.

Figure 1: Two types of new scrum teams in GCP
2. Developing Features Team

Before Agile:
GCP had traditional team structure. Team members have been practicing silos style of working for years. This is the reason why we had many experts in the project. Team members were experts in workflows and components and had a very long experience of working in their workflows or components. This situation advocated forming teams in silos. For example, all the developers and architects working on client component were grouped into two teams all the developers and architects working on server component were grouped together to form another team, and all the testers were grouped together as a separate team. Name of these teams also had a similar influence like client team, server team, and testing team. Again, every team had several component experts. Component experts were distributed in different countries.

GCP had several issues due to the components experts. Few of these issues are mentioned below:

i. Team members are not available for development even though all the team members have priority assignment.
ii. Development team works on multiple less priority tasks to get engaged with work.
iii. Delay in developing high priority requirements together.
iv. Loss of knowledge and expertise in case of attrition.

GCP has a similar situation for the testing team. The testing team gets development binaries for testing in bulk. They have to complete testing in less duration than the estimated time because the development phase has exceeded the estimated development time. Testers were not much involved in all the phases of the product development cycle starting from the requirement analysis phase till the code development phase. Testers were mostly busy in verifying and validating the binaries from the previous release. The testing team hence had to confirm the changes in hurry and multiple defects escaped unidentified to the production.

Dependent Team
The Development lead used to support the team members in taking a decision on their day to day activities. The team members thus developed a dependency on the Development lead. The team members were not able to make their day to day decision on their own.

After Agile:
As advised in LeSS framework, we targeted to create a feature team that is cross-functional, multi-skilled and self-managed.

1) Cross Functional Team:
As the first action, we restructured all the teams. New teams had combination component experts from client and server team. Each team has at least one expert tester. We renamed teams like Team 1, Team 2, and Team 3, etc.

This was the first baby step towards developing a cross-functional and multi-skilled team. We practiced this approach for several sprints along with some best practices from other agile methodologies. These are explained below:

a) Rewards and Recognition
At a project level, we awarded the teams who worked on maximum number of new components and workflows, created secondary or backup resources and actually used their slack time by indulging in activities like increasing the percentage of test cases automated, conducting knowledge sharing sessions, etc.

b) Knowledge Sharing
Every sprint, every team was given a target to share at least one of their team expertise with other team members. This practice makes a huge impact on distributing knowledge and breaking silos.

c) Pair Programming
It was not easy to get buy-in from every team member for pair programming. However, we started with few developers for pair programming and shown improvement in product quality knowledge shared.

d) Test Automation
The Scrum Master cum Part Product Owner involved testing team members in all the phases of development right from the requirement analysis phase till the product development phase. The testing team collaborates with the Part Product Owner and the Scrum Master cum Part Product Owner to develop acceptance case and automate them.

e) Slack Time
The Scrum Master cum Part Product Owner advised the team to keep some time for every sprint as the slack time and use this time in knowledge sharing, learning new components, workflows, supporting tester for automation etc. We also used the slack time for continuous improvement on domain knowledge, learning new skills, automating legacy test cases, improving code quality, addressing technical debts, etc.

It was difficult to get buy-in from the Chief Product Owner for few of these practices like pair programming, slack time and involving developer in developing test automation. We started with a very small volume, shown improvement to the Chief Product Owner, shared the best practices from the other agile projects and eventually with an improvement in product quality and faster delivery to the production, we got the consent from the Chief Product Owner.

2) Self-Managed Team:
The Scrum Master cum Part Product Owner sets the expectation to the team members for taking their decision with available for support whenever required. She/he also
encouraged the team to support each other. Few practices that helped in developing a self-managed team:

i. The Scrum Master cum Part Product Owner encourages brainstorming and involved the team members for decision-making.

ii. Respected advice and ideas from the team members and get buy-in from team.

iii. Scrum Master cum Part Product Owner ensured that the team is not punished for any incorrect decision and trained the team for taking a better decision in the specific situations.

These practices helped to develop cross-functional and self-managed feature team who was capable of developing any requirement and take decision for their day to day activities by themselves.

Weekly status meeting was a collaborative event among the development leads, Product Manager, and DESEs but did not involve the development team. This lead to communication gaps between the development team and DESE, which resulted in rework, schedule slippage, poor code quality due to last minute changes. It also led to an increase in customer complaints.

Refer [9] says that the daily stand up must be held with the development team. But the project needed transparency at a higher level too. Though distributed nature of the project was a hurdle, we came up with new events so that we could have everyone on the same page.

Defining new roles and responsibilities was not sufficient to improve team collaboration. We required better collaboration strategy like along with daily standup, we needed to improve collaboration among stakeholders.

The defects and new development features used to assign to development team by DESEs and Development lead, mostly based on team and lead expertise.

We did not have any process defined for demo and getting customer feedback. Also, the team was not sure on the point of contact for requirement clarification, though development leads used to clarify with their area of expertise.

**After Agile:**

**a) Daily Stand-up meeting:**

The Scrum Master cum Part Product Owner schedules a daily standup meeting. The Development team participates in this meeting in the presence of the Scrum Master cum Part Product Owner.

Since the Part Product Owner is not co-located with the development team, a twenty minutes meeting is scheduled between the Scrum Master cum Part Product Owner and the Part Product Owner. The Scrum Master cum Part Product Owner uses a digital image of the scrum team dashboard to demonstrate the progress. The Scrum Master cum Part Product Owner shares all the status received from the team, which answers the three questions mentioned in scrum guide [9], for each of the task. The Part Product Owner shares additional information with Scrum Master cum Part Product Owner, for example, if there is any critical topic/defect from end users or change in priority.

This extension to daily Scrum ensures that the team is in the right direction as per the customer’s expectation. It also builds collaboration between the Part Product Owner and Scrum Master cum Part Product Owner to bring completeness to the Product Owners role. Any concern from the team is communicated to the Part Product Owner and is checked on a daily basis. The Scrum Master cum Part Product Owner updates the critical topics that arise from the end user to the team in the next day’s daily stand up.
b) Weekly Scrum of Scrums:
A thirty-minute weekly Scrum of Scrums is scheduled by the Chief Scrum Master. The Chief Scrum Master and the Scrum Master cum Part Product Owners are the participants. The Scrum Master cum Part Product Owners answers the following three questions:

1. Which are the sprint backlog tasks that the scrum team completed since the last Scrum of Scrums?
2. Which are the sprint backlog tasks that the scrum team is currently working on?
3. Are there any impediments that need support from other Scrum teams?

This meeting not only brings transparency to the Chief Scrum Master and the management, but it also allows the Chief Scrum Master to address the communication and collaboration issues among the teams.

c) Bi-Weekly Scrum of Scrums:
The Chief Product Owner schedules a sixty-minute bi-weekly Scrum of Scrums. All the key stakeholders including all Scrum Master cum Part Product Owner, Part Product Owner and Chief Scrum Master, Release manager, etc are the participants. Scrum Master cum Part Product Owners and Part Product Owners share the progress in development and discuss product risk, new changes requests from the customer, etc.

This extension to the prescribed ceremonies helped in ensuring quick decisions on critical new topics. This improved confidence in the scrum teams that eventually contributed in empowering the scrum team and thus build a high performance team.

d) Backlog Grooming and Demos:
Scrum Master cum Part Product Owner sets up live meeting for duration of 45 minutes every day. Part Product Owners, Scrum Master cum Part Product Owners and the development team were the participants. The Scrum team used this time for demonstrating working software, requirements clarification, and discussion. The Scrum Master cum Part Product Owner used it effectively for backlog grooming. Only the required team members join this meeting. The Part Product Owner invites other participants like Chief Product Owner or key stakeholders based on the need. This is a dedicated window with overlapping time for both developers from India and other parts of the world.

This collaborated event between the Scrum Team and the Part Product Owner helped in addressing only the core issues related to the features that the user is looking for. This also increased the accuracy in converting the user requirements into features. This time-boxed discussion elevated communication and collaboration between the Part Product

iv. MEASURING IMPROVEMENTS

Survey on Communication and collaboration:
We are measuring the improvements in communication, collaboration and building a high performance team based on the response of the scrum team on a survey, which comprised of two questionnaires. We asked a set of questions to the team members and asked them to score on a scale of 1-5. 1 stands for “Strongly disagree” and 5 stands for “Strongly agree”. Each team member scored twice for each question, one for the situation in 2012 when the transition happened to an agile process and the other for the current situation (2015).

Around 75% of the all the scrum teams participated in the survey. We selected only those participants who have been in GCP on or before 2012. Participants are among the development team, all Scrum Master cum Part Product Owner, Part Product Owner, Bi-Serum Master, and Chief Scrum Master. The average experience of the participants (in GCP) is almost five years, and the total average working experience is almost nine years. We shared questionnaire and collected response to both the questionnaires anonymously.

**Questionnaire 1:** Eight questions from Jean Tabaka [2]

1. **Self Organization:** Is your team self organizing, rather than functioning in command and control (top-down organization)?
2. **Empowered to make decision:** Is the team empowered to discuss, evaluate and make decisions, rather than being dictated to by Scrum Master cum Part Product Owner, Chief Product Owner or anyone else?
3. **Belief in Vision and Success:** Do the team member understand the project vision and goals and do they truly believe that they can solve any problem to achieve any goals as a team?
4. **Committed Team:** Are team members committed to succeed as a team, rather than being committed to individual success at any cost?
5. **Trust each other**: Does the team have the confidence to continually work on improving their ability to act without fear, anger or bullying?

6. **Participatory decision making**: Is the team engaged in participatory decision making, rather than bending to authoritarian decision, making or succumbing to decision from others?

7. **Consensus Driven**: Are team decisions consensus driven, rather than leader driven? Do team members share their opinions freely and participate in the final decision?

8. **Constructive Disagreement**: Is the team able to negotiate through a variety of alternatives and impacts surrounding a decision, and craft the one that provides the best outcomes?

**Questionnaire 2**: Nine questions from James Shore [1]

1. **Working with Guess**: Do programmers ever make guesses rather than getting answers to questions?

2. **Information on Time**: Are programmers usually able to start getting information (as opposed to sending a request and waiting for a response) as soon as they discover their need for it?

3. **Clear Communication**: Do team members generally communicate without confusion?

4. **Trust Each Other**: Do nearly all team members trust each other?

5. **I-Know-All-Peer**: Do team members generally know what other team members are working on?

6. **Frequent Demo**: Does the team demonstrate its progress to stakeholders at least once per month?

7. **Customer Satisfaction**: Are all important stakeholders currently happy with the team's progress?

8. **Frequent Release**: Does the team provide a working installation of its software to the stakeholders to try at least once per month?

9. **Technical Ability**: Do all important stakeholders currently trust the team's ability to deliver?

**Team Response**: The average team score for each of the question is added in Table 2 & 3 and plotted in the bar chart in Figure 4 & 5.

### Table 2: Team Response on Questionnaire 1

<table>
<thead>
<tr>
<th>Questionnaire</th>
<th>FY'12</th>
<th>FY'15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Score</td>
<td>Score</td>
<td></td>
</tr>
<tr>
<td>Self Organization</td>
<td>2.8</td>
<td>3.4</td>
</tr>
<tr>
<td>Empowered to make decision</td>
<td>2.6</td>
<td>3.6</td>
</tr>
<tr>
<td>Belief in Vision and Success</td>
<td>3.2</td>
<td>3.9</td>
</tr>
<tr>
<td>Committed Team</td>
<td>3.6</td>
<td>3.9</td>
</tr>
<tr>
<td>Trust each other</td>
<td>3.1</td>
<td>4.1</td>
</tr>
<tr>
<td>Participatory decision making</td>
<td>2.4</td>
<td>4.0</td>
</tr>
<tr>
<td>Consensus Driven</td>
<td>2.8</td>
<td>3.4</td>
</tr>
<tr>
<td>Constructive Disagreement</td>
<td>2.8</td>
<td>3.2</td>
</tr>
</tbody>
</table>

### Table 3: Team Response on Questionnaire 2

<table>
<thead>
<tr>
<th>Q. No</th>
<th>YR'12</th>
<th>YR'15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Score</td>
<td>Score</td>
<td></td>
</tr>
<tr>
<td>Working with Guess</td>
<td>4.1</td>
<td>3.8</td>
</tr>
<tr>
<td>Information on Time</td>
<td>3.1</td>
<td>3.4</td>
</tr>
<tr>
<td>Clear Communication</td>
<td>3.0</td>
<td>3.9</td>
</tr>
<tr>
<td>Trust Each Other</td>
<td>4.1</td>
<td>4.4</td>
</tr>
<tr>
<td>I-Know-All-Peer</td>
<td>1.7</td>
<td>4.1</td>
</tr>
<tr>
<td>Frequent Demo</td>
<td>1.9</td>
<td>4.7</td>
</tr>
<tr>
<td>Customer Satisfaction</td>
<td>3.7</td>
<td>3.8</td>
</tr>
<tr>
<td>Frequent Release</td>
<td>1.6</td>
<td>3.9</td>
</tr>
<tr>
<td>Technical Ability</td>
<td>2.6</td>
<td>3.7</td>
</tr>
</tbody>
</table>

**Analysis of Team Responses from Questionnaire 1**: Better score on questions like ‘Self-Organization’, ‘Empowered to make decision’, ‘Participatory decision making’, ‘Consensus Driven’ and ‘Constructive Disagreement’ shows that the teams are motivated, self-organized and independent in taking decisions. Overall, these scores also shows that we improved in building a high performance team.
Better score on ‘Belief in Vision and Success’, ‘Committed Team’ and ‘Trust each other’ shows that we have improved in collaboration and communication.

**Analysis of Team Responses from Questionnaire 2:**
Better score on ‘Working with Guess’, ‘Frequent Demo’, ‘Clear Communication’ and ‘Frequent release’ shows that we have improved in collaboration and communication. It also displays our confidence during releases. It also highlights the effective rapport that has been built with the Chief Product Owner team who are an interface between the team and the customer.

Better Score on ‘I-know-All-Peer’ and ‘Trust Each Other’ shows that we have improved in building a high performance team. Scores on “Technical Ability” show that encouragement in Peer Programming, knowledge sharing sessions and formation of feature teams has improved the technical ability and confidence of the members.

**v. SUMMARY**

In this paper, we presented our key adaptions that has been done while embracing Agile Scrum in legacy GCP. We observed that these challenges were prominent in achieving project success. We demonstrated via this paper that adapting Scrum roles and responsibilities worked for us. The role of a scrum master and a product owner was shared among more than one person. We used scrum of scrums and daily collaboration meetings very effectively. This developed an excellent collaborative atmosphere for communication. Sharing a daily digital image of scrum board improved building trust and transparency and helped to empower the scrum and team.

These practices helped us in collaboration, communication and in building a high performance team, which is quite visible from self-assessment and the issue metrics. Questionnaire from Jean Tabaka and James Shores is very explanatory and we used them without any modification. The issue metrics also shows how it has transformed on its own and enhanced the user experience. It is also a privilege to share that there has been considerable appreciation in customer rating.

Finally, Scrum adaption and agility is a journey. We are exploring on adapting new tools, different representation of information radiators, electronics dashboard, etc. These tools will help us further improve collaboration and communication.

**References:**

[5] https://www.scrumalliance.org/community/articles/2014/july/7-mistakes-during-the-daily-stand-up-meeting
[12] Agile Estimating and Planning by Mike Cohn

5367