Collaboration and Communication in Agile Global Software Development

Karishma Bhatnagar  
Singapore Management University  
karishmab.2010@sis.smu.edu.sg

ABSTRACT
Companies have been moving toward creating products through globally distributed software development teams. They do so in the pursuit of saving product development costs and shortening product development lifecycles. Recently, they have increasingly started looking into adopting more agile methods of software development to be used via these globally distributed teams. Moving from centrally located teams to globally distributed teams that face time and cultural differences, and geographical distances has changed the method these teams prefer to use to communicate with other teams. Hence, project management procedures and companies’ collaboration methods are evolving to ensure that the communication between and within the teams is strong, efficient, and effective. This paper aims to review the formal and informal methods of communication used in collaboration and then evaluate the pros and cons of these methods, for projects applying agile practices to global software development (GSD).

Keywords
Global software development, agile software development, collaboration, communication tools, globally distributed teams

INTRODUCTION
Technological advancements, improved network infrastructure, and high time-to-market and time-to-profit pressures have led to a movement towards GSD. GSD is highly popular with companies because they can “offshore” the development of their software products to companies in developing economies that have much lower prices compared with those in developed economies. Compared with the traditional approach to software development, the globally distributed teams involved face geographic, temporal, and sociocultural differences that increase the challenges involved in collaboration (Holmstrom, Ó Conchuir, & J Ågerfalk, 2006). Furthermore, to assist and provide a more flexible approach to the offshore teams in managing the project requirements and client expectations, the companies have started using agile methodologies for software development. Agile software development is a collaborative, incremental, and iterative approach. Its focal point is software creation in a highly cost effective and timely manner. Agile development values interaction between the teams, customer collaboration, and fast responses to change in any project requirements (Serena Software Inc., 2007). Given the challenges of communicating across the globe with different teams and the need to do so continually throughout the agile software development cycle, it is obviously crucial for the teams to choose an appropriate type of collaboration. These globally distributed teams have to adopt different modes and decide between the two main kinds of collaboration, either formal or informal. Each form of collaboration requires different communication tools, and the effectiveness of these tools can provide insight into the form of collaboration best suited to the nature of a GSD project involving agile methodology.

GLOBAL SOFTWARE DEVELOPMENT
GSD is economically beneficial for companies because development centers are located in areas with cheaper labor, and these lower cost locales overcome the costs associated with carrying out the development globally (Fryer & Gothe, 2008). It provides companies with significant access to a larger pool of readily available talent at low cost, compared with the labor in developed countries. This reduction in cost serves as the primary reason many companies choose GSD instead of traditional localized software development. An additional reason for companies preferring to do GSD would be the setting that enables them to use the “follow the sun” workflow during software development. This “follow the sun” workflow essentially lessens the overall project duration. It comes into effect as work is handed off at the end of every day from one development center to the next located multiple time zones away (Carmel, Dubinsky, & Espinosa, 2010). Thus, work on the project continues around the clock, resulting in the product’s faster time to market. This is turn leads to a profitable workflow because earlier entry into the market with a good product can translate into higher profits.
Challenges Faced by Globally Distributed Teams

Although GSD has high monetary benefits, it is considerably demanding for a project team to operate as one when the subteams are globally distributed. The project teams typically have to overcome three main types of distances: geographical, temporal and sociocultural. All these distances have the potential to create misunderstandings and delay the project, which can lower the monetary gains derived through GSD. Hence, it is crucial to note what can be done to reduce the three types of distances, which is the topic explored in the following sections (Holmstrom, et al. 2006).

Geographical distance

The geographical distance between teams is measured by the time it would take one team member to reach the other team’s location physically. This distance emphasizes the need to use relevant media to communicate with and address any difficulties in finding a sufficiently good substitute for face-to-face interaction because it would be costly to relocate a team member whenever his or her expertise is required by various sub-teams.

Temporal distance

Temporal distances between globally distributed teams exist because there are time differences between the team locations. The larger the distance between two team locations, the smaller the time frame in which the teams can collaborate during standard working hours in each location. Hence, this creates requirements for each team to accommodate or work around this distance, which can be difficult if the mode of collaboration is unsuitable.

Sociocultural distance

With globally distributed teams, there will be a degree of misunderstanding between teams regarding the other team’s values and norms. That refers to the sociocultural distance between the teams. This can cause a lot of tension between the project teams when the globally distributed teams are not comfortable interacting with one another.

AGILE SOFTWARE DEVELOPMENT

Companies have simultaneously adopted agile methods for software development because these methods allow efficient modification of the product under development. The alternative of adhering to a traditional fixed-schedule lacks the responsiveness to adapt to changing customer requirements and technological advances or challenges. Furthermore, applying agile software development ensures that client-developer misunderstandings are minimal and that the high quality product developed satisfies all client expectations and requirements (Ekas, 2012).

![Agile Development Life Cycle](source.ibm.com)

For agile software development to be successfully implemented, the collaboration between developers themselves and between the developers and clients need to be frequent and extensive. Collaboration is key in the entire lifecycle of agile software development, as quoted in the Agile Manifesto, “*Business people and developers must work together daily throughout the project.*” and “*The most efficient and effective method of conveying information to and within a*
Collaboration and Communication in Agile Global Software Development

A development team is face-to-face conversation” (Beedle, et al. 2001). There is definitely a need to be constantly communicating with the various and relevant stakeholders involved in software development. This challenge can be illustrated through Figure 1 below (IBM, 2009), in which all the light blue boxes are stages within the agile development that requires the stakeholders to collaborate and discuss the future course of action. It can therefore be noted that without a good channel of communication through which frequent information can flow from client to developers and vice-versa, collaboration would not be possible.

Agile Global Software Development

When agile software development is applied to a project that is being developed globally, it brings together the benefits and challenges of GSD and agile development. A benefit of having agile global software development is that there is rapid development with minimal documentation throughout the project instead of having slow development with extensive documentation. The shorter time to production for the product provides the team opportunities to introduce the product to the market or use it earlier, which translates into greater business value. Major challenges for distributed teams embracing agile development include adapting processes meant for a single team working in one location to multiple teams across different time zones and geographies. Thus, overall using agile GSD is much harder compared with the agility of a team working from one location or just having a product being developed via traditional methodology through GSD.

So to overcome these challenges posed by agile global software development, effective collaboration from all the teams present in the process of GSD is required. As mentioned earlier, the distances have major impacts on the teams and hence on the project management. Three main aspects of the project management prone to heavy impact are: coordination between the teams, control over the teams, and communication between the project teams. Table 1 below summarizes the impacts on the teams and some of the actions necessary to reduce the distances between the teams.

<table>
<thead>
<tr>
<th>Temporal Distance</th>
<th>Geographical Distance</th>
<th>Sociocultural Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication</td>
<td>Use of technology-mediated communication required</td>
<td>Asynchronous communication between the teams</td>
</tr>
<tr>
<td>Coordination</td>
<td>Lack of awareness of other team increases effort to coordinate</td>
<td>Communication asynchronously is required, or shift working hours</td>
</tr>
<tr>
<td>Control</td>
<td>Cannot be managed by walking around</td>
<td>Asynchronous control over remote resources</td>
</tr>
</tbody>
</table>

Table 1: Issues Faced by Globally Distributed Teams

The table emphasizes that the communication between the teams is crucial to maintain communication, coordination, and control of the project. Companies then have to make a decision on the form of collaboration they would prefer for their projects, because that would determine the channels and tools used for communication with the stakeholders. The companies have to choose either an informal or formal mode of collaboration, which will be covered in the next segment of the report.

![Collaboration Tools For Global Software Development](Source: Serena Software, 2007)
COMMUNICATION TOOLS IN INFORMAL AND FORMAL COLLABORATION

Communication tools can be easily sorted into the two categories of formal and informal collaboration as shown in Figure 2 below (Serena Software Inc., 2007).

They are sorted as formal or informal based on the structure of the stored information and the level of access to the information itself to be used in the development of the software. Communication tools for formal collaboration have a fixed data structure, and the information they contain can only be used for specific purposes. Information within formal tools may also have restrictions as to which employees or users can access it. On the other hand, communication tools for informal collaboration store and provide greater access to loosely structured information (Alan, 2009).

Every software development project whether local or global uses a combination of formal and informal collaboration, but the success of a project can certainly depend on the extent to which formal or informal collaborative communication tools are used. Each collaboration method has its own set of pros and cons that affect the globally distributed teams and are summarized in the next section.

Formal Collaborative Communication Tools

*Pros*

Formal collaborative communication tools provide very accurate information regarding a project, if they are used correctly. The information provided has the ability to easily keep track of project progress, key performance indicators, and task assignments as most communication between the project team is tracked and logged. The fixed and well-defined data structure ensures that any data within will be perceived with consistency throughout the distributed team. Thus, the probability of miscommunication and misunderstanding will be lower throughout the globally distributed project team.

*Cons*

Formal tools are usually used during standard working hours because most communication is done asynchronously. Asynchronous communication means that the globally distributed teams do not communicate concurrently, so the information that is being accessed at any time would not be real-time. This also increases the risk of teams not following up at the scheduled time, or not following up at all, which might leave the teams having to endure long waits for responses from the other teams. Another downside of only using formal collaborative communication tools is that most of the communication is impersonal and lacks any display of emotion. The fixed data structure allows for little awareness about the other teams’ issues because only brief descriptions are provided in the tool. This can lead to miscommunication within the teams and thus negatively affect working relationships within the global team.

Informal Collaborative Communication Tools

*Pros*

Informal collaborative communication tools are certainly beneficial. Because they allow for synchronous communication, the information sent is in real time, and the response from the other team can be seen instantaneously if the team uses the tool effectively. This speeds up the process through which the project is developed and how issues are tackled immediately as they are detected. These tools also allow for a personal touch in the communication between the globally distributed teams, which is advantageous as thoughts and ideas comprise the main content of the information sent through. It allows the teams a better understanding and perspective of each other’s issues and capabilities.

*Cons*

A downside of informal collaborative tools is that there is too much unstructured information between the teams. Because there is a loose data structure, it can allow for various perceived notions of the same information, causing confusion among the globally distributed teams. This might exacerbate the misinterpretation of information caused by language barriers as well, which might then lead to misunderstandings between the teams.

FORMAL VS. INFORMAL COLLABORATION FOR AGILE GLOBAL SOFTWARE DEVELOPMENT

As seen from the pros and cons of each of the collaborative communication tools and as mentioned earlier, all projects do need to use both tools for formal and informal collaboration. However, when seen from the perspective of a project with agile GSD, the role of informal collaborative tools becomes increasingly vital. This arises because, without visual cues and the lack of awareness of the other teams’ issues such as their challenges, the capabilities of a development team are not obviously recognizable. The use of informal collaboration lowers the lack of awareness because it allows for communication on a more personal note. This in turn reduces the miscommunication and misinterpretations within the global team when compared with...
having purely formal collaboration. Having said that, the formal collaboration tools are of course used to ensure that there is efficient functioning of a software development team.

Also, because of the nature of agile practice and the high frequency with which information needs to flow within the globally distributed teams, it is definitely more efficient to have informal collaboration for day-to-day activities so that all the teams are on the same page and have access to the information they need so that decisions can be made without wasting time. This was clearly evident in my experience in a course – Global Software Project Management — in which we had to work together with students from the IT University of Copenhagen (ITU) to develop a Web application. Each team was assigned different parts of the Web application, with the team from ITU in charge of the backend development of the application and our team in Singapore was responsible for setting up the frontend. This experience was intriguing and thought-provoking because it allowed us to understand that informal communication through Skype and WhatsApp was important for us to understand the other team’s point of view because the other mode of communication gave us data and information but was insufficient in properly explaining the overall ITU team’s plan of action. It was only through the informal collaboration that we could really understand the entire project plan the other team had and provide timely updates to each other. However, the formal collaboration, which included a work breakdown structure and project scope document, was equally important for us to keep track of our progress and know how each team was contributing to the development of the product. It was overall a very educational experience because it provided insight into how each type of collaboration is necessary in agile GSD.

CONCLUSION

It is apparent that for a project with GSD applying agile practices, the global team should use formal collaboration to keep track of the project management related issues such as project schedules and metrics. This is also because it provides clear documentation and reduces confusion among the team. Informal collaboration is more efficient for discussions with the various stakeholders during the different stages of the agile development because it reduces the net time in which the teams analyze and reply to the information sent to them. Hence, in agile GSD it is important to have frameworks and tools that address the need for quick, impromptu discussions and also to have other formal forms of communication through which sign-offs for services rendered by the company permit decisions to be formalized.

REFERENCES


