Lameck Onyango Oyare
MSc. Software Engineering

Reg.SCT313 – 3506/2017

ICS 3104: SYSTEMS ENGINEERING

Continuous Assessment I

Most agile process models recommend face-to-face communication. Yet today, members of a software team and their customers may be geographically separated from one another. Do you think this implies that geographical separation is something to avoid? Can you think of ways to overcome this problem?

Do you think this implies that geographical separation is something to avoid?

Geographical separation cannot be avoided, due to the fact that many businesses are expanding to reach a wide customer base in order to significantly increase profit. Therefore, investment is being made to convert national market to global markets.

This scenario involves more competition and collaboration. Various challenges like more faults in the projects and scarcity of resources have to be dealt with.
Thus many organizations began setting up geographical separated Software Development facilities to solve these problems. This will help in the reduction of costs involved and access to skilled resources. Their main objective is to develop high quality products at lower cost than co-located developments by optimizing the resources. However, communication in the remote sites remains a challenge, as modern software development requires agile methodology, which produces high quality software, but recommends face-to-face communication.

Can you think of ways to overcome this problem?

Agile software development refers to a group of software development methodologies based on iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams. An agile method works very well in highly dynamic business and IT environment.

Both agile and distributed software developments are growing trends, as software business requires quicker quality production at a cheaper price. Geographical separation of development is a fact of life for many agile teams. Most of the agile methodologies (e.g. scrum) assume that the team is located in a single room. Unfortunately this principle does not fit in the real scenario where agile teams are also geographically separated across the globe. Factors like expanding business to new markets (Global markets through mergers and acquisitions), creating high quality employee pool, reduced costs through outsourcing to regions with cheaper development overheads are the main driving forces for organizations opting for distributed development.
We have various tools that can facilitate both formal and informal communication and project support on remote sites. Some of these tools will maximize communication

and expectation that it will take them some time to optimize around them. The suggested tools can be categorized by their main function:

- Social networking tools: different social software tools and social networking tools enable group interactions in different geographical locations, which also include everything from email to video conferencing. Examples are Facebook, twitter, Google etc. These tools can be categorized by their main function:
- Communication tools: e-mails, instant messengers Software configuration management tools: repositories and version controlling tools. A good example is GitHub, which offers an incredibly effective way to collaborate on development projects.
- Bug and issue tracking databases: that contains the information about bugs found
- Knowledge centers: containing technical references and frequently asked questions.
- Collaborative development environments: providing project workspaces and standardized worksets e.g. project repositories and project management tools are recommended as necessary solutions in distributed agile projects.

Telephone headsets are also needed. Collaboration will require many phone calls, and fatigue will quickly hinder such collaboration without a headset. I recommend having an extra headset on hand; you don't want to be working even for a few hours without one.
Online screen sharing software is also necessary, as it allows two or more people to work with the same application simultaneously. Presentations, whiteboard models, and pair programming are all easier with screen sharing software. I also recommend the use of digital tablets; they make diagramming easier.

Wiki software can be used to track information relevant to the team, ranging from requirements and design documents to coding standards and vacation schedules. They are easy to install and maintain. For a development team, the wiki can be deployed on an internal network and made available to remote users via a VPN, mitigating security risks. A remote team can successfully employ Extreme Programming even though XP encourages a co-located team. The co-located environment can be simulated. Morning

stand-up meetings can be accomplished with a conference call. Story cards can be maintained in a simple spreadsheet or a wiki. Pair programming can be done with telephone headsets and screen sharing software

Distribution of work: You will need to come up with a work plan where a certain component is allocated to developers in one location. This will be such that different locations will become over specialized in particular components. Distributed teams should continue to think about their work in the context of completing user stories not adding features to components. They need to consciously distribute tasks relating to a single story across the whole team, regardless of geography, and think in terms of user stories not system components.

Documentation: maintaining valuable documentation may also improve geographical located team collaboration process. For example providing user stories with use case diagrams in globally accessible backlogs helps to reduce misunderstandings and improve team collaboration. Various tools like issue tracker (e.g. Jira), project management tool (e.g. Scrum works) also helps in maintaining documentation and good transparency

Virtual private networks (VPN) are quite common in corporate networks and provide support for the remote user. This will enable remote to access common resources such as file systems, source control, and configuration management systems. Make sure this access is comfortable and reliable for all remote workers.

References

1. J.D Herbsleb and D. Moitra (2001), ”Global Software Development, IEEE Software”
http://conway.isri.cmu.edu/~jdh/collaboratory/research_papers/IEEE_SW_editorial_final.pdf

2. B. Sengupta, S. Chandra & V. Sinha, ” A Research Agenda for Distributed Software Development”, Proceedings of 28th International Conference on Software Engineering, Shanghai, China, 2006.

3. “Agile software development”, http://en.wikipidea.org/wiki/agilesoftware

4. Ian Somerville “Software Engineering, Ninth Edition”