

Project Performance Improvement - 15356
Individual Assignment

Charlotte Tanner – Student #11750179

Date of Submission – 15/06/2014

Abstract

The emergence of Open Source Software and the broader Communities of Practice are leading the way in terms of creating new *dynamics* and paths to *knowledge creation*. Whilst these communities of practice increasingly include global virtual teams, the teams themselves have not yet reflected the highly connected, non-hierarchical structures of Communities of Practice in their own methods and approaches to managing projects. Through the process of *reflective practice*, global virtual teams have within their power the ability to seek out new ways of thinking and *collaborating* not just within their own virtual teams, but also within the broader Open Source Software Community. The project environment of global virtual teams can create *barriers* to communication, but the shared *context* of the team itself, can act as a catalyst for transforming knowledge from *tacit* to *explicit* by increasing the points of connection within the team. By the sharing of knowledge, teams are able to build trust within the team, which leads to improved performance of the team. Whilst the emergence of new methodologies such as Agile within Project Management seeks to deal with the rapidly evolving development of new software, whereby the end ‘product’ is not always clearly defined when a project commences, does provide a workable approach to Project Management, problems can still arise within the way the team interrelates with one another. The next stage of the process should be a reflection on the way in which teams interact and how new knowledge is created.

Keywords and Phrases

Project environment, barriers and enhancers, reflective practice, dynamic knowledge creation,

Introduction

Through light speed advances in technology in the 21st Century, the phenomenon of the global virtual team reflects not only the increasing interconnectivity of people and communities through technology, it also reflects the economic reality of globalization whereby skills and experience can be sourced from a global marketplace. Despite the increasing use of global virtual teams within Project Management, the 20th Century hierarchical structures still persist within those teams (Lipnack & Stamps, 2000)ⁱ. If these teams are to survive in the global marketplace, new ways of thinking need to be developed about how teams are structured and how they relate to the wider communities of practice.

The topic of discussion within this paper is the transformation of the structures and approaches of global virtual teams through the lens of reflective practice. Through the use of a real world case study, this experiential learning process will provide a pathway to improving the performance of global virtual teams.

In covering the context of projects and the social capital that can be built within global virtual teams, this paper seeks to reflect the changing nature of teams and the shared context that can be used as the basis for dynamic knowledge creation (Nonaka, Toyama & Konno, 2000)ⁱⁱ.

This paper will firstly describe the chosen project case study of a global virtual team, working on the development of an enrolment management system utilizing an open source software platform and it will also describe the structure of the team and the approach taken at the time of the project. It will then describe what lessons were learnt through the process of reflective practice and it will outline the benefits that can be gained to the project outcome if the new approaches that are proposed are implemented. In conclusion the paper will suggest that

whilst the project team within the case study does utilize the highly adaptable Agile approach to Project Management, more work is needed on examining the way in which the team interacts and how new knowledge is created, as a way to deal with issues of team structure, communication, an inability to meet timelines and a lack of trust amongst team members.

Within software development, the complex layers of graphical user interfaces and programme coding that go to create software can often be categorised as either ‘front end development’ or ‘back-end development’. Within the context of this paper, these two terms are defined as follows; ‘front-end development’ refers to the client side of development including the graphical user interface (GUI). ‘Back-end development’ refers to the server side or administrative side of software coding (Wikipedia, 2014)ⁱⁱⁱ. This is also used as a method of categorizing the skill sets or teams that work on software development projects.

The Project

The ‘product’ the project team was working on was an enrolment management system and web interface developed for registered training organisations (RTO’s). This software combined an enrolment management system, customer relationship management database and web interface in the one package. The software was built on an open-source platform.

The company marketed the software to the education sector as a stand-alone product, which could either be installed by the client via a download link, or as a ‘cloud’ hosted service.

The project team consisted of Sydney based Project Leader, a back-end development team of four developers based in Belarus, one other back-end developer based in the Torres Strait, and a front-end developer based in India. Project management and quality assurance was based in Sydney; the team in Belarus also undertook Quality Assurance at the line of code level. Additional team members were utilized on an ad-hoc basis in Vietnam and India, if a particular milestone or sprint required additional expertise not present within the core virtual team.

The make up of the teams in each location was a mixture of individuals operating as freelance developers as well as small teams such as in Belarus who were part of a larger company whose core business was providing development services to a range of global clients.

The overall structure of the team was hierarchical in nature, with all team members reporting to the Project Leader. The Project Leader in Sydney set the strategic direction, prioritized the feature development within the development pathway as well as controlling the release cycle of the software to customers. This role also provided the link between the company and the broader open source community.

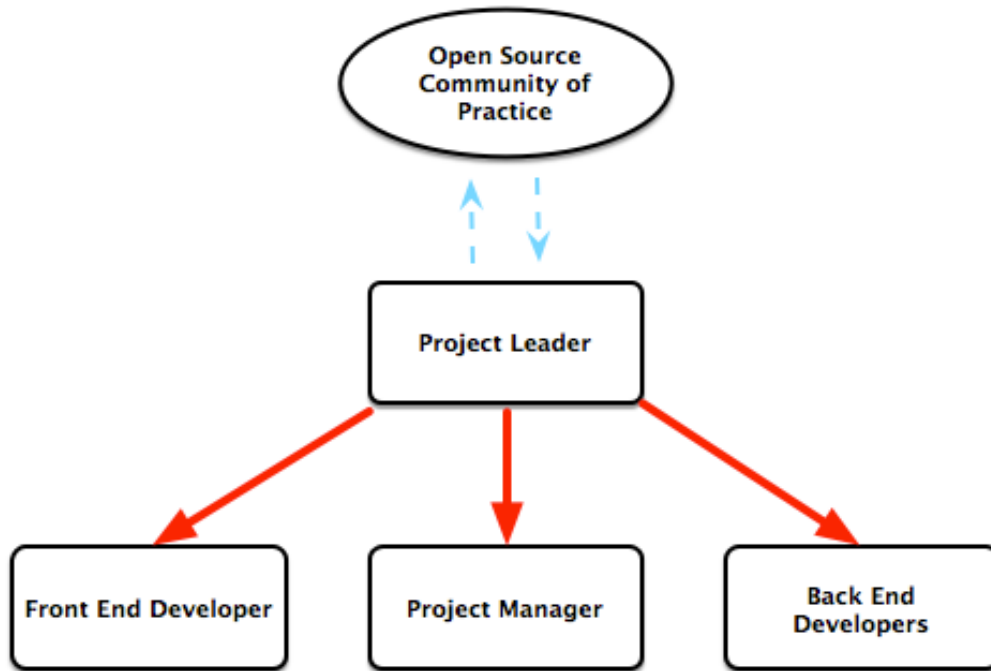


Figure 1: Existing Hierarchical Project Structure

The Approach

Due to the open ended nature of the overall development pathway of the software, an Agile ‘light’ approach was taken to manage the project elements and drive the delivery milestones. Taking its influence from Scrum based ‘sprints’ (Larson & Gray, 2011)^{iv} each milestone was mapped out 2 to 3 weeks in advance of commencement.

The steps involved in developing the next iteration of the software would run as follows:

1. The Project Leader would outline the key functionality that was to be included in the next release of the software.
2. Mock-Ups were created by the Sydney based project team of the GUI that was to be created, this creative process would help to inform or guide the back-end developers in terms of what this feature set needed to be able to do. These working models would also provide a template for the front-end developer to build a HTML version of the functionality.
3. A series of online discussions were held via video conferencing with the various teams around the world to discuss the brief, review the mock-ups provided and map out the time and resources required to complete the feature. This work estimation provided the duration of each of the ‘sprints’
4. Once the sprint had been formulated, the work was broken down into a series of tasks that could be assigned to individual team members to work on.
5. As each task within a sprint was completed, development teams undertook initial QA and testing, before the Sydney project team ran further QA tests focusing on usability.

Outside of the structured steps outlined above, less formal discourse took place between team members to work through the thoughts and ideas being formulated re: how to achieve the

desired outcome for each task within a sprint. This kind of discourse was usually done via instant messaging tools, either peer to peer or a group chat environment. Team members based in each location also held informal meetings to brainstorm ideas.

Lessons Learnt

The existing hierarchical structure of the Project Team, whereby the Project Leader has all sections of the team reporting to that role, placed an unnecessary burden on the Leader to be involved at the micro level of all decisions made within the team. It also limited the capacity for team members to interact horizontally with other team members, as all decisions needed to go via the Project Leader.

This hierarchical structure also meant that any pressure applied to the Project Leader, or if the Leader were taken out of the Project, it would have a catastrophic on the team as a functioning unit as the structure would simply collapse (Lipnack & Stamps, 1999)^v.

Whilst the Project Leader was aware that the existing structure was unworkable through reflection and observation of their own workload, the isolation caused by being at the top of the hierarchical structure meant that the Project Leader was not able to call upon the collective knowledge and experience of the group to find a more workable solution.

As Lipnack & Stamps identified, by transitioning from a hierarchical to a networked structure, the increased level of connectivity creates a more robust structure that is able to withstand pressure from within and without the project team. It also provides a path to create connections with communities of practice.

Of the overall issues encountered by the team, an inability to meet the 'sprint' deadlines proved to be a major impediment to the team's ability to deliver each new 'release' of the software on time, which therefore had a domino affect on each subsequent 'sprint', as they were also affected. Due to the diffuse spread of team members around the world, the Sydney based team was unable to foresee issues such as timeline breaches until the 'event' had occurred.

In seeking to identify through reflective practice why the team was not able to adhere to the prescribed deadlines, of the four approaches of thought within Action Learning (Yorks, O'Neil & Marsick, 1999)^{vi}, Experiential Learning is the most applicable within this project case study. By viewing the 'sprint' as the 'experience', the team can then reflect on the outcomes, reframe the experience and develop new ideas (Kolb, 1984)^{vii}.

Through critical analysis of these experiences, the Project Leader and Project Manager could reframe the 'sprint' process to develop greater transparency within Project tracking, allowing all team members to view the progress of each 'sprint' and for the Project Leader, it gave them an ability to set priorities at a more granular task level.

Another barrier that was encountered by the team was the individual behaviour of team members in reference to the quality and quantity of communication to the Project Leader and fellow team members. As virtual teams are so heavily reliant on electronic communication, a lack of communication on the part of individual team members can have a catastrophic effect on progress.

When assessing these ‘blocks’ in the process, whether they are cultural or perceptual (Megginson & Boydell, 1989)^{viii}, team members can benefit from reviewing and reflecting on past electronic communications and information repositories, both self-generated and the collective repositories of information. This serves as a tool for reflective practice at an individual and collective level, as well as providing newer team members the context and history needed to identify where they ‘fit’ into the overall project. Utilizing electronic communication as a form of reflection, like the writing of journals is utilizing writing as a way of learning (Boud, Keogh & Walker, 1985)^{ix}.

The way in which Quality Assurance and Control within the project environment was managed was that the Back End Development Team in Belarus managed the more automated process of developing and running test scripts across the software, this type of testing was highly technical in nature & tested the software at the level of individual lines of code. Whilst the Sydney based project team focused more on user testing at the client level, to ensure that the functionality as described within the documentation worked, as it should.

This structure of dispersing Quality Assurance and Control across different geographically dispersed sections of the team lead to a lack of connection and understanding of what each section of the team tested. This therefore lead to an increased risk of detailed line of code level test scripts overlooking a functional bug and for the user testing team, there was no transparency as to what aspects of the software code had been tested, increasing the likelihood of releases having technical bugs.

From a reflective practice perspective, ‘walking in the customer’s shoes’ (Hemetsberger & Reinhardt, 2006)^x, is an invaluable tool to ensure programmers are made aware of and consider the needs of the client when writing and testing software at the line of code level.

Another critical aspect of the Quality Assurance model was the lack of understanding or ‘knowledge transfer’ between the two sections of the team in Belarus and Sydney. This ‘dark side’ (Hase, Sankaran & Davies, 2006)^{xi} of knowledge management can have a negative impact on trust within the team and an inability to share knowledge.

Through experiential learning and the critical analysis of events such as ‘sprints’ and milestones, the Project Leader and Project Manager often sought administrative solutions such as changing the day to day processes in the production pipeline to fix issues that had arisen in terms of team coordination, communication and expertise. But by placing the emphasis on changing administrative processes rather than empowering the team members to utilize their own expertise to solve problems (Faraj & Sproull, 2000)^{xii}, it can jeopardise the teams’ ability to dynamically build knowledge and improve team performance.

Benefits of Proposed New Path

When seeking a model for building a better project team, one need not look any further than ‘Open Source Communities of Practice’, the democratic or networked structure of such entities, whereby each individual can contribute to collective tacit knowledge transfer through the process of interaction. (Hemetsberger & Reinhardt, 2006)^{xiii}

In transitioning from a hierarchical structure to a network structure, you create new connections within the team to encourage greater cohesiveness and strengthen the overall structure (Lipnack & Stamps, 1999)^{xiv}. By opening up new connections within the team, it enables the team to tap into the broader Open Source community through development platforms and Communities of Practice.

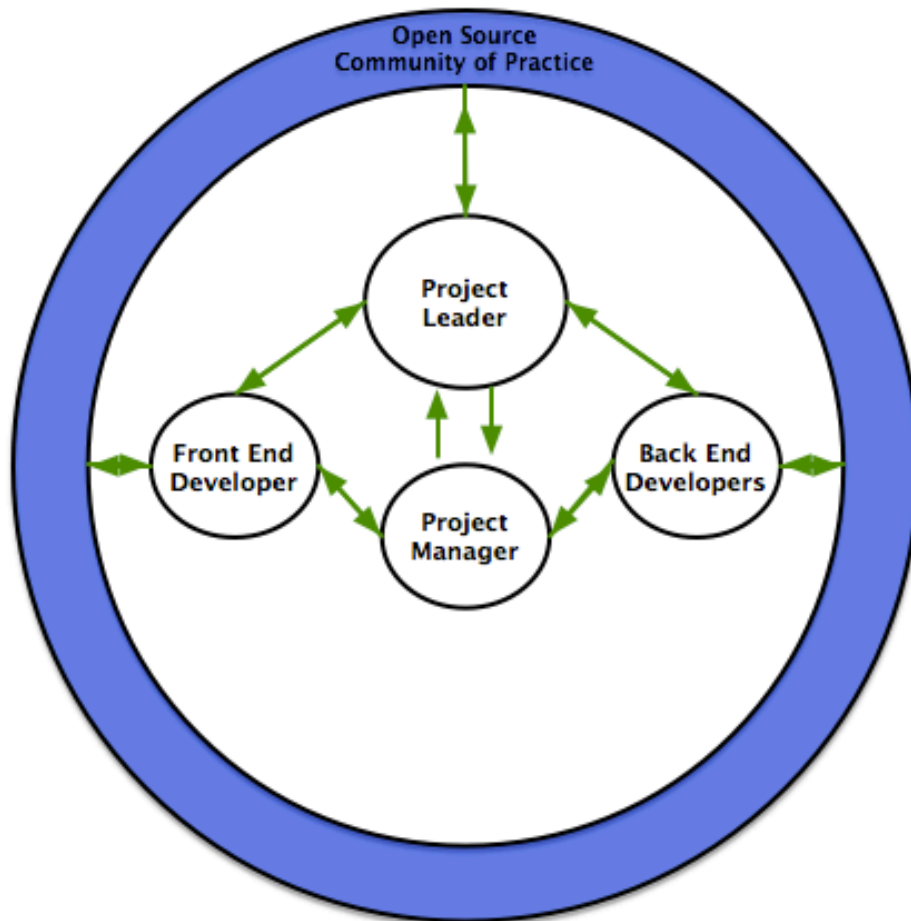


Figure 2: Proposed Networked Organisational Structure

In addressing the issue of ‘sprints’, which did not meet their prescribed deadline, the Project Leader and Project Manager began by assuming that each ‘sprint’ would be completed within its allotted timeline. They were not expecting the ‘sprints’ to exceed their time limits, this unexpected outcome led to a questioning of the approach or ‘reflection-in-action’ (Schon, 1987)^{xv}. The benefit of this process is that it then enabled a process of reframing how ‘sprints’ were managed and further experimentation led to a process which provided a more accurate estimate of work effort for each ‘sprint’.

Removing cultural or perceptual barriers to free and open communication within the team not only reinforces the connectivity of the team members; it has an added side benefit of building social capital amongst team members (Lipnack & Stamps, 1997)^{xvi}. As virtual teams rely so heavily on trust in one another, and trust cannot be imposed from on high, building social capital by extension builds trust and empowers the team to improve their overall performance.

The benefit to team members of seeking different perspectives such as that of the Client (Hemetsberger & Reinhardt, 2006)^{xvii}, within the context of the Quality Assurance and Control, also reinforces the lines of connection between the Project Team and its Stakeholders.

Encouraging team members to share knowledge within the context of Quality Assurance is an example of the knowledge creation process at work. Through the utilization of one of the four modes of knowledge conversion, that of Externalization which converts tacit knowledge to explicit knowledge, the team members can exist within the shared context (Ba) to create new dynamic knowledge (Nonaka, Toyama & Konno, 2000)^{xviii} for the benefit of the team and the wider organisation.

A virtual project team requires three basic elements, skills and expertise, administrative processes or structures and the ability to use that expertise as a resource to deliver the product.

When creating a new virtual project team within software development, the primary criteria for the recruitment of team members are their skills and expertise in Front End or Back End Development. But having the expertise within the team is not in itself the solution, as one must also know where that expertise lies and how to utilize it (Faraj & Sproull, 2000)^{xix}.

Conclusion

In summary, the structure of the project team and the approach taken within the chosen case study to develop an enrolment management system based on an Open Source Software platform, lead to issues of a hierarchical structure which placed far too much pressure on the Project Leader to be involved in all decisions, 'sprints' within the project milestones were not able to be adhered to, communication within the team could be impacted upon by individual team members not communicating and a lack of trust existed within the team itself.

Through the process of reflective practice, new possibilities could emerge in terms of the way in which the project team could be structured, experiential learning provided an alternate path to solve problems and an increase in connectivity within the team helped to build trust through the transfer of knowledge.

Whilst the project team had already transitioned to an Agile 'light' approach as the chosen Project Management methodology, to overcome the issue of software development projects where the final product is not clearly defined when the project commences. The team would benefit from exploring through reflective practice the approaches to team structures and increasing the connectivity and knowledge creation as utilized within Open Source Communities of Practice.

ⁱ Lipnack, J., Stamps, J., 1997, *Virtual Teams: People Working Across Boundaries with Technology*, 2nd Edition, Wiley & Sons, New York,

ⁱⁱ Nonaka, I., Toyama, R., & Konno, N., 2000, *SECI, Ba and Leadership: a Unified Model of Dynamic Knowledge Creation*, Long Range Planning, Elsevier Science Limited, Volume 33,

ⁱⁱⁱ Wikipedia contributors, 'Front and back ends', viewed 9th June, 2014,

http://en.wikipedia.org/w/index.php?title=Front_and_back_ends&oldid=610757898, Wikipedia, The Free Encyclopedia,

^{iv} Larson, E., W., Gray, C., F., *project management the managerial process* 5th edition, McGraw Hill Irwin, New York, NY, chapter 17, page 590

^v Lipnack, J., Stamps, J., 1999, *Virtual Teams: The New Way to Work*, Strategy & Leadership, Volume 27, no. 1, pp.14-19

-
- ^{vi} Yorks, L., O'Neil, J. & Marsick, V. J., 1999, 'Action Learning Through Theoretical Bases and Varieties of Practice', *Advances in Developing Human Resources*, vol. 1, no. 1, pp.1-18,
- ^{vii} Kolb, D., A., 1984, 'Experiential Learning – Experience as The Source of Learning and Development', Prentice Hall, New Jersey,
- ^{viii} Megginson, D., Boydell, T., 1989, 'A Managers Guide to Coaching', British Association for Commercial and Industrial Education, London
- ^{ix} Boud, D., Keogh, R., Walker, D., (editors), 1985, 'Reflection: turning experience into learning', Nichols, New York, Chapter 3 – Writing and Reflection, pp. 52-68
- ^x Hemetsberger, A., Reinhardt, C., 2006, 'Learning and Knowledge-building in Open-source Communities – a Social-experiential approach', *Management Learning*, Sage Publications, London, Volume 37 (2), pp. 187-214
- ^{xi} Hase, S., Sankaran, S., Davies, A., 2006, 'Overcoming barriers to knowledge management: visiting the dark side of organisations', *The actKM Online Journal of Knowledge Management*, Volume 3, Issue 1,
- ^{xii} Faraj, S., Sproull, L., 2000, 'Coordinating Expertise in Software Development Teams', *Management Science*, Volume 46, No. 12, pp. 1554-1568
- ^{xiii} Hemetsberger, A., Reinhardt, C., 2006, 'Learning and Knowledge-building in Open-source Communities – a Social-experiential approach', *Management Learning*, Sage Publications, London, Volume 37 (2), pp. 187-214
- ^{xiv} Lipnack, J., Stamps, J., 1999, 'Virtual Teams: The New Way to Work', *Strategy & Leadership*, Volume 27, no. 1, pp.14-19
- ^{xv} Schon, D., A., 1987, 'Educating the Reflective Practitioner', Jossey-Bass, San Francisco, Chapter 2, pp. 22-40
- ^{xvi} Lipnack, J., Stamps., J, 1997, 'Virtual Teams: People Working Across Boundaries with Technology', 2nd Edition, Wiley & Sons, New York,
- ^{xvii} Hemetsberger, A., Reinhardt, C., 2006, 'Learning and Knowledge-building in Open-source Communities – a Social-experiential approach', *Management Learning*, Sage Publications, London, Volume 37 (2), pp. 187-214
- ^{xviii} Nonaka, I., Toyama, R., & Konno, N., 2000, 'SECI, Ba and Leadership: a Unified Model of Dynamic Knowledge Creation', *Long Range Planning*, Elsevier Science Limited, Volume 33,
- ^{xix} Faraj, S., Sproull, L., 2000, 'Coordinating Expertise in Software Development Teams', *Management Science*, Volume 46, No. 12, pp. 1554-1568