

2008

PACIFIC NW SOFTWARE
QUALITY
CONFERENCE



COLLABORATIVE
QUALITY

OCTOBER 13-15, 2008

CONFERENCE PAPER EXCERPT
FROM THE

CONFERENCE
PROCEEDINGS

Permission to copy, without fee, all or part of this material, except copyrighted material as noted, is granted provided that the copies are not made or distributed for commercial use.

Quality Software Engineering: *Collaboration Makes the Experience*

Diana Dukart and Brian Lininger

Abstract

Well-executed collaboration can often make the difference between a quality software system and one that falls short. Strong collaboration skills are necessary for the varied roles software engineers are required to apply when undertaking software and information technology projects. In fact, close coordination and communication is needed in all aspects of the software process, from the initial customer requirements definition through the detective-like collaborative work required to triage and resolve errors in the final system. Any flaw in these lines of communication can greatly increase the risk of diminished quality in the end product.

During the process of completing the Oregon Master of Software Engineering (OMSE) Practicum Project, the authors applied a variety of collaboration styles and technologies commonly practiced on software engineering projects today. Project aspects addressed by such practices include distributed team member location, variability of member experience and skills, multiple modes of stakeholder integration, and constrained schedules and resources.

This paper examines the “lessons learned” from the full range of collaborative styles and technologies that were encountered during the project. These insights will provide other software professionals ideas and guidance on how to navigate similar challenges in their future collaborative software projects.

About the Authors

Diana Dukart is a senior software engineer with professional experience at Tektronix, Inc. She is a recent graduate of the Oregon Master of Software Engineering (OMSE) program at Portland State University and holds a bachelor’s degree in Computer Science from the University of Portland. Email: diana.dukart@comcast.net

Brian Lininger is a senior software engineer at Oracle, where he performs Quality Assurance activities on the Fusion Middleware Application Server. He is a recent graduate of the Oregon Master of Software Engineering (OMSE) program at Portland State University and holds a bachelor’s degree in Computer Science from the California State Polytechnic University at Pomona. Email: Brian.Lininger@Oracle.com

Introduction

Software development is inherently a social activity, as are most other types of product development. The core activities in any kind of product development are: understanding the requirements, defining the product specifications, and validating the correctness of the final product. Fred Brooks describes software development as:

“The essence of a software entity is a construct of interlocking concepts: data sets, relationships among data items, algorithms, and invocation functions... I believe the hard part of building software to be the specification, design, and testing of this conceptual construct, not the labor of representing it and testing the fidelity of the representation.” [1]

As Brooks states, the critical part of any software development project is the “conceptual construct”. For most software projects, a team of software engineers performs the project development, so the “conceptual construct” is a collaborative product influenced by each interaction of the members on the team. As a result, the collaborative experiences of the team will have a profound impact on the outcome of the software development project. Below are the experiences of a team of software engineers as they collaborate in the completion of a software development project as part of their educational practicum in software engineering.

Project Background

OMSE Overview & Requirements

The Oregon Master of Software Engineering (OMSE) program at Portland State University (PSU) is a graduate level software engineering education program designed for working software and information technology professionals. It provides participants both breadth and depth in the application of principles, methods, processes, and tools used in the dynamic and evolving software industry.

OMSE offers degrees, certificates, and professional courses tailored for working professionals. The degree program requires completion of 16 courses including a two term practicum experience. The practicum offers hands-on management and development experience in applying the skills learned, a variety of collaboration styles, and technologies commonly applied to software engineering projects today. Practicum participants are guided throughout the process by a member of the OMSE faculty and often work directly with an industry sponsor.

Sponsor Overview

Lifecom was the industry sponsor for the authors’ 2008 practicum project. They are a small Portland business developing software for clinical cognition patient diagnostics. Lifecom is revolutionizing medical informatics and patient care by developing a new form of artificial intelligence to provide knowledge, quality assurance, and safety to

medical practitioners. This ground-breaking software system is directly used for patient care as well as reducing administrative costs. With integrated patient records as well as built-in artificial intelligence using a large and current knowledge warehouse, Lifecom provides improvements on accuracy and timeliness of medical diagnosis.

Project Overview

As with all software systems, the Lifecom system needs to be updated periodically as new medical knowledge is obtained and added, as well as other enhancements. The practicum project was centered on this opportunity presented by Lifecom.

The Lifecom system runs on tablet computers used by medical practitioners. The applications running on these on-site tablets are what Lifecom needs the ability to update in a secure and timely manner. For these reasons, the practicum team named their project the Secure System Provisioning Project.

In addition to the main functional goal of updating the tablet applications, there are several ancillary goals. These include dealing with firewalls, running over the internet and the intranet, deploying custom updates for a specific customer, pushing out emergency updates, validation of file integrity, and producing an audit trail and version report.

Team Overview

The practicum team for this project was made up of four individuals who were completing the OMSE degree. The dynamics of this team lent well to the opportunity of gaining a wealth of collaboration experiences. These opportunities included:

- Distributed Team Communication
- Team Members with Different Backgrounds
- Learning a New Industry Sector
- Converging as a Team
- Team Expectations
- Work Product Management
- Working with Stakeholders

The following sections will delve into greater detail in each of these areas, providing a summary of the team's experiences during the project, their lessons learned and recommendations on how fellow engineers might approach similar challenges. Given that this was a newly formed team with members who had never worked with each other before, the experiences will generally be more valuable when forming a new team rather than for teams already working together.

Collaboration Experiences

A Distributed Team

Experiences

The OMSE program allows people to take classes remotely. For this practicum team, one member was based out-of-state with a job that took him all over the country during the course of the project. From the beginning, the team needed to figure out how to bring this individual into the project and allow him to contribute along with the rest of the team.

In addition to a permanently remote member, travel was necessary for another member of the team taking him overseas for three weeks during the project. Since the team was already set up to communicate remotely (at this point in the project the team regularly communicated via Skype), the travel was initially viewed as “not an issue”. Unfortunately, the team did not anticipate or plan for electricity shortages limiting communication with the team member traveling internationally. The result was delays on planned work.

On top of this, a third member of the team was “unavailable” for several weeks due to his child’s birth. As with the out-of-state member, this was planned from the beginning of the project. Unluckily for the team, it occurred the same time they were missing their other member overseas.

Lessons Learned: Communication & Risk Management

It is critical in planning for a project to include issues that may come up outside of the project. Risk management is essential in being able to control the many aspects that come up when running a project such as staying on schedule, unplanned events, and breakdowns in communication.

In general, the team did well in learning the new technologies to make remote communication work. Skype was chosen and integrated early in the project and it worked nicely for team meetings as long as the infrastructure was in place. The unfortunate experience of relying on Skype overseas when brownouts were occurring was not a problem with the technology, but rather a problem of the moment. In this case, moving the responsibilities of the critical tasks from the member traveling overseas to a local member temporarily would have been ideal. Although this was not done for this particular project, the team viewed it as a lesson learned to be used when executing future projects.

This lesson was similar to the other case of having a team member gone for family reasons. Planning ahead of time to move responsibilities to other available team members would have been easier for the members left with the work and possibly even kept the project on track. In addition, there could have been better scheduling on completion of work in anticipation of planned absences.

There were also lessons learned in working with people remotely. For certain tasks, such as brainstorming solutions, the optimal work environment is face-to-face communication. Ideas can be diagramed to communicate ideas more quickly, as well as non-verbal communication coming into play with hand gestures and facial expressions. The same work can be accomplished through remote means, but it takes more time. This extra time needs to be planned into the project schedule.

Recommendations

When a team includes remote members, there are several details that should be taken into consideration. First, any outside demands on the time or placement of team members needs to be included in the project plan. Communicating remotely must rely on many pieces of the system including not only the software, but the entire hardware infrastructure. With this being the case, there must be a contingency plan for when things do not work.

Second, the plan should always include a backup member to each person responsible for a critical piece of the project. This is to prepare for those unplanned events that take a person away from the job. The backup person can then step in and keep the project going and on schedule. When this occurs, completing the work the backup person had to put aside to pick up the critical tasks also has to be addressed. Although this type of issue can be different for each project, many times there is flexibility in when the non-critical work has to get done and a backup person at least keeps the project moving forward.

Lastly, extra time needs to be scheduled into the plan for communication. Although technologies like Skype are available and do aide in remote communication, it will never be as efficient as face-to-face meetings for some tasks.

Varied Backgrounds

Experiences

Each team member came from a different background. This included different industry sector experience as well as work experience and educational backgrounds. This diversity was an asset to the team because of the many skills that could be leveraged. However, at the same time, it became a stumbling block as people started assuming what they thought other team members already knew.

An example was the decision to use Web Services for a primary piece of the project solution. One member of the team already knew this technology intimately, but the others did not. Assuming the others knew more than they did, the knowledgeable member sometimes skipped critical details when explaining pieces of the solution. This caused confusion and misunderstandings within the team and negatively affected the necessary collaboration efforts.

This is just one example of many assumptions that were made throughout the project and each team member can claim at least one mistake. As teams try to gel and get to know each other, issues like this will arise and the experiences of this team were no exception.

Lessons Learned: Getting to Know Strengths and Weaknesses

Team members getting to know each other and learning each other's strengths and weaknesses in the beginning can help alleviate problems later on in a project. This not only includes learning which technologies each person knows, but also learning processes, industry experiences, and expectations of each member. Only after this initial assessment of the team is complete can the team begin to start assigning roles and responsibilities that really leverage the team's skills.

Collaboration can only take place when every member is in full understanding of the problem as well as knowing where each team member fits into the big picture. Establishing that foundation is essential for the success of a project as well as for the quality of the end product. This is especially true with such a diverse group of individuals who had never worked together on a project before.

Recommendations

When a new team comes together, they need to get to know each other first. A project kickoff meeting is a good first step in this direction. It not only presents the initial overview of the problem the team needs to tackle, but also begins the socialization between the team members.

A kickoff meeting is a good place for team members to talk about their background and experiences in the industry. Since this information is so important for a new team, the meeting agenda should explicitly include this discussion. Each member needs to bring up relevant knowledge that can help the project as well as areas where they have an opportunity to learn.

New Industry Sector

Experiences

Medical informatics was the industry sector of the practicum project. This is an exciting field to be involved in today because of the amount of growth and the endless opportunities it presents. In this case, no one on the team had any experience in this sector.

Fortunately, many of the skills needed to solve software problems are applicable across industry sectors. In fact, one of the main goals of the OMSE program is to teach such skills. The team was able to use this commonality to leverage the common language and processes learned in the OMSE program coursework. Each team member had completed the same core classes and this common foundation provided a framework to identify the important issues of the project confronting them.

The first important issue confronting the team was to understand the context and the goals of the project. Research became an important starting point in which all the members contributed. This built the foundation to the requirements elicitation and documentation efforts. This research not only included the specific problem domain, but also the medical informatics industry sector as a whole.

Lessons Learned: Learning Opportunities

The computer industry has grown a staggering amount since the early days of Alan Turing (English mathematician and philosopher, widely known as the father of modern computer science). [3] Used daily by most of the population, software has become a standard way of life. It is not plausible that any individual could know everything there is to know about every sector in the software industry. However, continuous improvement of software skills is a requirement because of the constant change. Therefore, learning about a different industry sector is an essential part in the career of software engineering. This project was just such a learning opportunity for the team.

Starting out as a team, the work of deciding which project to carry out or the initiation phase was completed. The team researched and read papers on current work in the medical informatics field. Challenges and future needs also came up in this exploration such as creating a National Framework for Healthcare and Electronic Medical Records. There are many laws and politics around this field as well such as the Health Insurance Portability and Accountability Act (HIPPA). None of these things actually became pertinent because of the nature of the project; however, the team did use what was learned in knowing what questions to ask the customers. It was only after asking questions, that the team knew things like HIPPA did not apply to the problem space.

Recommendations

Learning as much as possible about a project's industry sector is invaluable when eliciting requirements from the customer. Experts in any field have a tendency to leave out details that seem obvious to them, but are vital to the understanding of the problem for others not as well versed in the subject. Therefore initial research becomes essential before meeting with the customer. Once requirements elicitation begins, further research is also needed as subjects come up that are unclear to the engineers involved.

Converging as a Team

Experiences

Being able to successfully converge as a team in many ways comes down to having good communication, working together rather than as individuals, and having good leadership. A breakdown in any of these things can lead to floundering in a team's attempt to collaborate.

There are many ways in which a team can communicate; such as email, voice and video communication, and direct human interactions. While email is a good medium for transporting binary data, it falls short in communicating other important cues, such as sarcasm or voice inflections. It is important for teams to communicate regularly via a live mechanism. In addition, even voice communication (such as Skype) is not as productive as direct human interaction. Since the team was constrained to certain communication styles because of remote members, communication became a big challenge.

For one thing, voice as well as video still lacks the ability to whiteboard ideas, which is critical for architectural and design discussions. The team had no choice for their project and discovered that work can be accomplished electronically; however, it took more time than initially planned. Fortunately, other work tasks, such as requirements and planning, are more amenable to voice and video and did not impact the schedule.

The team converged in other ways, for instance, handling the administrative project tasks such as note-taking and configuration management. They decided to rotate these tasks, which worked well in getting acquainted with each other's work and forcing the members to work together.

Leadership was also rotated during the project. Each member was responsible for taking the lead role six weeks during the project. This gave each member a chance to exercise the leadership skills they had acquired through work experiences and skills learned in the OMSE program.

Lessons Learned: Communication and Management

There was some confusion and misunderstandings on project goals. Since the remote member could not attend local meetings with the customer, there was a limitation on how much he could take away from the notes written by the other members. In addition, the communications between the team members themselves were constrained to email and voice communication. In this case, more could have been done to explicitly talk about each member's assumptions and expectations on project goals. Not understanding each other's point-of-view led to some problems later in the project that could otherwise have been worked out earlier if better communication had happened. In some ways this was the result of the communication medium and in others; it was the result of the team learning how to work together.

As far as rotating tasks, the team learned that it worked okay for the small group of four, but did not feel it would work for a bigger team. In fact, four people were sometimes difficult to manage because of time constraints and other outside factors and responsibilities causing delays in communication and completion of work. A four-member team is close to the limit for this type of project management style to work.

Rotating the team leadership position did not work well in this project. There needed to be one person who always focused on the big picture in order to keep the work on course. For instance, when team members had conflicting project goals, a person

keeping an eye on the big picture could have picked up the signs of this conflict earlier than actually occurred. In rotating out the leader, this big picture focus was a little different for each cycle. In addition, setting up meeting agendas and following up action items could have happened better with a single person in this role.

Recommendations

There needs to be a single person with the specific role of focusing on the big picture and following up on work tasks. In line with this, roles and management style need to be adjusted for the type of project being addressed. In this case, the team was small, so rotating tasks worked. When this role is rotated among team members, there is a variation in the big picture view that will affect the direction of the project. The variation in task assignment and tracking leads to differing expectations among the team members, causing different working assumptions for the team members.

Communication of project goals and expectations should happen at the beginning of a project with these subjects explicitly called out. In addition, this type of communication should be revisited throughout the project to make sure everything is still on track and everyone is in agreement.

Team Expectations

Experiences

Whenever people are brought together for a project, each person brings with them unspoken expectations. These expectations can include gaining proficiency in a new technology or learning a new role in the organization. Unspoken expectations can work against a project if they remain unspoken, or if they are voiced in advance, can be leveraged to benefit the team.

This project team was no different in their possession of unspoken expectations. In some cases, expectations had no bearing on the outcome of a project; however, in this case a critical expectation was missed regarding how much the team could accomplish. After a scope for the project had been determined, the team moved forward with the project tasks but diverged approximately halfway through the project schedule. The problem was differing expectations of what the team could accomplish. Half of the team had set higher expectations and were proceeding to execute the project tasks in a manner that would allow them to exceed their planned scope. The other half of the team was executing their tasks to complete the project within the scope originally planned.

The result of these mixed expectations left the team frustrated and forced them to stop and evaluate their progress. It was at this point that the discussion of expectations occurred, taking into account the current state of the project. With this new insight, the team was able to refocus their efforts and complete the project as originally scoped.

Lessons Learned: Setting Expectations

Like all projects, communication is critical for success. It is important to identify the expectations of each team member at the outset of the project, as this will impact both technical and managerial decisions. In this instance, it would have been very useful to identify the differing expected project outcomes of each team member early in the project, as this could have been used by the team to avoid conflicts and manage project tasks more effectively. Knowing that half of the team had higher expectations would have allowed the team to partition their work in such a way that all members were satisfied.

Recommendations

As already discussed in the Different Backgrounds section, initiating a project kickoff meeting is a good start to any project. In this case, a discussion of each team member's expectations for the project needed to be included on the agenda as it would have averted much of the frustration felt by the team. The meeting needs to encompass all of the team members, but should be as hierarchy free as possible. The goal of this meeting is to allow the members of the team to state their expectations for the project so they can be used in planning and performing project tasks.

This exercise should not be used for employee evaluations but solely for the purpose of managing the expectations between team members and for allocating project tasks. In short, the main reason for knowing expectations is to gain insight and understanding of team members. In this example, differing expectations could have been the result of a team member expecting a new child during the project or another team member wanting a promotion as a result of the project's success. To get everyone working in the same direction, these expectations need to be explicitly discussed and managed.

Work Products

Experiences

Developing the Work Products within a distributed team can be challenging. Each team member was responsible for updating and/or reviewing each document multiple times during the course of its evolution. As each member of the team maintained a full-time work schedule and family life in addition to participating in this project, the schedule of each member was variable as well. As a result, the team needed a mechanism for managing the work being done on several documents by different members of the team on different schedules.

Lessons Learned: Document Management

Several mechanisms of collaborative document management were tried. One member used GoogleDocs for a document draft so that each team member could work on the document whenever it was convenient. The main issue with this was the rest of the team being unfamiliar with the tool and therefore, a learning curve had to be overcome before the entire team was effective using it. This included overcoming the security mechanisms that are necessary for protecting shared work.

Utilizing the merging capabilities of most source-code repository tools is another mechanism commonly used and was tried by the team. The team ran into issues; however, when any type of non-text document was used. A significant number of the documents the team produced were not simply text, but utilized diagrams and pictures to convey the intent and details of the system. As a result, a significant amount of work had to be redone when the repository could not resolve the conflict in the document versions.

One mechanism that worked well for the team was to use a Round-Robin style of work product updates. When work was required on a document, an explicit sequence was set for members to complete their updates. Each member would receive a notification that they could modify a specific work product and when their work was completed, they would notify the next team member in the sequence. This worked remarkably well for the team and became their default mechanism for the remainder of the project.

Recommendations

For a small sized team, a Round-Robin style of document management can be useful and effective. However, its use must be planned and explicit. A planned sequence must be stated at the outset and members must be held accountable for their work completion time as it can delay the entire team's work. Including the entire team on work completion notifications keeps the team informed of the current state and enforces a 'social justice' among team members. As already discussed, problems with communication or unavailability of team members must be part of the project plan as well and dealt with on a case-by-case basis.

Additionally, the ability to use common tools across a team cannot be ignored. If a process can be devised that allows the team to utilize their current tools and still perform the collaborative work necessary, this would be most desirable.

Multiple Stakeholders

Experiences

An experience of the team, which is likely common among most projects, was the existence of multiple stakeholders. In this case, there were two stakeholders that were looking for different outcomes from the project.

The first stakeholder, the project sponsor, was much like a typical customer. They provided as much input on system requirements as possible, participated in reviews, and provided feedback wherever possible. Their expected output from the project was the same as most customers, in this case, a set of work products that they could utilize in their company to extend their product-line. They had little interest in the project management side aspect, such as the work that went into generating a useful project plan and requirements document. They were interested in the end products: architecture, design, and prototype of the system.

The second stakeholder, the practicum instructor, was evaluating the project from a different perspective. In addition to providing input and evaluating the set of work products, the instructor's goal was to evaluate the team's effectiveness working together and their application of the knowledge the team had acquired during their coursework. The output from the team for this stakeholder was status updates and presentations that communicated the how and why the team had made certain decisions. In addition, the practicum instructor was interested in the internal work product qualities, such as the project plan and requirements documents, as part the evaluation process.

Lessons Learned: Stakeholder Input Benefits

Any software product that is going to be competitive in today's world needs a set of unique features that set it apart from what is already available on the market. With this being the case, each software project presents learning opportunities, such as applying software a little differently to a common problem or learning about a different industry sector.

In this case, the team learned both of those things and much of this input came from the project sponsor. It was a great opportunity for the team to learn from individuals who had been in the medical informatics sector for years. In addition, they were able to take other experiences in applying software updates and learn how to apply that knowledge in a new and different way.

As well as the input benefits from the project sponsor, the team also leveraged and benefited from the OMSE instructor's input. With this guidance and involvement, the team was able to maintain a macro-project focus and utilize processes that prevented them from falling into the typical pits of micro-task optimization.

Another learning experience for the team was dealing with the occasional conflict between the stakeholder inputs. In one instance, a disagreement of the detail of the project background came into question. One customer had an in-depth knowledge of the project domain, while it was new for the other. The lesson here was learning how to handle the expectations and needs from two very different perspectives. In this case, the team chose to include more detail so everyone could understand the project, even though this included details the one stakeholder already knew. In this way all parties were working from the same project basis.

Recommendations

Whenever a project has multiple stakeholders, there are two options:

- 1) View the different inputs as liabilities, or
- 2) View them as an opportunity to improve the project outcome.

In this case, the additional stakeholder inputs were an asset leveraged for the success of the project. Stakeholders provided information to the team that prompted them to ask further questions, define further requirements, and expose more views of the system.

As a result, the final product for both customers turned out more comprehensive and of better quality, a real-life example of “the whole is greater than the sum of the parts.”

In addition, differing inputs between stakeholders is a common occurrence that engineers will need to address. Collaboration skills again come into play, as the engineer must work with multiple stakeholders to come up with a solution. The job of the engineer after all, is to solve problems.

Conclusion

Collaboration is an essential piece to any software project for reasons of the inherently social activities involved. As such, there will be many experiences with lessons that can be taken away. In the case of the practicum project described in this paper, these lessons included learning how to work with remote members, leveraging different backgrounds, learning a new industry sector, converging as a team, handling team member expectations, managing project artifacts, and valuing customer input.

It is always less painful and more efficient to learn from the experiences of others. With the descriptions of experiences, lessons learned, and recommendations; this article hopes to assist professionals in similar circumstance to improve and enhance the quality of software project knowledge and understanding for the future.

References

[1] Brooks, Fred. “No Silver Bullet – Essence and Accidents of Software Engineering.” Proceedings of the IFIP Tenth World Computing Conference (1986): 1069-1076.

[2] Dukart, Diana, Lininger, Brian, Pierson, Derek, and Subramanian, Mahesh. “Secure Provisioning System.” OMSE Program Practicum Project. Portland State University, 2008.

[3] Lewis, Harry R., and Papadimitriou, Christos H. Elements of the Theory of Computation. New Jersey: Prentice-Hall, 1998.