Shiladitya Bhattacharyya

Sbhttch2 676424081

Professor Marinov

429

5 May 2014

Writing Assignment 6

This semester, the course staff for CS 428 has tasked the students with designing our own pieces of software and constructing said software following design processes of our choosing. To accomplish such an ambitious goal, the staff for CS 428 has laid out steps and principles for the class to follow so that the students make continuous progress on their respective projects. These guidelines also ensure the project ideas are feasible (with respect to the given time frame) in the first place. **Among the guidelines set out by the staff, the most important include writing a project proposal, forming a team, and following and documenting a certain design process which each team chooses after the team's formation**.

The first and most important task given out by the staff this semester assigned each student to form a pair with another student and write a project proposal outlining an idea and rough implementation of a possible project. Since we had the liberty to pick our own partners, I chose to work with a friend who also enrolled himself in CS 428 this semester. Together, we agreed upon a project which involved designing a piece of software which could help an individual track his daily habits and routines. However, we needed a lengthy deliberation to happen upon this idea and as a result, we did not have much time left to both write up our proposal with the proper amount of effort required and submit the finished proposal by the deadline. Since the deadline was non-negotiable, we decided to compromise our quality of work and submit a less than adequate proposal. Although completion determined the majority of the grade of the proposal and we did submit a complete proposal, the quality of the proposal disappointed us. The proposal served to persuade the staff as well as other students that our project deserved an entire semester's worth of work and consideration. **We knew that even though we had an interesting and thoughtful idea for a project, our writing could not demonstrate the value of the project it described to its reader**s.

For example, one of the sections of the proposal had to describe the motivation behind the project on which the team chose to work. This section is undoubtedly the most important among all the other sections of the proposal because if the reader finds the motivation of the project unclear or uninteresting, what reason has he to dedicate his time to the software project described by that proposal? Due to the lack of time, our "Motivation" section did not adequately reflect the benefits of designing software that helped one track his habits. Our ideas were present in our writing, but their organization lacked clarity and our vocabulary did not inspire the reader as well as we had hoped. If given the chance to rework the proposal, we would place most of our attention upon improving the organization of the content in our motivation and modifying the style of our entire proposal in general. Currently, it reads more like a narrative rather than a persuasive piece. **The obvious problem with this style lies in the fact that the proposal cannot achieve its ultimate goal: to persuade others to work on our project!**

As we expected, the course staff rejected our proposal. This meant that we needed to find a team whose proposal had both been accepted and interested us. This proved to be a rather arduous task because we found most of the accepted proposals uninteresting or too ambitious. **Thus, we instead shifted our focus into searching for an accepted pair with whom we would work well**. After some time, my partner recognized a pair of students he had worked well with during the previous semester so we requested to join their team. Thankfully, they accepted our request. Since the team would have good synergy, even though their project didn't appeal greatly to us , we reasoned that we could still design something useful and innovative.

Upon joining the team, we needed to complete multiple tasks. **For instance, we needed to meet all our members and discuss some fundamental issues such as when to schedule our weekly meetings, how to delegate work, and most importantly, which design process to use this semester**. In the previous semester, all the teams needed to follow the software design process known as "Extreme Programming", or XP for short. For some background, how a group of people should work together and the steps they should take when designing a piece of software are outlined by a software design process. This includes instructions for how to distribute work, what kind of coding style to follow, how often teams should meet, and many more administrative and technical rules.

XP exemplifies a class of design processes known as "Agile" processes. **This process dictates rules designed to allow the members of the team to begin coding as soon as possible as opposed to spending a lot of their time on discussing requirements of their project and other administrative issues**. More specifically, XP requires the team to schedule "iterations" where each iteration, the team will meet once and discuss what tasks they can complete that iteration as well as how well the previous iteration went for them. These iterations are typically two week intervals, but can be larger or smaller depending on the size and needs of the team. If the team thinks a certain task may require multiple iterations to complete, then XP dictates that the team should decompose the task into smaller subtasks, each of which the team can manage in one iteration.

Looking into finer aspects of Extreme Programming, XP also has specific requirements regarding how teams should code. The requirement that two members of the team to sit together at one machine and code whatever task the team assigned them is known as pair programming. XP advocates that pair programming speeds up the design process by ensuring that team members think through their coding ideas well and by reducing the number of syntactical mistakes made while coding. Another rule of XP which aides in accelerating the design process is the standard that pairs should always code "the simplest thing that works" first, rather than try and construct an all-encompassing complicated solution. XP says to revisit old code and refactor when necessary to handle emerging problems. **Further coding practices for which XP advocates include test driven development, automated unit tests, and continuous integration.**

After some discussion with my team, we have agreed to follow another Agile design process known as "Scrum". **Scrum and XP share many similarities including working in iterations, encouraging refactoring, and using continuous integration, but differ on other fundamental ideas**. For example, Scrum does not hold the standard that members of development teams should pair program. Furthermore, those developers who believe Scrum is the ideal software design process for development teams claim that projects designed following the guidelines of Scrum release quicker and have fewer bugs than those projects which teams design following other software design processes. Scrum followers justify their ideals by noting that Scrum specifically tailors to unfamiliar or unexpected situations that teams may encounter during the development process. These problems often disrupt traditional software development processes and push back the release date of the product under development. The design principles dictated by Scrum, however, encourage developers to always review their work and compare the direction they plan on taking against the requirements of the project at that time. That way, the developers never lose sight of the final goal and can make continuous progress towards completion.

Although we decided to follow Scrum, we do plan on making minor adjustments to our design process as the semester progresses. Our reason for not following all the principles of Scrum stems from the fact that our team does not have all the ideal criteria that teams following Scrum should have. For example, the project we plan on working on this semester requires a working knowledge of web development. Most of our members (including myself) are rather inexperienced in this field. On the other hand, Scrum as a design process works best when all the members of a design team have a nice grasp on the knowledge and technologies a certain project may require. For this reason, we plan on adopting the pair programming strategy from XP for the first few iterations so that those of our members who have more experience with web development may provide their help and guidance to some of our more inexperienced members. **After the majority of the team gets up to speed with most of the technologies required, we plan on switching back to a purer form of Scrum**.

In conclusion, I feel rather excited about learning more about how to implement the project I've been assigned. I believe writing the project proposal and having to join a development team gave me a glimpse into the steps required to begin any kind of development process, whether for software or some other type of project. Furthermore, I have wanted to learn web development for some time now and diving right into this semester’s project should certainly expedite my learning. **I look forward to these upcoming weeks and hopefully our team can design something meaningful over the short time period we have**.

1. Among the guidelines set out by the staff, the most important include writing a project proposal, forming a team, and following and documenting a certain design process which each team chooses after the team's formation
2. We knew that even though we had an interesting and thoughtful idea for a project, our writing could not demonstrate the value of the project it described to its readers
3. The obvious problem with this style lies in the fact that the proposal cannot achieve its ultimate goal: to persuade others to work on our project
4. Thus, we instead shifted our focus into searching for an accepted pair with whom we would work well
5. For instance, we needed to meet all our members and discuss some fundamental issues such as when to schedule our weekly meetings, how to delegate work, and most importantly, which design process to use this semester
6. This process dictates rules designed to allow the members of the team to begin coding as soon as possible as opposed to spending a lot of their time on discussing requirements of their project and other administrative issues
7. Further coding practices for which XP advocates include test driven development, automated unit tests, and continuous integration
8. Scrum and XP share many similarities including working in iterations, encouraging refactoring, and using continuous integration, but differ on other fundamental ideas
9. After the majority of the team gets up to speed with most of the technologies required, we plan on switching back to a purer form of Scrum
10. I look forward to these upcoming weeks and hopefully our team can design something meaningful over the short time period we have