CS 3212 Course Management Proposal

Josh Westmoreland, TA/GRA

03/24/2009

Note: this document assumes that the reader has at least a working knowledge of Scrum

Introduction

My proposal is as follows: We should use a modified form of Scrum to manage CS 3212, Software Engineering II, in the fall semester of 2009. Scrum in its pure form would not really be feasible for a classroom setting, even a software engineering class, but I believe if we modify it a bit then it could be quite useful for this class and for use in future classes. We would have to modify the nature of the scrum meeting, how we use the artifacts (burn down chart, sprint log, etc.), but I believe it can be done and done well. There also some other issues to be covered if this plan was to put in place, but I think those can be successfully dealt with as well.

The Nature of the course

The description of the CS 3212 from the University catalog is as follows:

"Software development methods for large scale systems. Management of software development projects. Software engineering standards. Students are expected to complete a large scale software project."

Scrum naturally lends itself to a course of this nature due to its intrinsic qualities. Scrum is arguable agile software development at its finest. It is tailor-made to handle large scale projects such as the one that students are expected to complete in this course. I believe Scrum to be the perfect method for managing this course.

Using Scrum to manage this course would also require the students to learn the details of agile, which they should already know from CS 3211, Software Engineering I, as well as the details of Scrum itself and other methodologies currently used by professional programmers. This fulfills the requirements of the students learning about "*Management of software development projects*" as well as "*Software engineering standards*".

How do we modify scrum to suit our needs?

The primary question associated with this proposal is how we modify Scrum to suit the needs of a class. I believe this can be done quite easily. I will break this down in this section.

Assigning Teams

- Quite simply, we randomly, or at least pseudo-randomly, divide up the students into groups of 4 to 5 and let them assign among themselves, or assign for them, the part of the project they are going to work on for the team.

Roles

- Product Owner
    - The Instructor, but this and the Scrum Master role can be shared to some degree by the Instructor and the TA
- Scrum Master
    - The TA, but this and the Product Owner role can be shared to some degree by the Instructor and the TA
- Scrum Team
    - The students, obviously, divided into teams working on the same project concurrently

The Sprint(s)

- Since each development cycle in Scrum, a sprint is usually about 4 - 6 weeks, or 30 - 45 days, depending on the preference of the development house, I suggest we have the students develop the project in 2 - 3 sprints with the length of the sprint to be determined by both the project and the length we decide upon for each sprint. At the end of each sprint we will have a short class-wide sprint retrospective, perhaps lasting as long as one class meeting, where we discuss the sprint, specifically what went well and what didn't as well as any other problems the individual teams might have had during the sprint such as participation, etc.
- To split this up even further, we could have "mini sprints" inside of each sprint lasting about a week each that way we could have something due for them every week as we have done in CS 3211 and this would make the proposed schedule work far better.

Schedule

- Since this a two-day-a-week class I suggest we do the following:
  Monday or Tuesday: Scrum meetings for each team (5 - 7 minutes each)
  Wednesday or Thursday: Lecture & due date for the current iteration/milestone

Artifacts

- The burn down chart

    - This is something we could have as class-wide so that everyone will know where the project currently stands
- The sprint backlog
    - We wouldn't have to come up with this completely before the sprint. We could work it similarly to how we do now and brainstorm on the lecture days since that's what we do primarily now in relation to the current project.
- The product backlog
    - Once again, this would be a class-wide thing that we could update as we complete requirements. As always with agile, we can brainstorm and add to or remove items from this as we see fit.

Isn't there supposed to be a scrum meeting every day?

- Yes, but there is a very simple way around this. ..
    - We set up a system and require the students to make a number of scrum reports every week, say at least 3 and as many as 5, on Moodle, or whatever system we decide to use, where they answer the three essential scrum questions
        1 - What did you do today?
        2 - What are going to do tomorrow?
        3 - What impediments are keeping you from completing your objectives?
    - We still have out formal scrum meeting with each team once a week.

<u>Issues that might arise</u>

There are at least a couple of things the students will need to know before this will become feasible:

- Subversion, and keeping up with the latest version of the project
- Assigning roles within a team, i.e. dividing up the work fairly

I think if the students can be taught how to do these things, there might be other issues I am not considering at the moment, I believe this course can be properly managed by the modified form of Scrum I am suggesting. Managing this course in this way would fulfill all of the requirements for the course, at least those I am privy to, and I believe our students would learn a great deal in the process.

<u>Conclusion</u>

I of course understand that conducting a class in this manner will require far more planning than I have done here. The purpose of this document is to merely be a rough outline of how to proceed.

Fin.