Review of Development Process

Description

The development process used during this project was agile. Features were typically added to the project incrementally (as opposed to iteratively) during 2–3 week long sprints.

Reflections on Agile

Throughout the term, group members felt that the overhead associated with the agile process was both a burden and a blessing to the development cycle.

Some aspects of agile development did improve our ability to develop significantly. For example, we felt that writing (and sometimes rewriting!) user stories as feature statements, and having to plan goals and assign tasks for each sprint, were all worthwhile, since they forced us to think about the high-level structure of the project on an ongoing basis.

Some aspects of agile development that struck group members as less productive. For example, in our group we felt that estimation of user stories was better settled through group discussion rather than Planning Poker, since some members were not yet familiar enough with CodeIgniter to be able to make realistic story estimates. We were also able to use these talks to help each other become acquainted with CodeIgniter and the MVC paradigm.

Some aspects of agile development struck us as redundant. For example, having a burndown chart seemed pointless when a glace at the project/sprint backlogs could tell us the same information.

Overall, we felt that a fully-featured agile development process would have been more appropriate for a larger-scale project, or for a development team with no distractions (i.e. other courses). Most of the person-hours devoted to this project were spent on the agile process itself, rather than developing the functionality of our product. Regular group meetings were difficult to maintain during midterm season.

Although Test Driven Development (TDD) was encouraged for this project, it was difficult to pursue due to the technologies we employed. To our knowledge, the CodeIgniter framework does not support unit and integration testing to the same extent as more traditional languages such as Java. Integrating automated testing into our development cycle was thus problematic, and ultimately deprecated in favour of repeated manual testing, often done collaboratively. In hindsight, not having followed TDD so rigorously resulted in a pattern of sharp development spikes, followed by long plateaus of testing. It allowed bugs to creep into the application (usually with respect to form validation), which we then had to hunt down and fix. With a test suite, these could have been caught much earlier. However, we feel the effort of writing automated tests would have cut down our development time even further, and our feature set would be much smaller as a result.

A Note on Communication Technologies

One technology we found especially useful for group communication was Facebook, since we all have smartphones that allow us to instantly reply to questions and comments. Many design decisions, answers to coding questions, or proposals for bug fixes were originally proposed in a Facebook post.

Google docs was also extremely useful for editing collaboratively. Our ‘Final Task List’ (a pre-deployment check-list of remaining tasks) was done entirely in Google docs.