XP Method

Ponder 04

# Reflection

## Viability

I love the XP method, and think that this method is viable for most small and middle sized development. It can be applied to large development efforts as well, but careful planning needs to be put into how to divide the large effort into multiple smaller efforts. I feel like this plan is viable for the contract described in the assignment.

## Resource Usage

We’re not really using our UX designers or our technical writer with this method, though in an actual company, we would probably be consulting with the UX designers as needed while developing. Other than that, I feel like we’re making excellent use of our developers, especially when pair programming is utilized well to spread knowledge throughout the whole team.

## Risks

If developer discipline is low, this method can degrade into the code and fix methodology, and the project is extremely likely to fail. This method is also great at bringing workplace rivalries/tensions to the forefront, and managing that conflict can be difficult.

# Meetings

## Release Planning Meeting

**Attendees:** Software Engineers, and the Customer. **Agenda:** Determine the user stories for the project and estimate each story in terms of ideal programming weeks [4]. Create spikes where estimates are uncertain [4]. Determine the project velocity [4]. Assemble this into a release plan. **Goal:** Lay out the overall plan for the project. **When:** The first work day after receiving the contract for the project, or after a significant project velocity change [4]

## Iteration Planning Meeting

**Attendees:** Software Engineers and the Customer. **Agenda:** Customer selects failed acceptance tests and user stories from the release plan from most value to least valuable [3]. Each user story has a point value, and the total value of selected stories cannot exceed the project velocity [3]. Make acceptance tests for the iteration around the stories and previously failed tests [2]. Allow Software Engineers to self-select tests and estimate their difficulty [3]. Assemble this into an iteration plan. **Goal:** Create an iteration plan based on tasks derived from a customer selected user story.  **When:**  Every other Monday at the start of the day.

## Standup Meeting

**Attendees:** Software Engineers **Agenda:** Discuss what each Software Engineer will be working on for the day. Organize pair programming. Identify pitfalls or other problems that will require a longer meeting to resolve. **Goal:** Get the team on the same page at the start of the day and organize our efforts. **When:** Daily at the start of the day.

# Documents

## Release Plan

**Author:** Customer and Software Engineers **Audience:** Customer and Software Engineers **Purpose:**  Lays out the overall project [4] **Deadline:** The end of release planning

## Iteration Plan

**Author:** Customer and Software Engineers **Audience:** Customer and Software Engineers **Purpose:** Plans which acceptance tests are to be used in the current iteration. **Deadline:** The end of iteration planning.

## User Story

**Author:** Customer [5] **Audience:** Software Engineers **Purpose:** Serves as requirements. 60 – 120 User Stories are selected for a release plan [4]. **Deadline:** Before the release planning meeting, and later as needed.

## Acceptance Test

**Author:** Customer and Software Engineers [6]  
**Audience:** Customer and Software Engineers **Purpose:** Black box tests to verify that a user story has been met. 1-3 weeks’ worth of acceptance tests are put into an iteration plan.  **Deadline:** Created as needed during iteration planning.

## CRC Cards

**Author:** Software Engineers [6]  
**Audience:** Customer and Software Engineers **Purpose:** Figure out how to design software that passes an acceptance test. **Deadline:** As needed while developing towards an acceptance test.

# Roles

## Software Engineers

**Qualifications:** A Software Engineer needs a BS degree in a Computer Science related field, to understand various methods of software development, including the XP model, to know how to assess and mitigate risks related to software development, and be able to design, code, and test software. **Responsibilities:** Aid in the creation of release plans, iteration plans, and acceptance tests. Pair program with another Software Engineer to develop software that passes acceptance tests, using CRC cards to design as necessary. **Who:** Owner, Abe, Britney, Claire, Doug, Emily, Frank, Grace, Holly, Ingrid, Jack, Keith and Larry.

# Checkpoints

## Iteration

**Time Estimate:** 1 – 3 weeks**,** as determined by the team, **Exit Criteria:** Whatever span of time as was determined by the team has passed.

# Bibliography

[1] Wells, D., “Extreme Programming: A gentle introduction”, [Online Available]: <http://www.extremeprogramming.org/index.html>

[2] Wells, D., “Manage Your Goals Instead of Activities”, [Online Available]: <http://www.agile-process.org/byfeature.html>

[3] Wells, D., “Iteration Planning”, [Online Available]: <http://www.extremeprogramming.org/rules/iterationplanning.html>

[4] Wells, D., “Release Planning”, [Online Available]: <http://www.extremeprogramming.org/rules/planninggame.html>

[5] Wells, D., “User Stories”, [Online Available]: <http://www.extremeprogramming.org/rules/userstories.html>

[6] Wells, D., “Acceptance Tests”, [Online Available]: http://www.extremeprogramming.org/rules/functionaltests.html