Spiral Method

Ponder 03

# Reflection

## Viability

I feel like this plan is the most viable plan discussed thus far. The Spiral method is so flexible that in can incorporate other methods as is deemed necessary during risk determination/evaluation. This flexibility helps improve the chances that the right methods are chosen to resolve the inevitable problems we will face.

## Resource Usage

Again, the flexibility of the Spiral Model shines here. We have a pool of developers that can be assigned to do whatever is needed. This requires competent software engineers, rather than your run-of-the-mill code monkey, but if you can trust your employees, you’re going to get better resource utilization with this method.

## Risks

One of the main risks of this method is related to the Team Lead. If your Team Lead doesn’t have a solid grasp of project management and software development, the Spiral method’s flexibility can become a hindrance rather helper. Hopefully the team will be making decisions as a whole, which would help mitigate that risk, but it’s still possible for the Team Lead to take things in an unproductive direction if he/she isn’t careful. One other risk of this method is “spiraling out of control”, or stated better, not stopping the project when it should be stopped.

# Meetings\*

## Objectives

**Attendees:** Team Lead, Developers **Agenda:** Identify objectives, alternatives and constraints. **Goal:** Determine if software can be feasibly developed to meet the objectives. **When:** The start of each phase.

## Risk Assessment

**Attendees:** Team Lead, Developers **Agenda:** Evaluate the risk of each alternative, choose an alternative, and form a development strategy to mitigate the risks associated with the chosen alternative. Record these details in an Operational Concept Document.  **Goal:** Evaluate risks, decide what to do, and decide how to do it. **When:** Immediately following the Objectives Meeting.

## Phase Planning Meeting

**Attendees:** Team Lead, Developers **Agenda:** Plan out the next phase, if another phase needs to happen. Create a requirements plan, development plan, or integration and test plan, depending upon the needs of the next phase. **Goal:** Plan the next phase. **When:** After validating/verifying the current round is complete.

\* There are likely to be many more meetings than this, but due to the flexible nature of the Spiral method, enumerating them here is infeasible. There will likely be a requirements meeting, a design meeting, test plan meeting validation meetings, and others, but the exact implementation depends upon the approach decided on in the Risk Assessment meeting.

# Documents\*

## Paper Models/Prototypes

**Author:** Developers **Audience:** Developers, Client **Purpose:** Uncover risks, demonstrate project/feature feasibility, and aid in design. **Deadline:** None. Models/prototypes are to be created on demand as needed.

## Operational Concept

**Author:** Attendees of the Risk Assessment meetings. **Audience:** Developers **Purpose:** Detail objectives, constraints, alternatives, risks, risk resolutions and results, and plans/commitments for the next phase.  **Deadline:** The end of the risk assessment meeting.

## Requirements Traceability Tool

**Author:** Developers **Audience:** Developers **Purpose:** Establishes the traceability between requirements, design elements, code elements and test cases [1].  **Deadline:** This is used throughout the project. Created over time throughout the project.

## Unit Development Folder

**Author:** Developers **Audience:** Developers **Purpose:** Collects all artifacts from a software unit into a folder and provides a management template for tracking the programmer’s scheduled and actual completion of each artifact [1].  **Deadline:** This is used throughout the project. Created over time throughout the project.

**\*** Other documents may need to be made based on the details decided on during the Risk Assessment meeting. Due to the flexible nature of the Spiral method, enumerating all of these documents is infeasible, but they are likely to include some kind of requirements, life-cycle plan, design, development plan, and integration/test plan documents.

# Roles

## Team Lead

**Qualifications:** The Team Lead needs experience managing projects, good communication skills, and a deep understanding of the Spiral Method, as well as fulfill the requirements of a Software Engineer. **Responsibilities:** The Team Lead is responsible for guiding the team through each round of the spiral model, determining when the spiral is complete, and presiding over the each of the meetings. The Team Lead reverts to a Software Engineer whenever he/she has time after fulfilling these responsibilities. **Who:** I will fulfil this role.

## Software Engineer

**Qualifications:** A Software Engineer needs a BS degree in a Computer Science related field, to understand various methods of software development, including the Spiral model, to know how to assess and mitigate risks related to software development, and be able to design, code, and test software. **Responsibilities:** Identify risks related to the current round of spiral development **Who:** Abe, Britney, Claire, Doug, Emily, Frank, Grace, Holly, Ingrid, Jack, Keith, and Larry.

# Checkpoints

## Round 0

**Time Estimate:** 3 days **Exit Criteria:** Round 0 has completed determining and evaluating objectives, alternatives and constraints; developed and verified the current product, and has planned the next phase.

## Round 1

**Time Estimate:** 2 weeks **Exit Criteria:** Round 1 has completed determining and evaluating objectives, alternatives and constraints; developed and verified the current product, and has planned the next phase.

## Round 2

**Time Estimate:** 4 weeks **Exit Criteria:** Round 2 has completed determining and evaluating objectives, alternatives and constraints; developed and verified the current product, and has planned the next phase.

## Round N

**Time Estimate:** 6 weeks **Exit Criteria:** The current round has completed determining and evaluating objectives, alternatives and constraints; developed and verified the current product, and has planned the next phase, if there is a next phase. (This phase repeats, incrementing N each time, until the project is complete)