02 Ponder: Spiral Application

Ryan Dockstader

# Meetings

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Meeting name** | **Who is required to attend?** | **What will be the agenda of the meeting?** | **What do you expect to accomplish during the meeting?** | **How often or on what event will the meeting be called?** |
| **Objectives meeting** | **Software Engineers** | **Discuss the objectives for the week** | **Identify Objectives** | **Every Other week on Monday Afternoon** |
| **Progress Report** | **Software Engineers** | **Discuss the progress made each week** | **Progress reports for development and test phase** | **Every other week on Friday, Opposite week as objectives** |
| **Project Review** | **Team Leads, Owner** | **Review and demo the project** | **Determine the progress of the project** | **Same day as the progress report, after that meeting** |
| **Risk Analysis** | **UX Designers, team leads, owner** | **Discuss risk potential for the current objective** | **Determine risk, and product viability** | **Same day as objectives meeting, before that meeting** |

# Documents

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Document name** | **Who will author the document(s)? Are there additional contributors?** | **Who will read the document(s)? Who is the intended audience?** | **What purpose does the documents(s) serve?** | **Is there a deadline for this document or a time period when the document is relevant/useful?** |
| *System Requirements* | *The technical writer as well as the lead developers* | *The client and the development team* | *List the requirements of the system as a whole* | *The deadline is 2 weeks in, and it will be useful through the entire process* |
| *Software Requirements* | *The technical writer as well as the lead developers* | *The client, owner and development team* | *List the requirements specific to the software* | *The deadline is 4 weeks in, and it will be useful through the entire process* |
| *Testing Report* | *The software testers will be the author of this group of documents* | *The coders* | *List any bugs found in a standardized format* | *No real deadline, and it will be useful in finding and squashing the bug* |

# Roles

|  |  |  |  |
| --- | --- | --- | --- |
| **Role name** | **What are the qualifications of the role? What must a member of your team be able to do to be qualified to operate in the role?** | **What are the responsibilities associated with the role? How can an individual working within this role know that they have done their job?** | **Who will be assigned to each role?** |
| Owner | *Own the company* | *Make sure everything in the company is running properly* | *Owner (Me)* |
| *Custodian* | *Good organizational and cleaning skills* | *Make sure everything is clean and stays stocked* | *Chrissy* |
| *Secretaries* | *Good organization skills, personable* | *Coordinate meetings and agendas, make sure everyone is staying on the same page and has what supplies they need* | *Stan, Sally* |
| *UX Designers* | *Good eye for design. Knows how to make things simple and look good, and stay within the constraints of software development* | *Design the user experience, Work with the testers to make sure everything flows well in the application* | *Ursula, Xavier* |
| *Technical Writer* | *Good grammar, knows how to edit and produce technical documentation* | *Create, edit, and produce all requirement documentation* | *Teri* |
| *Software Engineer Lead* | *Can produce good software, and help others produce good software* | *All regular software engineering responsibilities, as well as managing a division of the team* | *Abe, Britney, Emily* |
| *Software Engineer* | *Must be able to code and design good software. Must be able to work within constraints given by the UX designers* | *Produce good software* | *Claire, Doug, Frank, Grace, Holly, Ingrid, Jack, Keith, Larry* |

# Checkpoints

|  |  |  |
| --- | --- | --- |
| **Checkpoint name?** | **How long do you expect it to take to reach this checkpoint?** | **How will you know that the checkpoint is reached?** |
| *MVP Achieved* | *12 weeks* | *All base functions are working as expected, under the correct conditions* |
| *Product Hardened* | *16 weeks* | *The product will run without issue for all known scenarios* |
| *Product Released* | *18 weeks* | *The product has been released and accepted by the client* |

# Reflection

I think that there are a lot of glaring issues that come with this plan. In the perfect world, this would work great. But as we all know the world isn’t perfect. This plan doesn’t really take into account the fact that people aren’t going to be satisfied with a product that is 90% good when they have the opportunity to make it 91% good. So I could definitely see some issues surrounding that, and that is why I added in a few checkpoints along the way. Without checkpoints, I could really see this going on forever.

I think this could also be resolved through looking at it with a more scientific lens like I mentioned in one of my posts this week, but that can be very difficult for some people to do. Even in the science world people have problems letting go of their theories (which is sometimes really good, and sometimes not so great). I imagine that would be even worse in the software development sphere.

# Citations

B. Boehm, "A Spiral Model of Software Development and Enhancement," IEEE, xxx, pp. 61–72, May 1988,

[Online] Available: http://ieeexplore.ieee.org/xpls/abs\_all.jsp?arnumber=59&tag=1

# Rubric

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Exceptional 10% | Good  90% | Acceptable  70% | Developing  50% | Missing  0% |
| Accuracy  40% | It is completely obvious which development methodology is described. Any knowledgeable person would be able to identify the methodology based on this document. | There is nothing to add and nothing wrong; the development methodology is completely described. One part of the plan may be misclassified as **bold** or *red/italic* | There exists one small problem (factual error or missing component). | There exists one large or multiple small problems (factual errors or missing components). | Large parts of the development methodology are inaccurately described or missing. |
| Application  30% | It is obvious that real thought went into the application (*the red/italic part*) of the plan. | The development methodology is applied to the scenario in an uncontrived way. | Every aspect of the scenario is incorporated into the development methodology. | Large parts of the plan are overly vague, do not appear to be related to the scenario, or do not appear to be related to the development methodology. | No attempt was made to apply the development methodology to the scenario. |
| Reflection  20% | The reflection cuts to the heart of the strengths and weaknesses of the development methodology. | The strengths and weakness of the development methodology are clearly communicated. | One strength and one weakness is mentioned in the reflection. | Little thought or effort was put in the reflection part of the paper. | The reflection part of the paper is missing. |
| Professionalism  10% | The paper is easy to read and ideas are clearly communicated. | Everything is properly cited, there are no grammar or spelling errors, and writing style is "professional." | One instance of a spelling error, grammar error, incomplete citation, overly verbose, poor formatting, or poor writing. | A citation is missing where one is needed (plagiarism alert!). | Gross spelling/grammar errors or other aspects of the writing that make the paper difficult to read. |