02 Ponder: Waterfall 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?** |
| **Weekly Check in** | **Software Engineers, owner** | **Status update on the current work task** | **Keep everyone on the same track** | **Weekly** |
| **System Requirements Gathering** | **Client, Technical Writer, Dev leads** | **Gather requirements from the client** | **Get the requirements of the system** | **On project kick off** |
| **Software Requirements Gathering** | **Technical Writer, Dev Leads** | **Get software requirements out of system requirements** | **Get the requirements of the software** | **Upon completion of the system requirements document** |
| **Software Requirements Acceptance** | **Owner, client** | **Accept the software requirements** | **Review and accept the software requirements** | **At the end of week 4** |
| **Bug Handout** | **Software Engineers** | **Hand out the bugs returned from the testers** | **Hand out the bugs returned from the testers** | **End of each week from 14 onward** |

# 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? |
| **System Requirements Complete** | 2 weeks | *The system requirements document has been accepted by the company and the client* |
| **Software Requirements Complete** | 4 weeks | The software requirements document has been accepted the company and the client |
| **Analysis Phase complete** | 6 weeks | The analysis of the requirements has been completed |
| **Program Design complete** | 8 weeks | The program design has been completed and accepted by the coders |
| **Coding complete** | 12 weeks | The coding of the application has been completed, and handed off to the testers |
| **Testing Complete** | 14 weeks | The application has been fully tested according to the software requirments |
| **Recoding/Retesting** | **19 weeks** | **The software has been deemed to fit the requirements after a few additional rounds of coding and testing** |
| **Software Handoff** | **20 weeks** | **The software has been completed and handed off to the customer** |

# Reflection

I feel that this would be a very good way of producing good software for a one-and-done client. Someone who has a task that they would like automated, or a system they need to store information in and pull information out of. If I were to have financial stake in it though, I would have a pretty hard time spending all that time doing the requirements stuff early on.

It takes valuable development time away, especially if you only have one project rolling at a time. If you were able to get requirements for one while completing another, I think it could be a pretty good process. But, for just a one-time thing the requirements gathering can turn into a pretty big pain and possibly delay the whole project before it even begins.

# Citations

Royce, Winston W, “Managing the Development of Large Software Systems”

Proceeding of the IEEE, TRW, August 1970

# 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. |