05 Ponder: Scrum 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?** |
| **Release Planning** | **Team Leads, Owner, Client** | **Review the software, and create a release** | **Release of Program** | **Beginning of each Planned release/monthly** |
| **Sprint Planning** | **UX Designers, team leads, owner** | **Discuss risk potential for the current objective** | **Determine risk, and product viability** | **Beginning of each week/spring** |
| **Sprint Review** | **Software Engineers** | **Team up and Code** | **Team Up and code** | **End of each week/spring** |
| **Standup** | **Software Engineer teams** | **Review daily progress, daily plans** | **Review Daily progress, daily plans** | **Daily** |

# 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?** |
| **Weekly Requirements** | **The client and owner** | **The software engineers** | **List the requirements for the week** | **It needs to be created each week** |
| **Release Requirements** | **The client and the owner** | **The software engineers** | **List the requirement for the next release** | **It needs to be created at the beginning of each release cyle** |
| **Sprint Plan** | **Each lead/scru master** | **The software engineers** | **List the requirements fo the print** | **It needs to be created at the beginning of each sprint** |
| **Burn Down Chart** | **Scrum Masters** | **The software engineers** | **Show the list of items to complete in the sprint, and how they should be completed in a perfect world** | **It needs to be created at the beginning of each sprint, and updated each day to show progress** |

# 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/Product 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* |
| ***Scrum Master*** | ***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?** |
| **Release Plan Ready** | **A couple of days at the end of each release** | **The next release is fully planned out** |
| **Backlog Ready** | **A couple of days at the end of each release** | **The backlog is fully fleshed out** |
| **PBI Ready** | **A couple of days at the beginning of each release** | **The PBI is clarified, and various tasks identified** |
| **Sprint Ready** | **A couple of days at the beginning of each release** | **The backlog is excepted, burn-down chart created** |
| **PBI Done** | **This is a continuous checkpoint during the sprint** | **This is a continuous checkpoint during the sprint** |
| **Sprint Done** | **At the end of each sprint** | **All testing is done, and accepted. Product owner approves** |
| **Release Done** | **At the end of each release (assuming 3 releases each a month long, 3 months from start date)** | **Criteria is met and product is deployed to production servers** |

# Reflection

It was interesting to learn more about the earlier types of Agile development, but I do see some clear advantages of scrum over XP. I think it has some really good ideas but being client controlled is VERY difficult to achieve in real life. A client doesn’t really have as much of a finger to the pulse of a program and often will ask for more and more features as it goes along, such as “oh, that’s awesome. Can it also do X?” or “I was doing Y the other day, could the program also handle that?”. Due to that nature, unless you were collecting hourly pay and the deadlines were adjusted with each request I could see it causing some real issues.

# Citations

R. Jensen, "A Pair Programming Experience," CrossTalk, vol. 16, no. 3, pp. 22-24, March 2003

[Online] Available: http://static1.1.sqspcdn.com/static/f/702523/9292400/1289016242200/200303-0-Issue.pdf?token=tZDyBy7a9JOdod%2B2GFJ1YEPwVgc%3D

“What Is Extreme Programming (XP)?” Agile Alliance, 24 Sept. 2019, [www.agilealliance.org/glossary/xp/#q=~(infinite~false~filters~(postType~(~'post~'aa\_book~'aa\_event\_session~'aa\_experience\_report~'aa\_glossary~'aa\_research\_paper~'aa\_video)~tags~(~'xp))~searchTerm~'~sort~false~sortDirection~'asc~page~1)](http://www.agilealliance.org/glossary/xp/#q=~(infinite~false~filters~(postType~(~'post~'aa_book~'aa_event_session~'aa_experience_report~'aa_glossary~'aa_research_paper~'aa_video)~tags~(~'xp))~searchTerm~'~sort~false~sortDirection~'asc~page~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. |