Assignment 6

SE329 - Project Planning

Stamatios Morellas

7/2/2019

1. Communication Plan

Team-shared information:

1. Inputs from stakeholders
    a. Message content: *Any relevant feedback that the stakeholders give, in addition to new information*
    b. Delivery method: *Meeting & follow-up email*
    c. Frequency: *When necessary*
    d. Format: *Report*
    e. Owner: *Stakeholder(s)*
    f. Audience: *Project manager, Project team*
2. Ranked list of required features
    a. Message content: *The required features that the project will have, ordered by importance.*
    b. Delivery method: *Shared document*
    c. Frequency: *Single-time (unless features change throughout the project)*
    d. Format: *PDF*
    e. Owner: *Project manager*
    f. Audience: *Project team*
3. Sprint Backlog
    a. Message content: *Selected stories for the sprint*
    b. Delivery method: *Scrum board*
    c. Frequency: *Daily*
    d. Format: *Trello*
    e. Owner: *Project team*
    f. Audience: *Project manager*
4. Sprint end date
    a. Message content: *The end date of the sprint*
    b. Delivery method: *Scrum board*
    c. Frequency: *N/A*
    d. Format: *Date and time*
    e. Owner: *Product owner*

f. Audience: *Project manager, project team*

5. Team deliverables
   a. Message content: *The project deliverables*
   b. Delivery method: *Email*
   c. Frequency: *Monthly*
   d. Format: *PDF*
   e. Owner: *Project manager*
   f. Audience: *Project team*
6. Burndown/up charts
   a. Message content: *Work left / time relationship*
   b. Delivery method: *Email*
   c. Frequency: *Weekly*
   d. Format: *Graphical chart*
   e. Owner: *Scrum master*
   f. Audience: *Project team, Project manager*
7. Daily scrum meeting information
   a. Message content: *A daily discussion of progress in the development process*
   b. Delivery method: *Face-to-face*
   c. Frequency: *Daily*
   d. Format: *Meeting*
   e. Owner: *Scrum master*
   f. Audience: *Project team, project manager, product owner*
8. Sprint review information
   a. Message content: *A meeting that reviews the progress of the code sprint*
   b. Delivery method: *Face-to-face*
   c. Frequency: *1-4 Weeks*
   d. Format: *Meeting*
   e. Owner: *Project manager*
   f. Audience: *Project team*
9. Misunderstandings about the project goals and objectives
   a. Message content: *Any information that a team member is confused about*
   b. Delivery method: *Face-to-face or email*
   c. Frequency: *As soon as there is a misunderstanding*

    d. Format: *Meeting and/or electronic message*
    e. Owner: *N/A*
    f. Audience: *N/A*

10. Conflicts between team members
    a. Message content: *Issues between team members that may affect the progression of the project*
    b. Delivery method: *Face-to-face*
    c. Frequency: *Whenever necessary*
    d. Format: *Meeting*
    e. Owner: *Team member(s)*
    f. Audience: *Other team members, project manager*
11. Missed deadlines
    a. Message content: *Any important milestones that are late*
    b. Delivery method: *Email*
    c. Frequency: *When necessary*
    d. Format: *Report*
    e. Owner: Project team
    f. Audience: *Project manager, project owner*

<u>2. Quality Management Plan</u>

<u>Quality requirements</u>:

The following steps need to be taken to ensure proper quality management on the project:

1. Set QA requirements
2. Set rules for addressing defects
3. Set a process of evaluations and reviews
4. Decide on the tools to use in the project
5. Identify criteria for customer acceptance

There are 4 generic quality attribute requirements:

- Performance
- Scalability
- Security
- Usability

Some common defects that might occur are:

- Code bugs - *Low risk*
- Compliance with regulations - *High risk*
- Compliance with requirements - *Medium risk*

Code bugs are probably the most commonly encountered software defects, but they can be identified and solved with testing and reworking of the code. The severity will depend on the number of bugs that exist per 1000 lines of code or so. Severity will also depend on the impact they have on the overall performance, scalability, security, and usability of the software.

<u>Tracking tools</u>:

One way to manage reported defects are by having face-to-face meetings with the project manager and team members, and having a discussion. Another efficient way to do this would be through a *CI/CD (automated testing) pipeline*, which can help identify defects more precisely in the project, as well as alert customers and developers of existing issues. This will serve the purpose of revealing what defects have been resolved and which ones still exist within the project.

<u>Code review criteria</u>:

The following steps should be taken to ensure the establishment of an effective code review system:

- Set expectations for annotating and formatting code
- Integrate a system to capture metrics

- Use peer-reviewed documents to make sure everyone in the team is aligned on the objectives of the project
- Verify that existing defects are solved, instead of only identified
- Review fewer than 400 lines of code at a time (practice lightweight code review)
- Use checklists
- Use a code review tool (optional)