Taylor Shearin, Kevin Carmichael

CSCI 3350-201

Pine

10/31/2014

Software Management Plan

* Product Functionality

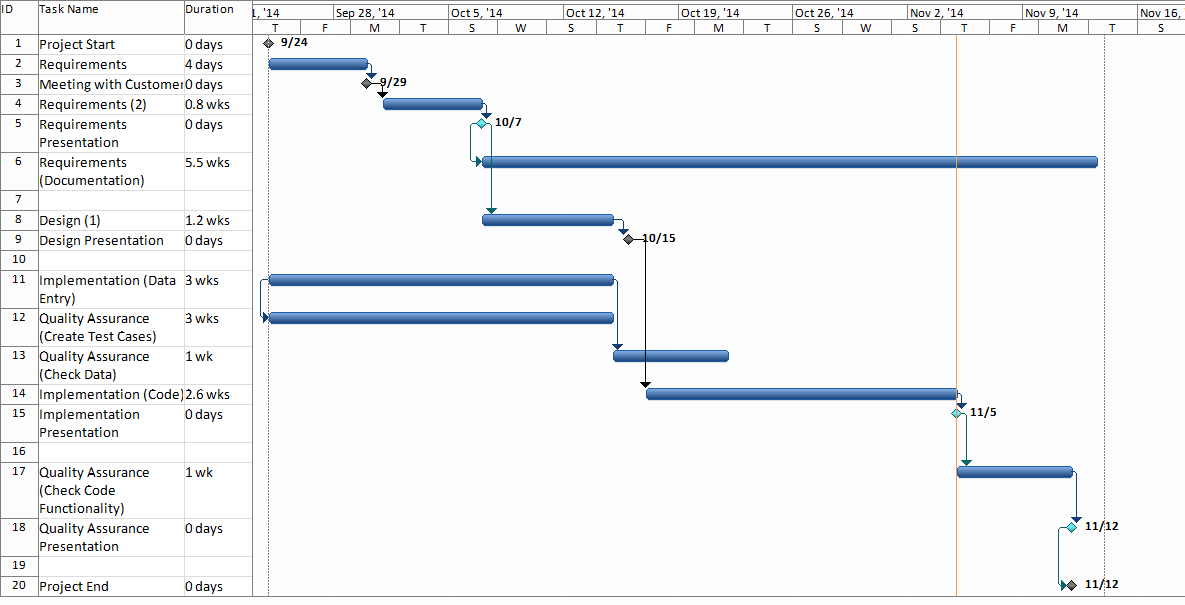
WoodWorks is an easy-to-use, robust woodworking guide and for Windows 7 users. WoodWorks is designed to be intuitive while providing full functionality when calculating various specifications of a given type of wood for use in projects. WoodWorks will display or utilize various properties of a given tree species, including a brief description, the specific gravity, the moisture content of a given piece of wood at certain densities, as well as coefficients of dimensional change. Using this information, WoodWorks will calculate dimensional changes and beam deflections to a user’s given values.

* High-Level Functionality

WoodWorks is intended to replace a cumbersome and obtuse book of documentation. The application will feature an intuitive user interface, allowing a user to perform a required task without the need to refer to any sort of documentation. Examples of user controls include combo boxes, text boxes, and select boxes, as well as labels providing formulas and pictures, displaying the types of trees displayed.

* Project Staffing
  + Requirements
    - Team Leader – Sherrie Dowdy
      * Jessica Horne
      * Saiping Chen
  + Design
    - Team Leader – Jonathan Barnett
      * Bradley Cross
      * Sergey Chibizov
  + Implementation
    - Team Leader - Josh Houston
      * Nicolas McMahon
      * Eric Blumenstoch
      * Tianran Hao
  + Quality Assurance
    - Team Leader – Tim Denton
      * Kristen Galloway
      * Kevin Patel
* Software Process

WoodWorks will be using the agile development methodology, specifically Rapid Protyping. The user interface is one of the most important pieces to this project, so getting a clickable interface in the customer's hands is one of our top priorities. Functionality will be built off of that initial discussion with the user. Each step will be moniroted by the Quality Assurance team, assuring the correct product is being built for the customer.



Schedule

* Metrics
  + Moisture-content percentage
    - This will most-likely be designed with minimal user input, allowing a user to simply select a value.
  + Wood Dimensions
    - These are decimal numbers that will be accepted through a user interface control.
* Project Risks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk** | **Current Position** | **Last Week’s Position** | **Weeks on the List** | **Resolution Progress** |
| Lack of domain knowledge on the part of the development team | 1 | N/A | 1 |  |
| Communication and organization between teams | 2 | N/A | 1 |  |
| Team availability constraints due to other projects | 3 | N/A | 1 |  |
| Requirement changes in later iterations requiring the reworking of previous iterations. | 4 | N/A | 1 |  |
| Mismatch between team skills and project needs | 5 | N/A | 1 |  |
| Scope Creep | 6 | N/A | 1 |  |
| Development team sandbagging | 7 | N/A | 1 |  |
| Experience with agile / rapid prototyping | 8 | N/A | 1 |  |
| Experience with development tools and implementation language | 9 | N/A | 1 |  |
| Unrealistic budget | 10 | N/A | 1 |  |

1. Lack of domain knowledge on the part of the development team

The software development team has no direct knowledge of woodworking, the product of woodworking, or the materials used in woodworking. Because of this lack of knowledge it will be very easy to misunderstand what the client states they need. Of equal or greater importance is the inability to differentiate between what the client states as a need and what they actually do need.

1. Communication and organization between teams

Due to the rapid prototyping method and multiple teams working in parallel even small issues in communication and organization between teams has the potential to delay the entire process.

1. Team availability constraints due to other projects

Because everyone on the team is assigned to multiple unrelated projects the potential exists for a single team to be overscheduled during key milestones on other projects (Classes). A delay in one stage of the development cycle would cascade to the next stage, either delaying the project or requiring another team to make up the lost time.

1. Requirement changes in later iterations requiring the reworking of previous iterations

Without at least a high level view of later stages of the project it is possible that additional features will require repeating previous work to facilitate the implementation of new features. This will increase the number of man-hours required to implement new features as well as requiring re-testing of previously tested features.

1. Mismatch between team skills and project needs

A skills necessary for project completion are concentrated in a minority of team members. This creates the potential for that minority to be responsible for the majority of the work. Should that occur it would result in poor team morale as well as creating a situation where a single team member could cause the project to fail.

1. Scope Creep

Agreeing to an ongoing software development project without knowing the final project goal, a rough final feature set, or the number of iterations that will be necessary for project completion makes it impossible to contain scope creep.

1. Development team sandbagging

Embarking on an ongoing software development project without a final project goal or defined final feature set may create a situation where the team believes that the only reward for completing tasks is the assignment of additional tasks. In that situation it is entirely possible that teams or team members may intentionally underperform in order to limit the amount of work that can be completed before the project deadline.

1. Experience with agile / rapid prototyping

The inexperience of the development team with the selected methods may lead to confusion and/or delays as the project moves from one stage to the next. This could negate any advantages gained from the use of Rapid Prototyping

1. Experience with development tools and implementation language

The majority of team members have limited or no experience with the implementation language, and only slightly more experience with the development tools to be used. Because of this it will be more difficult to properly design, implement, and test the project.

1. Unrealistic Budget

The existing budget may not be sufficient to fund completion of all undocumented features in future iterations. Without the well-defined and measurable goals on which to base a budget, either in dollars or man-hours, there is no possibility the budget will be realistic.

* Deliverables
  + Several iterations of the User Interface. This will be updated until the customer is happy with what is presented on-screen.
  + Some method of storing information stored in the book (values of various wood properties)
  + Fully-functional solution
  + Readme documentation
  + Project Presentation