John Hardy

3 February 2019

Prof. Johnson

Homework 1

1.1) The basic tasks that must be handled in order are: requirements gathering, high-level design, low-level design, development, testing, deployment, maintenance, and wrap-up.

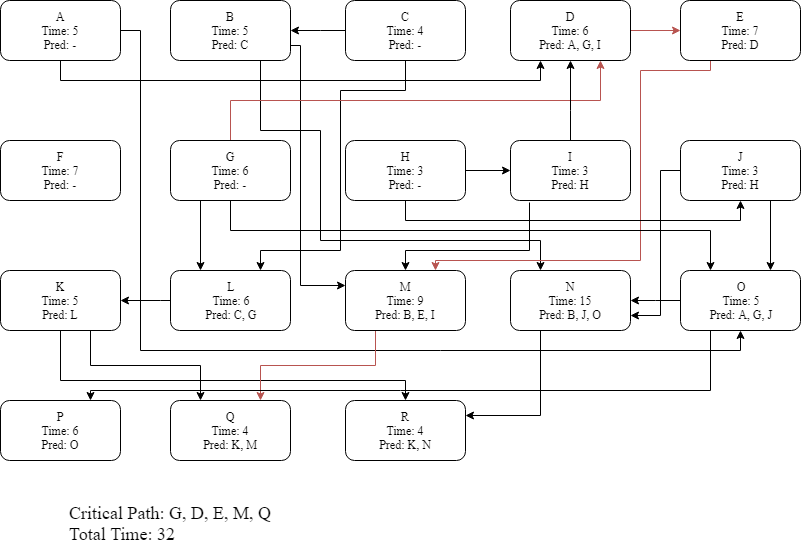
1.2) Each of these steps includes:

* Requirements gathering: This is a plan that tells the team what needs to be done and how long they have to do it.
* High-level design: This phase of the project focusses on big picture decisions such as which platform to use, how data should be stored, and any external interfaces (i.e. e-mail messages or text messages). This is then broken down for each sub-team to work on separately.
* Low-level design: This part includes individual descriptions of how each small piece of the project should work and execute without going into great detail.
* Development: Now the teams begin to get to work on each of the items planned out, continuing to refine the details until actual code can be generated.
* Testing: Once a chunk of code has been written and is ready to be incorporated, it is first tested by outside personnel who did not write the code themselves to check for bugs. Afterwards, the code is then incorporated into the rest of the project which is tested as well.
* Deployment: When all the tests pass, the code is packaged and deployed to the end user.
* Maintenance: Once the user uses the code, they will find bugs that will need to be fixed, this is where the software developer comes in and pushes out patches to the code in order to fix these issues.
* Wrap-up: This is the post-mortem of the project where you reminisce on what went well, what went wrong, and how you can prevent problems in the future.

2.4)https://docs.google.com/document/d/1RDNdRHuzC9RA3bGZkgsSJODfBlozvx8Y5RqhKQIhaYo/edit?usp=sharing

2.5) JBGE stands for “just barely good enough” and it has to do with how you should maintain code documentation. You don’t want to go into excessive detail about how your code works because then you will have to update the documentation a great deal whenever you change the code.

3.2)



3.4) 



3.6) These sorts of issues can be sorted out by extending the time allocated for all task by 5% and by explicitly adding in time for sick leave and other somewhat predictable events.

3.8) The biggest mistakes you can make while tracking a slightly behind task are:

1. Ignoring the task and expecting the developer to catch up later on. Often developers underestimate the task at hand and they need more time than is allocated initially.
2. Piling on more developers to speed up the completion. Adding more developer does not help since it takes a significant amount of time to get them up to speed on the task at hand, only slowing things down even more.

4.1) The five characteristics of good requirements are: clear, unambiguous, consistent, prioritized, and verifiable.

4.3) The categorized requirements are as follows:

* a. Allow users to monitor uploads/downloads while away from the office. – User Req.
* b. Let the user specify website log-in parameters such as an Internet address, a port, a username, and a password. – User Req.
* c. Let the user specify upload/download parameters such a number of retries if there's a problem. – User Req.
* d. Let the user select an Internet location, a local file, and a time to perform the upload/download. – User Req.
* e. Let the user schedule uploads/downloads at any time. – User Req.
* f. Allow uploads/downloads to run at any time.—Functional Req.
* g. Make uploads/downloads transfer at least 8 Mbps. —Non Functional Req.
* h. Run uploads/downloads sequentially. Two cannot run at the same time. —Non Functional Req.
* i. If an upload/download is scheduled for a time when another is in progress, it waits until the other one finishes. —Non Functional Req.
* j. Perform schedule uploads/downloads. —Functional Req.
* k. Keep a log of all attempted uploads/downloads and whether the succeeded. —Functional Req.
* l. Let the user empty the log. —User Req.
* m. Display reports of upload/download attempts. —Functional Req.
* n. Let the user view the log reports on a remote device such as a phone. —User Req.
* o. Send an e-mail to an administrator if an upload/download fails more than its maximum retry number of times. —User Req.
* p. Send a text message to an administrator if an upload/download fails more than its maximum retry number of times. —User Req.

The only requirements that are missing are business requirements and implementation requirements. Business requirements describe improvements in the system after it has been implemented (i.e. “Time spent on version control decreased by 50%”). Implementation requirements are temporary and will not exist after the system has been implemented (i.e. “Data transfer software migrates data from old data-bases to new one”).

4.9) Possible changes to hangman game (sorted using MOSCOW):

1. Increase the vocabulary available to the game.--Must
2. Added keyboard functionality on top of mouse click events.--Should
3. Include a difficulty setting to make the words longer or shorter.--Should
4. Add information about the category of the word to assist the user.--Could
5. Make Mr. Bones into a 3D asset that can be animated.—Won’t