

Matt here are my notes on the revisions to this document. - Jake

Yes, indeed Jake! Some suggestions: 1. Schedule. Define development releases for your project that incorporate a subset of the functionality for your project. These releases would constitute milestones for your project. THEN, couple testing of your project to these milestones to determine the testing schedule. In the first part of 481 we are going to revisit the roadmap and gantt charts for this very reason.

Alex did this part and I introduced it to the document. Most of it looked good to me but I didn't think too hard about it.

2. Regression testing. I didn't see too much in your test plan on how you will do regression testing. If you add new functionality into the next release, you often break features that tested as ok in the previous release, thereby requiring us to retest every new release for the functionality that should already be present. If we assume that the features that tested ok before are not broken we can optimize our time to focus on the newly added features; however this can be problematic, because the new features may have broken something. It happens all the time. Hence we use regression testing to ensure previously tested features/functionality are STILL functional. So repeating tests at each release is critical. In your next test plan, please be sure to address regression testing. As you add the risks/mitigation section, I would urge you to use the criticality of tests to be completed to determine what you might have to do to make up for potential development schedule slips. It's great you have that in there. We'll talk more about that in 481 as well. Good job guys.

Added to page 15 in section 4.3.4. I thought regression testing mean running some sort of statistical analysis, so I was confused for a while. I think don't think this will change the project much but it's in there now. (we were planning on doing this anyways I guess now it's written down).

Additionally, this revision of your spec should incorporate significantly more detail in your testing sections. You must have all sections defined sections with descriptions. You must have diagrams that show the composition of your design and how they will be evaluated. Your test plan must clearly articulate how you will do unit tests (software or hardware or both) as well as system testing. Your test descriptions must be clearly written so as to be understood by an external reviewer. Your test procedures must be updated such that they can be understood and executed by a test engineer. Finally, your test plan schedule should be updated to be consistent with your schedule and development release milestones.

I think I mostly hit this part in prior revisions. I couldn't think of much to add. I plan on breaking up the tables more later but I needed to get to other work instead so I left that for now. The system's testing table needs to be split up into separate tables. Personally I don't like the table format. I find it ugly. I know it keeps you from having to read it in text but I feel like there's so kind of other solution for this that meets the criteria half way that isn't so ugly.