INTRODUCTION: Project Scheduling Checklist

The checklist starts on the following page

What This Is

A checklist to use during the planning and scheduling phase of your project. Contains items to ensure your schedule includes all project work, such as cross-functional activities, testing, and more.

Why It's Useful

A project consists of more than the main technical or development or design work. Many activities from supporting groups must also be included, for the product or service to be truly ready for customers at the end of the project—and for your team to be committing to an accurate schedule based on everything that *really* has to be done before the project is over!

This checklist will help you make sure your project schedule includes everything it should. And before you've even gotten to detailed scheduling, it can remind you of items that should be included in the project scope and investigated during the planning phase.

How to Use It

1. Take a first look at the checklist early in the project, before you've even started detailed planning, to get reminders of items to consider for defining the scope of the project.
2. Read this checklist as you're starting your planning work, to be aware of the entire range of activities you should consider for your schedule.
3. Provide copies to your cross-functional team members as you ask them to contribute their own task lists and estimates to the full project schedule.
4. Reference the checklist during the planning phase as necessary to make sure you're getting all the needed details to have a full schedule.
5. Then be sure to check your schedule against it again as you near the end of planning. Make sure all items are covered so that your team commits to a full and accurate schedule!

The checklist includes both items that apply to any project, plus a number of items that are specific to technology development projects. You can customize this template to add items specific to your company and industry.

The checklist starts on the following page.

Project Scheduling Checklist

STOP! Did you read the minutes from past projects' Lessons Learned meetings to make sure previous problems aren't repeated and good ideas are used?

Early planning for getting out of the Investigation/Planning Phase:

[ ] Have you set a target date for getting out of this phase?

[ ] Do you have a schedule or task list/action item list to move the team through this phase?

[ ] Have you scheduled any training that should be done early to not impact the schedule later? (Training on tools, third party software, the development process, etc.)

Detailed schedule items and durations

[ ] Were the architecture and high-level hardware or software designs done to module level, sufficiently to:

__ accurately identify all the pieces of technical work to be done?

__ get an accurate estimate for each?

(Did the team agree up front on what level of detail in the architecture/design would be necessary to accomplish the above?)

[ ] For more uncertain tasks, did someone assess the difficulty or uncertainty and assign fudge factors to the task estimates?

[ ] Were interfaces defined well enough during this phase to find all pieces of technical work?

[ ] Has the team defined the desired content of any detailed specifications, to understand the scope of work and reviews, and ensure an efficient design process?

[ ] Are there back-up plans for risky areas, and trigger points for deciding to go to a back-up plan?

[ ] Are important design reviews shown as key milestones?

[ ] Is time included for design reviews and code reviews?

[ ] Is time included for teambuilding activities and celebrations later in the project? (Plan ahead for them so people will have time to go!)

[ ] Are "ancillary" tasks such as team member training scheduled?

[ ] Is time scheduled to train contractors on your development process and how it's being used on this project?

[ ] Is time scheduled to train any contractors, if they are entering project mid-stream?

[ ] Is time included for unit testing? (Note: ensure no untried code goes to any independent QA group)

[ ] Is time included for creation of test tools or testing environments, including automated tools and scripts (unit, integration, system functional, drivers, harnesses, instrumentation code)?

[ ] Is time included for writing and testing of SQA scripts by SQA members, and for participation and/or review by developers if needed?

[ ] Do alpha and beta test periods include time for deployment in field before testing starts?

[ ] Is there time for adequate system regression test during SQA, alpha testing, and beta periods?

(continued on next page)

Project Scheduling Checklist (continued)

Detailed schedule items and durations (continued)

[ ] Were testing schedules (integration, SQA, alpha, beta) constructed and estimated from the bottom up, instead of a high-level guess and/or one monolithic testing task?

(For example, use such assumptions as rough number of test cases to execute, number of testers, amount of retest once bugs are fixed, regression test time, etc.)

[ ] Was enough deployment planning done to show up the need for creating any special tools as part of the project, and are those tools in the schedule?

[ ] Was enough manufacturing planning done to show up the need for any special tools, and are those tools in the schedule?

[ ] Does the production phase schedule include realistic detail and resources for the work required to take the particular hardware product to volume manufacture, both pre and post-release?

Cross-functional tasks included

[ ] Have other group's activities have been thought through and put in the schedule?

__ User manuals and other publications

__ Operations (Test engineering, Mfg engineering, Purchasing, Manufacturing, etc.)

__ Field support

__ Marketing/Product Management

[ ] Did the Regulatory group participate in early design reviews and planning?

[ ] Does the schedule have the same level of detail as the Development parts of schedule?

Schedule construction:

[ ] No tasks larger than three weeks long? (Assumes a 9-12 month schedule.)

[ ] Are there milestones every 2-4 weeks for tracking? (Assumes a 9-12 month schedule.)

[ ] Is our development process being used appropriately for the project?

[ ] Have risk assessments been done and iterations included for risky areas? (For example, assume more than one round of design and test on a hardware or software module, algorithm, etc.)

[ ] Are iterations included for multiple design/document drafts and reviews as appropriate? (For