Managing your capstone project is one of the most important and often overlooked aspects of the development process. You will find that effort you put into planning, scheduling, and organizing your project will pay off in the end. Here is an outline of some proven techniques for planning, scheduling, and completing your project.

A project has five phases. Here's a brief summary of each:

Initiate
Articulate your vision for the project, establish goals, assemble your team, and define expectations and the scope of your project.

Plan
Refine the scope, identify specific tasks and activities to be completed, and develop a schedule and budget.

Execute
Accomplish your goals by leading your team, solving problems, and building your project.

Control
Monitor changes to the project, make corrections, adjust your schedule to respond to problems, or adjust your expectations and goals.

Deliver
Deliver/demonstrate your project to your audience (i.e. instructor and classmates), acknowledge results, and assess its success. Take the time to compose a written evaluation of the project and the development effort, in the form of a project report and a presentation.

The middle three phases are not sequential. You will find that you are constantly planning, executing, and controlling your project as necessary.

Aren't these phases really just common sense? In many ways, yes. But keep in mind that project development can be an unpredictable process. *Many* projects are delivered unfinished and incomplete, or are substantially over budget. The more effort you put into managing your project, the more you increase your chances of success.

These five phases of project management are explained in more detail below:

Initiate
Define your goals in writing as much as possible, so they make sense to everyone involved. You may think that your goals are obvious, but write them down anyway! Consider the following criteria for goals. They should:

- be agreed upon by everyone involved
- be realistic
- be specific
- be measurable (can you translate it to a calendar or Gantt chart? More on this later.)
- have a time component or deadline
- have a clearly defined set of responsibilities (can you assign this goal to someone, or a team?)

To identify your goals, first write down everything you can think of that might be a goal for your project. At this point, you are brainstorming; don't worry about including unrealistic or unlikely goals, or including ideas that don't meet the criteria above. But don't confuse goals with features.

Second, cross out anything on your list that has no direct bearing on your project.

Third, eliminate anything that is a step in achieving an end goal. You are trying to identify desired results, not the process itself.

Finally, does everything left on the list meet the six criteria above? Are they really what you want to achieve? Are they realistic?

Once you have a written list of clearly defined, realistic goals, make sure everyone on your team as well as the instructor, has a copy. This simple step goes a long way toward keeping people focused, motivated, and committed to your project.

Plan

Translate your goals into specific tasks and create a plan to complete those tasks. Unlike goals, tasks are identifiable steps that can be sequenced on a timeline or calendar and assigned to specific people.

Tasks should:

- be unambiguous. A task should be clear and simply stated. If a task cannot be described in a sentence or two, you might want to break it into two or more smaller tasks.
- be confined to a known time frame without gaps for other tasks in between
- include only work-related aspects
- be associated with a specific person or group of team members
- have a single point of sign-off (one person should be responsible for deciding when the task is done)

Don't worry about putting tasks in sequence yet. What is important is to identify what specifically needs to be done to build your project. Typical tasks in project design and development might include:

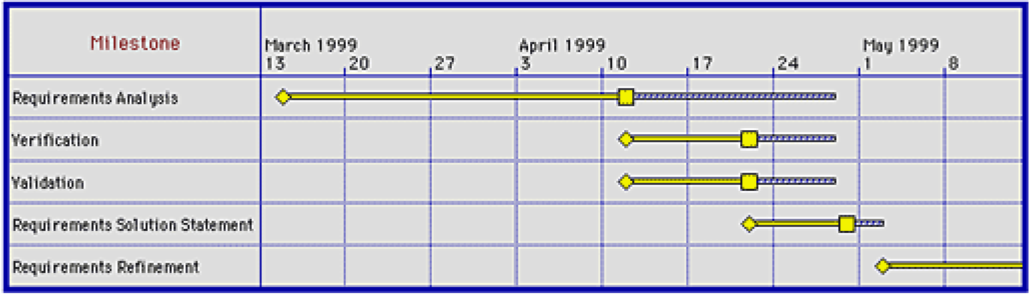
- build the mechanical device
- build the interface circuits for sensors and motors
- write C programs to test the sensors, motors and interface circuits
- design control algorithm by using UML activity diagrams
- write C program to implement the control algorithm
- implement network interface
- test and debug the complete project

Once you have a list of tasks, you can start making a plan to complete them. Many tasks are independent of one another (such as "build the mechanical device" and "write C programs to test sensors and motors"), but many are not. Organize the tasks in a logical sequence (though don't try to create a schedule just yet). Identify *who* will be responsible for each task and *what* resources they will need (components, test instruments, computer software, etc.). Pay attention to tasks that may depend on other tasks.

When you have a sequence, you can begin to translate it into a schedule. Try to identify milestones, sets of tasks that represent meaningful achievements. For example, "all sensors tested" or "mechanical assembly completed" or "all code written and tested" are good milestones for project development. When scheduling your project, you should assign specific dates by which these milestones will be completed.

You should use a tool like MS Excel or Project to help you plan. Simple projects can be planned easily with a calendar. Bar charts called Gantt charts are quite useful in showing how multiple tasks overlap or relate to each other. They also help you allocate workers' time effectively. It's wise to schedule non-critical tasks as early or as late in the schedule as possible.

Sample Gantt chart



You might want to create an "ideal" schedule and a "real" schedule that you constantly update as the project progresses. You must communicate changes in the schedule to everyone on the project promptly and regularly. Adjust the schedule whenever you change the definition or time frame of tasks, when the project scope changes, or when the project deadline changes.

So far, all you have is a pile of paper. You might feel like you haven't even *begun* the "real work" on your project. This is absolutely not true; you now have a carefully thought-out plan and schedule for the project that identifies goals, tasks, and milestones.

As a final step, set up an e-mail list or simple project web page where you can post updates, announcements, or other information. However, do **not** rely on e-mail for important communication, though it's fine for reminders or setting up meetings.

Execute

It may sound obvious, but start the project on time. If you start off late, you may not be able to catch up. Start formally, with an initial meeting of everyone involved where you discuss the work plan. Plan your next meeting at the kickoff meeting. It is very important to establish a regular meeting plan. Meetings go a long way to foster commitment to the project and respect for deadlines.

During the execution phase, everyone on the team will probably be off building mechanical device, interface circuits or writing code. As a project manager, you must stay in touch with your team members. This works both ways: others will need to stay in touch with you. You will also likely be the "content expert" for the project, so now is not the time for a research trip to Europe. Most people will expect you to check your e-mail at least daily, and to be available by phone.

If you are responsible for some of the tasks on the schedule, make sure you do them. Your contribution will be vital; if you don't provide your part of the project to the others on your team on time, they will not be able to complete (or perhaps even begin) their tasks. This will be especially important in the time between the project kickoff and the first meeting, for it sets the tone for the entire project.

You should require that everyone on the project send you regular status updates (and you should send updates to the team). Status updates will help you maintain and update the schedule, and coordinate interdependent tasks.

Control

No matter how much planning you do, you will need to be ready for change. Project development rarely goes as planned, so don't get discouraged if you get off schedule. Executing and controlling are two sides of the same coin.

Most important, don't lose touch with your team members during your project. This is all too easy to do when both you and they may have significant other responsibilities. It's crucial that you continue communicating with your team members and accommodate the changes. Make sure that your goals can accommodate change as well. It's better to step back and simplify your plan rather than jeopardize the whole project for the sake of one aspect of it.

A common problem is the "90 percent done" problem. Most of your project is in place, but those last few steps seem like they're taking forever to complete. Maybe the most difficult task was left for last, or enthusiasm has waned, or a key person has left the project. Try to identify exactly what remains to accomplish and come up with a detailed plan for doing it, but don't insist that people solve difficult problems or overcome creative blocks according to a schedule.

A project left unfinished can be an unpleasant burden; if you can't meet the deadline you've set, consider changing it or adjusting the scope of the project.

Deliver

Allow enough time to document your project as extensively as you can at the end of it. This is not only for your benefit, but for others in your class and the department. Be honest in your assessment of your project and those involved, including yourself.

- What goals were met and which weren't?
- Did you maintain effective communication among team members during the project?
- Were there specific technical hurdles that were overcome or not?
- Were there budget issues?
- Who did you solicit help or advice from, and was it useful?
- What advice would you give to others embarking on a similar project?

You should include a copy of your initial goals, task lists, and schedule in your final report. A before-and-after look at the project will be invaluable to you and your colleagues in future endeavors.