

# CIRCUIT CELLAR®

THE MAGAZINE FOR COMPUTER APPLICATIONS

## LESSONS FROM THE TRENCHES

George Martin

### Timing is Everything

If you've ever wanted to choke the person who made up a project design schedule, or if you are the one responsible for making the schedules, you might want to listen to what George has to say about the importance and benefits of making a realistic, achievable, and practical plan for bringing your design to completion.



or the past year, my articles have covered detailed issues in both hardware and software designs. These topics come from my direct experiences, which means I wrestled with those designs until I got them right. I'd like to shift gears now and present a topic I've seen many engineers struggle with—scheduling a project.

As a designer, most of your time is probably spent doing design work. To me, that's the fun part. But, there are other aspects of your job that you need to do well so that customers can be satisfied. For example, you may need to create a bill of materials so the purchasing department can procure the parts or you may need to work with the test department to create a factory test setup.

And usually, to start a design, someone has to make a schedule and a budget for management to approve the project. Someday, you're going to be the person scratching his/her head over when the prototype will be ready and how much it's going to cost.

I believe the best schedule and bud-

get you produce is one with no surprises. Sure, it would be easy to estimate that a project will take twice as long and cost twice as much as you think, but it's a competitive world. Your competitors aren't going to be that foolish. So, you've got to produce realistic schedules and budgets.

Scheduling and budgeting go hand in hand. For this article, I'll focus on scheduling and bring up budgeting where it makes sense.

#### NEAR-MISS SITUATIONS

What is a realistic schedule? You hear about a 90% confidence schedule, which means you're 90% confident you'll make the schedule. To my way of thinking, it also implies that you miss one out of every ten. Although that might be a good approach in a larger company with more flexible staffing, my business can't live with that. So, I approach scheduling with a more conservative view.

The basic building block of any schedule is a list of tasks to be completed. And, this is absolutely the most important thing to remember: you must create a list of all the tasks in your project. I can't stress this enough. If you leave out or forget anything, you've blown it! On your first attempt at scheduling, you'll probably leave something out. But don't worry, it's usually so painful that you never forget it again—at least, not that item.

#### GETTING THE MICRO VIEW

How do you ever come up with the number of hours needed to complete a design? Essentially, each task needs an estimate of how many hours it will take. My way is to break the larger tasks into smaller ones.

For instance, it does no good to estimate four weeks to design and test something like an RS-232 command processor. You need to break a project

like that into smaller components. For a command processor, I would break it down as follows:

- define the commands and their actions
- receive characters (interrupt routines)
- process characters (main processing routine)
- design processing routine for each command type

Depending on how many commands a project takes and how complicated each command is, I bet you can divide this project into tasks that take between four hours to two or three days each. When you total all these hours, you have something meaningful. You can discuss, defend, and measure your progress.

Another benefit of defining your tasks to this extent is the clear visibility you have of your project when you run into problems. When I get stuck, I just switch to the next most important task. Efficient progress takes place when you constantly work on the most important task. Breaking your design work into small tasks is difficult at first. But, after one or two projects, you begin to see patterns and the breakdown becomes easy.

In a nutshell: breaking a project into smaller tasks lets you see the real amount of work you have to do.

## MICROTASKING MANAGES PENNIES

As you define tasks, you should think about the costs associated with each task. For example, if purchasing printed circuit boards is part of your project, then you need to budget for the board costs. Divide the board costs into separate components, such as plotting, tooling, manufacturing, testing, and shipping. This level of detail helps you see the real cost of overnight deliveries and how such choices will have an impact on project costs.

## DECIDING WHAT'S BIG

Next, create milestones for your project. It's critical that milestones—points in a project schedule where everyone can agree on the state of the project—are verifiable. A milestone titled "completion of paper design" can

be verified by publishing the schematics. Milestones like "all parts received" or "breadboards built" are also easily verifiable.

Watch out for milestones like "design review" or "calibration software written." I don't believe they tell you anything. A design review may uncover a design or requirement flaw that takes days to fix. You may believe you've completed the design review, but essentially your schedule has just developed a loop. The same problem can occur with calibration software. Calibration software needs to be written, tested, and approved. Although the actual writing of the software is a task, testing and approval of the software make better milestones.

It's easy to fool yourself and others about a project being on schedule by selecting inappropriate milestones.

## A RELATIVE MEASURING STICK

I've included an actual schedule for you to download. The first thing to notice is that it's relative, which means the timeline is measured in weeks after the start of the project instead of actual calendar dates. If the start of the project is delayed, completion is also delayed.

When you're establishing the schedule, don't forget: the work we do is difficult. If management or sales needs to get something approved before you start, then they're part of the schedule. If they come in late with their part, they directly affect the start (and end) of the project. Although this is a small point, I think you get more respect by presenting a schedule in this manner and not trying to be the hero who absorbs all the late starts by working day and night.

Take a look at the milestones in my schedule. Do you have any questions about what they are, what they mean, and how you would verify each of them? No, I don't either.

## ESTABLISHING TASK DEPENDENCIES

To develop your schedule, you need to create task dependencies. With classic scheduling, every task depends on either a milestone or another task before it can be started. And, every task must be completed before a milestone is reached or another task can start. You

can get a number of books and programs to help you with this part. For large projects, such programs are indispensable. If you have several small projects that turn into a large project when you combine them, these programs are valuable.

What if you estimate that a task takes 40 hours to complete? Does that 40 hours represent five days at eight hours per day? What about other meetings you need to attend or what if production support keeps messing up? What about the extra time you work? Perhaps your 40 hours should represent four days. I can't define these parameters for you, but when I had a staff of engineers, I scheduled everyone at 80% or 32 hours per week of actual time. This seemed to work for me, but you need to adopt an approach for your own situation.

In my actual project, I have about 32 individual tasks. Not enough for a scheduling program. Instead, I just used a word processor with a fixed-pitch character font. Let's look closer at Figure 1 to see my first few tasks and milestones.

I overlapped establishing project requirements with completing a detailed design, which could be a recipe for disaster. It would be a more rational approach to complete the requirements then do a detailed design. But, in this case, my customer was specific and certain of his requirements. He never changed them through the project and, when I needed clarification, I got a prompt, clear, and consistent answer.

I put a milestone after the paper design review, which is something I said earlier was perhaps not such a good choice for a milestone. But, I needed to shorten the overall schedule. By making the paper design review a milestone, I could justify starting chassis design at that point. If I delayed the chassis design, I would have delayed the project. To make sure nothing slipped through, I involved the customer throughout the paper design task. I presented portions of the design as they were completed and had the customer freeze the design after review and start the chassis work as the paperwork caught up.

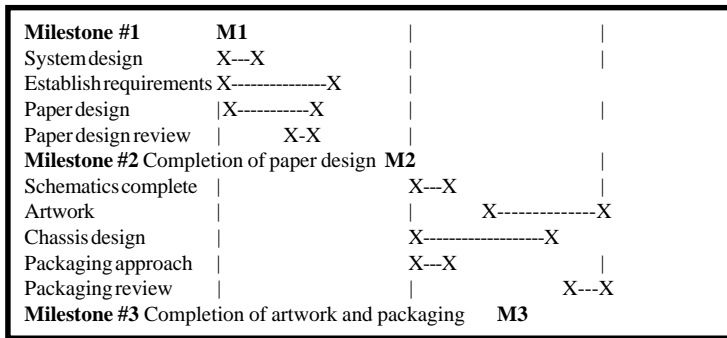


Figure 1—A closeup of my schedule gives you a chance to see how specific tasks overlap with each other.

## DEFINING CRITICAL PATH

The critical path involves the set of tasks that directly affect the duration of the project. A change in any critical-path tasks changes completion of the project. If you use a word processor for scheduling, tasks are not automatically linked to indicate which ones are on the critical path. After you identify critical-path tasks, you can start working to shorten the schedule.

Looking again at the complete downloaded schedule, notice that I put in three weeks of contingency after all the testing. Is this a cop-out? I don't think so. The first 16 weeks of this high-pressure intense project has no room for error. It's not a time to be cautious. The customer needs products as soon as possible.

So, just as taking extra money on vacation or leaving early with sufficient time to make a meeting are good practices, inserting a contingency period makes sense. If the project is on schedule with no surprises, then the contingency can be removed and the balance of the schedule moved up. Planning a contingency period also lets you keep some sanity in your personal life.

## AN EXACT FITTING

I've left out several areas that are specific to the work I do and how I do it. For example, I purchase parts on the Internet using a credit card (i.e., no formal purchase orders). You may have a purchasing department to support you.

Tailor your tasks and schedules to your situation. After several projects, your skills will be finely honed and you might even enjoy creating and maintaining schedules.

*George started his career in the aerospace industry in 1969. After five years at a real job, he set out on his own and cofounded a design and manufacturing firm. Typical systems that George designs include servo-motion control, graphical input and output, data acquisition, and remote control. George is a charter member of the Ciarcia Design Works Team and most recently, he's been working on the people-tracking system for Bill Gates's new house. You can reach him at [george.martin@worldnet.att.net](mailto:george.martin@worldnet.att.net).*

**Circuit Cellar, the Magazine for Computer Applications.** Reprinted by permission. For subscription information, call (860) 875-2199 or [subscribe@circuitcellar.com](mailto:subscribe@circuitcellar.com).