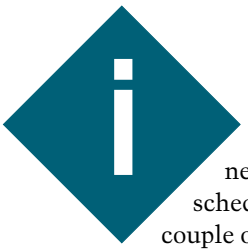
CIRCUIT CELLAR ONLINE

LESSONS FROM THE TRENCHES

George Martin

Scheduling Revisted

This month, George addresses the wide response he received for his past scheduling article. How can you schedule for something you've never done before? Well, keeping records is a good start. And, asking around to gather information isn't a bad idea. Even if it comes down to taking a shot in the dark, George gives us some tips about breaking the schedule down.



certainly hit a nerve with the scheduling article a couple of months ago. I had many replies and comments about the article, which was unusual because I don't write these in a manner that will prompt a reply. However, the comments ranged from, "Great timing! I've got to show this to the boss today" to "Well, it's about time someone put in writing what the real world is like." So I thought I'd try discussing the next step and make up a detailed work estimate and schedule.

Scheduling is especially difficult when you're trying to schedule work that you have never done before. For example, I just installed a network on my old DOS machine. I expected eight hours of work spread out over a week. I spent more time looking for the power adapter for the hub than installing the network. Less than two hours total! But, this is a perfect example of a scheduling assignment.

I had network experience with Windows 95/98. The folks who I share office space with attempted the same task a couple of months ago, and I took notes. I used this information for my estimate, but it took less time. That's always a pleasant surprise. If I didn't have part of the network running or know of my neighbor's experiences, I might have taken more time and been closer to my original estimate.

But, now I'm trying to add a printer to the network. I've spent about three hours a day for the last four days, and I've gotten nowhere. I called the shop that I get my equipment from, and they said they didn't have any trouble with this sort of installation. No magic required. I tried the installation once more and it worked. All systems recognized the printer. My point is, how do you schedule this? Well, my original estimate wasn't that far off after all.

The crux here is to make estimates, record your actual hours, and verify your accuracy. Keep repeating this on all your estimate efforts and you'll improve your estimating ability.

EXPERIENCE REQUIRED?

What really prompted me to "revisit" this topic so soon was a rush project for one of my customers. I suspect that the sales staff was attempting to show responsiveness and promised the end user a quick turnaround. It turned out that their customer supplied some units later than expected, and it turned into a rush project on Friday that didn't get completed on time. The internal turnaround time for the engineers on the project shrank from one week to four hours. We estimated one week for the work but didn't express that the task began one week from when we received the customer's material. Everyone remembered the delivery date and forgot our terms and conditions (re-

quirements and dependencies).

I heard a great comment on the radio today. It went something like, "If I don't deliver what the customer expects, then I've failed. If I give you one million dollars but you expected 10 million, then I've failed." The part that you should remember is to manage the customer's expectations. A good salesperson will do this at all times and never let the customer's expectations get out of line.

A good schedule from an engineer must also not let your manager's expectations get out of line. If you need a specification before designing the system, you and only you must ensure that no one thinks you can start anything before that specification is complete. It's just human nature to assume the best (that's what accountants are for, assuming the worst), so keep sales and management in line.

When I estimated the network, I drew upon experience. But, what if you don't have any experience in the project you're being asked to estimate? You can break it down into small and easily identifiable and measurable tasks.

AN EXAMPLE

Suppose you're designing and building a glue dispenser for placing glue dots for surface-mounted printed circuit boards. Marketing has asked for an estimate of how much and how long. The equivalent of, "the sky is falling, the sky is falling!"

Let's assume that it's PC-based and has a custom (ISA) plug-in card for driving and sensing. The work will fall into three areas: software, electrical, and mechanical. You don't have in-house mechanical, so you're going to have to go outside for that.

The best specification you could get was "make it as low-cost as possible but fast and accurate." You'll have to start with some assumptions. The PC can run any operating system you want, and the board sizes are reasonable, say $12^2 \times 12^2$. The positioning accuracy is $\pm 0.005^2$ over the entire work area, and the glue dispensing components are a readily available system from an existing manufacturer.

REQUIREMENTS

First, you have to put down some mechanical requirements to bind your project:

- the board mounting area has a straight edge for rotational orientation
- the board is held down with simple clamps
- the board has targets for registration
- calibration from week to week in the field is OK
- stepper motors meet the speed requirements. If this is not true, then you need servo motors, and there's a lot more expense involved
- you can purchase an X–Y table suitable for this project. (This is probably not true, but for this article let's pretend.)
- X–Y motion is with steppers and Z motion is with an air solenoid. You need shop air for the glue dispenser, so a small air solenoid is not asking too much.

Next, you have to list some electrical requirements:

- stepper motors have hardware drivers available
- limit switch and emergency stop switches are available on the X–Y table
- the PC's power supply can drive the motors
- one ISA card will fit all the interface electronics

And finally, some software requirements:

- the user can program the unit by moving the dispenser (with a joystick or keyboard input) and record locations and dispensing amounts
- the user can also enter an ASCII text file with program commands
- the program will be written in C and run under DOS
- calibration is a requirement
- speed of operation is not a major requirement in this device

This is a good first pass. I suggest you present this to management and make sure they understand what you're about to do. You should say things like, "Now, you realize that this won't run under Windows." If you don't point out these things, you'll be sorry. If marketing needs Windows, then you need to include that in your estimates. And really, if Windows takes more time and effort but is absolutely necessary, now is the time to have that discussion.

ESTIMATING

Let's first attempt to estimate the software. The first thing to do is to put this all into a spreadsheet so you can get accurate totals and keep track of changes (see Table 1). I never really learned how to use Microsoft Word's outlining capabilities, so a spreadsheet is easier for me to work with and modify. You now have a starting point to optimize.

The next step is to review the numbers. For each of the items, my estimate includes design, coding, compiling, unit testing, and system testing. Unit testing is testing the modules with custom inputs that cover the design issues. For example, in unit testing file names, I would try to enter valid names (both short and long) and also invalid names (perhaps some file names typically found on Macintosh systems or UNIX systems). There's no telling what mistakes the typical user might make.

My goal at to this point is to get something on paper that covers the software. I can now check to see if I left out anything and review the amount of work that I have estimated. So, let's take a more detailed look at how I arrived at these initial estimates.

USER INTERFACE

I'm assuming a good-old DOS text window type of look, only because I'm familiar with it and I can avoid getting into the pros and cons of a Windows*xx* discussion. The 40 h for defining screen graphics mode is a bit much, and upon review, I reduced that estimate to 20 h. I've included all the changes that are likely to occur in this item. Also, I would like to get a menu system that is a purchased package. No need to reinvent anything here. The last item is to build a static working demo where all

the pages appear and have the appropriate messages. The marketing department can walk through the menus and give their approval. And, it's much easier to make structural changes at this point than at any other time.

In the File Menu section, I reduced some of the numbers from 10 or 16 h to one day (8 h). In the File Compare area, I reduced the tasks for "differing points" items. The real work is in the "same points" items and probably just a simple code change to show differences.

I just sharpened my pencil a bit, and all these items are similar. So, revising goes on through all the items. As you can see, I took out 160 h total and saved one man month for this project.

BREAKING IT DOWN

When I worked for a large company, instead of scheduling 40 h per week for engineering, I scheduled only 32. That other 20% (8 h) went into general overhead. Production would need engineering help, as would sales. If you had one designer working on this project at 40 h per week, the time to complete would be 20.15 weeks, assuming no vacation, sick days, nor other interruptions. Currently, my independent status lets me schedule a full 40 h of effort per week. Be careful converting task estimated to calendar time.

Also, I did not link any of these tasks in a dependency relationship. Clearly you can't move motors until the hardware is in place. You can't even design a user interface until you have an agreement on the requirements. Therefore, the start date is not yesterday!

Did I leave anything out? Probably. Will there be surprises and changes to the plan? Yes. That is why you have the contingency.

How good are these estimates? Well, I've done most of these tasks before, so I'm confident with the numbers. But, what if you've never done something like a calibration routine before? How can you estimate that task? Ask your coworkers. Ask industry experts. Gather all the input you can and then take your best guess. Break that unknown task into its smallest steps, and measure your progress. You should be able to predict the outcome before you're halfway through.

Next month, I'd like to talk about making the switch from assembler to C and bringing up your first project. Until then, good luck and keep good records. You'll see the results on your first project.

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' new house. You can reach him at george.martin@worldnet.att.net.

Circuit Cellar, the Magazine for Computer Applications. Reprinted by permission. For subscription information, call (860) 875-2199, subscribe@circuitcellar.com or www.circuitcellar.com/subscribe.htm.

	First Pass	My Review		Chg
1. Design User Interface.				
a. Text based screen with pull down menus.				
Define Screen Graphics Mode.	40	20		-20
Find menu system that is compatible with Compiler.	20	20		0
Select Color Scheme.	20	20		0
Build Static Demo with all menus working.	40	40		0
b. File Menu includes:				
i. New.				
Long File name support?	10	8		-2
Create-Open a file.	16	16		0
ii. Open a file	10	8		-2
iii. Save a file	10	8		-2
iv. Save As a file	10	4		-6
iii. Compare.				
Compare programs.	20	20		0
Show Same Points.	20	20		0
Show Different points.	20	10		-10
List Same Points.	20	20		0
List Different Point.	20	10		-10
c. Edit Menu includes:				
Capture Points From Table.	40	30		-10
Delete Point.	10	8		-2
Insert Point.	10	8		-2
Modify Point.	20	4		-16
d. Orientation Menu includes:				
Align Lower Left Point.	10	8		-2
Align Upper Right Point.	10	4		-6
Rotate, Reverse X-Y.	20	16		-4
2. Design Files and File structures.				
Calibration File.	40	40		0
System Definition File.	20	20		0
Dispense File.	20	20		0
3. Design File Utilities.				
Print.	40	20		-20
Compare.	10	10		0
Load.	20	20		0
Edit.	20	20		0
4. Control Stepper Motors.				
Step + direction.	20	20		0
Step – Direction.	10	10		0
Go Home.	10	10		0
Go To X-Y.	10	10		0
5. Control Glue Dispenser.				
Head Up.	20	20		0
Head Down.	10	8		-2
Dispense.	10	10		0
6. Control Safety switches.				
Run to Limit Switch.	10	4		-6
Recover from Limit.	10	8		-2
Run to Over Travel.	10	4		-6
Recover from Over Travel.	20	10		-10
Test Limit Switches.	20	20		0
7. Allow User Programming.				
Learn from Existing board.	40	40		0
8. Allow Text File Programming.				
Read Customer File.	40	40		0
9. Calibrate Table.				
Design Calibration Fixture.	20	20		0
Design Calibration Software.	40	20		-20
Run calibration.	40	40		0
10. Create Test Software.				
Design Self Test Software.	40	40		0
Design Self Test Report.	20	20		0
11. Contingency.	80	80		0
	Totals			0
	hours	1046	886	-160
Assume 8 hours per day	days	130.75	110.75	-20
Assume 40 hours per week	weeks	26.15	22.15	-4
Assumer 4 weeks per month	months	6.5375	5.5375	-1

Table 1