

1. Automated data collection from transit vehicles (50%)

Your consulting company just received a contract to develop an automated passenger count information system for a transit agency. Counters are being placed on all buses in the fleet and will report data at the end of each day. The agency has retained your company to write the system to display the data in a Web-based system.

The selection committee thought your proposal was wonderful, and it additionally wanted to upload the data every evening to the system that also has automated fare data (from smart cards). You had written your proposal 6 months earlier as a stand-alone system without the ability to interface with other systems. The project was 18 months in duration and you feel that you can add the extra features within that time.

You meet with your boss the next day, who had participated in the final presentations and question-answer period. She said that she felt she had to promise an interface to the fare system to get the project. She also swallowed hard and said that she'd promised to deliver the automated data collection system, including the interface to the fare system, in 12 months, not 18 as the proposal had indicated. Your boss tells you that this will be a good challenge for you and that she has every confidence that you'll succeed. If you do, there will be a promotion to upper management for you. And there are probably some inessential features that can be eliminated from the system as proposed, so that should save some time. "Perhaps you should look for those first," she says.

You had asked for three people for the project, and your boss tells you that you can now have 4.5 people, to keep the cost the same. You feel that will cut your time by a few months, though not by 6 months, so you assume that the time estimate to complete the system is now approximately 14 months. You've read about some tools for generating database queries automatically; they seem appropriate for the project, so you assume that they will save another month, bringing the project time to 13 months. That seems close enough, so you go forward based on that plan. You assume that one of your developers can write the interface to the fare system in addition to their other tasks. "That's what it will have to be, at any rate," you say.

It takes you a month to assemble and organize your team, find offices and computers, buy and install software, and the like, which is very quick by company standards, since this is a high priority project. You now have 11 months left. The requirements are about 25 pages long, written at a high level. You assume you can flesh them out in two weeks, but people are on vacation and traveling, so it takes 4 weeks. You now have 10 months left. The requirements document turns into 100 pages, with about 40 database tables and 50 Web pages, most fairly complex. You meet with QA and test, who tell you they need your system two months before release, so that leaves you 8 months to build it. You will write the system in C# and SQL,

modern languages. Your developers are experienced, but only two of them have worked in C#; the others know similar languages. None have used the database query tool before; a couple of them have lingering commitments to a prior project.

You prepare a design 4 weeks after the requirements, leaving you 7 months to write the system. You've assigned your team of 4.5 people (5 individuals) as follows: 1 for database, 2 for Web site, 1 for the application logic and 1 person for the system interface. Each begins on their tasks and codes for 3 months before having an initial release; each component has about 1/3 of the required capabilities, and many of the more complex features have not been started. You are at month 6; you have 4 months left to complete 2/3 of the system; you ask everyone to work evenings and Saturdays, and they agree. It doesn't seem worth trying to tie the pieces together since they are so incomplete at this stage.

Three months later, at month 9, most components are 75% complete; one month is left before the system must be given to QA. You ask the developers to work Sundays also for the 6 weeks, to complete the system. They do, and you start integration testing with two weeks left in the schedule. When they begin, they discover that the query generator code is so inefficient that each page takes 3 minutes to load. The columns on the Web pages are not in the required order, since the query generator can only place variable length fields as the final columns of a report. The users, who are given a sneak preview, find this unacceptable.

Your developer in charge of the queries says that it will take her 6 months to code every Web page by hand. She will need to spend the next month improving performance as much as possible, a major task in itself. There are many glitches in the application, the Web pages, and the interface has not been tested at all with the fare system.

It takes a month for the system to successfully build with all components, but many things don't work correctly. You release it to QA at the end of month 10; in their first day of testing, they find 300 bugs. QA says it's not worth testing any more until the system is more complete.

It takes another 4 months to correct many of the reported bugs and to improve performance to a minimal level (each page takes 20 seconds to display). The interface to the fare system still doesn't work, but the customer agrees to take the system without it. QA still finds 100 bugs on its first test of the revised system. You are at month 14.

You are removed from the project; a new manager comes in, gives the team a long weekend off, examines the bug statistics, size of code, and reorganizes the effort. The project is rescheduled for an additional 4 months, with 40 hour weeks. Frequent integration is done, buggy nonessential features are removed, buggy modules are redesigned and rewritten completely. After 3 months of QA, training and installation, the system is released 9 months late, at month 21, to an unhappy customer.

-
-
-
-

Hand in

Hand in your answer via the Stellar Web site as a Microsoft Word document. There is not a single answer to these questions; you should provide a reasoned response, referencing material in McConnell. Cover only the major points.

2. Software installation (50%)

Install Visual Paradigm, SQL Server Express 2008 and Visual Web Designer 2008 software on your laptop computer. If needed, install Java runtime software, make sure that Microsoft security patches are being installed and that a firewall and virus protection are installed. See the reading on the Stellar Web site for detailed download and installation instructions.

