

## 1. A demand forecasting system

Pat, your manager, has returned from an MIT course, excited at having learned about new methods of demand forecasting for your organization. Your manager calls a meeting about the new project. “This demand forecasting system, which we’re buying from a vendor but also modifying for our organization, is critical to our future. We’ve anticipated demand changes very poorly in the past, and we have not had sufficient capacity in the right places at the right time to meet our customers’ needs. I know that our organization hasn’t supported logistics and transportation system efforts very well in the past, so this time we’ll give you everything you need. You’ll have good office space, good servers and software support, and you won’t have to attend most of the regular staff meetings.”

“That sounds excellent,” you say. “My staff, which is all recent MIT SM grads, is really enthusiastic. (1) They don’t have as much experience as I’d like, but their energy and smarts will more than make up for it. I’ll get (2) commitments from them to work really hard in exchange for allowing them to be really creative, so that we complete the project on time.”

- 1: Insufficient skill for fastest possible or maybe even nominal schedule
- 2: Commitments based on self-estimation of schedule are usually way off.

(3) “I’ve budgeted this as an 8 month project, and I’ve promised the Board of Directors that we’ll be done on time,” Pat said. “This is a very high visibility effort.”

- 3: No resource estimate based on function points.

Alex, from QA, says, “If we’re going to be using the system at the 8 month point, we need to have the software frozen at the 6 month mark, so we can write the test scripts. Since your system will interface with our current customer measurement software and our geographic information system (GIS), we’ll need time to test the interfaces.” Chris, from training, also chimes in: “We’ll need the system frozen at the 6 month point also, to prepare training materials for the staff.” You agree to the 6 month freeze.

As your staff of 4 analysts begins the project, they’re very energized by the challenge and the freedom of the effort. (4) They split up the project into pieces: Joe defined the interfaces with the other systems, and Karen designed the database and system interface components to achieve this. Kate configured the statistical modules of the vendor-supplied package, and specified some additional analytics that Jim would write as extensions. Karen liked to design user interfaces, so she took on that task. (5) You managed the project using a spreadsheet and simple project management software (Microsoft Project).

- 4: No integration in plan.
- 5: Inadequate management: it’s not about running MS Project.

(6) The staff took one month to be trained in the new software and to talk to the groups who did the current customer measurement and GIS systems. They discovered that most, but not all, the data needed to improve the demand forecasts was available, and that some existing data was coded incorrectly and grouped in ways that didn't work well. They negotiated changes with the customer measurement and GIS system groups, who promised the changes would be complete in two months (at the 3 month point in the project).

6: Unplanned task; poor process.

(7) The interfaces had to be redesigned when the other groups completed their work. Because they had to make the changes quickly, the other group tweaked the existing interfaces and queries to produce what was needed, even though in some cases much more data was sent than needed, and in other cases, the data had to be reorganized within the demand forecasting system to be usable. This took all of month 4, and a few compromises had to be made.

7: No spiral/agile process. No allowance for changes, which leads to excess rework.

The statistics modules then had to be redesigned because the compromises affected the models; this took month 5, and then month 6 to take care of some nagging problems. There were still a few spots that didn't seem to work properly, but the staff could make them work with care.

The user interface then had to be redesigned, and this took through month 7. Because the user interface wasn't ready, neither QA nor training could start their work. You said, (8) “Well, you really shouldn't need two months for just QA and training. It shouldn't take 25% of the effort just to do that; we've been busting our butts to build all this. You can do it in one month.” You go to your boss, and say that you really need her support in this, and that she'd promised you a lot of support for this project. She reluctantly agrees, and tells QA and training they have to be done in one month. (9) She tells the Board the project is still on schedule.

8: Shortchanged QA.

9: Lying.

Your team finally finished the first version of the customized software at the 8 month point and hands it over to QA and training. They can't get the system to run properly. Your staff sits with them and points out all the parameters that must be set in just one way, and the exact sequence in which all the analyses must be run. It's very complex, and QA and training both say that the users can't deal with it. (10) Also, any small error results in system crashes or crazy results.

10: Lack of integration, lack of process, lack of design, lack of QA...

An impasse results; you all meet with Pat the next week, during month 9. Pat asks for a demo of the system, which you haven't been able to provide until then. After the demo, she says that she can't understand how the system works and she couldn't use it either.

She clears her schedule, and sits with the five of you, plus QA and training, for the remaining four days that week, simplifying the system, throwing out many sophisticated models that don't quite work, streamlining the system interfaces (making them just one-way into the forecasting system, with nothing going back to the other systems, as originally intended), and eliminating half the user interface Web pages.

She's mad. She tells your team to work 12 hours a day, 6 days a week, and to finish the simpler system by the end of month 9. Your staff grumbles mightily but is embarrassed and does it. You finish in the middle of month 10. QA and training take the system, and find many bugs and inconsistencies. They also work overtime for the next 6 weeks and you finally deliver a simple system, not much different than the 'out of the box' configuration, at month 12.

You get a 1% raise for the year, and you feel very underappreciated. But when you interview for other jobs, no one is that excited to hire you...

**Questions. Answer each in a sentence or two. Be prepared to discuss them in class.**

- a. List as many errors that were made by this team as you can.**
- b. What did the team do right?**
- c. What project management method was used? Was it appropriate?**
- d. What should they have done to succeed?**

**Answers:**

**a. List as many errors that were made by this team as you can.**

Process: See above, and:

Rework: the system was done 3 times: initial, then again at month 4, and again at month 9

QA: started way too late. QA should start at project start

Development fundamentals: Poor requirements, design, implementation, QA cycle. Should have used spiral model to manage risk

Risk management: Not done

Lifecycle planning: Didn't take actual use into account very well

Customer orientation: Not really done

Product:

Product characteristics: Too complex

**b. What did the team do right?**

People:

Had good staff, though didn't account for lack of experience

Team organization was ok

Motivation was high

Process:

Resource targeting was ok

Product size:

Size was close to ok; if process had been better, product would have worked

Technology:

Probably a good choice to use vendor product and modify

**c. What project management method was used? Was it appropriate?**

There was no obvious method used, which is inappropriate.

**d. What should they have done to succeed?**

First, avoid the classic mistakes:

People: Few mistakes were made. Lack of user input is the largest one; users should have been involved more

Process: This was the major problem. Lack of realistic schedules and resource management and planning needs to be corrected. Risk management needed to be done (top risks list). Inadequate requirements and design; inadequate review; much more is needed. Too little QA; more is needed. No management controls really existed, even with Microsoft Project. Some research-oriented development occurred; research should have been done first.

Product was mostly ok, as was the technology.

**2. Demand forecasting system, version 3**

After a very shaky implementation of the first version of the demand forecasting system, the organization decided to upgrade it in next year's work plan. This didn't go very well either, but the system was more usable and more stable after a 12-month effort (that had, again, been scheduled for 8 months.) To correct still more problems, Pat asked you to make recommendations for version 3, now that everyone had two years of experience with the system. You were given a month to do the assessment.

Your report recommended that you rewrite major portions of the model. More GIS data was necessary than you had initially expected in version 1:

- Users wanted reports of demand changes overlaid on a map, not in text reports, which they found very difficult to use
- Users needed demand estimates in finer spatial detail than you had assumed, so the statistical methods had to be revised. There was barely enough data in each zone to do the statistics you'd implemented in version 1, and so the models had to be modified and simplified
- Users wanted shorter term forecasts and more recent actual data: they wanted yesterday's and last week's data, not just monthly and quarterly past data. This required a different interface to the customer measurement system.
- Users wanted less analysis and more creative display of data, a different system focus. The vendor system didn't support this very well.

Pat approved the approach; you had two of the original team members and two new team members for version 3; you estimated it would take 12 months, given past experience. Pat said you had to deliver in 11 months, so the Board could be sure the project was complete during this budget cycle and could be included in the annual report. You agree.

You estimated each part of the project with 30% more time than in versions 1 or 2, which would cover the overruns you had seen. However, when the team began working on version 3, tasks turned out to be more complicated than expected. The new team members took a long time to become familiar with version 2 of the system because it wasn't well documented. Tasks were taking about 60% longer, not 30% longer than versions 1 or 2. It was hard to predict when a task would be completed; overlooked items or issues cropped up unexpectedly and frequently.

You tell Pat, at the 4 month point that you are one month behind but can make up the time, because the issues are familiar by now. She doesn't believe you and hires a consultant, Bill, to review your work. Bill gives you some bad news: he estimates that you'll lose two more months on the schedule, and will deliver in 14 months, not 11. "Even though you are familiar with the GIS and customer management systems, and the vendor forecasting system, you still don't have management control of the effort. Surprises will crop up because of inadequate process; you can't make design tradeoffs because you don't know what they are; QA is still inadequate because it's left to the end; and there are many other problems."

Bill continues, "Version 3 will be a better system: it will be easier to use, because you understand your users; it will integrate with the other systems better, because you understand them better. But it will still have quality problems and you'll be late because of the lack of good process."

**Questions. Answer each in a sentence or two. Be prepared to discuss them in class.**

- a. At the 4 month point, what do you, Pat, do? You can have some additional resources; specify those you would like to have.**
- b. With your suggested actions, will you be able to deliver the system on time, in 11 months? Why or why not?**
- c. With your suggested actions, how certain will you be at month 8 whether you can deliver on time?**

**Answers.**

- a. At the 4 month point, what do you, Pat, do? You can have some additional resources; specify those you would like to have.**

Don't add any more analysts to your staff; it will make the project later. Do add a QA person immediately to work on requirements and design reviews, analysis reviews, and

test plans. Add a training person part time to begin on training materials and to comment on usability.

Move to a spiral model. Create a deliverable system at month 7, even if simple. Take 2 weeks for requirements, 2 weeks for design (month 5), use month 6 to implement the user and system interfaces, statistics and reports. Use month 7 for intensive QA and initial training materials.

After that, complete another spiral in months 8-11 and deliver that as version 3.

Improve your estimation and scheduling (we'll cover how to do it later, but understand you need to do it now).

Improve your planning and tracking, to your team and to Pat, your boss. Weekly visibility seems appropriate.

Miniature milestones are key to project recovery. Tasks are either done or not done; there are no "75% complete" tasks.

Implement technical fundamentals: requirements, design, implementation documents and reviews.

Make a top 10 risks list, from versions 1 and 2, and current issues in version 3. Assess them weekly and respond as needed.

**b. With your suggested actions, will you be able to deliver the system on time, in 11 months? Why or why not?**

With all the suggestions in part a, yes. If you add analysts, or don't implement good process, no.

**c. With your suggested actions, how certain will you be at month 8 whether you can deliver on time?**

If you use the spiral model, you will have very good information. If you can't create a deliverable system at month 7, you're unlikely to make it at month 11. If you can, you're likely to be fine in month 11. You may or may not meet all the expectations (requirements), but you can adjust expectations in month 8 based on what you were able to deliver in the first spiral.

MIT OpenCourseWare  
<http://ocw.mit.edu>

1.264J / ESD.264J Database, Internet, and Systems Integration Technologies  
Fall 2013

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.