Homework 1 solution
1.264, Fall 2013
Software process

1. Automated data collection from transit vehicles (35%)

The main points of the solution should be:

1. The boss and you changed the schedule from 18 months to 21 months somewhat arbitrarily. The boss doesn't understand software process—and should not have done this—and you should not have accepted it without principled negotiation.

2. With a 21 month schedule, a spiral model should have been used, bringing the system to a deliverable state every few months. You are implicitly using a waterfall model, which does not manage risk well.

3. Schedule is apparently the most important driver, and designing to schedule is probably the best approach, which is well supported by the spiral model. (You can also argue that cost is the key driver, but the case study notes that the transit agency is unhappy about the schedule, not the cost. You might convince them to increase the cost more easily than to slip the schedule.)

4. The team size should have been reviewed to ensure it fit with the project scope. This was not done. Almost certainly, the team needed another member to work on the farebox interface, and perhaps more. The effort should have been re-estimated, even if very approximately.

5. The project requirements were not rewritten; the project was not re-scoped with the new schedule in mind. Your team will work at a nominal rate at best, not fastest possible or efficient. You should re-estimate the project size (lines of code), schedule and team size.

6. The addition of an interface to a vendor with no participation in the project was a significant risk that should have had a management strategy. The transit agency should have been asked to accept some or most of the risk of their farebox vendor not performing. A risks list is important right from the start of the project.

7. You don't really control the software process; it is at level 1 (chaos). You should be at CMMI level 2 or 3, and should devote some resources at the start of the project to measure the resources, size, quality and other variables of the effort.

8. At the 6 month point, you have 15 months until the 21 point initial delivery point, which is often plenty of time to save a project. You can do three 5-month spirals, or more short spirals, to incrementally build the system. QA and documentation will get interim releases that they can use to test and write documentation. You can learn about other risks; you can find items to cut in each spiral, etc. You can simulate the

farebox, per the published interface, to decouple your effort from the farebox vendor. If you deliver your software that works with the simulator, it's then the farebox vendor's problem to fix their software.

9. In summary, the solution to these problems is a better software process based on developing good requirements, good designs, using the time and resource estimation methods to plan and scope the work, and using the spiral model and supporting methods to manage the implementation.

10. The process must involve top management, who must buy into the principles and the need to have software process; it must involve the transportation planning and business staff, who need to understand the process and their role in it; and, of course, it must involve the software staff.

2. Manufacturing process software (15%)

1. This is primarily a requirements issue. The consultant can't learn your manufacturing processes in 6 months, so you should have had a detailed requirements document before they began.
2. Agile processes that have little documentation are not suited for all projects. A phase in which (ideally) the company staff wrote requirements and then spent time educating the consultant would have resulted in the first few agile spirals being much more useful.
3. This is an example of a 'silver bullet' syndrome, in which a company thinks a tool or, in this case, an outside vendor, is a magic answer. There are no magic answers.
4. The company should have dedicated its own staff to work within the consultant teams as domain experts within each agile spiral.
5. To salvage this problem, the company needs to assign internal domain experts and perhaps some internal IT staff to work with the consultant.
6. It may want to use some longer spirals than the agile method uses, and it should produce written design documents that can have contributions from internal staff, and can be reviewed internally. More structure is probably needed.
7. The consultant staff should be allocated to tasks they can do well. These are probably coding tasks that have good requirements. The internal staff should do most of the requirements. Design should be a joint effort between internal staff and consultant staff.

3. Software installation. Checked by TA in person with you.

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.