The Development System for CSE403

People, Processes and Tools

To the class:

I much enjoyed my time with you. The key points (as far as I meant to touch them) are:

- *Software Engineering*, the techniques, tools, models, and activities that we use to make systems can themselves be objects of study – that's what the journaling is doing, is it not?
- One way to view *Software Engineering* is in terms of *The Development System* - the collection of *People, Processes* and *Tools* used to make systems. Treating this as a system, it can be studied using systems tools, as well as other, more human oriented tools.
- *The Development System* choices are often implicit. Making them explicit (a *Diagram of Effects* is one way) gives us more options when choosing how to go about building systems.

I noticed several things in your questions that were astute, and unusual in software development:

- There was interest in borrowing from other disciplines, and skepticism about Taylor-style "Scientific Management[1]" to developing systems. Some other sources:
    - Engineering, as a practice, vs. manufacturing as a practice.
    - Media production: magazine, newspaper, books, television and film.
    - Process and descriptive models from other disciplines. I've found useful from chemical engineering: chemical process design, kinetics, thermodynamics and process control.
    - I've found useful the models and approaches developed in abstract algebra, and calculus (for math majors – all proofs), information theory, game theory & economics.
    - The organization of this class, and many of the guest presenters are heavily influenced by the work of Jerry Weinberg, who applies and teaches techniques that originate in part in family therapy[2]. I have found that a consulting engagement is similar in structure to a therapeutic intervention (this observation is hardly original.)
- There's a lot of reflection on what works, and what doesn't in your questions. That's a good habit. Be aware that there's a fair amount of dogmatism in this field. You may well encounter people clinging to *the right way* in the face of incredible evidence that this isn't working.
- You wanted to get stuff right the first time – architecture, integration, team sizes, project plans and so on. I prefer to assume that there's a lot I don't know, then take a small first step that will create information. Iterative methodologies codify this approach. With projects, I treat unknowns as a conserved quantity – one that has to be consumed by The Development System, over the course of a successful project.

---

[1] Fredrick Winslow Taylor, who wrote a book *Scientific Management* about how to organize and optimize company practices. He took his models, mainly from time motion studies in discrete manufacturing and assembly.

Copyright © 1999, James Bullock, Rare Bird Enterprises jbullock@rare-bird-ent.com

My Journal Entry:

It seems only fair to share a *Journal Entry* for this experience with you:

Stepping in front of a class at this level is a bit daunting. I'm somewhat better, because of my interviewing and hiring at various top schools over the last several years. It's still a bit scary. I've trained myself in the discipline I'm about to talk about. But, I a reminded, when I encounter a group of top students, that I was not trained in this field.

The questions were interesting, as the class is clearly grappling with the realities of getting a non-trivial system built. David did something very insightful in organizing the class around *getting this thing built*. There's a model of good software engineering in that choice. "What are we trying to do here?" has got to have an answer. We went over this a lot in the *Roundtable on Project Management,* but I haven't seen it much in other literature. (III – that's his name, "III" - has a piece on project charters. Same problem space.)

The biggest thing I learned – or relearned – in this exercise is the strength of the link between software engineering practices, organizational design, and management. Tools reflect and enable a particular life cycle. I've got that. The choice of tools, life cycle, and so on is deeply intermingled with management practices and attitudes. I've got to watch out for that one. In my early career experiences at Techtran and Carrier, the management involved was sufficiently frustrated at the lack of delivery that they abdicated tool selection and use to developers. (The XP folks will say this is how it should be.) So we did what made sense, and shipped a lot of stuff. Since then I've encountered a large number of managers – former technical people – who know "how it works" and impose this on the people who work for them. Or the assorted believers in the big methodology – the one that specifies everything. Makes them all miserable.

What I did right . . .

I used the consulting model for my presentation vs. sticking rigidly to the ideas that David and I had discussed. The class seemed tired, and a bit preoccupied – not surprising at the end of a project, especially the first project for a given team. So you have to start by meeting them where they are, and use their current concerns to illustrate what I want to talk about. With luck, the written notes will emphasize some useful tools that we didn't get to.

When addressing questions, I was doing "diagram of effects" kinds of things, and then suggesting interventions – tools and tactics – that would have indirect effects on the question asked. I don't know if they noticed. I might have done the system diagrams to illustrate what I was doing with my words and questions. Questions about project management, and I talked about interfaces. Questions about architecture, and I talked about rapid, iterative deliveries. There's a diagram of effects behind each of these seeming non-sequiturs. Probably prepared them a bit, when I introduced a schematic version of the same thing. Some things I might have covered in terms of a diagram of effects:

- What happens as the requirements on a system, or environment (say, market) for a system change? What changes in the development system, to be successful?
- What changes with scale of the system being built? What are the various measures of "scale?" What are possible countermeasures, as system scale changes?

---

[2] Jerry was a personal student of the family therapist Virginia Satir, who was one of the first, if not the first, to apply systems thinking to theraputic interventions.

Copyright © 1999, James Bullock, Rare Bird Enterprises jbullock@rare-bird-ent.com

They sure got the stopwatch thing. Don't know if I was clear enough, though. Any particular amount of time may be the right amount or not. The point is to notice the amount of time being applied, and compare that to what seems important. There are at least two pieces of information there – something takes a long time, it's important to somebody. Something takes longer than you expect, you don't understand it's importance to someone.

The last thought is about journals and similar related practices. They work – as does any other kind of feedback – if they're interpreted in terms of some point of view at least. *The Development System* is constructed to create something, or a collection of things. Feedback about how it is working is interesting in terms of the stuff we're trying to build.

From a personal point of view, in journaling I've used these questions:

What have you done today,

To make your life better?

To make your self better?

To make your world better?

Copyright © 1999, James Bullock, Rare Bird Enterprises jbullock@rare-bird-ent.com