

# Faster, Better, Cheaper Ethical Perspectives on the News

Leon Tabak

30 May 2017

The largest computer programs rank among the most complex objects that human beings have ever designed and built. Software engineering is, at its limits, among our most ambitious undertakings. Learning how to manage complexity is one of the great challenges of software engineering.

We will be talking about how what software engineers have learned about how to manage projects ("agile") is finding application in all kinds of organizations.

## 1 Origins

- What led to the development of agile development?
- What are some of the promises and goals of agile development?
- Can you introduce our subject to an audience that may have little knowledge or interest in the details of engineering?
- What are some of the most important ways in which agile development differs from other methods of guiding the creation of a product or service from start to finish?

## 2 Distinguishing features

- What have we learned about the optimal size of a team? (A good team can be fed with two pizzas.)
- How frequent are your milestones and due dates?
- Do the members of your teams post notices of their progress and status for all to see? How do members of your team make themselves responsible to one another?
- How does the discipline change the relationships among members of developers and between developers and clients?
- Speak about the importance of getting some kind of product into the customer's hands early, getting feedback from the customer, and building (evolving) the product through a series of progressively better versions.
- Talk about how we assure quality in our products and services. When does testing begin? What is tested? Who is responsible?
- What are the working relationships between engineers and other members of your organizations? How do they work together? When in the course of a project does their collaboration begin?

You might talk about how software engineers on a team and the other (non-engineer) members of the team work closely with one another (rather than in isolated silos) and in parallel (rather than sequentially by passing documents over a wall).

- Have you seen agile principles and methods adopted by groups within your organization that are responsible for tasks other than the creation of software?

## 3 Focus on clients' needs

- How do the attitudes and expectations of members of team, as much as the specific steps they taking in planning and tracking progress, contribute to the success of the team?

- How do modern methods of creating software engage the development team with the clients from start through finish?
- To what extent is a focus on the clients' needs the heart of the agile philosophy? Can we then speak of the move to agile methods as a shift in ethical commitment as much as an adoption of new tools and procedures?
- Is there an ethical component to these attitudes and expectations?
- Can you talk a little about the focus on the client's needs that is at the center of agile methods?
- Is this focus on the client's needs an ethical commitment?

## 4 Attitudes, expectations, and habits

- There is a code of professional ethics for software engineers. Of course, it instructs us to avoid lying, cheating, and stealing, but it also asks us to do more. In our profession, we have defined ethical conduct to mean acting in the client's best interest, collaborating with colleagues, helping junior colleagues advance, and continuing our own education. How do such habits figure in a successful agile project?
- What are some of the key attitudes and habits that you seek among the people you recruit for your teams?
- How do you cultivate some of these non-technical skills among the members of your teams? Are colleges and universities doing enough to prepare students for this way of working?

## 5 Applications outside of software engineering

- Have you seen the vocabulary and discipline of agile development showing up in teams whose work has nothing to do with the development

of software?

- What have we learned in software engineering that might help organizations of all kinds become more responsive to the needs and wants of their clients?
- How can people who are delivering something other than software adopt these principles and practices?
- Many kinds of organizations, including public agencies, journalists, and makers of clothing, have adopted agile methods. What do you think they are learning from the experiences of software engineers?
- I frequently see agile methods defined in contrast to the “waterfall model.” How might our skepticism about the value of big, complex plans drawn up in great detail at the start of a project by a handful of experts apply outside of the engineering field?
- How does work within this discipline require a different set of expectations and responses to failure?
- We design iteratively because experience has taught us humility. We know that we cannot discern the whole solution at the start of a project. We expect to discover that some of our initial assumptions are wrong. Henry Petroski, a civil engineer, has written extensively about the role of failure in engineering. When you look at some of the big problems that our nation faces, do you see a need for a wider acceptance of the impossibility of getting things right on the first try?