Escaped from the Lab: Software Practices in Large Organizations

Dennis Mancl
Lucent Technologies

Steven Fraser
QUALCOMM

Ricardo Lopez
QUALCOMM

William Opdyke
Motorola

Greg Utas
Pentennea LLC

Abstract

What are some of the common practices for taking new ideas and converting them into products? What are the key obstacles that cause the failure of research prototypes turned into commercial products? This workshop examines the practices that have worked well and the approaches that can help developers and managers avoid customer problems.

Categories & Subject Descriptors D.2.9 Management, K.4.3 Organizational Impacts, K.6.3 Software Management

General Terms Management, Process

Keywords Prototypes, Reuse

1. Escaped from the Lab

An important dynamic in large organizations: one part of the organization over-commits (promises the earth, moon, and stars), and another part of the organization is forced to deliver. The extravagant promise of the Powerpoint presentation is converted into the trail of tears of the Gantt chart.

The grandiose project was originally supposed to be feasible. There were some small technology trials that proved out the basic ideas for low-volume transaction rates and simplified user interfaces. The product was supposed to be delivered in record time because of high rates of software reuse. So what went wrong?

One thing that compounds this effect: only a small fraction of the people working on software are actually in firms that consider themselves to be in the software business. This may lead a team to make inadequate choices of practices and approaches for the process of turning new ideas into products.

This workshop will explore the intersection of object oriented technology, high reliability and performance requirements, large organizations, and conflicts in the software development process. Some questions we will consider include:

- Prototyping: When can a prototype be "scaled up" to production? When is it just an experiment to be thrown away?
- Reusability: How to make an honest assessment of the potential reusability of a module or subsystem?
- Reliability and Availability: How do the reliability and availability requirements get captured and how do they become action items in the product plan?
- Frameworks: When is a framework part of the solution, and when does it become part of the problem?
- Contracting and Outsourcing: problems with development schedule, gaps in the feature set, stability issues, and shortcomings in the product's usability?
- Decision making: Is it possible to define the key issues early in the project lifecycle? What kinds of development processes and management processes are most useful for highlighting the gaps?

2. Some good practices

A few good practices are discussed in existing literature:

- In the development schedule, include extra time for "discovery costs". Managers need to plan for exploration and learning time when moving from a small prototype to a large product development team.
- When writing prototype applications to try out architectural ideas, make sure that the prototype is not a fully working application: whenever possible, leave out key pieces of functionality or build an awkward user interface, so that there is less temptation to ship the prototype as a product.
- Use an agile development process centered on user scenarios when transforming a simple application into a production-grade system. Make sure the final product includes both end-user scenarios and administrative scenarios (such as migrating data, performing backups, recovering from system crashes).

For more information on the subject of this workshop (including some of the workshop materials and the workshop's final report), see the workshop website:

http://mysite.verizon.net/~dennis.mancl/oopsla06.

OOPSLA'06, October 22–26, 2005, Portland, Oregon, USA.
ACM 1-59593-491-X/06/0010.