

Workshop: Beyond Green-Field Software Development: Reuse, Recycle, Refactor

Dennis Mancl

Alcatel-Lucent
Murray Hill, NJ, USA
dennis.mancl@alcatel-lucent.com

Steven D. Fraser

Cisco Systems
San Jose, CA, USA
sdfrazer@acm.org

Bill Opdyke

JP Morgan Chase
Chicago, IL, USA
opdyke@acm.org

1. Main Theme and Goals

Legacy code can be an asset in the software business, but it can also create a considerable amount of extra work. Most software is not a product of “green field” development: many development efforts must reuse code and interfaces from pre-existing systems. Change and growth in requirements and functionality occur in every system – and it is especially critical to manage the evolution process in modern agile software processes. The management of change and growth may in fact become a daily activity, so a well-defined set of techniques and tools for doing *reengineering* and *refactoring* are critical to success.

Leveraging existing software assets are even more important in today’s web-centric and cloud-based environment. It is easier and faster to build new “apps” by building adapters and wrappers for legacy systems and components.

Successful projects that use legacy code must pay attention to people, processes, and tools. The goal of this workshop is to identify the issues facing individuals, teams, and companies that need to build on their legacy base. We plan to explore these subjects:

- The techniques for wrapping and refactoring: software evolution techniques used to add new functionality to existing software systems – and which of the techniques are best for small systems, large systems, embedded systems, and open source software.
- The technical and organizational skills that are needed in a team that is constantly working with legacy sys-

tems and legacy modules.

- How to manage a software system that demands constant change and growth.
- Assessing the business value of refactoring and reengineering efforts.
- Impact of architecture on software reuse and evolution.
- How agile development practices affect the creation of long-lived software modules and components.
- How to measure the quality of reused, reengineered, and refactored code.
- When to abandon a legacy software reengineering effort.
- Exploration of some of the good “design for reuse” practices.

This workshop will cover some of the same topics as our OOPSLA 2003 workshop, and it will address some of the changes in the software industry in the past eight years:

- More tools-based support for refactoring and unit testing.
- More agile and iterative development in small teams.
- More interest in “green technologies.”
- Increased use of refactoring and reengineering to help support moving software functionality to new environments: web apps, smart phones, cloud services.

2. Abstract

There are many languages, tools, and design methodologies in the software community that are aimed at the creation of new software. But a lot of valuable software is the product of evolution, reuse, and reengineering. Some software is too expensive to “throw away and start over.” A skilled software team will have an arsenal of techniques at their disposal for adapting, evolving, and refactoring existing code and designs. Adapting legacy software is a kind of

recycling. If extending a legacy system is done well, it can help deliver business value sooner at a lower cost.

This workshop will explore some old and new techniques for building on existing code – wrapper classes, design patterns, test-driven approaches, refactoring tools, and others. The workshop will also address management issues: what factors to consider in the decision to reengineer or to build anew. This workshop revisits a topic discussed in an OOPSLA 2003 workshop.

3. Organizers

- Primary Contact - Dennis Mancl, Alcatel-Lucent, 600 Mountain Ave., Murray Hill, NJ 07974, USA
 - email: dennis.mancl@alcatel-lucent.com
- Steven D. Fraser, Cisco Research, San Jose, CA, USA
 - email: sdfraser@acm.org
- Bill Opdyke, JP Morgan Chase, Chicago, IL, USA
 - email: opdyke@acm.org

Dennis Mancl has been an internal consultant on OO design and agile development practices within Alcatel-Lucent and AT&T, with considerable experience in assisting software project teams with design patterns, requirements modeling, and reengineering existing software.

Steven Fraser joined Cisco Research in July 2007 as Director (Engineering). From 2005 to 2007, Steven was senior staff at QUALCOMM's Learning Center, with responsibilities for technical learning. From 2002 to 2004 Steven was an independent software consultant on tech transfer and disruptive technologies. Previous to 2002, Steven held a variety of software roles at Nortel and BNR including: Process Architect, Senior Manager (Disruptive Technology and Global External Research) and Software Reuse Program Prime. In 1994 he spent a year as a Visiting Scientist at the Software Engineering Institute (SEI). Steven holds a Doctorate in Electrical Engineering from McGill University in Montreal, Canada. Steven is a Senior Member of the IEEE and a member of the ACM. His work focuses on alleviating the "soft" challenges of developing and deploying products highly dependent on software.

Bill Opdyke has (throughout much of his career) worked in software and systems architecture roles. At JP Morgan Chase, he is involved in architecture issues for large financial systems. At Motorola, he was a distinguished member of technical staff, focusing on Information Assurance/ security and software architectures in support of government and public safety application development. He was also a technical lead on several advanced development projects that applied agile techniques at Bell Labs, and he spent several years teaching software engineering and project management courses (while a faculty member at North Central College). His doctoral research at the University of Illinois focused on object-oriented refactoring.

The workshop organizers have run nine workshops on various software engineering topics at previous OOPSLA and SPLASH conferences:

- Tackling the Discovery Costs of Evolving Software Systems (OOPSLA '02)
- Beyond Green-Field Software Development: Strategies for Reengineering and Evolution (OOPSLA '03)
- Challenges in Outsourcing and Global Development: How will your job change? (OOPSLA '04)
- Fostering Software Reliability in an Increasingly Hostile World (OOPSLA '05)
- Escaped from the Lab: Software Practices in Large Organizations (OOPSLA '06)
- No Silver Bullet – a Retrospective on Essence and Accident in Software Engineering (OOPSLA '07)
- Escaped from the Lab: Crossing the Gap from Invention to Practice (OOPSLA '08)
- Architecture in an Agile World (OOPSLA '09, SPLASH '10)

4. Anticipated Attendance

8-15 (maximum 20)

5. Advertisement

The workshop will be advertised by email to our personal networks of software professionals who are interested in software engineering and architecture issues, as well as to participants in previous OOPSLA/SPLASH workshops. Workshop details will be posted on the workshop website:
<http://mysite.verizon.net/dennis.mancl/splash11/index.html>

6. Participant Preparation

Pre-workshop readings will be posted on the workshop website – including a list of current articles relating to the workshop's topic. Workshop participants are requested to submit a short position paper in advance. This paper can include some discussion of the participant's experiences or some of the questions the participant wants to see addressed in the workshop.

7. Activities and Format

The workshop activities will be a series of brief position statements and experiences followed by a roundtable discussion. Tentative agenda:

- 8:30-9:30 -- short presentations of the position statements of the workshop participants
- 9:30-10:00 -- brainstorming session: gathering and prioritizing the main ideas and questions from the workshop participants

- 10:00-4:00 -- at least 3 iterative working sessions: based on the material prioritized in the brainstorming session
- [working lunch will take place somewhere in this interval]
- 4:00-5:00 -- wrap up: creation of a poster for the SPLASH poster session

The brainstorming and working sessions will be facilitated using a variety of techniques (NGT, Categorization, and others).

8. Post-workshop Activities

We plan to write a report and create a poster summarizing the most significant ideas shared and questions generated during the session. The poster will be displayed at the SPLASH poster session.

The report and poster will also appear on the workshop website after the conference:

<http://mysite.verizon.net/dennis.mancl/splash11/index.html>

9. Special Requirements

This is a relatively low-tech workshop – no special computer or video equipment will be required.