

Panel

Going Green with Refactoring: Sustaining the “World Wide Virtual Machine”

Steven Fraser
Cisco Research Center
San Jose
sdfraser@acm.org

Emerson Murphy-Hill
North Carolina State University
Raleigh
emerson@csc.ncsu.edu

Werner Wild
University of Innsbruck
Innsbruck
werner.wild@unibz.it

Joseph Yoder
The Refactory
Urbana-Champaign
joe@refactory.com

Bo Q. Zhu
Pattern Insight
Mountain View
bo.zhu@patterninsight.com

Abstract

This year's SPLASH theme is the "Internet as a World-Wide Virtual Machine" to highlight the challenges of continuous system growth, multi-site development, big data, with millions of interacting users and devices. This panel will discuss how refactoring systems can enable them to "go green" by reducing power consumption, increasing simplicity of interface, and/or extending system longevity in a world of constantly changing technology.

Categories and Subject Descriptors

K.0 Computing Milieux

General Terms Design, Experimentation, Standardization

Keywords Innovation, creativity, refactoring

1. Steven Fraser

STEVEN FRASER joined the Cisco Research Center as Director in July 2007 with responsibilities for fostering university research collaborations, managing PhD recruiting, and technology transfer. Prior to joining Cisco Research, Steven was a Senior Staff member of Qualcomm's Learning Center in San Diego, leading software learning programs and creating the corporation's internal technical conference (QTech Forum). Steven held a variety of technology strategy roles at BNR (Bell-Northern Research) and Nortel including: Process Architect, Senior Manager (Disruptive Technology and Global External Research), and

Advisor (Design Process Engineering). In 1994 he spent a year as a Visiting Scientist at the Software Engineering Institute (SEI) collaborating with the "Application of Software Models" project on the development of team-based domain analysis (software reuse) techniques. Fraser has organized multiple conference panels and was the Corporate Support Chair for OOPSLA'08 and OOPSLA'09. He was the Tutorial Chair for XP2008 and the Tutorial Co-Chair for ICSE'09. Fraser holds a doctorate in EE from McGill University in Montréal – and is a senior member of both the ACM and the IEEE.

2. Emerson Murphy-Hill

EMERSON MURPHY-HILL is an assistant professor at North Carolina State University. By conducting formative studies, building tools based on the findings, and then evaluating the effect that those tools have on software developers' work, his research aims to bridge the gap between the capabilities of tools and how software developers actually use them. He received his Ph.D. from Portland State University in 2009 and his B.S. from The Evergreen State College in 2001.

Emerson's research suggests that refactoring is a common practice among software developers, and that the types of refactorings that developers perform are also the types of refactorings that modern development environments offer. Although new kinds of refactoring tools may support developers in making new types of program transformations, this research suggests that developers are currently well supported by the existing tools.

Nonetheless, Emerson's research also suggests that existing refactoring tools are also underused -- perhaps 90% of refactorings that developers could do with modern refactoring tools are instead done manually. This is a problem because refactorings done by hand can be more error prone

and slower than those done with tools. One significant challenge, then, is to make refactoring tools that developers use frequently. Emerson believes this problem can be solved by improving the usability and discoverability of existing tools.

3. Werner Wild

WERNER WILD studied Computer Science and Mathematics at the University of Innsbruck and currently teaches at the Free University of Bolzano, the University of Innsbruck and the Management Center Innsbruck. Previous assignments include UNESCO, NIO Goa, ISS The Hague, UBS Switzerland, SwissRe Zurich, Joanneum Research Graz and others. His involvement with computers started 1972; he developed virtual machines, compilers, medical and financial applications, and is involved in the latest (agile) trends in Software Engineering. He is an organizer of workshops at international conferences, publishes together with Barbara Weber and is an elected official to the Austrian Chamber of Commerce in the Tyrol, identifying and tackling the challenges ahead of the Austrian IT industry. He loves to fly airplanes and holds a Commercial Pilot License in the US, including a current Instrument Rating.

Going "Green" is about sustainable solutions, from CSR (Corporate Social Responsibility) on the company level all the way down to the development of products consuming less energy per value unit delivered over the full product life cycle. This includes resources used during the development process, in the working product and when decommissioning. With the already hitting tidal wave of mobile devices the good old skills of considering code size and execution speed come to the forefront of desperately looked after skills, together with the new focus on reducing power consumption to extend useful battery life. In general, we fail as Computer Science educators to inspire the next generation of developers about the sustainability of their software and hardly any new engineer knows about the impact of a chosen implementation on maximum power used, life cycle energy consumption and its long term maintainability. However, teaching Refactoring for all kinds of "-ilities" (testability, changeability, scalability, ...) extends the students' mental software development model and gives them critical skills and tools needed to attack the issues mentioned above, right during development, not as an afterthought!

4. Joseph Yoder

JOSEPH YODER is a founder and principal of The Refactory, Inc., a company focused on software architecture, design, implementation, consulting and mentoring on all facets of software development. Joseph is an international speaker and pattern author, long standing member of the ACM, and the President of The Hillside Group, a group dedicated to improving the quality of software development. Joseph specializes in Architecture, Analysis and Design, C#, Java, Smalltalk, Patterns, Agile Methods,

Adaptable Systems, Refactoring, Reuse, and Frameworks. Joe is the author of many patterns, including being an author of the Big Ball of Mud pattern, which illuminates many fallacies in the approach to software architecture. Joe currently resides in Urbana, Illinois. He teaches Agile Methods, Design Patterns, Object Design, Refactoring, and Testing in industrial settings and mentors many developers on these concepts. He currently oversees a team of developers who have constructed many systems based on enterprise architecture in the .NET environment. Other projects involve working in both the Java and .NET environments deploying Domain-Specific Languages for clients.

Joe presents tutorials and talks, arranges workshops, and organizes leading technical conferences held throughout the world, including international conferences such as Agile, Agile Portugal, Encontro Ágil in Brazil, AOSD, CBSOft, JAOO, QCon, PLoP, AsianPLoP, SugarLoafPLoP in Brazil, OOPSLA, ECOOP, SATURN, and SPLASH. Joe thinks software is still too hard to change. He wants do something about this and believes that with good patterns and by putting the ability to change software into the hands of the people with the knowledge to change it seems to be on promising avenue to solve this problem.

Refactoring is a discipline approach to improve the overall design of the code. Refactoring is the process of changing software without altering its external behavior. Refactoring is usually done to make the code easier to extend or maintain and is done in the content of adding features to the software. A lot of small steps are taken to make the code better. Regression tests are critical to insure that the restructuring of the code did not break anything.

Refactoring software to make it maintainable and extendable has become a recognized best practice and has been accepted and incorporated into many Agile practices such as Scrum, TDD and XP. However, there are still many problems with the practice of refactoring. Quite often any type of redesign or evolution of code is considered "refactoring" the code.

Additionally there are many refactoring obstacles even in accepted Agile practices such as a rush to the next iteration and to add the next feature. It is important to support refactorings and include them as part of the regular development process as a lot of little steps before the code evolves to a state where refactoring can become more difficult. Knowing how and when to refactor is key towards being successful and also to reap the true benefits of keeping your code clean.

5. Bo Q. Zhu

BO Q. ZHU is Co-Founder and VP of Products at Pattern Insight. Bo is responsible for all products at Pattern Insight, from strategic planning to execution, through the entire product lifecycle. Pattern Insight is a VC backed enterprise software company that has developed a powerful data mining technology for analysis of source code to improve software quality, increase development productivity, and

reduce product cycle time. Code Insight, the company's flagship product, is currently used on tens of billions of lines of critical source code in industry leading companies such as Cisco, Qualcomm, Motorola, and Juniper. Bo obtained his Ph.D. in Computer Science from the University of Illinois at Urbana-Champaign. His interests lie in system research including using data mining with static code analysis to improve system reliability and manageability. He has published pioneering research papers in the top conferences such as SOSP and SIGMOD and holds several patents. He obtained his M.S. and B.S. in Computer Science from Nanjing University, China and previously worked at IBM Almaden Research Center and China's National Laboratory of New Software Technology.

Code reuse is a standard practice in modern software development. Practiced most commonly in the form of copy-paste programming, it has an inevitable side effect: Code bloat. Bloated code leads to poor code quality, increased software maintenance costs, as well as increased power drain on batteries in devices. In the mobile industry

in particular, device vendors have launched initiatives to pragmatically refactor codebases in an effort to combat quality problems and increase battery life. Step 1 in such refactoring processes is to accurately and efficiently identify candidate code clones. However, this is not an easy job. Studies show that 67% of code clones are not exact copies – with code often modified after reuse. Variable names are altered and statements are added deleted or modified. And this is not a matter of open source being the culprit: commercial code is as prone to this problem as open source. By combining data mining with static code analysis, Pattern Insight's Clone Detection solution quickly detects cloned patterns in code, even when code has been modified after reuse. It is fast, able to detect a greater quantity of clones than any other method (due to its fuzzy match capability), and easy to use. Our technology is used by the companies like Motorola and Qualcomm, that power a vast array of Android devices, to improve code quality, reduce bloat and maintenance and help make the world a bit more "green".