The Experimental Software Engineering Group: A Perspective

Victor R. Basili

Institute for Advanced Computer Studies
Department of Computer Science
University of Maryland

and

Fraunhofer Center - Maryland
Research Assumptions

• The study of software engineering is a laboratory science

• Understanding the discipline involves learning, i.e.,
  – observation
  – reflection, encapsulation of knowledge, model building
  – experimentation
  – model evolution over time

• The essential problem in software engineering is:
  – What is the appropriate process for developing a system with
    a specific set of properties given a set of constraints in a
    specific environment?

• Our essential question is:
  – Can empirical studies help address that problem?
Building a Basis for Empirical Study

• Can we measure and differentiate?
  – Can we measure and differentiate software products?
  – Can we empirically study the effects of processes?
  – Can we differentiate their effects, measure the differences?

• Can we improve the product and project?
  – Can we improve productivity and quality by manipulating process?

• Can we improve the process?
  – Can we use empirical studies to define improved processes and techniques?
  – Can we define techniques with different goals and empirically validate that they satisfy those goals?
  – Can a particular reading technique detect more of a particular class of defect than another reading technique?
Can we measure and differentiate?

• Questions of Interest
  – Can we measure and differentiate software products?
  – Can we empirically study the effects of processes?
  – Can we differentiate their effects, measure the differences?

• Study Context
  – University-based projects, student projects
  – Professional programmers

• Study methods
  – Individual Case Studies, Controlled Experiments

• UM Community
  – Joe Turner, Bob Reiter, Dave Hutchens, Rick Selby, Chris Lott, Oliver Laitenberger, Sivert Sorumgaard, ...
  – Filippo Lanubile, Adam Porter, Walcelio Melo, ...
Can we measure and differentiate?

Results

• Differentiated the effects of structured methods

• Developed and studied the effect of Iterative Enhancement Method

• Classified Experiments and Experimental Designs

• Differentiated the effects of testing vs. reading techniques

• Differentiated the effects of Mill’s Cleanroom methodology
Can we improve the product?

• Questions of Interest
  – Can we improve productivity and quality by manipulating process empirically study the effects of processes?

• Study Context
  – NASA/GSFC flight software projects (SEL)

• Study methods
  – Controlled Experiments, Case Studies, Multiple Projects
  – Qualitative Analysis

• UM/NASA/CSC Community
  – Dave Weiss, John Bailey, Rick Selby, Lionel Briand, Bill Thomas, Carolyn Seaman, Manoel Mendonca, Daniil Yakimovitch, Mike Stark, ...
  – Marvin Zelkowitz, Frank McGarry, Bill Agresti, Dieter Rombach, Rose Pajerski, Jon Valett, Scott Green, Gianluigi Caldiera, ...
Can we improve the product?

Results

• Developed the **Goal/Question/Metric Paradigm**

• Developed the **Quality Improvement Paradigm**

• Developed the **Experience Factory Organization**

• Demonstrated continuous improvement in the NASA/Software Engineering Laboratory
  – Decreased **Development Defect rates** by
  – **75%** (87 - 91) **37%** (91 - 95)
  – Reduced **Cost** by
  – **55%** (87 - 91) **42%** (91 - 95)
  – Improved **Reuse** by
  – **300%** (87 - 91) **8%** (91 - 95)
Can we improve the process?

• Questions of Interest
  – Can we use empirical studies to define improved process and techniques?
  – Can we define techniques with different goals and empirically validate that they satisfy those goals?
  – Can a particular reading technique detect more of a particular class of defect than another reading technique?

• Study Context
  – Industry, University, Individuals, ….

• Study methods
  – controlled experiments, case studies, observational studies

• Community
  – Forrest Shull, Zhijun Zhang, Jeff Carver, …
  – Guilherme Travassos, Ben Shneiderman, …
Can we improve the process?

Results

• Developed five families of reading techniques
  – parameterized for use in different contexts and
  – evaluated experimentally in those contexts

• Several have evolved based on various forms of empirical studies
  – Perspective Based Reading (PBR):
  – for detecting defects in requirements documents in English
  – Object Oriented Reading:
  – for detecting defects in object oriented design in UML
Can we improve the process?

Results

- Developed an approach for combining the results of several experiments to build our knowledge about software processes
  - We can **effectively design and study techniques** that are procedurally defined, document and notation specific, goal driven, and empirically validated for use
  - We can demonstrate that a **procedural approach** to a software engineering task could be more effective than a less procedural one under certain conditions (e.g., depends on experience)
  - A procedural approach to reading based upon **specific goals** will find defects related to those goals, so reading can tailored to the environment
Changes in Empirical Study Components

• Study Context
  – University Projects, Industry, Government, International Organizations

• Study methods
  – controlled experiments, case studies, structured interviews, observational studies

• Analysis Methods
  – correlations, regressions --> pattern recognition models
  – quantitative analysis --> qualitative analysis

• Measurement
  – interval and ratio, nominal and ordinal
  – characterize, evaluate, predict, control

• Study Goals
  – effect, feasibility, improvement
Building a Community

• We have created an environment in which
  – Many students worked on some aspect of the problem, building on the work of others
  – Many visitors brought and left with expertise
    • Ross Jeffrey, Giovanni Cantone, Markku Oivo, Sandro Morasca, Filippo Lanubile, Maurizio Morisio, Reider Conradi, ...

• Interchange occurred through (almost) weekly group meetings

• which have lead to international collaborations (ISERN)

• the Journal on Empirical Software Engineering (JESE)

• and the Fraunhofer Center for Experimental Software Engineering