Challenges in Software Evolution

Tom Mens

http://w3.umh.ac.be/genlog
Software Engineering Lab
University of Mons-Hainaut
Belgium

Challenges in Software Evolution

- The presented results are the outcome of the ChaSE 2005 workshop
  - Financed by ESF and ERCIM
  - April 2005, Bern, Switzerland

- Scientific goals
  - to discuss about and identify the main challenges in software evolution
  - to address the above goal from different points of view

European Research Consortium for Informatics and Mathematics
ERCIM

Classification of challenges

- Multidimensional classification
  - Time dimension
    - Short-term, medium-term, long-term research
  - Type of software evolution research
    - Managing software evolution
    - Understanding software evolution
    - Analysing software evolution
  - Interested stakeholder
    - Manager, end-user, developer, teacher, tool builder, software engineer, …
  - Type of artifact under study
    - Formalism, tool, model, language, process, people, …
  - Type of support provided
    - Same list as before…

Preserving and improving sw quality

- How can we provide tools and techniques that preserve or even improve the software quality, whatever its size, complexity, level of abstraction?

Michel Chaudron
Peter Ebraert
Kim Mens

time	research type	stakeholder	artifact	support
long	Controlling & supporting evolution	developer, project manager, end-user	Software system	tools, techniques, formalisms, processes

Supporting model evolution

Christian Lange

- Design and modeling tools provide little evolution support
- evolution techniques needed at higher level of abstraction
  - A&D models, SW architectures, requirements, …
- Model-driven engineering makes this challenge very relevant

time	research type	stakeholder	artifact	support
short	controlling, supporting	software engineer	models	tools, techniques, formalisms

Supporting co-evolution

Christian Lange,
Dennis Van Opzeeland

- We urgently need better techniques to achieve co-evolution
  - Synchronisation, consistency maintenance, inconsistency management, traceability, change propagation, …
- between different types of software artifacts or different representations
  - Programs and design models
  - Software and the organisation
  - Software and languages, tools, platforms

time	research type	stakeholder	artifact	support
medium	controlling, supporting	software engineer	any pair of related artifacts	tools, techniques, formalisms

Formal support for evolution

Kim Mens,
Tom Mens

- Some formal methods not amenable for an evolutionary setting
  - e.g., no incremental verification
- Formalisms for specific evolution activities needed
  - e.g., for refactoring

time	research type	stakeholder	artifact	support
medium - long	all	researcher	formalisms	formalisms

Need for empirical research

- empirical and statistical studies needed to assess impact of process models, tools, languages, and people's experience on software evolution

Christian Lange
Dennis Van Opzeeland

time	research type	stakeholder	artifact	support
long	analysing	researcher	any evolving artifact	statistical models, empirical studies

Need for evolution benchmarks

Kim Mens

- Commonly agreed representative benchmark or case studies to compare the developed formalisms, tools, techniques on relevant and typical problems
- Getting data from industrial setting is not easy, but there are many open-source, long-lived, industrial size projects

time	research type	stakeholder	artifact	support
short - medium	understanding, comparing	researcher	software system	benchmarks, exemplars, cases

Runtime evolution

Bram Adams

- There is a need for proper support of runtime adaptations that allow systems to evolve while they are running, without needing to pause them or shut them down
  - Reflective techniques, metadata

time	research type	stakeholder	artifact	support
short - medium	Controlling, supporting	developer, end-user	languages, execution platforms	languages, execution platforms, programs

Challenge: Teaching software evolution

- How can we introduce the ideas and techniques of evolution into our educational system?
  - What do we want to teach to our students?
  - How can we teach this?
  - Where does it fit in the CS curriculum?

time	research type	stakeholder	artifact	support
short	teaching	teacher, student	any	slides, exercices, case studies, tools, books, …

A common software evolution platform

- Proposed research solutions need to scale up to long-lived industrial-size software system
- Required tools are too complex to be built in isolation
- Need for a common platform, tool integration, exchange formats, standards and so on
- Candidates: MOOSE, Eclipse

time	research type	stakeholder	artifact	support
medium	Applied research	researcher	programs	tools, frameworks, platforms, standards, exchange formats

Evolution as a language construct

- Change should be a first-class entity in programming or modelling languages
- Evolution support is easier in dynamically typed languages with reflective capabilities

time	research type	stakeholder	artifact	support
short - medium	controlling, supporting	language designer, tool builder, researcher	languages	languages and programs

Supporting multi-language systems

- Many complex and large systems are built using 3 or more languages
- Evolution techniques should be parameterisable on or independent of the language

time	research type	stakeholder	artifact	support
medium - long	controlling, supporting	tool builder	languages, software systems	tools, standards

Integrating change into dev. process

- Change must be integrated into conventional development process models
- Some, like agile development, already embrace change as essential
- Others, like the staged life-cycle model, have explicit support for evolution

time	research type	stakeholder	artifact	support
medium	managing, controlling	manager, software engineer	software process model	software process model

Increasing manager's awareness

- Managers have to realise the importance and inevitability of software evolution
- Teach them how to plan, organise and control projects to cope with change

time	research type	stakeholder	artifact	support
short	motivating	manager, researcher		metaphors

Developing better versioning systems

- Many analyses of software evolution based on CVS or related tools
  - these weren't built for that purpose and don't store enough information
- New techniques and tools needed for recording the evolution of a system
- SCM is related

time	research type	stakeholder	artifact	support
short	analysing	tool builder	version control tools	tools

Integrating and analysing evolutionary data

- Scattered information about system changes
  - bug reports, source code, documentation, configuration files, …
- very large data sets, especially for long-lived systems
- Need efficient and heterogeneous techniques
- Extensible meta-models, data mining, and bioinformatics may be relevant

time	research type	stakeholder	artifact	support
medium	analysing	researcher	all relevant info about sw system's evolution	techniques, tools

A theory of software evolution

- Lehman has developed laws, but they need to be formalised and enriched
- May borrow ideas from evolutionary biology or linguistic evolution
- The what (noun) and how (verb) of evolution are still mostly disconnected
- How does the gathered data inform tools, techniques and formalisms?

time	research type	stakeholder	artifact	support
long	understanding, analysing	researcher	everything	theories, formalisms, laws, …