Software Quality Research: from Processes to Model-based Techniques

Bernhard Peischl
Softnet Austria

17th April 2015

From Processes to Model-based Techniques

- Internet of Things (IoT)
    - Digital economy
    - Features, functions, user interface solutions
    - User-centred elicitation of requirements
- Process-based software quality is gaining momentum
- Scientific disciplines: computer science, discrete mathematics, operations research, statistics, psychology, …, and economics

Model-based Engineering (MBE)

- Product-based evidence of software quality
- Productivity gains
- Model-based Development (MDD)
    - Code generation
    - Testing (MBT)
- Functional and partial views of the System under Test (SUT)
- Implementation-centred view
- HOW to perform a task rather than WHAT the task should achieve

Model-based Systems (MBS)

- Software quality tools that implement the HOW itself
- Tasks
  - Simulation
  - Prediction
  - Reasoning about the software product
- Prominent applications
  - Physics
  - Electrical engineering
  - Most recently in the field of software

Process-based software quality - Experience

- Bug and issue tracking for a vendor of simulation tools [2]
- User-centred software development in healthcare [3]
- Structured test case design for railway interlocking systems [1]

Model-based systems - Experience

- Facilitating the use of effort estimation techniques for end-users [4]
- Configuration of embedded software for electronic control units [5]

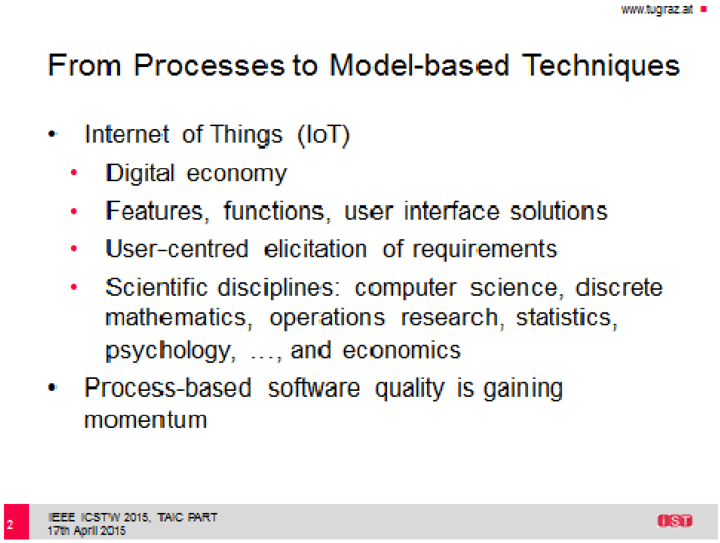
Model-based Testing - Experience

- Test case generation in the automotive industry [7, 8]
- Test case generation for VoIP telephony [6]
- Smart monkey testing with capture & replay tools [9]

Lessons Learnt

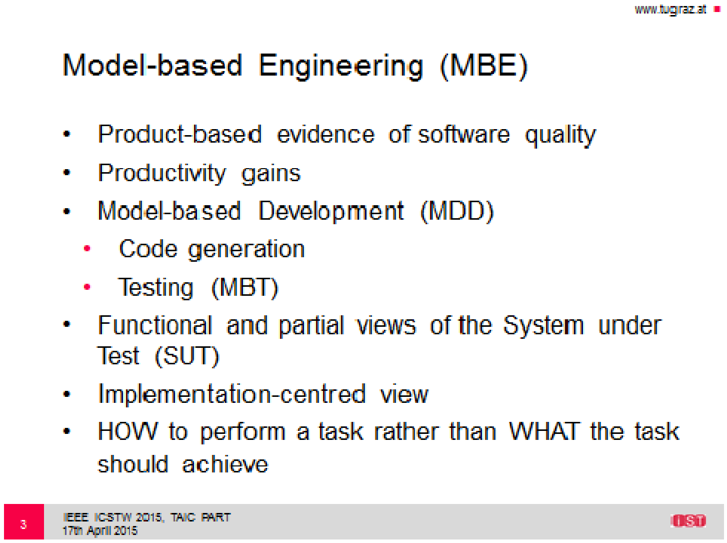
- The domain knowledge matters
- Simple and intuitive modelling is key to success
- Short feedback cycles inspire practical use
- Constraints, processes and human factors are essential for tool prototyping
- Pilot projects are more convincing than benchmarks
- A good solutions needs a lead user
- Generalization is the task of academia

Discussion



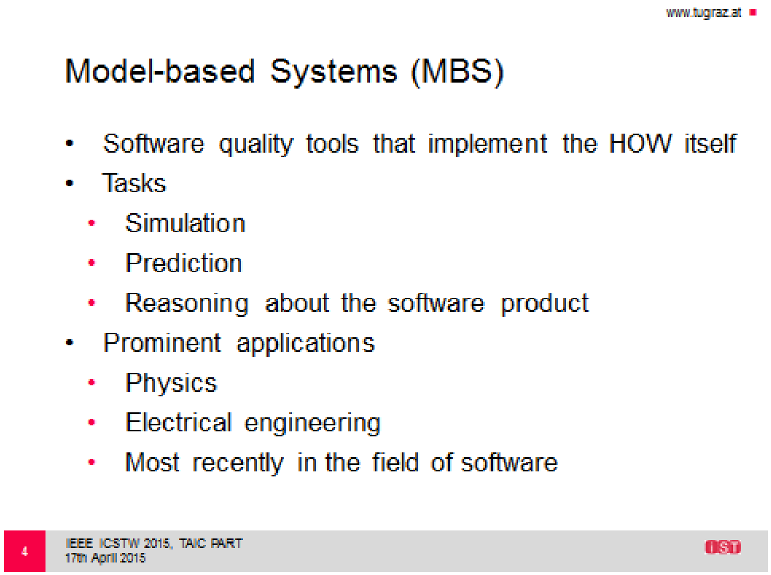
From Processes to Model-based Techniques

- Internet of Things (IoT)
  - Digital economy
  - Features, functions, user interface solutions
  - User-centred elicitation of requirements
  - Scientific disciplines: computer science, discrete mathematics, operations research, statistics, psychology, …, and economics
- Process-based software quality is gaining momentum



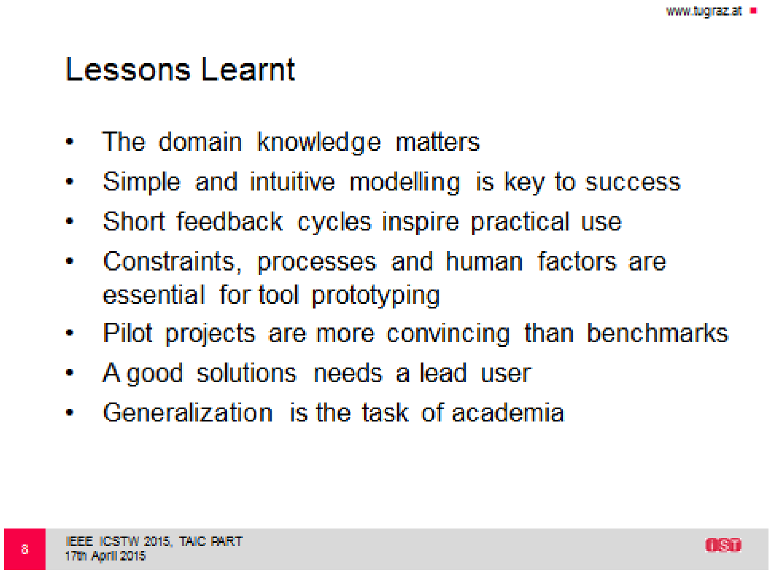
Model-based Engineering (MBE)

- Product-based evidence of software quality
- Productivity gains
- Model-based Development (MDD)
  - Code generation
  - Testing (MBT)
- Functional and partial views of the System under Test (SUT)
- Implementation-centred view
- HOW to perform a task rather than WHAT the task should achieve



Model-based Systems (MBS)

- Software quality tools that implement the HOW itself
- Tasks
  - Simulation
  - Prediction
  - Reasoning about the software product
- Prominent applications
  - Physics
  - Electrical engineering
  - Most recently in the field of software



Lessons Learnt

- The domain knowledge matters
- Simple and intuitive modelling is key to success
- Short feedback cycles inspire practical use
- Constraints, processes and human factors are essential for tool prototyping
- Pilot projects are more convincing than benchmarks
- A good solutions needs a lead user
- Generalization is the task of academia

Contact

Bernhard Peischl
Softnet Austria
Institute for Software Technology
Graz University of Technology
mailto: bernhard.peischl@soft-net.at

References

[1] A. Beer and B. Peischl, “Testing of safety-critical systems a structural approach to test case design,” in Advances in Systems Safety, C. Dale and T. Anderson, Eds. Springer London, 2011, pp. 187–211.

[2] A. Kalchauer, S. Lang, B. Peischl, and V. Rodela Torrents, “Business intelligence in software quality monitoring: Experiences and lessons learnt from an industrial case study,” in Software Quality. Model-Based Approaches for Advanced Software and Systems Engineering, ser. Lecture Notes in Business Information Processing, D. Winkler, S. Biffl, and J. Bergsmann, Eds. Springer International Publishing, 2014, vol.166, pp. 34–47.

[3] B. Peischl, M. Ferk, and A. Holzinger, “The fine art of user-centered software development,” Software Quality Journal, pp. 1–28, 2014.

[4] M. Nica, B. Peischl, and F. Wotawa, “Constraint-based configuration of embedded automotive software,” International Journal of Mass Customisation, vol. 3, no. 4, pp. 368–388, 2010.

[5] B. Peischl, M. Zanker, M. Nica, and W. Schmid, “Constraint-based recommendation for software project effort estimation,” Journal of Emerging Technologies in Web Intelligence, vol. 2, no. 4, 2010.

[6] B. Aichernig, B. Peischl, M. Weiglhofer, and F. Wotawa, “Protocol conformance testing a sip registrar: an industrial application of formal methods,” in Software Engineering and Formal Methods, 2007. SEFM

2007. Fifth IEEE International Conference on, Sept 2007, pp. 215–226.

[7] C. Schwarzl and B. Peischl, “Static- and dynamic consistency analysis of uml state chart models,” in Model Driven Engineering Languages and Systems, ser. Lecture Notes in Computer Science, D. C. Petriu,N. Rouquette, and Ø. Haugen, Eds. Springer Berlin Heidelberg, 2010,vol. 6394, pp. 151–165.

[8] C. Schwarzl and B. Peischl, “Test sequence generation from communicating uml state charts: An industrial application of symbolic transition systems,” in Quality Software (QSIC), 2010 10th International Conference on, July 2010, pp. 122–131.

[9] B. Hofer, B. Peischl, and F. Wotawa, “Gui savvy end-to-end testing with smart monkeys,” in Automation of Software Test, 2009. AST ’09. ICSE Workshop on, May 2009, pp. 130–137.