Computer Science Day 2008

Object-Oriented Software Systems

Henrik Bærbak Christensen
The OOSS Group Today

Central characteristics
– Software Systems
  • Software architecture, software engineering
– Empirical and Experimental Research Approach
  • Industrial Collaboration

Long history...
– Programming Language Design and OO
– Dev. Environment Design

http://www.ooss.dk
In- and out flux...
Two Projects Presented Today

**SA@Work: Software Architecture at Work**
- Empirical approach, studying *real* architects doing *real* architectures
  - Are we having the proper text-books?

---

**Flexible, Reliable Software**
- Constructive approach, a teaching book focusing on *real* programming of *real* architectures
  - *This is the proper type of text-book 😊!*
In the Natural Science Cycle

SA@Work: Software Architecture at Work
- Empirical approach, studying real architects doing real architectures
  - Are we having the proper text-books?

Flexible, Reliable Software
- Constructive approach, a teaching book focusing on real programming processes
  - This is the proper type of text-book ☺
SA@Work
Software Architecture in Practice in Danish Companies
Motivation

Software systems papers 😊
- Many ”Use my idea and get happier”
- Few ”What ideas are used and are they happy?”

Motivation: *Study the tribe rituals in the jungle!*

The ”jungles” are:

Cover a broad class of characteristics
- Large – small companies
- Products versus Projects
- Domains
- IT Platforms
- IT as Business versus IT as part of Business
Phases

Phase 1: Ethnographical Field Studies

- Focus: Architectural techniques, Arch. prototyping

Phase 2: Action Research (Ongoing)

- aSQA: Technique to formalize architectural quality
  - developed by Systematic Software Engineering
- AMM: Architectural Maturity Model
Initial Results

Phase 1:
- Classification of Architectural Techniques in general
- Study of Architectural Prototypes in particular
- Research: WICSA08, ECSA08, APSEC08, ICSE09, ...
- Industry: Confidential report, workshops, ERFA days

Phase 2:
- aSQA: not the “inventor” but “formalizer” and “verifier”
  - Describe technique + test it in B&O and Jyske Bank
- AMM: Close collaboration with SEI
- Research: ..., (collab. with SEI 😊)
- Industry: Concrete techniques, improvement initiatives
Flexible, Reliable Software
Realized two big challenges in (my) teaching

– *Doing it* is more than reciting the definitions at exam
  - *colour theory, anatomy knowledge, art history are important topics for a painter but they do not make you a great painter...*

  - Most software engineering books simply recite topics but do **not** show the process nor how each aspect relate to each other and to the complex task of large scale development...

– Paradigm for design: OOAD versus Design Patterns
  - *model of the real world* did not explain patterns 😞
The “Solution”

Paradigm: Compositional Software Design
– Applying 3 principles! Patterns are derived.

Make Programming Process Teachable
– Pattern language for the process itself
  • Test driven development / Agile process

Story Telling and Problem-based learning
– Techniques introduced in response to concrete problem

Spiral of increasing architectural complexity
– Reliability is improved by Test-Driven Development require Test Stubs that induce Variability Management supporting Flexibility implemented by Design Patterns derived from Principles of Flexible Design forming the Compositional Design Paradigm underlying Framework and Product-line Architectures supported by …
Steps in Responding to the Challenges

Research Papers


University Courses

– dSoftArk (Full time), 01-?
– PaSOOS (Part time), 99-?

Industry Courses+Talks


Kursus: Reliable and Flexible Software Explained: Test-driven Development, Patterns and Frameworks.

Kursus: Introduction to Design Patterns and UML.

Book Chapters

Michael E. Caspersen and Henrik Bærbak Christensen: CS1: Getting Started

Henrik Bærbak Christensen: Experiences with a Focus on Test in Teaching

Henrik Bærbak Christensen and Michael E. Caspersen: Frameworks and their Role in Teaching
Master/PhD in O OSS?

If you like to

– interested in large and complex software systems
– study topics highly relevant in Industry
  • to the degree of being buzzword compliant 😊
– study with a constructive, empirical focus
Questions?