

# Software Variability Management

**Jan Bosch**

*Professor of Software Engineering  
University of Groningen, Netherlands*

*[Jan@JanBosch.com](mailto:Jan@JanBosch.com)*

*<http://www.janbosch.com>*

## Background

- Head of Software Engineering group at University of Groningen - earlier head of RISE research group @ Blekinge Institute of Technology
- Dutch National Architecture Congress LAC 1999, 2000, 2001, 2002, 2003 (member organizing committee)
- general chair WICSA4, Oslo, June 2004
- IFIP/IEEE WICSA 4, Montreal, 2002 (programme co-chair)
- Software Product Line Conferences (SPLC's)
- EUREKA/ITEA ESAPS and CAFÉ projects, 300 person-years each, 99-01 and 01 -03, FAMILIES (03-05)
- EU FP5 IST projects:
  - STATUS – Software Architecture for Usability
  - ConIPF – Configuration of Industrial Product Families
  - Software variability management workshops and tutorials
  - Special issues on SVM in Science of Computer Programming and Software Process Improvement in Practice journals

# Software Engineering group

## Senior members

- prof. dr. ir. Jan Bosch
- dr. Jos Nijhuis (APr)
- dr. Rein Smedinga (APr)

## Ph.D. students

- Michel Jaring (2004)
- Eelke Folmer (2005)
- Sybren Deelstra (2006)
- Marco Sinnema (2006)
- Anton Jansen (2006)
- Ivor Bosloper (2007)
- Johanneke Siljee (2007)
- Jan Salvador van der Ven (2008)

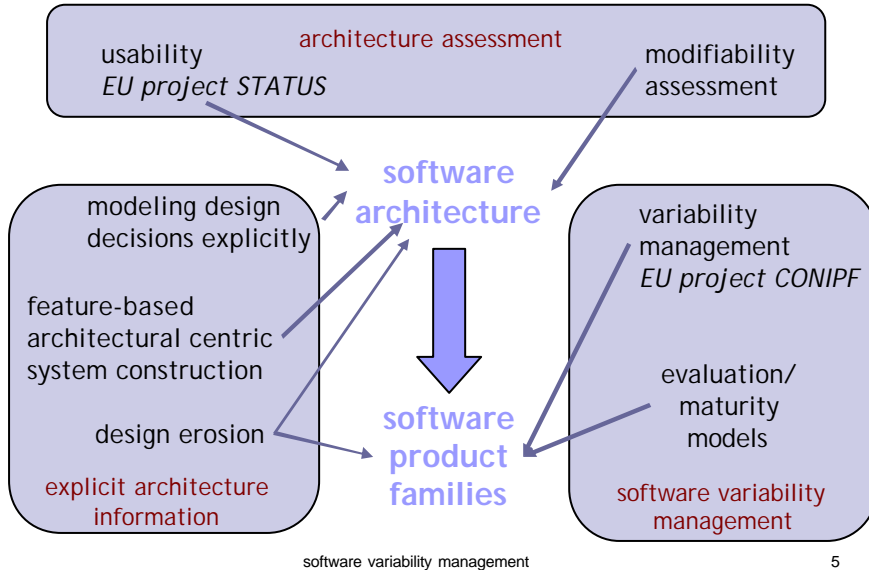
## Industrial Ph.D. students

- Rob van Ommering (Philips Research, 2004)
- Jan Gerben Wijnstra (Philips Research, 2004)
- Alessandro Maccari (Nokia Research/Mobile Phones)
- Ernst Kessler (Dutch Aerospace Laboratory)
- Sylvia Stuurman (Open Universiteit)

# Industrial Partners

- Ericsson Software Technology
- Althin Medical
- Axis Communications
- Symbian (earlier Ericsson Mobile)
- EC-Gruppen
- Securitas Larm
- Ericsson Software Architecture Research Lab
- Nokia
- CombiTech Software
- Vertis Information Technology
- Rohill Technologies  
currently ongoing cooperation
- Philips Research, Medical, Consumer Electronics and PDSL
- Baan
- Thales Naval Netherlands
- Robert Bosch GmbH
- Information Highway Group
- LogicDIS
- ICT Embedded

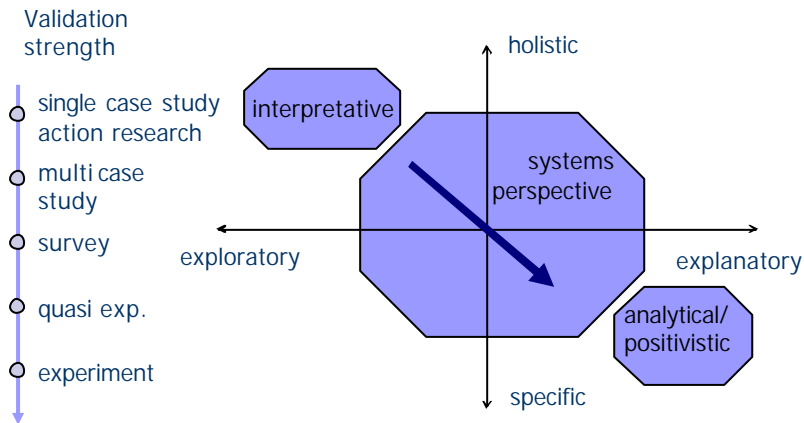
# Current Research Topics



# Research Topics

- software architecture design
  - software architecture assessment, especially *maintainability* and *usability*
  - design decisions for quality attributes
  - design for usability
- software product lines
  - variability management
  - organizational issues
  - scoping and business case analysis
- component-based software engineering

# Research Perspective

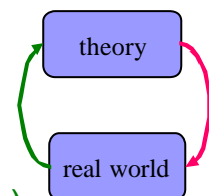


software variability management

7

# Research Methods

- analytical methods
  - formal models, theories, proofs, etc.
- empirical methods
  - experiment (quantitative)
  - survey (quantitative)
  - case study (qualitative or quantitative)
  - action research (qualitative)



software variability management

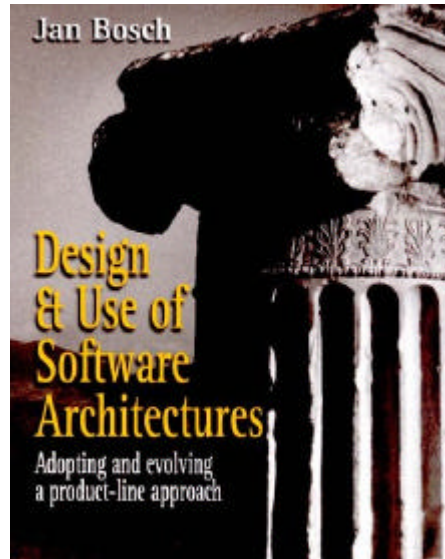
8

# Book!

Pearson Education

ISBN

0-201-67494-7



software variability management

9

# Overview

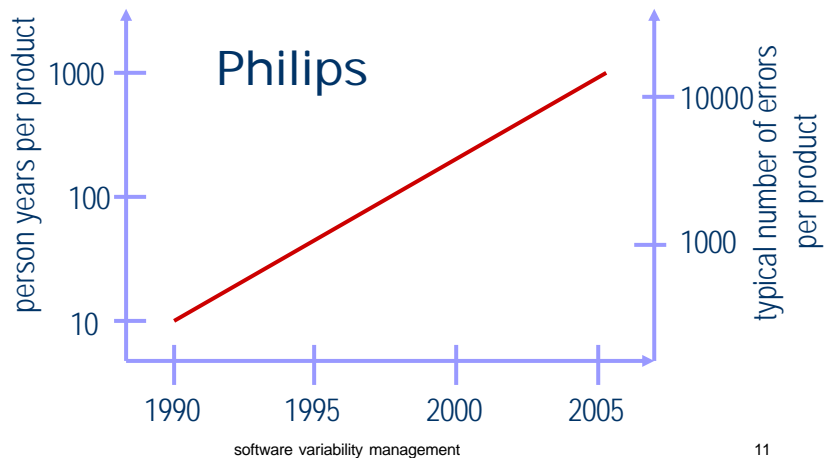
- role of software
- software reuse
- software components
- software architecture
- software product lines
- conclusion

software variability management

10

## Trend: software size

- software needs in products constantly increasing



11

## Role of software

- software constantly increasing in size and complexity
- software makes up *competitive edge*
- examples
  - Securitas Larm AB - fire alarm systems
  - business process reengineering and corporate information systems



## Trend: systems of systems

systems increasingly need to be integrated with other systems

- Information systems
  - from manual data exchange to behavioural integration
- embedded/technical systems
  - from stand-alone to signal exchange to behavioural integration



## Trend: Variability

Variability needs in software are constantly increasing

because

- variability moves from mechanics and hardware to software
- design decisions are delayed as long as economically feasible

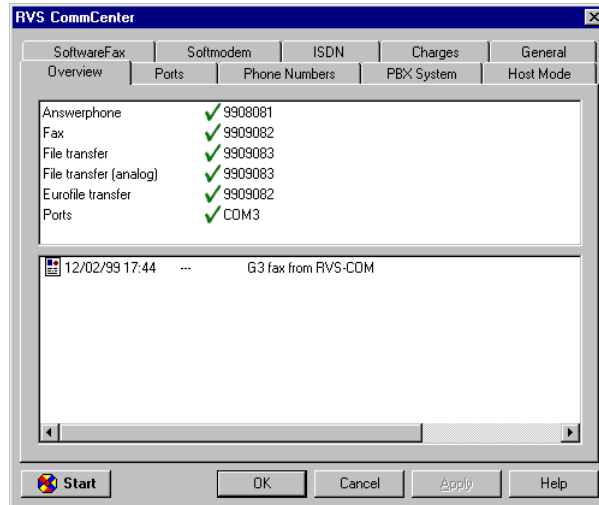
# Examples

- car engine controllers
- steer by wire
- telecom switches
- Transmeta's Crusoe chip

# Mobile Phones



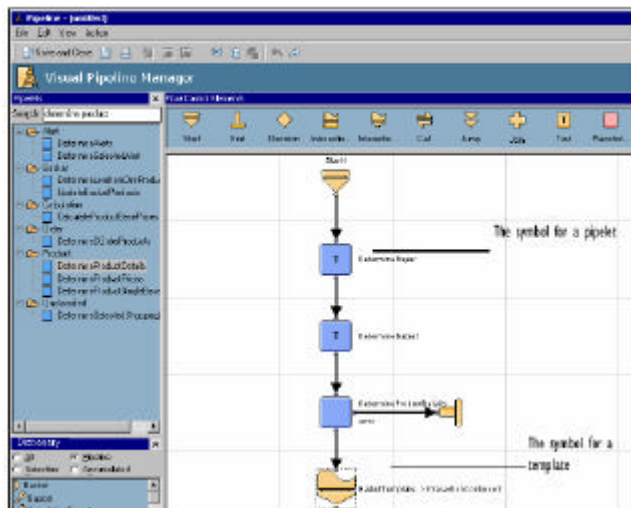
# Configuration Wizard



software variability management

17

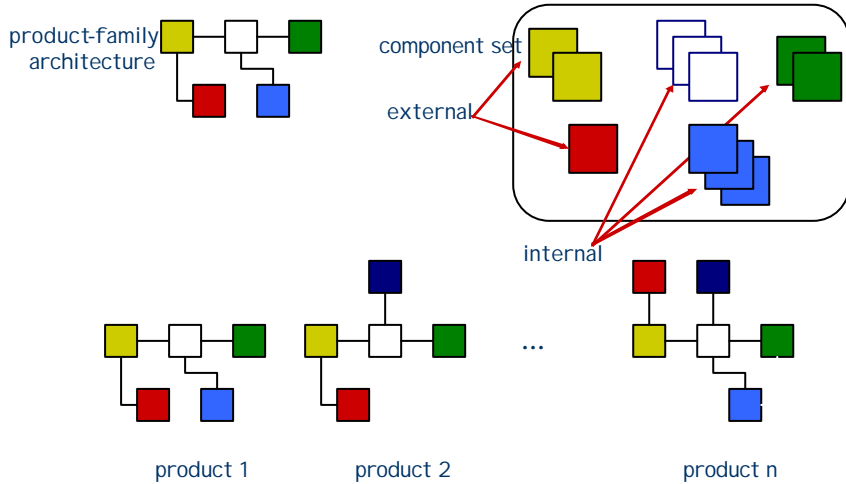
# BPM @ Intershop



software variability management

18

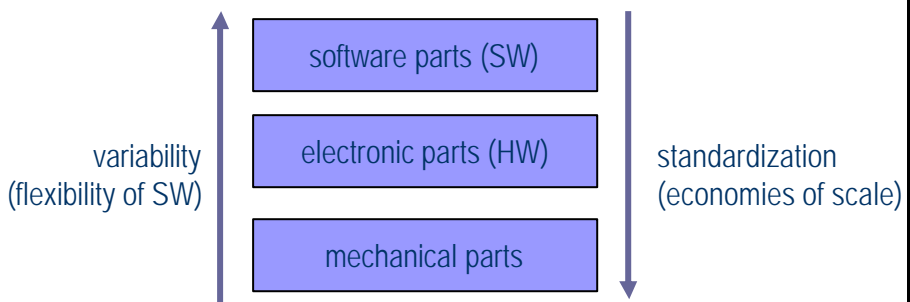
# Software Product Families



software variability management

19

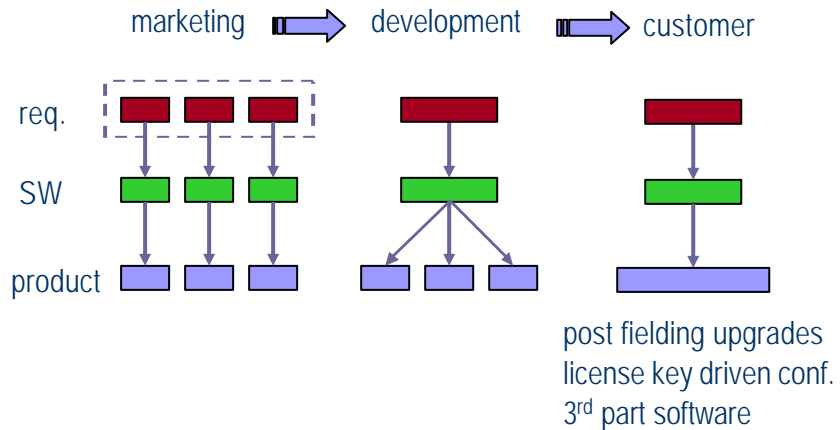
# Trend: Variability



software variability management

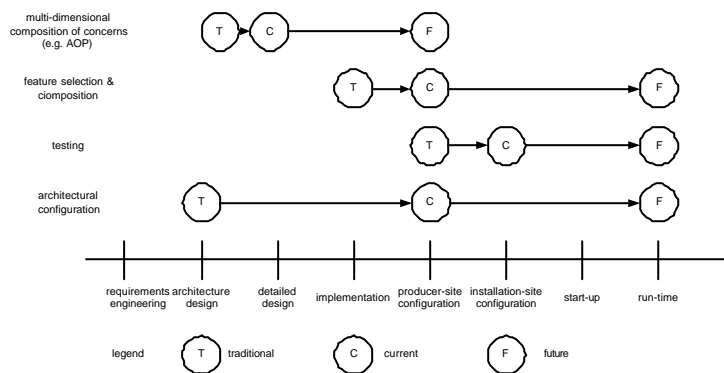
20

# Trend: Variability

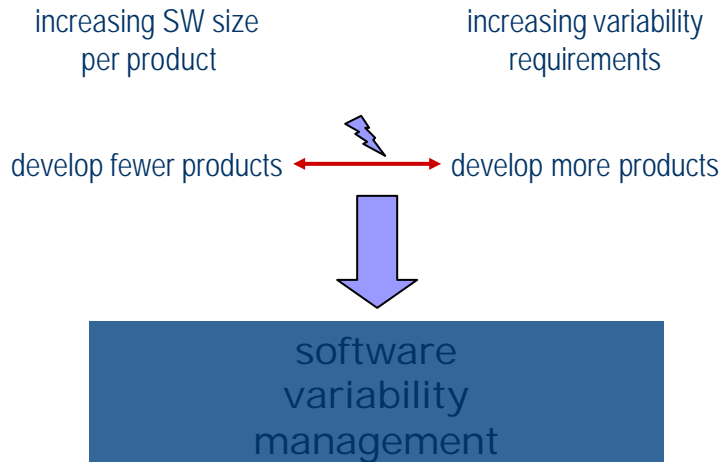


# Later Binding

- trend is towards later binding and increased automation



# Software Reuse



software variability management

23

# Software Reuse

- drastically decrease cost of software development and maintenance
- increase quality of software
- reuse of existing software is one of the most promising approaches
- construct applications by composing reusable software pieces

(cost ⇔ TTM ⇔ staff)

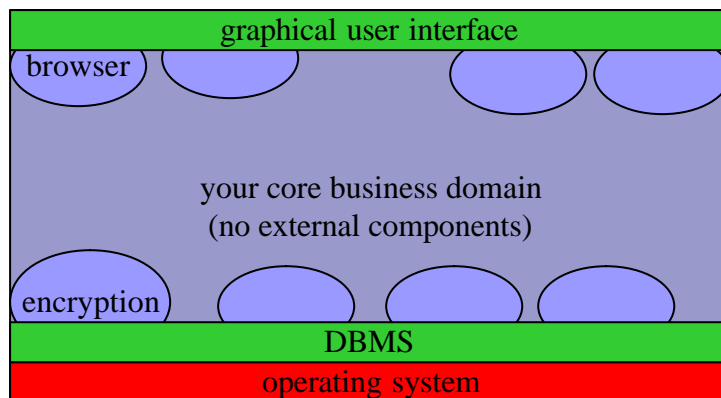
software variability management

24

# Examples of reuse

- code “scavenging” (copy & paste)
- operating systems
- compilers
- class libraries
- design patterns
- distribution systems, e.g., CORBA, COM, etc.
- graphical user interface (GUI) systems
- design reuse (expertise)
- object-oriented frameworks
- database management systems (DBMS)
- application generators

# Internal vs. External



# External Component Lifecycle

- part of application
- identified as domain by visionaries
- prototype development (universities, research labs)
- productification (often by small (new) companies)
- adoption by the “pragmatists”
- incorporation in “infrastructure”, e.g. operating system

# Software Architecture

- software architecture

*Architecture is the fundamental organization of a system embodied in its components, their relationships to each other and to the environment and the principles guiding its design and evolution.*  
[IEEE Standard P1471]

- Definition contains two main parts:
  - structural part – components and connectors
  - principles part – principles, design rules, design constraints, etc.

# Software Architecture and QAs

- the architecture of a software system constrains the quality attributes:
  - performance
  - maintainability
  - reliability
  - ...
- therefore, the driving quality attributes influence the software architecture (more than the functional requirements)

# Why software architecture?

first class representation of SA during

- design
  - stakeholder-based assessment
  - assessment of quality attributes
- implementation
  - configuration management
  - software product lines
- run-time
  - dynamic software architectures, post-fielding upgrades, dynamic reorganization, etc.

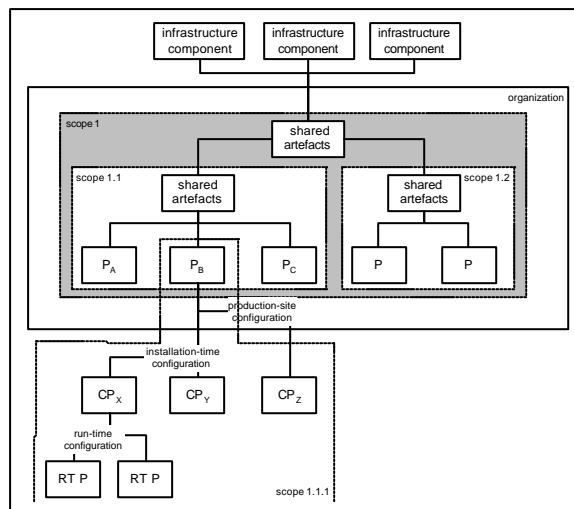
# Software Architecture Use

Three types of SA use:

- single system
- software product family
- standardized domain architecture component framework [szyperski 98]



## Example



# Conclusion

- software variability management and software reuse are intimately linked
- software variability management is evolving into a key challenge
- software variability management spans over multiple scopes of reuse and evolves over time

# Conclusion

- introduction to
  - software architecture
  - software components
  - software variability
- trends
  - size of software in systems constantly increasing
  - amount of variability increases, binding phase is delayed
  - move from rigid control (strict system paradigm) to mutual coordination and adaptation (loose system/ecology paradigm)
  - development is the exception, derivation and evolution are the norm