How to Make PhD in Software Engineering

Kai Koskimies
Introduction to Software Engineering Research
Spring 2008

Lecture Program

4.2	Introduction (Kai Koskimies)
4.2	Design science (Matti Rossi)
11.2	Action research and interviews (Tomi Männistö ja Marjo Kauppinen)
18.2	*How to make PhD in Software Engineering (Kai Koskimies)*
25.2	*Paper introductions (at 12-14)*
25.2	Model-driven testing (Mika Katara)
3.3	Model-driven design (Ivan Porres)
10.3	*How to make papers in Software Engineering (Kai Koskimies)*
17.3	Model-driven web services (Tarja Systä)
1.4	Empirical software engineering research (Claes Wohlin) ?
14.4	Architectures for user interfaces (Roope Raisamo)
28.4	Agile methods (Pekka Abrahamsson)
12.5	Domain-specific languages (Juha-Pekka Tolvanen)

Topics

Software engineering as research area
Purpose of PhD thesis
Finding topic/focus/viewpoint
Contribution: PhD styles
Different "schools" of SE research
Making PhD in industry
Conceptualization vs. tool development
Monograph vs. set of papers
Elements of PhD
Evaluation
Case studies

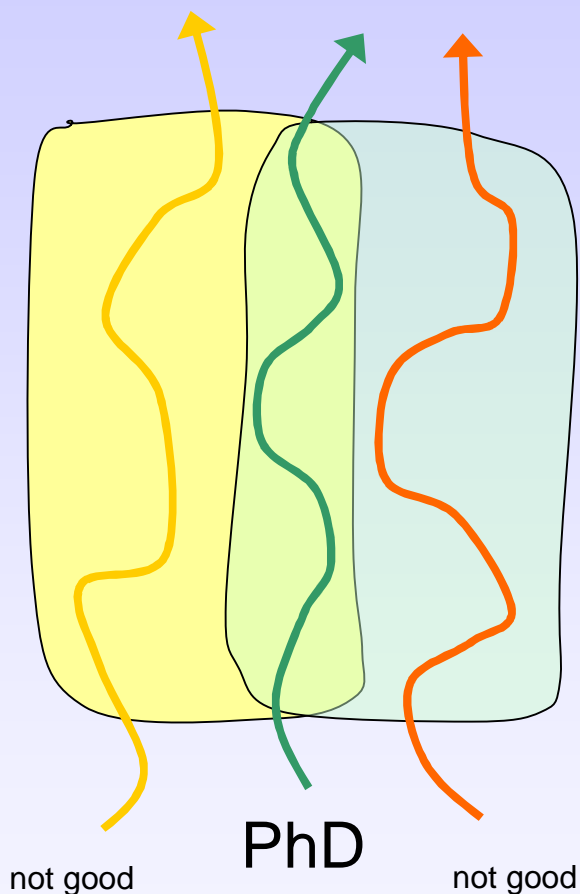# Software engineering as a research area

- software engineering world is a world of artifacts & people
- evolutionary laws
- hypes are typical in artifactual worlds
- strong social flavor
- rapidly evolving: problem areas become outdated
- practically oriented
- results should be relevant for software industry

Research world vs. actual world

Research world
- publications
- citations
- scientific methods
- academic
  communities
- gurus
- conferences
- ranking lists

Actual world
- new ideas
- tools, products
- projects
- applications
- economy
- industrial impact
- web

PhD

not good

not good

Purpose of SE research

- primary: improve software development processes, methods, techniques and tools;
  make progress in software engineering

- secondary: increase understanding of software engineering issues

Why do YOU want to make research?

TUT SE research philosophy

- Problem – Solution – Evaluation
- Evaluation is important, but not the beef
- Emphasis on new ideas, creativeness, practical relevance
- Typically constructive (techniques, methods, tools)

TTY Ohjelmistotekniikka

Purpose of PhD thesis

University viewpoint:

- show matureness
- show knowledge
- show creativity

Why do YOU want to make PhD?

Finding topic/focus/viewpoint

- good results are more likely in new areas
- a central theme would be good to identify early
- think big (real, big problems)
- small problem + big solution: not a good idea
- set of small problems + set of small solutions: ok
- topic often implied or affected by role in a project
- in industry: normally related to work

Contribution: PhD styles

- combination PhD ("Approach X to problem Y")
- delta PhD ("Improving technique X")
- empirical PhD ("Study of phenomenon X in industry")
- generalization PhD ("Unifying approach to problem X")
- method PhD ("Method/process for solving problem X")
- usually PhD applies several styles
- dont get fixed with styles or "research schools": just solve genuine problems

"What if somebody publishes similar ideas before me?"

- usually not a serious problem: independent research produces always sufficiently different results

- can be also a positive thing: confirms your ideas, provides relevant references, connections to the research community etc.

- if you are not aiming at a guru status but only at PhD, competition with similar ideas is not your biggest problem

Different "schools" of SE research (1)

Engineering-oriented school
- emphasizes contribution, problem-orientation,
  practical benefits & usability,
  concrete tools&methods, implementation,
  convincing case studies, constructive approaches
- research paradigms from technical sciences
- software industry seen as potential utilizer of research,
  "authentic" evaluation environment

Different "schools" of SE research (2)

Science-oriented school
- emphasizes use of existing literature, structure, experiments, queries, statistical analysis of results, case studies, understanding
- research paradigms from social sciences
- software industry often seen as target of research

Different "schools" of SE research (3)

Mathematics-oriented school
- emphasizes preciseness, formal presentations, proofs, generality and abstractness
- research paradigms from mathematics and logics
- industrial viewpoint often less emphasized
- software industry seen as a source of motivation

Different "schools" of SE research (4)

SE PhD theses are usually mixtures of these, although with clear emphasis on one

A balanced mixture can make a very good thesis

Making PhD in industry

- how to find understanding, time and funding
- typically concerns pragmatic problems in
  software development
- major problem: how to present results as a PhD
- good examples exist (Nokia)
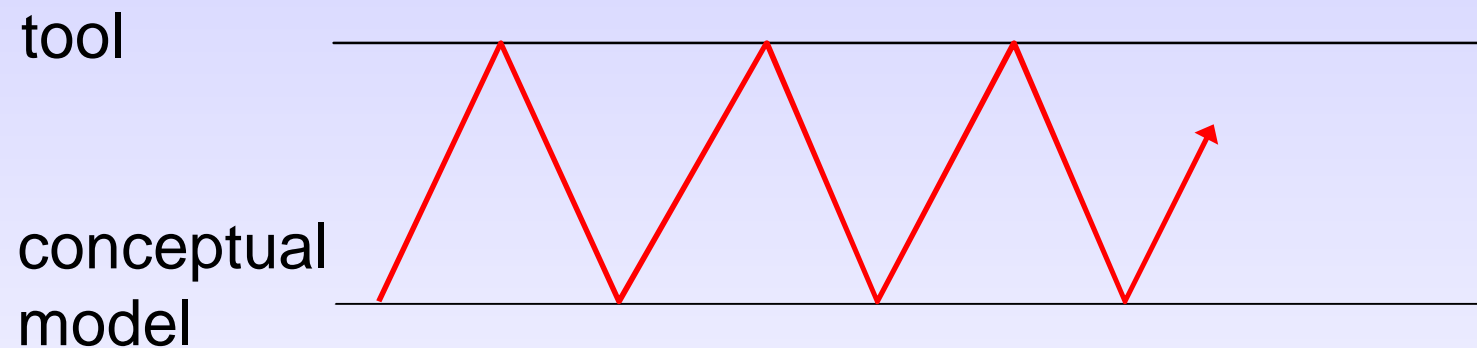- "infrastructural" work profile better than product-oriented

Conceptualization vs. tool development: the cowboy model

tool
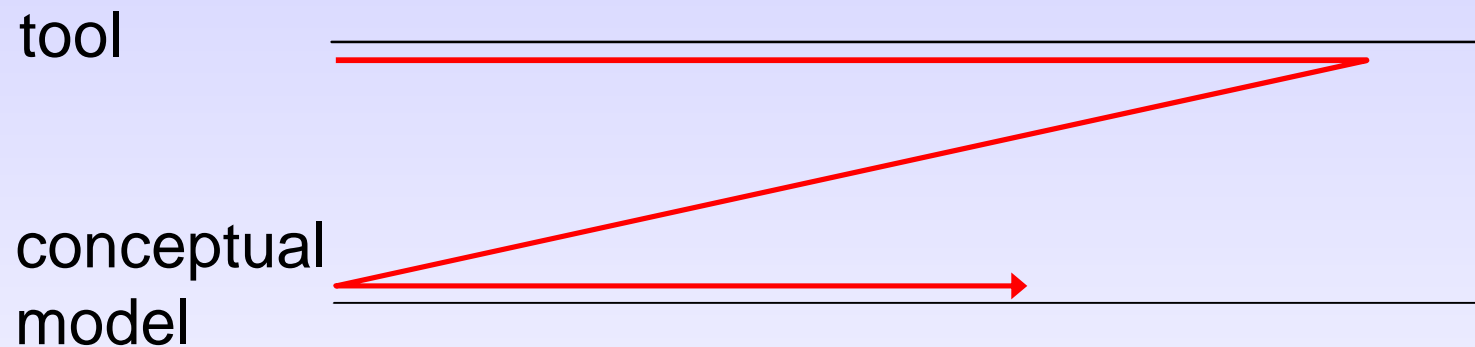
conceptual
model

Conceptualization vs. tool development: the philosopher model

tool
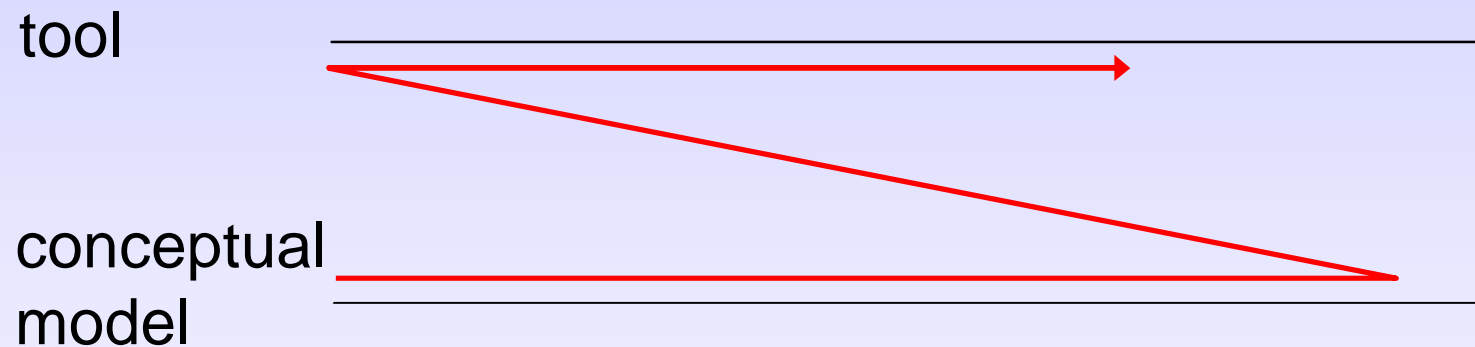
conceptual
model

Conceptualization vs. tool development: the zig-zag model

tool

conceptual model

Monograph vs. article-based

- good publications needed in both cases
- mixture possible: substantial introductory part
  (combines good but also bad sides of both)
- benefits of monograph
  - no need to wait for publications
  - things can be rethought, restructured and rewritten
  - more controlled project
- benefits of paper set
  - incremental process
  - no big bang: introduction can be *very* light
  - usually no problems of accepting as PhD
- narrow&deep topic good for monograph
- broad and shallow topic good for article-based

Papers in monograph & article based PhD

article-based
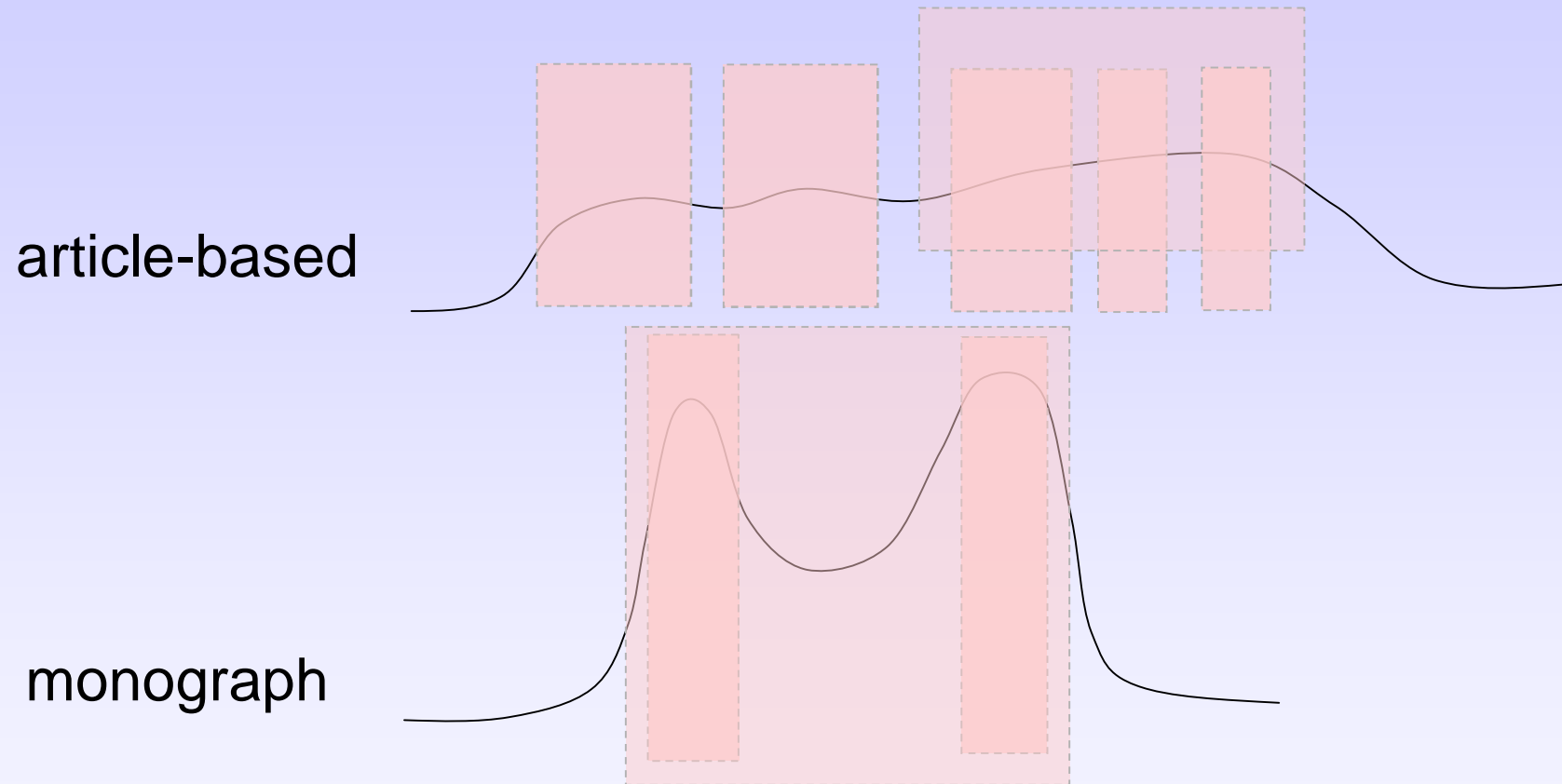
monograph

Article-based PhD styles

- Thin summary: each paper is a section as such, short introduction of 20-30 pages (e.g. Jan Bosch)

  advantages: straightforward, quick, relies on publications

- Thick summary: monograph, but backed-up with a set of publications

  advantages: topics can be rethought, decent evaluation can be included, more coherent

- Typically between these extremes: section for each research topic in publications

Typical parts of SE PhD thesis (1)

Introduction
- setting the scene
- problem statement: research questions
- basic approach
- main contributions
- introduction to the structure of the thesis

Research area survey (context)
- shows that the author knows the area
- positions the work on the map
- defines the needed concepts
- own contribution: clear, structured presentation, possibly summarizing tables, categorizations etc.

Typical parts of SE PhD thesis (2)

Abstract solution

- the beef of the work
- self-contained precise description of the solution
- often augmented with motivations and analysis
- preferrably no tool discussion
- toy examples

Implementation of the solution

- part of the evaluation: solution is implementable
- architecture-level system/tool description
- tool and its UI description, if part of the work
- analysis of implementation problems and their solutions
- high level description, not a system document or user guide
- toy examples

Typical parts of SE PhD thesis (3)

Evaluation
- typically case study -type: one or more realistic
  cases of applying the solution
- justifies the main result: evidence that problem "solved"
- analysis of the cases:
  performance analysis, usability analysis, efficiency analysis,
  quality analysis etc., depending on the research aims
- answering the research questions in the cases
- monograph vs. paper set

Related work
- relates other approaches to solve same or similar problems
- relies on the area survey part, some solutions may be
  outlined there
- main purpose is comparison, clarifying the contribution
  ("drawing the borderlines")

Typical parts of SE PhD thesis (4)

Conclusions
- summarizes the research questions that have been addressed
- summarizes the contributions
- summarizes essential shortcomings
- discusses next steps in this research direction

Evaluation in software engineering research

- There are no absolute truths in software engineering
- Usually it is not possible to really prove something in software engineering, since the goal is to improve industrial practices and tools
- The duty of academia in software engineering is to produce potential technology that has passed initial tests, worthy of entering the evolutionary IT world
- Evaluation in a PhD: initial test
- The final value of the technology will be determined in the evolutionary competition of IT world

Case studies

- originates from social sciences in last century
- typical approach also in medicine & law
- basic (original) motivation: statistical methods hide important
  details that can be seen only when investigating individual
  cases
- appropriate when statistical or analytical methods cannot be
  applied or do not reveal relevant information
- explains complex causal links in real life related to the phenomenon
- describes the real-life context in which the phenomenon occurs
- describes the phenomenon itself
- explores a phenomenon that has no clear set of outcomes
- basic criticism: generalization is questionable

Characteristics of case study research

- normally used to reveal information rather than to prove something
- holistic, in-depth investigation of selected cases
- cases selected so as to maximize what can be learned
- no required nor optimal amount of cases (one is enough)
- triangulation: getting confirmation for results by using several sources
- researcher often in key role

Case studies in SE

Why case studies are used in SE research?

- Statistical methods are often next to impossible
- Individual cases reveal more interesting information

Two different usages of case studies in SE research

- Investigate some existing phenomenon using a case study
- Investigate your own solution using a case study

Case studies in the evaluation part

Investigate your own solution using a case study

- the main contribution of the thesis is not the case study
- the purpose of the case study is
  - to explain the real life context of the solution
  - to find new facts about the solution in real applications
  - to explain the solution in more detail
  - to demonstrate the benefits of the solution in a real-life situation
  - to give "circumstantial evidence" about the general benefits of the solution (typical case)
  - to give "conditional evidence" about the general benefits of the solution (with these characteristics of the application, these are the benefits)
- actually this is a new type of usage of a case study in research

Analyzing case studies

- explain explicitly what you want to achieve
- be as analytical as you can (use tables, quantitative data, etc., if possible – performance, usability, effectiveness,…)
- give the essential properties of the case studies that may affect the results
- be as neutral as you can (discuss alternative techniques, observe drawbacks of your solution as well etc.)
- do not overestimate the evidence
- do not select the case studies especially amenable to your approach
- establish an infrastructure to collect the data