Seminar: Entwicklungsprozess von Software-Produktlinien

Sandro Schulze
Take a seat....buckle up...get started

Who I am?

Computer science background (e.g., interests, programming skills, scientific working)

Why I am here (and not in the very cool Seminar of XXX)?

What do I expect?
Introduction and Organization
Overview

- (Under)-Graduate level course (Bachelor, Master)

- Topics
  - Academic Writing
  - Presentation
  - Literature review

- Goal
  - Sharpen your soft skills (presenting the unknown to the unfamiliar)
  - Structured and goal-oriented working
  - Extracting and comparing information
  - Having fun ;)


Contents

• Several lectures
• Write an academic paper... “Current state and future challenges in X” (7-10 pages)
• Submit the paper
• Participate in a review process
• Present your work (conference style)
• Paper and presentation can be either German or English
Warning

• Take this course seriously!

• 5 Credit Points = 150 h, all within the semester (~10h/week)
• Focus on reading and writing
• Paper (first version) is due in 5 weeks!

• Plenty feedback and opportunities for improvement: Use it by starting with a good version

• Tough grading at the end
Deadlines (strict)

- November 13 (in 3 weeks): 5min presentation of topic and relevant literature
- November 16: Submission of outline
- November 30 (in 5 weeks): Submission of paper
- December 14 (within 2 weeks): paper reviews due (2x)
- January 18 (within 4 weeks): Submission of improved paper
- ???: Practice presentation; **Edit: no practice presentation**
- January/February: Final presentation
Grading

- Grades based on quality of
  - the final paper (50%),
  - the reviews (20%), and
  - the final presentation (30%)
- Only the final paper and presentation are graded
- Focus on formal criteria (style, cohesion, adequate references) instead of novel contribution
- Details in the corresponding lectures
- Participating in all lectures is mandatory; precondition for grading
Topics
1. Variabilität in Anforderungsspezifikationen
2. Formale Ansätze zur Variabilitätsmodellierung
3. Vergleich von konkreten Ansätze zur Variabilitätsmodellierung
4. Entwicklungsprozess von SPLs - vom Produkt zur Produktlinie
5. Kompositionale Ansätze zur Implementierung von Produktlinien
6. Annotative Ansätze zur Implementierung von Produktlinien
7. Testen von Produktlinien bzw. Variablen Softwaresystemen
8. Refactoring von Softwareproduktlinien
9. Anwendung und Realisierung von SPLs in der Praxis
10. Der Linux Kernel als Software-Produktlinie
Software Product Lines
Do you know examples for (Software) Product Lines?

Why to use the product-line approach?

Advantages? Disadvantages?
Car Product Line

Model | Ausstattung | Farbe, Interieur + Felgen | Editionen + Pakete | Sonderausstattungen | Zusammensetzung
---|---|---|---|---|---
Gebrauchte Automobile | Service & Zubehör | Faszination BMW

318i Touring
Gesamtpreis | Monatliche Rate
34,970,00 EUR

Fahrzeugdaten ausdrucken
Finanzierung und Leasing
Denken und Stellen
Hinweise zu Preisangaben

Modell | Code | Preis
---|---|---
Automatic Getriebe | 205 | 2,180,00 EUR

Klima, Heizung

Stellheizung mit Fernbedienung | 538 | 1,340,00 EUR
Sonnenschutzvergütung, Individuell | 781 | 380,00 EUR
Klimaautomatik mit Fensterautoklappen

Komfort+Luxus

Ablagenofset | 403 | 110,00 EUR
Armaturen, versenkbare | 44E | 150,00 EUR
Sitzpolsterung

Sandro Schulze
Seminar WiSe 2012/2013
Entwicklung von Software-Produktlinien
Sandro Schulze
Seminar WiSe 2012/2013
Entwicklung von Software-Produktlinien

Willkommen bei selve - the shoe individualizer
http://www.selve.net/index.js.html

selve Kollektion --> Style: casuals --> Modell: Opal

modell-details >> hier klicken

A. Erstes Oberleder
   - Veloursleder Sand
B. Veloursleder Bordeaux
   - Veloursleder Cognac
   - Veloursleder Sand
C. Futterleder
   - Beige
D. Absatz
   - Hufeisen Braun
E. Sohle
   - Gummisohle

>> Ändern
>> zurücklegen
Sandro Schulze  
Seminar WiSe 2012/2013  
Entwicklung von Software-Produktlinien
Entwicklung von Software-Produktlinien
Entwicklung von Software-Produktlinien
Software Product Lines

A software product line (SPL) is a set of software-intensive systems that share a common, managed set of features satisfying the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way.

Software Engineering Institute
Carnegie Mellon University
Software Product Lines

A set of program variants (software products),
...tailored to a common market segment (domain)
...with the goal of reuse common software artefacts

e.g., database product line for embedded system
Domain

• Programs of a product line tailored to application area
• Application area is called *domain*
• Horizontal domains
  • billing, inventory management, flight booking
• Vertical domains
  • Numeric algorithm, network drivers, GUIs, databases
What is a feature? (German: Merkmal)

- Domain abstraction
- Represents requirements, commonalities, variabilities amongst program variants
- Mean for communication between stakeholder
- Specification of variants
  - Feature selection as input for program generation
- BUT: there are concerns, which are NO features
Concern vs. Feature

• Concern
  • Every problem statement of interest

• Feature
  • Problem statement with specific meaning in the domain
  • Configuration option
Features in Databases

• Transaction management
• Log & Recovery
• Write access
• Persistency / In-Memory
• Place replace strategies LRU / LFU / Clock /...
• Sorting algorithms
• Records with variable length
• Grouping, aggregation
• Windows / Unix / NutOS / TinyOS / ...
Development of Product Lines

• Developing a product line instead of single applications
• Product lines covers requirements of whole domain
• Deviates from traditional development process and lifecycle
• Distinction between
  • Domain Engineering
  • Application Engineering
Software Lifecycle - typical

- Requirements
- Design
- Implementation
- Verification
- Maintenance
Domain Engineering

[...] is the activity of collecting, organizing, and storing past experience in building systems [...] in a particular domain in the form of reusable assets [...], as well as providing an adequate means for reusing these assets (i.e., retrieval, qualification, dissemination, adaptation, assembly, and so on) when building new systems.

K. Czarnecki and U. Eisenecker
Application and Domain Engineering
Development Effort


**Literature on Software Product Lines**


<table>
<thead>
<tr>
<th>Sandro Schulze</th>
<th>Seminar WiSe 2012/2013</th>
<th>Entwicklung von Software-Produktlinien</th>
</tr>
</thead>
</table>
Survey Papers

- Discuss different approaches/solutions for a problem
- Group/classify solutions, discuss commonalities and differences
- Try to be complete or at least select important representatives
- Aggregation and describing relationships is your contribution! (instead of a new solution to the problem)

- Which problems are still open
- Are there initial ideas for that?
Our Requirements

• Explain the problem (and its importance) to readers unfamiliar with the domain
• Discuss different solutions (by different research groups)
• Give a benefit compared to reading all papers
  • Discussion, comparison, or additional context
• Suggestions for future directions
• Keep it within 8 pages (IEEE CS 2 column format, including references)
Finding literature

- Library: Journals (online&print), also interlibrary lending (Fernleihe)
- scholar.google.com (use also link “All x versions”)
- ACM Digital Library (within university network)
- dblp.uni-trier.de

- Related Work therein
Examples of Survey Papers


Why to Publish
Research requires Writing

- Writing is fundamental part of research (do not underestimate!)
- > 30% of time spent on writing

- Writing is work, a profession (not talent, inspiration, …)
- Writing can be learned
- Practice, practice, practice…

- Some publications are expected when submitting PhD thesis
- PhD students that never focus on writing within the first years, hardly finish their PhD
- Scientists are often evaluated based on publications
Peer Review

• Scientific quality control
• Experts in the field review submitted papers, recommend acceptance/rejection and suggest improvements
• Quality reviews require high effort
• Only peer reviewed publications count
Conference vs. Journal

- **Journal**
  - More (long-term) impact
  - Wider target audience
  - Deeper reviews, long review process
  - More space (15-50 pages)
  - Typically research results of 1-5 years

- **Conference Proceedings**
  - Faster process
  - ca. 10 pages
  - Direct contacts and discussion at conferences; community
  - Audience: 30-200

- **Workshop**
  - Discussions; Community
  - Work in progress, ideas, first results, 4-8 pages
  - Audience: 10-20

- **Technical Report**
Ranking

• Different quality levels between conferences, workshop and journals
• Top-Level conferences receive over 300 papers, tough peer review, select only best (often <20%), often considered equivalent to journals
• Some local conferences and workshops accept almost everything, little quality control, some for money

One Example for Ranking:
<table>
<thead>
<tr>
<th>Sandro Schulze</th>
<th>Seminar WiSe 2012/2013</th>
<th>Entwicklung von Software-Produktlinien</th>
</tr>
</thead>
</table>


Tools

- Latex
- Word/Open Office with Bibliography-Tool such as Endnote or Citavi (URZ!)
- Use a spell checker + grammar checker, e.g. in Word
  - Latex users: latex2rtf
- SVN -> https://faracvs.cs.uni-magdeburg.de/svnadmin/
Literature

- Mary Jean Harrold, Axel van Lamsweerde - New Software Engineering Faculty Symposium, ICSE 2006 (Slides)
- Graham Horton – Schlüsselkompetenzen (Slides)
- William Cook – Academic Writing (Slides)
- Joseph M. Williams – Style: Toward Clarity and Grace, 1995
- Lyn Dupre – Bugs in Writing, 1998
- The Chicago Manual of Style