RE@40
Forty Years of Requirements Engineering – Looking Forward and Looking Back
April 24-28, 2017 • Kappel am Albis, Switzerland

Requirements Engineering for Software Product Lines

Julio Cesar Sampaio do Prado Leite
Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro
(PUC-Rio)

Support from:
Goal

Overview of the role of RE in the context of Software Product Line
Viewpoint

RE as a Concurrent Engineering Activity that may be seen as composed of the following parts: Elicitation, Modeling, Analysis, and Management.
Out There
Software Product Line

- Reduce time to market
- Increase Software Quality
- Extensive Planned Reuse
Fig. 4. In domain engineering, variability models and domain artifacts are created, which are used in application engineering to automatically generate software products based on feature selections.

**Objective:** Model the process for giving a talk on RE for SPL

**Viewpoint:** A researcher in RE, who needed an update on the topic.
Product line Book

SEARCH

1

Scholar

330 citations

Snowballing

Previous Knowledge

FOLLOW

2

Information Sources

Scholar

ACQUIRE

3

Introspection

Concepts
Process for this Talk

• Eliciting
  – Introspection (on my own previous knowledge)
  – Eliciting Information Sources (recent literature)
    • “Stand on the shoulders of giants”
    • Seed (Product Line Book 3305 citations)
    • Snowballing
    • Hand selecting over last 2 years
    • Some other specific queries
  – Document reading

• Modeling as general concepts
• Analyzing by you
• Managing by time
Information Sources


Information Sources


Information Sources


Elicitation

- Market-Driven
- Scoping (Negotiation, Conflicts, Cost)
- Domain versus Context
- Functional versus Non-Functional
- Domain Engineering vs Application Engineering
- Evolution (Reactive, Extractive)
Modeling

- Problem Space x Solution Space
- Feature-Oriented (tree based, constrained based)
- Goal Oriented
- Viewpoint Oriented
- Variation points (case; #ifdef; files; components; OR, XOR, ...)
- Evolution (reactive, extractive, tracing)
Analysis

- Feature Interaction
- Consistency Analysis
- Well formed Patterns
- Configuration (Verification, Rules)
- Testing (Requirements/Feature Based)
- Thum et al. on Verification
Management

- Value Investment
- Requirements Management vs Managing by Requirements
- Strategy (proactive, reactive, extractive)
- Configuration (Concurrent Management)
- Commitment
- Tools
Solid Gains

- Industry Acceptance
- Success Cases
- Industry Dependent
- ROI
- Tools
- Other opinions (lack of empirical work, failure on dealing with rapid change)
Future Challenges

- Inner Source (Reusing Open Source Experience)
- NFR driven Elicitation (QFD, Catalogs)
- Mining Software Repositories for Features
- Awareness Requirements Based Product Line
- Tools