A Product Line of Software Engineering Project Courses

Nicolas Guelfi, Alfredo Capozucca and Benoît Ries
University of Luxembourg, Esch-sur-Alzette, Luxembourg

Problem Definition

Introducing a software engineering (SE) project course in an education program is a challenging task.
Teaching teams of such courses face two main problems, among many others:

1. which notions should be covered by the project
2. how to design a course covering those notions

1. Coverage

Over the last few years, the software engineering community has developed some important education means.
The Software Engineering Body of Knowledge (SWEBOK) [1] classifies the SE knowledge using 15 knowledge areas (KA), 99 topics and 395 sub-topics. In our approach, the SWEBOK is used to select notions covered by the SE project courses variants.

Nb.	Knowledge Area Names
1	Software Requirements
2	Software Design
3	Software Construction
4	Software Testing
5	Software Maintenance
6	Software Configuration Management
7	Software Engineering Management
8	Software Engineering Process
9	Software Engineering Models and Methods
10	Software Quality
11	Software Engineering Professional Practice
12	Software Engineering Economics
13	Computing Foundations
14	Mathematical Foundations
15	Engineering Foundations

Figure 1: SWEBOK Knowledge Areas

2. Design

Variation Points

- SE knowledge areas
- SE tools & technologies
- Application domains
- Course administration and management

Product Derivation

- Top-down
- Bottum-up-and-down
- Hybrid

Qualities

- Functional suitability
- Performance efficiency
- Compatibility
- Reliability
- Security
- Maintainability
- Portability

Work Summary

The MESSEP approach offers to instructors a product-line approach to SE project courses allowing to derive its own course in an efficient way. Our approach is illustrated using several courses that we designed and performed in different universities and at different education levels. Since 2012, we have derived five SE project course variants using our approach in three different education institutions.

Reference Card

Property	Values
Course Variant Name	BINFO-SEP-and-SE2-2016-2017
Insitution	University of Luxembourg, LU
Education program	Professional Bachelor in Computer Science
ISCED Level [2]	BA 655 (Bachelor/Professional/First degree)
Schedule	14 weeks * 2 periods
Total learner's workload	10 hours / week
Periods	Period 1: [Sprint1 (5 weeks) + Sprint2 (3 weeks) + Sprint3 (3 weeks) + Sprint4 (3 weeks)] Period 2: [Sprint1 (5 weeks) + Sprint2 (3 weeks) + Sprint3 (3 weeks) + Sprint4 (3 weeks)]
Learners Team Size	4 nominal - [2,3] exceptional
Main SWEBOK KA Coverage	KA1 Software Requirements: (80%) KA9 Software Engineering Models and Methods: (75%) KA7 Software Engineering Management: (67%) KA11 Software Engineering Professional Practice: (63%) KA2 Software Design: (46%) KA3 Software Construction: (39%)
Main Market	Applications/Collaborative Applications/Team Collaborative Applications
Tools/Technologies with focus level	KA1 Software Requirements: UML (3), Eclipse (3), Excalibur (3), Latex (3), OCL (2), Texlipse (2), xindy (1), Texlive (1), Inkscape (1), PDFTk (1), ,PDF Reader (1), Xtext (0), Java SDK (0), Sirius (0), EMF (0) KA2 Software Design: UML (3), Eclipse (3), Latex (3), UML Designer (3), JustInMind (3), Texlipse (2), PDF Reader (1), xindy (1), Texlive (1), EMF (0), Sirius (0) KA3 Software Construction: Eclipse (3), e(fx)clipse (3), Java (3), MySQL (2), JavaFX (1) KA5 Software Maintenance: Atlassian JIRA (2), Atlassian Confluence Questions (2) KA6 Software Configuration Management: GitHub (3), Git (3) KA7 Software Engineering Management: ZenHub (3), GitHub (3) KA13 Computing Foundations: Windows (2), VirtualBox (2), Linux (2), Mac OS X (2)

Figure 2: Reference Card of an SE Project Course Variant

SE Course Product Variant Coverage Comparison Diagram

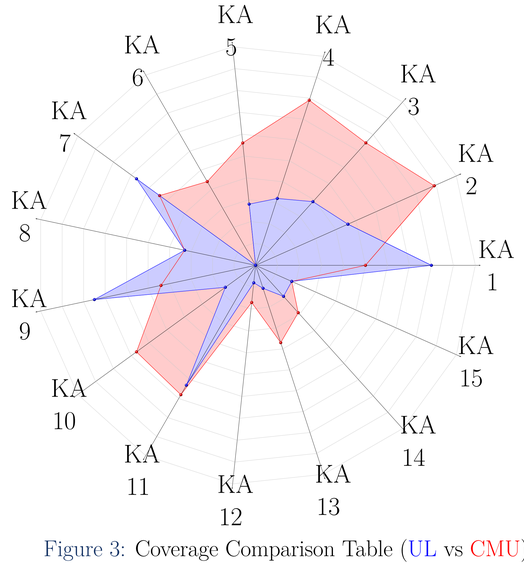

Figure 3: Coverage Comparison Table (UL vs CMU)

Conclusion & Future Work

This poster presents a method for the derivation of software engineering project courses. It reuses, from a conceptual viewpoint, the product line paradigm for its description, and is strongly based on the SWEBOK. The qualities of our method and of the already made derivations represent an improvement for deploying high-quality SE courses.
The future work planned will mainly focus on developing a tool-support to allow the education community to specify (exploiting our preliminary work on course specification using a domain-specific language [3]), derive, reuse and improve SE projects courses product lines.

References

[1] ISO/IEC.
Software Engineering – Guide to the Software Engineering Body of Knowledge (SWEBOK).
International Organization for Standardization, 2014.
ISO-IEC TR 19759-2014.

[2] UNESCO.
International Standard Classification of Education (ISCED) 2011.
UNESCO Institute for Statistics, 2012.

[3] Nicolas Guelfi, Benjamin Jahic, and Benoît Ries.
TESMA: Requirements and design of a tool for educational programs.
Information, 8(1), 2017.

Acknowledgements

The authors would like to thank Vladimir Itsykson, Evgeny Pyshkin, Bertrand Meyer, Cécile Péraire and all the students of the partner institutions for their help and support in the deployment of our SE project course product variants.

UNIVERSITÉ DU LUXEMBOURG