HELENA

Walking through the Method Zoo: Does Higher Education really meet Software Industry Demands?

Marco Kuhrmann, Joyce Nakatumba-Nabende, Rolf-Helge Schneider, Paolo Tell, Jil Klünder, Tayana Conte, Stephen G. MacDonell, Regina Hebig

Embassy of Sweden
Kampala

The “reality” of software development

Companies go for the best of both worlds **(Traditional, Agile, Hybrid)…**

- Managers need a safe harbor = traditional process
- Developers demand freedom = agile methods/practices
- Clients want their system fast = agile methods/practices

And here arise some problems…

- What is a suitable process combination (for me)?
- How to develop such a process combination?
- How to cover a complete dependable system lifecycle?
- How to train the people?
- How to ensure that the combined process is accepted?
- …

Idea

HELENA:

- Large scale international survey to study the use of **H**ybrid d**E**veLopm**EN**t **A**pproaches in software systems development
- What processes and process combinations are used in industry?
- Hybrid method use is common!

HELENA@SEET

- Do we prepare students for the reality that we found in HELENA?

➢ Replication study with teachers!

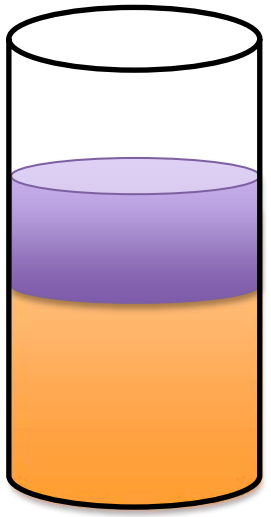
Objective

- Problem:
  - Given that it typically takes 3 to 5 years to educate students, it is hard for HEIs to catch up with industrial innovation cycles
  - It is impossible to teach everything in just one course.
- Analyze the current state-of-practice in software engineering education and compare it with industrial practice.
- Specifically - consider whether software development *frameworks*, *methods* and *practices* as used in industry are present in HEI courses. Our study aims to draw a big picture providing support for educators to evolve their courses to strategically balance basic knowledge and industry demands.

Demographics

Actually, there are 2 studies…

- HELENA (Stage 2)
  - International audience
  - Industry practitioners (projects/products)
  - Convenience sampling from 55 countries
  - Data collection: May-November 2017
  - Very comprehensive data ☺
- HELENA (Educator Survey)
  - International audience
  - Educators (universities etc.) from 17 countries
  - 241 personally invited participants
  - Data collection: September 2018



Participants

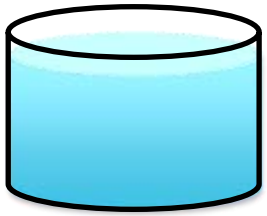
1467

732

Selected answers

691

Complete answers



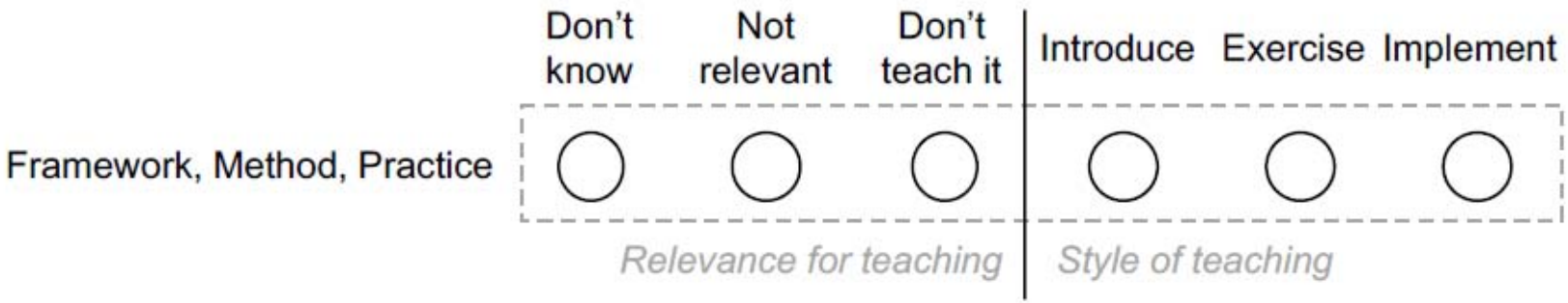
67

Cases, 63 out of 241 educators

Replication study

Some changes for academia replication survey

- Smaller survey: no focus on problems to solve with the process (maximum of 25 questions)
- Changed scales: no “how often used”



- Collected motivations for teaching choices

Research Questions

- RQ1: Which software and system development frameworks, methods and practices are taught in higher education?
- RQ2: To what extent does higher education cover and coincide with software and system development frameworks, methods and practices that are applied in industry?
- RQ3: Why do educators decide to cover the chosen development frameworks, methods and practices?

Research Approach

1. Survey Instrument

- Instrument development until August 2018, *incl. basic questionnaire design, analysis and integration of the HELENA questions, and question refinement and alignment*
- Internal instrument evaluation started in mid August 2018
- Identification of potential study participants for personal invitation (241 educators invited)
- Data collection: September 3-16, 2018
- Result: 152 total data points (67 complete)
- September 17, 2018: Begin of data analysis

2. Data Analysis

- Data Cleaning and Reduction
- Qualitative Analysis and quantitative Analysis

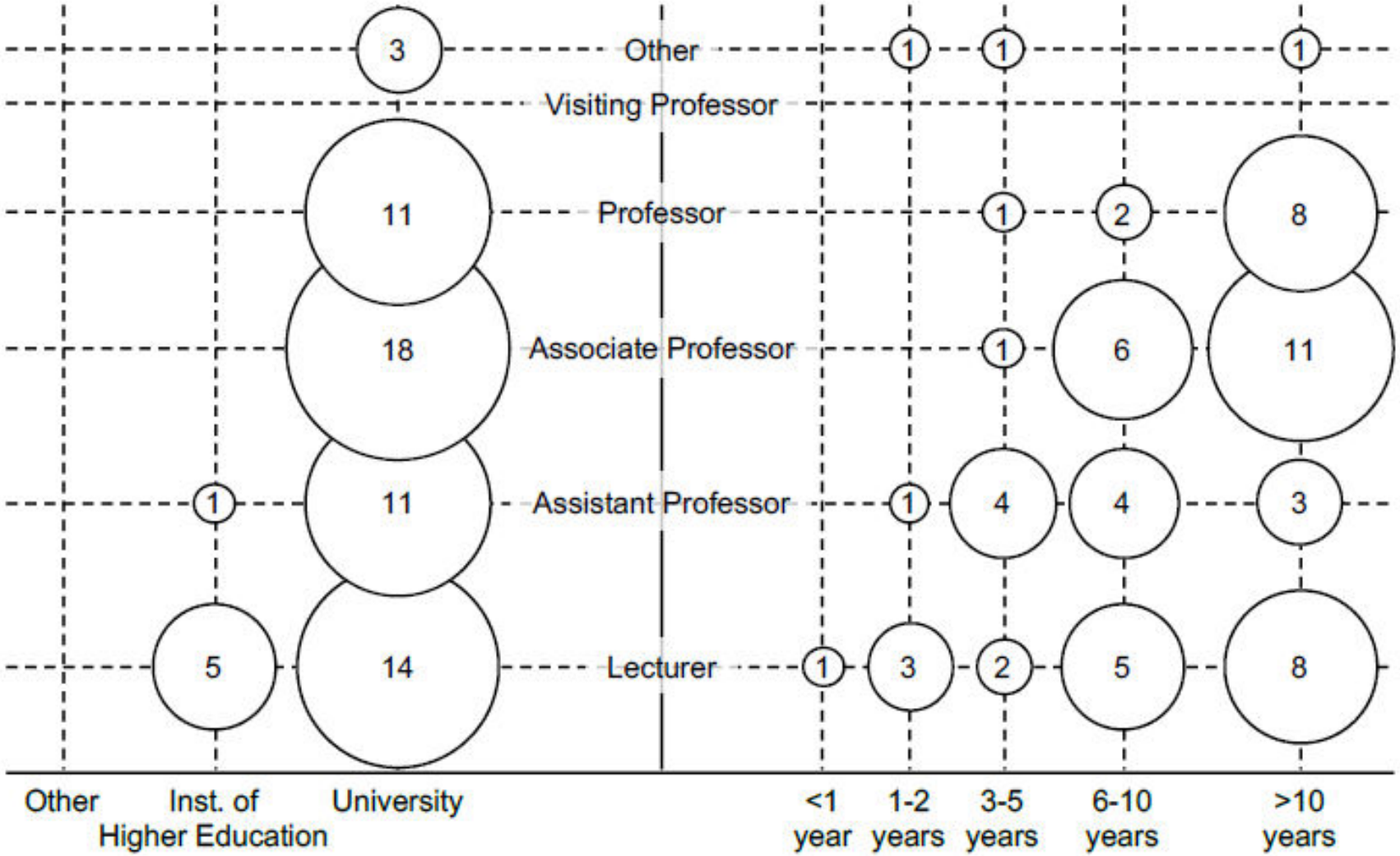
2a. Quantitative Analysis

- HELENA data selection and integration with survey results
- Descriptive statistics
- Comparative analysis

2.b Qualitative Analysis

- Data selection: course data and free-form text
- Thematic coding in 2 parallel groups

Population

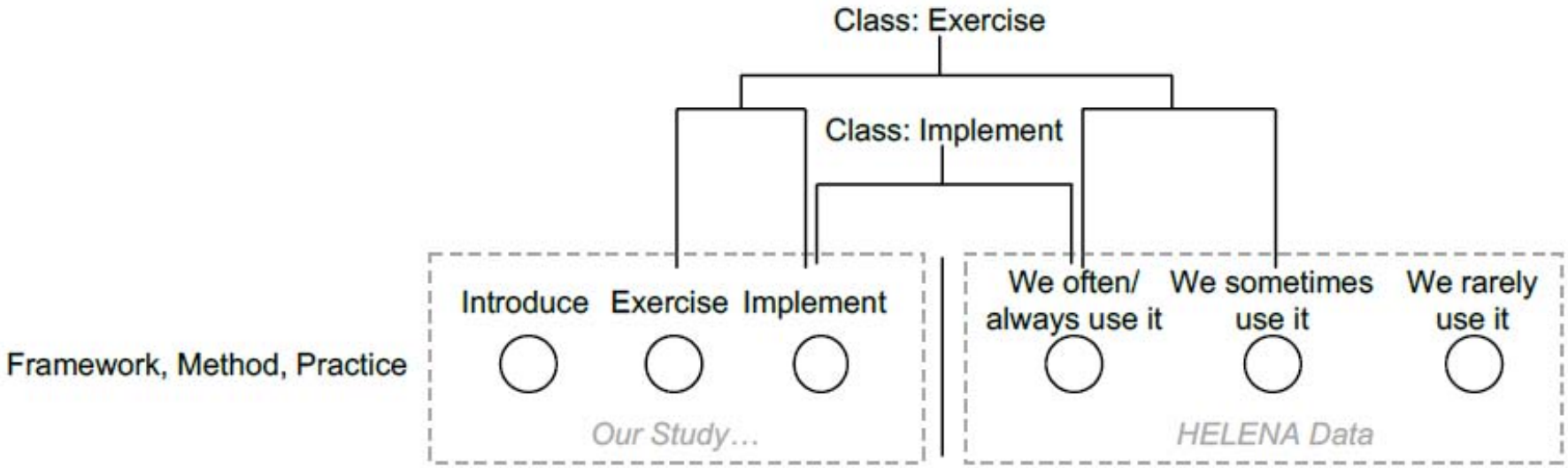

Other
Visiting Professor
Professor
Associate Professor
Assistant Professor
Lecturer
3
11
18
11
1
5
14
1
1
1
1
1
1
2
8
1
1
6
11
1
1
4
4
3
1
3
2
5
8
Other
Inst. of Higher Education
University
<1 year
1-2 years
3-5 years
6-10 years
>10 years

RQ1: Which software and system development frameworks, methods and practices are taught in higher education?

Rank	Framework/Method	This study		HELENA	
		n	%	n	%
1	Iterative Development	60	89.55	557	76.09
2	Classic Waterfall Process	44	65.67	404	55.19
3	Scrum	43	64.18	597	81.56
4	Extreme Programming (XP)	40	59.70	368	50.27
5	Kanban	28	41.79	468	63.93
6	Spiral Model	28	41.79	143	19.54
7	V-shaped Process	27	40.30	191	26.09
8	Rational Unified Process	25	37.31	143	19.54
9	Domain-driven Design	24	35.82	234	31.97
10	Feature-driven Development	23	34.33	263	35.93

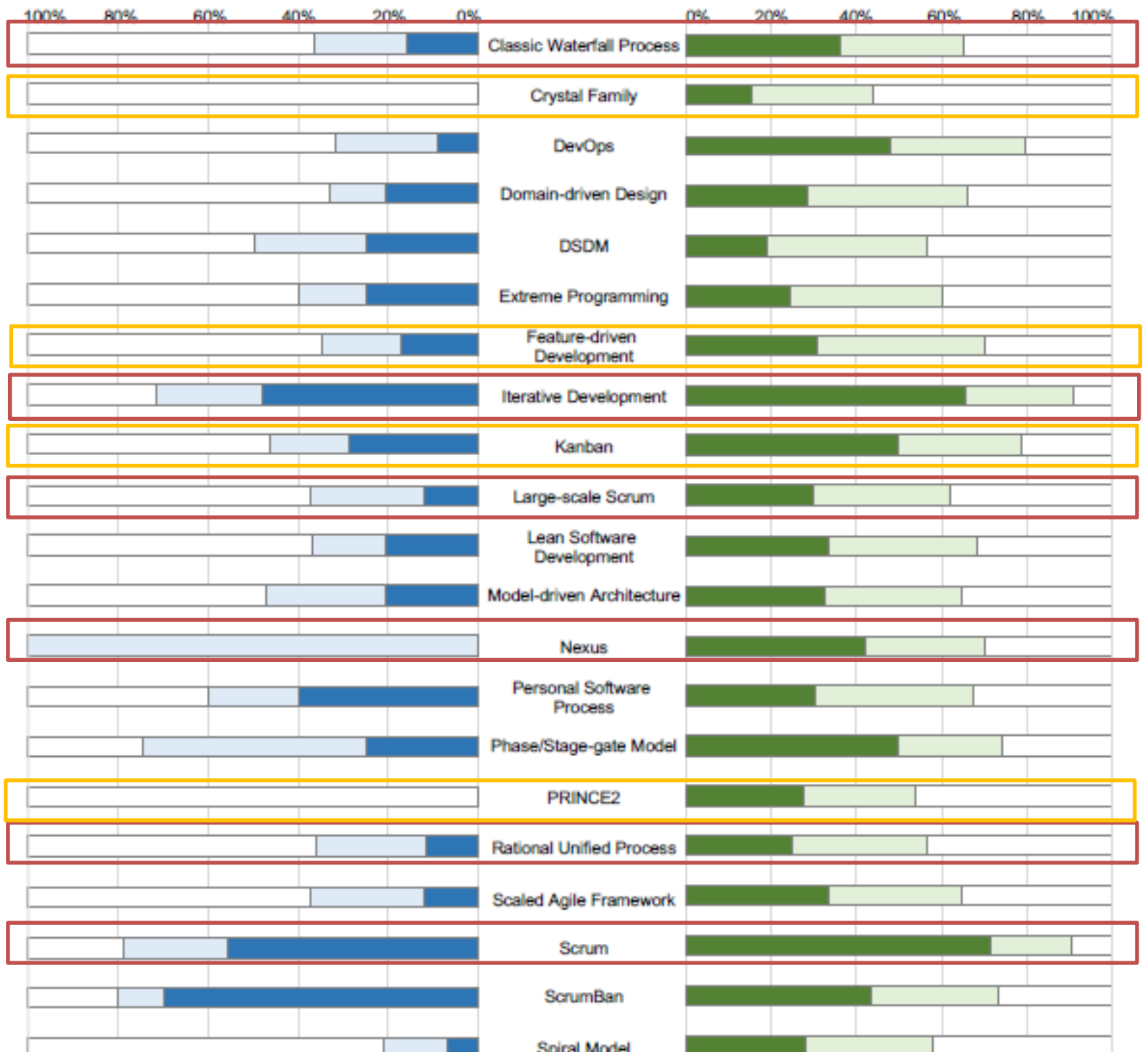
Rank	Practices	This study		HELENA	
		n	%	n	%
1	User Stories	54	80.60	579	79.10
2	Architecture Specifications	49	73.13	564	77.05
3	Prototyping	47	70.15	625	85.38
4	Coding Standards	46	68.66	666	90.98
5	Refactoring	46	68.66	608	83.06
6	Code Review	44	65.67	681	93.03
7	Pair Programming	44	65.67	473	64.62
8	Use Case Modeling	44	65.67	436	59.56
9	Test-driven Development	42	62.69	453	61.89
10	Continuous Integration	41	61.19	593	81.01
11	Iteration/Sprint Reviews	41	61.19	593	81.01
12	Iteration Planning	40	59.70	579	79.10
13	Automated Unit Testing	39	58.21	616	84.15
14	Backlog Management	37	55.22	592	80.87
15	Continuous Development	36	53.73	521	71.17
16	Collective Code Ownership	33	49.25	436	59.56

RQ2: To what extent does higher education cover and coincide with software and system development frameworks, methods and practices that are applied in industry?

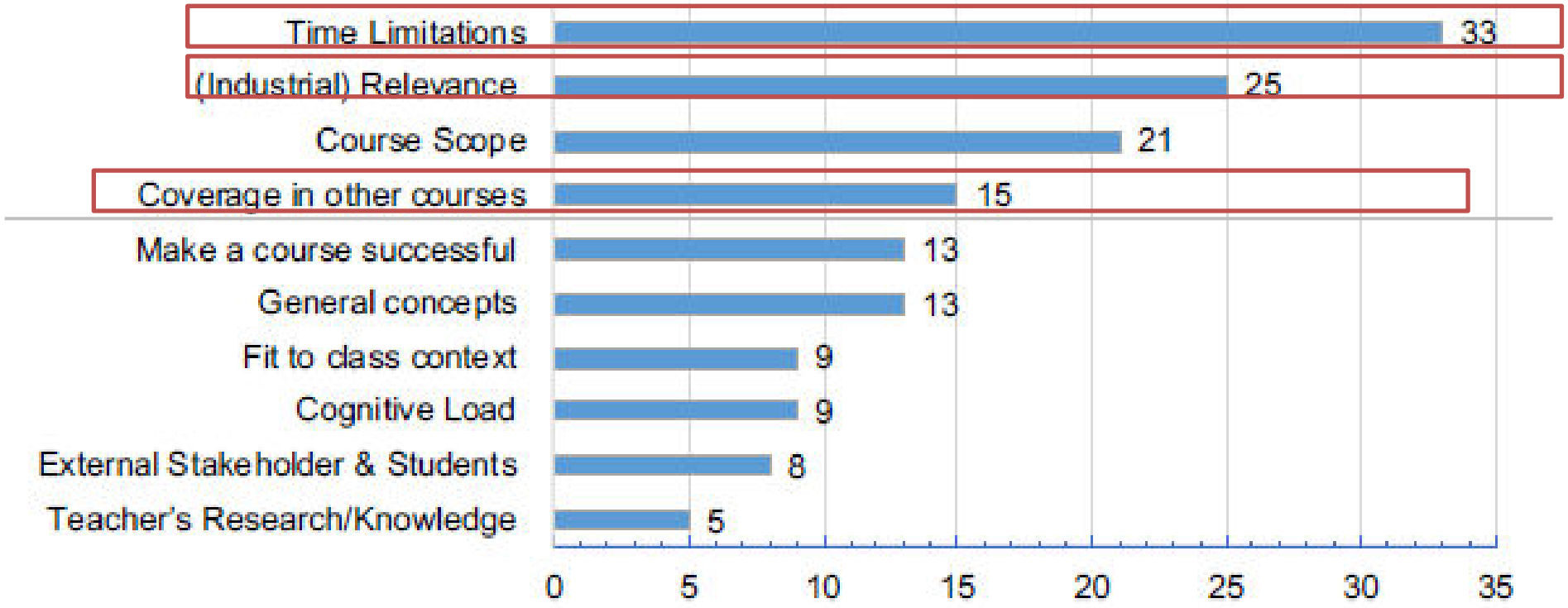


Class	Test	τ
Implement	z=7.0929, p-value $= 1.313 \times 10^{-12}$	0.6413766
Exercise	z=7.3317, p-value $= 2.273 \times 10^{-13}$	0.6581279

Education and Practice – Do they match?



RQ3: Why do educators decide to (not) cover the chosen development frameworks, methods and practices?



Takeaways for Teachers

Consider other courses in your program

- Do other courses earlier in the program already provide basic knowledge to build upon?
- Are there synergy potentials with other courses that rely on the same frameworks, methods and practices?
- Is it possible to develop “teaching sequences?

Less is more: Not every course needs to teach the full set of processes available

- Will the students be confronted with specific processes?
- Could guest lecturers from local industry provide required process knowledge?

- General concepts first.

Takeaways for Teachers

No teaching material for hybrid processes

- Can teaching collaborations, e.g. with other HEIs, provide the required process knowledge?
- Can guest lecturers from industry provide insights into local companies' specific processes?

The "Real-World Trap"

- Do student teams really select and follow a process?
- Do industry partners really help students implement the partner's process?

> "In addition I have not yet found a good course book that can be effectively used together with an iterative approach. (The teaching material tends to come in a format befitting the waterfall model, discussing elicitation, design, implementation, testing and deployment in a very sequential manner." [case 204]

Key Findings

- Teaching addresses both kind of approaches: the ‘’traditional’’ and the agile ones.
- Iterative development, the Waterfall process, Scrum and XP (eXtreme Programming) are taught by more than 50% of the educators.
- The list of practices taught by more than 50% of the educators comprises several items, including user stories, architecture specifications, code review and refactoring.

Key Findings

- Comparing the state-of-practice in teaching and the industrial practice shows - frameworks, methods and practices taught in higher education correspond with those used in industry.

- Educators cannot teach each development method, framework or practice due to time limitations, industrial relevance and course scope.

Kuhrmann, Marco, Joyce Nakatumba-Nabende, Rolf-Helge Pfeiffer, Paolo Tell, Jil Klünder, Tayana Conte, Stephen G. MacDonell, and Regina Hebig. "Walking through the method zoo: does higher education really meet software industry demands?." In *Proceedings of the **41st International Conference on Software Engineering: Software Engineering Education and Training**,* pp. 1-11. IEEE Press, 2019.

Walking through the Method Zoo: Does Higher Education really meet Software Industry Demands?

Marco Kuhrmann*, Joyce Nakatumba-Nabende†, Rolf-Helge Pfeiffer‡,
Paolo Tell‡, Jil Klünder§, Tayana Conte¶, Stephen G. MacDonell‖ and Regina Hebig**

*Clausthal University of Technology, Goslar, Germany, Email: marco.kuhrmann@tu-clausthal.de
†Makerere University, Kampala, Uganda, Email: jnakatumba@cis.mak.ac.ug
‡IT University of Copenhagen, Copenhagen, Denmark, Email: {ropf, pate}@itu.dk
§Leibniz University Hannover, Hannover, Germany, Email: jil.kluender@inf.uni-hannover.de
¶Federal University of Amazonas, Manaus, Brazil, Email: tayana@icomp.ufam.edu.br
‖Auckland University of Technology, Auckland, New Zealand, Email: smacdone@aut.ac.nz
**Chalmers | University of Gothenburg, Gothenburg, Sweden, Email: regina.hebig@cse.gu.se

***Abstract*—Software engineering educators are continually challenged by rapidly evolving concepts, technologies, and industry demands. Due to the omnipresence of software in a digitalized society, higher education institutions (HEIs) have to educate the students such that they learn how to learn, and that they are equipped with a profound basic knowledge and with latest knowledge about modern software and system development. Since industry demands change constantly, HEIs are challenged in meeting such current and future demands in a timely manner. This paper analyzes the current state of practice in software engineering education. Specifically, we want to compare contemporary education with industrial practice to understand if frameworks, methods and practices for software and system development taught at HEIs reflect industrial practice. For this, we conducted an online survey and collected information about 67 software engineering courses. Our findings show that development approaches taught at HEIs quite closely reflect industrial practice. We also found that the choice of what process to teach is sometimes driven by the wish to make a course successful. Especially when this happens for project courses, it could be beneficial to put more emphasis on building learning sequences with other courses.**

***Index Terms*—software development, software process, hybrid methods, survey research, education**

I. Introduction

Given the huge and growing number of concepts, methods, and technologies available for software and system development, many educators have to make decisions over what to include in or exclude from a curriculum and what not. In software engineering, among other things, the selection of software development frameworks, methods and practices constitutes a major challenge. Is it better to focus on the traditional models to thoroughly teach software engineering foundations? Is it better to teach "just enough" of methodology to serve a specific course or project context? Or is it better to largely ignore the fashions and teach the students how to learn and adapt any approach?

Software and system development is thus inherently diverse and a still growing number of application domains vitally depend on software. Consequently, software and system development has to fulfill a number of context-driven requirements. For instance, in the field of dependable systems, software is critical and has to comply with standards. Cars or medical devices are developed following such standards [1], [2], which "dictate" the way software is developed to a large extent, e.g., by defining quality management systems or test-, integration-, evaluation- and approval procedures. At the other end of the spectrum, startups work in the most pragmatic way possible to try out new ideas and to release a product as quickly as possible. Wasserman [3] states that *"...many startups [...] are completely unsystematic developing their MVP [minimum viable product], following coding practices most accurately described as 'hacking.' Products developed this way are often poorly architected, lack documentation..."*. He continues that, as soon as these startups attracted funding and customers, they need to grow and, hence, to establish all those procedures they skipped before to allow for a sustainable business. These extremes illustrate the field of tension in which educators must teach students: an extensive elaboration on standards for critical systems seems to be too much, but addressing the "hacking"-only dimension seems to be too little. Recent research confirms that companies use a multitude of different development approaches, which they combine in so-called *hybrid methods* [4], [5]. That is, in higher education, students should be prepared for this very situation, i.e., different development approaches should be subject to teaching.

Problem Statement and Objective: Given that it typically takes three to five years to educate students, it is hard for HEIs to catch up with industrial innovation cycles and, moreover, it is impossible to teach everything in just one course. Therefore, we analyze the current state of practice in software engineering education and compare it with industrial practice. Specifically, we consider whether software development frameworks, methods and practices as used in industry are present in HEI courses. Our study aims to draw a big picture providing support for educators to evolve their courses to strategically balance basic knowledge and industry demands.

Contribution: Based on an international online survey through which we collected information about 67 software en-

Thanks