UNIVERSITY OF RIJEKA
Faculty of Engineering

SOFTWARE FACTORY FOR STUDENT PROJECTS

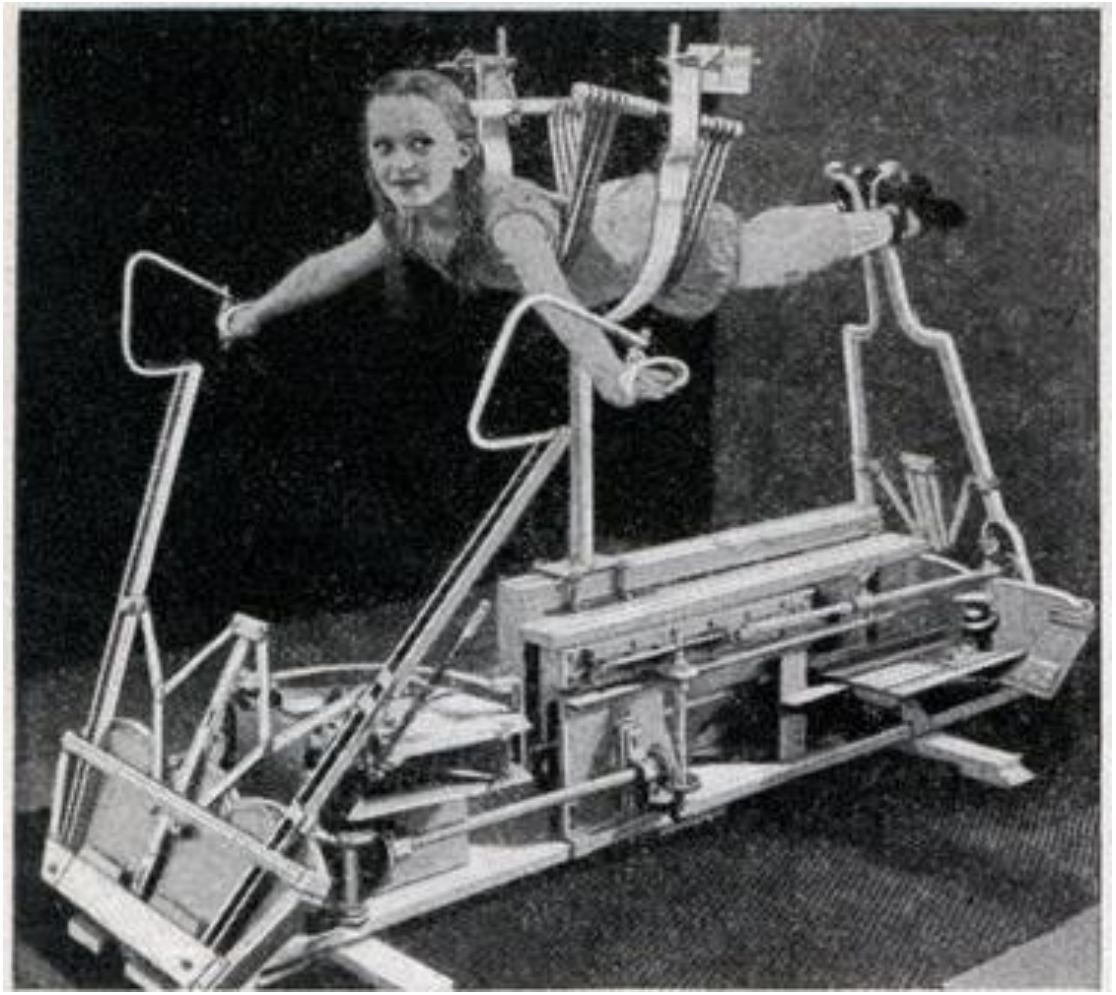
Tihana Galinac Grbac
Faculty of Engineering
University of Rijeka
Croatia

13th Workshop "Software engineering Education and Reverse Engineering", Bansko, Bulgaria, 26-31 August 2013

CHALLENGES IN TEACHING SOFTWARE ENGINEERING

- Motivation is the main vehicle for successful learning
- Main obstacles [SWFACT, SWFACT1] :
  - Lack of development visibility
  - Lack of uniform measures
  - Explosion of new technologies
  - Maturity of discipline (lack of discipline and repeatability)
  - Business need for reusability

HOW SHOLD WE TEACH SOFTWARE ENGINEERING?



Source: Swimming Students Learn Strokes From Machine Teacher, Modern Mechanix, Nov

LEARNING THEORIES [LT]

- **Learning by doing** - people learn task better by doing it than by hearing about it.
  - In majority of SWENG courses students organized in projects are developing some application
- **Learning through reflection** that is involving students into active discussion about applied theory
  - Students critical reflection on applied theory in practice, active dialog during lessons
- **Situated learning** based on learning by doing with focus on environmental factors such as product, context, culture, etc.
  - Industrial participant, large teams distributed across the globe, evolutionary projects

CURRICULUM GUIDELINES FOR PROGRAMS IN SOFTWARE ENGINEERING [GSEU, GSW]

UNIVERSITY OF RIJEKA
Faculty of Engineering

- Industry involvement
- Project based courses
- Attractive to students
- New technologies

FINDING THE BEST APPROACH

- Research question:
  - *'How to extend the boundaries of how educators, trainers, students and professional software engineers exchange and acquire software engineering knowledge and skills' [ICSE2014].*
- Study design:
  - 3 case studies: each representing one implementation scenario of software engineering course
  - Main driver of software engineering course is project
  - Students divided in teams 4-6
  - All student teams have same outputs to produce: project plan, requirement specification, design specification, code, test documentation, inspection reports, project report and presentation (using templates)
  - Weekly follow up with professor and assistant
- Evaluation:
  - After each case study students respond on questionnaire
  - Professor discussed learning outcomes with students
  - Final exam is used to confirm findings

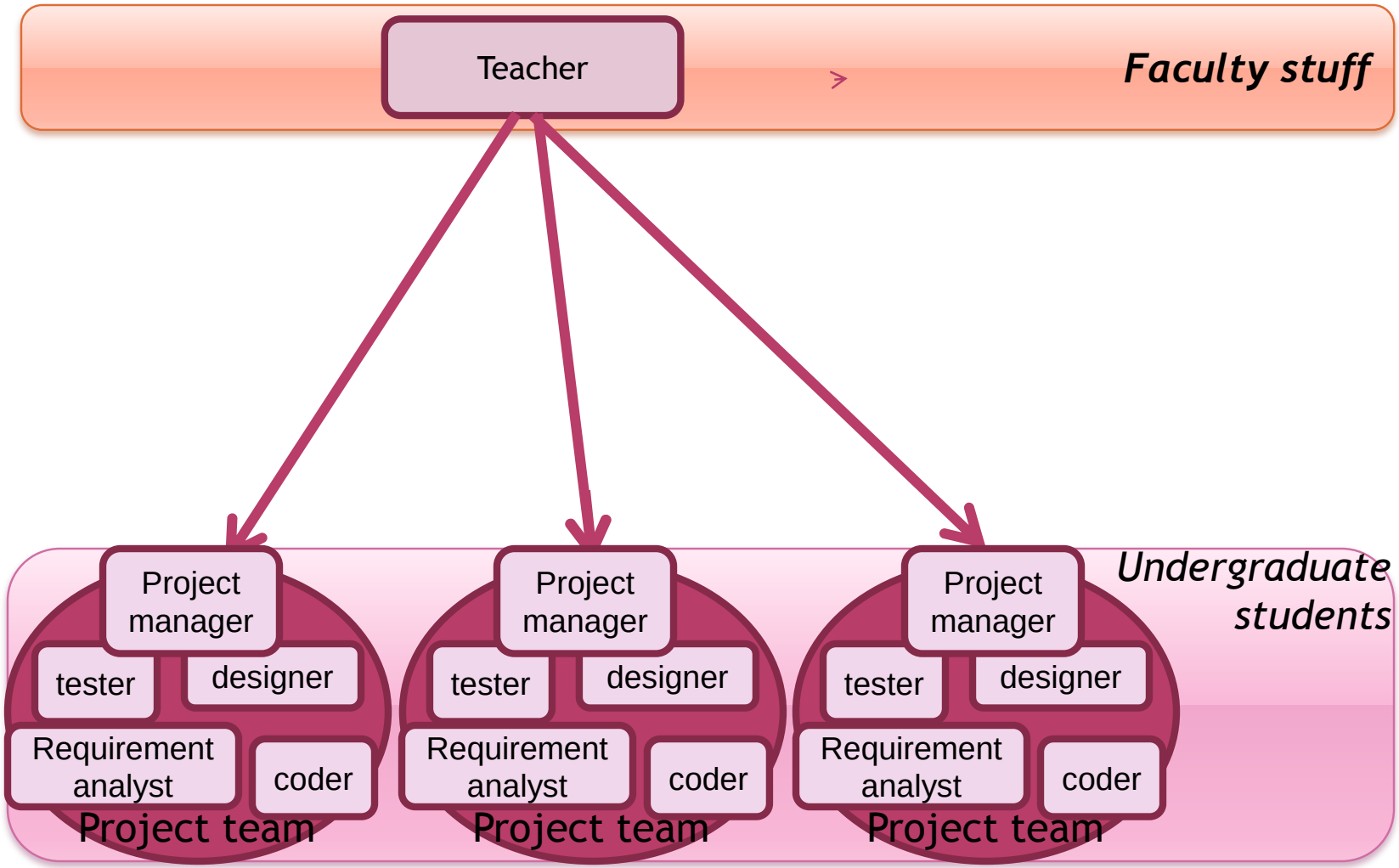
CASE 1

UNIVERSITY OF RIJEKA
Faculty of Engineering

Note that only courses with included software engineering knowledge are listed

Sem	Undergraduate study courses
I	Programming Computer Applications
II	**Software Engineering**
III	Computer Architectures Operating systems Algorithms and Data Structures
IV	Databases Computer Networks
V	Databases Computer networks Professional Practice
VI	Embedded Systems Web Applications Development Elective Project
	Graduate study courses
I	Advanced algorithms and data structures
II	**Software Engineering Management** **Project**

SOFTWARE PROJECT ORGANIZATION



STUDENT QUESTIONNAIRE

1-I fully DISAGREE with statement 5-I fully AGREE with statement	N	Mean	N	Mean
Relations between content of lessons and profession are clearly emphasized	43	4,47	52	4,5
New concepts are presented with practical examples	43	4,44	52	4,33
I was stimulated for independent work and critical reasoning	43	4,40	52	4,24
I was stimulated in active participation in lessons	44	4,39	52	4,2
I was motivated for gaining knowledge and learning the course content	42	4,26	52	4,2
My experience during this course was useful	44	4,25	52	4,41

STUDENT COMMENTS ABOUT COURSE

Positive side	Negative side
Lot of examples from our future working environment	This course would be more appropriate for later semesters, in 4 semester
I especially liked practical tasks and examination based on it, so I practiced the majority of course instead of learning by heart?	
Teamwork, project, object-oriented programming, Android	
Professional experience of professor	

Discussion with students and observations:
Project manager role is the most stressful and the most responsible role (in successful student projects PM did majority of work)
• Student experiences of student projects with Android are published in [Android]

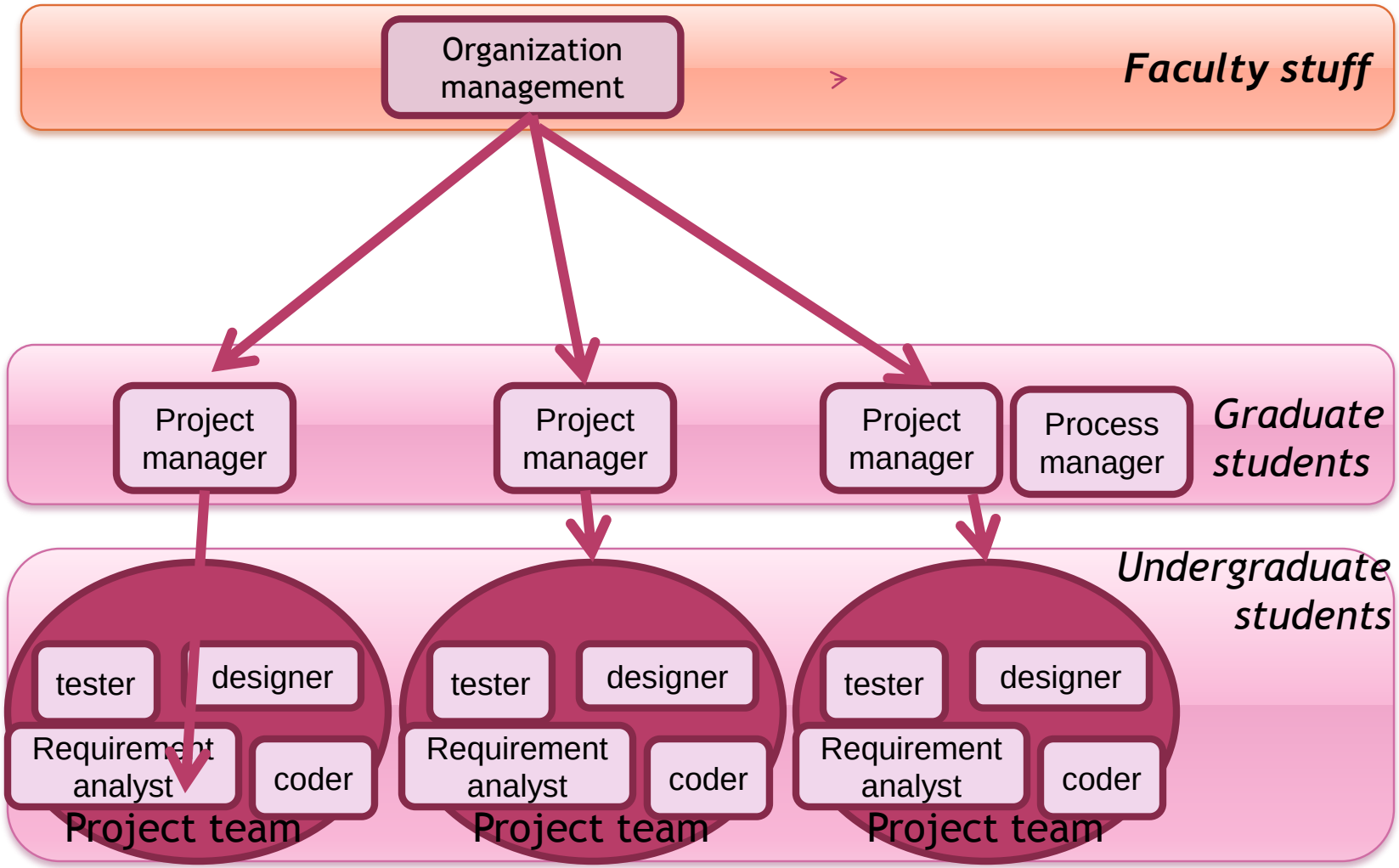
CASE 2

SOFWARE ENGINEERING @ RITEH

Sem	Undergraduate study courses
I	Programming Computer Applications
II	**Software Engineering**
III	Computer Architectures Operating systems Algorithms and Data Structures
IV	Databases Computer Networks
V	Databases Computer networks **Professional Practice**
VI	Embedded Systems Web Applications Development Elective Project
	Graduate study courses
I	Advanced algorithms and data structures
II	**Software Engineering Management** Project

BEST PRACTICE
ERICSSON SUMMER CAMP

SOFTWARE PROJECT ORGANIZATION



STUDENT QUESTIONNAIRE

1-I fully DISAGREE with statement 5-I fully AGREE with statement	N	Mean	N	Mean
Relations between content of lessons and profession are clearly emphasized	40	4,28	52	4,64
New concepts are presented with practical examples	40	4,23	52	4,64
I was stimulated for independent work and critical thinking reasoning	40	4,22	52	4,55
I was stimulated in active participation in lessons	40	4,00	52	4,36
I was motivated for gaining knowledge and learning the course content	40	4,10	52	4,27
My experience during this course was useful	40	4,13	52	4,27

STUDENT COMMENTS ABOUT COURSE - SOFTWARE ENGINEERING

Positive side	Negative side
All	Short introductiory basics in java and XML before project execution
Project for developing Android application	

STUDENT COMMENTS ABOUT COURSE - SOFTWARE ENGINEERING MANAGEMENT

Positive side	Negative side
Leading students from undergraduate study in projects	This course would be more appropriate for later semesters of undergraduate students
Practical examples	

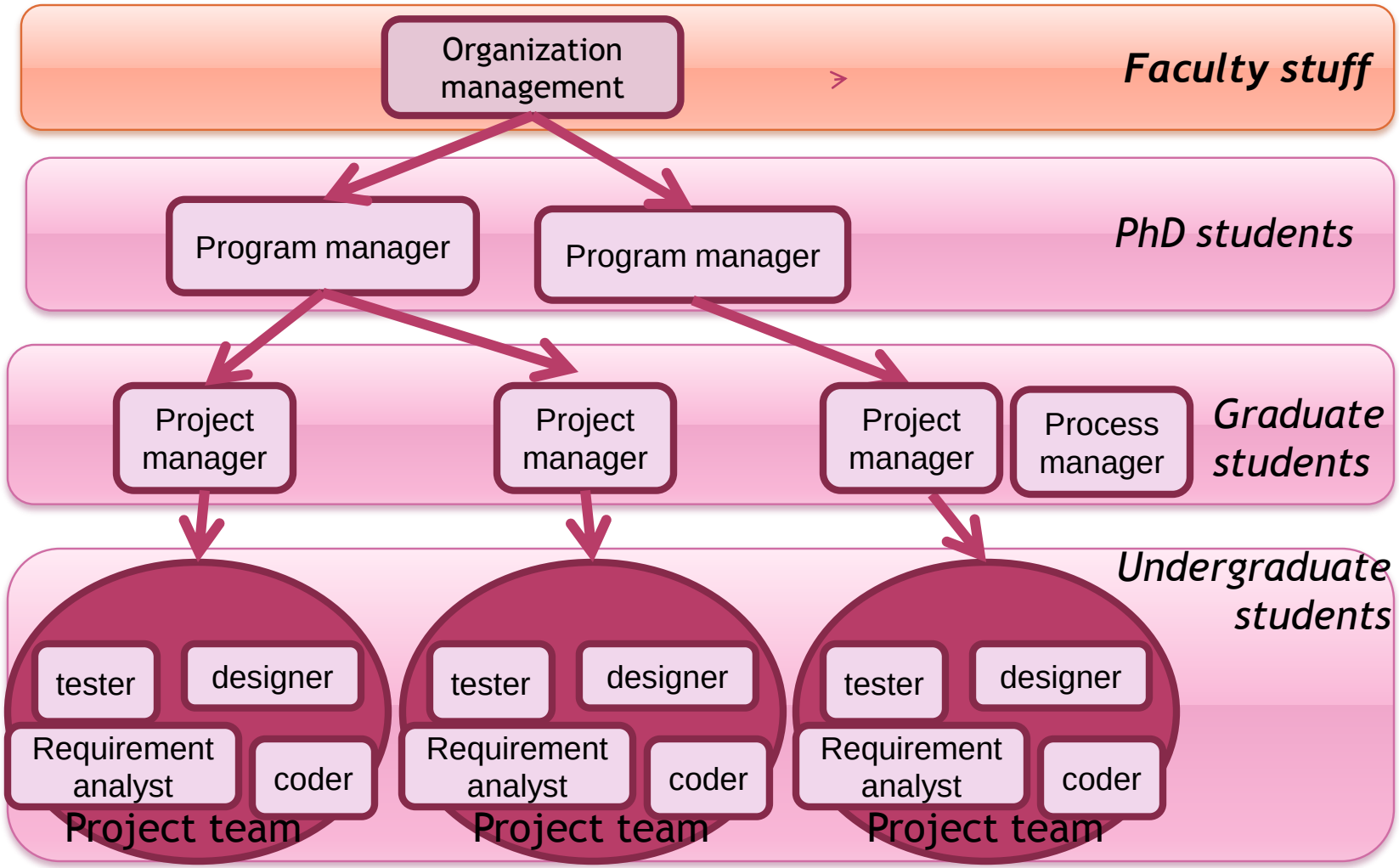
Discussion with students and observations:
Students do not understand concept of reusability. Very low usage of tools and techniques that helps with this issue.
Student with relevant industrial practice have shown better understanding and were more successful in projects.

CASE 3

SOFWARE ENGINEERING @ RITEH

Sem	Undergraduate study courses
I	Programming Computer Applications
II	~~Software Engineering~~ **Computer networks**
III	Computer Architectures Operating systems Algorithms and Data Structures
IV	Databases ~~Computer Networks~~ Software engineering
V	Databases Computer networks Professional Practice
VI	Embedded Systems Web Applications Development Elective Project
	Graduate study courses
I	Advanced algorithms and data structures
II	**Software Engineering Management** **Project**

SOFTWARE FACTORY ORGANIZATION [SF]



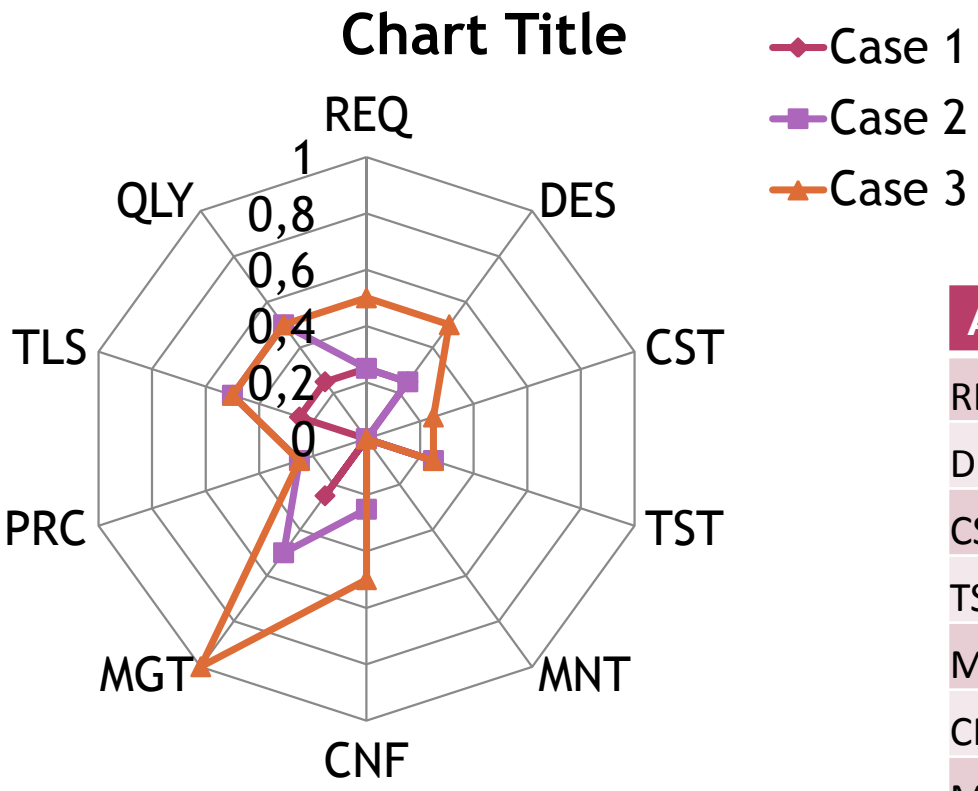
CASE 3. LEARNING OUTCOMES

- Students used to diminish importance of systematic planning of system
- Do not understand why systematic planning is needed, common saying was:
  - It would be much faster to code immediately this part of functionality, then describing it in detail
- Hard to convince students into meaningfulness of this task and that reflected on quality of outputs
- Conclusions: Students learn about software engineering principles but ignore to use them
- Majority of students respect only tools that allows them easy coding and fast solution

CASE 3. LEARNING OUTCOMES

- Reusability
  - We start to talk about need for configuration management tools

COVERAGE OF SWEBOK



Pt	Description
0	No coverage
1	Some coverage
2	Medium coverage
3	Significant coverage

Abb.	Description
REQ	Software requirements
DES	Software design
CST	Software construction
TST	Software testing
MNT	Software maintenance
CNF	Software configuration management
MGT	Software engineering management
PRC	Software engineering process
TLS	Software engineering tools and methods
QLY	Software quality

CONCLUSION

- Students reaction on models based on the selected learning theories were very positive
- Better organized project with higher complexity, involving more students turned to be positive practice
  - Student like projects,
  - Get more involved in theoretical lessons,
  - Wider set of knowledge areas is covered
  - Able to better understand lessons

REFERENCES

[LT] E. O. Navarro, A. van der Hoek, On the Role of Learning Theories in Furthering Software Engineering Education, *Software Engineering: Effective Teaching and Learning Approaches and Practices, IGI Global, 2008.*

[GSEU] ACM and IEEE (ACM/ IEEE Computer Society Joint Task Force on Computing Curricula). 2004. Software engineering 2004: Curriculum guidelines for undergraduate degree programs in software engineering. <http:/ /www.acm.org/education/curricula-recommendations>

[GSE] ACM and IEEE (ACM/ IEEE Computer Society Joint Task Force on Computing Curricula). 2004. Software engineering 2004: Curriculum guidelines for graduate degree programs in software engineering. <http:/ /www.acm.org/education/curricula-recommendations>

[GSwE2009] Pyster, A. (Ed.), Graduate Software Engineering 2009 (GSwE2009): Curriculum Guidelines for Graduate Degree Programs in Software Engineering, Integrated Software & Systems Engineering Curriculum Project, Stevens Institute of Technology, September 30, 2009. <http:/ /www.gswe2009.org>.

[SF] J. D. Tvedt, R. Tesoriero, K. A. Gary. The Software Factory: Combining Undergraduate Computer Science and Software Engineering Education, *pp.633-642. IEEE Computer Society, (2001)*

[SWFACT] M. A. Cusumano. The Software Factory: A Historical Interpretation. *IEEE Software 6(2):23-30 (1989)*

[SWFACT1] H. Bratman and T. Court, The software factory, IEEE Computer, 1975, pp.28-37.

[Android] R. Grgurina, G. Brestovac, T. Galinac Grbac: Development environment for Android application development: An experience report. MIPRO 2011: 1693-1698

QUESTIONS?