Universiteit Leiden
Campus Den Haag

Centre for Innovation The Hague

Academic Education of Software Engineering Practices

Towards Planning and Improving Capstone Courses Based upon Intensive Coaching and Team Routines

CSEE&T 2013, San Francisco, CA

Christoph Johann Stettina <stettina@liacs.nl>
Zhao Zhou <zzhou@liacs.nl>
Thomas Bäck <baeck@liacs.nl>
Bernhard Katzy <bkatzy@liacs.nl>

Dit project is mede gefinancierd met steun van het Europese Fonds voor Regionale Ontwikkeling van de Europese Unie.

Christoph Johann Stettina

- MSc Computer Engineering (Dipl.-Inf.)
- MA Project Management
- Past: 4 years R&D engineering at Nokia
- Now: Process Coach, PMO & PhD Candidate

Interest: R&D Management

- Knowledge Creation and Innovation
- Project Management
- Teamwork

Introduction

- Academic Education of SE Practices
- Practical education & academic reflection

What we know

- Learning stages: declarative and procedural (Anderson, 1982)
- Students struggle with process
- Learning in team works well (Richards, 2009)

Agile Practices as Team Routines

- Learning through repeated interaction
- Support both learning stages
- Agile: SE practices in a single framework (Hazzan and Dubinsky, 2007)

Objectives

- Agile practices provide a framework to address procedural knowledge, but how to make it academic?

Research Questions

1. Course: How can we plan software engineering courses so that using agile process improvement techniques we can improve education and contribute to research at the same time?

2. Experiment: What are the implications of individual intra-team stand-up meetings on coaching success and team satisfaction compared to bigger inter-team stand-up meetings?

Study Context: SDPM Course

- Master-level Capstone: SE & PM
- Real-world: From idea to demonstrator
- Declarative knowledge: Regular Lectures
- Procedural knowledge: Intensive Coaching

Coaching Routine

- Stand-up Meetings (5-15min)
- Iteration Reviews
- Guide, feedback: Process, Content, Teamw.

Methodology: Embedded Experiment

- 30 students, 6 iterations, 6 teams, 2 groups
- SIndividual: Individual Stand-up meetings
- SUnited: Collective Stand-up meetings
- Better knowledge transfer and interaction?

Project Planning and Initial Design
02-02-2011: (Session 1) Introduction *16-02-2011*: (Session 2) Project Bid *22-02-2011*: (Session 3) Project Plan
Development
29-02-2011: Sprint 1 *07-03-2011*: Sprint 2
Delivery
15-03-2011: System Demonstration and Trade Fair

Methodology: Data Collection

Project Questionnaire

This research questionnaire is anonymous and answers will not affect your grades. Please answer honestly.

Scale: 1-Completely dissatisfied, 2-Mostly dissatisfied, 3–Somewhat dissatisfied, 4-neither satisfied or dissatisfied, 5-Somewhat satisfied, 6-Mostly satisfied, 7-Completely satisfied

Date: [_____], Group: [___]

How satisfied are you with the project? (This current project in this course and within your project group)

Completely dissatisfied ☐ ☐ ☐ ☐ ☐ ☐ ☐ *Completely satisfied*

How satisfied are you with the amount of work?

Completely dissatisfied ☐ ☐ ☐ ☐ ☐ ☐ ☐ *Completely satisfied*

How satisfied are you with the teamwork in your team?

Completely dissatisfied ☐ ☐ ☐ ☐ ☐ ☐ ☐ *Completely satisfied*

How satisfied are you with the innovativeness in your team?

Completely dissatisfied ☐ ☐ ☐ ☐ ☐ ☐ ☐ *Completely satisfied*

How satisfied are you with the information exchange in this project? (In general, expectations, requirements, issues..)

Completely dissatisfied ☐ ☐ ☐ ☐ ☐ ☐ ☐ *Completely satisfied*

Comment(s): _____

Qualitative data:

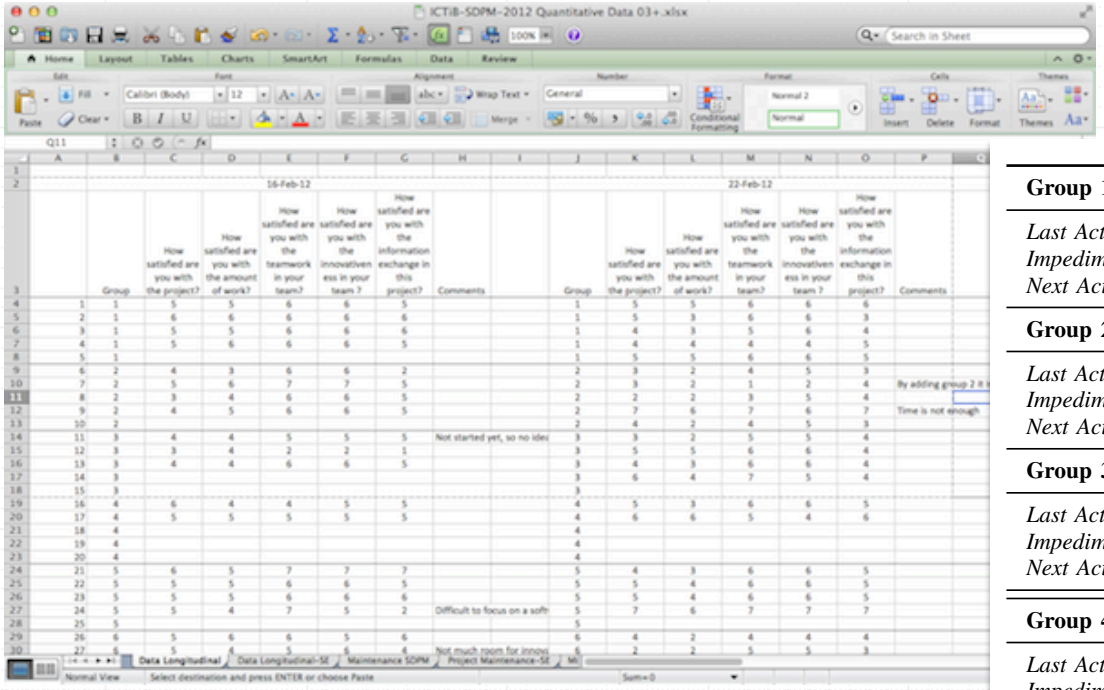
Observations, informal interviews, artifacts

Quantitative questionnaire (weekly):

Comparable Likert scale data on satisfaction:

- How satisfied are you with the project?,
- How satisfied are you with the teamwork in your team?
- How satisfied are you with the information exchange in this project?

Data Samples:



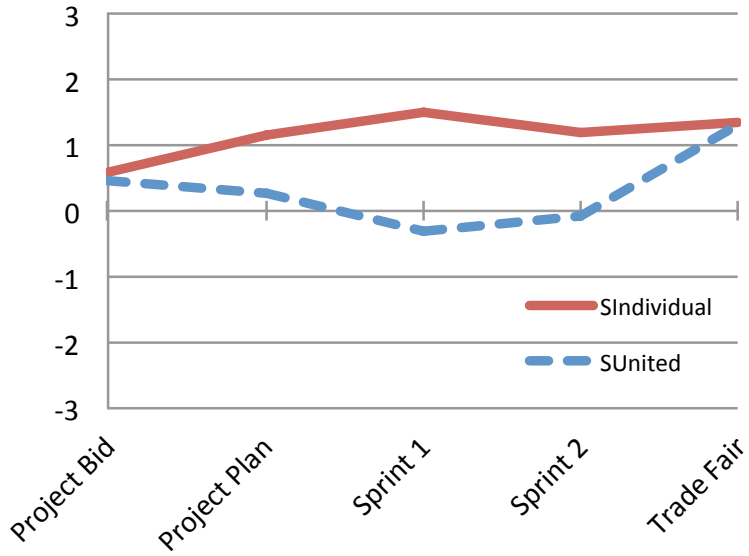
Longitudinal data Excel

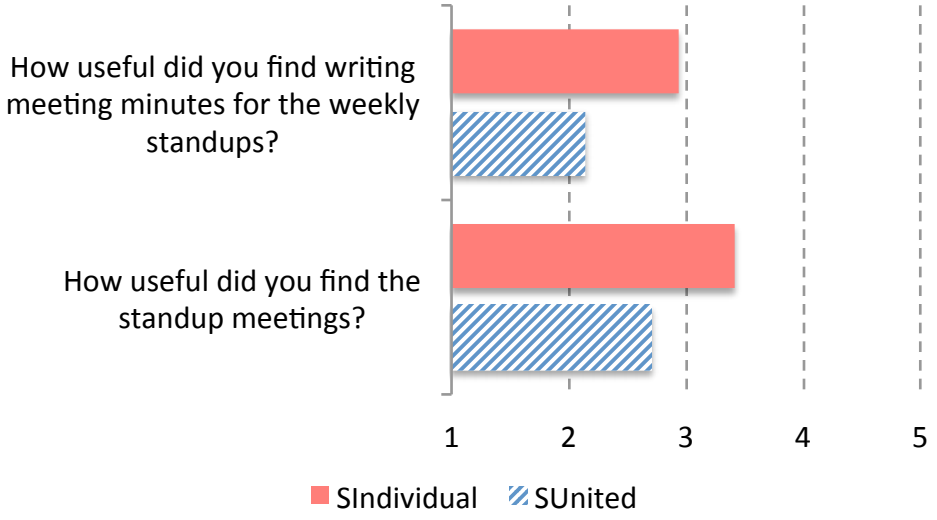
30 (students) x 6 (sprints)
Allows t-test for significant difference!

Stand-up notes

Group 1	
Last Actions:	Project plan, Kentico CMS
Impediments:	-
Next Actions:	Implementation, easy requirements first
Group 2	
Last Actions:	Project plan, UML Sequence Diagram
Impediments:	Time
Next Actions:	Functional Design, Implementation, Technical Design
Group 3	
Last Actions:	Project plan
Impediments:	-
Next Actions:	Interface Prototype
Group 4	
Last Actions:	Project plan, exploring platform -¿ requirements
Impediments:	Tight schedule, balance between documentation and development
Next Actions:	Page layout, reduce text main page, OpenStudy
Group 5	
Last Actions:	Project plan, decided on key deliverables, decided on local and stable demonstrator
Impediments:	Final constraint: time, C only known to two people
Next Actions:	Follow project plan, Divide work, Start on monday
Group 6	
Last Actions:	Project plan, High level software specifications; Defined implementation strategy with Java
Impediments:	TIme pressure, Platform unknown, Need to learn
Next Actions:	Work on the demonstrator, set up development environment, Need to verify if Java is the best option for implementation

Results: Experiment





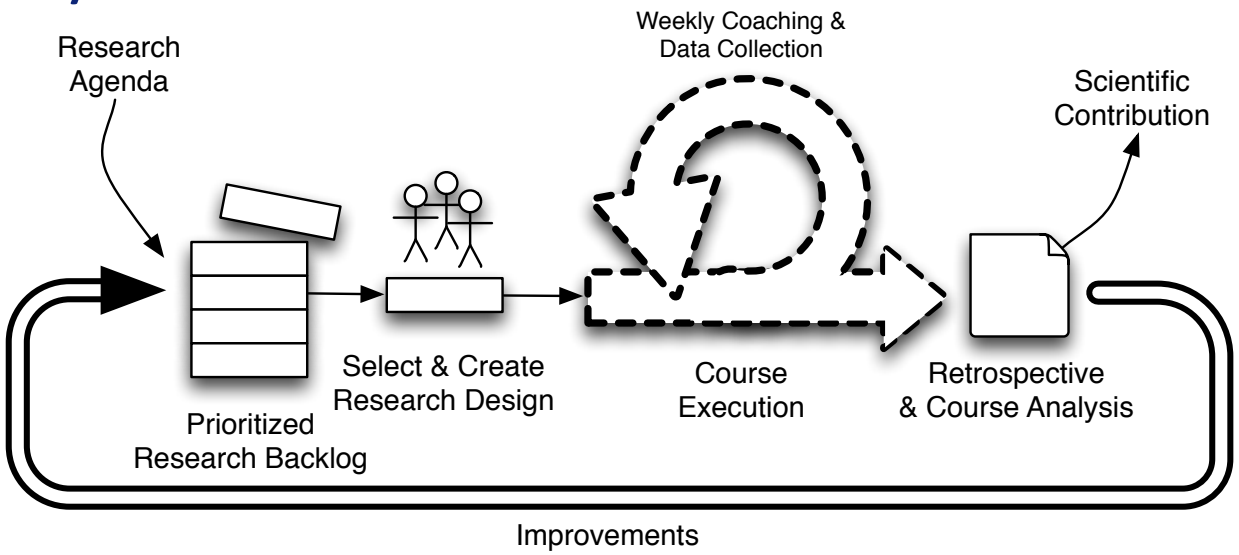
Sindividual: More satisfied, longer more elaborated discussions
SUnited: Waiting for the next group to finish, groups coming late

Significant: Satisfaction with project & information exchange
Not significant: Satisfaction with teamwork

Discussion RQ1:

How can we plan software engineering courses so that using agile process improvement techniques we can improve education and contribute to research at the same time?

- Intensive coaching using notion of team routines
- Explore concrete SE techniques in context (Collaboration, Google Docs, Dropbox)
- Intensive coaching justified by contribution to science and PhD maturity



Discussion RQ2:

What are the implications of individual intra-team stand-up meetings on coaching success and team satisfaction compared to bigger inter-team stand-up meetings?

- Individual groups more focused and on time
- Possible knowledge gain overridden by less satisfaction
- Team should feel comfortable for a good knowledge exchange and interaction
- Standups: Identification of impediments early on (Sharp and Robinson, 2007)

Conclusions:

Course

- Our experience balancing practical coaching and academic reflection
- Planning and improving capstone courses based on intensive coaching and notion of routines
- Contributes to student and educator/PhD maturity

Experiment

- SUnited: Knowledge gain overridden by lass satisfaction
- Intensive coaching shorter and more appealing

Data Collection Method

- Approach allows quantitative data collection even with smaller groups (longitudinal)

Conclusions → Future Work

Increasing importance of routines in creating knowledge

→ How improve to study routines in-crass?

→ How to visualize/model the practices?

Collaboration amongst coaches in bigger groups

→ How do these results relate to bigger group size?

→ How to embed peer-assessment?

→ How to address different student learning types?

Questions ?

Thank you for your attention!

stettina@liacs.nl

References

J. R. Anderson. Acquisition of cognitive skill. Psychological Review, 89:369–406, 1982.

O. Hazzan and Y. Dubinsky. Why software engineering programs should teach agile software development. SIGSOFT Softw. Eng. Notes, 32(2):1–3, Mar. 2007.

D. Richards. Designing project-based courses with a focus on group formation and assessment. Trans. Comput. Educ., 9:2:1–2:40, March 2009.

H. Sharp and H. Robinson. An ethnographic study of xp practice. Empirical Softw. Engg., 9(4):353– 375, Dec. 2004.

H. Taylor. Role-play cases for teaching interviewing skills in information systems analysis. In Proceedings of HERDSA Annual International Conference, pages 1–9, 1999.