

# Unreined Students or Not: Modes of Freedom in a Project-Based Software Engineering Course

Øystein Nytrø, Anh Nguyen Duc\*, Hallvard Trætteberg,  
Madeleine Lorås, Babak Amin Farschian

\* University of South Eastern Norway  
Norwegian University of Science and Technology

# Overview

- Introduction
- Research questions
- Research methodology
- Course settings
- Results
- Recommendations

# Introduction

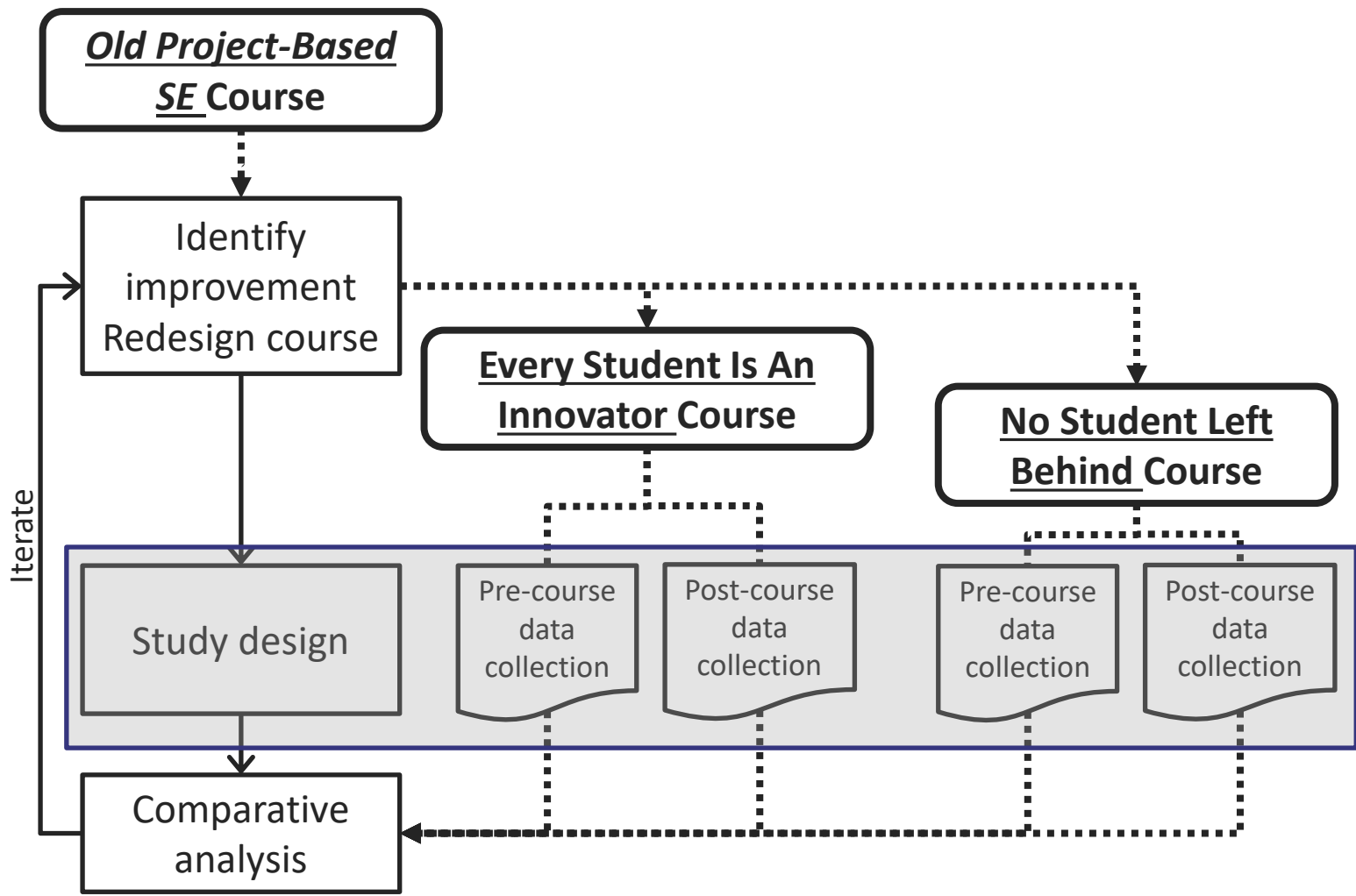
- Educating software engineering (SE) students is challenging:
  - Balance of ensuring practical competence while still providing future-proof research-based knowledge
  - Balance of lucid research-based lectures and running a smooth and large-scale lab operations
- Project-based courses motivate students to collaborate, explore, take responsibility and learn for themselves
  - Educators need the right levels of constraints, theory, tools, and infrastructure for the course

# Course setting

- Large class size
- Progressive enhancement of curriculum
- Project-based course
- Trained, but lower-grade TA staff

# Research Questions

- *RQ1: What dimensions of students' choices are possible to accommodate in a large, project-based SE course?*
- *RQ2: What are the strengths/weaknesses associated with such freedom?*



# Overview of courses' schedule

	Week no.	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	W16	W17	
<b>ESIAI</b>	<b>Lectures</b>	Process-SEMAT, MobileD	Process - Metric	Architecture	Requirement	Software quality	Testing	Project management	SEMAT (2)	Secure SE							
	<b>Process</b>				Sprint 0	Sprint 1		Sprint 2		Sprint 3		Sprint 4			Sprint 5		
	<b>Deliverables</b>			Project plan Backlog Poster			Sprint 1 demo		Sprint 2 demo		Sprint 3 Demo		Team practice card	Final delivery Video Report			
	<b>Assessment</b>			100%													
<b>NSLB</b>	<b>Lectures</b>	Intro, Tech	Scrum, CI, Req, GitLab	VC, Testing, Framew.	Agile meth., Testing	Safety, Quality, CI	Cardiology, SO, servers	Enterprise arch., UML	Archi., Gamific., Security								
	<b>Process</b>			Sprint 1				Sprint 2				Sprint 3					
	<b>Deliverables</b>			GitLab config	Project plan	MC Test					Demo 1				Demo 2	MC Test	
	<b>Assessment</b>			10%	10%	10%	10%			10%	10%			10%	20%	10%	

# Every Student is an Innovator



Dimensions	Fixed	Flexible
<b>Project setting</b>	Project theme Report template Overall architecture	Project idea Product backlogs Detailed architecture
<b>Process</b>	Number of Sprints Duration of Sprints Delivery of each Sprint Weekly supervision meeting	Process metrics Adopted practices Team communication and meeting Quality assurance
<b>Technology</b>	Version Control System	Programming language (frontend, backend) Servers, Database
<b>Supervision</b>	Weekly meeting Delivery	N/A
<b>Assessment</b>	Video-presentation and report eval. at end of course	N/A



# No Student Left Behind

- Fixed project problem
- Template code in GitLab
- Assigned group TA's, frequent deliverables
- Continuous assessment under senior supervision

<b>Dimensions</b>	<b>Fixed</b>	<b>Flexible</b>
<b>Project setting</b>	Overall architecture and server functionality	Domain, application, user functions
<b>Process</b>	Scrum with sprints	Roles, organization
<b>Technology</b>	GitLab support, setup and working code templates, example service stack setup.	Students could decide on prog. language, version control, service stack etc. Most would follow provided examples and templates.
<b>Supervision</b>	Given according to defined deliverables, process and content requirement. TA's trained in templates and examples.	Ample resources and agile and eager staff. Full flexibility in use of staff with time, location and medium.
<b>Assessment</b>	Continuous, structured, assessment of various types of deliverables. Individual multiple choice tests.	Senior staff would receive complaints about unfair deliverable evaluation, and could intervene if valid complaint.

# Answering RQ1 – Possible dimensions to accommodate

- all of the dimensions (1) project setting, (2) process, (3) technology, (4) supervision, and (5) assessment allow and need a certain level of flexibility to cope with the variety of projects, students' experience and TA's experience
- freedom of choice is both inspiring and challenging but must be balanced with precise control in order to reach learning objectives and maintain fairness.

**Answering RQ2 –  
strengths/  
weaknesses  
associated with  
such freedom**

	<b>Strength</b>	<b>Weakness</b>
<b>No Student Left Behind</b>	Thorough and predictable coverage. High, average level of competence. Effective TA involvement.	Too many details that may be relevant to everybody. Hard work and less fun, negative appreciation of innovation.
<b>Every Student Is <u>An</u> Innovator</b>	Fun and motivating. Lifelike and realistic learning. Exposure to innovative thinking in teams.	Uncertain individual learning outcomes. Hard to control resource use. Little cross-team communication. Ineffective TA's. Overwhelming.

# Recommendations

- Freedom of technology and method choices reduce the value and validity of TA aid and assessment.
- Freedom of problem selection increases involvement, and time spent.
- Freedom of team arrangement increases team competitiveness and potential student lockout or team failure.
- Freedom (lack) of precise deliverable content and form makes assessment non-transparent and subjective.