

Using eXtreme Programming in a student environment

A case study



Karlstad, the 10th of June

Agenda



- Motivation
- eXtreme Programming
- Goal of the dissertation
- The evaluated course
- XP Practices
- Analysis model
- Conclusion building
- Significant results
- Conclusion
- Outlook

Motivation



- Increasing complexity of software projects
- Increasing time to market pressure
- Shorter live cycles
- Productivity needs to be increased
- Agile methods have been developed
 - RUP, eXtreme Programming, Scum, etc...
- Enable employees to choose

eXtreme Programming



- Defined by Kent Beck in 1999
- Modified by Kent Beck in 2004
- Based on values, principles and practices
- Social and professional aspects
- Support by the psychologist Cynthia Andres
- Bug free development
- Short release cycles
- Customer becomes architect

Goal



- Is it possible to „teach“ agile methods?
- What methods can be taught?
- What is a good way of teaching agile methods?



The evaluated course

- 2nd year course in computer science
- Programming was known (4 courses)
- 10 students
 - XP ~ Gladwell max of 12
 - KAU 10 is typical
- Not all (15 out of 24) practices
- Goal: virtual story wall

XP Practices



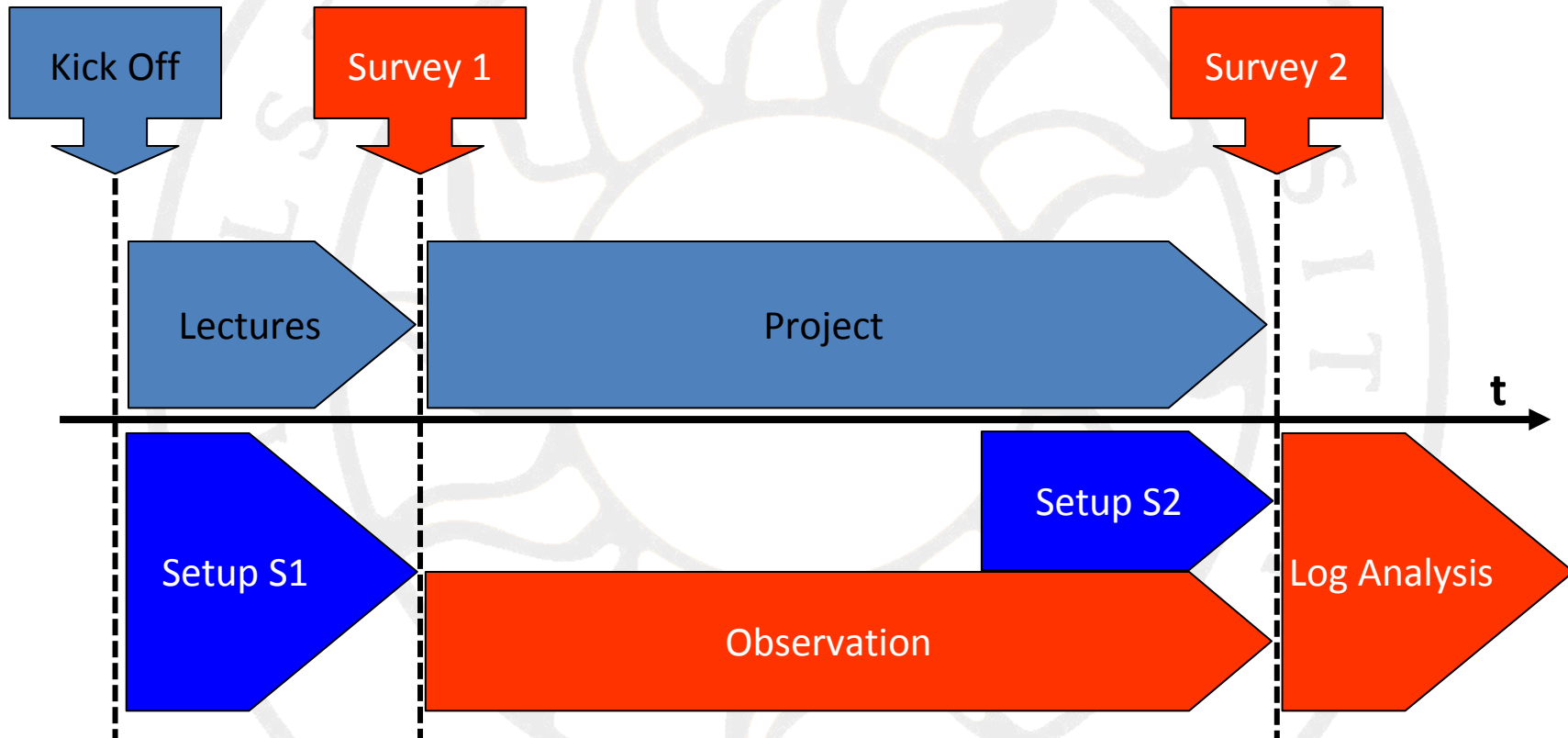
Taught

- Sit Together
- Informative Workspace
- Energized Work
- Pair Programming
- Stories
- Weekly Cycle
- Slack
- Ten-Minute Build
- Continuous Integration
- Test-First Programming
- Incremental Design
- Shared Code
- Code & Test
- Single Code Base
- Negotiated Scope Contract

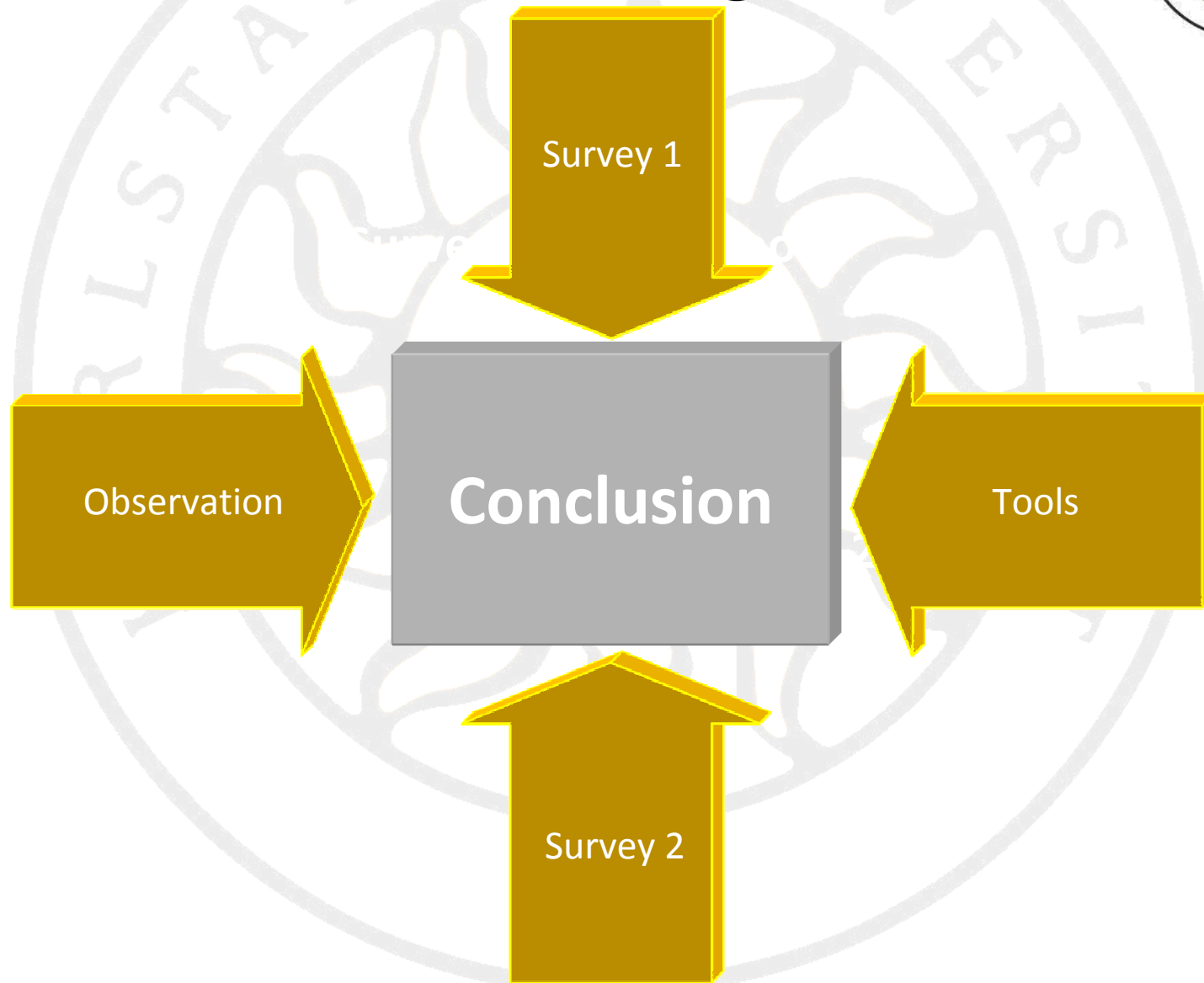
Not taught

- Real Customer Involvement
- Incremental Deployment
- Team Continuity
- Shrinking Teams
- Root-Cause Analysis
- Shared Code
- Code & Test
- Single Code Base
- Daily Deployment
- Negotiated Scope Contract
- Pay-Per-Use

Analysis model



Conclusion building



Significant results



- Practices not well applicable
 - Test First Programming, Slack, Informative Workspace, Negotiated Scope Contract
- Practices not understood
 - Negotiated Scope Contract, Slack

Significant results



- Discrepancy between subjective view and reality
- Project specification
 - Code & Test
- Project size caused misleading effects
 - Ten-Minute-Build & Continuous Integration

Conclusion



- Agile methods can be taught at universities
- Advantages must be carried out
- The concept based on theory & practical use is good but offers room for improvement
- The idea of cycles for lectures and practical use seems to be useful
- Role game meets its borders (e.g. Negotiated Scope Contract)

Outlook



- Scrum ...
- Further course evaluations
- Control group comparison
- Different agile methods
- Distributed eXtreme Programming

Opposition



Zak, your turn

Backup

