

# 10 Years of Agile Lab Courses for International Students

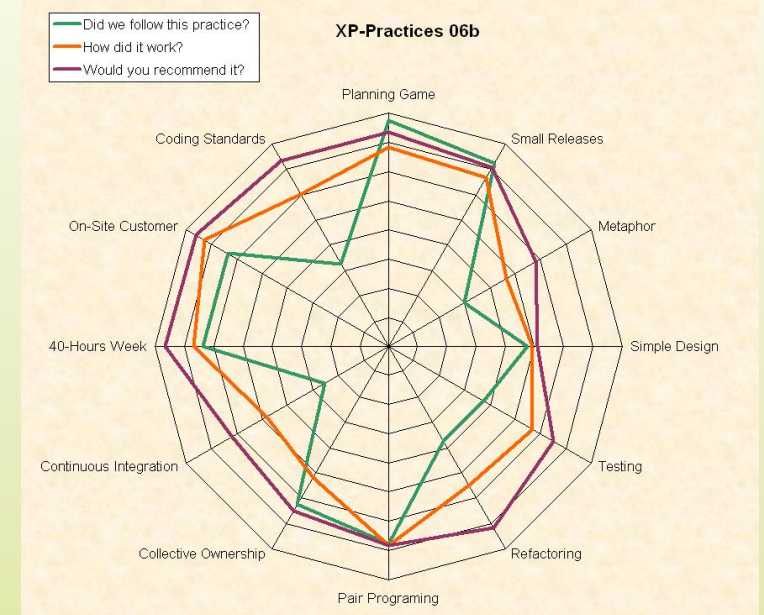
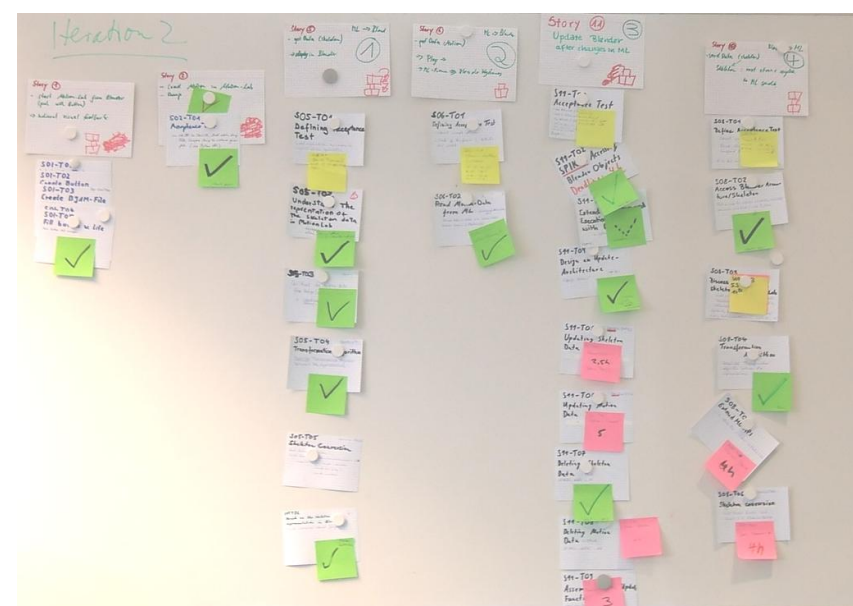
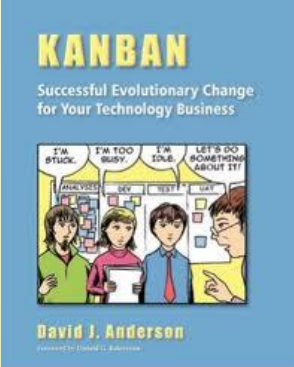
## Eclipse Plug-Ins for Developers



Addition of the Kanban Method  
<http://youtu.be/kcEX31U4MFk>



Feedback from Matthias Bohlen  
(<http://mbohlen.de/>)



## Collaboration with other departments

2010cn: "Mafia Game"  
[Nanjing, China] (We consulted only)



2011b: Cultivate goes Android

2009a: Improved Cultivate

2008a: Improved UI for JTransformer

2007cn: Editor for ad-hoc Network Simulations / Requirements Management Tool [Nanjing, China]



2007a: Integration of an Animation Studio & Motion Library



## Context Sensitive Mobile Applications

2012b: GeoQuest

2010b: Context Sensitive Adaptation of Webpages

2009b: Enhanced Mobile Gaming (YouAnt)

2008b: Context Sensitive Mobile Gaming (Scotland Yard to go)

2007b: Context Sensitive Mobile Gaming (Scotland Yard to go)

2006b: Context Sensitive Mobile Navigator (CSI Navigator)

2005b: Context Sensitive Mobile Application (CSI PimPro)

Copyright für den Text  
bei der GI. Erschienen in  
der Veröffentlichung, der  
GI-Jahrestagung, 2013.  
SAG WAS Workshop

## References

[MSK04] H. Mügge, D Speicher und G. Kniessel. Extreme Programming in der Informatik-Lehre - Ein Erfahrungsbericht. In Informatik 2004 – Beiträge der 34. Jahrestagung der GI, Lecture Notes in Informatics, 2004.

[NIS13] J. Nonnen, J. P. Imhoff und D. Speicher. Kanban im Universitätspraktikum - Ein Erfahrungsbericht. In Andreas Spillner und Horst Lichter, Hrsg., SEUH, Jgg. 956 of CEUR Workshop Proceedings, Seiten 91–98, 2013.

[Roy70] W.W. Royce. Managing the development of large software systems: concepts and techniques. Proc. IEEE WESTCON, Los Angeles, Seiten 1–9, August 1970. Reprinted in Proceedings of the Ninth International Conference on Software Engineering, March 1987, pp. 328–338.

## Reflective Improvement for Responsible Students

Software of realistic complexity requires a broad range of knowledge. We leverage on the number of students by asking everyone to become an expert in one area. The seminar, a wiki as a knowledge base and consequent pair programming allows the knowledge to expand into the team. The students are offered the responsibility to estimate the effort a certain functionality requires. With our support they learn that they can accept this responsibility. Planning Poker during planning is very helpful in this respect and some progress tracking gives feedback about the quality of their estimates. Daily stand-up meetings make sure that problems become obvious early and everyone shares the latest insights that are of value to the team. Each iteration ends after the presentation to the customer with a retrospective. In a respectful atmosphere everyone can discuss the challenges freely and actions addressing them can be found.

## What did we learn?

The students are typically very satisfied with the course, praise the friendly and productive atmosphere and are surprised by the improvement of their skills. Nevertheless we are not always successful from the start, so that we can offer as well some lessons learned: There need to be some breadth in the product so that two or more somewhat independent functional areas can be addressed by the students in parallel. Setting up a testing framework for a complex technology is beneficial and can be supported but not completed by a single course of four weeks. There is a limit to what can be estimated appropriately. If the team is in doubt that something takes more than two days it is probably too big to be estimated. Advocating that finishing functionality should be favored over investing into new one is not enough as the final integration seems too tedious and challenging. Visualizing the workflow and placing limits on the work in progress finally solved this issue for us [NIS13].

20.09.13  
in Bonn

Schedule for Agile/XP-Lab @ b-it/International Program of Excellence (First Iteration) (as planned on 27.08.2008)						
	Wednesday	Thursday	Friday	Monday	Tuesday	Wednesday
9:00	Breakfast	Breakfast	Breakfast	Breakfast	Breakfast	Breakfast
9:15						
9:30	"Stand up"	New Stories	Stand up	Stand up	Stand up	Stand up
9:45	Introduction		Development	Development	Development	Development
10:00	Vertical Slice					(and Preparation of the Presentation)
10:15		Estimation				
10:30						
10:45						
11:00						
11:15						
11:30						
11:45		Release plan				
12:00		Task definition				
12:15						
12:30						
12:45						
13:00	Lunch	Lunch	Lunch	Lunch	Lunch	Lunch
13:15						
13:30						
13:45						
14:00	"Presentation"	Task definition	Seminar	Seminar	Seminar	Development
14:15						
14:30	XP Game		Stand up	Stand up	Stand up	Presentation
14:45			Development	Development	Development	
15:00						
15:15						
15:30						
15:45						
16:00						
16:15						
16:30						
16:45						
17:00		Stand up				
17:15		Development				
17:30						
17:45	Wrap up					Iteration Review

Legend in Keywords	
Recreation	
Breakfast	Shared, agreed on participation
Lunch	Usually shared in a cafeteria near by
Customer Negotiation	
New Stories	Presented by the customer
Release plan	Selected and ordered by the customer
Presentation	Developers present implemented stories
Planning	
Estimation	Planning Poker, Discussions
Task definition	Definition, Estimation, Commitment
	Peer review of task definitions
Bootstrapping the Process	
Introduction	Give overview of the lab and the day
Vertical Slice	Code simple application through all levels
XP Game	The game to learn the planning game
Being Self-Adapting	
"Stand up"	Welcome each other, Explain Stand up meetings
"Presentation"	Present the vertical slice
Wrap up	Recap what was learned/observed
Stand up	Yesterday, Today, Obstacles? Burndown!
Seminar	Ruby, ObjectiveC, Code Reviews, Techniques, Reports
Review	Keep These, Problems, Try These
Acting	
Development	Tasks, Spikes, Design, Communication
	Pair Programming, Simplicity, High Code Quality
	Test-First, Code Coverage, Continuous Integration
	Even more seminars if required
	Stand up on demand
	Acceptance tests defined (and automated)
	Introduction into tracking
Practices that are special to this iteration are set in <i>italics</i> . Practices that are of special importance are set in <b>bold</b> .	
© 2007-13, university of bonn - Institute of Computer Science III	

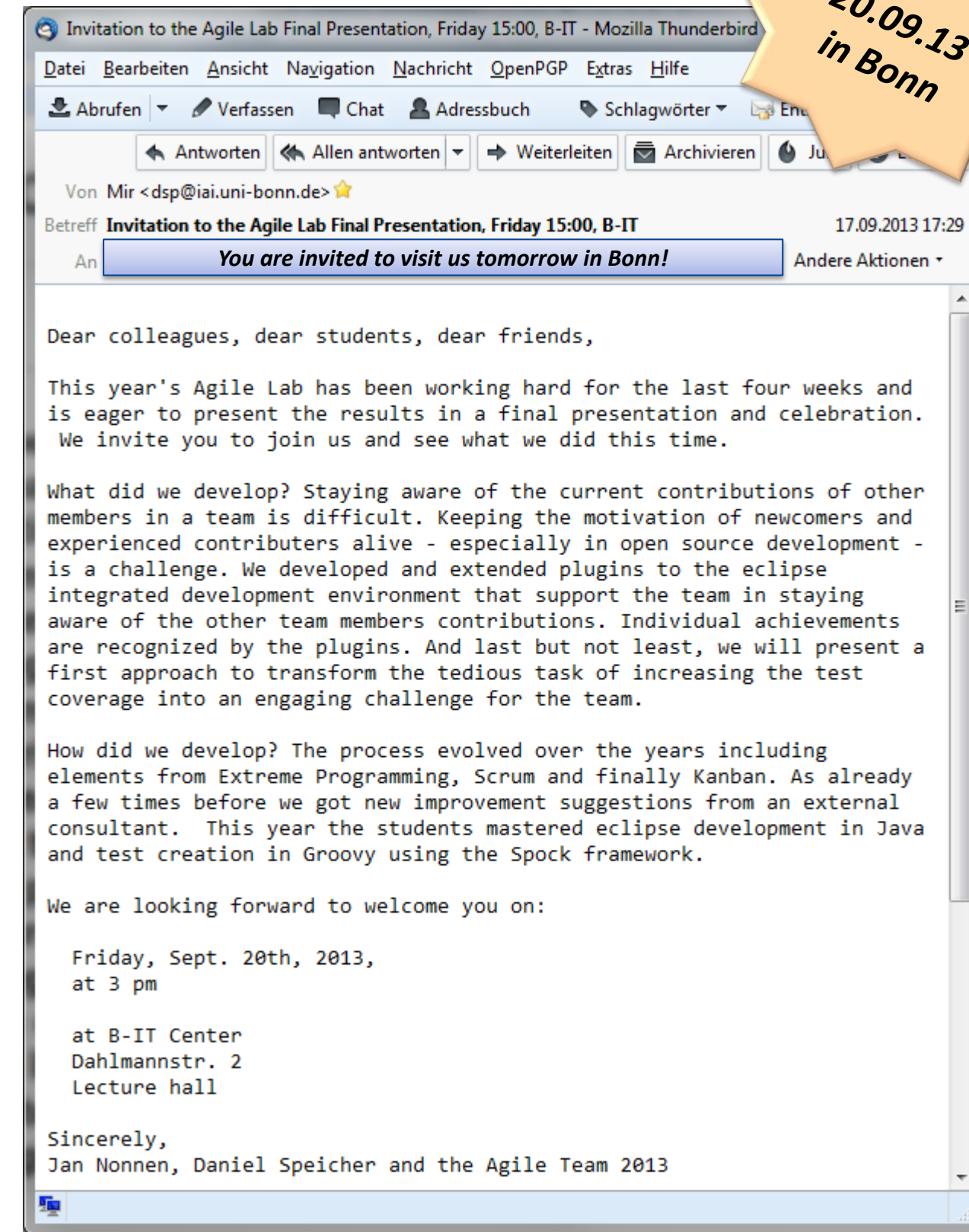
2005a: Visual Tool for Refactoring to Pattern (Cultivate, PatchWork)

2004b: Program Analysis by Logic Meta Programming (JTransformer, Cultivate)

2004a2: Tool Support for Pattern Management (PatchWork)

2004a1: Synchronized Logic Representation of Java Code (Jtransformer)

2003b: Improved Editor for Conditional Transformations (ConTraCT)



Some more foundations of our labs:

XP is [... an] attempt to reconcile humanity and productivity (R. Jeffries, Extreme programming explained)

Handle so, daß du die Menschheit, sowohl in deiner Person, als in der Person eines jeden andern, jederzeit zugleich als Zweck, niemals bloß als Mittel brauchst.  
(I. Kant, Grundlegung zur Metaphysik der Sitten)