Agile Methods

Ship Happens! - using our practical approach to developing great software

Dr. Till Bay
Geschäftsleiter

Dr. Michela Pedroni
Projektmanager
who we are
Company Profile.

what we do
Software Systems.

how we work
Our Process.
GOALS: you will learn...

what agile methods are

how they can be put to practice

that agile methods ≠ no process
About Comerge AG

**In 2008**
Founded by Till Bay and Bernd Schoeller.

- Official ETH Spin-Off
- Constant growth in the last 3 years
- Currently 12 employees
About Comerge AG

> 8 software engineers (ETH Zürich, FH)
> 1 Art Director / Designer
> 1 Administrative Staff
> 1 Apprentice
> 1 Advisor
About Comerge AG

Software you like

- **Business Software:** Eclipse Rich Client Platform
- **Mobile:** iOS & Android
- **Web:** Google Web Toolkit, J2SE, Drupal
Why? We believe in code.

We believe in code, not in doc
We want to see IT.

- Meaningful systems are too complex to specify
- We will always forget something, so we make sure we’re fit to add stuff and change stuff anytime
- Great challenges require great tools and great people
Our Customers

BEKB | BCBE

mobility

car sharing

ASA
Autism Society of America

CREDIT SUISSE

ifit

AXA INVESTMENT MANAGERS
What we do.

Mobility
Responsible for the complete rebuild of the car sharing platform.

going live: July 25 2011!
What we do.

Eclipse RCP
Use of proven robust technologies.
What we do.

Architecture
Scalability for Seamless Processes.
What we do.

Origo
Cloud-based Agile Software development platform (more than 17'000 users)
What we do.

Architecture
All the tools you need

Origo TeamWorkspace
GOALS: you will learn...

- what agile methods are

how they can be put to practice

that agile methods ≠ no process
Origins

> Erich Gamma
IBM Research /Ubilab
(former IT lab of UBS)

> Kent Beck
http://agilemanifesto.org
Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more.
The ingredients

- get customer involved
- build constantly
- deliver tested quality
- talk face-to-face
- everybody should be able to keep the pace
- embrace change
- automate technical excellence
- maximize the amount of work not done (simplicity!)
- self-organizing teams
- reflect on the way you work
The ingredients

▷ get customer involved
▷ build constantly
▷ deliver tested quality
▷ talk face-to-face
▷ everybody should be able to keep the pace
▷ embrace change

▷ automate technical excellence
▷ maximize the amount of work not done (simplicity!)
▷ self-organizing teams
▷ reflect on the way you work

→ Agile Software Development!
GOALS: you will learn...

what agile methods are

➔ how they can be put to practice

that agile methods ≠ no process
In practice

- timeboxed iterations
- user stories
- self organizing teams
- daily scrum
- test driven development
- continuous integration
- shipping process
- peer reviews
- retrospectives
- parties and lifestyle
In practice

- timeboxed iterations
- user stories
- self organizing teams
- daily scrum
- test driven development
- continuous integration
- shipping process
- peer reviews
- retrospectives
- parties and lifestyle

Tools, tools, tools!
Timeboxed Iterations

- Planning: 2-3 days
- Implementing: ~4 weeks
- Testing: 1 week

- Always the same length
- No asynchronous requests
- Ship happens at the end of the iteration
Planning Iterations

1. Define goals and principles
2. Identify stories
3. Define enhancements
4. Hand over to team leaders
5. Review plan

- Create a project site on the intranet
- Stories do not mention anything technical
- Assign them to teams and iterations
- No new stories or enhancements came up
- Team leaders and product manager
- Changed or new story
- Changed or new enhancement
Gathering User Stories

> As a [Describe Role of Person], I want to [describe activity], so that [add a context and a reason why this is of value]

> Example:

As a Mobility customer, I want to be able to use my smartphone to extend my current reservation, while being in a meeting.
Gathering User Stories

**Extreme Programming**

I can't give you all of these features in the first version.

And each feature needs to have what we call a “user story.”

Okay, here’s a story: You give me all of my features or I’ll ruin your life.
Self organizing Teams

Product manager

Customer

Team leader

Team leader

Team member

Team member

Team member

Team member

Team member
Working with Issues

> Team Leader triages issues and assigns to team members

> Issues are owned by teams together

> For everything we do, we create an Issue
Triaging Issues

Take issue from inbox

Reproduce issue

Plan issue
Fix issue
Close as fixed
Verify fix

Can reproduce
Regression test if feasible

Can't reproduce
Verifying

Don't know how to reproduce
Request details
Wait for feedback

Can't reproduce
No feedback

Close as fixed
Close as duplicate or wontfix
Close as works for me

Team leader assigns inbox triager

Close as works for me

Fix issue

Close as xed

Can't reproduce

Close as works for me

Not fixed
Everybody keeps pace
keep the meeting short!

9.45 am
# The Scrum Board

The Scrum board is a tool used in Agile software development to visualize the progress of tasks. The board typically consists of three columns: To Do, In Progress, and Done. Each column represents the status of tasks:

- **To Do**: Tasks that need to be started.
- **In Progress**: Tasks currently being worked on.
- **Done**: Tasks that have been completed.

## Columns
- **RCP**
- **ACP**
- **WEB**
- **MNG**

## Tasks

<table>
<thead>
<tr>
<th>Name</th>
<th>To Do</th>
<th>In Progress</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td>Berno</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Etienne</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bernd</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Danni</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Till</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bernd</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The board also includes sticky notes and cards to track specific tasks and their statuses. The colors of the cards and notes are used to indicate additional information or priorities.
The
TDD
Test Driven Development

- Take bug
- Take enhancement

- Write unit test
- Implement issue

- Commit
- Wait for build to succeed

- Resolve as fixed

Build fails

Example Stats:

- Web: 243
- RCP: 423
- DB: 278
- Test Total: 944
Continuous Integration

Project MonoRCPtycho

Building the MonoRCP with Tycho

Workspace

Last Successful Artifacts
- ch.mobility.mono_releng-0.14.0-SNAPSHOT-win32-win32-x86.zip
- mobsys2.0-Coop-setup.exe
- mobsys2.0-Coop-update.exe
- mobsys2.0-Coop.zip
- mobsys2.0-PSProd-setup.exe
- mobsys2.0-PSProd-update.exe
- mobsys2.0-PSProd.zip
- mobsys2.0-PSTest-setup.exe
- mobsys2.0-PSTest-update.exe
- mobsys2.0-PSTest.zip

Test Result Trend

Chuck Norris doesn't need an OS.

Recent Changes

Latest Test Result (no failures)

Upstream Projects

MonoACNtycho
Shipping Process

- Deploy release candidate
- Write test plan
- Test release
- Verify fixes
- Fix day(s)
- Deploy release candidate
- Build issues meeting
- Team discussion, teamlead(s) decide
- Build is not good
- Review for each fix required

- Build is good
- Release
- Request input from stakeholders
- Team discussion, teamlead(s) decide
- Review for each fix required
Test plan
Shipping Process

M9: Incidents & Activities

- Week 1
- Week 2
- Week 3
- Week 4
- Week 5

Number of changes to source code

Release M9

39
let’s make ship happen

M9: Incidents & Activities

Number of changes to source code

Number of reported issues

Release M9

Week 1  Week 2  Week 3  Week 4  Week 5
let’s make ship happen

M9: Incidents & Activities

- Number of changes to source code
- Number of reported issues

- Freeze M9
- Testingday M9
- Release M9

Week 1  Week 2  Week 3  Week 4  Week 5

41
Peer Reviews

1. **Build is not good**
   - Fix day(s)
   - Deploy release candidate
   - Verify fixes

   **Review for each fix required**

2. **Request input from stakeholders**

3. **Team discussion, teamlead(s) decide**
   - Review for each fix
   - Required

4. **Build is good**
   - Open bottle of champagne
   - Retrospective meeting
   - Triage issues
   - Meeting
Retrospectives

Before

Product manager invites to the meeting

Each participant collects 3 good + 3 bad items on the wiki page

Each participant reports

Discussion

Take notes on the wiki

Product manager invites to the meeting

Each team leader reports if the milestone plan is fulfilled

Each participant reports 3 good + 3 bad things

What has improved since last time?
Each participant collects 3 good + 3 bad items on the wiki page.

Each team leader reports if the milestone plan is fulfilled.

Discussion

Process improvements

What has improved since last time?
Each participant collects 3 good + 3 bad items on the wiki page.

Before

During

After

Product manager assigns tasks to update the content of the process wiki.

What has improved since last time?

Each team leader reports if the milestone plan is fulfilled.

Each participant reports 3 good + 3 bad things.

Process improvements

Discussion

Take notes on the wiki
Parties and lifestyle

Choose an idea for an event

Budget the event

Where, when, who?

Send invitations
Create flyer
Book at the chosen location
Give task to everyone at Comerge if needed

Additional info

Any potential new employees?

Get the party started!

Take pictures
Upload to Comerge Flickr / Facebook

Gather feedback

Brainstorming
Ask Comerge
Spy on employees
Ask people from other companies

Perhaps doodle
Our Process

> agile process
   (embrace change)

> always a concrete product
   (a release is a feature your software must have)

> iterations bring transparency
   (get used to working)

> automated testing
   (build on your mistakes, lay the ground for change)

> get things done
   (focus on the interesting things)
Troublemakers?

The Problems

Shared resources
Last minute feature-requests
Bad testers

What are the solutions?
Troublemakers?

THE SOLUTIONS

Shared resources

> Participation in Scrum-Meetings
> Write down all management tasks as issues
> Bring them to meeting
> Earn respect
Troublemakers?

THE SOLUTIONS

Last minute feature requests

> Trade for something else

> Implement ways to avoid them, e.g. by having a constant flow of releases coming
Troublemakers?

Bad Testers

> Improve process
> Automate more of the testing process
Our Process

THE FOUNDATION

> feature-richness
> delivery on time
> quality software
Our Process

THE FOUNDATION

> feature-richness
> delivery on time
> quality software

Pick two, manage with the third!
GOALS: you will learn...

what agile methods are

how they can be put to practice

→ that agile methods ≠ no process
What you get of the book:

> Project management patterns
> How to use tools to enable people
> What to do when things go wrong
CONCLUSION

We showed how to use tools and process to deliver software on time with quality.

Without getting killed by the tools or the doc.

And without killing the people!

Please ask us about the details.
Questions?

Need a job? Look for an internship?

> Contact us!
  michela.pedroni@comerge.net  till.bay@comerge.net