Software Reuse in Agile Projects

Net.ObjectDays 2003

Hasko Heinecke, Christian Noack,
Daniel Schweizer

Daedalos International AG, Copyright © 2003



Motivation

1. A software engineering promise: Reuse helps to develop software
   - in shorter time
   - at lower costs
   - of a higher quality
2. An agile processes promise: Software products that are
   1. simpler
   2. cheaper
   3. better suiting requirements, especially when they change

➔ Agile Reuse?

Agile Projects

"The cost of change may not rise dramatically over time."

- Iterative Process
- 4 Values: Communication, Feedback, Simplicity, Courage
- *„Do the simplest thing that could possibly work"*
- *„Embrace change"*
- *„You ain't gonna need it"*

➔ And Reuse?

Reuse I

"Software reuse is the systematic practice of developing software from a stock of building blocks…to achieve substantial benefits in productivity, quality and business performance.

- Systematic
- Building blocks (assets)
- *Time-to-market*
- *Costs*
- *Quality*
- *Operations and evolution*

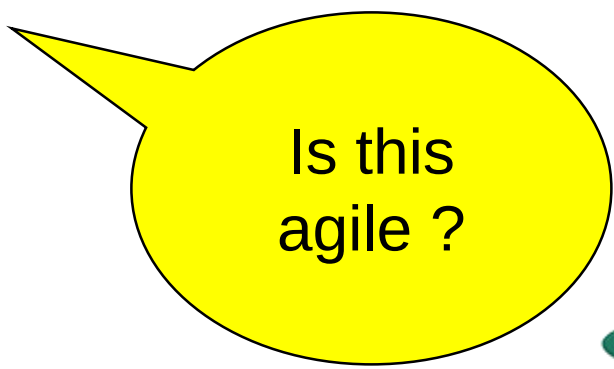
➔ And where is the problem?

Reuse II

Some Problems:

- How do I know about reusable assets?
- How do I provide my own assets to others?
- How do I produce assets that can be reused?

➔ Standards
➔ Project reviews and documentation
➔ Cataloguing and archiving
➔ Architecture-police



Agile Reuse?

Check advantages of reuse against :

- Values in XP
- Practices in XP
- Principles of Agile Manifesto

➔ We discuss some interesting aspects

Discussion

Full discussion in paper. Here are some examples of aspects.

- ➔ An obvious one (Simplicity)
- ➔ A pragmatic one (Measure of Project Progress)
- ➔ A not very well understood one (Courage)

Simplicity (XP)

"Simplicity--the art of maximizing the amount of work not done--is essential."
(Principle in the Agile Manifesto)

- ! Do as much as you have to do
- - Don't do anything unnecessary
- - Don't do anything from which you think you will need it later
- - You will never guess what is tomarrow

➔ Business Value First (XP hidden value)
➔ Don't plan for reuse - no functionality in advance

Progress ≅ Working Software

"Working software is the primary measure of progress."
(Principle in the Agile Manifesto)

- + Working software over comprehensive documentation – it is easier to reuse software than documentation
- ! Not the amount of software but amount of implemented stories counts (3rd party and reused components can help)
- - Frameworks usually do not work by their own

➔ Don't plan for reuse – implement working solutions to requirements of current iteration

Courage (XP)

Change requires courage

- + Broad test coverage motivates refactorings on reused components - *Feedback*
- + I do not reuse what I do not understand - *Simplicity*
- + In accordance with the original developers or the onsite customer I have more confidence to change things - *Communication*

➔ Courage is the resulting force of the other values
- without them courage is foolhardiness (Ger: Tollkühnheit)

➔ Courage can boost reuse

Conclusion

Agile Reuse is possible and makes sense

- It’s a question of implementation and organization
- Stick to agile Values and Principles

➔ Simplicity, Refactoring
➔ Business Value First
➔ Testing, Courage
➔ Individuals and Interactions over Processes and Tools

The Paper

Contains also:

- Beginning of a classification of Software Reuse
- Full discussion of Values, Practices, and Principles
- References

Outlook

Currently in work:

- How do I apply Agile Reuse in practice?
- Success stories
- Pitfalls

➔ Contribution to a future conference

Merci und uf Wiederluege

Daniel Schweizer
Daedalos Consulting AG
Zürichbergstrasse 7
8032 Zürich
Switzerland

Tel. +41 1 4810720
Fax +41 1 4810724
Daniel.Schweizer@daedalos.com