

Software Engineering 2008

Peter Dolog
 dolog [at] cs [dot] aau [dot] dk
 5.2.47
 Information Systems
 February 7, 2008

Goal of the Lecture

To give you an introduction to a course organization
 To give you an introduction to the main concepts

- Iterative Software Development
- Agile Software Development

Objectives

The main goal of this course is to facilitate basic understandings of:

- The main phases in software development
- Significant approaches to organize the phases
- Planning and management of the phases
- Influence of the software process organization on the company processes
- Reflecting on own practice in software projects

Service to You

Lectures 30 + 30 min

- I present, you question, comment, ...

Tutorials 30 min

- You present, your peer students comment, question,...; I guide and facilitate

Exercises 2x45 min

- You analyze, think, reflect, write, I and teaching assistant observe

Why so?

- By practicing and analysing the concepts with regard to what you have done you learn more and deeper
- You will be directly prepared for the exam (at least those who present 😊)

Waterfall Model

```

graph TD
    R[Requirements] --> D[Design]
    D --> I[Implementation]
    I --> T[Test]
    D --> R
    I --> D
    T --> I
  
```

Complete Requirements Not Known From Start

Fig. 5.1 rates of change on software projects

Source: *Applied Software Measurement*, Capers Jones, 1997. Based on 6700 systems

The Requirements Challenge

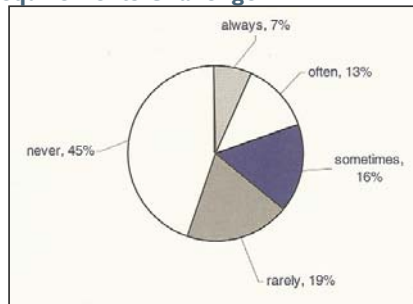


Fig. 5.3 actual use of requested features

Iterative Development

Background & Problems

waterfall: response to ad-hoc code-and-fix 1960's, Winston Royce (proponent of iterative dev.) - e.g. DoD standard 2167

system should be clearly specified before its design and implementation

clients are not sure what they want

details will only be revealed during development

as the product develop, clients change their minds

external forces (competitor's product)

-> high change rate, not predictable manufacturing

Manufacturing vs. Development

Predictable Manufacturing	New Product Development
It is possible to first complete specifications, and then build.	Rarely possible to create upfront unchanging and detailed specs. and then build.
Near the start, one can reliably estimate effort and cost.	Near the beginning, it is not possible. As empirical data emerge, it becomes increasingly possible to plan and estimate.
It is possible to identify, define, schedule, and order all the detailed activities.	Near the beginning, it is not possible. Adaptive steps driven by build-feedback cycles are required.
Adaptation to unpredictable change is not the norm, and change-rates are relatively low.	Creative adaptation to unpredictable change is the norm. Change rates are high.

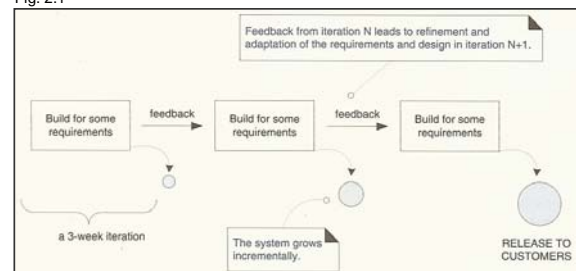
ID - Iterative Development

ID is an approach to build software of several iterations each iteration a mini-project

- requirements, design, implementation, test
- iteration release: a stable, integrated and tested partially complete system

IID - Iterative and Incremental Development

Fig. 2.1



Iterative and incremental development

Agile methods are a subset of iterative and evolutionary methods

The key practices:

- risk-driven and client-driven
- timeboxing
- evolutionary and adaptive development
- evolutionary requirements analysis
- evolutionary and adaptive planning rather than predictive planning

Peter Delsing - SCIE - Introduction 11

Risk-and Client-Driven Iterative Planning

What to do in the first/next iteration?

Risk-driven: the riskiest, most difficult elements for the early iterations

Client-driven: the choice of features comes from client, the currently highest business value

Peter Delsing - SCIE - Introduction 12

Timeboxing

the practice of fixing the iteration end-date and not allow it to change

if short of time, reduce the scope and leave features to the next iteration

no adding of new tasks to iteration

Project variables: scope (tasks), quality, resources (people), time

Peter Delsing - SCIE - Introduction 13

Evolutionary and Adaptive Development

requirements, plan, estimates, solution evolve and are refined

requires feedback from users, tests, developers

Peter Delsing - SCIE - Introduction 14

Two contract phases

Fig. 2.6

EFFORT, COST, or SCHEDULE ESTIMATES

4.0x

2.0x

Final Actuals x

0.5x

0.25x

iter 1 iter 2 iter 3

Phase 1. Fixed Time, Fixed Price. e.g., 8 weeks, \$200,000

Phase 2. Fixed Price

iter 1 iter 2 ...

fixed price bidding for phase 2

10% of the final software

- 3 requirements workshops

- software development to build a core architecture and obtain information for future estimation.

- Most high-level requirements

- Realistically, only 75% of the detailed requirements, or less

Peter Delsing - SCIE - Introduction 15

Incremental delivery with iterations

Fig. 2.7

4 weeks 2 weeks

iteration 1 iteration 2 ... iteration 10

Jan - June

Incremental Delivery 1

3 weeks 3 weeks

iteration 1 iteration 2 ... iteration 7

July - Dec

Incremental Delivery 2

deliver

deliver

Peter Delsing - SCIE - Introduction 16

Let's look at the example

I need a volunteer with a project to look at
He should present the case for 10 minutes that the others
understand what is going on
We will try to reflect on what was happening

Example from a Student Data Integration and Software Arch.

Digital Library
Multiple Data Sources
Presenting to different People
What is the library domain – to understand
Data types and how to present (video, audio, but also
activities...)
Huge discussions with librarians
Understanding the real world of library was the challenge (2
month)
They did not tell anything of features
After 2 month – design
Design on was interactive
Transforming features from domain understanding
Iterations not fixed
Risky features first

Agile Development

The Agile Manifesto

<i>Individuals and interactions</i>	<i>over process and tools</i>
<i>Working software</i>	<i>over comprehensive documentation</i>
<i>Customer collaboration</i>	<i>over contract negotiation</i>
<i>Responding to change</i>	<i>over following the plan</i>

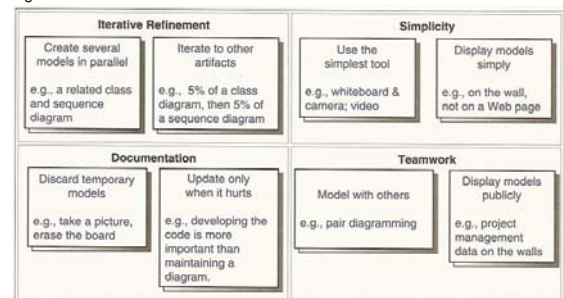
*That is, while there is value in the items on the right,
we value the items on the left more*

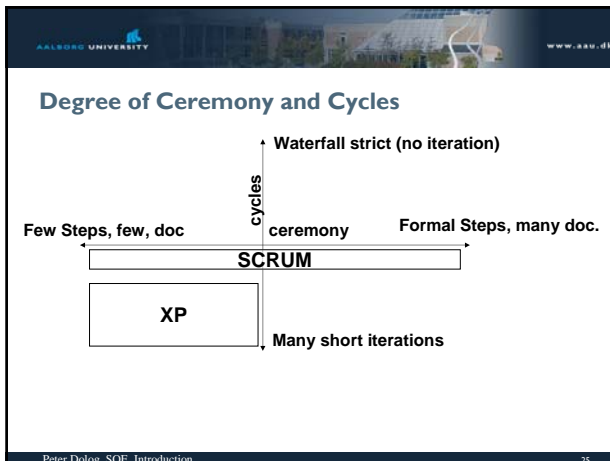
The Agile principles and agile management

1 early and continuous delivery of valuable software	8 agile processes promote sustainable development
2 welcome changing requirements, even late	9 sponsors, developers, and users maintain constant pace
3 deliver working software frequently	10 attention to technical excellence and good design enhances agility
4 business people and developers together daily	11 simplicity – the art of maximizing the amount of work not done is essential
5 motivated individuals	12 the best architectures, requirements and designs emerge from self-organizing teams
6 face-to-face conversation	13 regular reflections in the team on how to become more effective
7 working software is the primary measure of progress	

Agile Modeling

Fig. 3.3





Cockburn scale

↓ Criticality

Life (L)	L6	L20	L40	L100
Essential Money (E)	E6	E20	E40	E100
Discretionary Money (D)	D6	D20	D40	D100
Comfort (C)	C6	C20	C40	C100

People

Peter Dalsgaard, SCIE, Introduction 37

Summary

Course Organizations
Iterative Development
Agile Development

Questions and answers

Peter Dalsgaard, SCIE, Introduction 37

Exercise

Reflect on basic ideas for iterative and agile development in your selected project

I would like to receive by Tuesday evening: team number, team room, team members, project name, project abstract, team contact (for those not grouped yet, please send immediately when you are grouped)

I would like to receive 2 volunteers to present at the tutorial

To my E-mail address: @ first slide

Peter Dalsgaard, SCIE, Introduction 38

Assignment

Choose a project of your own from your past experience where you participated

Look at the project from the iterative software development perspective based on readings from previous lecture

Reflect on the project, i.e. was your project already iterative, if yes how, what would you change in the project to make it more iterative, what would be a benefit for that? How big was your iteration? Was it fixed? What about communication? Did you do face to face meetings? How often and with whom? ...

Prepare 10 min presentation on your reflections

Document reflections in a report

Peter Dalsgaard, SCIE, Introduction 39