COM3005J: Agile Processes
Chapter 1: **Context**

Dr. Anca Jurcut
E-mail: anca.jurcut@ucd.ie

School of Computer Science and Informatics
University College Dublin

Beijing-Dublin International College

Context: Overview

- Introduction to Week 1
- The Agile Manifesto
- Agile Methods
- Official Agile Principles
- Agile Values
- Assignment 1: Context Quiz (to be uploaded on cs moodle soon)

Chapter 1: Context

- **Introduction to Week 1**
- The Agile Manifesto
- Agile Methods
- Official Agile Principles
- Agile Values

Introduction to Week 1(1)

- I hope you enjoy this course.
- Agile processes are one of the most important ideas on the software scene today.
- Everywhere in the industry, people are looking at agile methods and often practicing them or some form of agile development.
- Every software employer is going to give preference to candidates that master agile concepts.
- This course will enable you to master them and decide for yourself what they can do for you, for your projects, and for your career.

Introduction to Week 1 (2)

- Nowadays, many presentations of agile processes are partisan. They really want you to kneel down and to start praying and adopt this or that agile method.
- This course is not preaching one method or another
  - is preaching software quality and software productivity

!!! So we're going to see what agile processes are about. But we're not going to advertise for a particular method.

  - and in fact-- surprise, surprise-- in agile processes, there are some good aspects and some not so good aspects as well, as in every human endeavour.

- It is very important to keep one's cool and to decide what is best for you, for your project.
- So this is going to be an analysis, without any bias, of agile processes as they exist today.

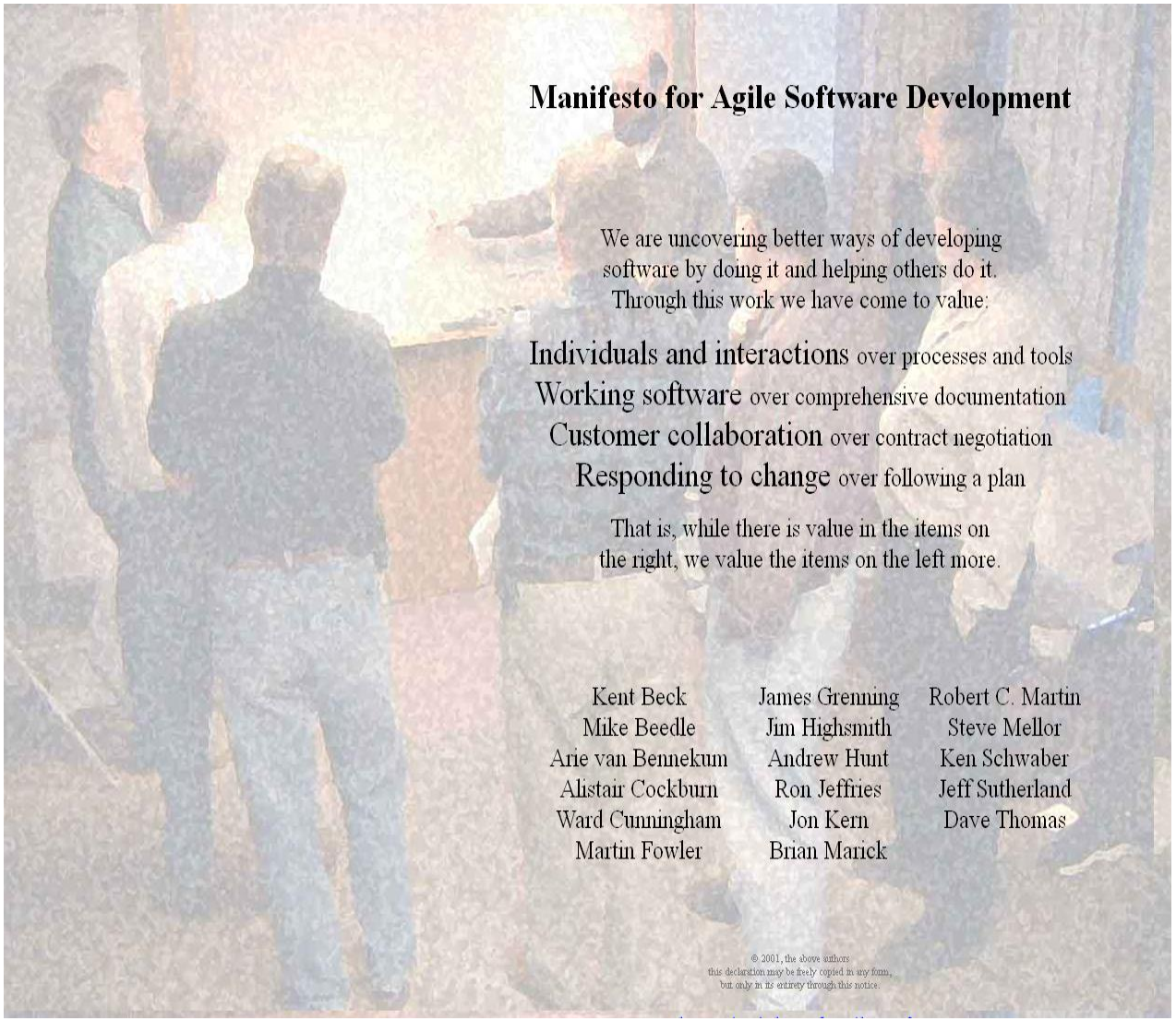
Chapter 1: Context

- **The Agile Manifesto**
- Agile Methods
- Official Agile Principles
- Agile Values

The Agile Manifesto

- The Agile Manifesto is the 17-year-old text which first presented Agile ideas. And it's remarkable to see how aptly it still characterizes the Agilist's view today!
- Agile processes came into prominence at the time of the publication of the so-called Agile Manifesto in 2001. This was, as the name suggests, a pamphlet, a manifesto, designed to attract the world's attention on the need to develop software differently.
- The manifesto was the work of a number of people that are in fact software consultants.
- The manifesto gained a lot of visibility, and has continued to be quoted widely since then.

Agile manifesto



agilemanifesto.org

Twelve Principles (from the Manifesto)

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity— the art of maximizing the amount of work not done —is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Chapter 1: Context

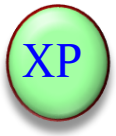
1. ….
2. **The Agile Manifesto**

What we have seen:
A set of principles,
defining Agile as seen by its creators

Chapter 1: Context

- The Agile Manifesto
- **Agile Methods**
- Official Agile Principles
- Agile Values

Agile Methods



(Extreme programming) Kent Beck

Mary Poppendieck

Alistair Cockburn

Schwaber & Sutherland

XP: Extreme Programming

- Came out in the 90s.
- XP - a reaction against the culture (process, plans, diagrams) that was predominant in software engineering circles at the time.
  - e.g. UML (Unified Modeling Language), or CMMI (Capability Maturity Model Integration - a standard for defining best practices in software development).
- Drawing attention to the fact that what really matters in the end is the programs, and of course, the programming and the programmers.
- XP made a major contribution by rehabilitating, the work of programmers, and putting programs and programming at the center of software development.

Lean Software

- An attempt to apply to software a number of ideas and principles which have proved their value in other engineering fields, in particular, in the car industry.
- There's the famous set of practices developed in particular, by Toyota in Japan, which have been very influential, not just in the automobile industry, but throughout industry, in particular industries, that make material things.
- Poppendiecks applied these ideas to software, emphasizing in particular the need to get rid of what they call waste.
- Poppendiecks claim that we should also be on the lookout for waste in software, and get rid of things like for example, useless documentation, which they view as waste, in order to concentrate on stuff that is going to be actually delivered to the customer.

Crystal

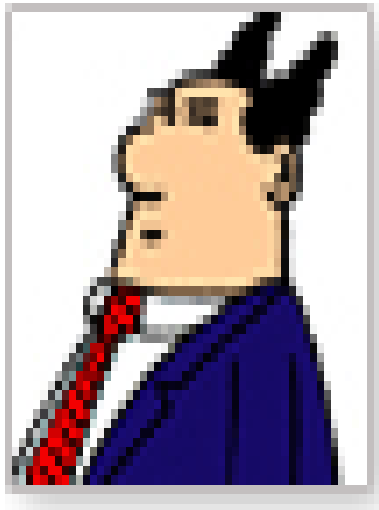
- A combination of agile ideas and more traditional ideas.
- Crystal is actually the name for a set of methods characterized by various degrees of importance of process in management.
- Alistair Coburn, who developed the Crystal set of methods is trying to combine the best of Agile with a best of more process oriented approaches.

.

Scrum

- Dominate the scene in recent years.
- Less technical, not so much software-focused than XP.
- A management method which emphasizes ideas such as the importance of self-organizing teams as opposed to teams that are closely managed by a manager, and the importance of a specific kind of short release iterations known as **Sprints.**
- It's important to note that in practice nowadays when we talk about Agile, we really talk about Scrum.
  - This view is not entirely accurate… a lot of ideas from the other methods, in particular, XP.
  - When we continue at a high level of abstraction, at a bird's eye view of what Agile methods are, it's important to know that initially with XP, one could view the use of Agile methods as a reaction against Dilbert's boss. So, so to speak, against the management aspect in favour of the view of the Dilberts of this world.

XP: The revolt of the cubicles



Negotiated scope contract

XP

Source: Beck 05

"Write contracts for software development that fix time, costs, and quality but call for an ongoing negotiation of the precise scope of the system. Reduce risk by signing a sequence of short contracts instead of one long one."

"You can move in the direction of negotiated scope. Big, long contracts can be split in half or thirds, with the optional part to be exercised only if both parties agree. Contracts with high costs for change requests can be written with less scope fixed up front and lower costs for changes"

Chapter 1: Context

1. ....
2. ....

3. The Agile Methods

What we have seen:

A number of agile methods have emerged
They share the basics
but differ in their goals and emphases

Chapter 1: Context

- The Agile Manifesto
- Agile Methods
- **Official Agile Principles**
- Agile Values

Twelve Principles

Source: Agile manifesto

What about testing?

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity — the art of maximizing the amount of work not done — is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Redundancy

Practice

Redundancy

Assertion

Assertion

Assertion

Wrong

Practice

Finishing a Design

“It seems that the sole purpose of the work of engineers, designers, and calculators is to polish and smooth out, lighten this seam, balance that wing until it is no longer noticed, until it is no longer a wing attached to a fuselage, but a form fully unfolded, finally freed from the ore, a sort of mysteriously joined whole, and of the same quality as that of a poem. It seems that perfection is reached, not when there is nothing more to add, but when there is no longer anything to remove.”

(Antoine de Saint-Exupéry, *Terre des Hommes*, 1937)

Steve Jobs, 1998

“That's been one of my mantras — focus and simplicity. Simple can be harder than complex: You have to work hard to get your thinking clean to make it simple. But it's worth it in the end because once you get there, you can move mountains.”

(Steve Jobs in an Interview to Business Week's in '98)

Chapter 1: Context

1. ....
2. ....
3. ....
4. **Official Agile Principles**

What we have seen:

The official principles give us a good starting point for understanding Agile,
but they fall short of explaining exactly what the method is

Chapter 1: Context

- The Agile Manifesto
- Agile Methods
- Official Agile Principles
- **Agile Values**

A More Precise Description

- Agile values
- Agile principles
  - Managerial
  - Technical
- Agile roles
- Agile practices
  - Managerial
  - Technical
- Agile artifacts

Agile values

A New, reduced role for manager

B No “Big Upfront” steps

C Iterative development

D Limited, negotiated scope

E Focus on quality, achieved through testing

Chapter 1: Context

1. ….
2. ….
3. ….
4. ….

5. **Agile Values**

What we have seen:

Beyond specific principles, practices and artifacts, Agile is defined by general values, representing a vision of how development should proceed