Agility in Software Development

Ashleigh Brilliant (https://www.ashleighbrilliant.com)

Dr J Paul Gibson, Dept. INF, TSP, Evry, France

http://blog.dilbert.com
Agile Software Engineering - survey for CSC4102 (Part I)
Structure of the Presentation

1. Software Process and the Software Life Cycle
2. History of Agile: More or less a process?
3. Agile Fundamentals
4. Comparing Agile Methods
5. Agile Resources

An extended version of this presentation can be found in: “Pour aller plus loin”
1. Software Process and the Software Life Cycle

How important is a software development process?

"... the quality of the people on a project, and their organization and management, are much more important factors in the success than are the tools they use or the technical approaches they take."


ISO 9000 - Quality management
http://www.iso.org/iso/iso_9000

ISO/IEC JTC 1/SC 7 - Software and systems engineering

ISO/IEC 12207 Systems and software engineering - Software life cycle processes

CMMI maturity model for Software Process Improvement (SPI)
http://cmmiinstitute.com

http://dilbert.com/search_results?terms=Iso+9000
History of Agile: Philosophy and People

Agile software development isn’t a set of tools or a single methodology, but a philosophy put to paper in 2001 with an initial 17 signatories.

Kent Beck  
Mike Beedle  
Arie van Bennekum  
Alistair Cockburn  
Ward Cunningham  
Martin Fowler

James Grenning  
Jim Highsmith  
Andrew Hunt  
Ron Jeffries  
Jon Kern  
Brian Marick

Robert C. Martin  
Steve Mellor  
Ken Schwaber  
Jeff Sutherland  
Dave Thomas

"Many people may think that agile is just another software development process. Although that is true to a degree, there is a lot more to agile than just a process or just a set of practices. Agile (or agility) is more of a mindset - a way of thinking about software development."

3. Agile Fundamentals

Agile: Values, Principles and Methods
3. Agile Fundamentals

**Manifesto for Agile Software Development**

http://agilemanifesto.org/

We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value:

- Favour Individuals and interactions over processes and tools
- Favour Working software over comprehensive documentation
- Favour Customer collaboration over contract negotiation
- Favour 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.
3. Agile Fundamentals

Agile: Principles highlighted in CSC4102

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

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.

9. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
Agile Software Engineering - survey for CSC4102 (Part II)
3. Agile Fundamentals

Agile: The Other Principles

2 Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

4 Business people and developers must work together daily throughout the project.

8 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 behaviour accordingly.
Differences between CSC4102 and Agile in Industry

CSC4102 focuses on technical aspects; Agile in Industry also considers non-technical issues such as management, costs, resources, business, marketing, etc... in an agile framework

CS4102 had to compromise a purely agile approach in order to meet its teaching objectives (because the software engineers participating in the development had to learn about things that are normally taken for granted in industrial application of agile methods)
Some of the most popular Agile methods:

Scrum
Lean Development
Extreme programming (XP)
Adaptive Software Development (ASD)
Agile Modeling
Crystal Methods
Dynamic System Development Methodology (DSDM)
Feature Driven Development

Most relevant to CSC4102
4. Comparing Agile Methods

**Agile techniques which motivated aspects of CSC4102**

<table>
<thead>
<tr>
<th>Short “Sprints”</th>
<th>Scrum</th>
<th>Lean Development</th>
<th>Extreme programming (XP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Short, high frequency meetings</th>
<th>X</th>
<th>X</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Working code ASAP</th>
<th>X</th>
<th>X</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Build integrity through tests</th>
<th>X</th>
<th>X</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Increments - as stories or use cases</th>
<th>X</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Continuous integration</th>
<th>X</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Pair programming</th>
<th>X</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>See the whole</th>
<th>X</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Refactoring design</th>
<th>X</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Prioritizing Increments</th>
<th>X</th>
<th>X</th>
</tr>
</thead>
</table>

cf. “Pour aller plus loin” for details
5. Agile Resources

http://agilemanifesto.org/

http://www.agilealliance.org/

http://en.wikipedia.org/wiki/Agile_software_development

http://scrummethodology.com/

http://www.extremeprogramming.org/

http://www.leansoftwareinstitute.com/

An Overview of Feature-Oriented Software Development

An extended version of this presentation - containing links to many additional resources - can be found in: “Pour aller plus loin”
QUESTIONS ....?