

Software Development Methodologies

Bahman Zamani

PhD Candidate
Dept. of Computer Science and Software Engineering
Concordia University

3/3/2006

Software Development Methodologies

1/16

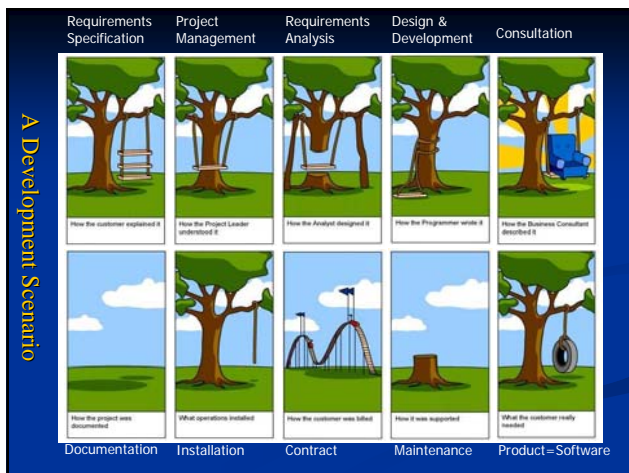
Overview

- A Development Scenario
- What is a Methodology?
- The Scope of a Methodology
- Methodology History
- Heavyweight Methodologies
- Agile Methodologies
- Agile vs. Heavyweight
- Conclusion

3/3/2006

Software Development Methodologies

2/16



What is a Methodology?

- Provides a means for
 - managing the development process
 - Problem understanding
 - Software Design
 - Software invention
 - Supporting that process by
 - Managing Risk
 - Measuring Quality

3/3/2006

Software Development Methodologies

4/16

What is a Methodology? (Cont'd)

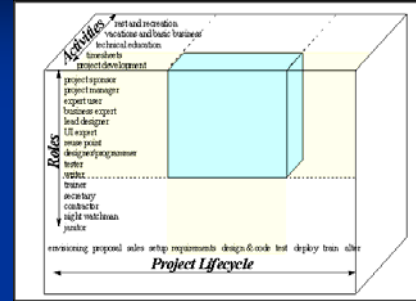
- Definition: A methodology is a recommended set of
 - Phases
 - Procedures
 - Rules
 - Techniques
 - Tools
 - Documentation
 - Management
 - training
 used to develop a system.
- Abstract: An organized, documented set of rules and practices

3/3/2006

Software Development Methodologies

5/16

The Scope of a Methodology



3/3/2006

Software Development Methodologies

6/16

Methodology History

- Pre-Methodology Era (1960s and 1970s)
 - Ad hoc software development (Cowboy coders)
 - **Problem:** The Software Crisis (1969)
 - **Solution:** Standardize software development process and rely on a methodological approach

3/3/2006

Software Development Methodologies

7/16

Methodology History (Cont'd)

- Early Methodology Era (late 1970s, early 1980s)
 - Focus on identification of phases and stages
 - Software Development Life Cycle (Waterfall Model)
 - **Problem:** inflexibility, resisting changes
 - **Solution:** More focus on user involvement

3/3/2006

Software Development Methodologies

8/16

Methodology History (Cont'd)

- Methodology Era (1980s and 1990s)
 - Definition of methodology: A methodology is a recommended set of phases, procedures, rules, techniques, tools, documentation, management, and training used to develop a system.
 - Different Methodologies emerged:
 - Structured, Prototyping, OO, etc.
 - **Problem:** Too much discipline, many failed projects
 - **Solution:** New methodologies, or no methodology at all!

3/3/2006

Software Development Methodologies

9/16

Methodology History (Cont'd)

- Post-Methodology Era (late 1990s - ...)
 - **View1:** Go back to ad hoc , trail-and-error manner
 - **View2:** Looking for better approaches
 - Development using tools
 - Improved OO approaches
 - Evolutionary development (**Agile**)
 - External development (ERP, CRM)
 - Outsourcing
 - One methodology for every project

3/3/2006

Software Development Methodologies

10/16

Heavyweight Methodologies

- Many rules, practices, and documents
- Require discipline and time to follow correctly
- Bureaucratic and laborious
- Software crisis problems still remain
 - 80% of systems are late and over-budget
 - 40% of developments fail
 - 10% - 20% of systems meet their success criteria

3/3/2006

Software Development Methodologies

11/16

Agile Methodologies

- Agile = Lightweight
- A reaction to delayed, over-budget, collapsing projects
- Only a few rules and practices
- Between no process and too much process
- Less document-oriented, but rather code-oriented

3/3/2006

Software Development Methodologies

12/16

Agile Methodologies (Cont'd)

- What makes a method Agile?
 - Incremental
 - Cooperative
 - Straightforward
 - Adaptive
- When we use Agile?
 - Requirements are uncertain or volatile
 - Developers are responsible and motivated
 - Customer is involved and understands

3/3/2006

Software Development Methodologies

13/16

Agile Methodologies (Cont'd)

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.

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

3/3/2006

Software Development Methodologies

14/16

Conclusion

- For a project, methodology is mandatory.
- There is no silver bullet!
- "ONE SIZE FITS ALL" is wrong!
- The methodology weight must match with the product characteristics.

3/3/2006

Software Development Methodologies

15/16

Thank You!

3/3/2006

Software Development Methodologies

16/16