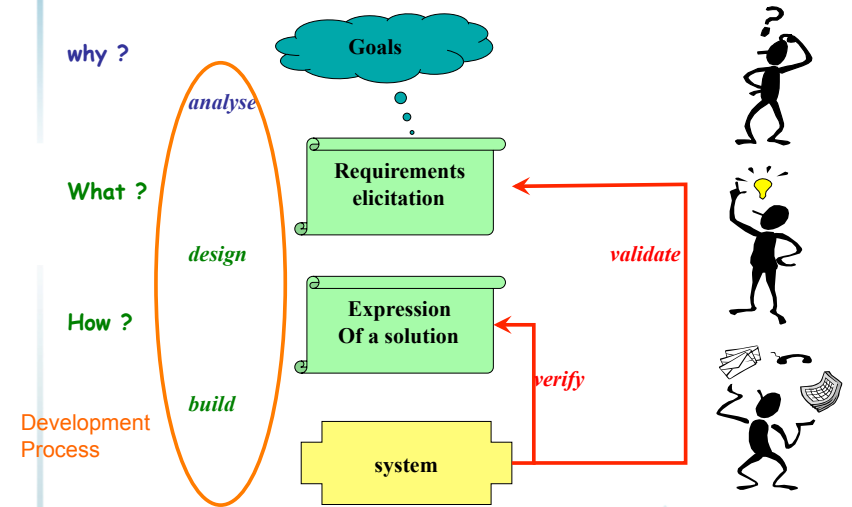




Design Process engineering

Sophie Dupuy-Chessa

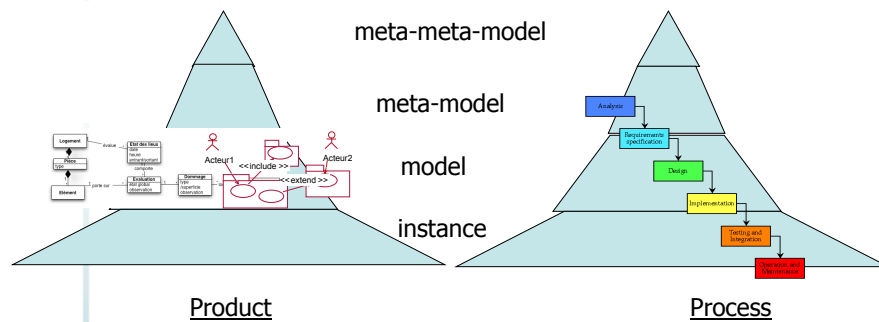
A classical engineering approach



OCTOBER 2014
MOSIG /SIGAL

2

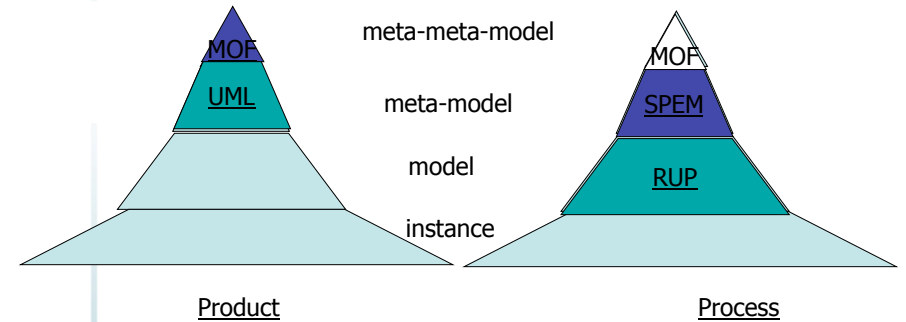
Engineering methods



OCTOBER 2014
MOSIG /SIGAL

3

Engineering methods



OCTOBER 2014
MOSIG /SIGAL

4

Process engineering lectures

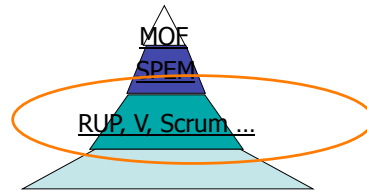
Goals:

- Reuse and adaptation of development process
=> Users in the processes

Approach:

– Process models

- How to compare process models?
 - Practice with a tool



What is a process?

« A process is a *set of activities that are interrelated* or that *interact with one another.* » (ISO 9001:2000)

=> *it is not a procedure.*

- A process answers to the questions: What to do? For which added value?
- A procedure answer to the questions: How to do? When? Who?

=> *it is not a life cycle model*

- a life cycle model is a framework of processes and activities concerned with the life cycle that may be organized into stages, which also acts as a common reference for communication and understanding

Activity : set of cohesive tasks of a process

Software Life Cycles

• 3 normalized Life cycle models

- Waterfall
- Spiral
- Iterative or incremental

• Other well-known life cycle models

- V-model
- semi-iterative models
- ...

Software Life Cycles

- Waterfall
- Spiral
- Iterative or incremental
- V-model
- semi-iterative model
- ...

↳ 2 families

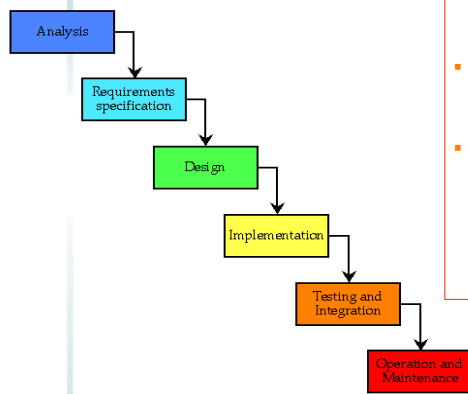
• Linear cycles

- Based on the Waterfall model
- **Temporal** breaking down of the project
- **TOP-DOWN**

• Iterative cycles

- generally based on prototyping
- Structural breaking down of the project
- **BOTTOM-UP**

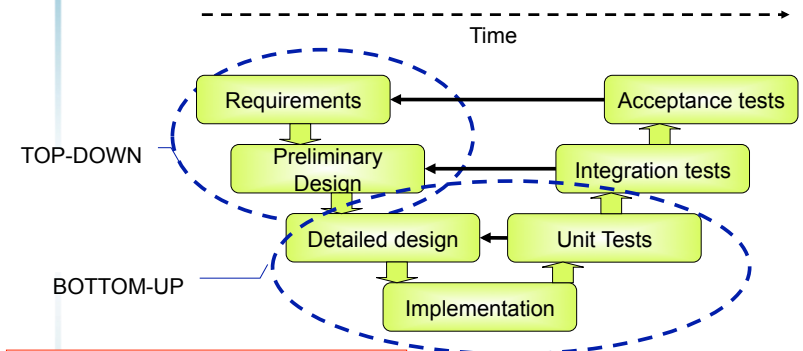
Linear cycles Waterfall Model



- Principles:
 - Each activity is realized once.
 - No feedback
- Advantage:
 - The whole system delivered at one time.
- Drawbacks:
 - Requirements badly understood
 - Changing requirements
 - System too big to be developed in one shot

► Adapted for small projects with a well defined domain

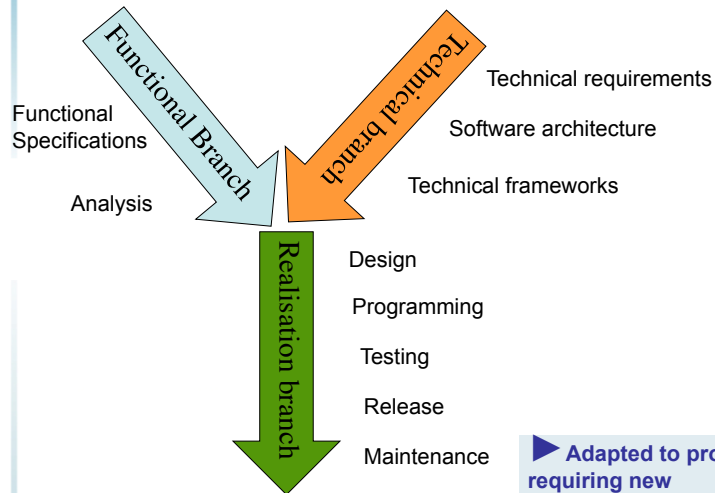
Linear cycles V-model



- Amelioration of the Waterfall model
- Limit feedback risks
- Think about testing for each system decomposition

► Adapted to projects with a well defined domain

Linear Cycles Y cycle



► Adapted to projects requiring new technologies

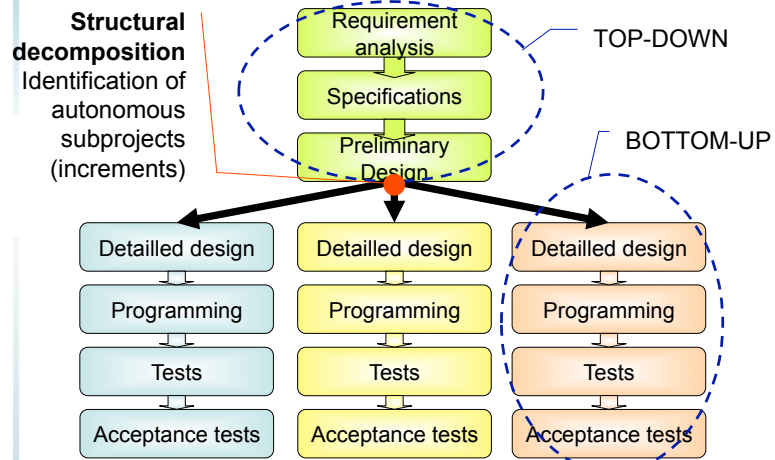
Exercise

Open the <http://design-methods.net> website

Read the description of the waterfall model in the « Methods » tab

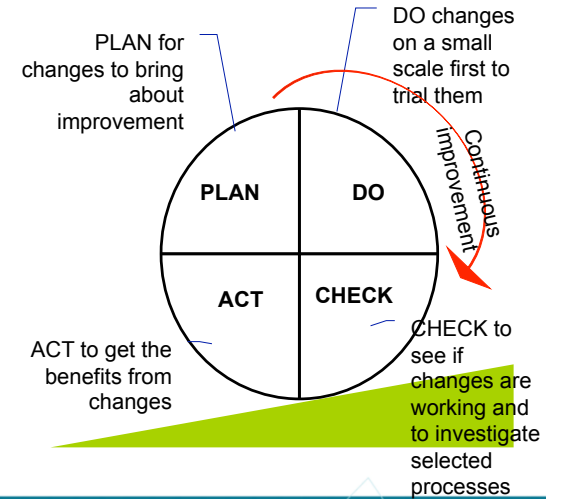
What do you learn about the waterfall model?

Semi-Iterative model (RAD, FDD, ...)

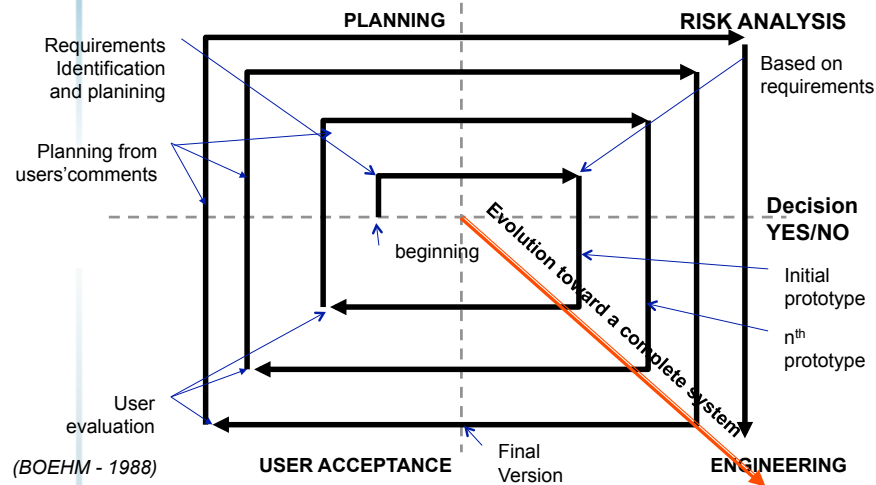


Foundations

- Based on the **DEMING wheel**
 - PDCA method (Plan Do Check Act)
 - Continuous improvement in quality



Spiral Model



Exercise

Open the <http://design-methods.net> website

Read the description of the spiral model in the « Methods » tab

What do you learn about the spiral model?

Spiral model

- Principle: realize successive version of the system
 - While refining requirements
 - All requirements are identified, but they are not stable.
 - By producing a robust version for a set of requirements
 - By working in parallel on different versions (several teams)
- Advantages
 - A first operational version is available.
 - The system is built by increments.
 - Use feedbacks to refine requirements
- Drawback
 - Risks from Requirements not stable

Adapted to complex and uncertain projects

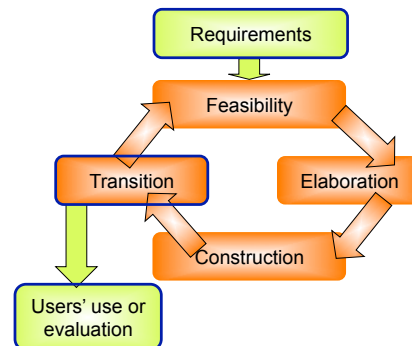
Iterative and incremental model

- Principle: realize successive version of the system
 - While refining requirements
 - Not All requirements are identified, but they are stable.
 - By producing a viable version for a set of data
 - By working in parallel on different versions (several teams)
- Advantages
 - A first operational version is available.
 - The system is built by increments => flexibility
- Drawbacks
 - Risks from Misunderstood Requirements
 - Evolution of requirements

Adapted to complex projects

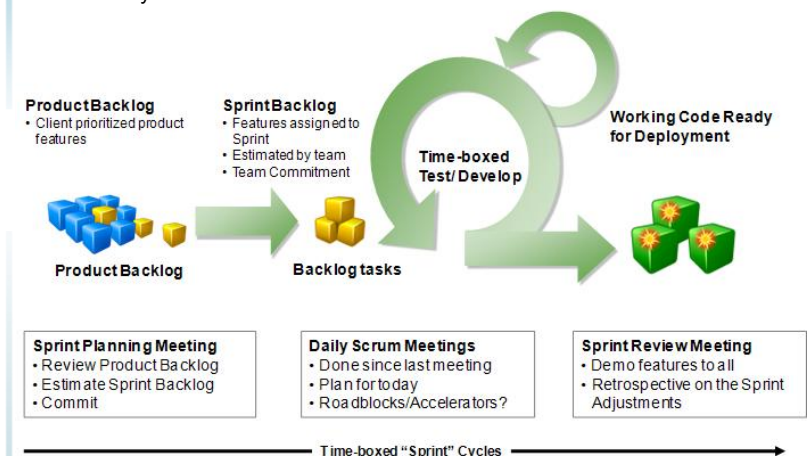
Iterative and incremental model

- It can be NOT an iterated V-cycle
 - Short iteration
 - => iteration on a product (doc, test, code ...)
 - Feasibility <-> Specifications
 - Acceptance of a new requirement
 - Elaboration <-> Design
 - Imagine the solution
 - Construction <-> Prototype Development
 - Transition <-> test
 - Deliver to the customer



Classical agile cycle

SCRUM cycle



Exercise

Open the <http://design-methods.net> website

Read the description of SCRUM in the « Methods » tab

What do you learn about SCRUM?

Iterative cycle Agile Manifesto

Individuals and interactions **over** processes and tools

Working software **over** comprehensive documentation

Customer collaboration **over** contract negotiation

Responding to change **over** following a plan

► Iterative incremental
and collaborative
approach

<http://agilemanifesto.org/>

Iterative cycles

Agile Manifesto - Principles

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.

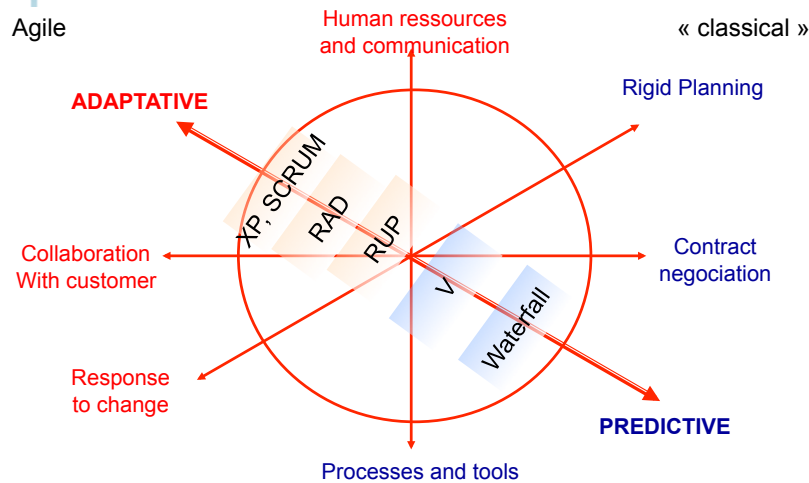
<http://agilemanifesto.org/>

Iterative cycles Agile cycle

- Principle: realize successive version of the system
 - While refining requirements
 - All requirements are identified, but they are not stable.
 - Short cycle (21 days)
- Advantages
 - Customers' satisfaction
 - Better visibility of project progress
- Drawbacks
 - No documentation (maintenance, change in a team ...)
 - Difficulty for standardisation

► Changing
requirements for
projects with user
facing

Overview

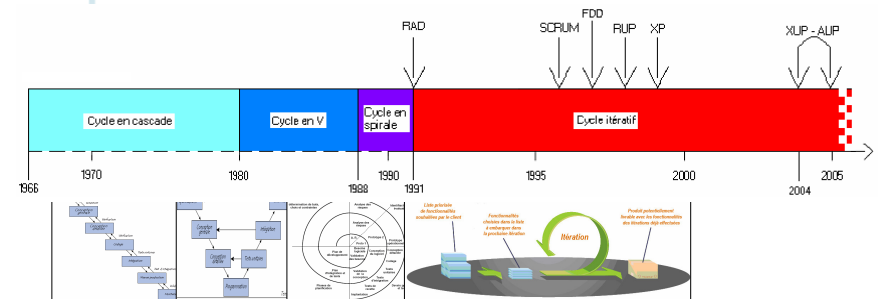


OCTOBER 2014
MOSIG /SIGAL

S. Dupuy-Chessa

25

Chronology

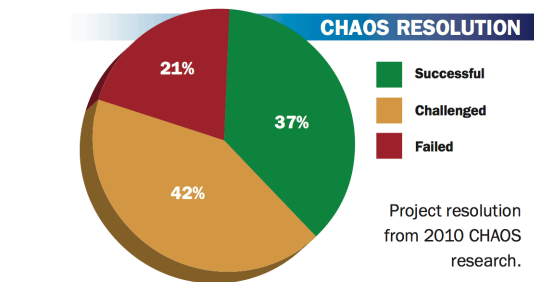


OCTOBER 2014
MOSIG /SIGAL

S. Dupuy-Chessa

26

Projects resolution



	2002	2004	2006	2008	2010
Successful	34%	29%	35%	32%	37%
Challenged	51%	53%	46%	44%	42%
Failed	15%	18%	19%	24%	21%

<http://versionone.com/assets/img/files/CHAOSManifesto2012.pdf>

OCTOBER 2014
MOSIG /SIGAL

S. Dupuy-Chessa

27

Project factors of success

FACTORS OF SUCCESS	POINTS
Executive Management Support	19
User Involvement	18
Clear Business Objectives	15
Emotional Maturity	12
Optimization	11
Agile Process	9
Project Management Expertise	6
Skilled Resources	5
Execution	4
Tools and Infrastructure	1

<http://versionone.com/assets/img/files/CHAOSManifesto2012.pdf>

OCTOBER 2014
MOSIG /SIGAL

S. Dupuy-Chessa

28

Toward method engineering



- Cycles are just one aspect of a method.
- Development Methods are numerous (1000 methods identified in 2001).
 - How to choose one?
- They never used as they are.
 - Need for flexibility (Method spectrum from Harmsen, Brinkkemper and Oei)

