Software Processes

Session 2

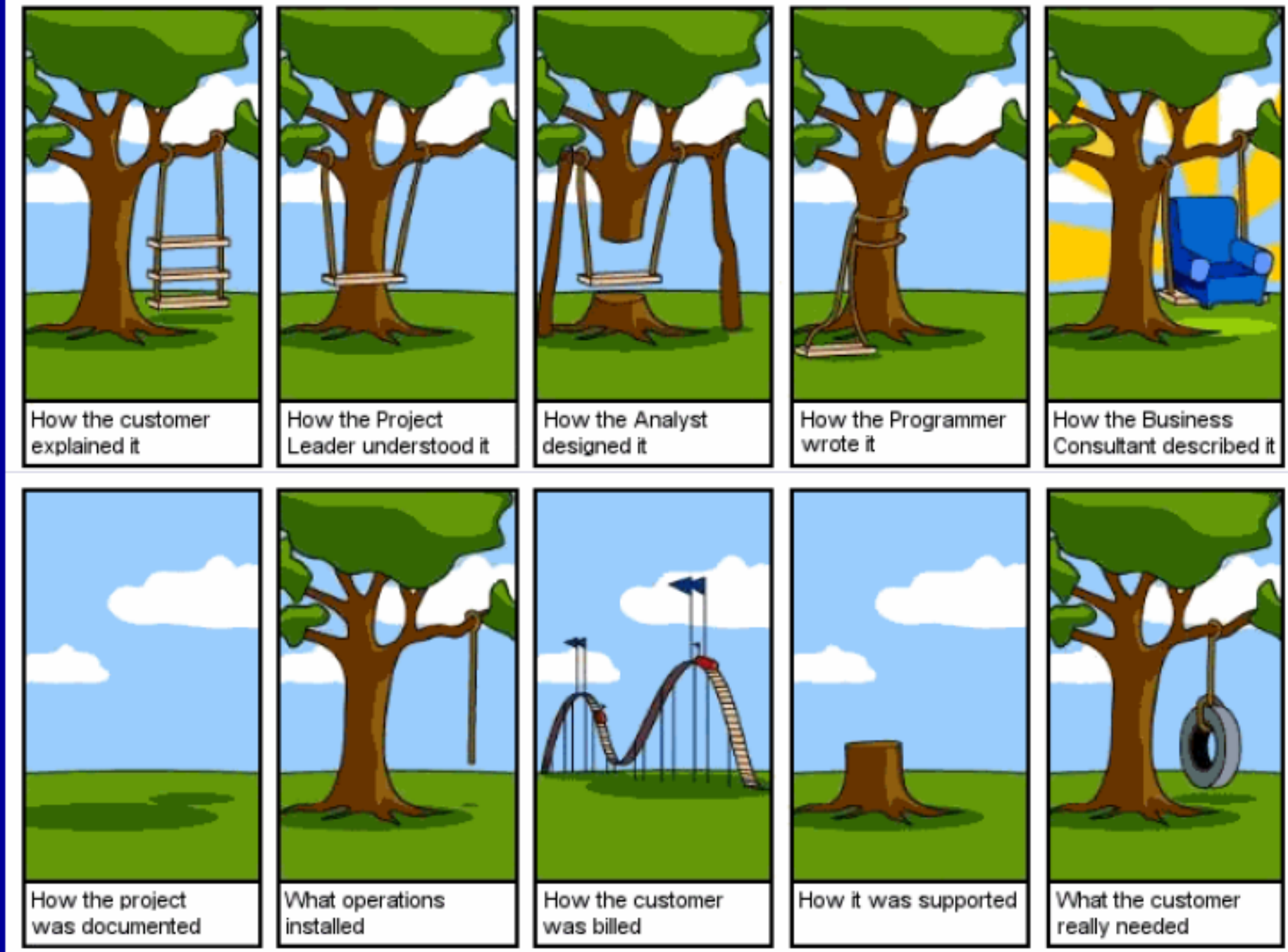
MobileSenegal et Mobile4Senegal

http://mobilesenegal.org

http://facebook.com/mobilesenegal

http://twitter.com/mobilesenegal

DEM


How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed

How to choose a process?

How to define a process?

How to audit a process?

DEM

The software development process

- A structured set of activities required to develop a software
  - Requirements
  - Design
  - Implementation
  - Testing
  - Evolution and maintenance

Software = product

Process = HOW to develop the software

- A software development process model is an abstract representation of a process, a description from some particular perspective

DEM

Generic software development process models

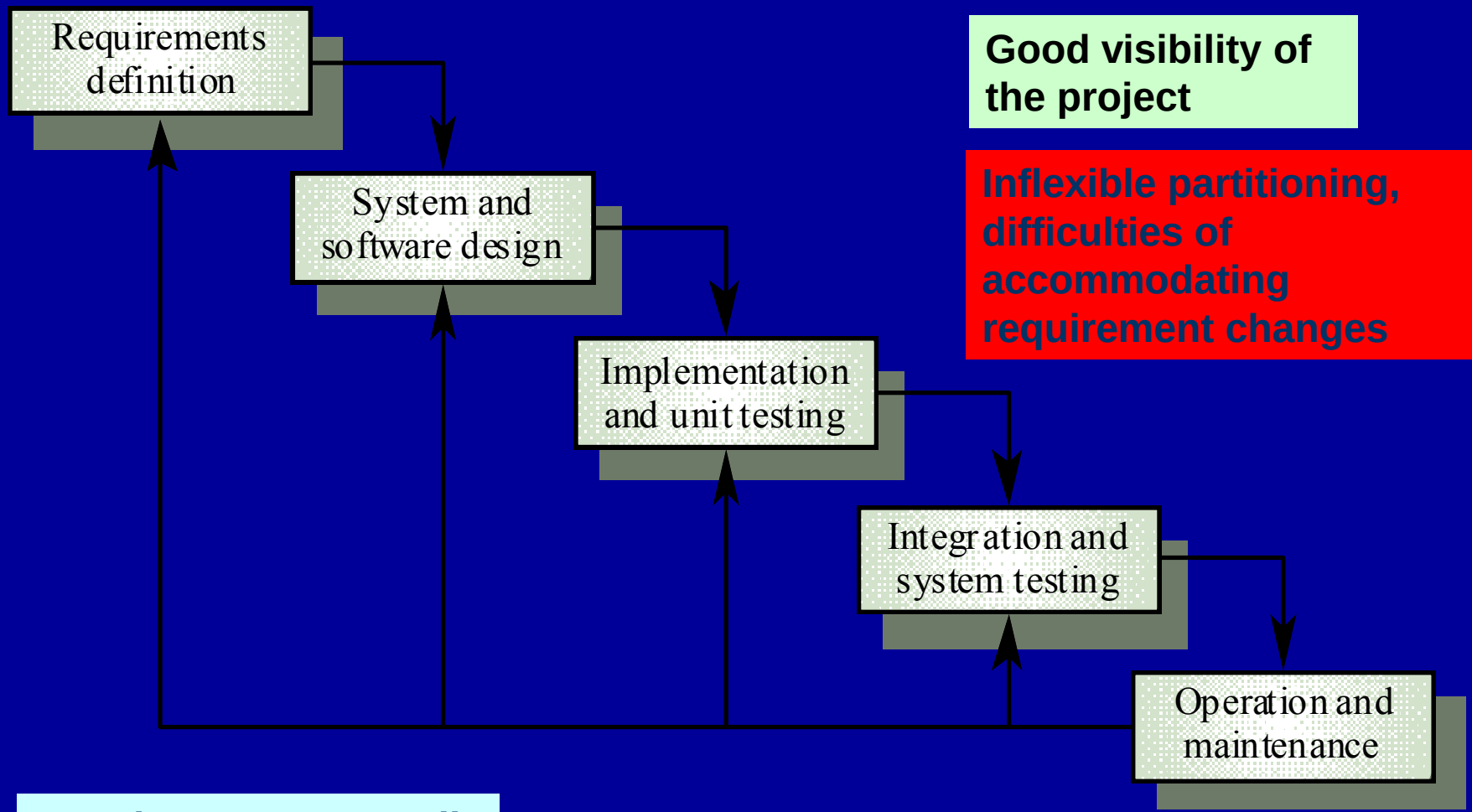
- Waterfall development
- Evolutionary development
- Formal development
- Reuse-based development
- Iterative development
- Incremental development
- Spiral development
- Rational Unified Process (RUP)

DEM

Agile methodologies

- Extreme Programming (XP)
- Scrum
- Cockburn's Crystal Family
- Open source

Waterfall development



Good visibility of the project

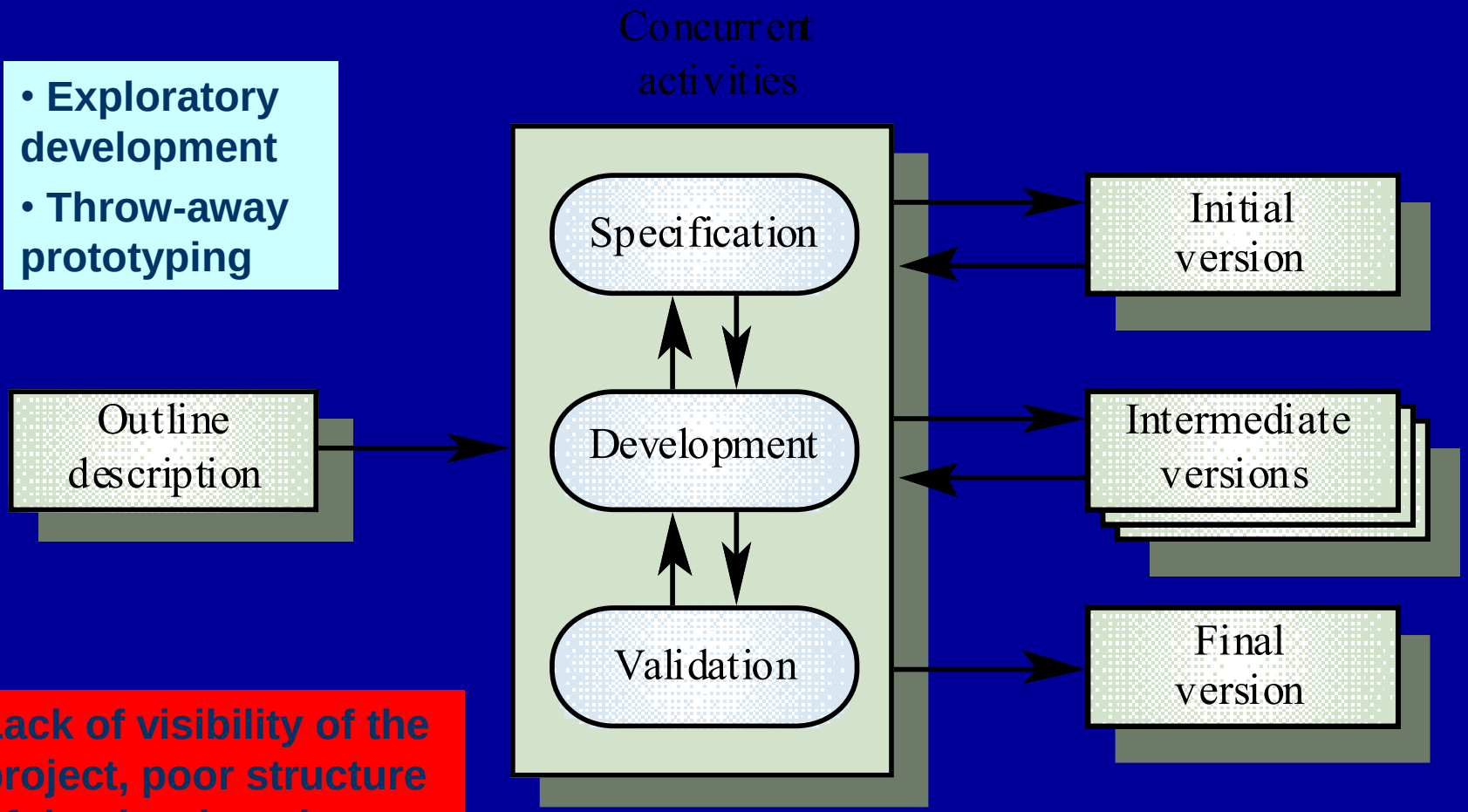
Inflexible partitioning, difficulties of accommodating requirement changes

Requirements are well-understood

DEM

Evolutionary development

- **Exploratory development**
- **Throw-away prototyping**



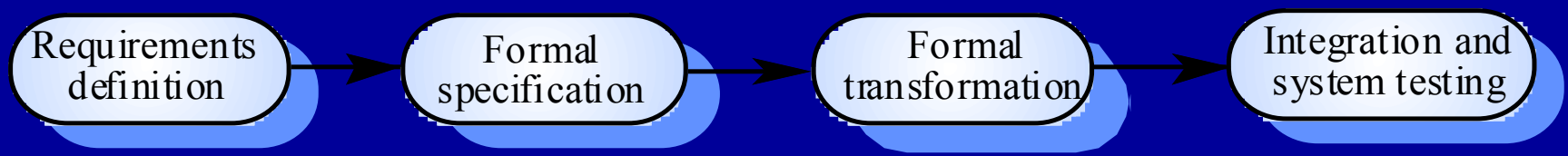
Lack of visibility of the project, poor structure of the developed software, short life systems

DEM

User interface development

Formal development

Formal verification to prove correctness



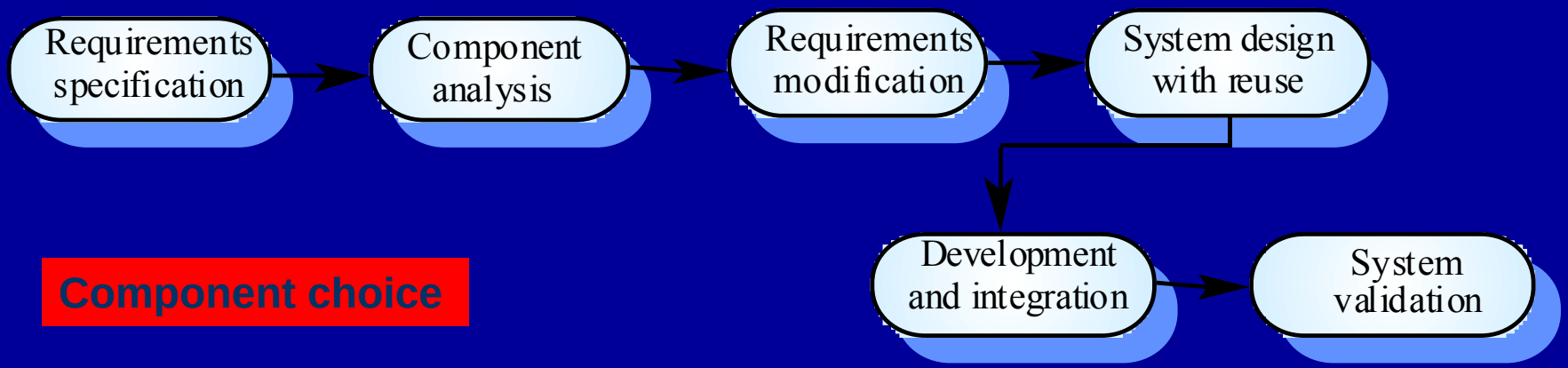
Executable program

Formal methods skills required, difficulties to formalize some aspects of the software

DEM

Reuse-oriented development

Reuse paradigm



Component choice

Web services

DEM

Iterative development

- Process iteration **Time**
  - Overall life-cycle is composed of multiple iterations
  - Each iteration goes through all activities required to develop a software
  - Each iteration can use a particular process model

DEM

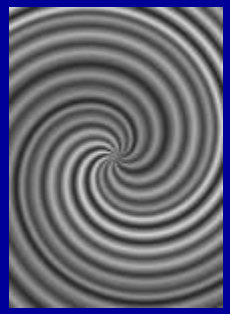
Incremental development

- Software increment
  - A set of new features to be added incrementally (step by step)
  - Features are prioritized
- Software developed as a series of delivery

Feature(s)

Lower risk of failure because delivery at each increment, highest priority features are the most tested

DEM

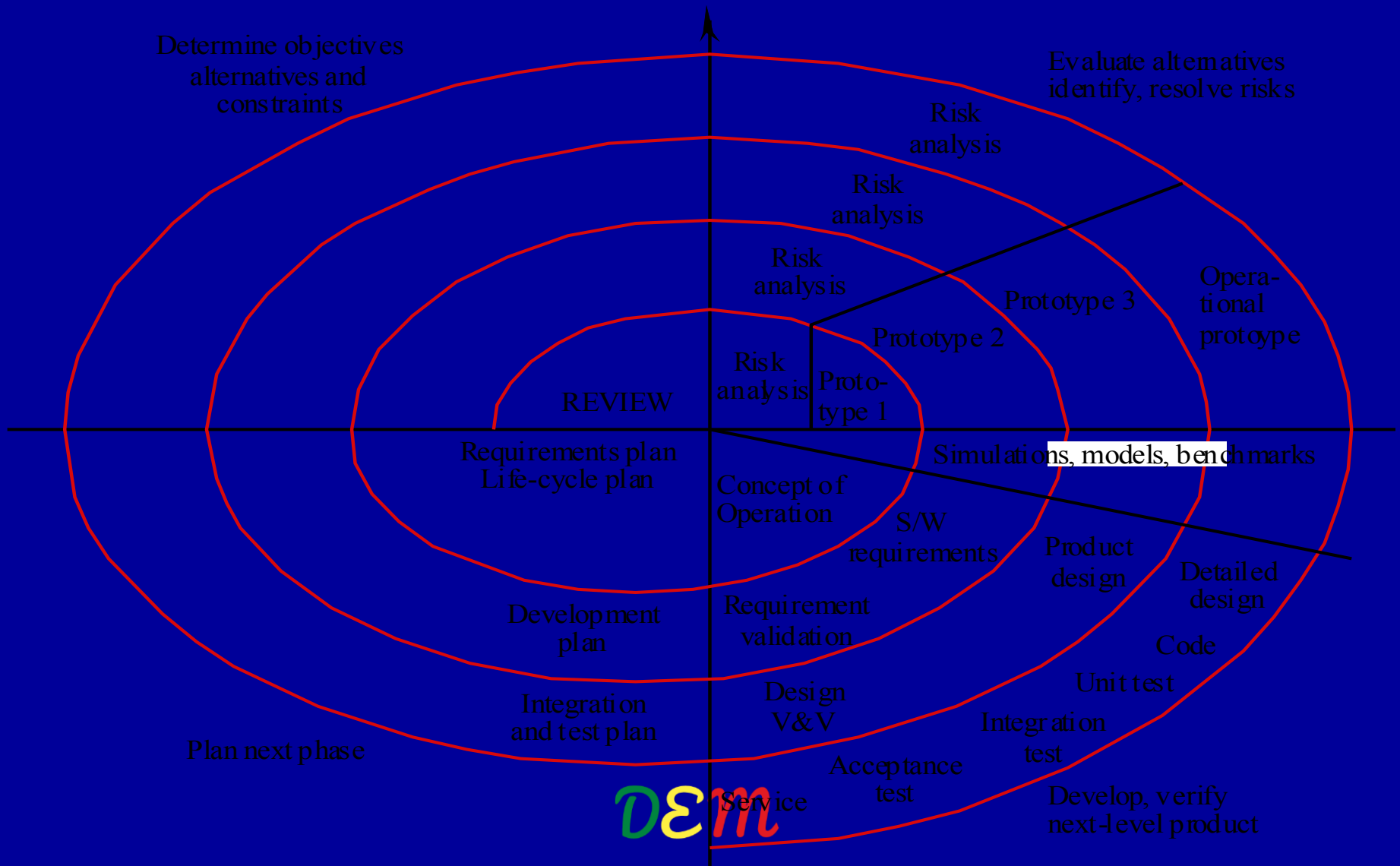


Spiral development

- Spiral view rather than sequence with backtracking
- Each loop is a phase of the process
  - Objectives for the phase are defined
  - Risks for the phase are assessed
  - Planning for the phase is done
- Choice of any development process

DEM

Spiral development



Agile methodologies

- Reaction to generic models
- Agile Manifesto – February 2001
  - http://agilemanifesto.org
  - Individuals and interactions over processes and tools
  - Working software over comprehensive documentation
  - Customer collaboration over contract negotiation
  - Responding to change over following a plan

DEM

Extreme Programming (XP)

- Kent Beck and Ward Cunningham
- Payroll project for Chrysler in 1996
- XP begins with four values
  - Communication
  - Feedback
  - Simplicity
  - Courage
- 12 practices
- Strong emphasis on testing first
- Iterations of 1 to 3 weeks

DEM

User stories

- A story mentions one feature or constraint of the desired software
- The stories are estimated by the developers (1 to 3 weeks) and prioritized by the customer
- When all stories have been implemented, the software is complete
- The stories drive the creation of tests
- Around 80 stories for a release

DEM

Principles

- Customers define application features with user stories
- XP teams put small code releases into production early
- XP teams use a common system of names and descriptions
- Teams emphasize simply written object-oriented code that meets requirements

DEM

Principles

- Designers write automated unit tests upfront and run them throughout the project
- XP teams frequently revise the overall code design, a process called refactoring
- Programmers work side by side in pairs, continually seeing and discussing each other's code
- All programmers have collective ownership of the code and the ability to change it

DEM

Principles

- XP teams integrate code and release it to a repository every few hours and never hold on to it longer than a day
- Programmers work at a sustainable pace, with no extended overtime
- A customer representative remain onsite throughout the development project
- Programmers must follow a common coding standard so all the code in the system looks as if a single individual wrote it

DEM

Pair Programming

- Two side-by-side programmers (driver and navigator) at one computer and rotate
- Collaboration on the same design, algorithm or test
- Improves productivity, quality of software and is fun
  - More than twice as fast
  - Less bugs - quality
- Remote pair programming

DEM

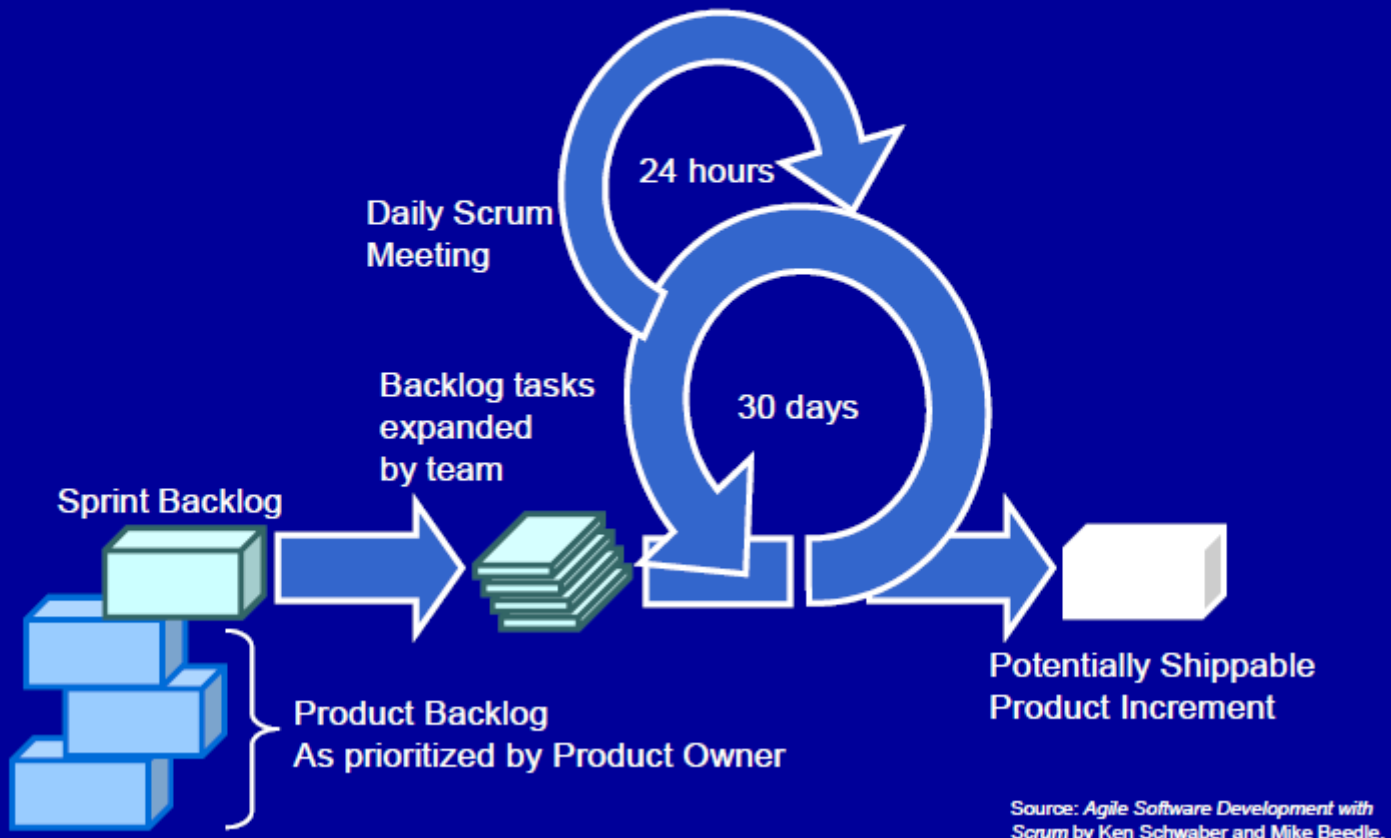
Scrum

- Developed in management in 1986
- Adapted to software development in 1993
- Jeff Sutherland and Ken Schwaber are at the origin of the Scrum framework in software development
- Empirical challenges cannot be addressed successfully in generic models
- Focus on maximizing the team's ability to respond in an agile manner to emerging challenges
- No specific process prescibed
- Short iterations (sprints) of 30 days where the software is designed, coded and tested
- Daily 30-minute stand-up meeting (Daily Scrum)
- The requirements are expressed in a backlog composed of user stories and tasks
- The Product Owner is the owner of the requirements
- The Scrum Master facilitates Scrum and removes impediments

DEM

Rally Project Management Software
IBM Rational Team Concert / Jazz

Overview of Scrum



Source: *Agile Software Development with Scrum* by Ken Schwaber and Mike Beedle. And Cohn

DEM

Roles

Product Owner: Set priorities

ScrumMaster: Manage process, remove blocks

Team: Develop product

Stakeholders: observe & advise

Key Artifacts

Product Backlog
- List of requirements & issues
- Owned by Product Owner
- Anybody can add to it
- Only Product Owner prioritizes

Sprint Goal
- One-sentence summary
- Declared by Product Owner
- Accepted by team

Sprint Backlog
- List of tasks
- Owned by team
- Only team modifies it

Blocks List
- List of blocks & unmade decisions
- Owned by ScrumMaster
- Updated daily

Increment
- Version of the product
- Shippable functionality (tested, documented, etc.)

Key Meetings

Sprint Planning Meeting
- Hosted by ScrumMaster; ½-1 day
- In: Product Backlog, existing product, business & technology conditions

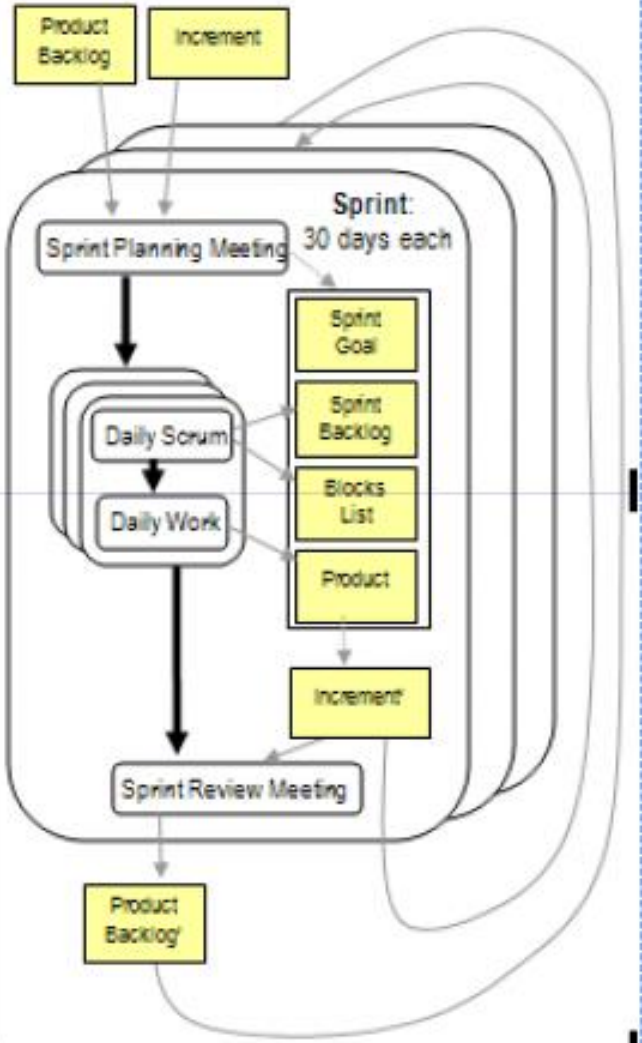
1. Select highest priority items in Product Backlog; declare Sprint Goal
2. Team turns selected items into Sprint Backlog

- Out: Sprint Goal, Sprint Backlog

Daily Scrum
- Hosted by ScrumMaster
- Attended by all, but Stakeholders don't speak
- Same time every day
- Answer: 1) What did you do yesterday? 2) What will you do today? 3) What's in your way?
- Team updates Sprint Backlog; ScrumMaster updates Blocks List

Sprint Review Meeting
- Hosted by ScrumMaster
- Attended by all
- Informal, 4-hour, informational
- Team demos increment
- All discuss
- Hold retrospective
- Announce next Sprint Planning Meeting

Development Process



DEM

Choosing the process model

- One process does not fit all!
- Choosing generic methodologies (predictive)
  - Large size team
  - Fixed price contract with requirements front end
  - Critical applications
- Choosing agile methodologies (adaptive)
  - Average size team
  - When you are used to a ad-hoc process
  - Responsible and motivated developers
  - Requirements are uncertain and volatile
  - Customer willing to get involved

DEM

References

- Software Engineering. Sommerville. Last version.

DEM

XP game

- Enough talk, let's play the XP game
- http://www.xp.be/xpgame.html

DEM