



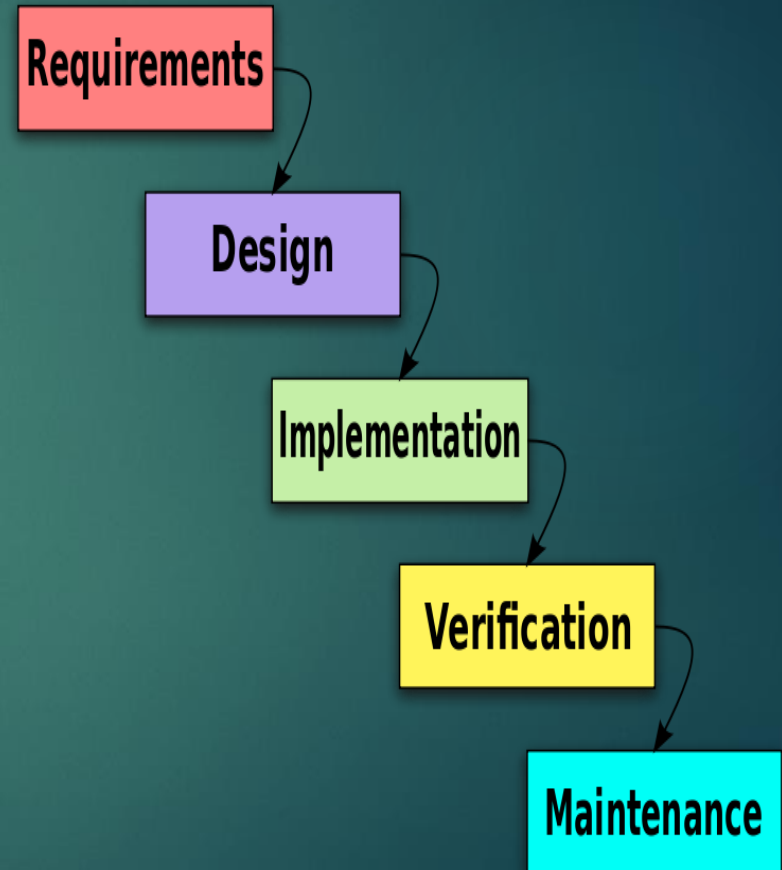
# Software Development methodologies

# Methodologies examples

- ▶ Waterfall
- ▶ Prototype model
- ▶ Incremental
- ▶ Iterative
- ▶ V-Model
- ▶ Spiral
- ▶ Scrum
- ▶ Cleanroom
- ▶ RAD
- ▶ DSDM
- ▶ RUP
- ▶ XP
- ▶ Agile
- ▶ Lean
- ▶ Dual Vee Model
- ▶ TDD
- ▶ FDD

# Waterfall

- ▶ Sequential design process
- ▶ Progress is seen as flowing steadily downwards (like a waterfall) through SDLC



- 
- The image contains three hand-drawn sketches of software interfaces:
- Top Left: Find Profile**
    - Buttons: Find, Close
    - Fields: Name (with a dropdown arrow), Description
    - Buttons: Search, Cancel, Return Selected
  - Top Right: Customer Order - Look Within Orders**
    - Buttons: Help, Close
    - Fields: Customer Name (with a dropdown arrow), Order Number (with a dropdown arrow)
    - Buttons: Show Full Header
  - Bottom: Bill Line**
    - Buttons: Change Location, Product Information, Inventory History, Cost Calculator, Comments
    - Table:
 

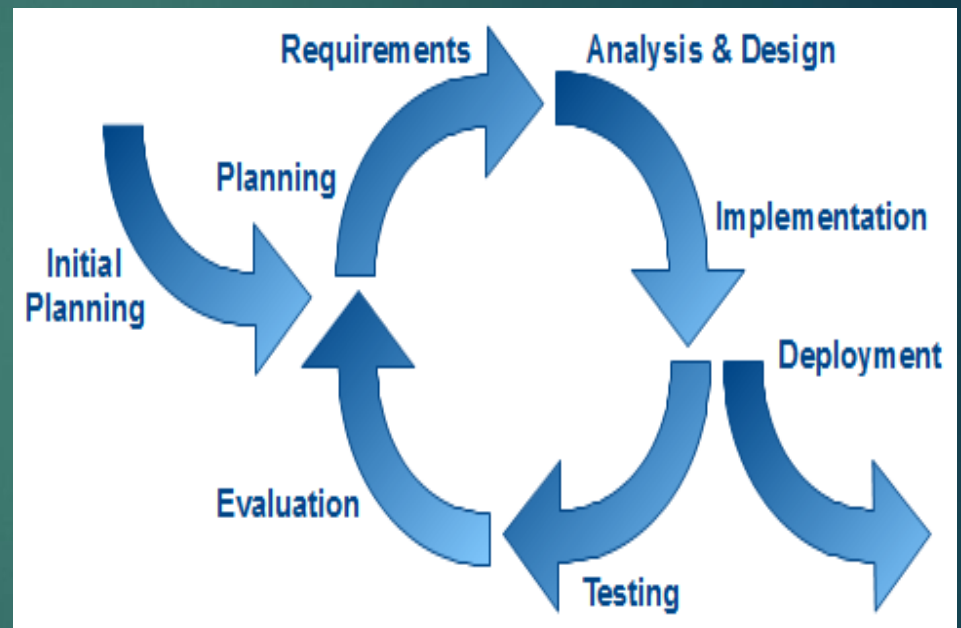
Item	Description	Quantity	Unit Price	Cost
P25-04	Single del station order	5	32.45	162.25
P24-07	Three del station order	0	14.70	123.45
224-01	Del station Truckload order	1	450.00	450.00

# Incremental Build Model

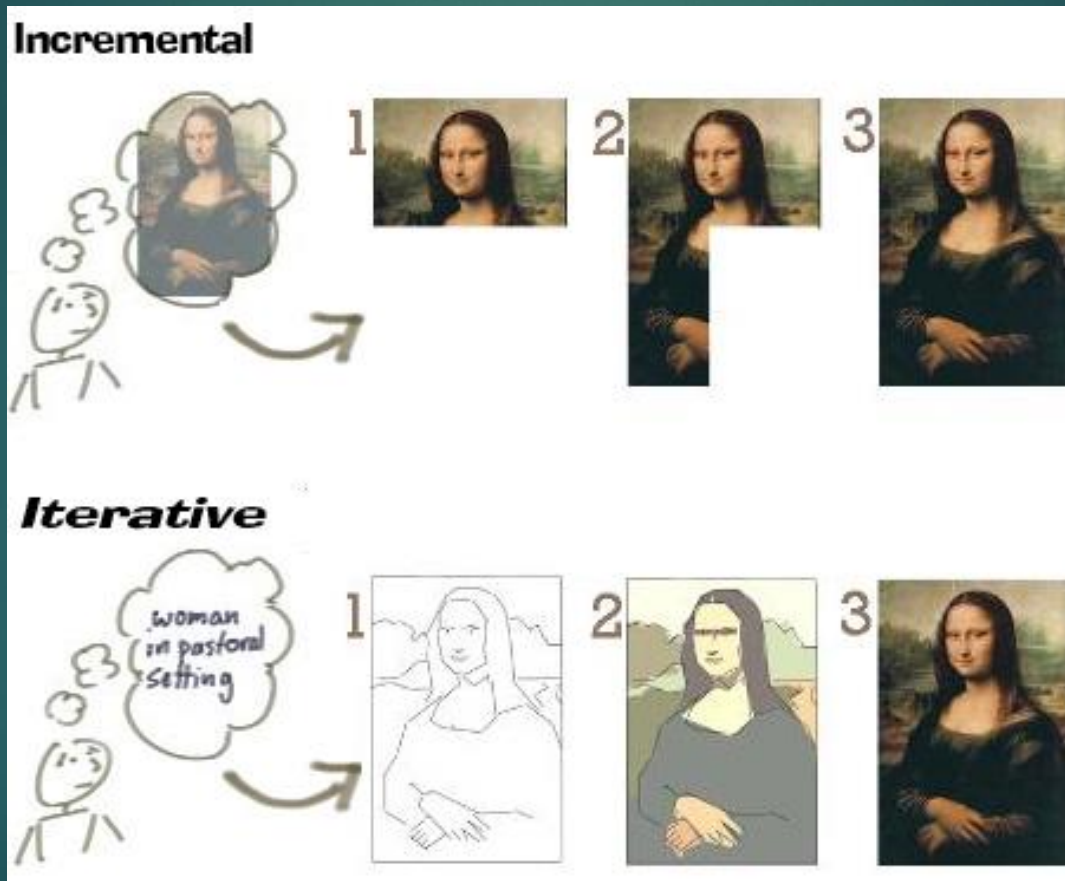
- ▶ The model is designed, implemented and tested incrementally (a little more is added each time).
- ▶ Finished when satisfies all the requirements.
- ▶ Combines the elements of the waterfall model with the iterative philosophy of prototyping.

# Iterative and Incremental Development

- ▶ Iterative and incremental development is any combination of both iterative design or iterative method and incremental build model for development.



# Incremental vs. Iterative





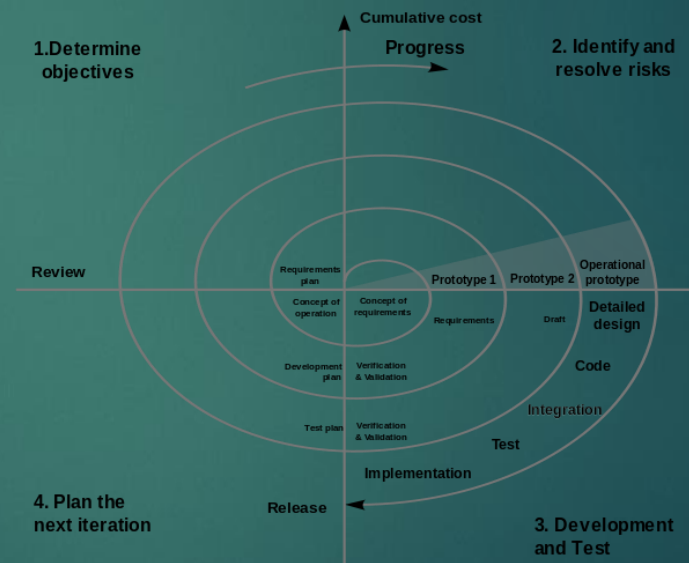
# A Bit Different Understanding





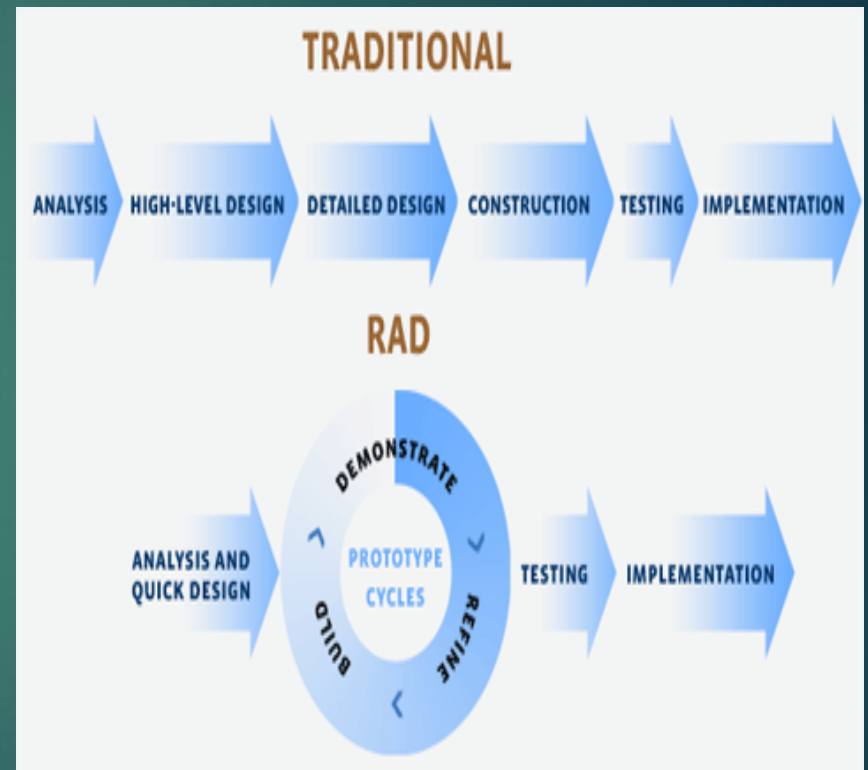
# Spiral Model

- ▶ Combining elements of design and prototyping-in-stages
- ▶ Combines the features of the prototyping and the waterfall model
- ▶ The spiral model is intended for large, expensive and complicated projects
- ▶ Advantages of top-down and bottom-up concepts



# RAD

- ▶ Minimal planning and fast prototyping.
- ▶ Developing instead of planning
- ▶ The lack of pre-planning generally allows software to be written much faster, and makes it easier to change requirements.



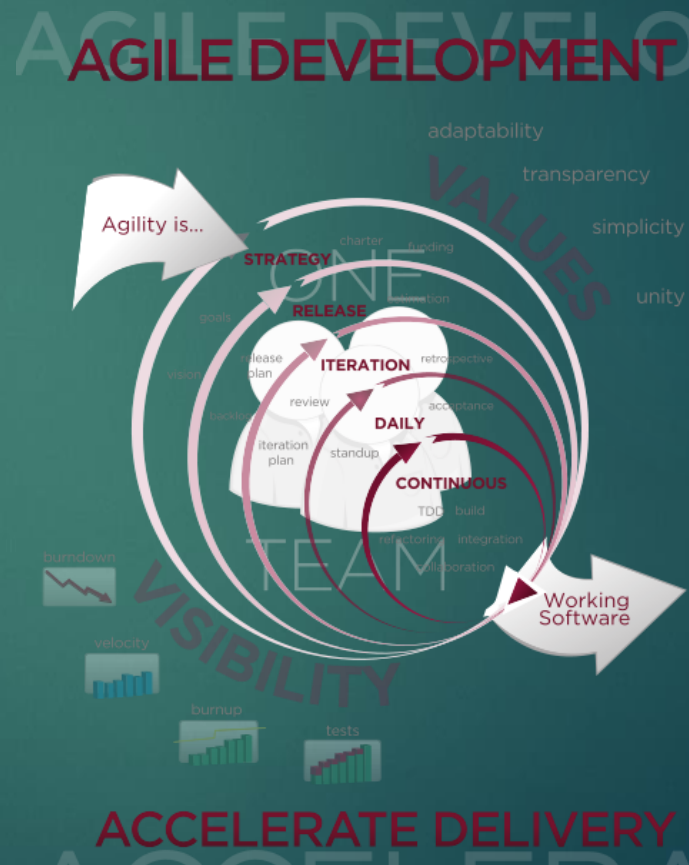
# Cleanroom



- ▶ The Cleanroom process embeds software development and testing within a statistical quality control framework.
- ▶ Mathematically-based software development processes are employed to create software that is correct by design, and statistical usage testing processes are employed to provide inferences about software reliability.
- ▶ This systematic process of assessing and controlling software quality during development permits certification of software fitness for use at delivery.

# Agile

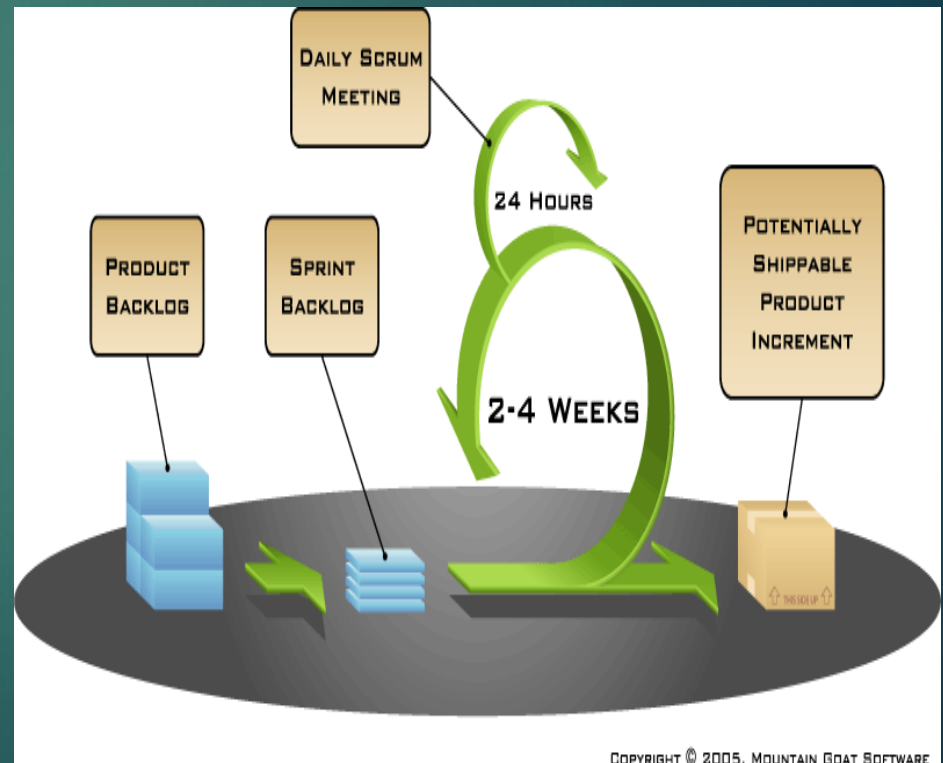
- ▶ Group of software development methods
- ▶ Based on iterative and incremental development
- ▶ Most important phrases
  - ▶ self-organizing, cross-functional teams
  - ▶ adaptive planning,
  - ▶ evolutionary development and delivery,
  - ▶ a time-boxed iterative approach,
  - ▶ rapid and flexible response to change.
- ▶ A conceptual framework
- ▶ The Agile Manifesto in 2001.





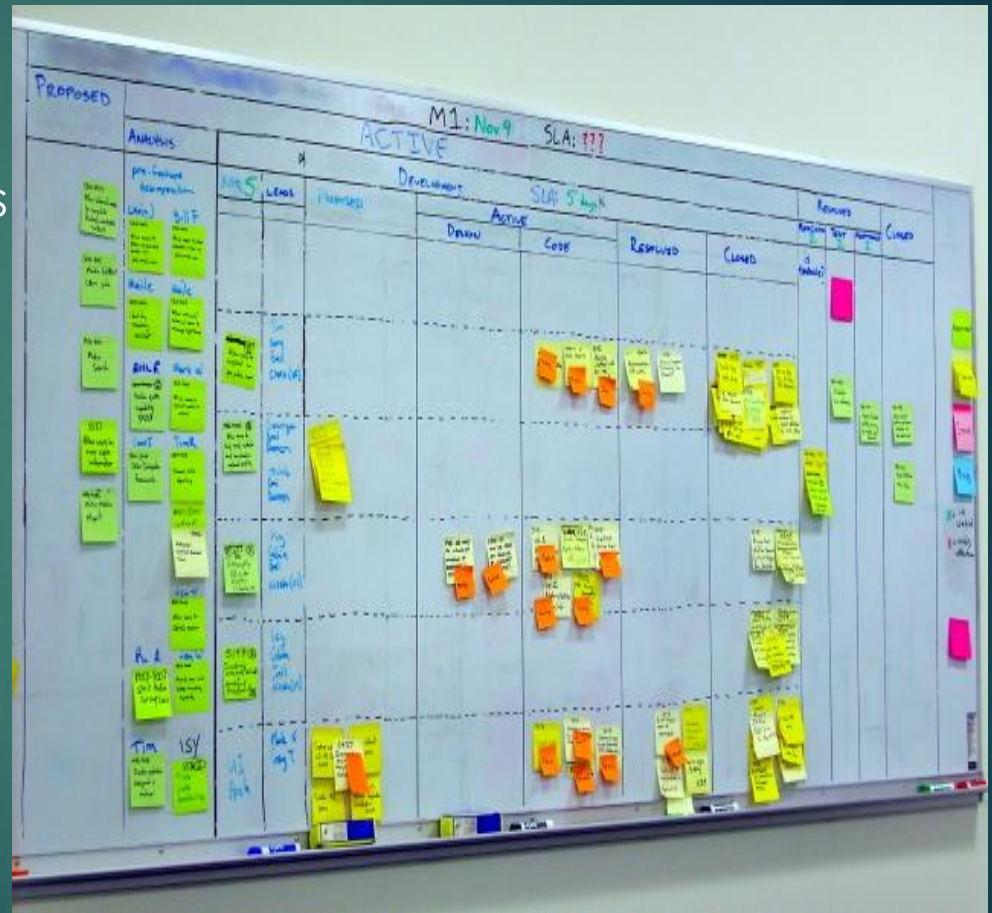
# Scrum

- ▶ Scrum is an iterative and incremental agile software development framework
- ▶ A flexible, holistic product development strategy
- ▶ Development team works as an atomic unit
- ▶ Opposing to sequential approach



# Lean (Kanban)

- ▶ A translation of lean manufacturing principles and practices
- ▶ Toyota Production System,
- ▶ Today part of Agile community.





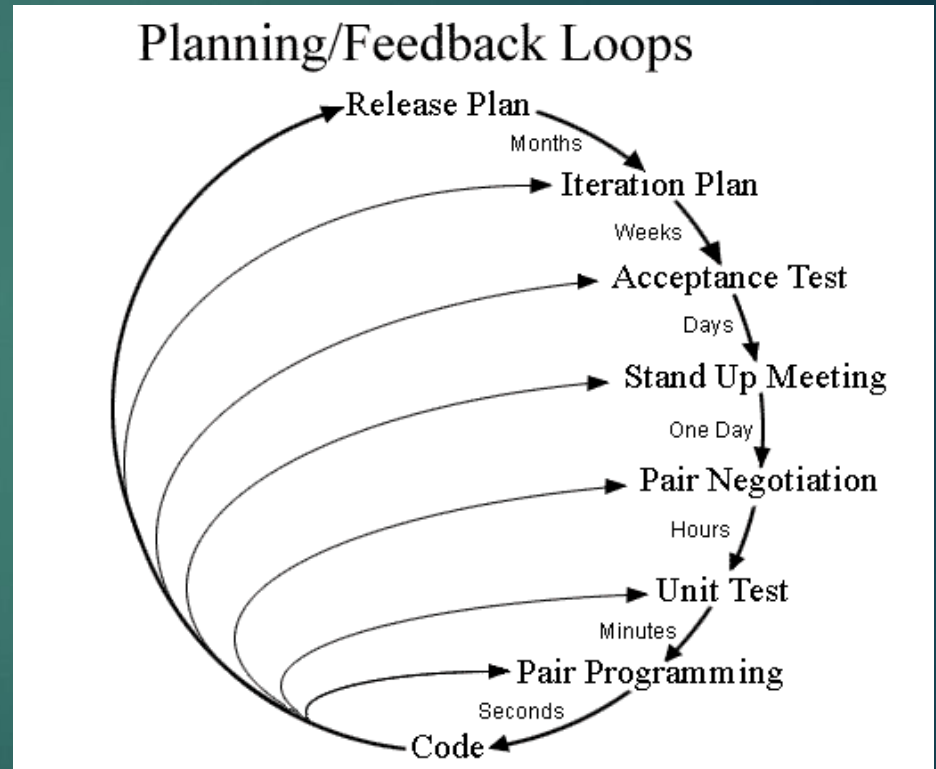
# Lean Principles



1. Eliminate waste
2. Amplify learning
3. Decide as late as possible
4. Deliver as fast as possible
5. Empower the team
6. Build integrity in
7. See the whole

# Extreme Programming (XP)

- ▶ Improve software quality and responsiveness to changing customer requirements
- ▶ A type of agile software development
- ▶ Frequent "releases" in short development cycles
- ▶ Introduce checkpoints where new customer requirements can be adopted.



# XP Concepts (examples only)

- ▶ Pair programming
- ▶ Planning game
- ▶ Test-driven development
- ▶ Continuous integration

# DSDM

- ▶ An agile project delivery framework, primarily
  - ▶ DSDM fixes cost, quality and time at the outset and uses the MoSCoW prioritization of scope
  - ▶ Pareto principle
- ▶ **M - MUST:** Describes a requirement that must be satisfied in the final solution for the solution to be considered a success.
  - ▶ **S - SHOULD:** Represents a high-priority item that should be included in the solution if it is possible. This is often a critical requirement but one which can be satisfied in other ways if strictly necessary.
  - ▶ **C - COULD:** Describes a requirement which is considered desirable but not necessary. This will be included if time and resources permit.
  - ▶ **W - WOULD:** Represents a requirement that stakeholders have agreed will not be implemented in a given release, but may be considered for the future.

# Test-driven development (TDD)

- ▶ Relies on the repetition of a very short development cycle: first the developer writes an (initially failing) automated test case that defines a desired improvement or new function, then produces the minimum amount of code to pass that test, and finally refactors the new code to acceptable standards.
- ▶ Test-first programming concept of extreme programming in the beginning
- ▶ Today standalone methodology

# Feature-driven development (FDD)

- ▶ Iterative and incremental development process.
- ▶ An Agile method
- ▶ Driven from a client-valued functionality (feature) perspective
- ▶ Mostly part of other methodologies

# Rational Unified Process (RUP)

- ▶ An iterative software development process framework created by the Rational Software Corporation (IBM)
- ▶ Not a concrete prescriptive process, but an adaptable framework, intended to be tailored by the development organizations
- ▶ Expected to select elements of the process that are appropriate



# V-model

- ▶ The V-model is an extension of the waterfall model.
- ▶ Show the relationships between development phases and test phases
- ▶ Time and project completeness vs. level abstraction

