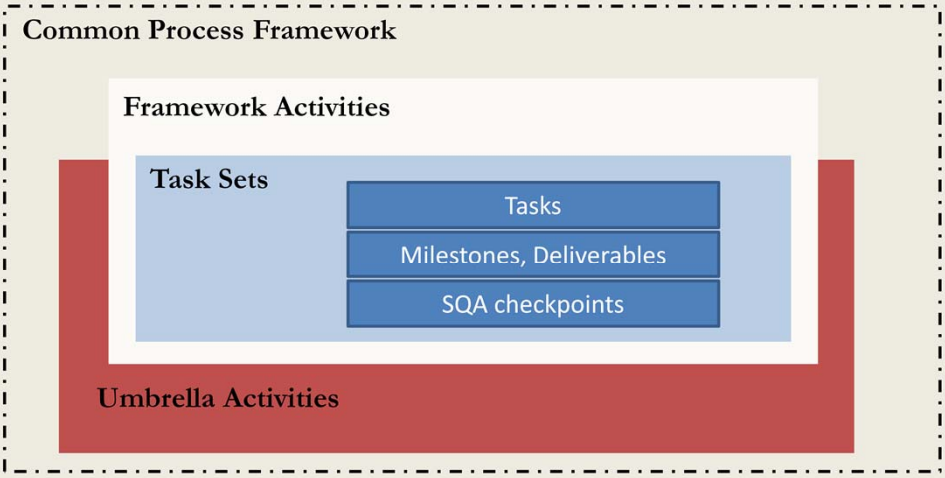
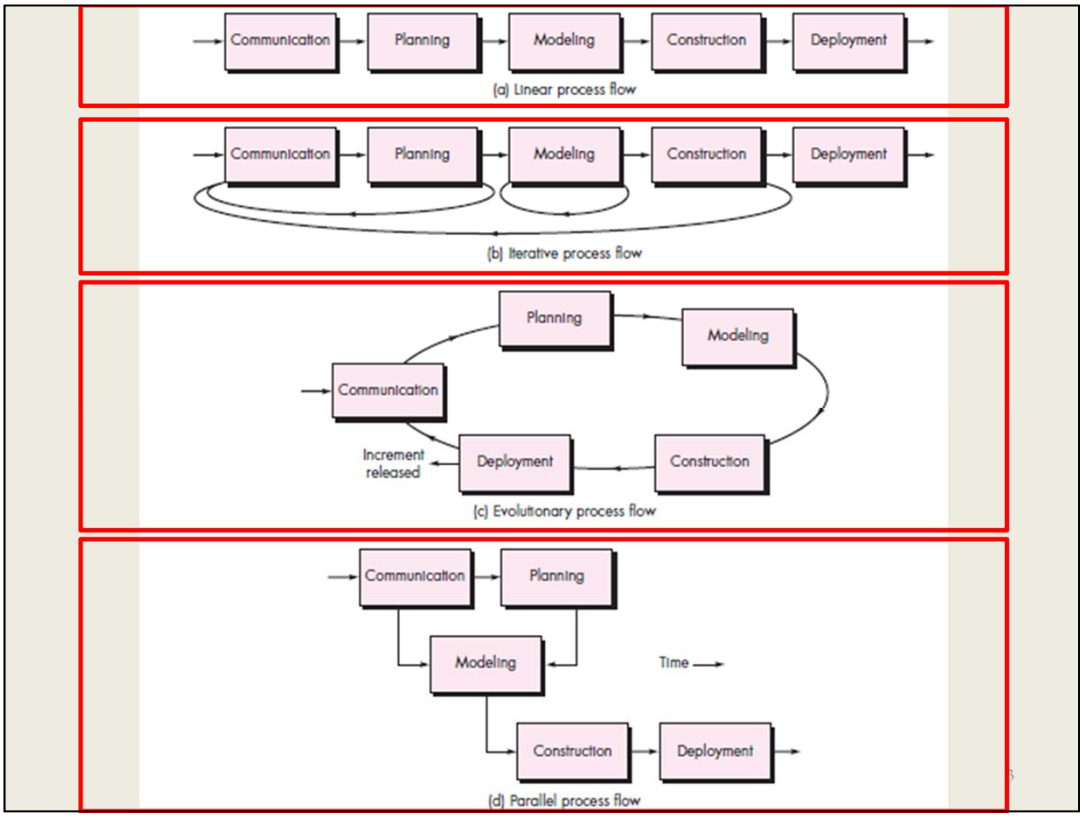
Software Process
(*process models*)

Software process: a generic view



- Process flow
  - describes how the framework activities and the actions and tasks in each activity are organized with respect to *sequence and time*

2


Communication
Planning
Modeling
Construction
Deployment
(a) Linear process flow
Communication
Planning
Modeling
Construction
Deployment
(b) Iterative process flow
Planning
Modeling
Communication
Increment released
Deployment
Construction
(c) Evolutionary process flow
Communication
Planning
Modeling
Time
Construction
Deployment
(d) Parallel process flow

Categories of Process Models

- Prescriptive model
  - Sequential model
  - Incremental model
  - Evolutionary model
- Specialized process model
  - Component-based model
  - Formal method model
  - Aspect-Oriented process model
- Unified process model
  - Software development with UML

Prescriptive Process Models (1)

5

Prescriptive: traditional

Challenges in Building Software Processes

Have a right process built-in worth half of journey in software development, but, it is far more easier to say than done, some reasons:

- Different development organizations have different products, which have different requirements on quality and schedule.
  - E.g. Consumer software applications v.s. avionic software.
- Customers vary from company to company, they have different understanding on,
  - Requirements.
  - Schedules.
    - E.g. Web application v.s. embedded applications.
- Different organizations have different marketing/pricing/maintenance strategies.
  - E.g. from quality-oriented release cycle to twice a year.
- Teams and developers have their own work habit.

6

iPhone game app and an autopilot system in a 747, or whatever fancy stuff NASA is using on their mars rover

Prescriptive Process Models

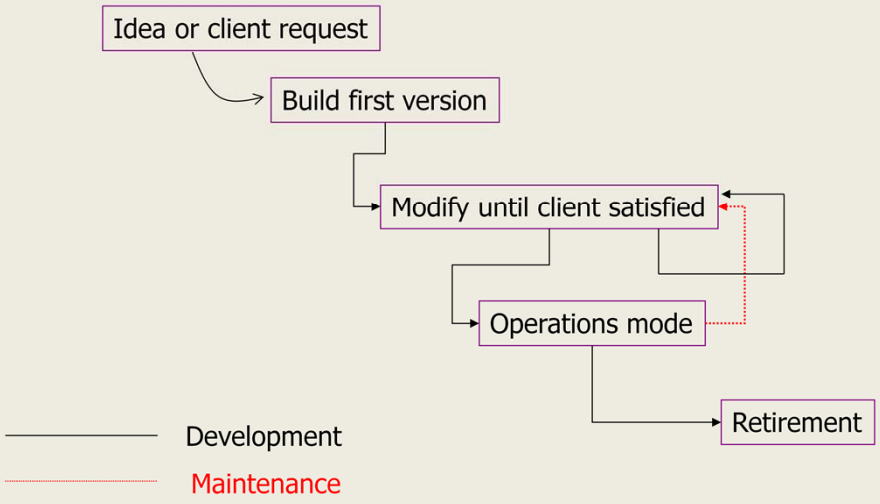
- Prescriptive process models provide formula to organize software development processes. They do,
  - Advocate an orderly approach to software engineering.
  - Summarize best practices in software processes.
- These models reflect different philosophies and viewpoints. Before proceeding, remember
  - No silver bullet. Each process model has its own characteristics, strength and weakness.
  - Best process models are those customized for individual organization's need.
    - You may even mix and match these process models.
- Process models provide blue-prints to implement your own software process.

Categories of Prescriptive Models

- Prescriptive models
  - Sequential models.
    - Waterfall models.
    - V-model
  - Incremental models:
    - Incremental model.
    - Rapid Application Development (RAD) model.
  - Evolutionary models
    - Prototyping.
    - Spiral model: risk-driven process.

8

Ad-hoc: Build-and-Fix model



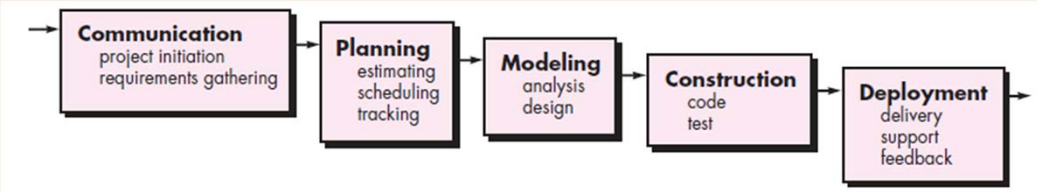
9

Build-and-Fix model

- Development is done in an unordered and demand-driven fashion,
  - Product is constructed without specifications.
  - There is no explicit design stage.
    - An ad-hoc design will likely evolve in the mind of the developer.
  - No verification and validation stage.
- May work for small projects, but,
  - Development activities are not organized and easily result in chaos.
  - No measurements are taken towards quality of software.

10

Waterfall Model



- Classic software process model (life cycle)
  - Developed in 70s for aerospace and defense projects.
- Features: a linear pipeline model with multiple stages.

11

Waterfall Model

- Framework activities are:
  - Performed in sequence
  - Result of one flows(falls) into the next
- Advantages
  - Disciplined approach.
    - Software processes are phased by framework activities.
  - Quality assurance is built into process.
    - Quality assurance/review are performed as last step of each stage.
    - Testing/analysis can be a separate stage or share a stage with implementation.

12

the staged development cycle enforces discipline: every phase has a defined start and end point, and progress can be conclusively identified (through the use of milestones) by both vendor and client.

The emphasis on requirements and design before writing a single line of code ensures minimal wastage of time and effort and reduces the risk of schedule slippage, or of customer expectations not being met.

Waterfall Model

- Drawbacks.
  - Fundamental flaw: Assumption that each stage can and must be completed before the next one occurs
    - Counterexample: user may need to see finished product to express true requirements!
  - Limited or no interaction and feedback among stages.
    - No refinement on software design based on user's reaction to initial product prototype.

13

Customer has to be patient

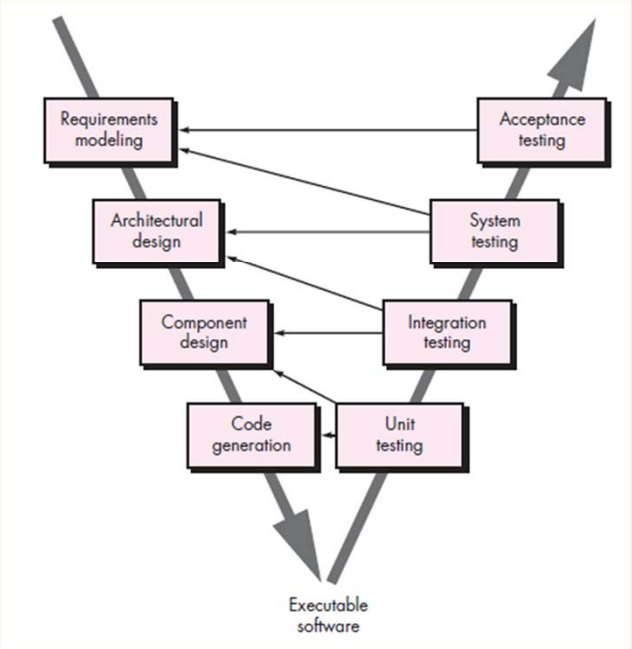
Useful scenarios?

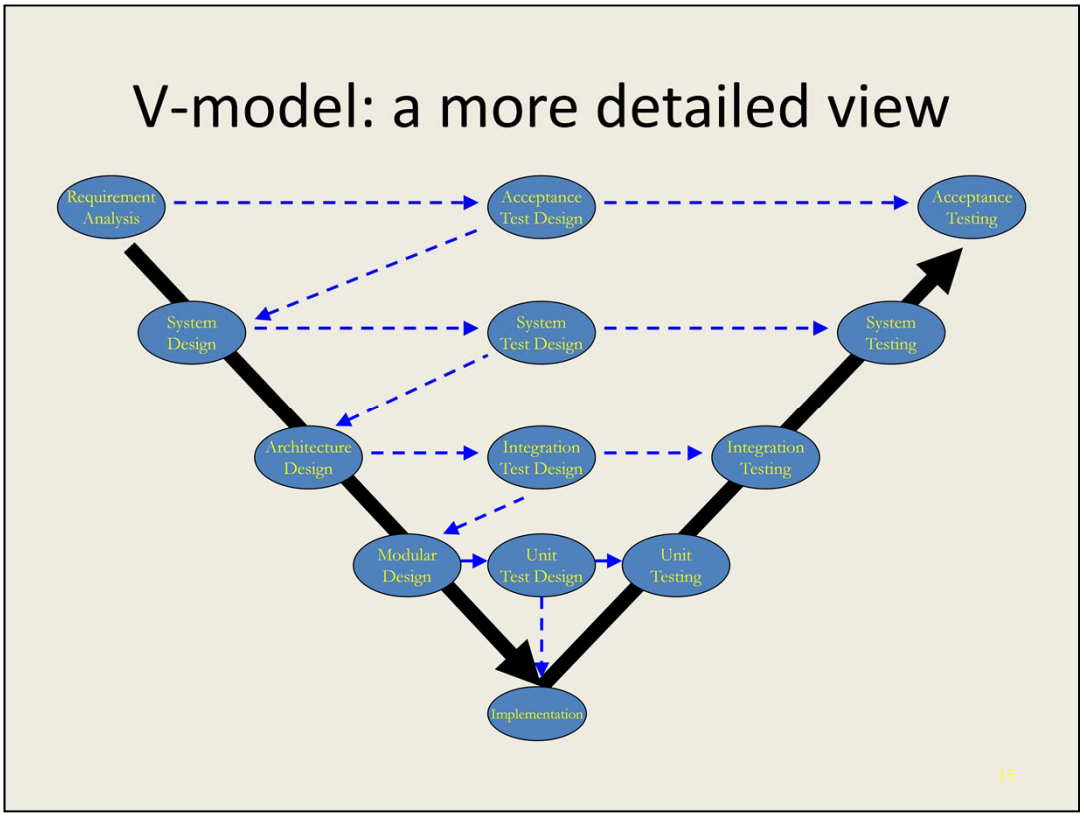
Feasibility of design. Problems won't be realized until implementation.

designs that look feasible on paper turn out to be expensive or difficult in practice, requiring a re-design and hence destroying the clear distinctions between phases of the traditional waterfall model.

the waterfall model implies a clear division of labor between, say, "designers", "programmers" and "testers"; in reality, such a division of labor in most software firms is neither realistic nor efficient.

V-Model: for Verification and Validation





Left side: decomposition of specification and implementation. Each step is a refinement of previous step with more implementation details.

Right side: quality control. To verify the implementation with respect to their implementation.

Unit testing: smallest testable unit;

integration: testing the separate modules/components together to reveal

errors in interfaces and interactions; black box

System testing: compare the system specifications against the system; REASONS: first time testing wrt requirements, not technical specification – customer and user might be different; errors only available in the whole system

Acceptance: Tested by users/customers. Determine accpeptance.

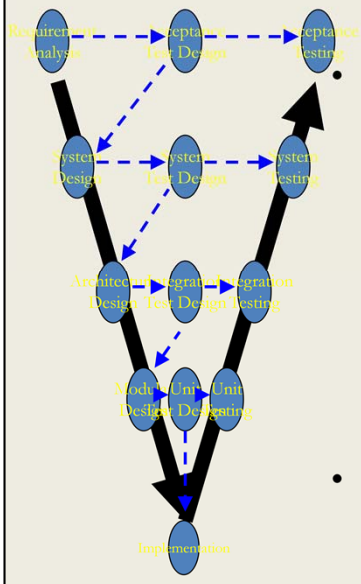
V-model

- V-model is a process model centralized for quality assurance activities.
  - "The world according to a quality manager"
- First proposed for German federal administration and defense projects.
  - Have since been adopted by many leading software companies as the model for software engineering with an emphasis on product quality.
  - Have many variants.
  - V stands for "Verification and Validation"

16

Medical device industry

V-model

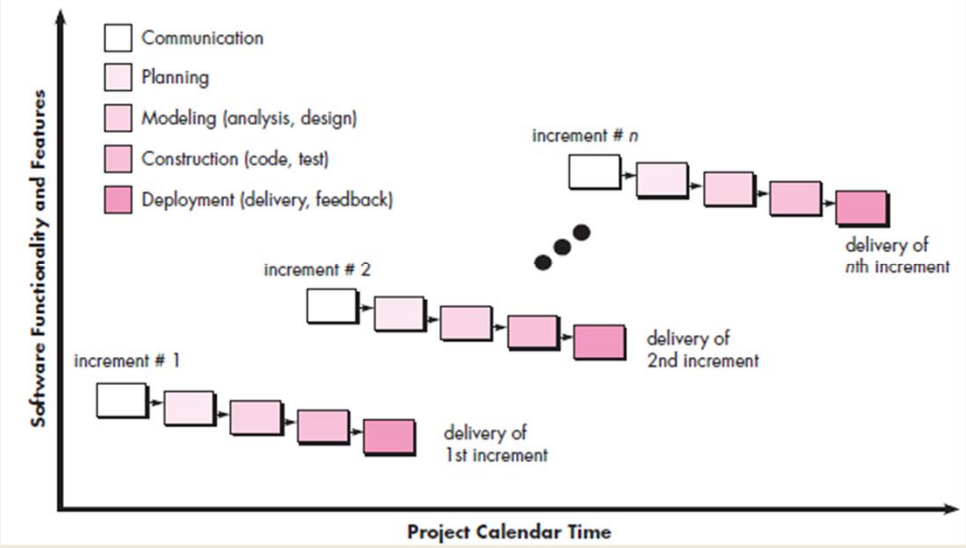


- The left side of “V” is the decomposition of specification and implementation.
  - Development efforts can be seen as the decomposition of requirements.
  - Decomposition has many steps:
    - Requirement Analysis
    - System designs
    - Architecture designs
    - Modular designs
    - Implementation
  - Each step is a refinement of previous step with more implementation details.
- The right side of “V” is the V&V activity to verify the implementation according to its requirements.
  - Tests are developed to test each step of decomposition.

17

1. Too simple, false sense of security.
2. Inflexible – but has been changed
3. Testing methodologies

Incremental Model



Each stage of development process is a mini-waterfall
- *combing linear and parallel process flows*

18

Incremental Model

Software process is a sequence of linear development processes, each of which uses a waterfall model.

- Features are prioritized and increments defined.
- Product is designed, implemented, and integrated as a series of incremental *builds.*
- A new build integrates code from previous build and new code.

19

Incremental Model

- Advantages:
  - Maintenance is no longer a single stage.
    - Maintenance phases becomes subsequent sequences of mini-waterfall models.
    - Every iteration is supposed to improve the quality of software and/or add new features.
  - Spreading risks
    - Unlike linear development process in waterfall model, each cycle ends with a usable system.
    - Development can be stopped when quality and functionalities of software reaches the goal.

20

Priority of features

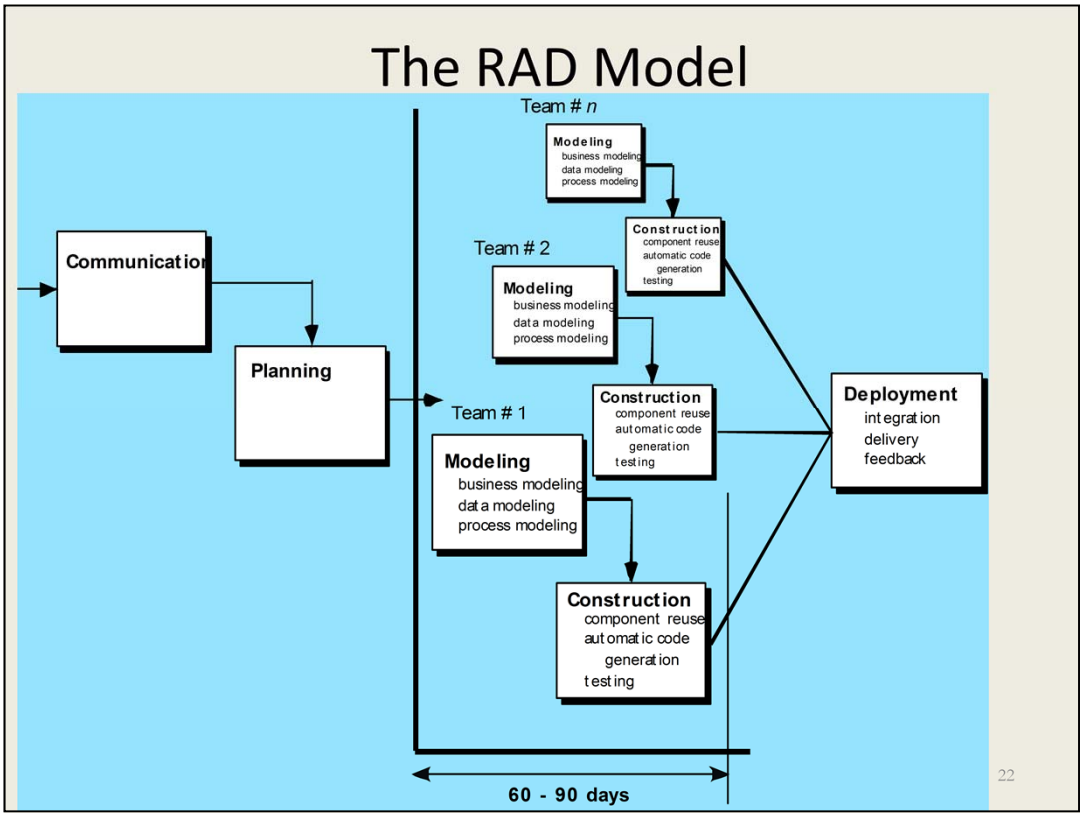
Incremental Model

Drawbacks

- Limit feedback and interaction.
  - Each cycle is still a waterfall model. Suggestions to current release can only be incorporated into the next product release.
- Stress on product scheduling.
  - Requiring an operational product at each increment will put stress on overall schedule of product development.

21

Heroes of might and magic 6



Rapid Application Development

High-speed adaptation of waterfall, achieved by using a component-based construction approach.

If requirements are well understood and project scope is constrained 60 to 90 days

The RAD model

- An incremental software process model that emphasizes a short development cycle and component-based design.
  - Shared communication stage to establish consensus on a component-based design.
  - Components are developed concurrently by different teams using waterfall model.
  - Used primarily in information systems world, where tasks are well understood.
- Advantages.
  - Short development time.
  - Development activities can be concurrently carried out by different teams.
- Potential problems with high risk projects
- Works best in well understood environment

The RAD model

- Drawbacks
  - Works best only in well understood environment, where teams develop consensus on component-based design.
  - Potential problems with high risk projects.
    - The duration of project is decided by the component that takes the most time to develop.
  - Stress on human resources.
    - Having teams concurrently working on a project will require a large pool of developers.

24

Cannot be properly modularized;

New application makes heavy use of new technology

Large but scalable, human resources; people need to be committed to the very intense working environment

Summary

- Process flows
- process models
  - Challenges
  - Categorization
- Prescriptive process model
  - Waterfall & V-model
  - Incremental model
  - RAD