Software Process

Overview

- What is **software process?**
- Examples of process models
- Agile software development

Software Process

- Definition [Pressman]
  - a framework for the tasks that are required to build high-quality software.
  - to provide stability, control and organization to an otherwise chaotic activity

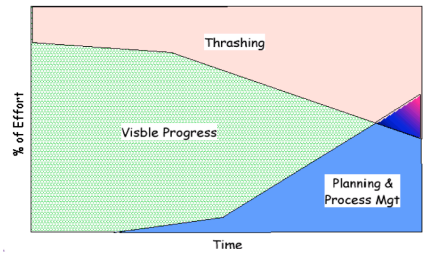
Code-and-Fix Process

- The first thing people tried in the 1950s
  1. Write program
  2. Improve it (debug, add functionality, improve efficiency, …)
  3. GOTO 1
- Works for small 1-person projects and for some CS course assignments

Problems with Code-and-Fix

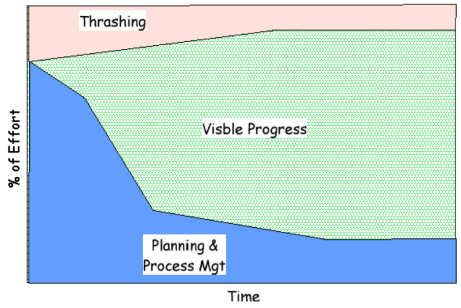
- Poor match with user needs
- Bad overall structure – No blueprint
- Poor reliability - no systematic testing
- Maintainability? What's that?
- What happens when the programmer quits?

Code-and-Fix Process



From McConnell, After the Goldrush, 1999

A More Advanced Process

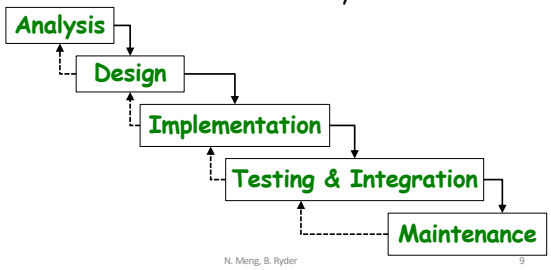


Examples of Process Models

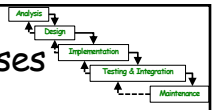
- Waterfall model
- Prototyping model
- Spiral model
- Incremental model

Waterfall Model

- The "classic" process model since 1970s
  - Also called "software life cycle"



Waterfall Phases



- Analysis: Define problems
  - requirements, constraints, goals and domain concepts
- Design: Establish solutions
  - System architecture, components, relationship
- Implementation: Implement solutions
- Testing and integration: Check solutions
  - Unit testing, system testing
- Maintenance: the longest phase

Key Points of the Model

- The project goes through the phases sequentially
- Possible feedback and iteration across phases
  - e.g., during coding, a design problem is identified and fixed
- Typically, few or no iterations are used
  - e.g., after a certain point of time, the design is "frozen"

Waterfall Model Assumptions

- All requirements are known at the start and stable
- Risks(unknown) can be turned into known through schedule-based invention and innovation
- The design can be done abstractly and speculatively
  - i.e., it is possible to correctly guess in advance how to make it work
- Everything will fit together when we start the integration

Pros and Cons

- Pros: widely used, systematic, good for projects with well-defined requirements
  - Makes managers happy
- Cons:
  - The actual process is not so sequential
    - A lot of iterations may happen
  - The assumptions usually don't hold
  - Working programs are not available early
    - High risk issues are not tackled early enough
  - Expensive and time-consuming

When would you like to use waterfall?

- Work for big clients enforcing formal approach on vendors
- Work on fixed-scope, fixed-price contracts without many rapid changes
- Work in an experienced team

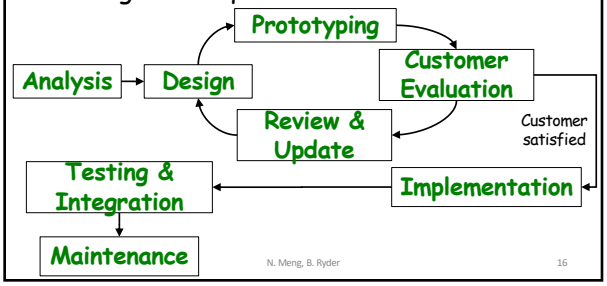
Observation

Standish group 1995

- Top three reasons for at least partial failure projects
  - lack of user input
  - incomplete requirements, and
  - changing requirements

Prototyping Model

- Build a prototype when customers have ambiguous requirements

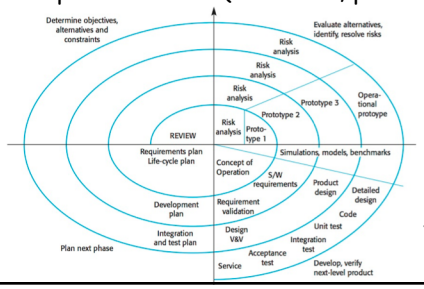


Key Points of the Model

- Iterations: customer evaluation followed by prototype refinement
- The prototype can be paper-based or computer-based
- It models the entire system with real data or just a few screens with sample data
- Note: the prototype is thrown away!

Spiral Model

- A risk-driven evolutionary model that combines development models (waterfall, prototype, etc.)



Spiral model (SOM)

Spiral Phases

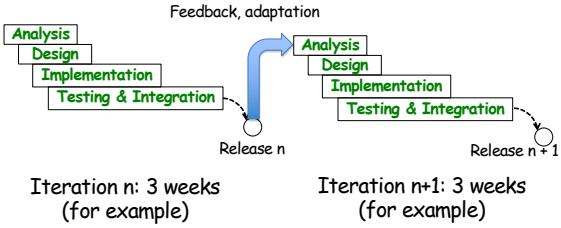
- Objective setting
  - Define specific objectives, constraints, products, plans
  - Identify risks and alternative strategies
- Risk assessment and reduction
  - Analyze risks and take steps to reduce risks
- Development and validation
  - Pick development methods based on risks
- Planning
  - Review the project and decide whether to continue with a further loop

Key Points of the Model

- Introduce risk management into process
- Develop evolutionary releases to
  - Implement more complete versions of software
  - Make adjustment for emergent risks

Incremental Model

- A sequence of waterfall models



Key Points of the Model

- Iterative: many releases/increments
  - First increment: core functionality
  - Successive increments: add/fix functionality
  - Final increment: the complete product
- Require a complete definition of the whole system to break it down and build incrementally

Agile Software Development

- A timeboxed iterative and evolutionary development process
- It promotes
  - adaptive planning
  - evolutionary development,
  - incremental delivery
  - rapid and flexible response to change

Any iterative method can be applied in an agile spirit.

Key Points of Agile Modeling

- The purpose of modeling is primarily to understand, not to document
- Modeling should focus on the smaller percentage of unusual, difficult, tricky parts of the design space
- Model in pairs (or triads)
- Developers should do the OO design modeling for themselves
- Create models in parallel
  - E.g., interaction diagram & static-view class diagram