

Software Development Processes

February 2, 2021

Overview

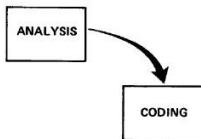
- 1 Background
- 2 Waterfall
- 3 Improvements
- 4 Attempts at fixing Waterfall

What is a process?

- Definitions from the web:
 - procedure: a particular course of action intended to achieve a result
 - a systematic series of actions directed to some end
 - a sustained phenomenon or one marked by gradual changes through a series of states
 - a continuous action, operation, or series of changes taking place in a definite manner
- What is the difference between a process and an algorithm?

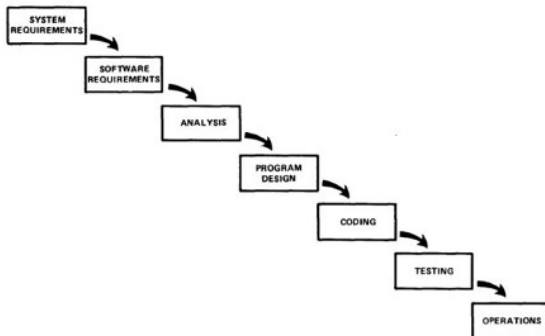
Royce's Seminal Paper

- Context
 - What was he building?
 - What tools were available to him?
- Two Steps



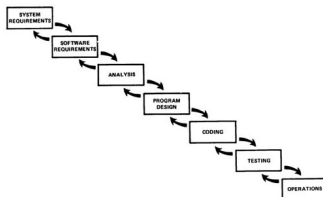
- Sufficient for small projects where developers are the users
- Notes that customer is willing to pay for both parts because each has a valuable product

Waterfall



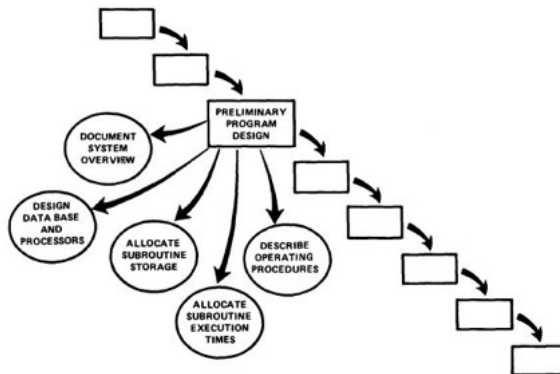
- Two steps of requirements gathering (system and software)
- Additional steps (analysis and testing) are “treated separately . . . because they are distinctly different . . . must be planned and staffed differently.”

Back Edges



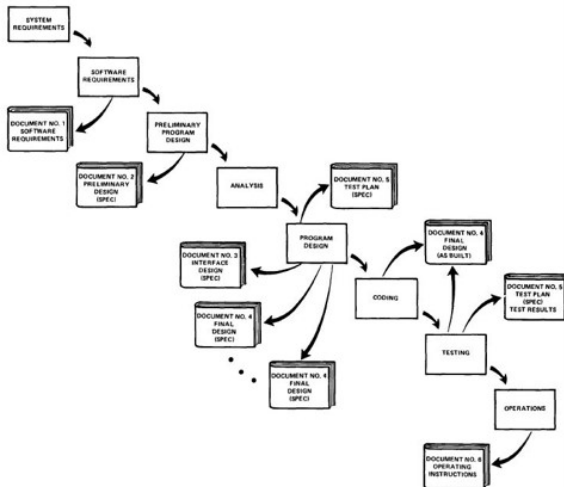
- Current interpretation of this diagram is “when we recognize there is a defect, we should go back to that step and fix it.”
- Royce says it should ALWAYS happen! “as each step progresses and the design is further detailed, there is an iteration with the preceding and succeeding steps”
- How does Royce assess this process?
 - “...risky and invites failure”
 - Testing is late
 - Timing, storage, and I/O speeds cannot be predicted accurately
 - Missing those constraints can affect the ability to meet requirements resulting in complete re-design
 - Is that still true today?
 - Are there constraints we can miss that could cause complete re-design?

Revision Step 1



- Put preliminary design between requirements and analysis
- Analysis is detailed math from his problem space
- Real-time requirements and storage capacity are critical for him
- Creates new role: program designer

Revision Step 2 - Document the Design

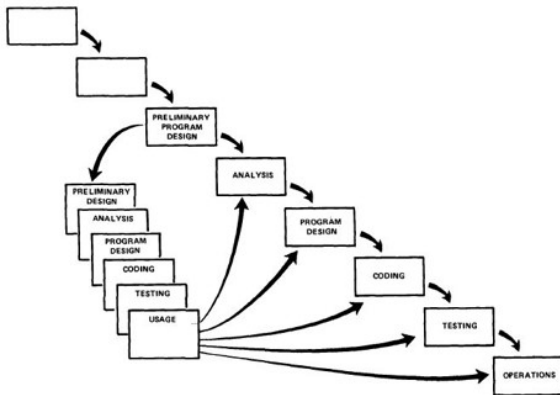


(Discussion on next slide)

Revision Step 2 - Document the Design

- What is the motivation behind this step?
 - Communication
 - So we will be able to track progress
- Documentation IS specification IS design until coding starts
- Look at the Figure: “design (spec)” vs. “design (as built)”

Revision Step 3 - Do It Twice

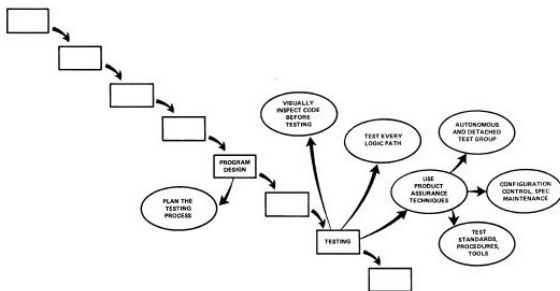


(Discussion on next slide)

Revision Step 3 - Do It Twice

- Justification is based on real-time and storage restrictions
- Can we justify it today?
 - Complexity of systems. Even without those restrictions, today's systems are "not precisely analyzable."
- What does he put in the first pass?
 - "forget the straightforward aspects of the design which aren't worth studying at this early point"

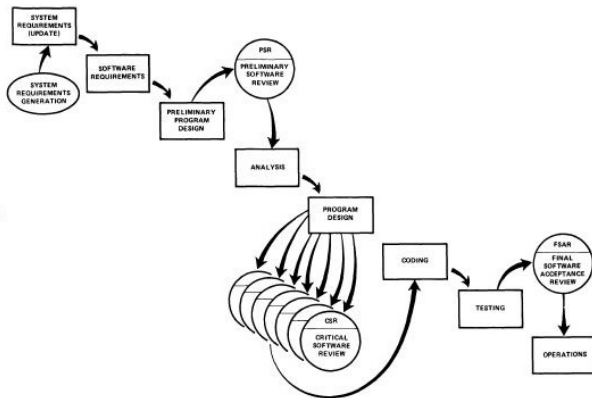
Revision Step 4 - Plan, Control, and Monitor Testing



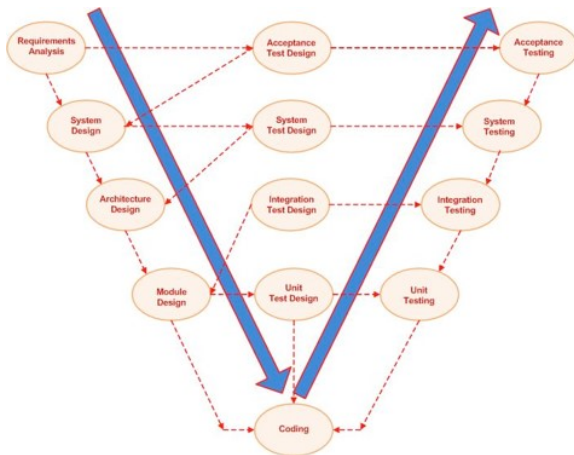
- Created Test Specialists
- “most errors are of an obvious nature that can be easily spotted by visual inspection” - code review by a second person is part of his “testing”
- “Test every logic path through the code”

Revision Step 5 - Involve the Customer

- “What a software design is going to do is subject to wide interpretation even after previous agreement.”
- And the result is:



V-Model



Essentially using test design as a review of early documentation

Boehm's Spiral

