CS 430 - Lecture 24 - Review and Wrap-Up

Collin Roberts

December 6, 2023

Outline

1. Historic / Economic Aspects
2. Maintenance (esp Post-Delivery)
3. Why There Is No Phase for
    1. Planning
    2. Testing
    3. Documentation
4. The OO Paradigm

1. Change is Inevitable
2. Iteration and Incrementation (which drives the remainder of the items in the list)
3. Other Life-Cycle Models
    1. Code-And-Fix
        1. This was the prevailing model pre-Waterfall / Classical.
        2. Under this model there was no change management at all!
    2. Waterfall / Classical
    3. Rapid Prototyping
    4. Open Source
    5. Agile Processes
    6. Synchronize and Stabilize
    7. Spiral
4. No One Life-Cycle Model dictated by SW-CMM

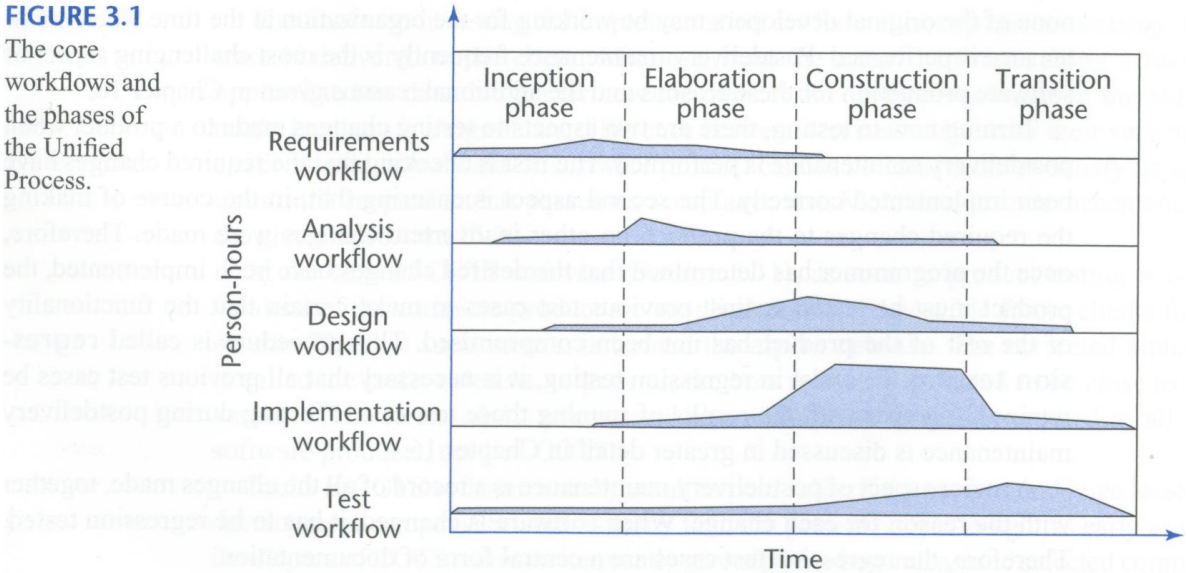

FIGURE 3.1
The core workflows and the phases of the Unified Process.

1. Postdelivery Maintenance
2. One- and Two-Dimensional Life-Cycle Models
3. Capability Maturity Models (SW-CMM specifically)

1. Democratic
2. Classical Chief Programmer
3. Modified Chief Programmer
4. Teams for Life-Cycle Models
    1. Synchronize and Stabilize
    2. Agile Processes
    3. Open Source
5. No One Team Organization dictated by P-CMM

1. Stepwise Refinement
2. Cost-Benefit Analysis
3. Divide and Conquer
4. Separation of Concerns
5. S/W Metrics
6. CASE tools
7. Version/Configuration Control

1. Quality Issues
   1. SQA
   2. Managerial Independence
2. Non-Execution-Based Testing (Reviews)
   1. Walkthroughs
   2. Inspections
3. Execution-Based Testing
   1. Best Practice: Determine expected results **before** you execute your first test.

4. What to Test
    1. Utility
    2. Reliability
    3. Robustness
    4. Performance
    5. Correctness
5. Testing versus Correctness Proofs
    1. There will be **no** correctness-proving on the final exam.
    2. When correctness-proving can be justified is fair game for the final exam.
6. Who Should Perform Execution-Based Testing? Answer: SQA!

N.B. Use our definitions from the Lecture Notes, NOT the text definitions here

1. Cohesion
2. Coupling
3. Encapsulation
4. Abstract Data Types
5. Information Hiding
6. Objects
7. Inheritance, Polymorphism, Dynamic Binding

1. Reusability
   1. Impediments to Reusability
   2. Objects and Reusability
   3. Types of Re-use
      1. Library (toolkit)
      2. Application Framework
      3. Design Patterns
2. Portability
   1. Impediments to Portability
      1. Hardware Incompatibilities
      2. Operating System Incompatibilities
      3. Numerical System Incompatibilities
      4. Compiler Incompatibilities
   2. Objects and Portability

1. Estimation
    1. Metrics for Size of a S/W product - Function Points
    2. Estimating Duration - Intermediate COCOMO
2. Project Management
    1. Testing
    2. Training
    3. Documentation
    4. CASE Tools
    5. Testing the SPMP

Student Course Perception Surveys

- Please fill out your course evaluations at `http://perceptions.uwaterloo.ca`
- All the best on your final exams before ours!