ITEC420: Software Engineering
Lecture 8: Case study: IBM tools, GIT and Final Exam

Box Leangsuksun
SWECO Endowed Professor,
Computer Science
*Louisiana Tech University*
box@latech.edu

CTO, PB Tech International Inc.
naibox@gmail.com
Software Engineering talk

- Title: IBM Software Engineering tools
- By Abena Primo, PhD Student, Louisiana Tech
- Time & Place: 5:30pm, Jan 21 2014, Bogard Hall
Additional Talk (optional)

- GITHUB is a web-based hosting service for software development projects that use the Git revision control system.

![GitHub Homepage](https://github.com)
The Retrospect: Class Review
Product Life cycle (RUP)
View of Software Engineering

User Needs/Inputs – Specification or Requirement

Technical: Design/Analysis Implementation

Project/Product Management:

Manufacturing/Quality Control

Marketing

Sale
Requirements and Specifications

- User Inputs/expectations are translated into agreement documents among users/customers and various stakeholders in SE lifecycle

- Can be legal documents between client and supplier

- How do we know whether the software product will meet the expectation?
Requirements and Specifications

• Functional Requirements
  – Tangible Needs
  – E.g. your order processing system, online store with shopping cart.

• Non-Functional Requirements
  – Performance (how well your system can perform, # transaction)
  – Reliability (how long your system can run w/o failure or what is uptime?)

• How do we know whether the software product will meet the expectation?
Capture requirement

- Reach agreement on system context
  - provided by customers
  - Vision statement (e.g. from marketing/product team)
  - Survey or research
- Communication.
- Articulation.
- Clarity.

- Come up with Abstractions of a given problem domain

- Arrive at actions representing/involving the abstractions (USE-CASES)
Use case process & notation

• Identify actors
• Brain-storm actions that will lead to features/promises to customers
• Refine use-cases and add exception cases
• eg. A doctor clinic
Recap: The Analysis Workflow
Purpose of Analysis & Design

• To transform the requirements into a design of the system to-be.

• To evolve a robust architecture for the system.

• An Architecture document & later design documents (for each subsystem, modules)

• To adapt the design to match the implementation environment, designing it for performance.
Analysis Artifacts (UML)

Robustness Diagram:
Usually contains attributes, not operations.

- Boundary Classes
- Control Classes
- Entity Classes
Example of Robustness
Diagram: Login usecase

Authenticated

User: John Doe
Pin: 1234
Account number: 111111111
Account value: $25,678.56
Address: PO box 7543
Ruston, La 71270

Validation

ATM Keypad

Actor

Account
Recap: Design Workflow
GOAL OF DESIGN WORKFLOW

EXPANSIONS OF THE ANALYSIS MODEL

• MOSTLY PHYSICAL

• MORE DETAILED

• SHOWS THE DECISIONS

• Capture how internal working of the system at the object-level
DESIGN MODEL

• Class Design
  – Class Diagram

• Use Case Realization-Design
  – sequence diagram or
  – collaboration diagram
  – Statechart diagram
  – Activity diagram
Example: class diagram

Superclass
(parent)

GroundVehicle
- weight
- licenseNumber
- register()

Subclasses

Car
- size

Truck
- tonnage
- getTax()

Person
- owner

Generalization

Trailer
Sequence DIAGRAM (in design workflow)

Actor 1

- login()
- enter user and password()
- validate user login()

Home page
- display()

Login page
- display()

Account
Sample of Deployment Diagram

This diagram is excerpted from intro to UML 2 deployment diagram, http://www.agilemodeling.com/artifacts/deploymentDiagram.htm
Implementation Work Flow
Activities in Implementation workflow

• Fundamental goal is to build a working version of system.
• Implement the Design in terms of components.
  – Write code with controlled quality
• Plan System Integrations
• Unit test
• Integration
The Test Workflow
Main Goals

• To ensure that the system offers a high degree of quality before it’s delivered to customers
• Result: Test Model
• Use-case oriented testing
• Quality Gate
  – Entrance Criteria
  – Exit Criteria
Activities & Artifacts

• Test Case
  ➢ Black-box testing-use cases
  ➢ White-box testing-use case realizations-design
  ➢ Integration testing-system level

• Test Procedure
  How to perform

• Test Evaluation & Acceptance Criteria
Sample of Exit Criteria for system test

- There must be 95% SVpass for a total test cases in the current release.
- No severity 1 opened change requests
- All code must be ITpassed in the change control
- Mean Time To Failure must be greater than 2 days (non-functional requirements)
Supporting Workflows in Context
Process vs Supporting Workflow

- Process Workflow
- Development activities
- Stronger emphasis in specific phases

- Supporting workflow
- Management and infrastructure activities
- Equal emphasis in all phases
Supporting Workflows

- Environment Management
- Change & Configuration Management
- Project & Risk Management.