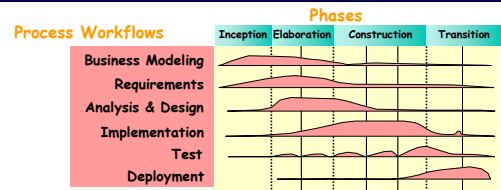


Elaboration

Rational Unified Process

(a form of controlled iteration)



where are you in the arcade game?

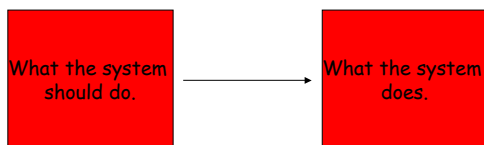
Essential Processes

- Requirements
 - Design
 - Implementation
 - Test
- } Elaboration

Iterative Development

- Phase i
 - Requirements:
 - What are you going to do?
 - How are you going to test it?
 - Design: How will you do it?
 - Implementation: Do it!
 - Test: Does it work?
 - Transition to phase i+1:
 - Integrate results into final project
 - Test integration

Requirements in Iterative Development



Requirements evolve over the course of the project.

Types of Requirements: FURPS+

- Functional: features, capabilities
- Usability: human factors, help, documentation
- Reliability: frequency of failure, recoverability, predictability
- Performance: response times, throughput, accuracy, availability, resource usage
- Supportability: adaptability, maintainability, configurability

Use-Case Analysis

- What is it?
 - An informal, user-friendly, technique useful for *functional* requirements analysis and specification
- From where did it come?
 - Ivar Jacobson, a Swedish software engineer at Ericsson, now with Rational, in a *method* called OOSE (Object-Oriented Software Engineering). Originally called "Usage cases"
- Now "part of" UML

Casual Use-Case

- Order from catalog
 - **Description:** Customer calls to order items from the catalog. The sales rep. identifies the item numbers, verifies that the items are in stock, and confirms the order with the customer, giving him the order number. The sales rep. then forwards the order to the Shipping dept.

Definition of "Use Case"

- "The specification of sequences of actions that a system, subsystem, or class can perform by interacting with outside actors"

(*UML Reference Manual*, Rumbaugh, Jacobson, and Booch).

Purpose of a "Use Case"

- "to define a piece of behavior of a [system or subsystem or class] ***without revealing the internal structure*** of the [system]"

(UML Reference Manual, Rumbaugh, Jacobson, and Booch).

Characteristics of Use-Case Analysis

- **Use-cases:** The specific ways in which the system is used.
- Each use-case expresses a "complete thought" or end-to-end transaction.
- A use case is a "black-box" specification; it does not deal with internal structure.
- A use case case may address high-level or low-level behavior.

The "who" in Use-Case Analysis

- **Actors:** anyone or anything with behavior in the use case.
- **Initiator:** the actor who/which initiates an interaction with the system to achieve a goal.
- **Stakeholder:** someone or something with a vested in the behavior of the system.

Actors

- Actors are characterized by *role* played in use case.
- One person can play different roles.
- One role can be filled (at different times) by several different persons or groups.
- An actor need not be a living thing; it could be another subsystem.

Casual Use-Case

- Order from catalog
 - **Actors:** customer, sales rep., shipping dept.
 - **Initiator:** customer
 - **Description:** Customer calls to order items from the catalog. The sales rep. identifies the item numbers, verifies that the items are in stock, and confirms the order with the customer, giving him the order number. The sales rep. then forwards the order to the Shipping dept.

Description format: flow-of-events

- 1 Customer calls to order from catalog.
- 2 Sales representative identifies item numbers.
- 3 Sales representative verifies stock.
- 4 Sales representative confirms order.
- 5 Sales representative gives order number to Customer.
- 6 Sales representative passes order to Shipping.

Flow-of-event *iteration*

- **For each** item to be ordered:
 - Sales representative checks catalog number.
 - Sales representative verifies stock.
 - Sales representative records item.

Flow of Events *extensions*

Main success scenario

- 1 Customer calls to order from catalog.
- 2 Sales representative identifies item numbers.
- 3 Sales representative verifies stock.
- 4 Sales representative confirms order.
- 5 Sales representative gives order number to Customer.
- 6 Sales representative passes order to Shipping.

Extensions

- 2a Customer's catalog is outdated and item number is invalid. Sales representative tells client item is no longer available.

Questions to ask

- **Scope:** What is the system under discussion (SuD)?
- **Actors:** Who is involved?
- **Initiator:** Who has the goal?
- **Level:** How high or low-level is that goal?

Pink Monster demo

Identify use cases

- Play game
- Move pink monster
- Change fruit
- Launch fruit
- End game

Play game

- Scope: What is really the system under discussion (SuD)?
- Actors: Who is involved?
- Initiator: Who has the goal?
- Level: How high or low-level is that goal?

Play game

- Description: Player launches game. Intro message asks player if she needs instructions. Game screen appears. Player choose fruit. Player positions pink monster. Player launches fruit. Score is updated. Process repeats. Player closes window to end game.

Play game

1. Player launches game.
2. Message box asks if player wants instructions.
3. Player says no.
 - 3a. Player says yes.*
4. Player positions pink monster.*
5. Player chooses fruit.*
6. Player launches fruit.*
7. Fruit misses target.
 - 7a. Fruit hits CS cart. *
 - 7b. Fruit hits other target*
8. Player repeats 4-7.
9. Player closes window to end game.

* Details are provided in separate use case.

Identify use cases

- Play game
- Provide game instructions
- Move pink monster
- Change fruit
- Launch fruit & miss
- Launch fruit & hit CS cart
- Launch fruit & hit other target
- End game

Move Pink Monster

1. Player uses left and right arrows to move pink monster.

Note: we are specifying the interface!

Scoping the project

- Identify as many use cases as you can
- Elaborate on high-level use cases
- Expand your use case list
- Identify use cases that specify "core" functionality
- Elaborate on those use cases
- Iterate until you converge to a small collection of use cases that capture the essential functions

Iterate

- Get customer feedback on your use cases.
- Repeat until you and the customer agree on the "core" system.

Pink Monster "core"

- Play game
- Move pink monster
- Launch fruit & miss target
- Launch fruit & hit target
- End game

Artifacts

- Fully elaborated use cases that define the core game.
- Additional use cases that define future enhancements.
- Major risks
- Other risks

Addressing Risks

- Brainstorm with your teammates on ways to eliminate or reduce each risk.
- Develop a concrete plan with measurable outcomes and specific artifacts.
- Assign tasks to group members.
- Establish deadlines.
- Set the next meeting time.
- Carry out your assignment.
- Assess your results.
- Meet with your group and present your results.
- Reassess risks.

Example

- Risk: POP???
- Plan: Complete the lab exercise Z has prepared to help acquaint us with POP.
- Who should do it: everyone on team
- When should it be done by: noon on Saturday
- When will we meet: noon on Saturday
- At the meeting: Compare results. Is there anything you couldn't figure out? If so, what are you going to do about it?