

---

# Organizing Software Development

Advanced Programming  
Barbara Russo

---

---

---

---

---

---

---

---

---

## Content

---

- Introduction
- The three beasts
- Software lifecycle
- Plan-driven and agile approaches

---

---

---

---

---

---

---

---

## Introduction

---

- Software development is a combination of **mutually dependent activities**
- Activities often **overlap in time and scope**
- **Good organization** is the key towards success

---

---

---

---

---

---

---

---

## The three beasts

- 3 major problems faced in developing software
  - Uncertainty
  - Irreversibility
  - Complexity
- Occurs before, during, and after development



Dante running off the three beasts

---

---

---

---

---

---

---

---

---

---

## Uncertainty

- Communication **issues** among different parties
  - For example: unexpected problems with libraries discovered by developers only at coding

---

---

---

---

---

---

---

---

---

---

## Irreversibility

- **Time** and most **resources** are not recoverable
  - For example: decision of allowing upgrades or levels of support provided cannot be later changed

---

---

---

---

---

---

---

---

---

---

## Complexity

- Lots of information to keep for a project
- **Coordination** between different groups (i.e. customers, managers, developers, etc.) is important
- + Uncertainty and irreversibility make coordination harder



28/02/14

CASE

7

---

---

---

---

---

---

---

---

## The 3 beasts in the software project

- A software project
  - Starts with vague idea of what to be developed
  - Ends with codes of various degrees of satisfaction to the original idea
- Uncertainty in idea
- Irreversibility in time
- Complexity in making idea into code

28/02/14

CASE

8

---

---

---

---

---

---

---

---

---

**STRUCTURING SOFTWARE DEVELOPMENT  
TO CONTROL FOR THE THREE BEASTS**

---

---

---

---

---

---

---

---

---

## Product lifecycle

- It is a **sequence of stages** for a product development
  - From conception to disposal/maintenance
- PL reduces
  - uncertainty of customers
  - complexity of development
- It comes from experience, it is not fixed for all the projects

28/02/14

CASE

10

---

---

---

---

---

---

---

---

## Example of product lifecycle

- Building a home
  - Analyze requirement of customers
  - Design home
  - Purchase land
  - Commission development to builder
  - Ensure the building meet standards set by different parties (personal and governmental)
  - Maintenance (housekeeping)



28/02/14

CASE

11

---

---

---

---

---

---

---

---

## Software lifecycle (1/2)

- Aiming for building quality software products in well organized manner
- Stages:
  - Are a common reference for all the parties
  - Are not fixed; vary with different projects
  - Are not necessarily implemented in linear fashion
  - Can be backtracked

28/02/14

CASE

12

---

---

---

---

---

---

---

---

## Stages

- **Requirement Elicitation and Analysis:** desiderata collected from customers and then formalized in a consistent and coherent way
- **Design:** a representation of entities and their relations and/or status (often graphical)
- **Implementation:** code developed
- **Testing:** system tested for correctness
- **Maintenance:** bug fixes, new features, new versions

28/02/14

CASE

13

---

---

---

---

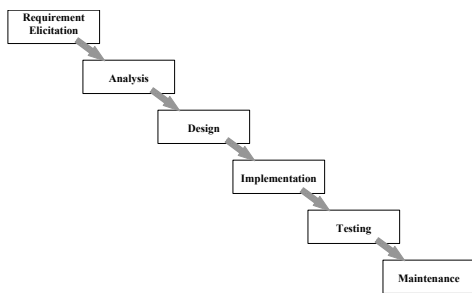
---

---

---

---

## Example of lifecycle



28/02/14

CASE

14

---

---

---

---

---

---

---

---

## The concept of stakeholder

- The stakeholders are key representatives of the groups who have vested **interest** in the system to be developed or have direct and indirect influence on the **requirements**

---

---

---

---

---

---

---

---

---

## Key stakeholders

- Key stakeholders
  - Customer
  - Manager
  - Developer
- Communication and mutual understanding of each other's role are important
  - Pitfall: uncertainty



---

---

---

---

---

---

---

---

## Customer

- “what to do”
- Desires and pays for the product
- Provides description of requirements
- Tests the end product



---

---

---

---

---

---

---

---

## Manager

- “when to finish and for how much”
- A link between customer and developer
- Negotiates with customer (timeline and price)
- Plans and oversees the project
- Set constraints of time and effort for developer



---

---

---

---

---

---

---

---

## Developer



- “how to do”
- Builds the product
- Aims for the satisfaction of customer

---

---

---

---

---

---

---

---

## Modern approaches to development

- They adopt the **divide et impera** paradigm:
  - Clearly **separate different stages**
  - Formalize the **case scenarios of by-products** to prevent misunderstanding
  - **Detect and correct defects** as early as possible
  - **Division of labour** among developers
  - **Combine** the implementation of **by-products** to get the original product

---

---

---

---

---

---

---

---

## Plan-driven approach

- Aim: reduce uncertainty and complexity
- Guiding development via long term plans to address uncertainty and manage complexity
- Strict end-to-start dependencies among stages
  - Irreversibility increases inevitably as project progresses

---

---

---

---

---

---

---

---

## Agile approach

- Original idea and plans changed progressively
- Changes in development is essential, not evil
- Accepts and addresses the 3 beasts rather than avoiding them
- It is up to developers to decide which approach is more suitable for the project

28/02/14

CASE

22

---

---

---

---

---

---

---

---

## The Agile Manifesto

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

- <http://agilemanifesto.org/>

---

---

---

---

---

---

---

---

## Example of Agile Methods

- Example of agile methods
  - eXtreme Programming
    - <http://www.extremeprogramming.org/>
  - SCRUM
    - <http://www.scrumalliance.org/>
  - Test Driven Development
    - Beck, K. Test-Driven Development by Example, Addison Wesley, 2003

---

---

---

---

---

---

---

---



## Reference book

- Pekka Abrahamsson, Agile Software Development Methods: Review and Analysis

---

---

---

---

---

---

---

---

---

## Lessons learned

- There is a need to engineering software
- Three beasts hamper the success of a software project
- To control them
  - the concept of software life cycle has been introduced – divide et impera
  - The concept of stakeholder has been emphasized
  - Methods of development have been outlined
- We will use an agile approach

---

28/02/14

CASE

26

---

---

---

---

---

---

---

---

## Next lesson

- The Object Oriented Paradigm

---

28/02/14

CASE

27

---

---

---

---

---

---

---

---