Software Engineering

Architecture, Design and Patterns

Wladmir Cardoso Brandão
www.wladmirbrandao.com

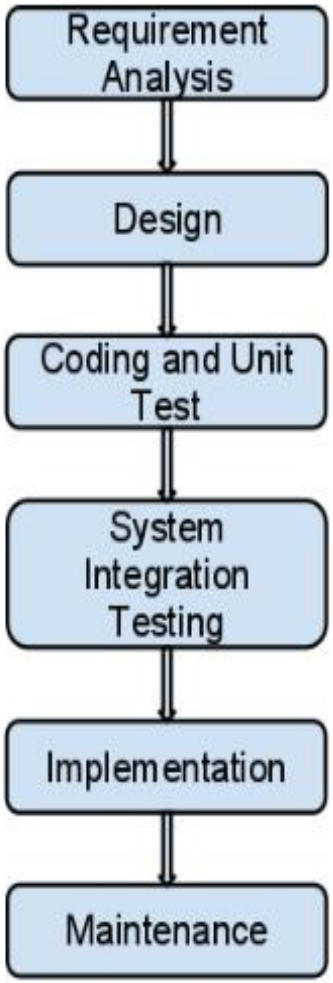
Department of Computer Science (DCC)
Institute of Exact Sciences and Informatics (ICEI)
Pontifical Catholic University of Minas Gerais (PUC Minas)

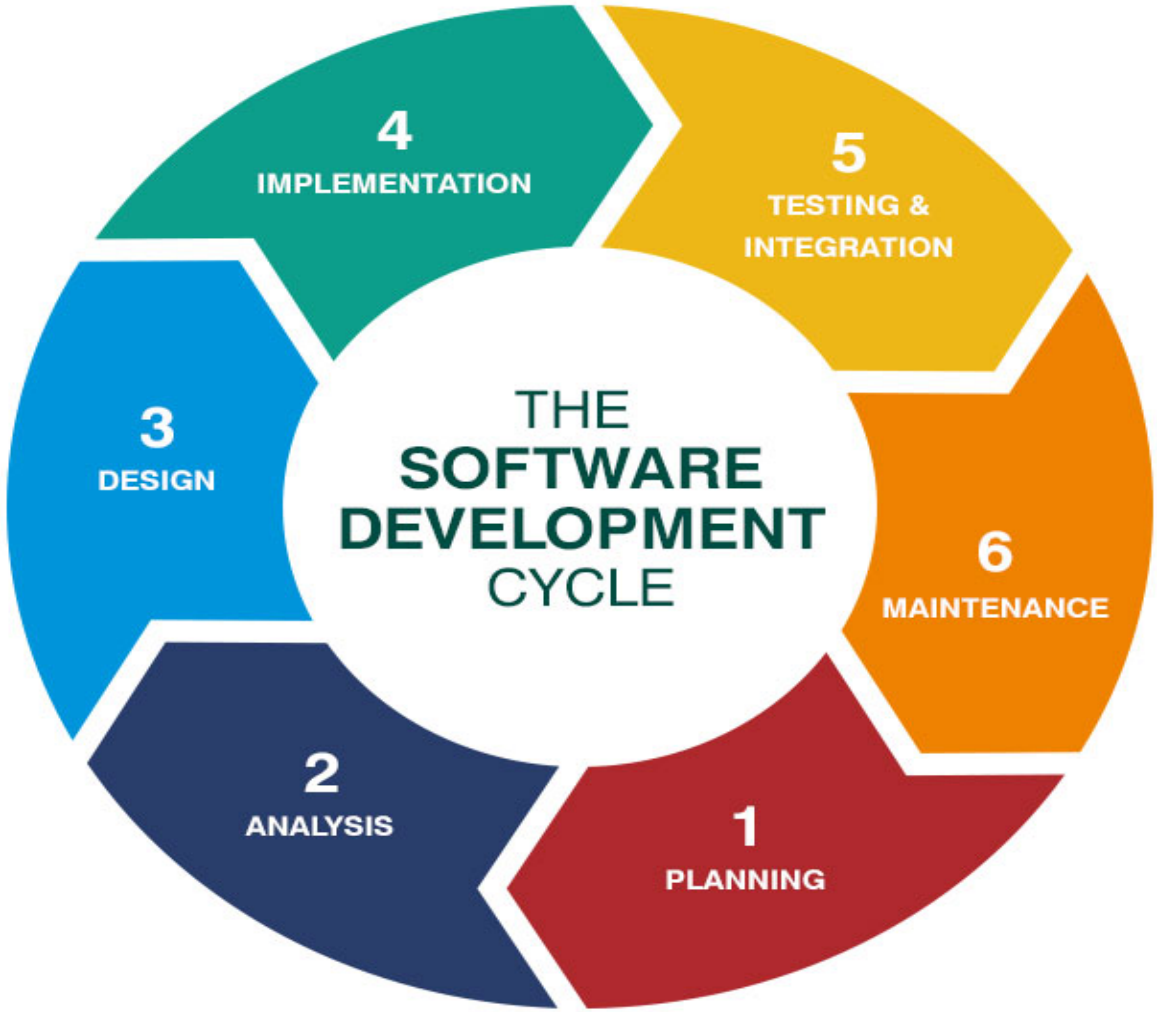
The author thanks the support to:
Students who willingly help me in this task

Chapter 03

Software Development Process

Software Development Life Cycle





Software Development Life Cycle

Initiation

Begins when a sponsor identifies a need or an opportunity. Concept Proposal is created

System Concept Development

Defines the scope or boundary of the concepts. Includes Systems Boundary Document. Cost Benefit Analysis. Risk Management Plan and Feasibility Study.

Planning

Develops a Project Management Plan and other planning documents. Provides the basis for acquiring the resources needed to achieve a soulution.

Requirements Analysis

Analyses user needs and develops user requirements. Create a detailed Functional Requirements Document.

Design

Transforms detailed requirements into complete, detailed Systems Design Document Focuses on how to deliver the required functionality

Development

Converts a design into a complete information system Includes acquiring and installing systems environment; creating and testing databases preparing test case procedures; preparing test files, coding, compiling, refining programs; performing test readiness review and procurement activities.

Integration and Test

Demonstrates that developed system conforms to requirements as specified in the Functional Requirements Document. Conducted by Quality Assurance staff and users. Produces Test Analysis Reports.

Implementation

Includes implementation preparation, implementation of the system into a production environment, and resolution of problems identified in the Integration and Test Phases

Operations & Maintenance

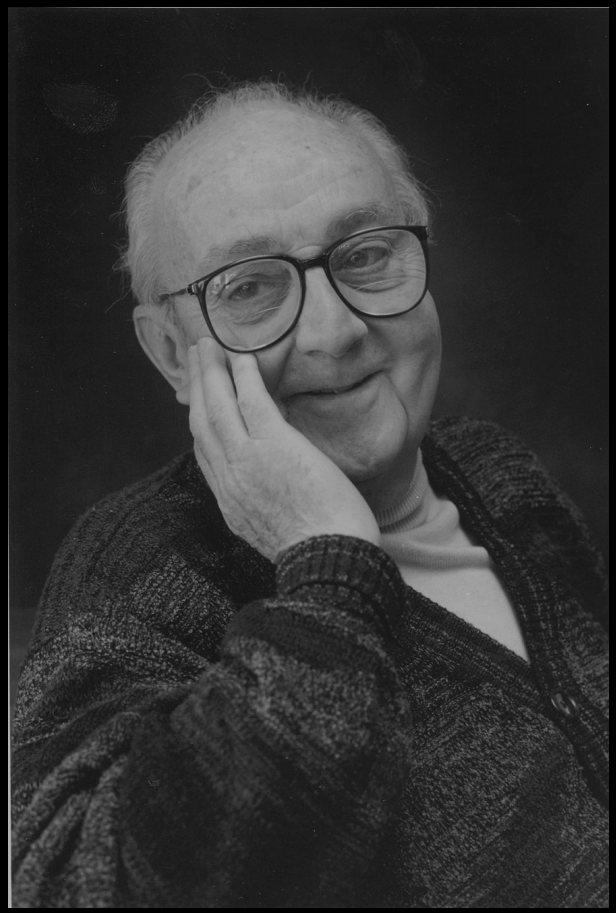
Describes tasks to operate and maintain information systems in a production environment. includes Post-Implementation and In-Process Reviews.

Disposition

Describes end-of-system activities, emphasis is given to proper preparation of data.

Modeling

Model	
Analytic	Design
Describe a problem	Describe a solution
Logical	Physical
Support many designs	Single implementation
Informal and generic	Formal and detailed

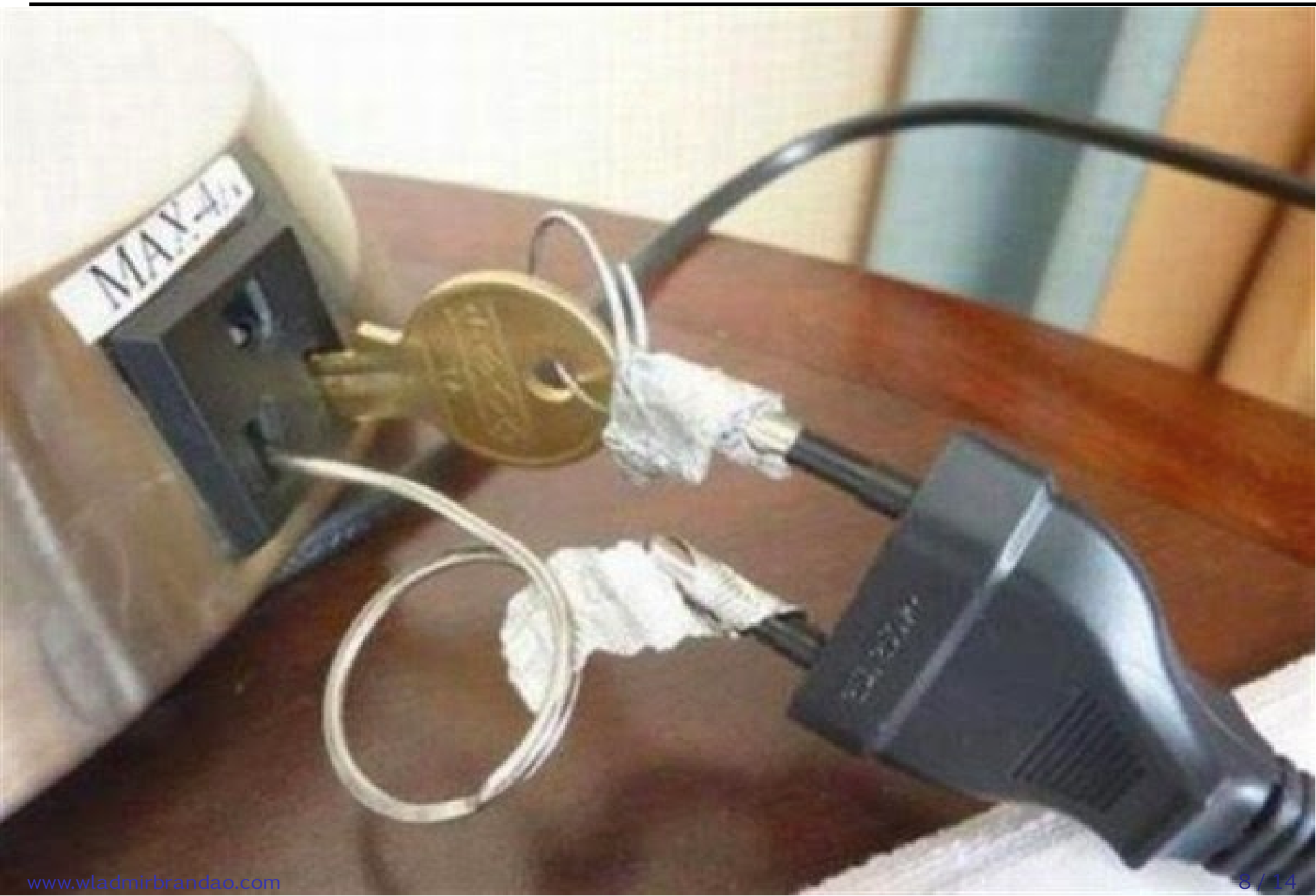


"All models are wrong,
but some are useful"

George Box, 1978

Why Modeling?

Why Modeling?



Models and Modeling

- Models → **representations** for entities
- To put one thing in the place of another, e.g., physical reductions, graphics, text and art
- Provide simplification, concept test
  - Complexity reduction → capture only fundamental aspects of objects
  - Conceptual convergence → make the different ideas about the object convergent, facilitating the communication process
  - Prediction and simulation → allow an approximation of the actual object, often impossible to perform in the real world
- Modeling → action of building models

Analysis x Design

- SE evolves:
  - Analysis → decomposing large problems into smaller, understandable pieces (abstraction)
  - Synthesis → building large software systems from smaller building blocks (composition)
- For solving problems, we apply:
  - Techniques → a formal "recipe" for accomplishing a goal, typically independent of the tools, e.g., automated builds, configuration management, software testing
  - Tools → an instrument or automated system for accomplishing something in a better (efficient, accurate, faster) way, e.g., maven, git, jenkins

Analysis x Design

- For solving problems, we apply:
  - Procedures → a combination of tools and techniques to produce a particular product
  - Paradigms → a particular philosophy or approach for building a product
- OO x Structured → Both approaches use similar things (requirements, design, code, editors, compilers), but think about the problem in different ways

References I

- Sommerville, Ian
  Software Engineering.
  10ed, 2016.
- Larman, Graig
  Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development.
  3ed, 2004.
- Pressman, Roger; Maxim, Bruce
  Software Engineering: A Practitioner's Approach.
  8ed, 2014.
- Booch, Grady; Rumbaugh, James; Jacobson, Ivar
  The Unified Modeling Language User Guide.
  2ed, 2005.

References II

Gamma, Erich; Helm, Richard; Johnson, Ralph; Vlissides, John

Design Patterns: Elements of Reusable Object-Oriented Software.

1ed, 1994.

Buschmann, Frank; Rohnert, Hans; Stal, Michael; Sommerlad, Peter; Meunier, Regine

Pattern-Oriented Software Architecture, A System of Patterns.

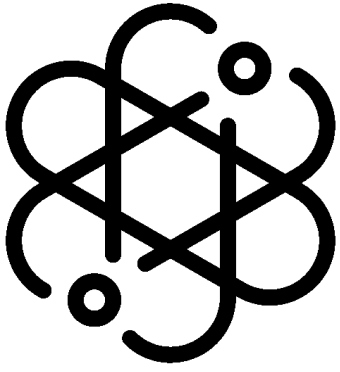
1ed, 1996.

Frederick P. Brooks

The Mythical Man-Month.

Proceedings of the IFIP Tenth World Computing Conference, 1986.

THANK YOU



IRIS
RESEARCH

Wladmir Cardoso Brandão
www.wladmirbrandao.com
wladmir@pucminas.br

"Science is more than a body of knowledge. It is a way of thinking."
Carl Sagan