U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Software Engineering

T2 - Software Processes

3MIEIC04_B

(20th October 2017)

Bárbara Silva **up201505628**@fe.up.pt

Julieta Frade **up201506530**@fe.up.pt

Ventura Pereira **up201404690**@fe.up.pt

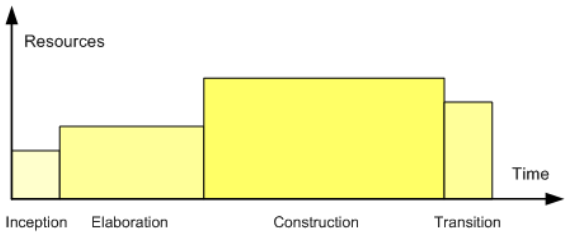
Índice

Unified Process

The **Unified Software Development Process** or Unified Process is a popular iterative and incremental software development process framework. The best-known and extensively documented refinement of the Unified Process is the Rational Unified Process (RUP). Other examples are OpenUP and Agile Unified Process. Unified process can be applied to different software systems with different levels of technical and managerial complexity across various domains and organizational cultures.

The first book to describe the process was titled The Unified Software Development Process and published in **1999** by **Ivar Jacobson, Grady Booch and James Rumbaugh.**

The Four Phases



The life of a software system can be represented as a series of cycles. A cycle ends with the release of a version of the system to customers.

Within the Unified Process, each cycle contains four phases. **A phase is simply the span of time between two major milestones, points at which managers make important decisions** about whether to proceed with development and, if so, what's required concerning project scope, budget, and schedule.

1. Inception

Primary goal: establish the case for the viability of the proposed system.

Tasks:

- Defining the scope of the system (that is, what's in and what's out).
- Outlining a candidate architecture, which is made up of initial versions of six different models.
- Identifying critical risks and determining when and how the project will address them.
- Starting to make the business case that the project is worth doing, based on initial estimates of cost, effort, schedule, and product quality.

Major milestone: Life-Cycle Objectives.

The indications that the project has reached this milestone:

- The major stakeholders agree on the scope of the proposed system.
- The candidate architecture clearly addresses a set of critical high-level requirements.
- The business case for the project is strong enough to justify a green light for continued development.

2. Elaboration

Primary goal: establish the ability to build the new system given the financial constraints, schedule constraints, and other kinds of constraints that the development project faces.

Tasks:

- Capturing a healthy majority of the remaining functional requirements.
- Expanding the candidate architecture into a full architectural baseline, which is an internal release of the system focused on describing the architecture.
- Addressing significant risks on an ongoing basis.
- Finalizing the business case for the project and preparing a project plan that contains sufficient detail to guide the next phase of the project (Construction).

Major milestone: Life-Cycle Architecture.

The indications that the project has reached this milestone:

- Most of the functional requirements for the new system have been captured in the use case model.
- The architectural baseline is a small, skinny system that will serve as a solid foundation for ongoing development.
- The business case has received a green light, and the project team has an initial project plan that describes how the Construction phase will proceed.

3. Conception

Primary goal: build a system capable of operating successfully in beta customer environments.

Tasks: involve building the system iteratively and incrementally, making sure that the viability of the system is always evident in executable form.

Major milestone: Initial Operational Capability.

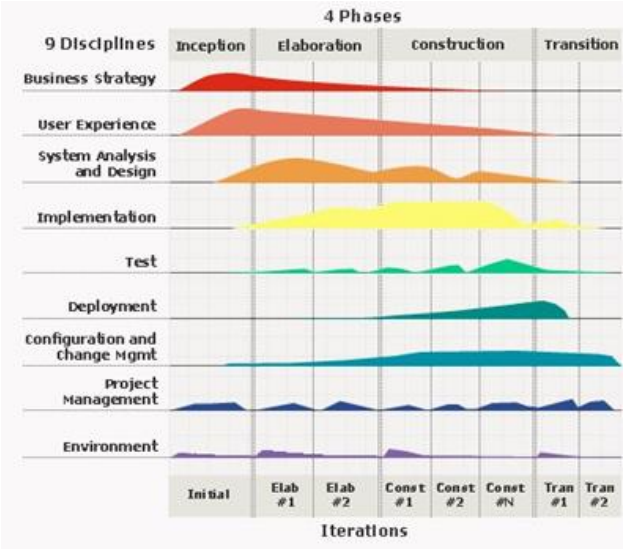
The indications that the project has reached this milestone: if a set of beta customers has a more or less fully operational system in their hands.

4. Transition

Primary goal: roll out the fully functional system to customers.

Tasks: correcting defects and modifying the system to correct previously unidentified problems.

The major milestone: Product Release.



Characteristics

Iterative and incremental

The Elaboration, Construction and Transition phases are divided into a series of timeboxed iterations. **Each iteration results in an increment, which is a release of the system that contains added or improved functionality compared with the previous release**.

Use-case driven

In the Unified Process, use-cases are **used to capture functional requirements and refine the content of the iterations**. Each iteration has a set of use cases or requirements scenarios throughout the implementation, testing, and development time.

Architecture-centric

Since no single model is sufficient to cover all aspects of a system, **the Unified Process supports multiple architectural models and views**. One of the most important deliverables of the process is the executable architecture baseline which is created during the Elaboration phase. This partial implementation of the system serves to validate the architecture and act as a foundation for remaining development.

Risk-focused

The Unified Process requires **the project team to focus on addressing the most critical risks early in the project life cycle**. The deliverables of each iteration, especially in the Elaboration phase, must be selected in order to ensure that the greatest risks are addressed first.

Spiral

The **spiral model** is a **risk-driven process model generator for software projects**. Based on the unique risk patterns of a given project, the spiral model guides a team to adopt elements of one or more process models, such as incremental, waterfall, or evolutionary prototyping. This model was first described by **Barry Boehm** in his **1986** paper "A Spiral Model of Software Development and Enhancement".

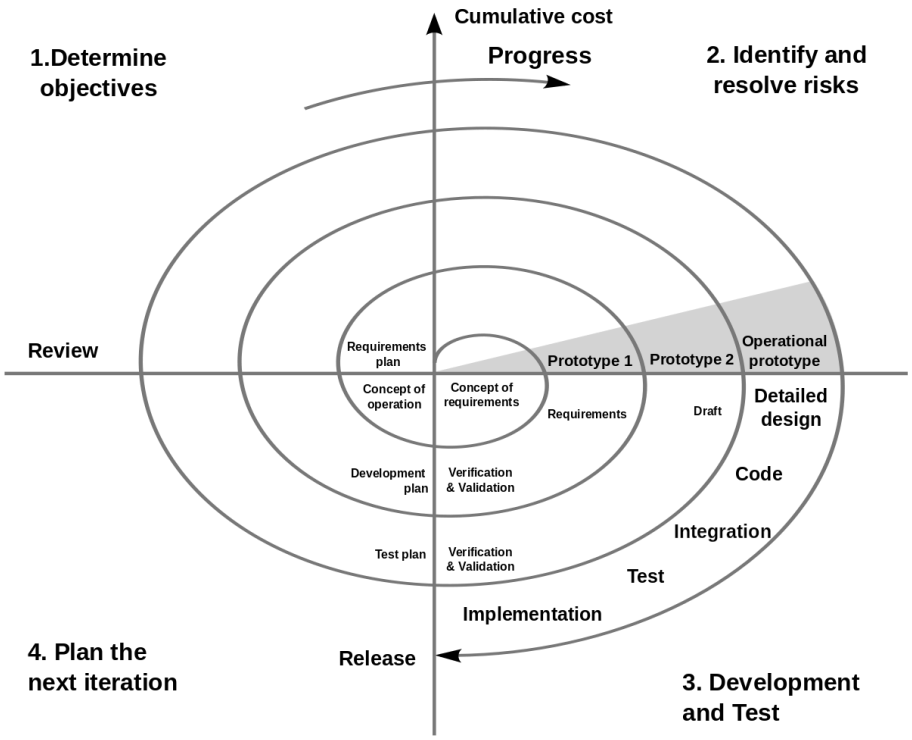


Figure 1: Spiral model (Boehm, 2000). A number of misconceptions stem from oversimplifications in this widely circulated diagram, with a few errors.

Boehm identified a **number of misconceptions arising from oversimplifications** in the original spiral model diagram, such as:

- The spiral is simply a sequence of waterfall increments.
- All project activities follow a single spiral sequence.
- Every activity in the diagram must be performed, and in the order shown.

Authentic applications of the spiral model are driven by cycles that always display **six characteristics** and **four phases**.

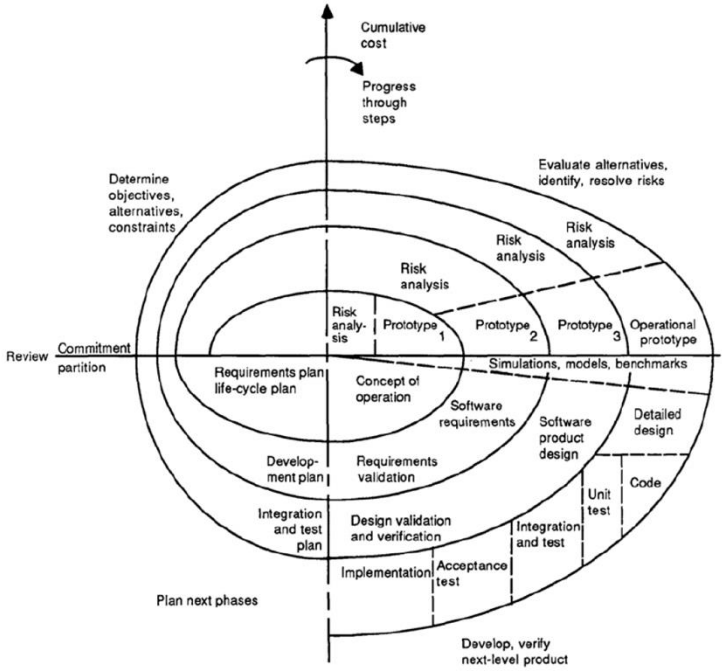


Figure 2: Spiral model of the software process.

The Six Invariants

- Define artefacts concurrently.
- Perform four basic activities in every cycle.
- Risk determines level of effort.
- Risk determines degree of details.
- Use anchor point milestones.
- Focus on the system and its life cycle.

As mentioned above, the spiral model has **four phases**. A software project repeatedly passes through these phases in iterations called Spirals.

- Phase 1: **Identification**
- Phase 2: **Risk Analysis**
- Phase 3: **Engineering and Evaluation**
- Phase 4: **Planning**

When to use Spiral method

- When costs and risk evaluation is important.
- For medium to high-risk projects.
- Long-term project commitment unwise because of potential changes to economic priorities.
- Users are unsure of their needs.
- Requirements are complex.
- New product line.
- Significant changes are expected (research and exploration).

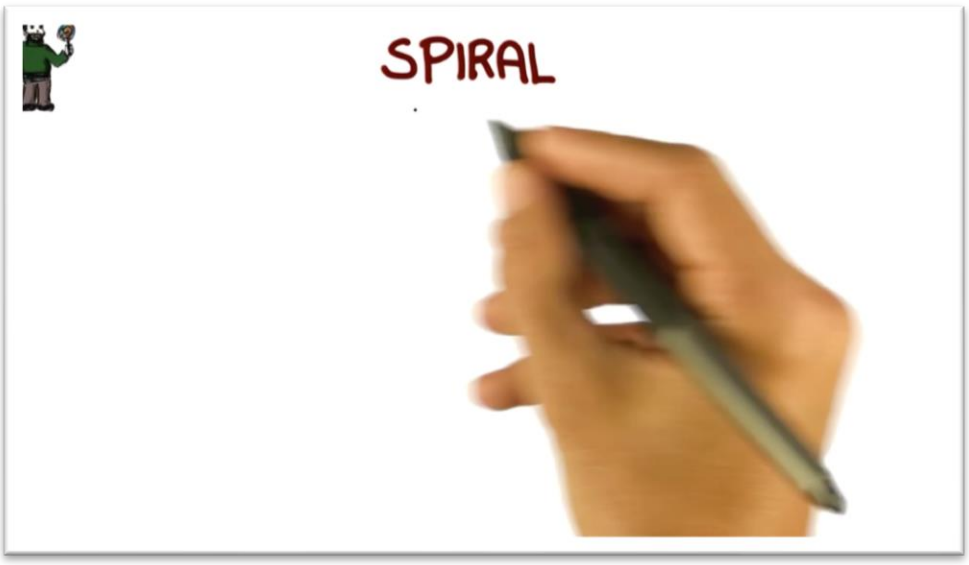
Advantages

- Changing requirements can be accommodated.
- Allows extensive use of prototypes.
- Requirements can be captured more accurately.
- Users see the system early.
- Development can be divided into smaller parts and the risky parts can be developed earlier which helps in better risk management.

Disadvantages

- Management is more complex.
- End of the project may not be known early.
- Not suitable for small or low risk projects and could be expensive for small projects.
- Process is complex.
- Spiral may go on indefinitely.
- Large number of intermediate stages requires excessive documentation.

Tutorial



Youtube video: https://youtu.be/mp22SDTnsQQ

Sources

Unified Process

- https://pt.slideshare.net/guy_davis/unified-process
- http://www.methodsandtools.com/archive/archive.php?id=32
- https://www.techopedia.com/definition/3885/unified-process-up
- https://pt.wikipedia.org/wiki/Processo_unificado
- http://www.informit.com/articles/article.aspx?p=24671&seqNum=7

Spiral

- https://en.wikipedia.org/wiki/Spiral_model
- https://www.tutorialspoint.com/sdlc/sdlc_spiral_model.htm
- http://istqbexamcertification.com/what-is-spiral-model-advantages-disadvantages-and-when-to-use-it/

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Software Engineering

Unified Process

3MIEIC04_B

(20th October 2017)

Bárbara Silva **up201505628**@fe.up.pt
Julieta Frade **up201506530**@fe.up.pt
Ventura Pereira **up201404690**@fe.up.pt

Unified Software Development Process

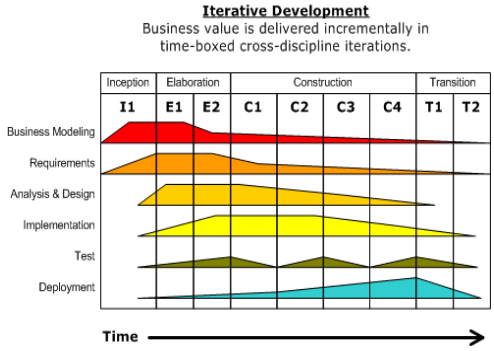
The Four Phases

1. Inception
2. Elaboration
3. Conception
4. Transition

Characteristics

What's a software phase?

- Series of cycles.
- Each cycle contains four phases.
- Span of time between two major milestones.



Inception

The first phase

Inception

Goal

Viability of the proposed system.

Tasks

- System's scope.
- Outlining a candidate architecture.
- Indentify risks.
- Start the business.

Inception

When can we move on?

Before moving on torwads the next step, the Major milestone must be achieved:

- Do the major stakeholders agree on the scope of the proposed system?
- Is a specific set of critical high-level requirements adressed by it's architeture?
- Is the business case enough for a green light?

Elaboration

The second phase

Elaboration

What do we want to achieve at this stage?

Build is the key word. As engineers, the ability to build the new system, given some constrains, such as the financial, schedule ones, must be established.

Elaboration

What exactly do we need to do?

- Understanding the majority of the remaining functional requirements.
- To expand the previous candidate architecture into a full architectural baseline. Make it an internal release of the sstem focused on describing the architecture.
- Adressing major risks on na ongoing basis.
- To finish the business case and prepare a project plan, detailed enough, to guide us through the next phase.

Elaboration

When can we move on?

The major milestone must be achieved.

- The user case model must capture most of the functional requirements.
- The architetural baseline must be a small system that will serve as a solid foundation.
- Business case has received a green light, there's a project plan for the next phase.

Conception

The third phase

Conception

What's the primary goal?

Create a system capable of operating in beta customer environments.

How?

Build the system iteratively and incrementally, making sure that the viability of the system is always evidente in executable form.

Conception

What must be achieved?

The initial operational Capability.

How do we know?

A set of beta customers has a more or less fully operational system in their hands.

Transition

The final phase

Transition

What's the purpose?

The roll out of the fully functional system to customers.

How do we do it?

Correct defects, modify the system to correct previously unidentified problems.

Transition

What comes next?

Product release!

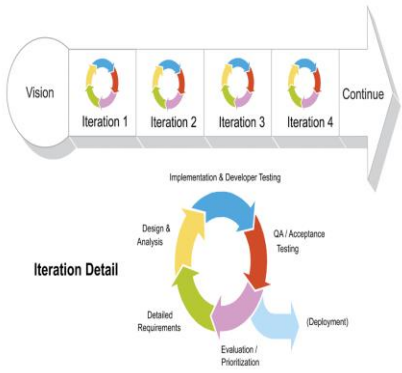
Characteristics

Iterative and Incremental

The last three phases that we've seen are divided into a series of timeboxed iterations.

Each iteration results in an increment.

It represents added or improved functionalities compared with the previous release.



Use-case Driven

- Used to capture functional requirements.
- Refine the content of the iterations.
- Set of use cases or requirements scenarios.

Architecture-centric

Since no single model is suficiente to cover all aspects of a system, the Unified Process supports multiple architectural models and views.

- Created during the Elaboration phase.
- Partial implementation of the system serves to validate the architecture and act as a Foundation for the rest of the development.

Risk-focused

- The project team must focus on adressing the most critical risks early in the project life cycle.
- Each iteration, especially in the elaboration phase, must be select to ensure the greatest risks are adressed first.