

Engenharia de Software

José Cascais Brás

Tópico 3



UNIVERSIDADE
LUSÓFONA

Engenharia de Software

Learning Objectives



**SOFTWARE PROCESS
MODELS**

PROCESS ACTIVITIES

**COPING WITH
CHANGE**

**PROCESS
IMPROVEMENT**

AGENDA

1

**THE SOFTWARE
PROCESS MODEL**

2

**SOFTWARE
PROCESS
MODELS**

3

**PROCESS
ACTIVITIES**

4

**COPING WITH
CHANGE**

5

**PROCESS
IMPROVEMENT**

6

KEY POINTS

THE SOFTWARE PROCESS

A structured set of activities required to develop a software system

Many different software processes but all involve:

- ❑ **Specification** –defining what the system should do;
- ❑ **Design and implementation** –defining the organization of the system and implementing the system;
- ❑ **Validation** –checking that it does what the customer wants;
- ❑ **Evolution** –changing the system in response to changing customer needs.

Conceito

→ **Processo de desenvolvimento de *software*** (*software development process*, ing. / *processo de desenvolvimento de software*, br.)

O processo de desenvolvimento de *software* consiste na definição sistemática de etapas distribuídas de comunicação, cooperação, execução e gestão tendo como objetivo construir, de forma eficiente e eficaz¹, um novo produto de *software* devidamente verificado e validado.

THE SOFTWARE PROCESS

A software process model is an abstract representation of a process. It presents a description of a process from some particular perspective.

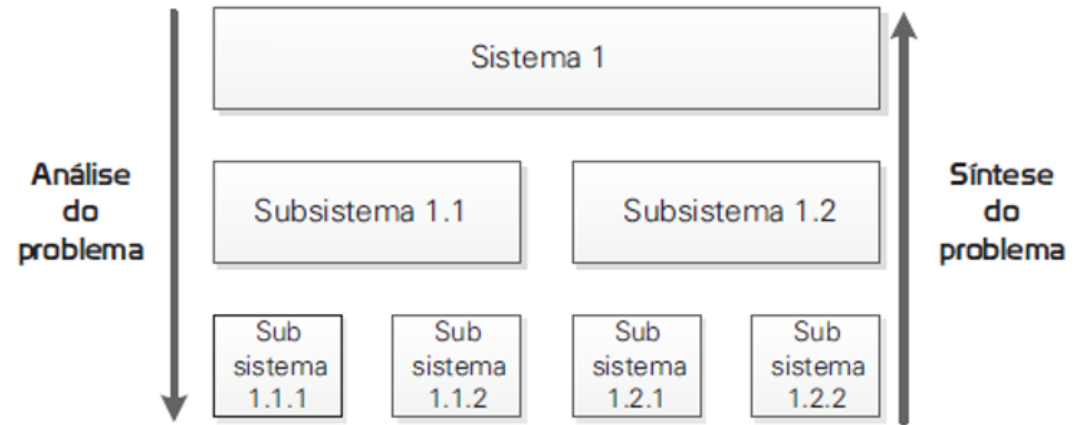


FIGURA 2.1 – DUAS SOLUÇÕES PARA LIDAR COM A COMPLEXIDADE

Fonte: Sérgio Guerreiro, *Introdução à Engenharia de Software*, FCA – Editora de Informática

SOFTWARE PROCESS DESCRIPTIONS

When we describe and discuss processes, we usually talk about the activities in these processes such as specifying a data model, designing a user interface, etc. and the ordering of these activities.

Process descriptions may also include:

PRODUCTS

Which are the outcomes of a process activity

ROLES

Which reflect the responsibilities of the people involved in the process.

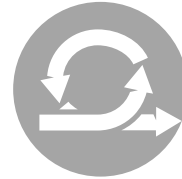
PRE and POST CONDITIONS

Which are statements that are true before and after a process activity has been enacted or a product produced.

PLAN DRIVEN AND AGILE PROCESSES



Plan driven processes are processes where all of the process activities are planned in advance and progress is measured against this plan.



In agile processes, planning is incremental and it is easier to change the process to reflect changing customer requirements.

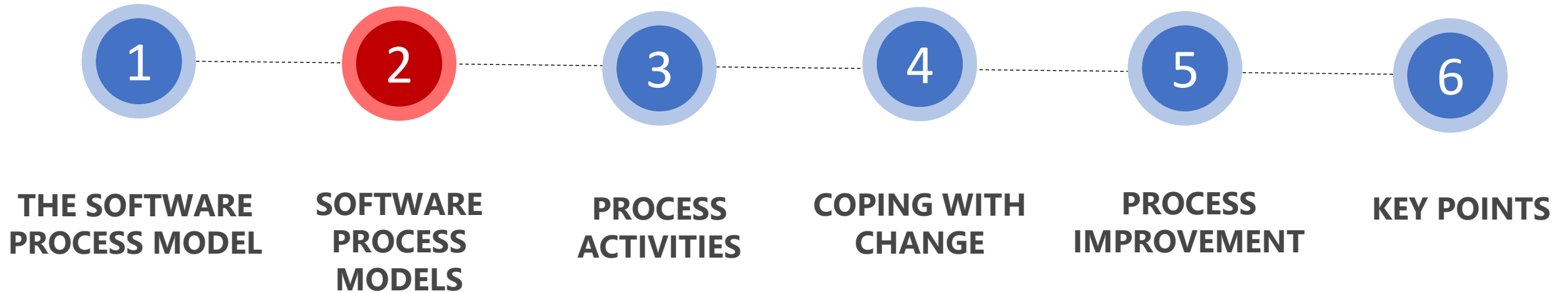


In practice, most practical processes include elements of both plan driven and agile approaches.



There are no right or wrong software processes.

AGENDA



SOFTWARE PROCESS MODELS



THE WATERFALL MODEL

Plan driven model. Separate and distinct phases of specification and development.



INCREMENTAL DEVELOPMENT

Specification, development and validation are interleaved. May be plan driven or agile.



INTEGRATION AND CONFIGURATION

The system is assembled from existing configurable components. May be plan driven or agile.

In practice, most large systems are developed using a process that incorporates elements from all of these models.

THE SOFTWARE PROCESS

DEFINITION: STAKEHOLDER vs USER

Conceito

→ **Interveniente** (*stakeholder*, ing. / *interveniente*, br.)

Um interveniente de um processo de desenvolvimento de *software* é uma ou mais pessoas que cumprem um papel ativo no processo de desenvolvimento e que permitem que as atividades de comunicação, cooperação, execução e gestão sejam executadas ao longo do tempo.

Fonte: Sérgio Guerreiro, *Introdução à Engenharia de Software*, FCA – Editora de Informática

Conceito

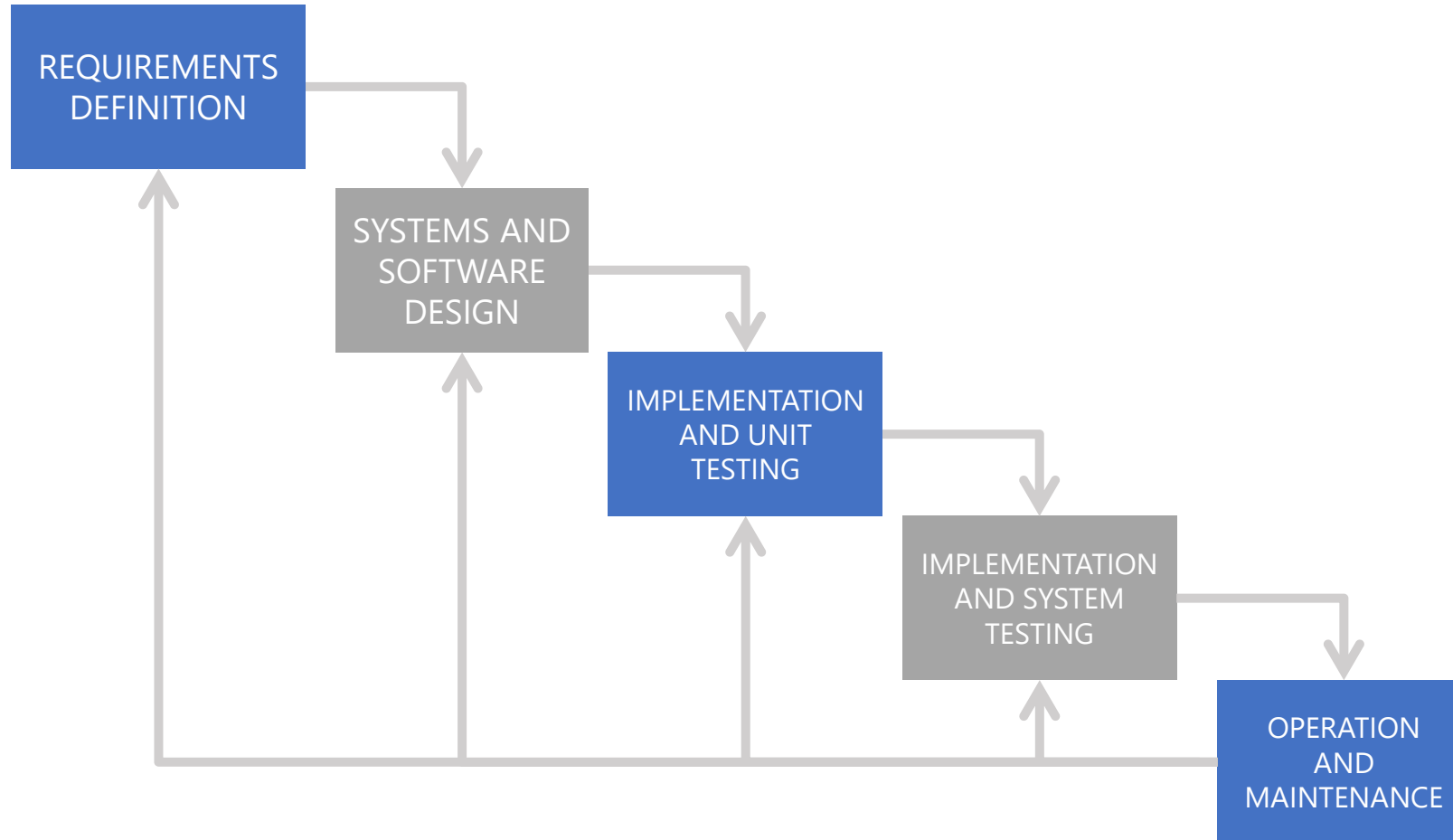
→ **Utilizador final** (*end user*, ing. / *usuário*, br.)

Os utilizadores finais utilizam as funcionalidades do produto de *software* para realizarem partes dos processos de negócio em que estão envolvidos. Os utilizadores finais necessitam de formação para utilizarem os produtos de *software* de forma correta.

Fonte: Sérgio Guerreiro, *Introdução à Engenharia de Software*, FCA – Editora de Informática

THE SOFTWARE PROCESS

WATERFALL SOFTWARE DEVELOPMENT PROCESS MODEL

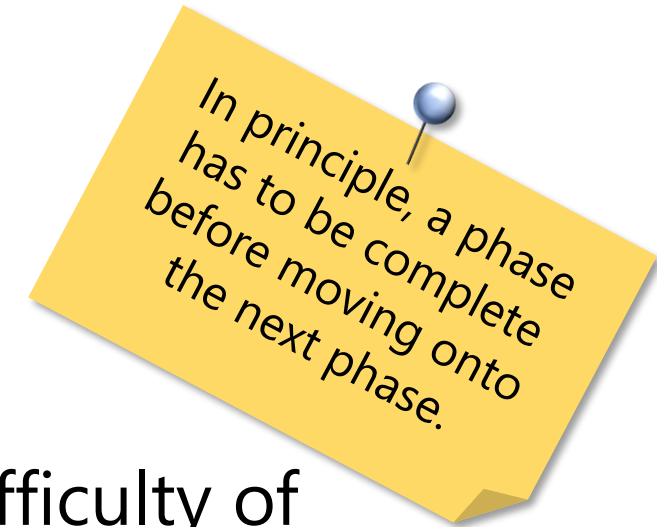


WATERFALL MODEL PHASES

❑ There are separate identified phases in the waterfall model:

- **Requirements** analysis and definition
- System and software **design**
- **Implementation** and **unit testing**
- **Integration** and **system testing**
- **Operation** and **maintenance**

❑ The main drawback of the waterfall model is the difficulty of accommodating change after the process is underway.



WATERFALL MODEL PROBLEMS

- ❑ Inflexible partitioning of the project into distinct stages makes it difficult to respond to changing customer requirements.
 - Therefore, this model is only appropriate when the requirements are well understood and changes will be fairly limited during the design process
 - Few business systems have stable requirements.
- ❑ The waterfall model is mostly used for large systems engineering projects where a system is developed at several sites.

In those circumstances, the **plan driven** nature of the waterfall model helps coordinate the work.

HW FAILURE PROB DURING ITS LIFE CYCLE

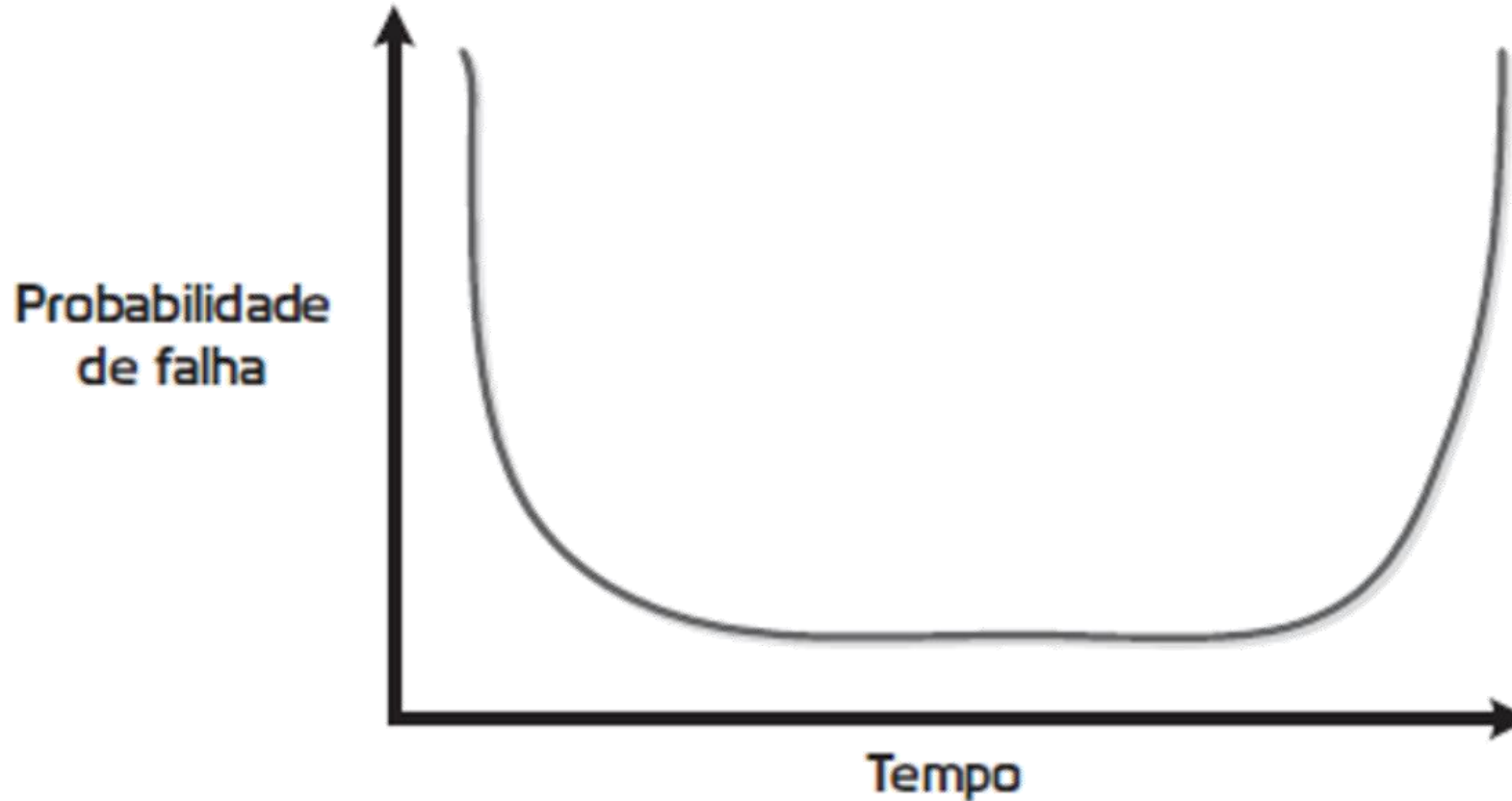


FIGURA 2.2 – EVOLUÇÃO DA PROBABILIDADE DE FALHA DO HARDWARE AO LONGO DO SEU TEMPO DE VIDA

Fonte: Sérgio Guerreiro, *Introdução à Engenharia de Software*, FCA – Editora de Informática

SW FAILURE PROB DURING ITS LIFE CYCLE

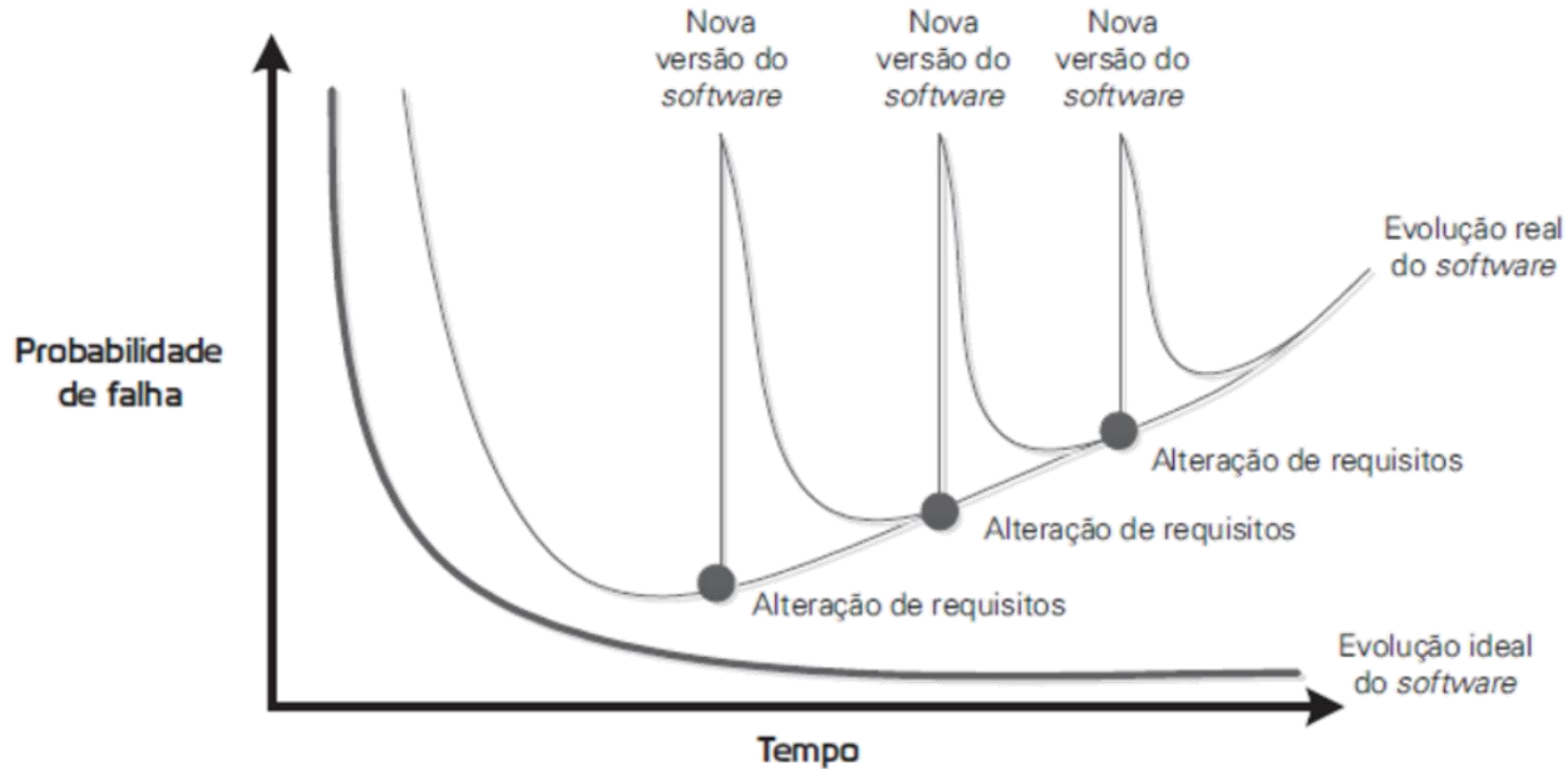


FIGURA 2.3 – EVOLUÇÃO DA PROBABILIDADE DE FALHA DO *SOFTWARE* AO LONGO DO SEU TEMPO DE VIDA

Fonte: Sérgio Guerreiro, *Introdução à Engenharia de Software*, FCA – Editora de Informática

PROTOTYPING SOFTWARE DEVELOPMENT PROCESS MODEL

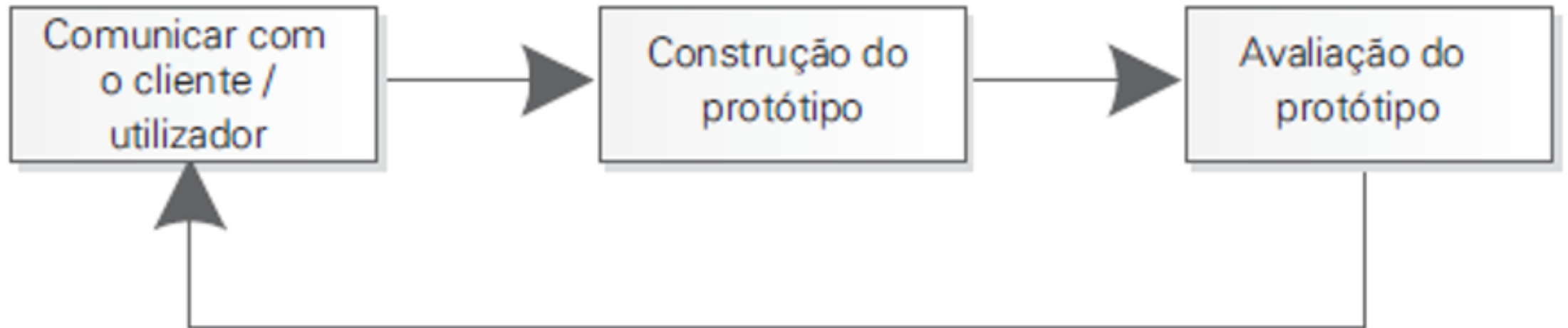
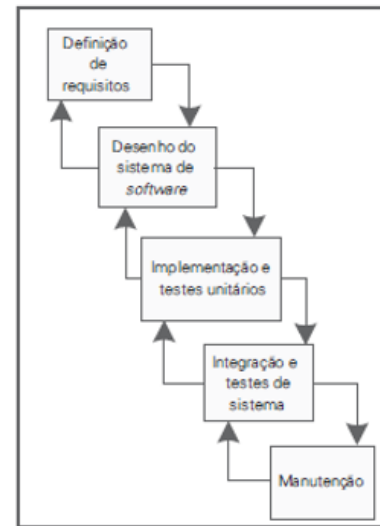


FIGURA 2.5 – PROCESSO DE DESENVOLVIMENTO POR PROTÓTIPOS

Fonte: Sérgio Guerreiro, *Introdução à Engenharia de Software*, FCA – Editora de Informática

RAD SOFTWARE DEVELOPMENT PROCESS MODEL



Múltiplas equipas de desenvolvimento, de pequena dimensão, a trabalharem em simultâneo



Tempo total reduzido →

FIGURA 2.6 – PROCESSO DE DESENVOLVIMENTO RÁPIDO DE APLICAÇÕES

Fonte: Sérgio Guerreiro, *Introdução à Engenharia de Software*, FCA – Editora de Informática

INCREMENTAL SOFTWARE DEVELOPMENT PROCESS MODEL

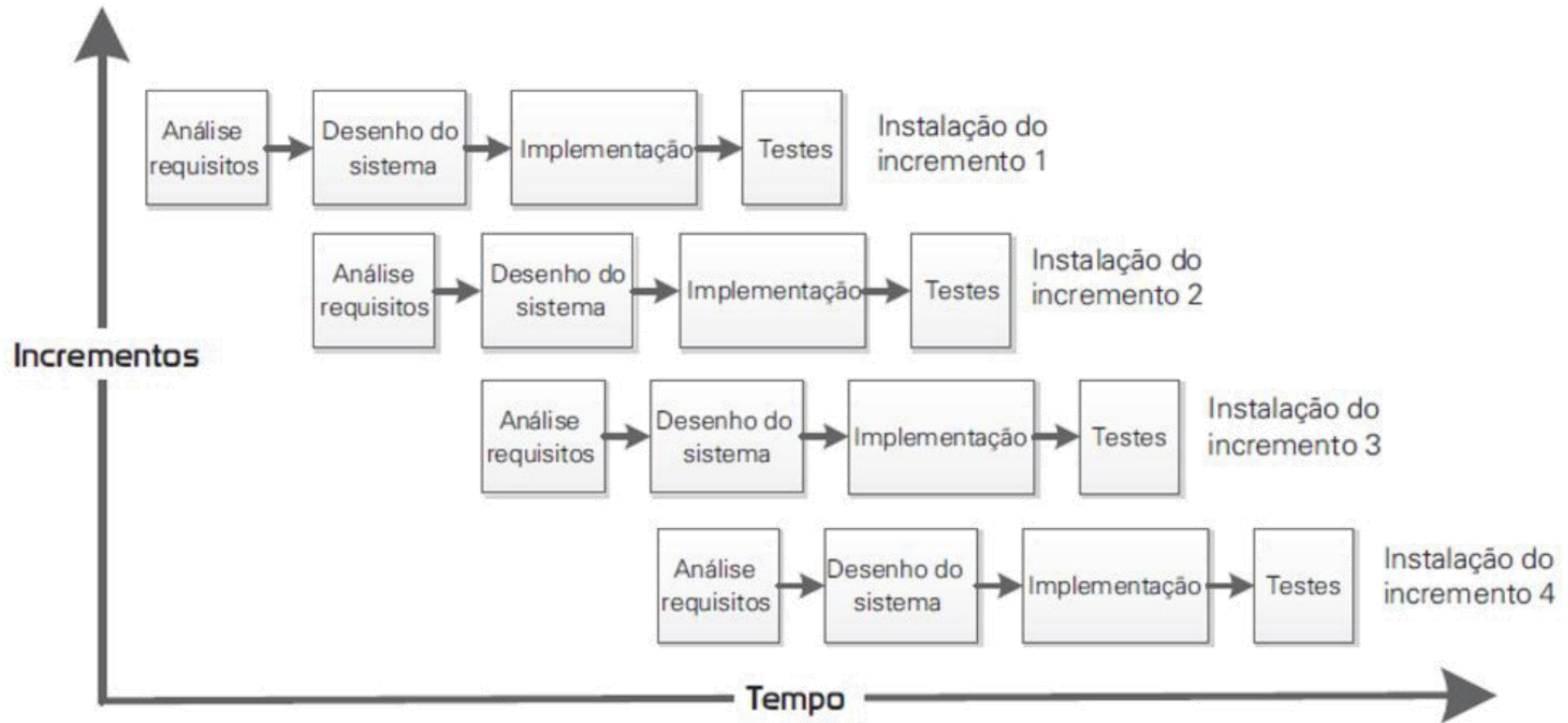


FIGURA 2.9 – PROCESSO DE DESENVOLVIMENTO INCREMENTAL

INCREMENTAL DEVELOPMENT BENEFITS

The cost of accommodating changing customer requirements is reduced.

- ❑ The amount of analysis and documentation that has to be redone is much less than is required with the waterfall model.

It is easier to get customer feedback on the development work that has been done.

- ❑ Customers can comment on demonstrations of the software and see how much has been implemented.

More rapid delivery and deployment of useful software to the customer is possible.

- ❑ Customers are able to use and gain value from the software earlier than is possible with a waterfall process.



INCREMENTAL DEVELOPMENT PROBLEMS

THE PROCESS IS NOT VISIBLE.

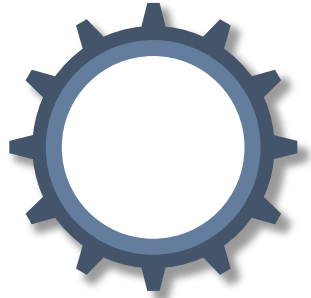
Managers need regular deliverables to measure progress. If systems are developed quickly, it is not cost effective to produce documents that reflect every version of the system.



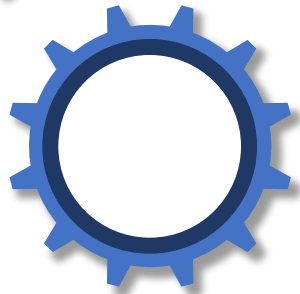
SYSTEM STRUCTURE TENDS TO DEGRADE AS NEW INCREMENTS ARE ADDED.

Unless time and money is spent on refactoring to improve the software, regular change tends to corrupt its structure. Incorporating further software changes becomes increasingly difficult and costly.

REUSE-ORIENTED SOFTWARE ENGINEERING

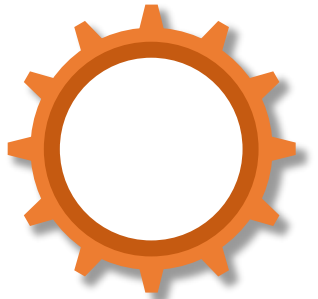


Based on **systematic reuse** where systems are integrated from existing components or COTS (Commercial-off-the-shelf) systems.



Process stages

Component analysis;
Requirements modification;
System design with reuse;
Development and integration.



Reuse is now the standard approach for building many types of business system

TYPES OF REUSABLE SOFTWARE



Stand alone application systems (sometimes called COTS) that are configured for use in a particular environment.

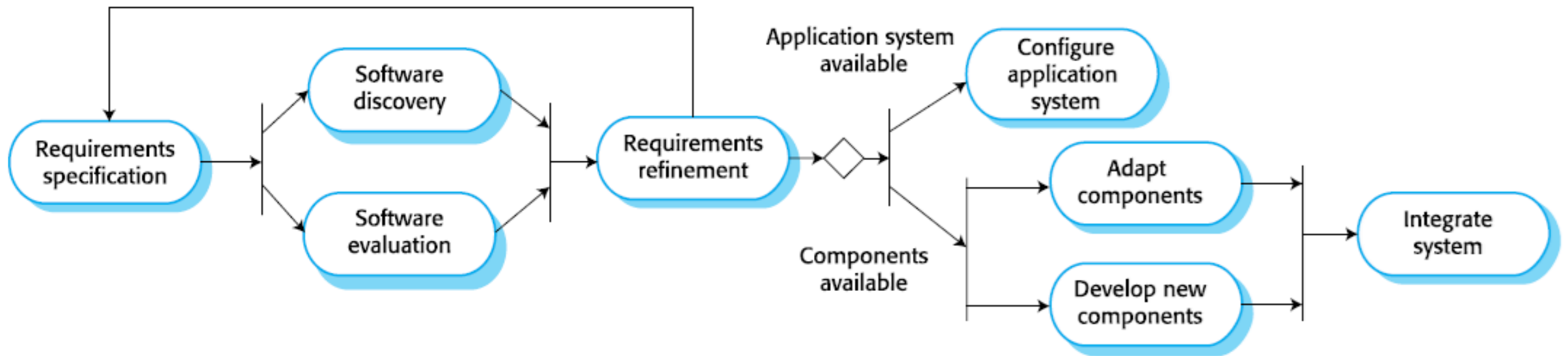


Collections of objects that are developed as a package to be integrated with a component framework such as .NET or J2EE.

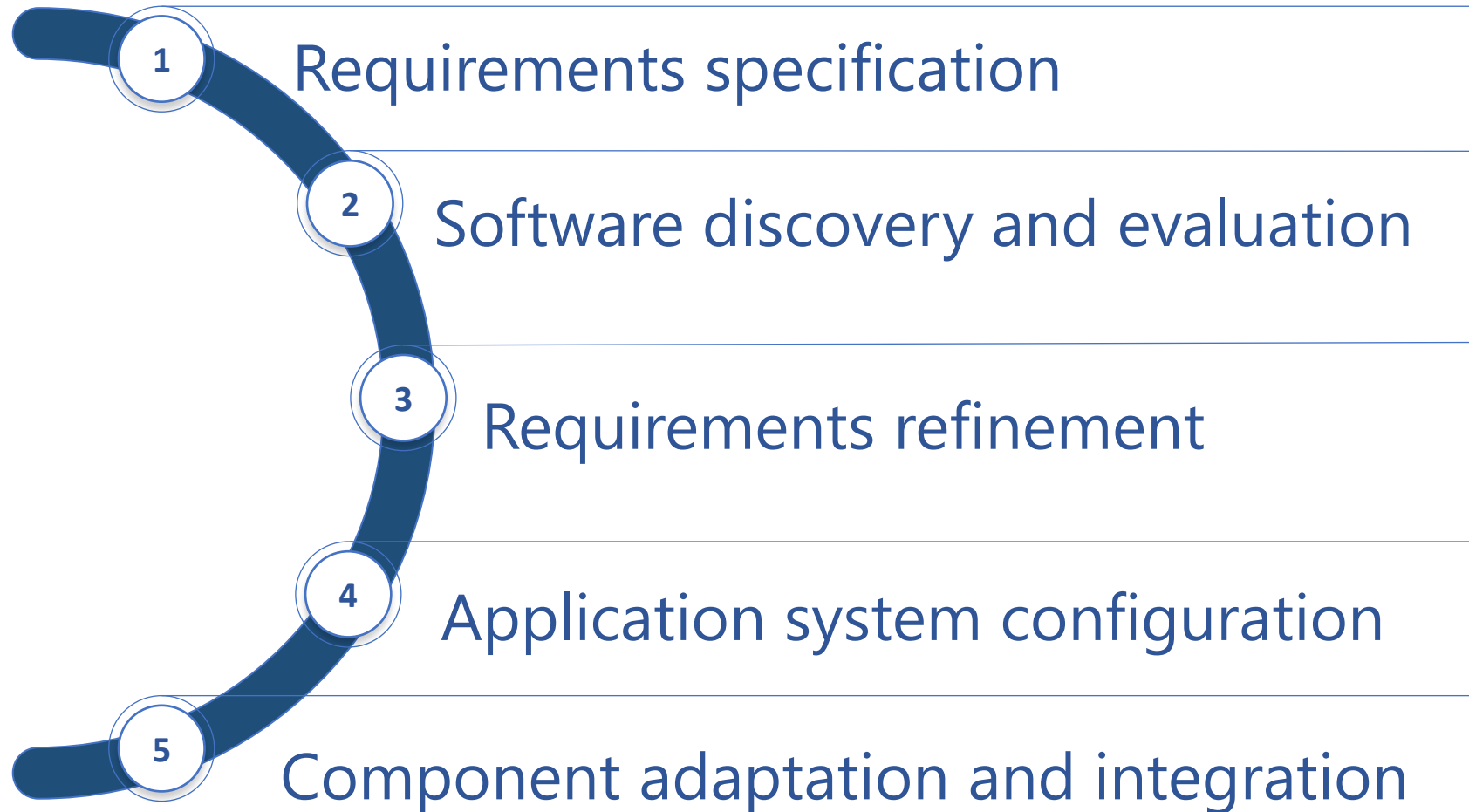


Web services that are developed according to service standards and which are available for remote invocation.

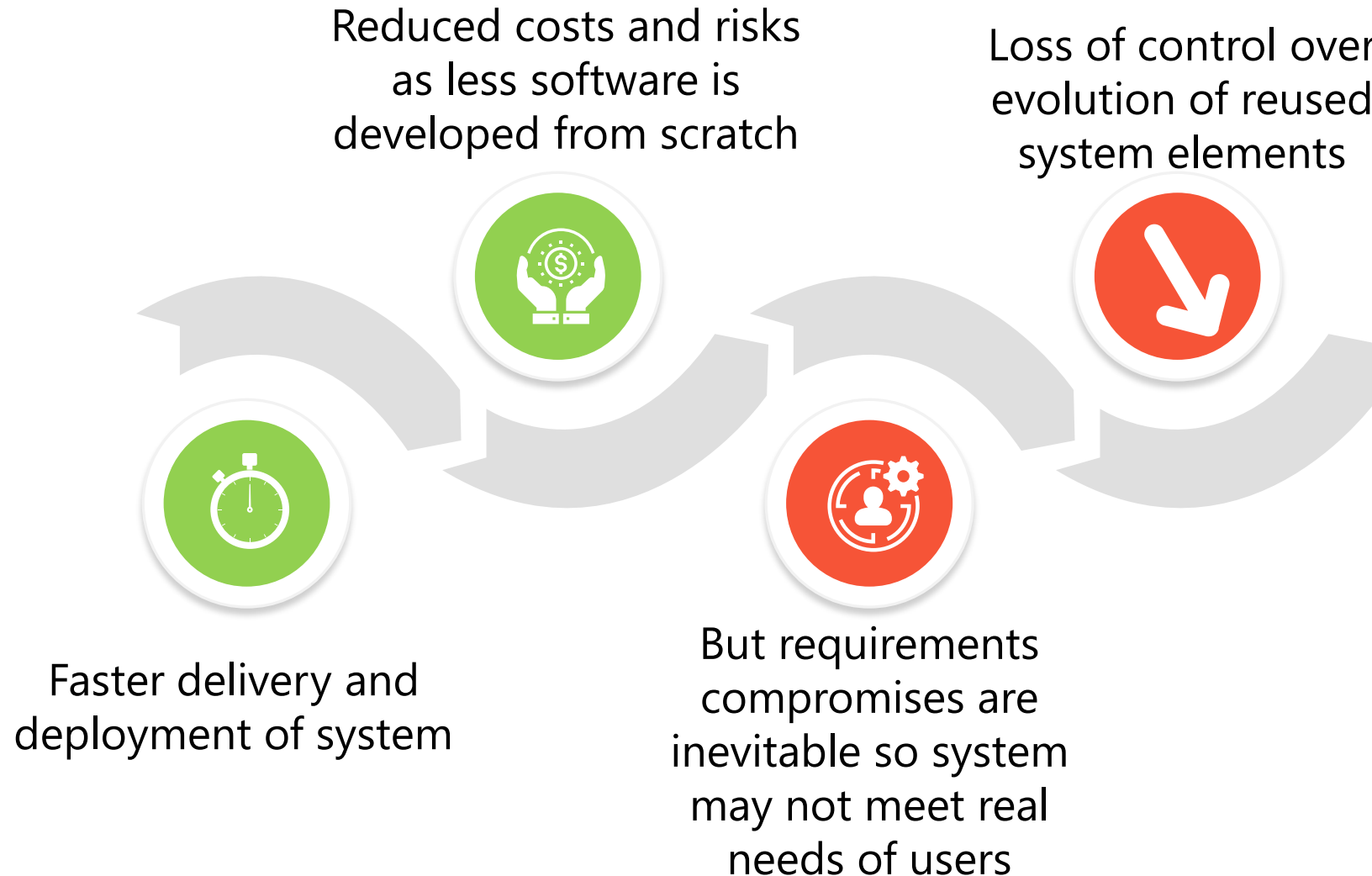
REUSE-ORIENTED SOFTWARE ENGINEERING



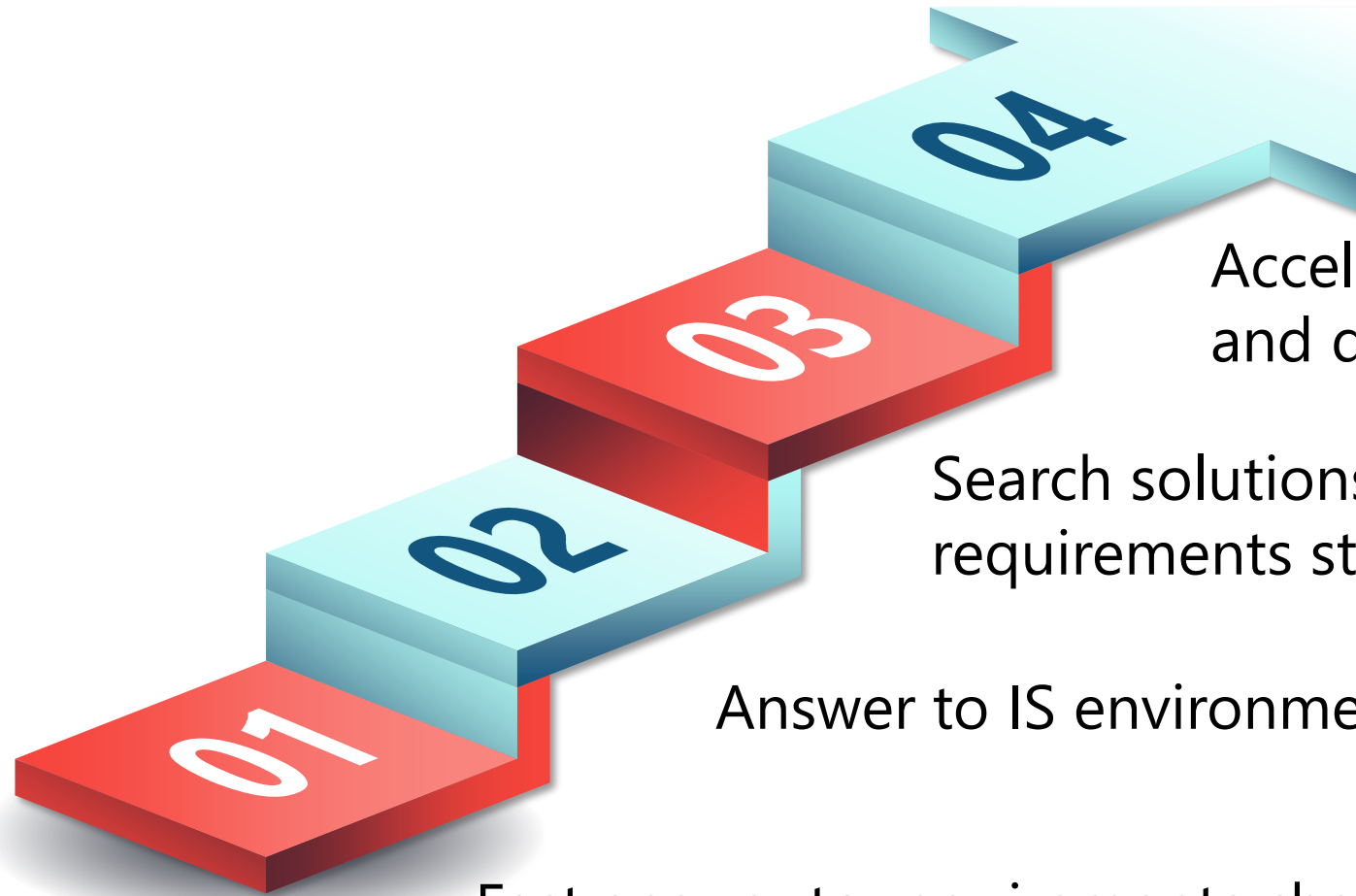
KEY PROCESS STAGES



ADVANTAGES AND DISADVANTAGES



CHALLENGES FOR THE AGILE SOFTWARE DEVELOPMENT PROCESS MODELS



Fast answer to requirements change imposed by organizations

Answer to IS environment changes

Search solutions be hound the business requirements stated objectives

Accelerate the software development and deployment process

SPIRAL WIN-WIN SOFTWARE DEVELOPMENT PROCESS MODEL

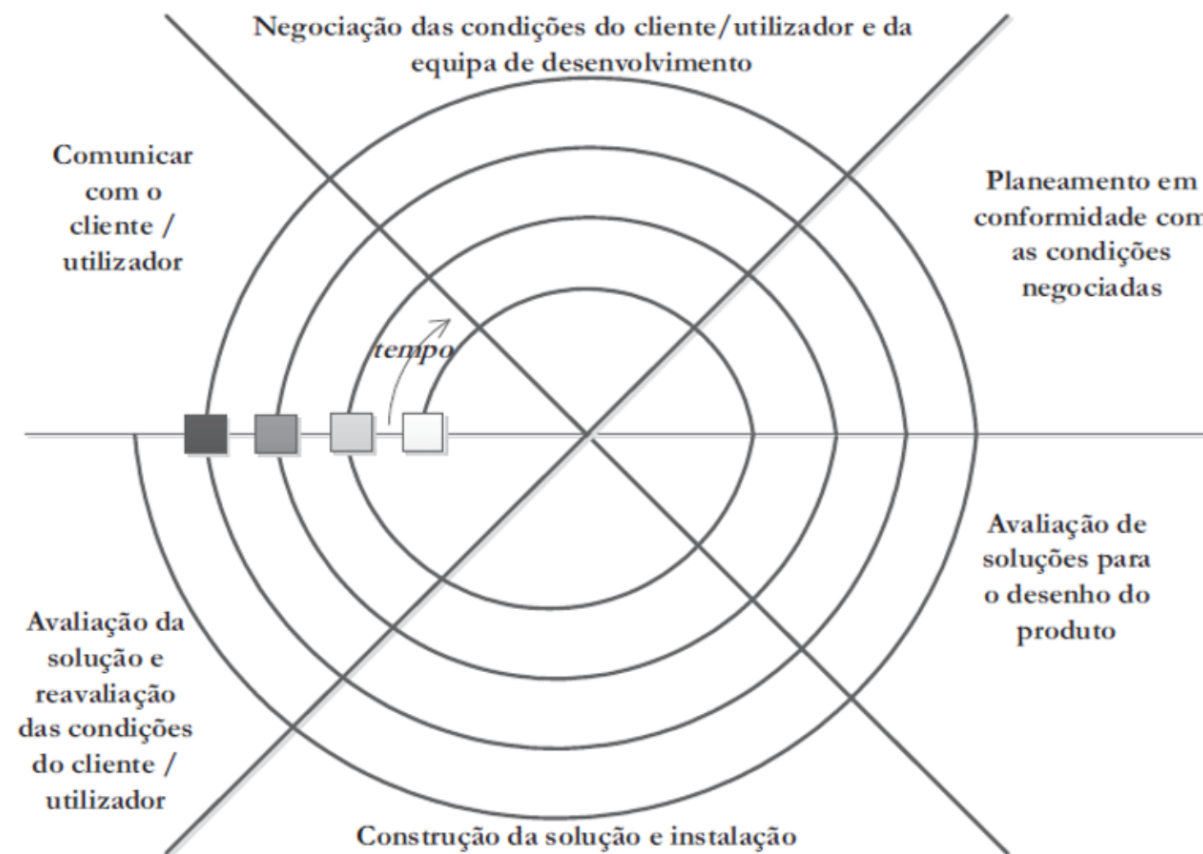


FIGURA 2.8 – PROCESSO DE DESENVOLVIMENTO EM ESPIRAL COMBINADO COM A ABORDAGEM *WIN-WIN*

REQUIREMENTS ACHIEVEMENT COMPARISON

CLASSIC APPROACH VS AGILE APPROACH

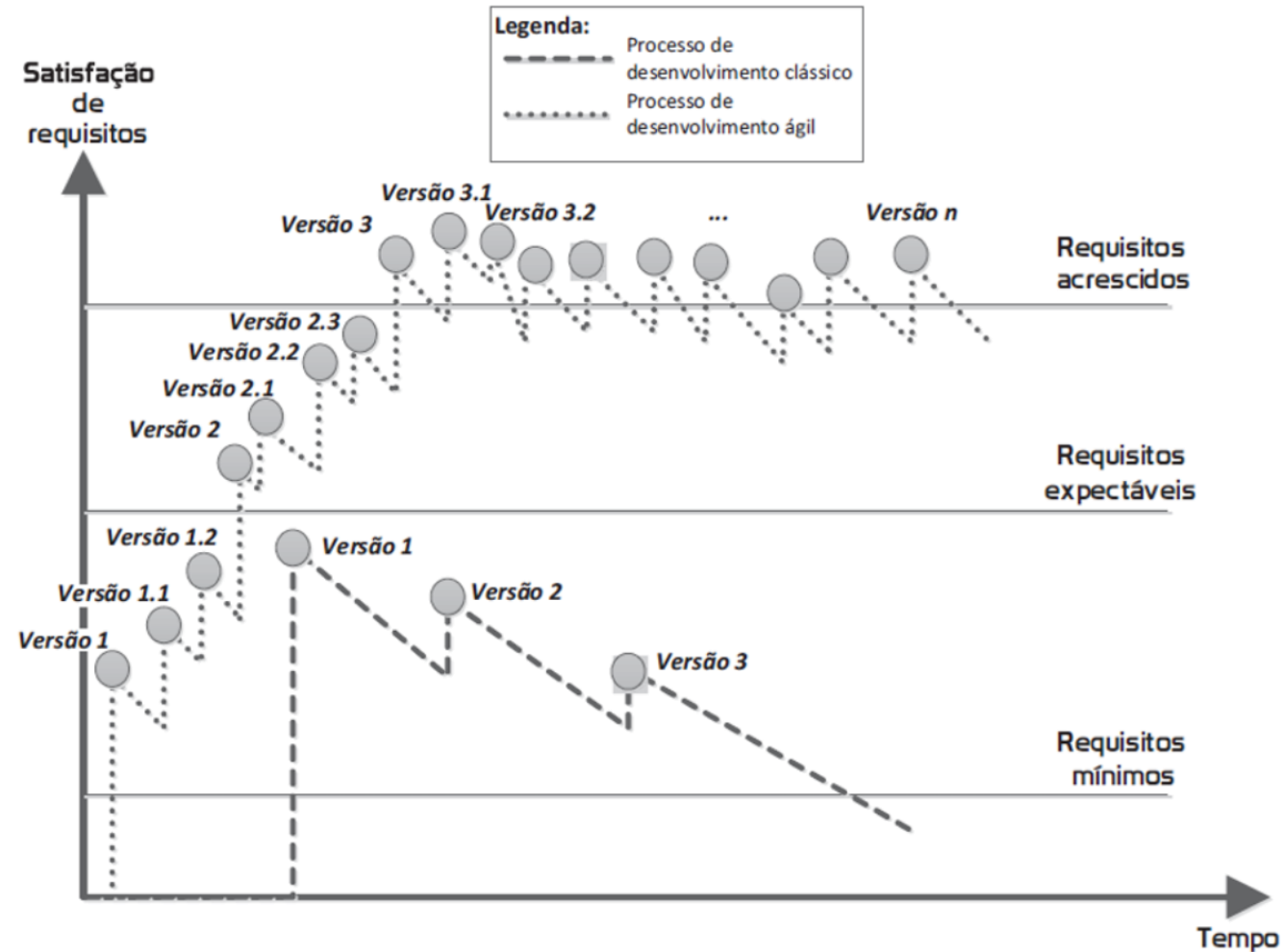
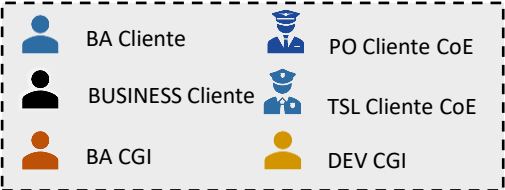


FIGURA 2.11 – COMPARAÇÃO DA SATISFAÇÃO DE REQUISITOS ENTRE PROCESSO DE DESENVOLVIMENTO CLÁSSICO E ÁGIL

Fonte: Sérgio Guerreiro, *Introdução à Engenharia de Software*, FCA – Editora de Informática

SCRUM with Agile

SDLC



UAT
Testes de aceitação do RPA desenvolvido com a área em conjunto com os BA do Cliente e da CGI.

Apenas após a aceitação da área do caderno de testes apresentado é que é solicitada a passagem a produção.

QA
Apresentação do RPA desenvolvido ao BA do Cliente. Assegurar que é desenvolvido de acordo com os requisitos descritos no caderno de desenho e tem de seguir as normas de qualidade definidas. A demonstração é efetuada seguindo os testes definidos no caderno de testes.


Quality Assurance & Code Review

Subir no reportório o seguinte:
a. software da automação
b. Caderno funcional e outra documentação que sirva de suporte à automação (PDD, ...)

GIT Lab

DEV
Desenvolvimento da solução descrita no Caderno de Desenho, segundo os requisitos e normas de qualidade definidas.
Elaboração do caderno de testes – da responsabilidade do BA da CGI + Dev da CGI. PO e BA do Cliente participam nas reuniões diárias.

PRD
O pedido da passagem a produção é efetuado pelo BA da CGI ao BA do Cliente que efetuará a formalização da subida a produção para o CoE

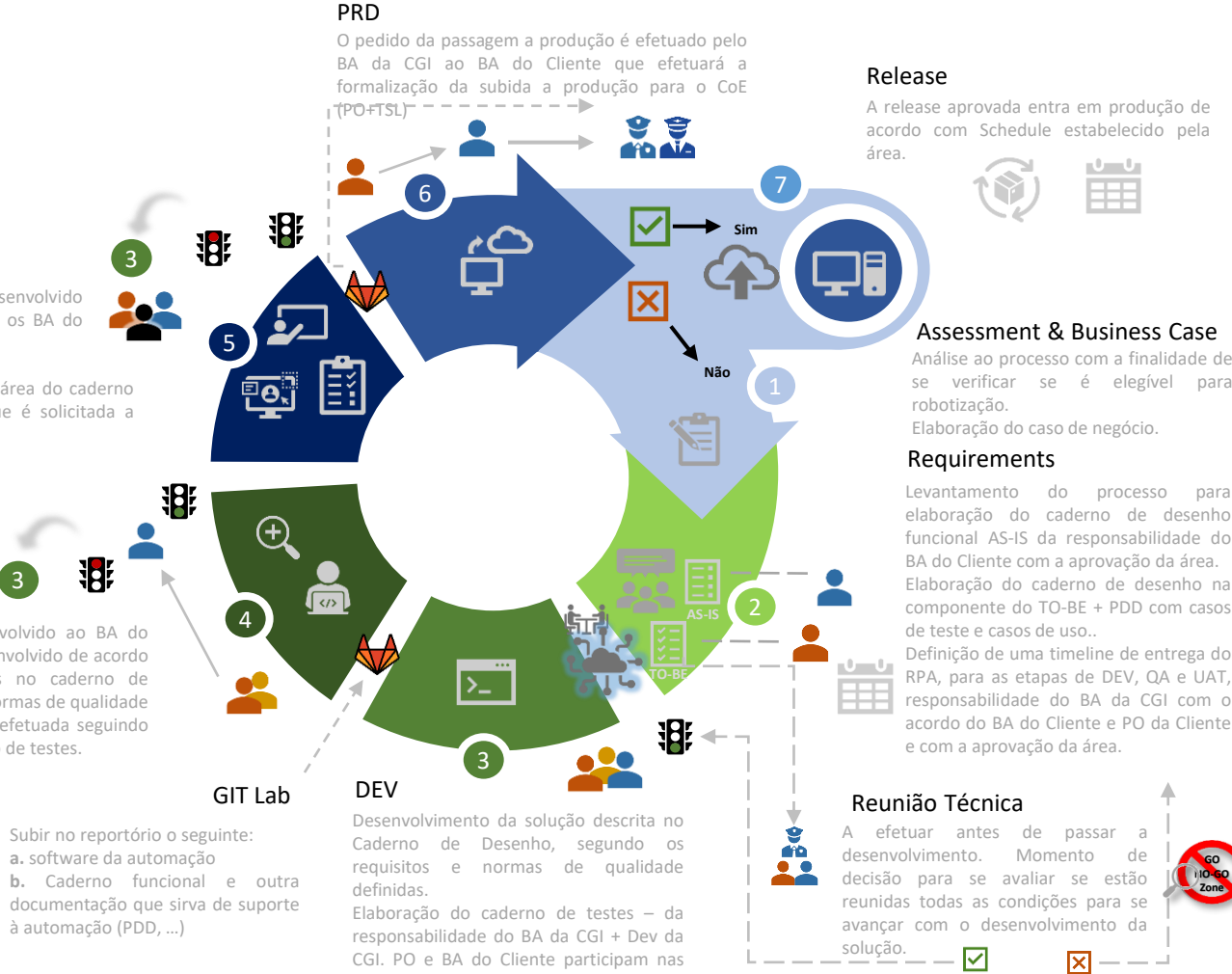
Release
A release aprovada entra em produção de acordo com Schedule estabelecido pela área.

Assessment & Business Case
Análise ao processo com a finalidade de se verificar se é elegível para robotização.
Elaboração do caso de negócio.

Requirements
Levantamento do processo para elaboração do caderno de desenho funcional AS-IS da responsabilidade do BA do Cliente com a aprovação da área.
Elaboração do caderno de desenho na componente do TO-BE + PDD com casos de teste e casos de uso..
Definição de uma timeline de entrega do RPA, para as etapas de DEV, QA e UAT, responsabilidade do BA da CGI com o acordo do BA do Cliente e PO da Cliente e com a aprovação da área.

Reunião Técnica
A efetuar antes de passar a desenvolvimento. Momento de decisão para se avaliar se estão reunidas todas as condições para se avançar com o desenvolvimento da solução.

Test Cases + Use Cases (é necessário verificar e assegurar que existem casos de teste para todos os casos de uso). Integração na arquitetura de RPAs e validação do cenário de desenvolvimento do TO-BE. Acessos a aplicações em produção e em desenvolvimento. Reutilização de módulos e componentes já existentes. Estado da arte dos ambientes de desenvolvimento (produtivo e não produtivo)



Use Cases vs Test Cases

*Um **caso de uso** captura os requisitos de negócios e do utilizador relacionados às funções do sistema - ou seja, como os utilizadores interagem com o sistema. O objetivo de um **caso de uso** é ajudar a equipa de desenvolvimento a entender exatamente o que os utilizadores esperam que o RPA faça.*

*Um **caso de uso** descreve todos os caminhos possíveis por meio de uma determinada interação utilizador / sistema, incluindo o fluxo básico e quaisquer fluxos alternativos ou de exceção. O fluxo básico (ou "natural") é aquele que atende às necessidades do utilizador.*



- › "Fluxos alternativos" são caminhos adicionais aceitáveis, mas não os mais comuns, frequentes ou desejáveis.
- › Um único caso de uso pode descrever muitos caminhos diferentes.
- › "Fluxos de exceção" são aqueles que não atendem às necessidades do utilizador devido a erros como informações ausentes ou dados inválidos e servem para o RPA dar feedback da sua execução.
- › Os casos de uso e seus requisitos são definidos e preparados pela área dona do processo que o RPA vai executar.

Use Cases vs Test Cases

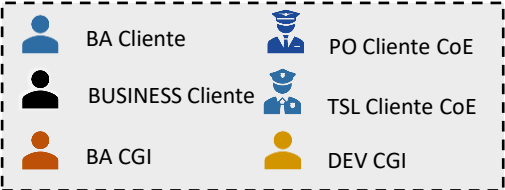
*Os **casos de teste** são usados para validar se os requisitos foram atendidos. O analista de garantia da qualidade provavelmente desejará testar o sistema completamente, configurando um **caso de teste** individual para cada funcionalidade descrita em cada um dos casos de uso. No mínimo, eles configurariam **casos de teste** separados para o "bem sucedido", cada caminho alternativo e cada caminho de exceção. Provavelmente também haveria vários casos de teste para o fluxo bem sucedido - um para cada situação que faria com que regras de negócios diferentes fossem invocadas.*



- › Os casos de uso são fornecidos aos programadores para que eles possam desenvolver a solução, e os casos de teste são fornecidos aos "beta testers" para que eles possam validar se a solução corresponde aos requisitos.
- › Portanto, os casos de uso geralmente fornecem a entrada para o desenvolvimento de casos de teste. Mas, embora os dois casos se possam sobrepor bastante, eles não são exatamente a mesma coisa.
- › A área dona do processo é quem valida se os testes efetuados são válidos e são dados como aceites.

SCRUM with Agile

SMLC



5 UAT

Testes de aceitação do RPA desenvolvido com a área em conjunto com os BA do Cliente e da CGI.

Apenas após a aceitação da área do caderno de testes apresentado é que é solicitada a passagem a produção.

4 QA

Apresentação do RPA desenvolvido ao BA do Cliente. Assegurar que é desenvolvido de acordo com os requisitos descritos no caderno de desenho e tem de seguir as normas de qualidade definidas. A demonstração é efetuada seguindo os testes definidos no caderno de testes.

Quality Assurance & Code Review

GIT Lab

Subir no reportório o seguinte:
a. software da automação
b. Caderno funcional e outra documentação que sirva de suporte à automação (PDD, ...)

PRD
O pedido da passagem a produção é efetuado pelo BA da CGI ao BA do Cliente que efetuará a formalização da subida a produção para o CoE (PO+TSL)

Release

A release aprovada entra em produção de acordo com Schedule estabelecido pela área.

1 Change Request

Análise ao pedido efetuado. Determinar se é uma manutenção evolutiva, corretiva ou urgente. Determinar prioridade em função da sua urgência, criticidade e impacto no negócio.

2 Requirements

Levantamento do processo para elaboração do caderno de desenho funcional AS-IS da responsabilidade do BA do Cliente com a aprovação da área.
Elaboração do caderno de desenho na componente do TO-BE + PDD. Definição de uma timeline de entrega do RPA, para as etapas de DEV, QA e UAT, responsabilidade do BA da CGI com o acordo do BA do Cliente e PO da Cliente e com a aprovação da área.

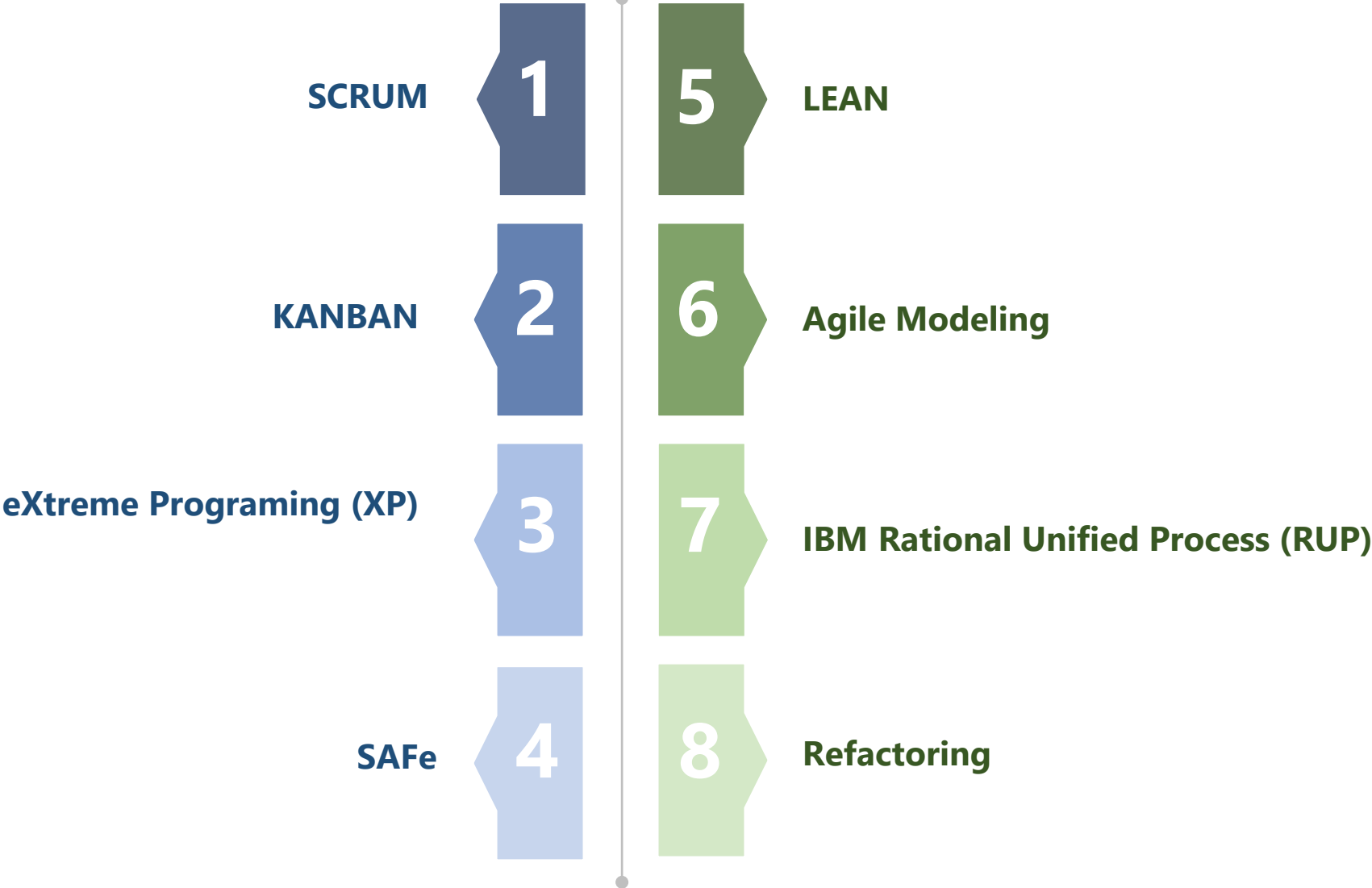
Reunião Técnica

A efetuar antes de passar a desenvolvimento. Momento de decisão para se avaliar se estão reunidas todas as condições para se avançar com o desenvolvimento da solução.

<<Questões Técnicas apenas>>

Test Cases + Use Cases (é necessário verificar e assegurar que existem casos de teste para todos os casos de uso). Integração na arquitetura de RPAs e validação do cenário de desenv. do TO-BE. Acessos a aplicações em produção e em desenvolvimento. Reutilização de módulos e componentes já existentes. Estado da arte dos ambientes de desenvolvimento (produtivo e não produtivo)

AGILE SOFTWARE DEVELOPMENT PROCESS MODELS



Q&A

