

Desenvolvimento Colaborativo de Software

Introdução aos Modelos Ágeis
(Scrum)

Rui Sousa

Instituto Politécnico do Cávado e do Ave

2023/2024

Rui Sousa | ✉ rsousa@ipca.pt

Engenharia de Software

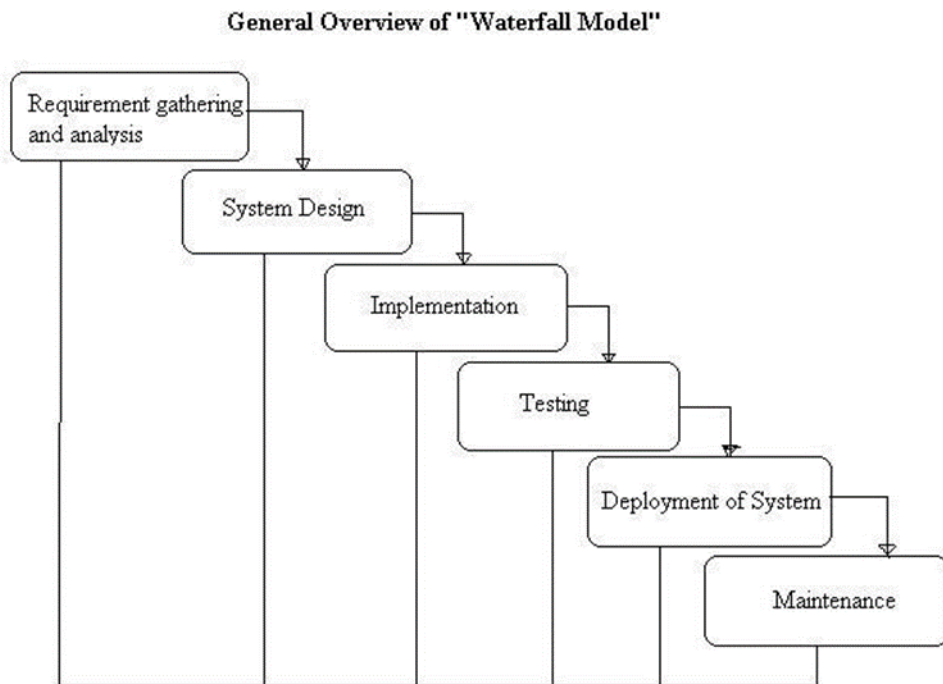
- Perceber o problema
- Planear a solução
- Executar o plano
- Analisar o resultado

Modelos tradicionais

- Modelo em “cascata” (waterfall) ou modelo sequencial
- Modelo em V
- Modelos interativos:
 - Incremental
 - Espiral
- Modelos baseados em prototipagem

Modelos tradicionais

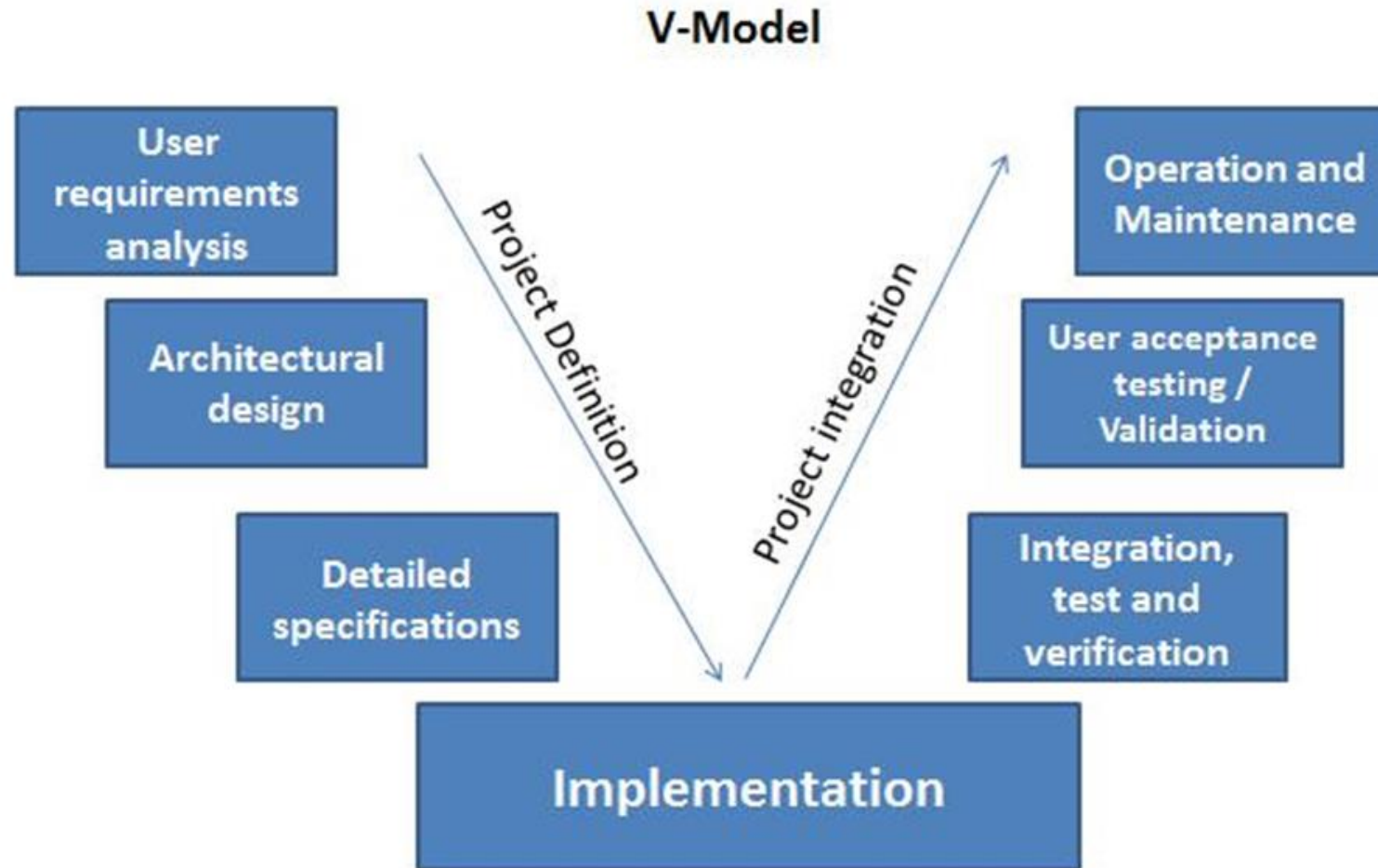
Modelo em cascata (waterfall) ou modelo sequencial



- Dos primeiros modelos propostos
- Aplica-se em projetos de grandes dimensões
- Não é um bom modelo para alterações (mudança): âmbito, custos, duração
- Obriga a uma definição correta dos requisitos
- Permite integrar várias equipas, em vários estágios do projeto
- Só avaliamos o projeto no final.

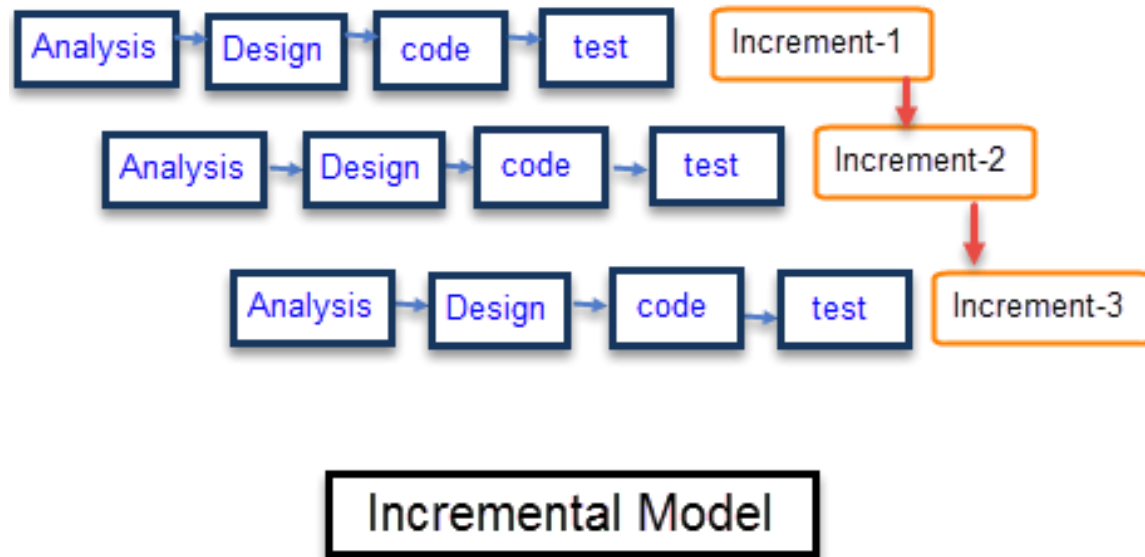
Modelos tradicionais

Modelo em V



Modelos tradicionais

Modelo em Interactivo



Modelos tradicionais

Modelos baseados em prototipagem

- Permite que o cliente “perceba” o software que está a ser desenvolvido/gerado antes de obter o produto final.
- Tem uma natureza interactiva.
- É a criação de um protótipo. Significa que é um projecto “rápido”, de pequena duração.
- Foca na demonstração de um conceito ou funcionalidade para mostrar ao utilizador funcional.
- O utilizador avalia o protótipo e refina os requisitos.



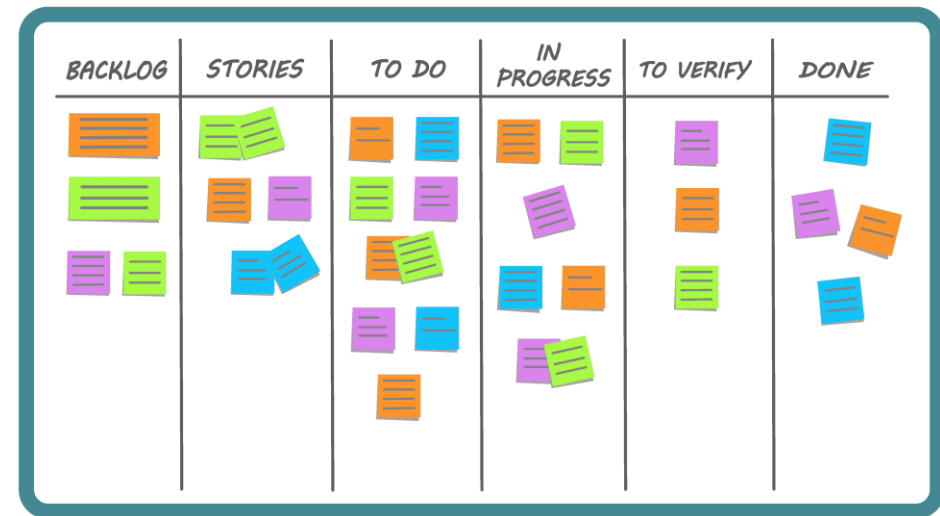
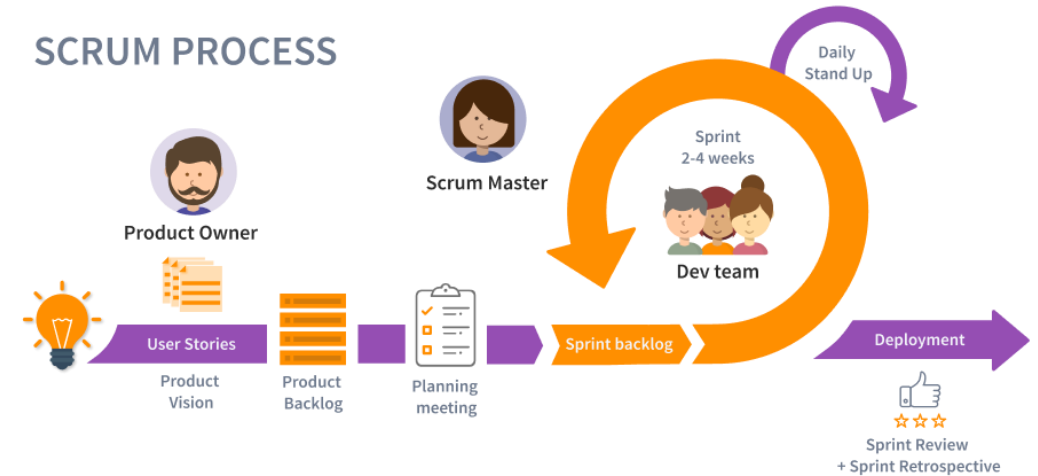
Modelos Ágeis

Nos modelos ágeis, que estão focados na mudança de requisitos, destacam-se algumas frameworks:

- SCRUM
- KANBAN



Exemplo: Kanban Board

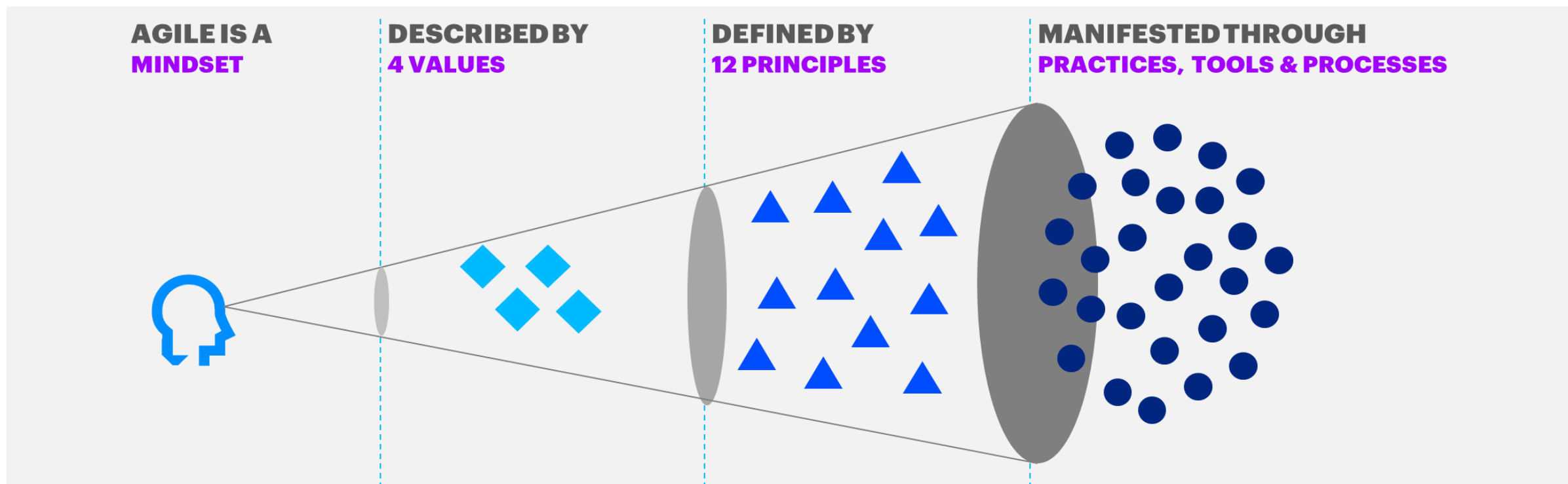


Exemplo: Kanban Board

Modelos Ágeis

O que é o Agile?

De forma geral, o **Agile** é uma cultura, uma mentalidade e uma forma de trabalhar (em equipa ou em colaboração). É descrito fundamentalmente pelos 4 valores, 12 princípios do Manifesto Agile e pode ser **explorado por meio de um número ilimitado de práticas, processos e estruturas**.



Modelos Ágeis

Valores do Agile

O ser humano faz parte do processo

Pressupostos de base:

- É difícil prever a mudança de requisitos;
- O projeto de produção de software evoluiu ao longo do processo, não é estanque;
- Não é possível planear em detalhe todas as tarefas do projeto.



Ou seja, embora haja valor nos itens da direita, valorizamos mais os itens da esquerda.

O manifesto ágil: <http://agilemanifesto.org>

Modelos Ágeis

Princípios do Agile

Princípios associados ao Manifesto Agile.

CUSTOMERS



1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.



2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.



3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

ENVIRONMENT



4. Business people and developers must work together daily throughout the project.



5. Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done.



6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

CULTURE



7. Working software is the primary measure of progress.



8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.



9. Continuous attention to technical excellence and good design enhances agility.

EXCELLENCE



10. Simplicity - the art of maximizing the amount of work not done - is essential.



11. The best architectures, requirements, and designs emerge from self-organizing teams.



12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Princípios do Agile

Porquê o Agile?

Por causa do contexto, do ambiente em que **desenvolvemos e entregamos os nossos produtos digitais**, por causa do **produto** e da **natureza dos de seus requisitos**, o software, e por causa das capacidades (skills), características e princípios da(s) equipa(s) que constroem esses produtos, os programadores:

Context/Environment:

there is no news saying we live in a VUCA (Volatility, Uncertainty, Complexity and Ambiguity) world, this means that enterprises, organizations, and people live in a world that has volatility, uncertainty, complexity, and ambiguity, it describes the situation of constant, unpredictable change that is now the norm in certain industries and areas of the business world. So, the context is a **VARIABLE** not a constant.

Nature of Software requirements or Business needs:

No software product is ever exactly, perfectly satisfactory, real software development is like shooting at a moving target, and this, has to do with the nature of business or requirements. So, we need to start accepting that **change (in business needs and hence in the product) is inevitable.**

Nature of the Product, the Software and the People, the Developers:

software development is mainly a design (not a construction) activity, thus hard to plan and cost, but also because of that it requires creative and talented people. A design activity means that basic materials will keep changing rapidly, but depending heavily on people's talent and creativity, means also that we will increase uncertainty as individuals are hard to predict and quantify.

Framework SCRUM

O que é o SCRUM?

- É uma forma de estruturar o processo de desenvolvimento
- É um Framework dinâmica na gestão de projetos:
 - Não é considerada uma metodologia, pois não interfere diretamente nas atividades de desenvolvimento de software;
 - Pode-se considerar uma forma de gestão de projetos;
- Desenvolvimento **iterativo** e **incremental**
- Principais objetivos:
 - focar na entrega de valor ao cliente;
 - Software funcional no final de cada interação;
- O Scrum estimula as equipes a aprender com as experiências, a se organizar para resolver um problema e a refletir sobre os êxitos e fracassos para melhorar sempre.
- Começa a ser um padrão no mercado.

Guia do Scrum: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Portuguese-European.pdf>

Framework SCRUM

Razões para usar o SCRUM?

- Controlo eficiente e eficaz do trabalho
- Trabalho em equipa
- Definição concreta de objetivos e atividades
- Redução de prazos e custos

- Melhora o ROI
- Satisfação do cliente

Framework SCRUM

Caracterização do SCRUM? (1/2)

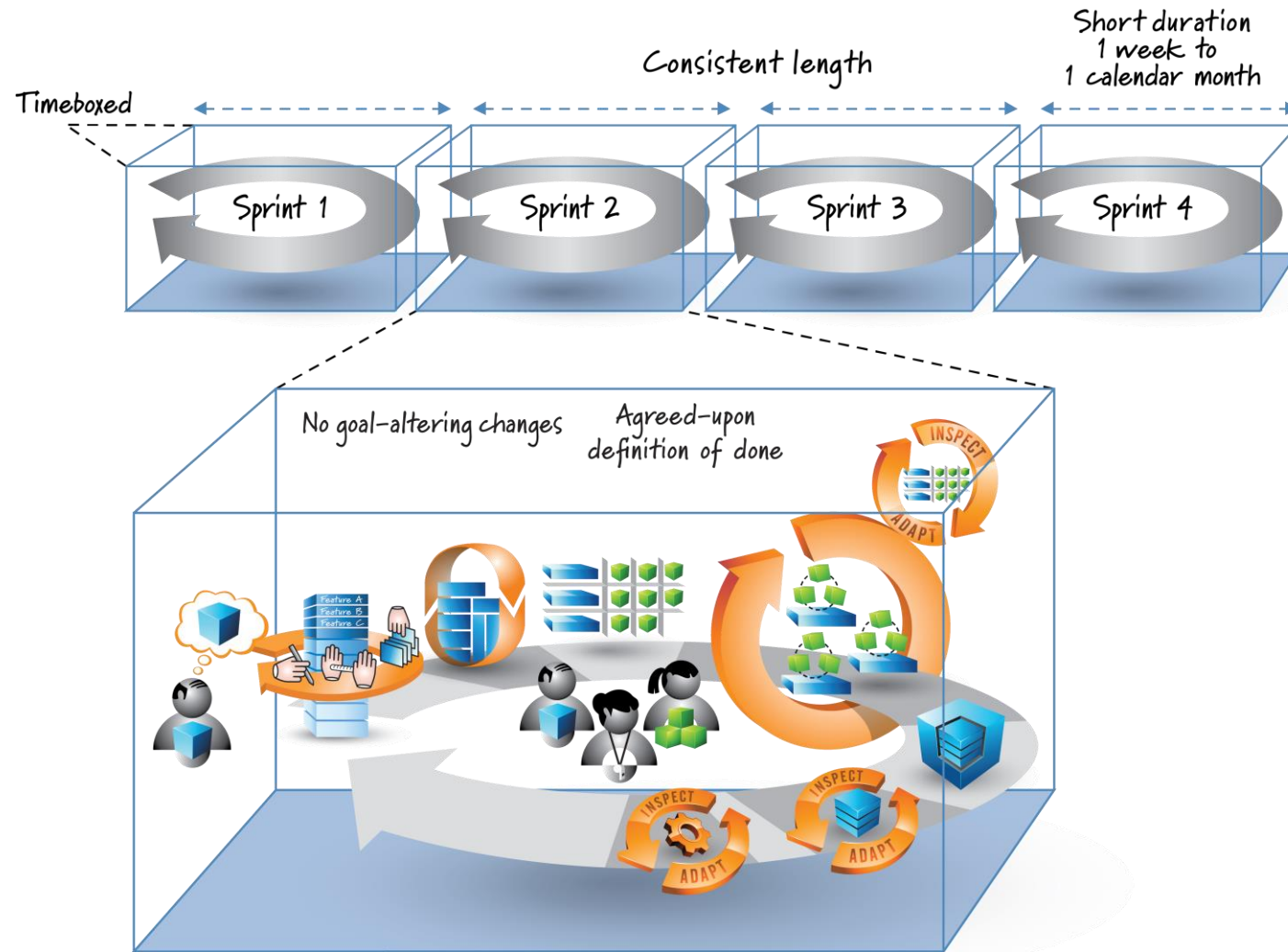
- Orientado os objetivos:
 - “sprint” → corresponde a uma meta curta, clara e realizável
 - No final de cada interação é necessário ter software funcional
- Focado na entrega de valor ao cliente:
 - Cada interação tem apenas as atividades mais prioritárias;
 - O inicio é sempre pelos requisitos mais prioritários, na visão do cliente;
- Controlo e clareza:
 - Não existem dúvidas do que está feito e do que está por fazer;
- Time-Boxing:
 - As atividades têm sempre uma duração fixa

Framework SCRUM

Caracterização do SCRUM? (2/2)

- Equipas auto-organizadas
- Aceita a mudança como parte do processo
- Flexível:
 - No final de cada interação é revisto o processo para melhorar na seguinte;
- Generalista:
 - É generalista o suficiente para poder adaptar-se a qualquer projeto
 - Inclusive é possível adaptar para outras áreas para além do desenvolvimento de software

Visão geral do SCRUM



SCRUM - Organização

PRINCIPIOS

ASPECTOS

ARTEFACTOS

EVENTOS (time-box)

PROCESSOS (fases)

FUNÇÕES

SCRUM - Princípios

PRINCIPIOS	Controle empirico do processo	transparência, inspeção e adaptação		
	Auto-organização	conscientização, articulação e apropriação		
	Colaboração	conscientização, articulação e apropriação	<p>As dimensões principais do trabalho colaborativo são:</p> <p>Consciência—Os indivíduos que trabalham juntos precisam estar cientes do trabalho um do outro.</p> <p>Articulação—Os colaboradores devem dividir o trabalho em unidades, dividir as unidades entre os membros do time, e em seguida, assim que o trabalho for concluído, devem reintegrá-lo.</p> <p>Apropriação—Adaptação de tecnologia para a própria situação; a tecnologia pode ser usada de uma maneira completamente diferente do que esperado pelos designers.</p>	<p>A necessidade de mudanças devido a requisitos mal esclarecidos são minimizadas. Por exemplo, durante os processos de Criar a Visão do Projeto, Desenvolver os Épicos, e Criar o Backlog Priorizado do Produto, o Dono do Produto colabora com os stakeholders para respectivamente criar a Visão do Projeto, os Épicos e o Backlog Priorizado do Produto. O que garante um entendimento claro entre os membros Time Central do Scrum sobre o trabalho que será necessário para a conclusão do projeto.</p>
	Prioritização baseada em valor			
	Time-boxing	Sprints, reuniões diárias, reuniões de planejamento e revisão de Sprint		
	Desenvolvimento interativo			

SCRUM - Aspectos

ASPECTOS	Organização	Papeis Centrais	Dono do Produto Scrum Master Equipa Scrum
		Papeis não essenciais	Partes Interessadas (Stakeholders) Scrum Guidance Body (SGB) Fornecedores Dono do produto chefe Scrum master chefe
	Justificação do negócio	Elaborado pelo Dono do Produto	
	Qualidade	Requisitos de negócio	Backlog do produto (prioridade do produto) Testes de qualidade Desenvolvimento, testes e documentação MELHORIA CONTINUA
	Mudança	As partes interessadas mudam de ideia ao longo do projeto. Estas mudanças são bem vindas	
	Risco	Positivo = Oportunidades Negativo = Ameaças	Probabilidade de ocorrência de cada risco Impacto potencial do risco ocorrer Risco = Probabilidade x Impacto

SCRUM - Artefactos

ARTEFACTOS	Declaração de visão do projeto	Ambito Objetivos Metas Justificação de existência	Representa a visão dos patrocinadores
	Backlog do Produto com Prioridades	Requisitos Funcionais Requisitos Não Funcionais	Responsável: Dono do produto
	Backlog do Sprint		
	Epic	Histórias de utilizador (casos de uso) de alto nível, no backlog do produto	
	User Story	Ao nível de uma funcionalidade:	
	Gráfico de Burn Down	$Y = \text{Tempo Restante} \mid X = \text{numero de dias}$ (Gráfico do Sprint)	
	Scrum Task Board	User Story To do On going Testing Ready	

SCRUM - Eventos

EVENTOS (Time Box)	Sprint	1 a 6 semanas Possuir uma meta estabelecida Scrum Master: - Remover os impedimentos Dono do produto pode cancelar Falta ou excesso de tempo pode ser revisto pela equipa scrum e pelo dono do produto	
	Sprint Planning Meeting	Duração: - 8 horas para sprint de 1 mês; - 2 horas por cada semana de sprint; Duas partes: - Etapa 1: definição do objetivo (dono do produto explica) - Etapa 2: estimativa de trabalho (backlog da sprint e o scrum task board)	1. O que será feito? 2. Como será feito? Determinar as histórias do backlog do produto a incluir na sprint (Equipa Scrum)
	Daily Scrum	15 minutos de reunião Perguntas base: - O que eu fiz no projeto desde a última reunião? - O que vou fazer até à próxima reunião? - Quais os impedimentos que tenho?	Scrum Master e a Equipa Scrum Diariamente o gráfico de burn down deve ser atualizado
	Sprint Review	Responsável: Dono do produto Duração: 4 horas para sprint de 1 mês A equipa apresenta os itens desenvolvidos segundo os critérios de aceitação O dono do produto valida, segundo os critérios de aceitação	Quando um item é rejeito, entra novamente no backlog do produto O dono do produto discute o estado do backlog do produto
	Sprint Retrospective	Responsável: Scrum Master Duração: 4 horas para sprint de 1 mês Responde: - O que funcionou? Para repetir; - O que não funcionou? Para corrigir;	
	Refinamento do Backlog do Produto	Responsável: Dono do Produto; Não tem tempo, logo não é time box Estimativa de atividades: equipa scrum Prioridade das atividades: dono do produto	

SCRUM - Processos

PROCESSOS (fases)	INICIAR	Criar a visão do projeto	Caso de negócio do projeto -> Declaração da visão do projeto; Dono do produto é identificado; Reunião de stakeholder
		Identificar o Scrum Master e as partes interessadas	Scrum Master Stakeholders
		Formar a equipa scrum	Responsabilidade do Dono do Produto, com o auxílio do Scrum master
		Desenvolver os épicos	Declaração de visão do projeto -> Épicos
		Criar o backlog do produto com prioridades	Épicos -> backlog do produto; Estabelecer os critérios de pronto;
		Conduzir o planeamento da versão	
	PLANEAMENTO E ESTIMATIVA	Criar as histórias de utilizador	Feito pelo Dono do produto; Envolver a equipa Scrum; Adicionar ao backlog do produto;
		Aprovar, estimar e comprometer as histórias de utilizador	Dono do produto: define as histórias a incluir no sprint; Scrum master e equipa scrum: estimam os esforços necessários e comprometem-se a entregar de acordo com os requisitos do cliente;
		Criar as tarefas	Scrum master e Equipa Scrum
		Estimar as tarefas	Time Central de Scrum
		Criar o backlog da Sprint	Time Central de Scrum -> na reunião de planeamento da Sprint
	IMPLEMENTAR	Criar os entregáveis	Equipa Scrum Scrum Board
		Conduzir a reunião diária	Scrum Master Equipa Scrum
		Refinar o backlog do produto	Dono do produto
		Convocar o Scrum de Scrums	Para grandes projetos, com várias equipas scrum
	REVISÃO e RETROSPECTIVA	Demonstrar e validar a Sprint	Equipa Scrum, apresenta o resultado ao dono do produto e a outras pessoas relevantes; Garantir a aprovação do dono do projeto, para todos os entregáveis da Sprint
		Retrospectiva da Sprint	Scrum Master e equipa scrum realizam uma reunião para analisar o resultado de um sprint; Conhecido como lições aprendidas
	RELEASE (versão)	Envio de entregáveis	Os entregáveis são enviados para os stakeholders relevantes
		Retrospectiva do projeto	stakeholders e equipa central do scrum pontos de melhoria para projetos futuros

SCRUM - Funções

Scrum Master

Um **Scrum Master** é responsável por garantir que uma equipa Scrum esteja a operar da forma mais eficaz possível com os valores Scrum.

Isso significa que eles mantêm a equipa alinhada com os princípios de SCRUM, planeiam e lideram reuniões e resolvem quaisquer obstáculos que a equipa possa enfrentar.

Os Scrum Masters também podem desempenhar um papel mais amplo dentro de uma organização para ajudá-la a incorporar os conceitos do Scrum no seu trabalho.

O Scrum pode ser diferente de organização para organização e de equipa para equipa, tornando as tarefas específicas de um Scrum Master variadas.

Um Scrum Master pode ter as seguintes responsabilidades:

- Facilitar reuniões diárias do Scrum (também chamadas de “standups diários”).
- Liderar reuniões de planeamento de sprint. Conduza revisões “retrospectivas” para ver o que deu certo e o que pode ser melhorado no sprint seguinte.
- Acompanhe os membros da equipa, por meio de reuniões individuais ou outros meios de comunicação.
- Gerencie os obstáculos que surgem para a equipa, comunicando-se com as partes interessadas fora da equipa.

SCRUM - Funções

Product Owner

O Product Owner é responsável por maximizar o valor do produto resultante do trabalho da Scrum Team.

A forma como isto é feito pode variar muito entre organizações, ScrumTeams e indivíduos.

O Product Owner pode fazer o trabalho acima referido ou pode delegá-lo a outros. Independentemente disso, o Product Owner permanece responsável.

O Product Owner é uma pessoa, não um comité. O Product Owner pode representar as necessidades de muitos intervenientes no Product Backlog. Aqueles que desejam alterar o Product Backlog podem fazê-lo, tentando convencer o Product Owner.

O Product Owner é também responsável pela gestão eficaz do Product Backlog, que inclui:

- Desenvolver e comunicar explicitamente o Product Goal;
- Criar e comunicar claramente os itens do Product Backlog;
- Ordenar os itens do Product Backlog; e,
- Assegurar que o Product Backlog é transparente, visível e compreendido.

SCRUM - Funções

Development Team

Os Developers são as pessoas da Scrum Team que estão empenhados em criar qualquer aspecto de um Increment utilizável em cada Sprint.

As competências específicas necessárias aos Developers são frequentemente amplas e variam com o domínio de trabalho. Contudo, os Developers são sempre responsáveis por:

- Criar um plano para o Sprint, o Sprint Backlog;
- Incutir qualidade, aderindo a uma Definition of Done;
- Adaptar o seu plano todos os dias na direção do Sprint Goal; e,
- Responsabilizarem-se uns aos outros como profissionais.

Q&A