

Guia das Metodologias Ágeis

ticapp.gov.pt

Índice

1.	Introdução.....	4
1.1.	Organização.....	4
2.	Metodologias Ágeis	5
2.1.	Metodologias Tradicionais	5
2.2.	Metodologias Ágeis.....	6
3.	Scrum	7
3.1.	Papéis	7
3.1.1.	ScrumMaster.....	7
3.1.2.	Product Owner.....	7
3.1.3.	Equipa.....	8
3.2.	<i>Sprint</i>	8
3.3.	Artefactos.....	9
3.3.1.	<i>Product Backlog</i>	9
3.3.2.	<i>Sprint Backlog</i>	11
3.4.	Reuniões.....	13
3.4.1.	<i>Sprint Planning</i>	13
3.4.2.	<i>Daily Meeting</i>	13
3.4.3.	<i>Sprint Review</i>	14
3.4.4.	<i>Sprint Retrospective</i>	15
3.5.	<i>Framework</i>	16
3.6.	Começar.....	17
3.6.1.	<i>Para saber mais</i>	18

Referências a Outros Documentos

ID	Descrição	Autor

Histórico do Documento

Versão	Descrição	Data	Autor
1.0	Versão Inicial	18-03-2021	TicAPP

Acrónimos e Abreviaturas

Acrónimos	Descrição

1. Introdução

Este documento tem como objetivo descrever metodologias ágeis que se consideram ser uma boa prática na gestão / implementação de projetos e que deverão ser aplicadas em projetos.

1.1. Organização

Este documento encontra-se organizado nas seguintes secções:

- **Introdução**

Secção com a introdução ao documento onde são referenciadas informações como o objetivo e organização do mesmo.

- **Metodologias Ágeis**

Secção onde se descreve o que é uma metodologia ágil.

- **Scrum**

Secção onde se descreve a metodologia Scrum.

2. Metodologias Ágeis

2.1. Metodologias Tradicionais

Uma metodologia tradicional pressupõe a existência de um conjunto de fases na gestão / implementação de um projeto que são rígidas, perfeitamente delimitadas e com execução sequencial. A metodologia “Em Cascata” é o exemplo clássico de uma metodologia tradicional:

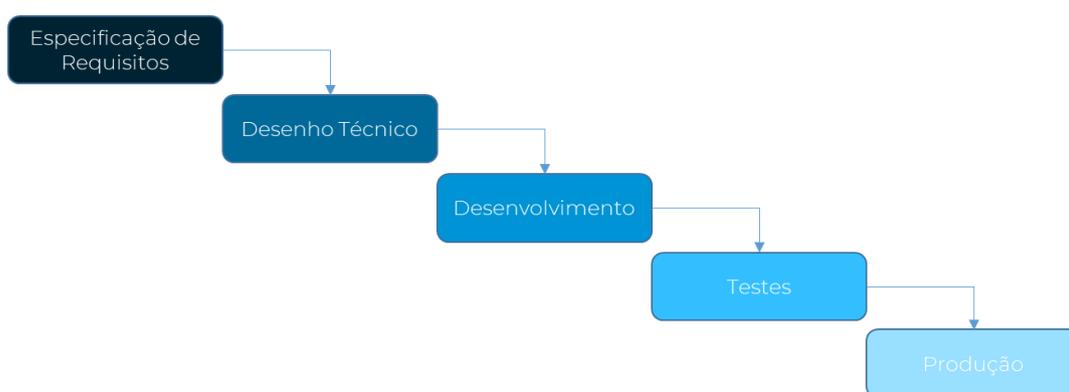


Figura 1 – Fases da metodologia “Em Cascata”

As metodologias tradicionais foram amplamente utilizadas na gestão / implementação de projetos ao longo dos anos. Com a experiência da sua utilização foi possível identificar as seguintes desvantagens:

- Cada fase do desenvolvimento do projeto tem de ser concluída previamente ao início da fase seguinte;
- Todos os requisitos têm de ser identificados durante a fase de Análise do projeto;
- Não prevê alteração aos requisitos definidos inicialmente, pelo que é difícil incorporar necessidades adicionais do negócio ao longo do projeto;
- Existe um grande intervalo de tempo desde o início do projeto até que o *Sponsor* finalmente tem contacto com o produto final e pode validar o mesmo;
- Determina a produção de documentação excessiva, o que torna a tarefa de atualização da mesma muito difícil;
- O foco e esforço desenvolvido é no processo e não no produto final, nem na sua adequação às necessidades dos *stakeholders*.

2.2. Metodologias Ágeis

As metodologias ágeis surgem da necessidade de ultrapassar as dificuldades e desvantagens de aplicação das metodologias tradicionais na gestão / implementação de projetos. A metodologia ágil pressupõe **períodos de tempo curtos entre cada entrega**, com o objetivo de garantir uma entrega antecipada e contínua de software suscetível de avaliação. A implementação da nova solução é assim **iterativa e incremental**, permitindo desde muito cedo confirmar se o objeto entregue vai ou não ao encontro das necessidades e proceder às respetivas correções com baixo risco e custo.

São apontadas as seguintes vantagens na utilização de metodologias ágeis na gestão / implementação de projetos:

- Abordagem alternativa às metodologias tradicionais de desenvolvimento de projetos, permitindo colmatar as desvantagens associadas à sua utilização;
- Reduz a complexidade do projeto através da divisão em iterações incrementais com entregáveis bem definidos (designados *Sprints*);
- São definidos períodos de tempo curtos entre cada entrega;
- Existe uma entrega antecipada e contínua de *software* suscetível de avaliação pelos *stakeholders*;
- Promove a deteção precoce de falhas no processo de desenvolvimento e a sua rápida resolução com menor risco e custo;
- Melhora a comunicação e maximiza a cooperação entre todos os intervenientes;
- Estabelece um conjunto de regras e práticas de gestão.

Alguns exemplos de metodologias ágeis são:

- Scrum;
- Kanban;
- Extreme Programming (XP);
- Lean Development;
- Crystal.

3. Scrum

Neste capítulo, irá ser descrita a metodologia Scrum, bem como todos os conceitos inerentes à sua utilização.

3.1. Papéis

A metodologia Scrum pressupõe que os seguintes papéis sejam atribuídos e desempenhados pelos intervenientes no projeto:

- *ScrumMaster*;
- *Product Owner*;
- Equipa.

3.1.1. ScrumMaster

O *ScrumMaster* tem como principal objetivo a criação e manutenção de uma Equipa saudável, com desempenho ótimo e que aplica a metodologia Scrum corretamente.

Estas são as principais responsabilidades que o *ScrumMaster* deve assegurar:

- Garantir que a metodologia Scrum é aplicada corretamente;
- Ser o facilitador das reuniões diárias / de revisão / retrospectiva;
- Garantir que o foco da Equipa se mantém nos objetivos definidos para o *sprint*;
- Identificar e resolver obstáculos / impedimentos ao desempenho ótimo da Equipa;
- Analisar a produtividade da Equipa.

3.1.2. Product Owner

O *Product Owner* tem como principal objetivo garantir que o produto entregue vai ao encontro das necessidades de negócio e que cada *sprint* contribui para a mais valia do produto final.

Estas são as principais responsabilidades que o *Product Owner* deve assegurar:

- Gerir o *Product Backlog* e garantir que este está sempre atualizado e priorizado;
- Representar e gerir os interesses dos *Stakeholders*;

- Definir e garantir o alinhamento de todos os intervenientes no projeto sobre a visão do produto;
- Garantir que o trabalho entregue em cada *sprint* é uma mais valia para o negócio;
- Decidir sobre a criação de uma *release* oficial, que poderá entrar em exploração;
- Decidir excecionalmente abortar um *sprint* em caso de alterações ao negócio que o inviabilizem.

3.1.3. Equipa

A Equipa tem como principal objetivo garantir a execução e cumprimento dos objetivos definidos para o *sprint*.

Estas são as principais responsabilidades que a Equipa deve assegurar:

- Ser autónoma na definição dos seus compromissos;
- Ser autónoma na gestão e organização do seu trabalho;
- Ser autónoma no cumprimento dos objetivos do *sprint*;
- Garantir a execução dos itens prioritários do *backlog*;
- Ser multidisciplinar a fim de dar resposta às necessidades do *sprint*.

3.2. *Sprint*

Sprint é um conceito da metodologia Scrum que é definido pelas seguintes características:

- Corresponde a um período fixo de tempo durante o qual se pretende produzir um entregável com valor acrescido para o negócio;
- O período fixo de tempo pode variar de 1 a 4 semanas, sendo tipicamente definido como 2 semanas;
- Inclui todas as tarefas necessárias à produção do entregável que é definido como objetivo do *sprint*;
- A Equipa é totalmente responsável pela concretização dos objetivos e compromissos assumidos para o *sprint*, sem interferências externas;

- Exceccionalmente, o *sprint* pode ser abortado se deixar de fazer sentido, nomeadamente em caso de alteraões ao negcio que o inviabilizem;
- Enquanto decorre o *sprint*, o *Product Owner* trabalha adicionalmente sobre os prximos itens do *Product Backlog*, a fim de realizar o processo de *grooming* ou refinamento, onde so descritos os vrios itens com detalhe para que possam ser posteriormente estimados pela Equipa.

3.3. Artefactos

A metodologia Scrum pressupe que os seguintes artefactos sejam produzidos e geridos ao longo do projeto:

- *Product Backlog*;
- *Sprint Backlog*.

3.3.1. Product Backlog

A lista do trabalho a ser executado e implementado para a construo do produto final  designada de *Product Backlog*, sendo definida pelas seguintes caractersticas:

- O *Product Backlog*  criado e gerido pelo *Product Owner*;
- Pode ser constitudo por uma lista de requisitos, *user stories*, funcionalidades, tarefas ou *bugs*;
- O seu foco est na mais valia que o item da lista trar para o negcio;
- Corresponde a uma lista ordenada e priorizada por valor acrescido e risco que representa para o negcio (ordenao descendente);
- Cada item da lista contm informao sobre os respetivos critrios de aceitao pelo *Product Owner* / *Stakeholders*, a que a respetiva implementao ter de responder e  alvo de duas estimativas: a estimativa inicial ou de alto nvel (atribuda durante a definio de *roadmaps*) e a estimativa detalhada (atribuda na definio dos objetivos e compromissos dos *sprints*);
- A execuo de cada item da lista tem de respeitar a Definio de Feito, ou o conjunto de tarefas que a Equipa ter de realizar para que o item da lista possa ser considerado terminado (ex. cumpre critrios de acessibilidade; testes unitrios).

O *Product Backlog* permite assim à Equipa partir o produto final em partes mais pequenas, de modo a construir o produto de forma iterativa e incremental, ao longo de curtos períodos de tempo que se designam *sprint*.

Backlog 55 of 88 issues visible [Clear all filters](#) Create sprint ...

FE Alertas de erro para Imagem + Anexos	2 VERSIONS	Submissão de Proposta...	PART50-319	↑	
Frontend					
BE Pesquisa e Filtros - Lista propostas/ pr	2 VERSIONS	Pesquisa e Filtros Pági...	PART50-74	↑	8
Backend					
BE Pesquisa e Filtros - Lista propostas/ f	2 VERSIONS	Pesquisa e Filtros Pági...	PART50-102	↑	5
Backend					
BE Pesquisa e Filtros - Lista propostas/ f	2 VERSIONS	Pesquisa e Filtros Pági...	PART50-101	↑	5
Backend					
BE - Melhorar logging da aplicação	2 VERSIONS		PART50-148	↑	13
Backend					
BE BO - Estado propostas/ projetos	2 VERSIONS	BO - Estado das propo...	PART50-258	↑	2
Backend					

Figura 2 – Exemplo de Product Backlog

Estimativas

O *Product Backlog* é estimado em unidades de valor abstratas, tais como *story points*, com base na utilização de um modelo de pesos relativos. O significado dos *story points* é determinado pela Equipa (ex. níveis de complexidade de implementação).

Esta estimativa pode ser obtida através de várias técnicas, por exemplo, por aplicação do método *Planning Poker*. Na aplicação deste método, cada elemento da Equipa tem um conjunto de cartas com a seguinte sequência 0, 1, 2, 3, 5, 8, 13, 20, 40 e 100 disponível para sua seleção. Após conhecimento de todos os detalhes sobre o item da lista e esclarecidas todas as questões, cada elemento da Equipa seleciona uma carta que corresponde à estimativa que considera deva ser atribuída ao item. No caso de existirem seleções divergentes entre os vários elementos da Equipa, tem de haver um momento de discussão das várias estimativas, para que consigam chegar a um consenso da estimativa a atribuir.

3.3.2. Sprint Backlog

A lista das tarefas que a Equipa tem de executar durante o *sprint*, a fim de que exista um incremento de funcionalidade do produto no final do *sprint*, é designada de *Sprint Backlog*, sendo definida pelas seguintes características:

- O *Sprint Backlog* é definido durante a reunião de *Sprint Planning*, sendo criado e gerido pela Equipa;
- A lista das tarefas selecionada para execução durante o *sprint* resulta das tarefas prioritárias do *Product Backlog*;
- Cada item da lista tem uma estimativa detalhada associada.

The screenshot shows a Jira Sprint Backlog for 'PART50 - Sprint 5'. At the top, it indicates '13 of 29 issues visible' and 'Clear all filters'. There are three status indicators: 56 (grey), 33 (blue), and 9 (green). A 'Plan sprint' button is visible. The sprint dates are '12/Mar/20 4:29 PM' to '26/Mar/20 4:29 PM'. Below the header, there are several team avatars. The main area contains a list of tasks with the following details:

Task Name	Category	Version	Label	ID	Priority	Estimate
BE Detalhe proposta/ projeto - Contactos do Pr	Backend	2 VERSIONS	Detalhe Proposta	PART50-154	↑	1
BE Campo TEMA	Backend	2 VERSIONS	Submissão de Proposta...	PART50-292	↑	8
BE Criar ID numérico para a propo	Backend	2 VERSIONS	Submissão de Proposta...	PART50-179	↑	5
Adicionar tabelas de histórico de alterações	Backend		PART5.0	PART50-154	↑	13
FE Campo TEMA	Frontend	2 VERSIONS	Submissão de Proposta...	PART50-293	↑	2
FAQ's - Estilo dos CSS	Frontend	2 VERSIONS	DI - FAQ's	PART50-286	↑	

Figura 3 – Exemplo de Sprint Backlog

A Equipe mantém as tarefas do *Sprint Backlog* atualizadas diariamente em relação à sua execução, o que permite o acompanhamento do *Sprint Burndown Chart*:

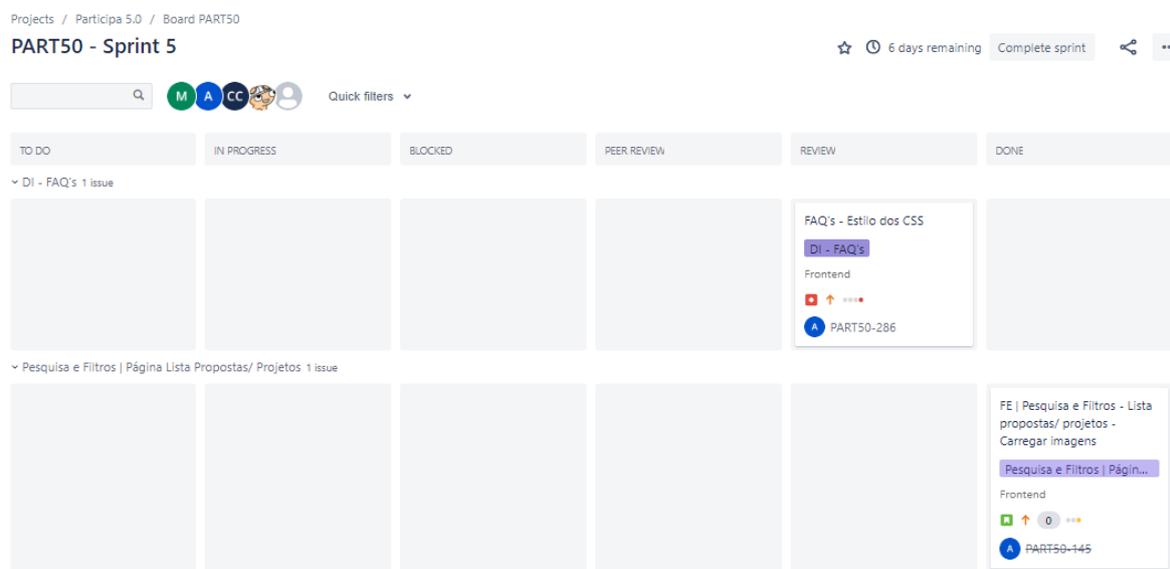


Figura 4 – Exemplo de Board de atualização de Sprint Backlog

Sprint Burndown Chart

O *Sprint Burndown Chart* permite acompanhar o progresso do trabalho realizado pela Equipe durante o *sprint*, verificando qual o trabalho que ainda não foi realizado e qual a probabilidade de conseguir cumprir com os compromissos assumidos e atingir os objetivos definidos inicialmente para o *sprint*.



Figura 5 – Exemplo de Sprint Burndown Chart

3.4. Reuniões

A metodologia Scrum pressupõe a realização das seguintes reuniões ao longo do projeto:

- *Sprint Planning*;
- *Daily Meeting*;
- *Sprint Review*;
- *Sprint Retrospective*.

3.4.1. *Sprint Planning*

A reunião de *Sprint Planning* tem como objetivo a definição dos compromissos e objetivos do *sprint*, sendo definida pelas seguintes características:

- Na reunião de *Sprint Planning* estão presentes o *Product Owner*, *ScrumMaster* e a Equipa;
- Esta reunião ocorre no início de cada *sprint* e tem a duração máxima de 4 horas;
- Nesta reunião, o *Product Owner* apresenta à Equipa os itens prioritários do *Product Backlog* (apenas são apresentados itens que já foram alvo do processo de *grooming* ou refinamento);
- A Equipa esclarece todas as dúvidas que tenha sobre os itens apresentados pelo *Product Owner*;
- A Equipa divide os itens apresentados pelo *Product Owner* em tarefas, estima detalhadamente cada tarefa, identifica as tarefas que considera poder cumprir durante o *sprint* e cria o *Sprint Backlog*.

Da reunião de *Sprint Planning*, resultam os compromissos assumidos pela Equipa para cumprir a execução do *Sprint Backlog*.

3.4.2. *Daily Meeting*

A reunião *Daily Meeting* tem como objetivo a partilha do progresso individual na execução do *sprint*, sendo definida pelas seguintes características:

- A *Daily Meeting* é uma reunião diária com a duração máxima de 15 minutos e onde estão presentes o *ScrumMaster* e a Equipa, devendo realizar-se todos os dias à mesma hora e no mesmo local;

- Cada membro da Equipa tem um máximo de 2 / 3 minutos para partilhar o progresso do seu trabalho, respondendo às seguintes questões:
 - Em que trabalhou desde a última reunião (nas últimas 24 horas)?
 - Em que vai trabalhar até à próxima reunião (nas próximas 24 horas)?
 - Tem algum obstáculo / impedimento à realização das suas tarefas e ao cumprimento dos objetivos do *sprint*?
- No caso de partilha de obstáculos / impedimentos, apenas devem ser identificadas as situações, não deverão ser detalhadas nem discutida a respetiva solução nesta reunião. Se necessário, marcar uma reunião adicional para debater este assunto.

Durante a reunião *Daily Meeting*, o *ScrumMaster* tem um papel fundamental:

- Garantir que a Equipa cumpre as regras das reuniões diárias;
- Garantir que a Equipa mantém o foco e não se desvia dos objetivos da reunião diária;
- Identificar os obstáculos / impedimentos referidos pela Equipa, tendo ele próprio a responsabilidade de os endereçar e resolver.

3.4.3. *Sprint Review*

A reunião de *Sprint Review* tem como objetivo a revisão das funcionalidades entregues no presente *sprint*, sendo definida pelas seguintes características:

- Na reunião de *Sprint Review* estão presentes o *Product Owner*, *ScrumMaster*, Equipa e *Stakeholders*;
- Esta reunião ocorre no fim do *sprint* e tem a duração máxima de 4 horas, sendo previsível a duração de 1 hora por cada semana da duração do *sprint*;
- Nesta reunião, a Equipa apresenta as funcionalidades implementadas durante o *sprint*, a fim de que o *Product Owner* e os *Stakeholders* possam validar se as funcionalidades entregues foram implementadas corretamente.

A reunião de *Sprint Review* permite assim rever os seguintes tópicos em relação ao *sprint* atual:

- Os objetivos definidos previamente;
- Os compromissos assumidos pela Equipa;

- O trabalho concluído;
- O trabalho por concluir;
- As decisões tomadas durante o *sprint*;
- O trabalho realizado e entregue;
- As questões / dúvidas que subsistem na Equipa.

O *feedback* obtido, sobre as funcionalidades implementadas, ajuda a definir as tarefas seguintes da Equipa (definidas na próxima reunião de *Sprint Planning*), podendo ter de considerar:

- Não conformidades ou pedidos de alteração que foram identificados;
- Repriorização do *Product Backlog*;
- Alterações à composição da equipa.

3.4.4. *Sprint Retrospective*

A reunião de *Sprint Retrospective* tem como objetivo a realização de uma retrospectiva sobre o presente *sprint* com o objetivo de melhorar o processo, sendo definida pelas seguintes características:

- Na reunião de *Sprint Retrospective* estão presentes o *Product Owner*, *ScrumMaster* e a Equipa, sendo dinamizada pelo *ScrumMaster*;
- Esta reunião ocorre após a reunião de *Sprint Review* e tem a duração máxima de 1:30 horas;
- Nesta reunião, a Equipa responde às seguintes questões:
 - O que correu bem neste *sprint*?
 - O que não correu bem neste *sprint*?
 - Que obstáculos / impedimentos foram prejudiciais ao desempenho da equipa;
 - Que ferramentas, processos ou práticas podem ser utilizados para melhorar o desempenho da equipa no próximo *sprint*?

3.5. Framework

Todos os conceitos descritos nos capítulos anteriores, são trabalhados e devem ser considerados em conjunto no que concerne à aplicação da metodologia Scrum e respetiva *framework*:

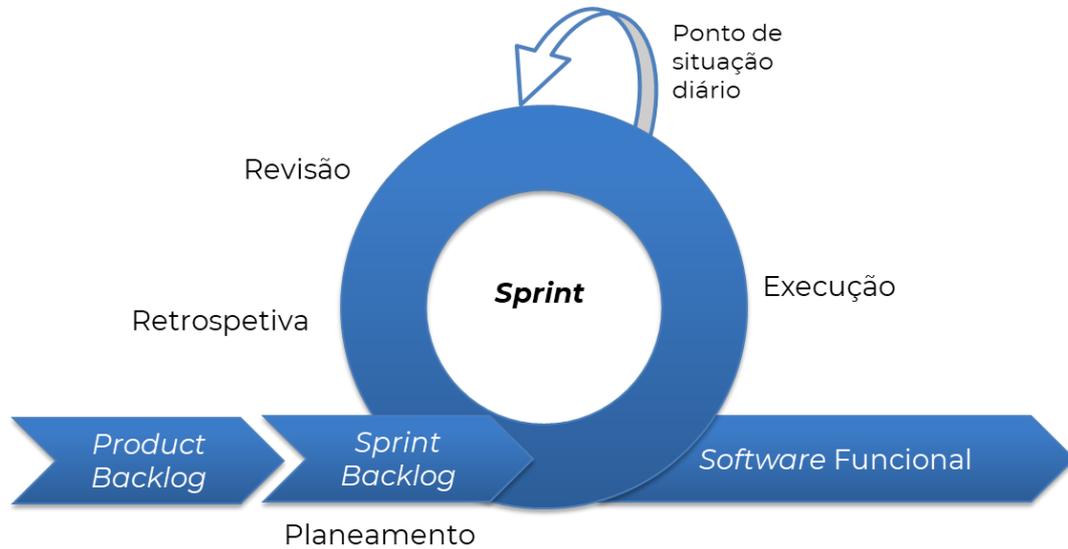


Figura 6 – Framework da metodologia Scrum

Considerando uma perspetiva temporal, a figura seguinte descreve os momentos do *sprint* em que deverão ser realizadas as reuniões que fazem parte da metodologia, bem como as respetivas fases do processo que lhes correspondem (considerando como exemplo um *sprint* de duas semanas de duração):

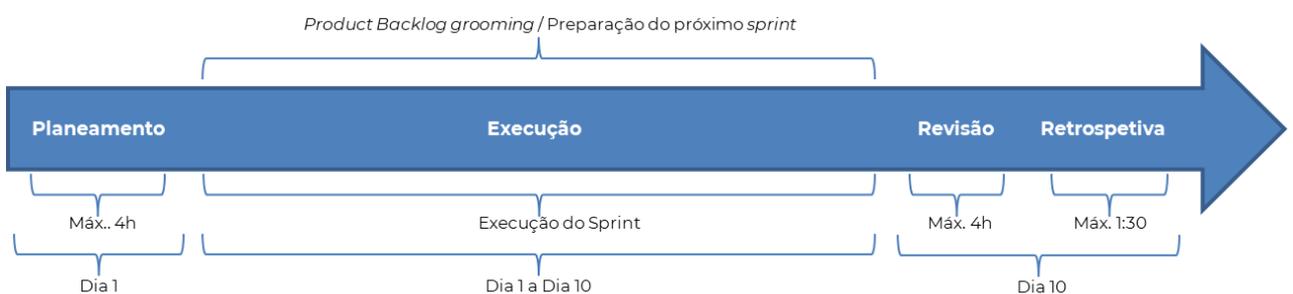


Figura 7 – Reuniões e fases do processo na metodologia Scrum

3.6. Começar...

A fim de iniciar a aplicação da metodologia Scrum na organização / entidade, deverá ser garantida a execução dos seguintes passos (resultantes da aplicação da respetiva *framework*):

- Identificar um *Sponsor* com senioridade na organização / entidade;
- Identificar o responsável pela implementação da metodologia na organização / entidade;
- Decidir e comunicar em que data se irá iniciar a aplicação da metodologia;
- Já em contexto de projeto, decidir quem irá desempenhar cada um dos papéis envolvidos na metodologia;
- Criar um *Product Backlog* priorizado e estimar todos os itens do mesmo;
- Definir a Equipa de projeto e remover todos os obstáculos / impedimentos à sua alocação;
- Agendar no calendário todas as reuniões envolvidas na metodologia;
- Criar um *Sprint Backlog* e começar a trabalhar;
- Garantir que a reunião diária acontece todos os dias à mesma hora e no mesmo local;
- Garantir que existe um entregável com valor para o negócio no final de cada *sprint*;
- Analisar os gráficos de desempenho da Equipa;
- Apresentar e validar o resultado de cada *sprint*;
- Analisar os resultados obtidos e tomar as ações corretivas necessárias;
- Repetir, melhorando os resultados de cada *sprint*;
- Resistir ao stress de adaptação a uma nova metodologia;
- Manter a disciplina e os resultados rapidamente aparecerão.

3.6.1. Para saber mais...

Os seguintes *links* poderão ser úteis na procura de um maior conhecimento sobre a metodologia Scrum:

- <https://www.agilealliance.org/>
- <https://www.scrumalliance.org/>
- <https://www.scruminc.com/>
- <https://www.scrum.org/>
- <https://www.mitchlacey.com/>