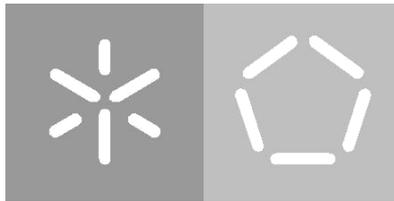


**Universidade do Minho**  
Escola de Engenharia  
Departamento de Informática

Patrícia Sofia Duarte Rocha

**Como Compatibilizar Paradigmas  
Estruturados da Gestão de  
Projetos com Metodologias Ágeis**

Agosto 2017



**Universidade do Minho**  
Escola de Engenharia  
Departamento de Informática

Patrícia Sofia Duarte Rocha

**Como Compatibilizar Paradigmas  
Estruturados da Gestão de  
Projetos com Metodologias Ágeis**

Dissertação de Mestrado  
Mestrado Integrado em Engenharia Informática

Dissertação realizada sob a orientação de  
**João Varajão**

Agosto 2017

---

## AGRADECIMENTOS

---

A todas as pessoas que me auxiliaram, motivaram e encorajaram possibilitando a conclusão desta dissertação, o meu sincero agradecimento.

Agradeço ao meu orientador e supervisor na empresa, João Varajão e Pedro Engrácia, respetivamente, pelo auxílio e orientação. Em particular ao professor Varajão, pela referência do tema, pelos conselhos e pela disponibilidade ao longo desta fase árdua de trabalho.

À eSPap pela oportunidade apresentada, que permitiu desafiar e expandir o meu conhecimento numa área que considero cativante e com impacto no sucesso dos projetos.

Um agradecimento especial à minha família pela força e paciência. Às companhias habituais que sempre me apoiam um obrigado cheio de carinho e sentimento. Se cheguei aqui hoje foi por vos ter ao meu lado, muito obrigada...

---

## RESUMO

---

As organizações dispõem atualmente de standards e de guias de boas práticas de gestão de projetos bem estruturados e com ampla adoção. Tal é o caso, por exemplo, da ISO 21500 e do PMBoK 5. Não obstante, no contexto dos projetos de tecnologias e sistemas de informação realizados com recurso a metodologias ágeis, surgem novos desafios relacionados com a compatibilização das práticas ágeis com os paradigmas estruturados da gestão de projetos. A presente dissertação está focada em procurar respostas para a questão de investigação “Como compatibilizar paradigmas estruturados da gestão de projetos com metodologias ágeis”, através de uma abordagem metodológica baseada na *Design Science Research* (DSR). Como resultado do trabalho realizado, apresenta-se um novo método de gestão de projetos desenvolvido com base no PMBoK e no *Scrum*. Este método faz a conciliação das metodologias de gestão de projetos estruturadas com metodologias ágeis.

**Área de Aplicação:** Gestão de Processo de Software.

**Palavras-Chave:** Gestão de Projetos, Método, Metodologia, Desenvolvimento de *Software*, Ágil, PMBoK, *Scrum*.

---

## ABSTRACT

---

Today organizations and companies maintain well-structured and widely adopted project management standards and practices. Such is the case of ISO 21500 and PMBoK 5. Nevertheless, when using agile methodologies within the context of projects of information systems and technologies, new challenges arise regarding the conciliation of the agile practices with structured paradigms of project management. This thesis is focused on finding answers to the research question "How to combine structured paradigms of project management with agile methodologies", through the usage of Design Science Research (DSR) methodological approach. As a result of the work carried out, a new project management method based on PMBoK and Scrum is presented. This method combines the usage of structured project management methodologies with agile methodologies.

**Application Area:** Software Project Management.

**Keywords:** Software Project Management, Method, Methodology, Software Development, Agile, PMBoK, Scrum.

---

# CONTEÚDO

---

AGRADECIMENTOS .....	i
RESUMO .....	ii
ABSTRACT .....	iii
LISTA DE FIGURAS .....	viii
LISTA DE TABELAS .....	ix
SIGLAS E ACRÓNIMOS .....	x
1. INTRODUÇÃO .....	11
1.1. ENQUADRAMENTO .....	11
1.2. MOTIVAÇÃO.....	12
1.3. OBJETIVOS .....	12
1.4. PROCESSO DE INVESTIGAÇÃO .....	12
1.5. ESTRUTURA DA DISSERTAÇÃO .....	13
2. ESTADO DA ARTE .....	15
2.1. GESTÃO DE PROJETOS .....	15
2.2. METODOLOGIAS E STANDARDS DA GESTÃO DE PROJETOS.....	17
2.2.1. <i>PMBok</i> .....	17
2.2.2. <i>PRINCE2</i> .....	19
2.2.3. <i>APMBok</i> .....	22
2.2.4. <i>P2M</i> .....	23
2.2.5. <i>ICB</i> .....	25
2.2.6. <i>ISO 21500</i> .....	27
2.3. METODOLOGIAS ÁGEIS.....	28
2.3.1. <i>Scrum</i> .....	29
2.3.2. <i>Kanban</i> .....	31
2.3.3. <i>Extreme Programming (XP)</i> .....	32
2.3.4. <i>Dynamic Systems Development Method (DSDM)</i> .....	34
2.3.5. <i>Lean Development</i> .....	36
2.4. METODOLOGIAS DE GESTÃO DE PROJETOS ARTICULADAS COM METODOLOGIAS ÁGEIS .....	37

3. NOVO MÉTODO DE GESTÃO DE PROJETOS .....	39
3.1. FUNDAMENTOS DO MÉTODO .....	39
3.2. DESCRIÇÃO GLOBAL DO MÉTODO.....	40
3.3. PROCESSOS DO MÉTODO.....	42
3.4. PAPÉIS E RESPONSABILIDADES.....	46
3.4.1. <i>Gestor de Projeto</i> .....	46
3.4.2. <i>Scrum Master</i> .....	47
3.4.3. <i>Product Owner</i> .....	48
3.4.4. <i>Equipa de Desenvolvimento (elementos)</i> .....	48
4. FASE DE INICIAÇÃO .....	50
4.1. DEFINIR O STATEMENT OF WORK .....	52
4.2. DESENVOLVER O BUSINESS CASE .....	53
4.3. ELABORAR O PROJECT CHARTER .....	55
4.3.1. <i>Justificar a Existência do Projeto</i> .....	56
4.3.2. <i>Definir os Objetivos de Alto Nível</i> .....	57
4.3.3. <i>Identificar os Requisitos de Alto Nível</i> .....	57
4.3.4. <i>Definir os Deliverables</i> .....	57
4.3.5. <i>Identificar as Restrições e os Pressupostos</i> .....	58
4.3.6. <i>Identificar as Milestones</i> .....	58
4.3.7. <i>Construir a Timeline</i> .....	58
4.3.8. <i>Identificar os Recursos Necessários</i> .....	59
4.3.9. <i>Estimar o Orçamento</i> .....	59
4.3.10. <i>Esquematizar a Organização</i> .....	59
4.3.11. <i>Identificar os Riscos</i> .....	60
4.3.12. <i>Categorizar os Riscos</i> .....	60
4.3.13. <i>Elaborar o Plano de Resposta aos Riscos</i> .....	61
4.3.14. <i>Definir os Critérios de Avaliação</i> .....	61
4.3.15. <i>Identificar os Fatores de Sucesso</i> .....	61
4.3.16. <i>Identificar e Caracterizar os Stakeholders</i> .....	62
4.3.17. <i>Identificar Preliminarmente as Aquisições</i> .....	63
5. FASE DE PLANEAMENTO .....	64
5.1. ELABORAR O PLANO DE PROJETO.....	65
5.1.1. <i>Identificar os Requisitos</i> .....	67
5.1.2. <i>Definir o Âmbito</i> .....	69
5.1.3. <i>Detalhar as Restrições e os Pressupostos</i> .....	70
5.1.4. <i>Definir o Não-âmbito</i> .....	70

5.1.5. Identificar os Deliverables.....	71
5.1.6. Definir as Atividades.....	71
5.1.7. Sequenciar as Atividades.....	72
5.1.8. Estimar os Recursos das Atividades.....	73
5.1.9. Estimar as Durações.....	74
5.1.10. Construir o Cronograma.....	75
5.1.11. Estimar as Reservas.....	78
5.1.12. Determinar o Orçamento.....	78
5.1.13. Desenvolver o Plano de Recursos Humanos.....	79
5.1.14. Definir a Organização.....	80
5.1.15. Criar o Plano de Gestão de Risco.....	81
5.1.16. Planejar Medidas de Qualidade.....	82
5.1.17. Definir o Plano de Gestão do Sucesso.....	83
5.1.18. Definir a Gestão dos Stakeholders.....	83
5.1.19. Desenvolver o Plano de Gestão da Comunicação.....	84
5.1.20. Planejar Aquisições.....	85
5.2. DEFINIR O PRODUCT BACKLOG.....	86
5.2.1. Identificar as User Stories.....	88
5.2.2. Validar as User Stories.....	89
5.2.3. Priorizar as User Stories.....	89
5.2.4. Aceitação do Product Backlog.....	92
6. FASE DE EXECUÇÃO.....	93
6.1. PLANEAR A SPRINT.....	95
6.1.1. Grooming Meeting.....	95
6.1.2. Sprint Planning Meeting.....	99
6.2. EXECUTAR A SPRINT.....	103
6.3. CONCLUIR A SPRINT.....	107
6.3.1. Sprint Review.....	108
6.3.2. Sprint Retrospective.....	109
7. FASE DE MONITORIZAÇÃO E CONTROLO.....	111
7.1. CONTROLAR O ÂMBITO.....	112
7.2. CONTROLAR O TEMPO.....	113
7.3. MONITORIZAR O ORÇAMENTO.....	114
7.4. MONITORIZAR A EQUIPA DO PROJETO.....	116
7.5. MONITORIZAR E CONTROLAR OS RISCOS.....	117
7.6. ASSEGURAR A QUALIDADE.....	118
7.7. CONTROLAR OS FATORES DE SUCESSO.....	119

7.8. GERIR AS EXPECTATIVAS DOS STAKEHOLDERS .....	119
7.9. CONTROLAR AS COMUNICAÇÕES .....	120
7.10. LEVAR A CABO AS AQUISIÇÕES .....	121
7.11. MONITORIZAR AS AQUISIÇÕES .....	122
8. FASE DE CONCLUSÃO.....	124
8.1. FECHO DE CONTRATOS E AQUISIÇÕES.....	124
8.2. CONSOLIDAÇÃO DAS LIÇÕES APRENDIDAS .....	124
8.2.1. <i>Closing Meeting</i> .....	124
8.2.2. <i>Recolha de Opiniões dos Stakeholders</i> .....	125
9. CONCLUSÃO .....	126
MATERIAL DE SUPORTE .....	128
I. GLOSSÁRIO .....	128
II. REFERÊNCIAS/BIBLIOGRAFIA.....	131

---

## LISTA DE FIGURAS

---

Figura 1 - Abordagem Metodológica .....	13
Figura 2 - Ciclo de vida de um projeto .....	16
Figura 3 - Project Management Tower .....	25
Figura 4 - Eye of Competence .....	26
Figura 5 - Scrum .....	30
Figura 6 - Kanban Board .....	31
Figura 7 - Ciclo de Vida de um Projeto DSDM .....	35
Figura 8 - Estrutura de um Projeto .....	41
Figura 9 - Esquema da Fase de Iniciação .....	51
Figura 10 - Esquema da Fase de Planeamento .....	65
Figura 11 - Exemplo de WBS.....	70
Figura 12 - Exemplo de uma PBS .....	71
Figura 13 - Notação Gráfica da Nomenclatura de Dependências.....	73
Figura 14 - Exemplo de ReBS .....	74
Figura 15 - Diagrama de Gantt.....	76
Figura 16 - Nomenclatura do Diagrama de Precedência .....	76
Figura 17 - Construção do Diagrama de Precedências .....	77
Figura 18 - Gráfico do Método de Kano .....	91
Figura 19 - Esquema da Fase de Execução .....	94
Figura 23 - Exemplo de uma Task Board.....	104
Figura 24 - Sprint Burndown Chart, Esforço Ideal .....	105
Figura 25 - Sprint Burndown Chart .....	106

---

## LISTA DE TABELAS

---

Tabela 1 - Fatores de Sucesso dos Projetos .....	16
Tabela 2 - Desafios nos Projetos .....	17
Tabela 3 - Áreas de Conhecimento Abordadas na Construção do Project Charter .....	56
Tabela 4 - Áreas de Conhecimento Abordadas na Construção do Plano de Projeto .....	66
Tabela 5 - Áreas de Conhecimento Abordadas na Construção do Product Backlog .....	87
Tabela 6 - Áreas de Conhecimento Abordadas na Grooming Meeting .....	96
Tabela 7 - Áreas de Conhecimento Abordadas na Sprint Planning Meeting .....	100
Tabela 8 - Áreas de Conhecimento Abordadas na Sprint Execution .....	103
Tabela 9 - Áreas de Conhecimento Abordadas no Sprint Closing .....	108

---

## SIGLAS E ACRÓNIMOS

---

APM	Association for Project Management.
APMBoK	APM Body of Knowledge.
DSDM	Dynamic Systems Development Method.
ENAA	Engineering Advancement Association of Japan.
eSPap	Entidade de Serviços Partilhados da Administração Pública.
EVM	Earned Value Management.
GP	Gestão de Projetos.
ICB	Individual Competence Baseline.
IPMA	International Project Management Association.
ISO	International Standardization Organization.
P2M	Project & Program Management.
PMBok	Project Management Body of Knowledge.
PMI	Project Management Institute.
PMO	Project Management Office.
PO	Product Owner.
PRINCE	Projects in Controlled Environments.
RAD	Rapid Application Development.
SaaS	Software as a Service.
SM	Scrum Master.
SP PMO	Serviços Partilhados de Project Management Office.

---

## INTRODUÇÃO

---

Neste capítulo é enquadrado o tema, enunciada a motivação e objetivos da dissertação, sucintamente descrito o processo de investigação e, por último, apresentada a estrutura seguida do documento.

### 1.1. Enquadramento

Este tema foi proposto pela Entidade de Serviços Partilhados da Administração Pública (eSPap) em conjunto com a Universidade do Minho.

A missão da eSPap é assegurar o desenvolvimento e a prestação de serviços partilhados no âmbito da Administração Pública, dando, assim, suporte à gestão documental e de processos a outros organismos.

Dentro da sua oferta de serviços, a eSPap disponibiliza Serviços Partilhados de Project Management Office (SP PMO) para o Setor Público, alicerçados tecnologicamente numa plataforma transversal de suporte à gestão estratégica e de projetos, disponibilizada em regime de SaaS (*Software as a Service*), visando o alinhamento estratégico dos projetos, a otimização da sua execução e a maximização dos respetivos benefícios. Os principais drivers do serviço são (eSPap 2016): Definição e divulgação de metodologias e de boas práticas; Gestão integrada do portefólio de programas e projetos, com manutenção de toda a documentação relacionada; Articulação com as entidades envolvidas, tendo em vista a coordenação da execução dos projetos em curso; Monitorização da execução com foco no cumprimento do custo, tempo, âmbito e qualidade e na maximização dos benefícios; Otimização da afetação de recursos humanos, financeiros, e tecnológicos; Produção de indicadores de desempenho do portefólio dos projetos da Organização; Identificação atempada de riscos e antevisão dos desvios de execução.

É no contexto do SP PMO que se enquadra o presente trabalho.

## 1.2. Motivação

O SP PMO dispõe atualmente de *standards* e de guias de boas práticas da gestão de projetos, mas a sua utilização conjunta com metodologias ágeis nem sempre é óbvia e facilmente articulada. Surge, assim, a necessidade de se desenvolver um novo método de gestão de projetos referencial que auxilie a utilização conjunta das boas práticas estruturadas da gestão de projetos, com metodologias ágeis.

## 1.3. Objetivos

A presente dissertação está focada em procurar respostas para a questão de investigação “Como compatibilizar paradigmas estruturados da gestão de projetos com metodologias ágeis”, tendo como objetivo principal o desenvolvimento de um método que auxilie a utilização conjunta das boas práticas estruturadas da gestão de projetos, com metodologias ágeis.

Identificam-se como objetivos de suporte:

- Caracterizar o estado da arte relativamente a metodologias estruturadas da gestão de projeto;
- Caracterizar o estado da arte relativamente a metodologias ágeis;
- Identificar cenários de articulação das metodologias estruturadas da gestão de projetos, com metodologias ágeis;
- Propor um método de gestão de projetos que possibilite a articulação das metodologias estruturadas da gestão de projetos, com metodologias ágeis.

## 1.4. Processo de Investigação

No desenvolvimento deste trabalho foi seguida a metodologia *Design Science Research* (DSR) [1], conforme a Figura 1.

O processo iniciou-se com a identificação pela eSPap da necessidade de compatibilização de metodologias estruturadas de gestão de projetos com metodologias ágeis, o que correspondeu à “Consciência do Problema”. Foi apresentado e contextualizado o problema, bem como analisado e caracterizado o estado da arte correspondente. De seguida foi feita

a proposta de solução que, neste caso, correspondeu a um novo método de gestão de projetos para compatibilização de metodologias. Após a sua sugestão, foi iniciado o seu desenvolvimento, tendo sido o método caracterizado e detalhado. A fase de avaliação do referencial, dado o tempo limitado para o desenvolvimento desta dissertação, foi orientada para o estudo da adequação da proposta. O processo terminou com a conclusão do trabalho realizado, tendo sido apresentados os resultados obtidos.

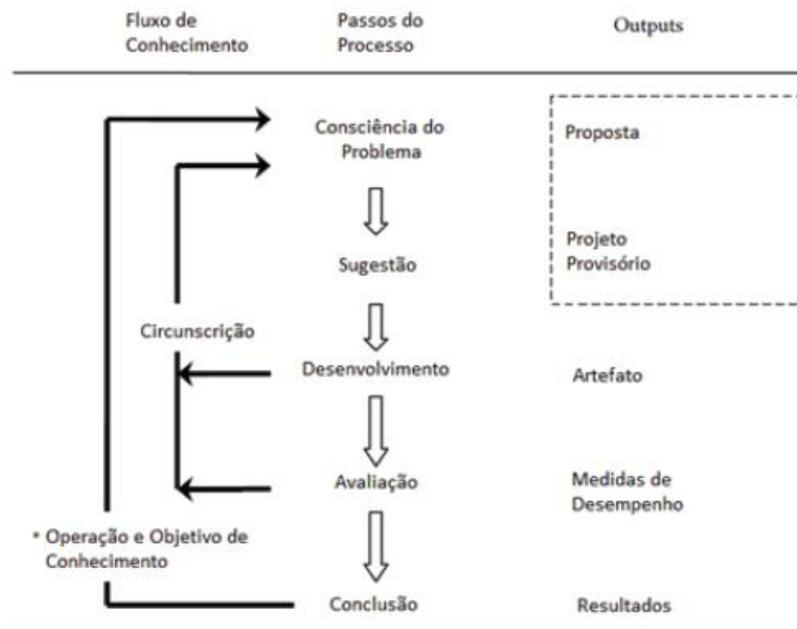


Figura 1 - Abordagem Metodológica  
Adaptado de [1]

## 1.5. Estrutura da Dissertação

No primeiro capítulo, introdutório, é apresentado o enquadramento, a motivação, os objetivos e o processo de investigação associado ao desenvolvimento desta dissertação e, por último, é detalhada a sua estrutura.

No segundo capítulo é apresentado o estado da arte dividido em três partes: gestão de projetos (conceitos fundamentais, importância para as organizações, principais metodologias e *standards* de gestão de projetos utilizados atualmente); metodologias de desenvolvimento ágeis (conceitos e caracterização das metodologias principais); por último, é abordado o conceito de metodologias de gestão de projetos conciliadas com metodologias ágeis de desenvolvimento.

O terceiro capítulo apresenta os fundamentos utilizados para a construção do novo método de gestão de projetos, a descrição global do método, os respectivos processos gerais e os papéis e responsabilidades associados. Do capítulo quatro ao capítulo oito são apresentadas as fases que constituem o método proposto, detalhando os processos envolvidos, ou seja, descrevendo a sua implementação, as várias técnicas e processos envolvidos, os fatores a considerar e os artefactos que deverão ser produzidos.

A dissertação termina no capítulo nove onde são expostas as conclusões que sintetizam o trabalho realizado, são abordadas algumas limitações relevantes e, por último, é apresentado o trabalho que se pretende realizar futuramente.

---

## ESTADO DA ARTE

---

Neste capítulo é apresentado e caracterizado o estado da arte dividido em três partes: gestão de projetos, metodologias de desenvolvimento ágeis e, por último, metodologias de gestão de projetos conciliadas com metodologias ágeis de desenvolvimento.

### 2.1. Gestão de Projetos

A Gestão de Projetos é atualmente reconhecida e aplicada em uma grande diversidade de áreas. No entanto, neste documento o foco encontra-se na Gestão de Projetos aplicada ao desenvolvimento de *software*.

Em primeiro lugar é preciso clarificar alguns conceitos fundamentais para a compreensão desta dissertação. Note-se que estas definições são apresentadas no PMBoK [2]:

- Projeto: esforço temporário empreendido para criar um produto, serviço ou resultado único. Note-se que temporário implica existir um início e fim bem definidos;
- Gestão de Projetos: consiste na utilização de conhecimentos, técnicas, metodologias e ferramentas nas fases de um projeto em prol do sucesso do próprio, ou seja, para atingir o seu objetivo;
- Gestor de Projeto: responsável pela gestão de um ou mais projetos e pelo cumprimento dos seus objetivos.

O PMBoK identifica a seguinte estrutura típica de um projeto [2]:

- Iniciação – fase de abertura do projeto, em que é estudada a sua viabilidade e definido um plano a um nível mais genérico;
- Planeamento – fase em que são detalhados todos os aspetos significativos e importantes para o projeto, sendo produzido um plano de projeto;

- Execução – fase de produção dos *deliverables*, tendo por guia o plano construído na fase anterior;
- Conclusão – fecho do projeto, terminando-se contratos e, quando necessário, apresentados relatórios de conclusão, lições aprendidas, etc.



Figura 2 - Ciclo de vida de um projeto  
Fonte: PMBoK [2]

Existe ainda o Controlo, que feito transversalmente no projeto, inicia-se, normalmente, a partir do planeamento para perceber se o projeto está de acordo com o plano estipulado.

A Gestão de Projetos está diretamente ligada ao sucesso dos projetos. A Tabela 1 e Tabela 2, presentes no *Chaos Report* [3, 4] do *Standish Group*, apresentam os principais fatores de sucesso e os principais desafios de um projeto, respetivamente.

Fatores de Sucesso	Respostas
1. <b>Envolvimento dos utilizadores</b>	15.9 %
2. <b>Apoio da gestão executiva</b>	13.9 %
3. <b>Requisitos claros</b>	13.0 %
4. <b>Bom planeamento</b>	9.5 %
5. <b>Expectativas realistas</b>	8.2 %
6. <b>Milestones simples</b>	7.7 %
7. <b>Funcionários competentes</b>	7.2 %
8. <b>Responsabilidades claras e bem definidas</b>	5.3 %
9. <b>Visão e objetivos claros</b>	2.9 %
10. <b>Funcionários empenhados e concentrados</b>	2.4 %
<b>Outros</b>	13.9 %

Tabela 1 - Fatores de Sucesso dos Projetos  
Fonte: Chaos Report [3]

Desafios	Respostas
1. Falta de feedback dos utilizadores	12.8 %
2. Requisitos incompletos e mal especificados	12.3 %
3. Alterações dos requisitos	11.8 %
4. Falta de suporte por parte da gestão executiva	7.5 %
5. Tecnologia inadequada	7.0 %
6. Falta de recursos	6.4 %
7. Expectativas irrealistas	5.9 %
8. Objetivos pouco claros	5.3 %
9. Prazos irrealistas	4.3 %
10. Novas tecnologias	3.7 %
<b>Outros</b>	<b>23.0 %</b>

Tabela 2 - Desafios nos Projetos  
Fonte: Chaos Report [3]

Tendo em conta que a Gestão de Projetos estipula medidas para quase todos os fatores apresentados (âmbito, custo, *stakeholders*, tempo, etc.), pode ser concluído que a gestão de projetos é um elemento essencial para o sucesso dos projetos.

## 2.2. Metodologias e Standards da Gestão de Projetos

Nesta secção são abordados alguns *standards* e metodologias de gestão de projeto mais comumente utilizados atualmente: PMBoK, PRINCE2, APMBok, P2M, ICB e ISO 21500.

### 2.2.1. PMBoK

O *Project Management Body of Knowledge* (PMBoK) é um referencial de Gestão de Projetos mundialmente reconhecido e utilizado, havendo mais de seiscentos mil profissionais certificados, em mais de 185 países [5]. Proposto pelo *Project Management Institute* (PMI), contém um conjunto de *standards*, *guidelines* e regras sobre gestão de projetos.

O PMBoK é estruturado em áreas de conhecimento da gestão de projetos, identificando dez áreas:

1. Integração – conjuga os processos e atividades das restantes áreas de conhecimento. A existência desta área reforça que todas as outras devem ser agregadas para a obtenção de um resultado positivo e consistente, e não ser desenvolvidas e estudadas individualmente. Esta área é responsável pela coesão e

coerência das restantes. Podemos considerar que esta é a área mais genérica da gestão de projetos, definindo os procedimentos e atividades a utilizar na gestão do projeto;

2. Âmbito – define a contextualização do projeto, incluindo a sua necessidade, impacto e objetivos, tudo o que deve ser produzido no projeto (*deliverables*) e o caderno de encargos a cumprir ao longo do projeto. Procura definir o que faz ou não parte do âmbito do projeto e quais serão os *outputs* do mesmo;
3. Tempo – estrutura e define as atividades do projeto, incluindo como se sequenciam, e estima as suas durações para formar o cronograma do projeto. Define ainda os recursos necessários para realizar as atividades;
4. Custo – determina o orçamento do projeto a partir dos diversos recursos necessários para executar as atividades definidas e necessárias para o sucesso do projeto;
5. Recursos Humanos – define a equipa do projeto, a sua hierarquia, e as responsabilidades dos vários elementos da equipa, bem como necessidades de formação e regras de trabalho, como por exemplo, horários ou definição de vestuário proibido/apropriado;
6. Risco – identifica os riscos inerentes ao projeto, ameaças e oportunidades, analisa os riscos de forma quantitativa e qualitativa, identificando os riscos que são significativos e, em relação a estes, define um plano de resposta de forma a explorar as oportunidades e minimizar as ameaças;
7. Qualidade – procura assegurar que o produto proporciona satisfação a quem o utiliza, definindo *standards* que deverão ser cumpridos e os testes que serão usados para a verificação destes *standards*;
8. *Stakeholders* – identifica pessoas, grupos e/ou organizações que possam ter impacto no projeto ou ser afetadas pelo projeto e desenvolve mecanismos e estratégias para lidar com estes elementos;
9. Comunicação – define que entidades necessitam de que tipo de informação e como é que a comunicação vai ser estabelecida;
10. Contratações – define, executa, administra e encerra todas as contratações necessárias para o desenvolvimento do projeto.

Para cada uma destas áreas são identificados processos. Por processo são definidos *inputs*, ferramentas e técnicas a aplicar, assim como *outputs*.

O PMBoK é um referencial aplicável a qualquer projeto, porque possui boas práticas, técnicas e regras de gestão de projetos genéricos. Citando a autora *Kathy Schwalbe* no livro *Information Technology Project Management* [6]: “*The PMBOK® Guide is a standard that describes best practices for what should be done to manage a project. A methodology describes how things should be done ...*”. Isto é, o PMBoK é um standard que responde ao “o quê?”, enquanto que uma metodologia dita o “como?”. Por esta razão, é possível conjugar as orientações do PMBoK com diferentes metodologias de gestão de projetos.

### 2.2.2. PRINCE2

O PRINCE2 [7] (*Projects in Controlled Environments*) é uma metodologia estruturada de gestão de projetos criada, tal como o PMBoK, com base nas contribuições de diversos profissionais que lidam com gestão de projetos nas suas carreiras profissionais.

Os seus princípios e estratégias são bastante genéricos, o que leva a que possam ser aplicados a virtualmente qualquer projeto. Contrariamente ao PMBoK, possui uma abordagem estruturada onde são definidos e explicados os processos, responsabilidades e mecanismos a utilizar no projeto. Isto torna o PRINCE2 uma metodologia de gestão de projetos, respondendo assim ao “como?” referenciado anteriormente.

A abordagem do PRINCE2 baseia-se em sete princípios, sete temas e sete processos.

Os princípios fornecem, segundo a página oficial do PRINCE2 [8], uma *framework* de boas práticas de gestão de projeto. São genéricos o suficiente para aplicar a qualquer tipo de projeto e totalmente compatíveis com os *standards* encontrados no PMBoK.

Princípios do PRINCE2 são os seguintes:

1. *Continued Business Justification* – um projeto tem de fazer sentido, tem de existir expectativa de retorno de valor em relação ao tempo, esforço e recursos despendidos no projeto;
2. *Learn from Experience* – decidir e aplicar estratégias é sempre mais fiável se existir material de suporte e algum conhecimento prévio de projetos semelhantes ou com fatores semelhantes. No artigo “*Top 10 Reasons Why Systems Projects Fail*” [9] são referidos três pontos-chave para o sucesso de um projeto, um dos quais é, citando, “*Solid technical leadership by someone who has successfully completed a similar project*”. Existir *background* sobre fatores semelhantes e resultados obtidos, problemas encontrados, etc. é fundamental e pode decidir o sucesso ou o fracasso de um projeto;

3. *Define Roles and Responsibilities* – é fundamental a existência de uma hierarquia estruturada e de responsabilidades bem definidas e distribuídas;
4. *Manage by Stages* – este princípio é a representação da tão célebre frase “*divide and conquer*”, atividades ou problemas complexos são geridos com maior facilidade se forem fracionados em pequenos fragmentos;
5. *Manage by Exception* – o PRINCE2 assume que os supervisores do projeto são pessoas com pouco tempo e com muito trabalho. Só quando existe um desvio fora da área de tolerância definida para qualquer um dos fatores restritivos do projeto é que devem ser notificados os supervisores. Isto impede que se recorra constantemente às entidades supervisoras;
6. *Focus on products* – definir requisitos, *deliverables* e padrões de qualidade. É preciso saber o que é esperado do produto e suas funcionalidades, para que se definam as atividades do projeto;
7. *Tailor to the environment* – adaptar a metodologia às necessidades do projeto em questão é essencial. Todos os projetos são únicos e possuem, por isso, necessidades diferentes no que se refere à sua gestão. Cabe ao gestor de projeto entender o que deverá ou não ser aplicado.

Os temas são secções mais práticas e podem ser vistos como áreas de conhecimento, onde é descrito como proceder dentro de cada área. Estes temas devem ser abordados no início do projeto e ao longo do mesmo deve ser feita a monitorização e controlo sobre cada um dos temas.

Os temas propostos no PRINCE2 são os seguintes:

1. *Business Case* – justificar o projeto determinando a razão da sua existência, o impacto que terá após ser concluído e mostrando que o projeto vale a pena ser prosseguido e é possível concluir com sucesso;
2. *Organisation* – definir os papéis e responsabilidades dos envolvidos no projeto, ou seja, definir a organização para que seja claro quem é responsável por dada tarefa. O desenvolvimento do projeto deve ser estruturado de forma a que a atribuição de tarefas seja simples;
3. *Quality* – definir os padrões de qualidade a cumprir permite o foco no desenvolvimento do projeto e verificar se o produto é adequado às necessidades;
4. *Plans* – esquematizar como os objetivos vão ser atingidos, tratando das áreas de custo, qualidade, âmbito, calendarização e possíveis benefícios;

5. *Risk* – identificar, avaliar e controlar eventos associados ao projeto, quer ameaças, quer oportunidades;
6. *Change* – aceitar mudanças em vez de as prevenir, mas apenas aceitar mudanças justificadas e que fazem sentido ser realizadas;
7. *Progress* – controlar e monitorizar o projeto permite identificar desvios do plano e reagir atempadamente a essas alterações.

Os sete processos enquadram-se no ciclo de vida de um projeto. Para cada fase definem o que deve ser feito pelo gestor de projeto, mantendo a abordagem mais prática do PRINCE2:

1. *Starting Up a Project (SU)* – justificar a existência do projeto, definir a equipa e distribuir responsabilidades, e planificar como o projeto vai ser levado a cabo;
2. *Initiating a Project (IP)* – identificar o caderno de encargos e como vão ser tratadas as áreas de tempo, custo, qualidade, âmbito, benefícios e risco do projeto;
3. *Directing a Project (DP)* – gerir a iniciação, gerir os recursos, monitorizar progresso, fornecer direções e concluir o projeto são algumas das tarefas de dirigir um projeto;
4. *Controlling a Stage (CS)* – o gestor de projeto deve monitorizar e construir relatórios sobre o progresso das atividades e corrigir os desvios ou problemas que surjam durante o desenvolvimento do projeto. O gestor de equipa é responsável pela coordenação das atividades de trabalho diárias e de intermediário para a comunicação entre equipa e gestor de projeto;
5. *Managing Product Delivery (MP)* – define a comunicação entre gestor de projeto e gestor de equipa que deve consistir em aceitar uma tarefa, facultar a execução da tarefa e realizar a entrega do resultado;
6. *Managing Stage Boundaries (SB)* – o gestor de projeto marca reuniões com a equipa para registar lições aprendidas para a próxima etapa do projeto, os supervisores do projeto vão decidir se o projeto continua ou não a cada ponto de análise (*stage gate*). Assim, é necessário planificar a etapa seguinte do projeto, atualizar o plano e o modelo de negócio e reportar aos supervisores o final de cada etapa ou de um desvio fora da tolerância estabelecida;
7. *Closing a Project (CP)* – identificar que o projeto cumpriu os objetivos e o caderno de encargos, definir que ações vão ser necessárias após a sua conclusão, libertar recursos e entregar o produto ao cliente.

De reforçar que esta metodologia é compatível com os *standards* do PMBoK, sendo que podem ser utilizados no mesmo projeto. Como já foi indicado, o PMBoK oferece *standards*

de boas práticas da gestão de projeto, já o PRINCE2 oferece uma metodologia estruturada para aplicar na gestão de projetos.

O PRINCE2 continua sem fornecer um método prático de gestão de projetos, ou seja, não fornece técnicas e planos detalhados sobre como gerir o projeto, precisamente para poder ser aplicado a projetos cuja área não está restrita a grupos específicos.

### 2.2.3. APMBoK

O APMBoK surgiu por existir a opinião de que o PMBoK não respondia a todas as necessidades de conhecimento dos gestores de projeto. O seu conteúdo está dividido em sessenta e nove tópicos, distribuídos em quatro secções [10]:

1. *Context* – esta secção encontra-se dividida em duas áreas: *governance*, que aborda os procedimentos e aspetos culturais que precisam de ser tratados para melhorar a capacidade de concluir os projetos com sucesso; e *setting*, que lida com os fatores organizacionais fora dos limites do projeto, que possam ter um impacto significativo na forma como o projeto é levado a cabo. Comparando com o PMBoK, que refere que no desenvolvimento do projeto é preciso ter estes temas em conta, mas não menciona como os gerir, o *Context* do APMBoK é mais relacionado com a gestão de programas e de portfólios sendo que se refere a áreas da organização e empresa e não aos projetos;
2. *People* – com o objetivo de promover a motivação entre os intervenientes humanos do projeto e definir técnicas de controlo e coordenação de pessoal, esta secção realça as *softskills*, modelos e processos sobre comportamentos a tomar, e define a melhor abordagem profissional em relação às entidades humanas envolvidas no projeto;
3. *Delivery* – esta secção foca áreas comuns às abordadas ao longo PMBoK: o âmbito; a calendarização; o financiamento; o risco; a qualidade e os recursos. No entanto, o PMBoK tem uma estruturação de áreas mais fracionada e mais delineada que o APMBoK. Por exemplo, considerando as aquisições, no PMBoK existe a área das aquisições especificamente vocacionada para a gestão das aquisições, no APMBoK, a explicação de como se deve proceder às aquisições encontra-se dentro da definição dos recursos. Existe uma divisão de mais baixo nível no PMBoK. Outra característica que distingue o APMBoK e o PMBoK, é que no primeiro um mesmo

tópico pode ser abordado em diferentes secções e no segundo existe uma separação limitada por áreas de conhecimento;

4. *Interfaces* – abordando áreas legais, de contabilidade e de recursos humanos, tem como objetivo que cada gestor saiba o suficiente sobre estes temas para conseguir obter um resultado positivo nos seus projetos, programas e portfólios. Esta é uma área que não está presente (com exceção dos recursos humanos) no PMBoK.

O APMBok está orientado não só para os gestores de projeto, mas também para os gestores de programas e de portfólios. Assim sendo, abrange áreas e temas que não se encontram no PMBoK, nomeadamente técnicas e *softskills* sobre como lidar com as pessoas (negociação, comunicação, etc.). Porém, é importante referir que o PMI tem guias orientados especificamente para programas e para portfólios.

O PMBoK aborda as áreas de conhecimento em secções distintas, ou seja, apresenta uma divisão clara entre as diferentes áreas. O APMBok segue outra abordagem, o seu conteúdo divide-se nas quatro secções já referidas e, em cada uma, aborda diversas áreas de conhecimento.

#### 2.2.4. P2M

Desenvolvido pela *Engineering Advancement Association of Japan* (ENAA), tem como principal objetivo assegurar a criação de valor para as empresas.

O P2M, "*A Guidebook of Project and Program Management for Enterprise Innovation*" [11] estrutura-se em quatro secções:

- *Project Management Entry* – secção na qual é apresentado o enquadramento dos profissionais da gestão de projetos na sociedade; os requisitos, responsabilidades e funções que um gestor de projeto e um gestor de programa devem deter; a história da gestão de projetos e como é que esta se aplica na sociedade; e, por último, são dadas indicações sobre uma melhor utilização do guia;
- *Project Management* – é estabelecida a *framework* de gestão de projetos que o P2M utiliza como referência, são definidas as *skills* que um gestor de projeto deve possuir e são ainda identificados detalhes e funções inerentes à gestão de projetos como, por exemplo, o esquema típico do trabalho de um gestor de projeto;
- *Program Management* – fornece conceitos inerentes à gestão de programas, e oferece uma *framework* base para a gestão de programas, incidindo nos princípios, atividades e fatores a ter em conta.

- *Project Segment Management* – nesta última secção são definidos onze segmentos ou áreas de conhecimento que estruturam os diferentes temas a ter em conta na gestão de projetos:
  1. Estratégia – visa a criação de valor para a empresa, utilizando estratégias baseadas na visão corporativa da empresa;
  2. Finanças – o objetivo é construir um esquema financeiro viável e economicamente interessante, através da combinação de várias alternativas e opções;
  3. Sistema – procura identificar questões ambíguas ou inesperadas com maior precisão, com base no modelo “*systems approach*”;
  4. Organização – visa construir e manter uma organização de projeto motivada para atingir os seus objetivos, estabelecendo flexibilidade em relação a alterações e atingindo um nível de produtividade maior;
  5. Objetivos – clarifica os objetivos do projeto e define o trabalho que tem de ser realizado para os atingir, garantindo satisfação do cliente e permitindo a priorização de objetivos;
  6. Recursos – estabelece como se vão gerir e adquirir os recursos para o desenvolvimento do projeto;
  7. Riscos – define o plano de controlo e de resposta aos riscos do projeto.
  8. Tecnologia da Informação – estabelece a estratégia de utilização de tecnologias da informação e comunicação no projeto;
  9. Relações – procura estabelecer relações mutuamente benéficas com os *stakeholders* e definir que tipo de relação é preciso manter com cada um dos *stakeholders* para que a satisfação dos mesmos seja atingida;
  10. Valor – apresenta o valor associado ao projeto. Procura identificar potenciais oportunidades e criar valor relevante a partir da experiência e conhecimento acumulado;
  11. Comunicação – estabelece a estrutura e plano de comunicações necessários para passagem de informação de forma eficiente e eficaz, proporcionando apoio à resolução de emergências ou problemas no projeto.

A Figura 3 ilustra a estruturação seguida pelo P2M, que corresponde ao que foi enunciado nesta secção.

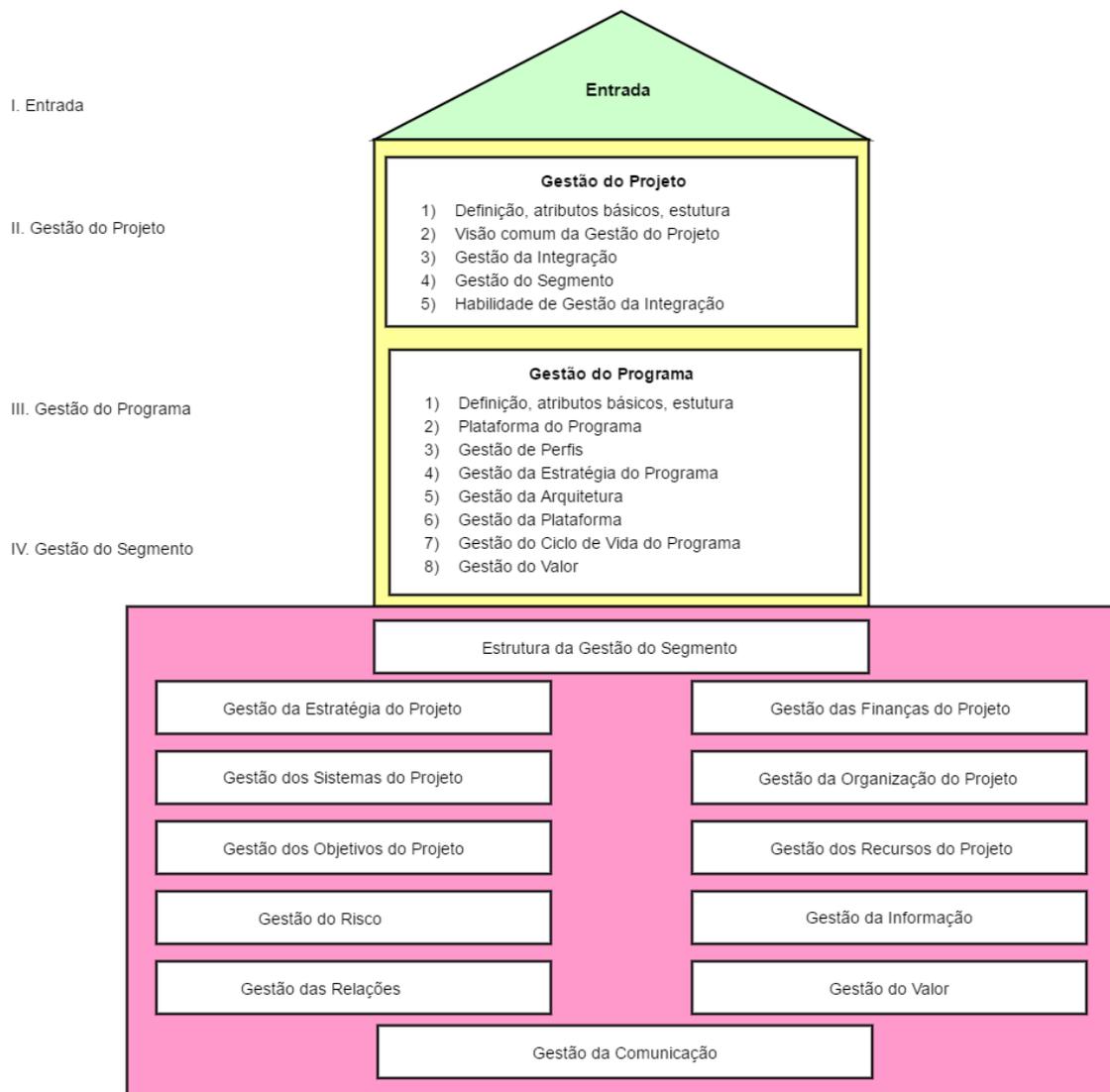


Figura 3 - *Project Management Tower*  
 Fonte: P2M [11]

### 2.2.5. ICB

O ICB, “*Individual Competence Baseline*” [12], é um guia de gestão de projetos direcionado a gestores de projeto, programa ou portfólio, que mantém uma estruturação generalista o suficiente para poder ser aplicado a projetos de várias áreas.

Na Figura 4 é apresentado o “*Eye of Competence*” definido pelo ICB, que estabelece o universo de competências necessárias para a gestão de um projeto, programa e/ou portfólio. Estas competências dividem-se em três áreas: perspectiva, pessoal e prática. Cada uma das áreas identifica um conjunto de aspetos necessários para uma gestão competente

e, em conjunto, definem o *skill set* que um gestor de projeto, programa e/ou portfólio deve apresentar:

- Perspetiva – métodos, ferramentas e técnicas de interação necessárias para que indivíduos, organizações e/ou sociedades iniciem e apoiem projetos, programas e portfólios;
- Pessoal – capacidades pessoais e de relação (*soft skills*) necessárias na participação ou liderança de um projeto, programa ou portfólio;
- Prática – métodos, ferramentas e técnicas a utilizar em projetos, programas e portfólios.



Figura 4 - *Eye of Competence*  
Fonte: ICB [12]

O ICB define vinte e nove elementos de competência, cada um deles associado a uma das áreas: cinco elementos associados à área "*perspetiva*"; dez à área "*pessoal*"; e catorze à área "*prática*". Para cada um dos elementos de competência é dada a sua definição, explicado o seu propósito, identificadas as capacidades inerentes ao mesmo, e fornecidas medidas para verificar a presença desse elemento de competência.

Note-se que a sequência de passos que o ICB apresenta é apenas (e deve ser considerada como) uma sugestão. Para além disso, não define qualquer técnica exata para a realização de cada passo, ficando ao critério do gestor de projeto escolher o método mais apropriado ao projeto que está a gerir.

### 2.2.6. ISO 21500

Estabelecida em 2012, a norma ISO 21500 [13] é generalista o suficiente para poder ser aplicada a qualquer tipo de organização e de projeto.

É estruturada da seguinte forma:

1. Âmbito – define o âmbito associado à ISO 21500, nomeadamente o porquê da sua existência, onde se aplica, quem poderá estar interessado e quem beneficia com a utilização e conhecimento deste *standard*.
2. Termos e Definições – esta secção define e identifica termos que são específicos à prática da gestão de projetos.
3. Conceitos da Gestão de Projetos – define conceitos-chave da execução da gestão de um projeto, como o conceito de projetos, operações, ambiente de projeto, etc.
4. Processos da Gestão de Projetos – define dez áreas de conhecimento de um projeto:
  - Integração;
  - *Stakeholders*;
  - Âmbito;
  - Recursos;
  - Tempo;
  - Custo;
  - Risco;
  - Qualidade;
  - Aquisições;
  - Comunicações.

A ISO 21500 segue a abordagem caracterizada pelo PMBoK, como se verifica pelos processos que define, que são compatíveis com as áreas de conhecimento definidas no PMBoK.

### 2.3. Metodologias Ágeis

Em fevereiro de 2001 foi criado o manifesto *Agile*, que definiu formalmente o avanço dos métodos ágeis no desenvolvimento de *software*. Este manifesto possui doze princípios e quatro valores [14]:

- *Individuals and Interactions over Processes and Tools* (Pessoas e Interações acima de Procedimentos e Ferramentas);
- *Working Product over Comprehensive Documentation* (Produto Funcional acima de Documentação Completa);
- *Customer Collaboration over Contract Negotiation* (Colaboração com o Cliente acima de Negociação e Contratos);
- *Responding to Change over Following a Plan* (Capacidade de Resposta a Mudança acima de Seguir um Plano).

De uma forma geral, este manifesto tenta fornecer uma alternativa ao conceito de projetos desenvolvidos sobre *documentos estáticos e necessidades inalteráveis*, defendendo que existir incerteza inicial é normal num projeto e a capacidade de adaptação às necessidades do cliente fornece uma flexibilidade essencial ao sucesso do projeto e decorrente satisfação do cliente.

A típica estrutura de uma metodologia ágil é estabelecida por iterações, que se repetem ao longo do desenvolvimento do projeto até que este esteja completo. Tipicamente inicia-se com uma fase de levantamento, seguida de uma fase de planeamento, outra de execução e uma última de avaliação e retrospeção sobre o que foi feito.

Este tipo de metodologias promove constante reavaliação ao trabalho desenvolvido, existência de feedback constante entre os intervenientes com poder e interesse no projeto, aceitação de alterações justificadas e que tragam algum tipo de valor acrescentado, comunicação e cooperação frequente entre equipa e *stakeholders*, e estabelece entregas fracionadas e, por isso, mais rápidas de produto.

Este tipo de metodologia de desenvolvimento é vantajosa em projetos com incerteza elevada e que possam envolver alterações futuras. No entanto, a aplicação de metodologias ágeis a projetos de alto risco onde existe uma necessidade de maior certeza como, por exemplo, desenvolvimento de *software* aeroespacial, regra geral pode não ser adequada [15]. Projetos de maior risco e complexidade exigem uma maior quantidade de documentação e mais detalhada. As vantagens das entregas iterativas tornam-se menos

relevantes porque existe necessidade de testes globais e extensos ao produto completo e, dada a dimensão do projeto, as equipas são normalmente de grande dimensão, o que desrespeita o conceito de pequenas equipas para promoção de comunicação.

### 2.3.1. *Scrum*

O *Scrum* representa um desenvolvimento iterativo e incremental, que fornece flexibilidade em relação a alterações relacionadas com o projeto e maior rapidez nas entregas de partes do produto ao cliente. Os elementos de uma *Scrum team* são [16]:

- *Product Owner* – responsável por fazer a ligação entre a *Scrum team* e os *stakeholders*. O *product owner* possui conhecimento do domínio do problema que é aplicado para auxiliar a restante equipa a tomar decisões mais adequadas. É este que valida o trabalho desenvolvido e decide as prioridades nas iterações, especificando as *user stories*, que podem ser vistas como os requisitos a desenvolver na iteração.
- *Scrum Master* – responsável por liderar a equipa de desenvolvimento, não interfere nas decisões de especificações de implementação, mas aconselha os elementos da sua equipa e fornece possíveis soluções. É ainda responsável por garantir que os princípios do *Scrum* são respeitados. Com conhecimento técnico do domínio da solução, deve guiar a equipa para as melhores decisões e soluções para os problemas de desenvolvimento que eventualmente surjam.
- Equipa de Desenvolvimento – responsável pela execução e implementação da solução, tem como função ser auto-organizada e independente. Assim, cada elemento decide que *user story* vai desenvolver sem que lho peçam ou atribuam.

Numa primeira fase, o *product owner* constrói o *product backlog* priorizado para o projeto. Este *backlog* define os requisitos do projeto, os quais são, a cada iteração, reavaliados e priorizados. As iterações denominam-se *sprints* e são compostas por [16]:

- *Sprint Planning* – no início de uma *sprint* é realizada uma reunião com todos os intervenientes do projeto, sendo então construído o *sprint backlog* (conjunto de *User Stories*, que corresponde ao que deve ser desenvolvido e implementado na *sprint*) tendo em conta a capacidade de execução da equipa. Isto corresponde a tirar a parte do topo do *product backlog* que, estando priorizado, corresponde a iniciar o desenvolvimento das características mais importantes do projeto naquele momento.

- *Daily Scrum* – a cada dia existe uma reunião de curta duração (15 minutos no máximo), em que cada membro da equipa explica o que tem estado a desenvolver, o que pretende fazer de seguida e se existe algum problema que o impeça de cumprir o seu objetivo.
- *Sprint Review* – reunião existente no final de uma *sprint* para rever o trabalho que foi realizado e o que ficou por realizar. Para além disto, nesta reunião é apresentado aos *stakeholders* o produto obtido após a *sprint* de forma a que o resultado atual possa ser avaliado.
- *Sprint Retrospective* – tal como a anterior é uma reunião realizada no final da *sprint* onde é feita uma reflexão sobre o *sprint* terminado, por forma a retirar lições vantajosas para as *sprints* seguintes.

A Figura 5 representa o processo enunciado nos parágrafos anteriores.

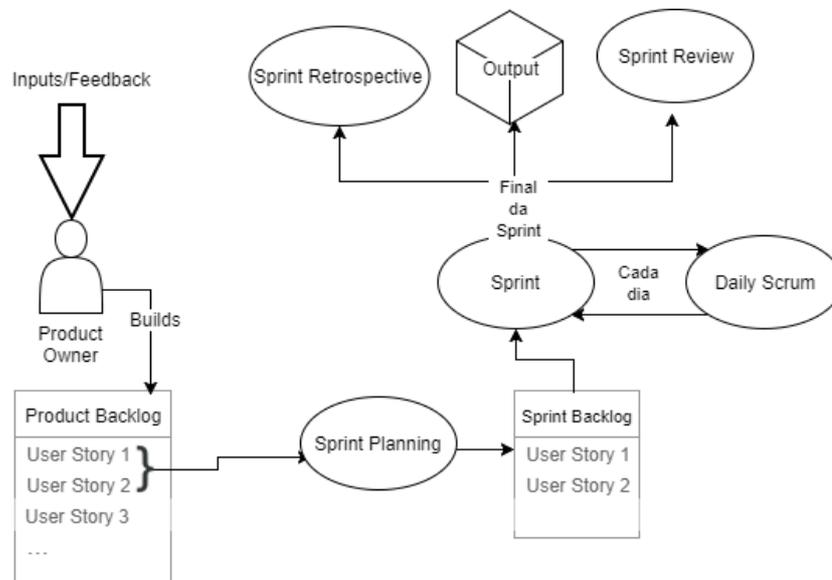


Figura 5 - Scrum

Um item só se considera pronto quando a *definição de completo* (*Definition of Done*) é cumprida. Esta definição é construída pela equipa e auxilia a verificar se uma *user story* está a ser ou não cumprida.

### 2.3.2. Kanban

A metodologia *kanban* promove a existência de um *kanban board* – ver Figura 6 – de forma a que seja visível o *workflow* do projeto. Ao contrário do *Scrum*, o *kanban* não é tão estruturado em termos de processo de desenvolvimento, ou seja, não existem restrições temporais nem iterações definidas, assim como não existe o conceito de “papéis”. No entanto, permite que sejam realizadas entregas e alterações em qualquer altura do projeto.



Figura 6 - Kanban Board

O *kanban* baseia-se nos seguintes princípios [17, 18]:

1. *Start with what you do know* – o *kanban* não define, tal como já foi referido, uma estrutura detalhada para o desenvolvimento. Assim sendo, pode ser utilizado com aquilo que a equipa já conhece, não afetando o desempenho dos elementos da equipa.
2. *Agree to pursue incremental, evolutionary change* – para promover uma introdução livre de diferenças radicais que podem afetar o desempenho da equipa, o *kanban* defende que devem ser introduzidas pequenas alterações em vez de alterações radicais que mudam todo o processo a que os trabalhadores estão acostumados.
3. *Respect the current process, roles, responsibilities and titles* – os processos, papéis e responsabilidades aplicados antes de introduzir o *kanban* possuem valor. Assim, a alteração deve ser incremental e não de forma radical, dado que tal pode trazer desconforto aos membros e afetar o seu desempenho e produtividade.
4. *Encourage acts of leadership at all levels* – a promoção de atitudes de liderança entre os membros da equipa ajuda a que cada trabalhador se sinta apto para partilhar ideias e, com esta partilha, pode ser melhorada a metodologia utilizada.

Estes princípios têm como objetivo guiar quem pretenda utilizar o *kanban* no seu processo de desenvolvimento.

Para além, disto seguem-se ainda seis princípios base [17, 18]:

1. *Visualize the workflow* – criar uma visão comum das tarefas e atividades a cumprir permite uma gestão mais eficaz por parte dos membros da equipa sobre o trabalho a realizar.
2. *Limit Work-In-Progress* – o *kanban* estabelece um sistema de desenvolvimento incremental e progressivo. Ao limitar a quantidade de trabalho em desenvolvimento facilita-se a visualização do *kanban board* e evitam-se problemas com a troca de tarefas e constante priorização.
3. *Manage Flow* – cada tarefa do *kanban board* deve ser monitorizada e, através dessa monitorização, pode definir-se a rapidez de desenvolvimento e identificar problemas no *workflow*.
4. *Make Process Policies Explicit* – é importante que todos os elementos tenham noção de como funcionam os processos na empresa. Se tal for alcançado, qualquer elemento consegue ter uma discussão sobre dado tema referente aos processos e são mais facilmente identificados problemas em relação à metodologia e processos utilizados.
5. *Improve Collaboratively, Evolve Experimentally* – através de modelos e métodos científicos, devem-se identificar os problemas da metodologia atual e criar possíveis soluções.
6. *Implement Feedback Loops* – a existência de *feedback loops* tem como objetivo comparar os resultados previstos com os obtidos e, face a esta análise, retirar conclusões sobre os processos seguidos e ajustar procedimentos.

Alguns dos princípios do *kanban*, tanto de fundação como princípios base, variam consoante a referência. Isto acontece porque existe uma agregação de princípios, ou seja, em vez de seis princípios apresentam-se cinco, onde o último é a agregação de dois outros princípios. Nesta secção optou-se por manter todos os princípios individualmente identificadas nas referências [17, 18].

### 2.3.3. *Extreme Programming* (XP)

O objetivo principal da *extreme programming* (XP) é organizar a equipa de forma a produzir mais eficazmente *software* de maior qualidade e que corresponda às expectativas dos clientes.

Esta metodologia define as seguintes entidades: *developer*, que corresponde a um programador; *customer*, que é o cliente do projeto; *manager*, responsável pelo planeamento da solução, monitorização do desenvolvimento, identificação de problemas, etc., podendo ser considerado um gestor de desenvolvimento; por último, *coach*, é equivalente a um *Scrum master* no *Scrum*, que garante que os princípios e valores da XP são respeitados, etc.

Esta metodologia define quatro atividades [19]:

1. *Coding* – estabelece que o único *deliverable* realmente importante é o produto de *software*. Para além disso, promove a comunicação por código, ou seja, transmitir um problema que exista do domínio da solução é mais fácil através da programação. Isto pode ser feito se os elementos da equipa forem todos fluentes neste tipo de linguagem. O mesmo não se aplica para questões do domínio do problema (por exemplo: comunicação com o cliente ou utilizadores sem conhecimentos de programação).
2. *Testing* – salienta a importância da existência de testes unitários e de integração para garantir a qualidade técnica da solução, e testes de qualidade e satisfação perante os *stakeholders*, para identificar se as suas expectativas e necessidades são cumpridas.
3. *Listening* – a comunicação com os clientes e utilizadores é crucial para entender as suas necessidades e formular soluções mais adequadas. Deve existir disponibilidade por parte da equipa para esclarecimento de questões técnicas aos clientes e utilizadores numa linguagem que lhes seja familiar.
4. *Designing* – as últimas atividades necessitam de uma atividade inicial de esquematização e conceção de soluções, caso contrário, durante o desenvolvimento, a probabilidade de encontrar problemas é superior. A necessidade de conceção de soluções é diretamente relacionada à complexidade do problema em questão: quanto mais complexo o projeto, mais vantagem existe num planeamento e estudo inicial.

Para além destas quatro atividades são definidos cinco valores [19, 20]:

1. *Communication* – comunicar constantemente propaga conhecimento e facilita a ajuda entre os colegas de equipa. A comunicação constante com os clientes e utilizadores promove uma maior perceção das suas necessidades e o desenvolvimento de soluções mais adequadas.
2. *Simplicity* – o trabalho vai incidir sobre o que é pedido e nada mais. Isto aumenta a rapidez do desenvolvimento e usa de forma mais eficaz os recursos do projeto.

3. *Feedback* – existir um *feedback* a cada iteração é importante para entender que alterações são necessárias e construir uma solução mais apropriada às necessidades dos clientes.
4. *Respect* – todos os elementos devem ser tratados com respeito e todos devem ser valorizados. Tal contribui para a motivação da equipa e, conseqüentemente, para uma maior produtividade e melhor desempenho.
5. *Courage* – falhar é algo normal e que não deve ser escondido. Este reconhecimento permite que a equipa tenha um maior à vontade para procurar ajuda e comunicar problemas.

São ainda definidas regras a seguir que estabelecem o processo de desenvolvimento: *Planning, Managing, Coding, Designing* e *Testing*, que respeitam tanto o definido nas atividades, como os valores da metodologia. Estas regras estabelecem o conjunto de passos a cumprir.

#### 2.3.4. *Dynamic Systems Development Method (DSDM)*

A metodologia *Dynamic Systems Development Method (DSDM)*, baseada numa outra metodologia ágil (RAD – *Rapid Application Development*), é indicada para projetos com duração, orçamentos e recursos bastante limitados, que têm de ser respeitados mesmo com a permissão de alterações aos requisitos. Até 2014, a metodologia DSDM era constituída pelos seguintes princípios [21]:

1. *Active user involvement;*
2. *Teams must be empowered to make decisions;*
3. *Focus on frequente delivery;*
4. *Criterion for accepted deliverable (Fitness for Business);*
5. *Iterative and incremental development;*
6. *All changes during development must be reversible;*
7. *Requirements are base lined at high level;*
8. *Testing is integrated throughout the life cycle;*
9. *Collaborative and co-operative approach.*

Porém, em 2014, os princípios foram reformulados para satisfazer as necessidades de evolução das empresas *agile*. Esta reformulação permitiu utilizar DSDM com métodos de gestão clássicos de projeto e com metodologias complementares ágeis como, por exemplo, o *Scrum* [22].

Assim, os princípios passaram a ser os seguintes [23]:

1. *Focus on the business need;*
2. *Deliver on time;*
3. *Collaborate;*
4. *Never compromise quality;*
5. *Build incrementally from firm foundations;*
6. *Develop iteratively;*
7. *Communicate continuously and clearly;*
8. *Demonstrate control.*

A DSDM recomenda a utilização das técnicas e processos MOSCOW, *Timeboxing*, prototipagem, *Testing*, etc., no projeto e explica como os utilizar; sugere treze papéis detalhando as respetivas responsabilidades e área de interesse a que pertencem; identifica fatores de sucesso, as suas vantagens, e técnicas para os gerir; e define as fases que estabelecem o ciclo de vida típico de um projeto e os processos que devem ocorrer em cada uma dessas fases (ver Figura 7) [23].

O facto de existir um grande detalhe associado a esta metodologia pode ser visto uma desvantagem. Dado que os projetos têm características únicas, um método específico e com grande detalhe restringe o ambiente de projeto a que pode ser aplicado.

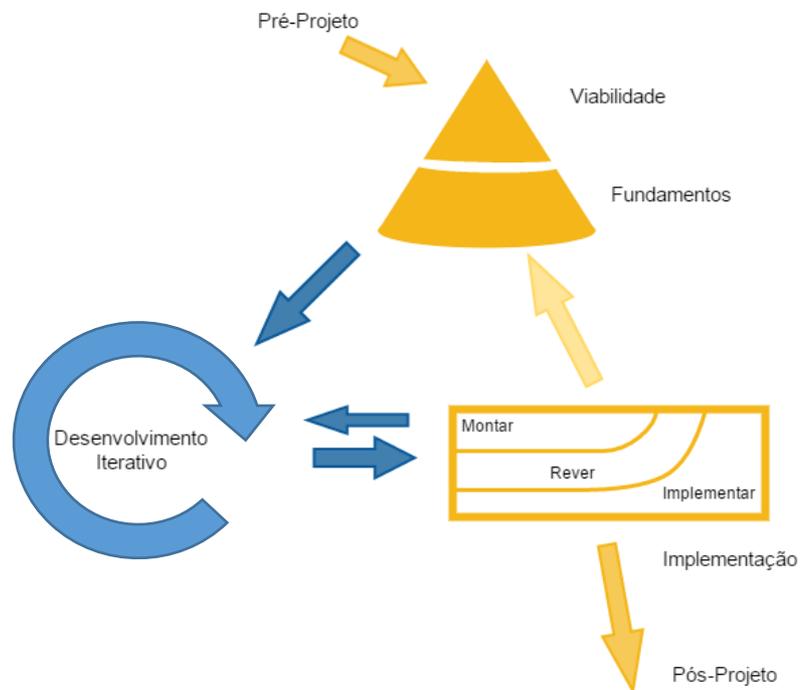


Figura 7 - Ciclo de Vida de um Projeto DSDM  
Fonte: Agile Business Consortium [22]

### 2.3.5. *Lean Development*

A metodologia *Lean Development* de desenvolvimento de *software* estabelece os seguintes princípios para atingir qualidade, velocidade do projeto e satisfação do cliente [24]:

1. *Eliminate waste* – esta metodologia dita que uma tarefa que não traga valor ao produto é desperdício. Estas tarefas podem ser trabalho de desenvolvimento extra, tarefas da área de gestão, documentação, etc. Toda a tarefa que seja incorporada no projeto e não traga valor é considerada desperdício. Assim sendo, o *Lean Development* propõe que sejam utilizados métodos de rastreio de desperdício e que o valor inerente às tarefas seja claro.
2. *Build in quality* – este princípio defende a criação de resultados de qualidade logo no início do desenvolvimento e não codificar e depois corrigir e/ou otimizar. Este princípio é diretamente associado à metodologia *Test Driven Development* para garantir qualidade de produto. A vantagem é clara: no final do processo de desenvolvimento não existe uma fase de testes e correção, porque esta fase foi distribuída ao longo do desenvolvimento.
3. *Create knowledge* – a existência de pequenas iterações de desenvolvimento aumenta a propagação de conhecimento e torna o desenvolvimento mais rápido. Além disto, nas reuniões de validação sobre o que foi realizado (mais frequentes em pequenas iterações), consegue-se perceber mais sobre o domínio do problema e sobre como satisfazer as necessidades do cliente.
4. *Defer commitment* – sabendo que na maior parte dos projetos de desenvolvimento de *software* existe sempre um determinado grau de incerteza, atrasar o mais possível as decisões incertas é benéfico, porque, entretanto, pode já ter sido diminuída ou eliminada a incerteza referente à decisão. Uma decisão tomada antecipadamente sem necessidade pode levar a um desperdício de recursos, tendo em conta a incerteza envolvida nos projetos. Deve ser construída uma arquitetura de solução flexível e tolerante que satisfaça as necessidades de negócio.
5. *Deliver quickly* – a existência de entregas rápidas implica que a equipa trabalhe em pequenas partes do problema. Isto aumenta o foco da equipa e reduz o *stress* dos elementos. Para além disto, quanto mais rápida for a entrega de *software*, mais rapidamente se obtém *feedback* sobre o mesmo. Assim, as próximas iterações já terão mais *feedback* a incorporar e a probabilidade do resultado ir de encontro às necessidades do cliente é maior.

6. *Respect people* – com o objetivo de formar uma equipa motivada, eficiente e criativa, este princípio defende que os indivíduos devem ser respeitados. Uma pessoa que se sinta respeitada tem uma maior motivação para trabalhar e opinar sobre o trabalho corrente. Técnicas de *team building* podem ser vantajosas e promover o respeito entre colegas de equipa.
7. *Optimize the whole* – “*look at the bigger picture*” para obter uma solução adequada e que vá de encontro ao que se quer obter, é necessário ter em conta os processos de mais alto nível que os projetos individuais apoiam. É preciso gerir os programas e as suas relações para que os objetivos estejam claros e coerentes entre eles e para que o produto final corresponda ao que o cliente necessita.

Existem algumas técnicas que promovem estes princípios, incluindo o desenvolvimento por iterações, *testing*, *value stream mapping*, atividades de *team building*, etc.

## 2.4. Metodologias de Gestão de Projetos Articuladas com Metodologias Ágeis

A necessidade de existir uma metodologia de gestão de projetos é fundamental para o sucesso dos projetos [25, 26] e já foi referida anteriormente nesta dissertação. Assim, assumindo uma metodologia de desenvolvimento ágil, será necessário perceber quais são as metodologias de gestão de projetos compatíveis (ou orientadas ao desenvolvimento ágil) que poderão ser utilizadas. A escolha de uma determinada metodologia será diretamente influenciada pelo ambiente de trabalho do projeto e pelas normas e regras dos *stakeholders* envolvidos.

A aplicação de metodologias de gestão de projetos ditas clássicas na gestão de projetos, em conjunto com metodologias ágeis de desenvolvimento, é também uma possibilidade válida a explorar. Tenha-se o exemplo da conjugação do PMBoK com o *Scrum*: pode observar-se que a construção do plano de projeto (segundo uma abordagem *rolling-wave*) pode corresponder à construção das *sprint plannings*; definir o âmbito pode corresponder à construção do *backlog*, etc. Em suma, as várias atividades do *Scrum* podem ser apoiadas e incorporadas numa gestão de projetos baseada no PMBoK.

Note-se o caso da alteração recente da metodologia DSDM [27], suscitada pelas necessidades atuais das empresas e organizações. Com a alteração, a metodologia tornou-se mais compatível com os métodos de gestão de projetos convencionais, como o PMBoK

ou PRINCE2. Isto é uma evidência da necessidade atual da compatibilização de metodologias de gestão de projetos ditas tradicionais com metodologias ágeis. Uma organização que harmonize a utilização das várias metodologias consegue potenciar as vantagens de ambas.

As soluções atuais de gestão de projetos com utilização de metodologias ágeis são muito orientadas à metodologia ágil [28, 29] , isto é, procuram definir a utilização da metodologia ágil e explicar e caracterizar as vantagens da sua integração. Porém, não fornecem uma metodologia de gestão de projetos estruturada que auxilie a prática da gestão sobre estas metodologias. Existem, contudo, soluções com estratégias interessantes [30, 31], mas que não englobam todos os fatores relevantes no que toca à gestão de projetos. Subsistem realmente desafios no que toca à introdução de metodologias ágeis numa empresa [32], no entanto, nesta dissertação, procura-se dar apoio à prática de uma gestão de projeto estruturada conciliada com uma metodologia ágil e não apenas introduzir os leitores às práticas de metodologias ágeis.

De reforçar a ideia de que as metodologias clássicas e ágeis não são incompatíveis, sendo viável e desejável articular uma metodologia clássica de gestão de projetos com uma metodologia ágil de desenvolvimento para levar a cabo um projeto.

---

## NOVO MÉTODO DE GESTÃO DE PROJETOS

---

Neste capítulo são apresentados os fundamentos utilizados para o desenvolvimento do novo método de gestão de projetos, é apresentada uma descrição global do método proposto e, por último, são enunciados os processos do método alusivos à sua implementação.

### 3.1. Fundamentos do Método

Para o desenvolvimento do novo método de gestão de projetos foram escolhidos o *Scrum* como metodologia ágil e o PMBoK como *standard* de gestão de projetos.

O *Scrum* reflete-se positivamente na produtividade, qualidade do produto, satisfação do cliente, redução de custos e motivação da equipa [33-35], características diretamente relacionadas ao sucesso de projetos [3, 36]. É, por isto, a metodologia ágil base escolhida para o desenvolvimento do novo método. Note-se que utilizar uma metodologia não implica que todos os projetos a cumpram rigidamente. Ou seja, variações na metodologia consoante as necessidades do projeto são expectáveis: "*ScrumMasters have only to refer back to Scrum theory to find Scrum practices that can be readily adapted to handle even the most complex projects*" [37].

O PMBoK foi escolhido como base para a componente de gestão de projetos por ser um guia utilizado mundialmente [38] e pela sua capacidade de influência no sucesso dos projetos ao agregar as melhores práticas de gestão [39].

Esta decisão teve ainda em conta o facto de a eSPap utilizar *Scrum* e PMBoK nos seus projetos, o que facilita o processo de introdução e a adaptação das equipas ao novo método. Ambos definem a base estrutural a utilizar no desenvolvimento do método. Porém, não significa que não possam ser incorporadas técnicas e *guidelines* pertencentes a outras

metodologias. Tenha-se o exemplo da *Extreme Programming* e *Scrum*: a primeira terá influência na qualidade do código produzido e a segunda na produtividade e valorização do produto [40].

### 3.2. Descrição Global do Método

Globalmente identificam-se as seguintes fases para o projeto:

4. Iniciação do Projeto – fase onde é estabelecida a definição do projeto e obtida a autorização para avançar com o mesmo. Esta fase é composta pelos seguintes processos globais:
  - 4.1. Definir o *Statement of Work*;
  - 4.2. Desenvolver o *Business Case*;
  - 4.3. Elaborar o *Project Charter*.
5. Planeamento do Projeto – fase onde são definidos aspetos cruciais para o desenvolvimento do projeto com sucesso como, por exemplo, âmbito, custo e tempo. É dividida nos seguintes processos:
  - 5.1. Elaborar o Plano de Projeto;
  - 5.2. Definir o *Product Backlog*.
6. Execução do Projeto – fase em que decorre o desenvolvimento de cada um dos *deliverables* de forma a cumprir os objetivos do projeto. Esta fase é decompõe-se nos seguintes processos:
  - 6.1. Planear a *Sprint*;
  - 6.2. Executar a *Sprint*;
  - 6.3. Concluir a *Sprint*.
7. Monitorização e Controlo do Projeto – aplicada em paralelo à execução, tem como objetivo monitorizar o progresso do projeto e tomar medidas de ação/reação quando necessário;
8. Conclusão do Projeto – fase onde são terminadas as contratualizações realizadas no projeto, estabelecendo o término do mesmo e consolidadas as lições aprendidas.

Tenha-se em consideração que os processos identificados são apenas aqueles que dizem respeito à gestão de integração e que a numeração utilizada representa os capítulos onde estes são detalhados para mais fácil referência. Na Figura 8 é possível observar a organização referente ao que foi acima descrito.

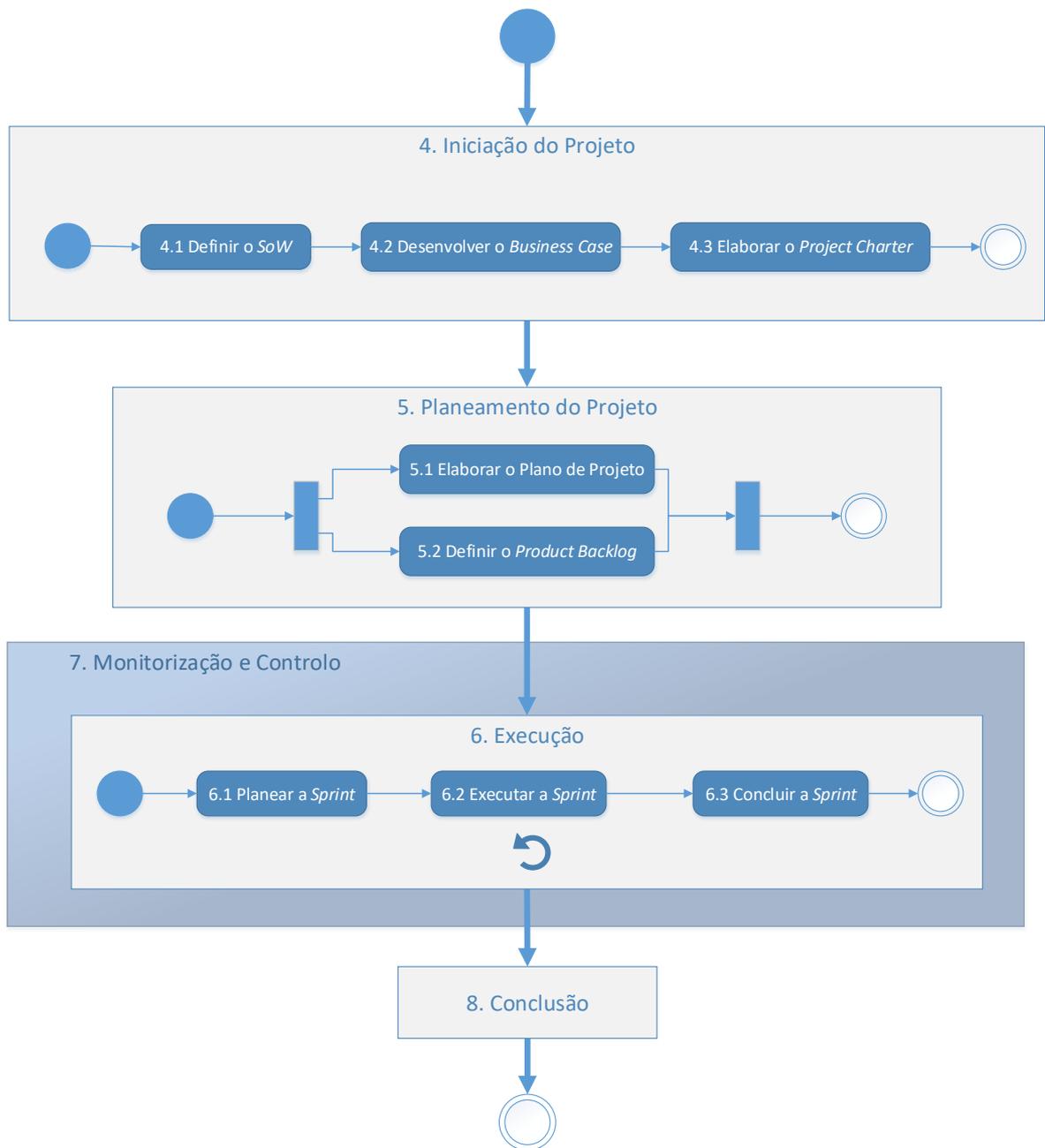


Figura 8 - Estrutura de um Projeto

### 3.3. Processos do método

A tabela seguinte apresenta os processos que constituem o novo método que procura compatibilizar as metodologias clássicas de gestão de projeto com metodologias ágeis de desenvolvimento, organizados em áreas de conhecimento. De salientar a introdução de uma área de conhecimento suplementar, a área do Sucesso [41].

Fase	Área	Processo
4. Iniciação (I)	Integração (I)	4.1 Definir o <i>Statement of Work</i> 4.2 Desenvolver o <i>Business Case</i> 4.3 Elaborar o Project Charter
	Âmbito (Â)	4.3.1 Justificar a Existência do Projeto 4.3.2 Definir os Objetivos de Alto Nível 4.3.3 Identificar os Requisitos de Alto Nível 4.3.4 Definir os Deliverables 4.3.5 Identificar as Restrições e os Pressupostos
	Tempo (T)	4.3.6 Identificar as <i>Milestones</i> 4.3.7 Construir a <i>Timeline</i>
	Custo (C)	4.3.8 Identificar os Recursos Necessários 4.3.9 Estimar o Orçamento
	Recursos Humanos (RH)	4.3.10 Esquematizar a Organização
	Risco (R)	4.3.11 Identificar os Riscos 4.3.12 Categorizar os Riscos 4.3.13 Elaborar o Plano de Resposta aos Riscos
	Qualidade (Q)	4.3.14 Definir os Critérios de Avaliação
	Sucesso (Su)	4.3.15 Identificar os Fatores de Sucesso
	<i>Stakeholders</i> (St)	4.3.16 Identificar e Caracterizar os <i>Stakeholders</i>
	Comunicação (Co)	—
	Aquisições (A)	4.3.17 Identificar Preliminarmente as Aquisições

<b>5. Planejamento (P)</b>	Integração (I)	5.1 Elaborar o Plano de Projeto 5.2 Definir o <i>Product Backlog</i>
	Âmbito (Â)	5.1.1 Identificar os Requisitos 5.1.2 Definir o Âmbito 5.1.3 Detalhar as Restrições e os Pressupostos 5.1.4 Definir o Não-âmbito 5.1.5 Identificar os <i>Deliverables</i> 5.2.1 Identificar as <i>User Stories</i> 5.2.2 Validar as <i>User Stories</i> 5.2.3 Priorizar as <i>User Stories</i> (Âmbito) 5.2.4 Aceitação do <i>Product Backlog</i>
	Tempo (T)	5.1.6 Definir as Atividades 5.1.7 Sequenciar as Atividades 5.1.8 Estimar os Recursos das Atividades 5.1.9 Estimar as Durações 5.1.10 Construir o Cronograma 5.1.11 Estimar as Reservas 5.2.3 Priorizar as <i>User Stories</i> (Tempo)
	Custo (C)	5.1.12 Determinar o Orçamento 5.2.3 Priorizar as <i>User Stories</i> (Custo)
	Recursos Humanos (RH)	5.1.13 Desenvolver o Plano de Recursos Humanos 5.1.14 Definir a Organização
	Risco (R)	5.1.15 Criar o Plano de Gestão de Risco 5.2.3 Priorizar as <i>User Stories</i> (Risco)
	Qualidade (Q)	5.1.16 Planejar as Medidas de Qualidade
	Sucesso (Su)	5.1.17 Definir o Plano de Gestão do Sucesso
	<i>Stakeholders</i> (St)	5.1.18 Definir a Gestão dos <i>Stakeholders</i>
	Comunicação (Co)	5.1.19 Desenvolver o Plano de Gestão da Comunicação
	Aquisições (A)	5.1.20 Planejar Aquisições

<b>6. Execução (E)</b>	Integração (I)	6.1 Planear a <i>Sprint</i> 6.2 Executar a <i>Sprint</i> 6.3 Concluir a <i>Sprint</i>
	Âmbito (Â)	6.1.1.2 Realizar a <i>Grooming Meeting</i> 6.1.1.3 Atualizar o <i>Product Backlog</i> 6.1.2.2 Especificar as <i>User Stories</i> (Âmbito) 6.1.2.3 Realizar a <i>Sprint Planning Meeting</i> (Âmbito) 6.3.1 <i>Sprint Review</i>
	Tempo (T)	6.1.1.4 Atualizar a Calendarização do Projeto
	Custo (C)	6.1.1.5 Atualizar o Orçamento do Projeto
	Recursos Humanos (RH)	6.2.1 <i>Daily Scrum</i>
	Risco (R)	6.3.2 <i>Sprint Retrospective</i> (Risco)
	Qualidade (Q)	6.1.2.2 Especificar as <i>User Stories</i> (Qualidade) 6.1.2.3 Realizar a <i>Sprint Planning Meeting</i> (Qualidade)
	Sucesso (Su)	6.3.2 <i>Sprint Retrospective</i> (Sucesso)
	<i>Stakeholders</i> (St)	6.1.1.1 Preparar a <i>Grooming Meeting</i> ( <i>Stakeholders</i> ) 6.1.2.1 Preparar a <i>Sprint Planning Meeting</i> ( <i>Stakeholders</i> )
	Comunicação (Co)	6.1.1.1 Preparar a <i>Grooming Meeting</i> (Comunicação) 6.1.2.1 Preparar a <i>Sprint Planning Meeting</i> (Comunicação)
Aquisições (A)	—	

<b>7. Monitorização e Controlo (M)</b>	Integração (I)	–
	Âmbito (Â)	7.1 Controlar o Âmbito
	Tempo (T)	7.2 Controlar o Tempo
	Custo (C)	7.3 Monitorizar o Orçamento
	Recursos Humanos (RH)	7.4 Monitorizar a Equipa do Projeto
	Risco (R)	7.5 Monitorizar e Controlar os Riscos
	Qualidade (Q)	7.6 Assegurar a Qualidade
	Sucesso (Su)	7.7 Controlar os Fatores de Sucesso
	<i>Stakeholders</i> (St)	7.8 Gerir as Expectativas dos <i>Stakeholders</i>
	Comunicação (Co)	7.9 Controlar as Comunicações
	Aquisições (A)	7.10 Levar a Cabo as Aquisições 7.11 Monitorizar as Aquisições
<b>8. Conclusão (C)</b>	Integração (I)	–
	Âmbito (Â)	8.2 Consolidação das Lições Aprendidas (Âmbito)
	Tempo (T)	8.2 Consolidação das Lições Aprendidas (Tempo)
	Custo (C)	8.2 Consolidação das Lições Aprendidas (Custo)
	Recursos Humanos (RH)	8.2 Consolidação das Lições Aprendidas (Recursos Humanos)
	Risco (R)	8.2 Consolidação das Lições Aprendidas (Risco)
	Qualidade (Q)	8.2 Consolidação das Lições Aprendidas (Qualidade)
	Sucesso (Su)	8.2 Consolidação das Lições Aprendidas (Sucesso)
	<i>Stakeholders</i> (St)	8.2 Consolidação das Lições Aprendidas ( <i>Stakeholders</i> )
	Comunicação (Co)	8.2 Consolidação das Lições Aprendidas (Comunicação)
	Aquisições (A)	8.2 Consolidação das Lições Aprendidas (Aquisições) 8.1 Fecho de Contratos e Aquisições

Nos próximos capítulos os vários processos são descritos em detalhe, organizados por fase do projeto. De notar que a repetição de alguns processos ao longo de diferentes áreas de conhecimento é justificada nesses capítulos.

### 3.4. Papéis e Responsabilidades

Para além da possível existência de um *Project Management Office* (PMO) ou de outras entidades administrativas da empresa que supervisionam, auxiliam e/ou interagem com os projetos, associados ao método proposto estão previstos os seguintes papéis:

- Gestor de Projeto;
- *Scrum Master*;
- *Product Owner*;
- Equipa de Desenvolvimento (elementos).

Cada papel tem associado diversas responsabilidades e, nesta secção, por forma a clarificar a importância de cada um dos papéis, são apresentadas as implicações e encargos a eles associados.

#### 3.4.1. Gestor de Projeto

O gestor de projeto é responsável por estudar, planear e monitorizar o desenvolvimento do projeto para que se consigam atingir os objetivos do mesmo [42].

Genericamente, o gestor de projeto é responsável por:

- Estudar e formar um plano de projeto que direciona o desenvolvimento do projeto de forma a maximizar a sua probabilidade de sucesso;
- Reportar sobre o estado de projeto a entidades supervisoras;
- Construir e manter atualizada toda a documentação associada ao projeto (por exemplo: SoW, Plano de Projeto, Planos de Contingência dos Riscos);
- Aplicar técnicas e boas práticas da gestão de projetos para procurar o sucesso do projeto;
- Monitorizar e controlar o desenvolvimento do projeto segundo o plano desenvolvido, identificando desvios no projeto e reagindo de acordo;

- Exercer capacidades interpessoais e outras *soft skills* que auxiliem na interação com os *stakeholders* e beneficiem o desenvolvimento do projeto.

#### 3.4.2. *Scrum Master*

O *Scrum Master* é o “treinador” da equipa no que concerne à metodologia de desenvolvimento. Tipicamente, é o elemento mais conhecedor da metodologia e o mais experiente na sua aplicação. O *Scrum Master* está encarregue de:

- Assegurar que os princípios da metodologia de desenvolvimento são respeitados;
- Remover obstáculos e impedimentos que surjam ao longo do desenvolvimento;
- Maximizar, em conjunto com o *Product Owner*, o valor do resultado final do projeto;
- Assegurar que o resultado a ser produzido corresponde às expectativas do cliente e respeita o que foi acordado;
- Procurar otimizar a aplicação da metodologia de desenvolvimento;
- Procurar melhorar a produtividade e desempenho da equipa de desenvolvimento;
- Organizar e manter atualizada toda a informação relevante sobre o desenvolvimento do projeto;
- Reportar, sempre que necessário, ou quando pedido, sobre o estado de projeto ao gestor do projeto;
- Manter a equipa de desenvolvimento motivada e identificar necessidades de formação;
- Promover a auto-organização da equipa de desenvolvimento;
- Apresentar capacidades interpessoais e outras *soft skills* que o permitam liderar e promover a produtividade da equipa ao longo do projeto.

### 3.4.3. *Product Owner*

Representante do cliente dentro da equipa de desenvolvimento, o *Product Owner* procura maximizar o valor do resultado final do ponto de vista de negócio, certificando que o mesmo respeita as necessidades do cliente. O *Product Owner* é responsável por:

- Priorização das tarefas a desenvolver de forma a que sejam produzidas as funcionalidades mais valiosas primeiro;
- Maximizar, em conjunto com o *Scrum Master*, o valor do resultado final do projeto;
- Verificar a adequação da solução a desenvolver segundo o domínio de negócio e necessidades do cliente;
- Validar e responder a dúvidas da equipa de desenvolvimento no âmbito de negócio;
- Definir critérios de aceitação e de qualidade sobre o resultado final;
- Reportar sempre que necessário ao gestor de projeto, cliente ou outro elemento supervisor da área de negócios, sobre o decorrer do projeto;
- Possuir boa capacidade de comunicação, negociação e outras *soft skills* necessárias para a valorização do resultado final e interação com diversos *stakeholders*.

### 3.4.4. Equipa de Desenvolvimento (elementos)

A equipa de desenvolvimento é responsável pela implementação do resultado final para conseguir atingir os objetivos do projeto. Os elementos da equipa de desenvolvimento responsabilizam-se por:

- Produção de porções funcionais que, quando integradas, constituem o produto final resultante do projeto;
- Respeitar e cumprir as normas, regras e objetivos definidos para o projeto;
- Respeitar os valores da metodologia e os colegas de trabalho;
- Ter a capacidade de auto-organização e de motivação própria;

- Estimar o trabalho a desenvolver tendo em conta as suas capacidades;
- Transmitir problemas ou outro tipo de questões com impacto no desenvolvimento e eficácia de produção ao *Scrum Master*;
- Atualizar o estado das tarefas a seu encargo;
- Reportar, quando necessário, a outros elementos do projeto (tipicamente ao *Scrum Master* ou ao *Product Owner* no caso de existirem dúvidas de negócio e/ou valorização do produto).

---

## FASE DE INICIAÇÃO

---

A fase de iniciação constitui a preparação do projeto, é aqui que se define o projeto de forma a obter a autorização para o levar a cabo.

O projeto é iniciado pela abertura de um *Statement of Work (SoW)*, onde é justificada e detalhada a natureza do projeto.

Nesta fase é também desenvolvido o estudo de negócio (*Business Case*), que detalha a viabilidade e justifica a existência do projeto e é construído o *Project Charter*, documento utilizado para suporte à decisão de continuar ou abandonar o projeto.

Na Figura 9 é apresentado o esquema de processos associado à fase de iniciação.

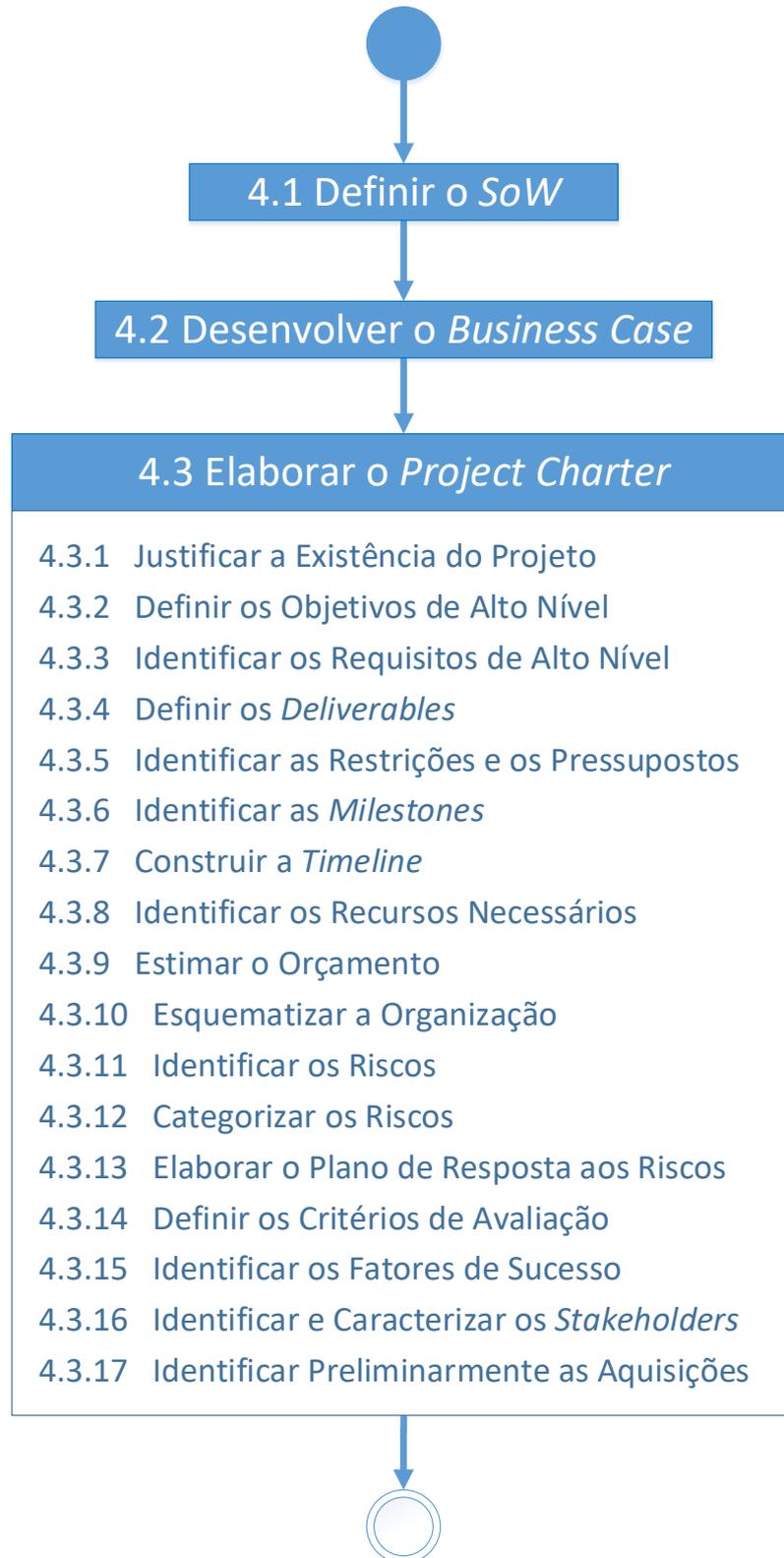


Figura 9 - Esquema da Fase de Iniciação

#### 4.1. Definir o *Statement of Work*

Desenvolvido pelo *Project Initiator* (elemento promotor do projeto), o *Statement of Work* (*SoW*) é um documento formal sobre qual todas as partes envolvidas devem acordar e assinar.

O documento do projeto deve ser iniciado por um resumo executivo. O resumo executivo oferece uma explicação curta e sucinta sobre o que trata o documento e que detalhes relevantes devem ser considerados. Para os elementos que monitorizam globalmente os projetos da empresa cujo tempo é precioso, existir um resumo que destaca a informação global relevante do documento é crucial e apreciado. O *SoW* e todos os documentos apresentados nesta dissertação seguem esta prática, porém, para evitar repetições, não será mencionado o resumo executivo na estrutura dos restantes documentos.

##### **Inputs:**

- Informação acordada com o cliente durante as reuniões no que toca ao âmbito, cronograma, principais pressupostos e preçário;

##### **Estrutura**

O *SoW* deve abordar os seguintes temas:

- Necessidade do projeto – nesta secção devem ser respondidas questões como “Este projeto é necessário porquê?”, “Que vantagens vai trazer o projeto?”. Deve ser explicado o problema ou oportunidade de negócio que dá origem ao projeto;
- Âmbito do projeto – com a descrição do âmbito devem ser respondidas as questões “O que vai ser feito no projeto?” e “O que não vai ser feito no projeto?”, ou seja, descrever o trabalho que será necessário realizar no projeto para se obter o produto final. Além de apresentar o que vai ser feito, deve esclarecer-se o que não vai ser realizado, para que o cliente entenda o que está compreendido (ou não) no projeto;
- Principais objetivos – “O que se pretende atingir com este projeto?”, estes objetivos devem ser o mais simples e claro que for possível, para além de específicos e realistas;
- *Deliverables* – “O que vai ser produzido com o projeto”, “O que é que o cliente vai receber?” e “Quais são as características do produto final?” são as perguntas respondidas nesta secção, onde é exposta a listagem do que vai ser entregue e

as características cada um dos *deliverables* deverá cumprir. A caracterização dos *deliverables* é importante para que as partes interessadas acordem sobre o que deve constar dos produtos principais do projeto, por exemplo, detalhes de desempenho e/ou segurança;

- Critérios de aceitação – esta secção estabelece os critérios que o cliente utilizará aquando a aceitação do produto. O produto deverá ser avaliado em função destes critérios;
- Plano estratégico de alto nível – o plano estratégico responde às questões: “Quanto tempo vai demorar o projeto?”, “Qual o custo do projeto?”, “Que recursos serão necessários?”, etc. O objetivo é que se entenda a estratégia para levar a cabo o projeto.

Cada projeto poderá necessitar de mais ou menos detalhe na documentação, o mesmo acontece no *SoW*. Podem ser adicionados tópicos como a localização onde o projeto vai decorrer, a equipa do projeto, normas que o resultado final deve respeitar, detalhes de pagamento ou de contratos, etc. ou serem retirados tópicos que, no projeto em questão, não sejam relevantes.

## 4.2. Desenvolver o *Business Case*

O estudo de negócio é importante para perceber o valor de negócio que o projeto trará à organização e, com isto, demonstrar a sua viabilidade. É neste documento que se apresentam os benefícios inerentes ao projeto, demonstrando o seu valor.

O *Business Case* é tipicamente construído pelo *Project Sponsor*.

Note-se que poderá ser previamente realizado um estudo de mercado completo cujas conclusões conduzam ao aparecimento do projeto. Este documento tomará por base esse estudo e explicará o valor de negócio do projeto.

### **Inputs:**

- *Statement of Work*;
- Estudo Prévio de Mercado (se existente).

## Estrutura

A construção do *Business Case* é bastante complexa e aborda uma série de temas ligados à área de negócio, tanto que, tipicamente, desenvolver o *Business Case* implica complementar o documento à medida que é encontrada mais informação.

Sendo que o objetivo não é definir rigidamente como se desenvolve um *Business Case*, mas apenas apresentar direções para o construir, apresentam-se os temas que o documento deverá esclarecer:

- Problema/oportunidade onde o projeto vai incidir – o primeiro passo trata-se de responder à questão “Porque surgiu o projeto?”. Pode ter sido para resolver um problema que se identificou ou para introduzir algo novo. Descrever a razão ajuda a entender os motivos que justificam a continuação do projeto.
- Benefícios e limitações – “Que vantagens trará o projeto?”, apresentar os benefícios que advêm da implementação do projeto. Caso existam, as limitações devem também ser apresentadas;
- Análise de sensibilidade – identificar quais as influências, restrições e riscos e medir o impacto de cada um desses riscos no projeto. Podem eventualmente ser descritos cenários distintos para ilustrar o impacto de cada um dos fatores apresentados;
- Análise financeira – apresentar estimativas de custos, recursos, tempo e receitas. Todos os valores financeiros do projeto devem ser expostos nesta secção: investimento inicial, rendimento esperado, custo de venda (se aplicável), etc.;
- Análise de mercado – na análise de mercado deve constar a descrição do mercado atual, que inclui a análise de concorrência e a análise de potencial de mercado. Para além disto, devem ser apresentados os pressupostos considerados na obtenção dos valores de mercado, que justificam a posição estratégica que se tenciona tomar. Este tema é aplicado a projetos cuja intenção é lançar para o mercado um produto ou serviço, caso contrário não deve ser considerado;
- Estratégia de negócio – apresentar todas as atividades necessárias ao desenvolvimento do projeto e o plano associado. Isto inclui não só atividades de desenvolvimento do projeto, como atividades de promoção, marketing, parcerias estratégicas, etc.

### 4.3. Elaborar o Project Charter

O *Project Charter* auxilia na tomada de decisão sobre continuar/abandonar o projeto. Assim, este documento, deverá conter os detalhes necessários à obtenção da autorização do projeto.

A construção do *Project Charter* incide em várias áreas de conhecimento definidas pelo PMBoK [2]. A Tabela 3 apresenta os processos segundo a área em que se enquadram.

Âmbito (Â)
4.3.1 Justificar a Existência do Projeto
4.3.2 Definir os Objetivos de Alto Nível
4.3.3 Identificar os Requisitos de Alto Nível
4.3.4 Definir os <i>Deliverables</i>
4.3.5 Identificar as Restrições e Pressupostos
Tempo (T)
4.3.6 Identificar as <i>Milestones</i>
4.3.7 Construir a <i>Timeline</i>
Custo (C)
4.3.8 Identificar os Recursos Necessários
4.3.9 Estimar o Orçamento de Alto Nível
Recursos Humanos (RH)
4.3.10 Esquematizar a Organização
Risco (R)
4.3.11 Identificar os Riscos
4.3.12 Categorizar os Riscos
4.3.13 Elaborar o Plano de Resposta aos Riscos
Qualidade (Q)
4.3.14 Definir os Critérios de Avaliação

Sucesso (Su)
4.3.15 Identificar os Fatores de Sucesso
Stakeholders (St)
4.3.16 Identificar e Caracterizar os <i>Stakeholders</i>
Aquisições (A)
4.3.17 Identificar Preliminarmente as Aquisições

Tabela 3 - Áreas de Conhecimento Abordadas na Construção do *Project Charter*

### **Inputs Principais:**

- *Statement of Work*;
- *Business Case*;
- Fatores internos da empresa – como registos da organização, normas, equipa, recursos existentes, contratos, etc.;
- Fatores externos relevantes – legislações aplicáveis, preçários, etc.

### **Técnicas Principais:**

- Consulta de peritos;
- Consulta de relatórios de lições aprendidas;
- Consulta de outra documentação relevante – documentação de fatores internos ou externos.

#### 4.3.1. Justificar a Existência do Projeto

A justificação da existência do projeto esclarece a necessidade inerente ao projeto e detalha o problema ou oportunidade que o tornam necessário.

**Área de Conhecimento:** (Â) Âmbito.

### **Perguntas-chave:**

- Por que razão surgiu o projeto?
- Que problema ou oportunidade se pode associar à criação do projeto?

#### 4.3.2. Definir os Objetivos de Alto Nível

Os objetivos de alto nível são os objetivos principais que se esperam cumprir com o desenvolvimento do projeto.

**Área de Conhecimento:** (Â) Âmbito.

**Perguntas-chave:**

- O que deverá o projeto conseguir?
- O que se espera obter com o projeto?

#### 4.3.3. Identificar os Requisitos de Alto Nível

O levantamento de requisitos de alto nível é feito com base nas reuniões com o cliente. Aqui, procura-se descobrir as características funcionais e não funcionais de alto nível que se esperam do produto final.

**Área de Conhecimento:** (Â) Âmbito.

**Perguntas-chave:**

- O que deve o resultado final realizar?
- Que características se esperam do produto/serviço obtido com o desenvolvimento do projeto?

#### 4.3.4. Definir os *Deliverables*

Os *deliverables* correspondem ao que o deve ser obtido (produto, serviço ou outro resultado) no projeto, isto é, os artefactos e resultados que o cliente vai receber com o desenvolvimento do projeto.

**Área de Conhecimento:** (Â) Âmbito.

**Perguntas-chave:**

- O que será entregue ao cliente?
- Que produtos, serviços ou outros resultados do projeto são esperados?

#### 4.3.5. Identificar as Restrições e os Pressupostos

As restrições clarificam os limites do projeto e os pressupostos representam informação que se pensa ser verdadeira.

**Área de Conhecimento:** (Â) Âmbito.

**Perguntas-chave:**

- O que deve o produto final respeitar/cumprir?
- Que fatores podemos assumir como certos?
- Que restrições limitam o projeto? (Recursos? Tempo? Tecnologia?...)

#### 4.3.6. Identificar as *Milestones*

As *milestones* devem ser caracterizadas com uma descrição e ter uma data associada. Estas *milestones* são pontos importantes de controlo no decorrer do projeto.

**Área de Conhecimento:** (T) Tempo.

**Perguntas-chave:**

- Quais as *milestones* do projeto?
- Qual a data associada a cada uma das *milestones*?
- O que define a *milestone*?

#### 4.3.7. Construir a *Timeline*

Após identificadas as *milestones* é construída a *timeline* do projeto. Para isto, são identificadas as atividades de alto nível associadas ao desenvolvimento do projeto e, posteriormente, são estimadas as suas durações e estabelecida a sequenciação de atividades esperada. Note-se que tudo isto é feito a um nível alto de detalhe, apenas para obter uma noção global sobre a *timeline* do projeto.

**Área de Conhecimento:** (T) Tempo.

**Perguntas-chave:**

- Quais as atividades de alto nível que definem o projeto?

- Quanto tempo se estima que dure cada uma das atividades do projeto?
- Como se espera que seja estabelecida a sequenciação das atividades?
- Quanto tempo se prevê que o projeto dure?

#### 4.3.8. Identificar os Recursos Necessários

Por forma a estimar o orçamento é preciso identificar os recursos necessários ao projeto associando-lhes um custo e quantidade necessária.

**Área de Conhecimento:** (C) Custo.

**Perguntas-chave:**

- Que recursos são necessários para levar a cabo o projeto?
- Quanto custa cada um dos recursos necessários?

#### 4.3.9. Estimar o Orçamento

É elaborada uma estimativa de custos baseada na listagem de recursos necessários para o desenvolvimento do projeto. A listagem é também apresentada justificando o orçamento atribuído ao projeto.

**Área de Conhecimento:** (C) Custo.

**Perguntas-chave:**

- Qual o custo total calculado com base nos recursos?
- Qual se pensa ser o orçamento do projeto?

#### 4.3.10. Esquematizar a Organização

A equipa do projeto é determinada na fase de iniciação, incluindo o gestor de projeto, que pode ou não ser o mesmo profissional que elabora toda a documentação de gestão na iniciação do projeto.

O *Project Charter* associa, na secção da organização do projeto, os vários elementos da equipa de projeto aos cargos que vão desempenhar, descrevendo as responsabilidades do cargo em questão.

É também apresentado um organograma com a estrutura hierárquica da organização.

**Área de Conhecimento:** (RH) Recursos Humanos.

**Perguntas-chave:**

- Que cargos existem neste projeto?
- Que elemento da equipa está atribuído a dado cargo?
- Que responsabilidades se associam a cada cargo?
- Como se esquematiza hierarquicamente a organização do projeto?

4.3.11. Identificar os Riscos

A identificação dos riscos permite perceber as condições (ameaças e/ou oportunidades) a que o projeto está sujeito.

**Área de Conhecimento:** (R) Risco.

**Perguntas-chave:**

- Que ameaças podem vir a afetar o projeto?
- Que oportunidades podem ser aproveitadas para benefício do projeto?

4.3.12. Categorizar os Riscos

A categorização dos riscos envolve atribuir um valor de probabilidade e de impacto a cada um dos riscos identificados. É construída uma matriz de riscos associada que, baseada no impacto e probabilidade de cada risco, torna visualmente mais fácil a identificação dos principais fatores de risco do projeto. Note-se que a escala de medição do impacto e probabilidade é apenas de alto nível, utilizando valores como "baixo", "médio" e "alto".

**Área de Conhecimento:** (R) Risco.

**Perguntas-chave:**

- Qual é a probabilidade de um dado risco acontecer?
- Qual o impacto das ameaças e oportunidades no projeto?

#### 4.3.13. Elaborar o Plano de Resposta aos Riscos

Após serem identificados os riscos de maior relevância ao projeto são pensadas nas melhores técnicas e soluções para gerir dado risco.

**Área de Conhecimento:** (R) Risco.

**Perguntas-chave:**

- O que fazer perante este risco?
- O que podemos fazer para prevenir dada ameaça?
- O que podemos fazer para estimular/aproveitar dada oportunidade?

#### 4.3.14. Definir os Critérios de Avaliação

Esta secção do *Project Charter* abrange os métodos de avaliação a aplicar ao resultado do projeto (acordados com o cliente).

**Área de Conhecimento:** (Q) Qualidade.

**Perguntas-chave:**

- Como vai ser avaliado o resultado final?

#### 4.3.15. Identificar os Fatores de Sucesso

É importante identificar os fatores que maximizam as hipóteses de sucesso do projeto (quando aproveitados) para que, como o nome indica, o projeto tenha maiores hipóteses de sucesso.

**Área de Conhecimento:** (Su) Sucesso.

**Perguntas-chave:**

- Que fatores são cruciais ao sucesso do projeto?
- O que deve ser feito para maximizar as hipóteses de sucesso do projeto?

#### 4.3.16. Identificar e Caracterizar os *Stakeholders*

Os *stakeholders* são pessoas e/ou organizações sobre as quais o projeto tem impacto. O impacto que estas entidades têm no projeto justifica a sua identificação formal para poder gerir a sua influência no projeto da melhor forma.

**Área de Conhecimento:** (St) *Stakeholders*.

##### **Perguntas-chave:**

- O projeto tem impacto sobre quem?
- O que pretendem obter do projeto cada um dos *stakeholders*?
- Que posição toma cada *stakeholder* em relação ao projeto?
- Que poder/influência possui cada *stakeholder* sobre o projeto?

O registo de *stakeholders* inclui a definição de cada *stakeholder* e da respetiva estratégia de gestão. É também construída a matriz de *stakeholders* em função do interesse, poder e posição de cada *stakeholder* em relação ao projeto e justificados esses parâmetros para cada um dos *stakeholder*.

Este documento relaciona-se diretamente ao sucesso do projeto, se o projeto possuir um rácio de *stakeholders* negativos (querem que o projeto falhe) muito superior aos restantes *stakeholders*, é natural que o progresso do projeto seja dificultado, o que torna vital a existência de uma gestão dos *stakeholders* eficaz.

Podem ser definidos, para cada *stakeholder*, os seguintes dados:

- Nome;
- Contactos;
- Influência no projeto;
- Interesse e posição relativa ao projeto;
- Tipo de *stakeholder*;
- Empresa/departamento a que pertence;
- Cargo e funções que desempenha;
- Expectativas em relação ao projeto;
- Estratégia de gestão do *stakeholder*, definida com base na informação anterior.

#### 4.3.17. Identificar Preliminarmente as Aquisições

A última secção do *Project Charter* envolve a identificação das aquisições necessárias ao desenvolvimento do projeto identificando também as hipóteses de obtenção do recurso e médias dos valores de mercado.

**Área de Conhecimento:** (A) Aquisições.

**Perguntas-chave:**

- Que aquisições é necessário levar a cabo?
- Que opções existem para a aquisição dos recursos necessários?

---

## FASE DE PLANEAMENTO

---

A fase de planeamento, como o próprio nome indica, envolve o planeamento de todas as vertentes relevantes ao projeto como âmbito, custo, tempo, etc. Tal ajuda a atingir com sucesso os objetivos do projeto e, durante a fase de execução, monitorizar e controlar o avanço do projeto.

Note-se que não é expectável que ao longo do projeto tudo decorra como planeado, alterações ao plano inicialmente definido são perfeitamente normais e quase impossíveis de evitar. É por esta razão que é útil recorrer a uma metodologia ágil para o desenvolvimento do projeto.

Na Figura 10 são apresentados os processos referentes à fase de planeamento.

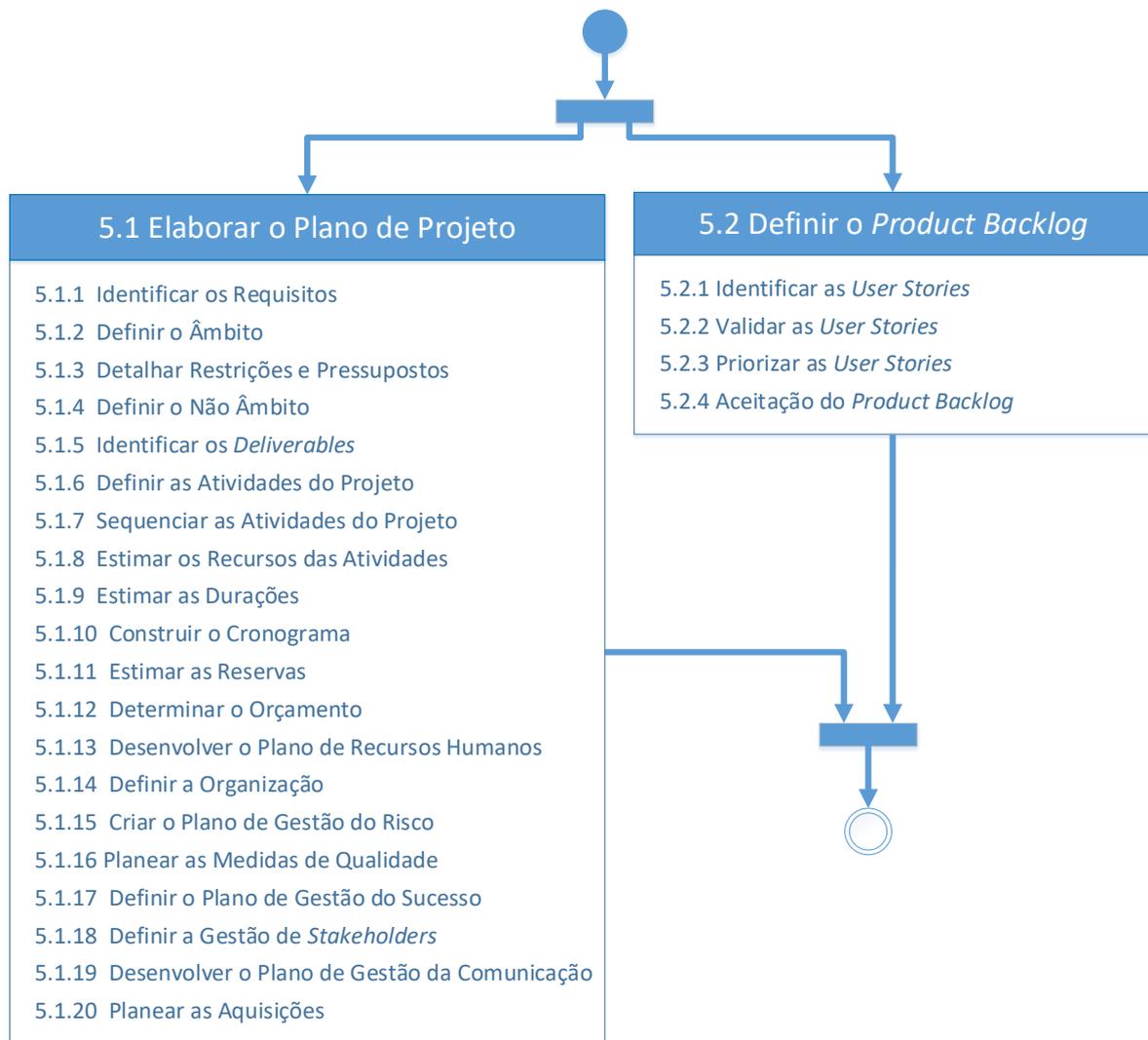


Figura 10 - Esquema da Fase de Planejamento

## 5.1. Elaborar o Plano de Projeto

O plano de projeto é construído pelo gestor de projeto e detalha o projeto em várias dimensões complementares. É este o documento utilizado pelo gestor de projeto para monitorizar e controlar o projeto durante o seu desenvolvimento.

O plano de projeto detalha vários elementos já abordados no *Project Charter*. Por esta razão, o *Project Charter* é utilizado como referência para a construção do plano de projeto.

A Tabela 4 define os processos envolvidos na construção do plano de projeto segundo áreas de conhecimento.

Âmbito (Â)
<ul style="list-style-type: none"> <li>5.1.1 Identificar os Requisitos</li> <li>5.1.2 Definir o Âmbito</li> <li>5.1.3 Detalhar as Restrições e Pressupostos</li> <li>5.1.4 Definir o Não-Âmbito</li> <li>5.1.5 Identificar os Deliverables</li> </ul>
Tempo (T)
<ul style="list-style-type: none"> <li>5.1.6 Definir as Atividades</li> <li>5.1.7 Sequenciar as Atividades</li> <li>5.1.8 Estimar os Recursos das Atividades</li> <li>5.1.9 Estimar as Durações</li> <li>5.1.10 Construir o Cronograma</li> <li>5.1.11 Estimar as Reservas</li> </ul>
Custo (C)
<ul style="list-style-type: none"> <li>5.1.12 Determinar o Orçamento</li> </ul>
Recursos Humanos (RH)
<ul style="list-style-type: none"> <li>5.1.13 Desenvolver o Plano de Recursos Humanos</li> <li>5.1.14 Definir a Organização</li> </ul>
Risco (R)
<ul style="list-style-type: none"> <li>5.1.15 Criar o Plano de Gestão de Risco</li> </ul>
Qualidade (Q)
<ul style="list-style-type: none"> <li>5.1.16 Planejar Medidas de Qualidade</li> </ul>
Sucesso (Su)
<ul style="list-style-type: none"> <li>5.1.17 Definir o Plano de Gestão do Sucesso</li> </ul>
<i>Stakeholders (St)</i>
<ul style="list-style-type: none"> <li>5.1.18 Definir a Gestão dos <i>Stakeholders</i></li> </ul>
Comunicação (Co)
<ul style="list-style-type: none"> <li>5.1.19 Desenvolver o Plano de Gestão da Comunicação</li> </ul>
Aquisições (A)
<ul style="list-style-type: none"> <li>5.1.20 Planejar Aquisições</li> </ul>

Tabela 4 - Áreas de Conhecimento Abordadas na Construção do Plano de Projeto

**Inputs:**

- *Project Charter*;
- Fatores e documentação interna relevantes;
- Fatores e documentação externa relevantes.

**Técnicas:**

As técnicas aqui descritas aplicam-se genericamente aos processos de construção do plano de projeto. No entanto, existem atividades e técnicas específicas associadas a cada um dos processos, estas são apresentadas na secção referente ao processo em questão.

- Consulta de peritos;
- Análise de documentação.

#### 5.1.1. Identificar os Requisitos

**Área de Conhecimento:** (Â) Âmbito.

Nesta fase são aplicadas técnicas de levantamento de requisitos para identificar as características e funcionalidades a desenvolver no projeto.

A listagem dos requisitos deve ser elaborada em conjunto com o *Product Owner* (PO). O PO é o representante da área de negócio do projeto, por este motivo, é da sua responsabilidade auxiliar o gestor de projeto a determinar os requisitos a incluir no projeto.

**Perguntas-chave:**

- Que funcionalidades vão ser desenvolvidas no projeto?
- Que características deve o resultado final possuir/respeitar?

**Técnicas:**

Nesta secção são apresentadas algumas técnicas de levantamento de requisitos que auxiliam o gestor a identificar os requisitos do projeto. Cada técnica apresentada tem vantagens e desvantagens em relação às outras técnicas, cabendo ao gestor identificar que técnicas utilizar segundo o ambiente do projeto em questão.

- Análise de domínio – Esta técnica é aplicada pelo gestor de projeto se este considerar que não existe conhecimento suficiente sobre o domínio do problema.

O gestor agrega informação sobre o problema e o seu domínio e, com esta informação, constrói o modelo de domínio. O modelo de domínio é uma representação do contexto do problema a partir da utilização de UML que identifica:

- Objetos do domínio – tenha-se por exemplo uma aplicação para gestão de hospitais, temos como objetos: “Doutor”, “Enfermeira”, “Paciente”, “Sala de Cirurgia”, “Consulta”, etc.;
  - Associações entre objetos – mantendo o mesmo exemplo tem-se “Doutor utiliza sala de cirurgia” como uma associação;
  - Atributos pertencentes aos objetos – continuando com o caso hipotético acima definido temos como atributos para o objeto “Paciente”: “Número do Cartão de Cidadão”, “Nome”, etc.
- Dinâmica de grupo – O gestor pode recorrer à marcação de sessões de dinâmica de grupo com as partes interessadas para identificar requisitos do projeto. *Brainstorming* é um tipo particular de sessão de dinâmica de grupo onde se procuram ideias inovadoras para incorporar no projeto.
  - Trabalho Colaborativo – Nos casos em que é impossível reunir presencialmente, invalidando sessões de dinâmica de grupo, esta técnica é alternativa. O trabalho colaborativo estabelece interação entre as partes interessadas de forma assíncrona e distribuída. Normalmente é utilizado *software* de apoio que facilita a recolha da informação e o levantamento de requisitos.
  - Entrevistas – Entrevistar as partes interessadas de forma a entender que tarefas precisam de ver cumpridas e que tipo de suporte terá o resultado final de fornecer.
  - Inquéritos – Tal como a entrevista, aplicar inquéritos a potenciais utilizadores permite recolher informação relevante ao levantamento de requisitos. Esta informação é, posteriormente, estatisticamente tratada para obter requisitos.
  - Etnografia – Estudar o ambiente dos utilizadores e perceber o seu comportamento usual permite identificar características e funcionalidades que o produto final precisará de facultar. Esta técnica pode ser feita por observação ou por aprendizagem.
  - Modelação representativa – construção de modelos como *Story Boards* ou *Personas* que retratam as necessidades dos potenciais utilizadores e permitem descobrir requisitos.

- Prototipagem – se já estiverem definidos determinados requisitos, mas o gestor considere as definições atuais demasiado vagas, pode recorrer-se à utilização de protótipos que representem soluções potenciais. O protótipo é apresentado às partes interessadas para discussão permitindo especificar a definição dos requisitos considerados vagos e potencialmente levantar outros requisitos.

### 5.1.2. Definir o Âmbito

#### **Área de Conhecimento:** (Â) Âmbito.

A definição do âmbito envolve descrever o projeto, a sua área de atuação, os seus propósitos e os seus objetivos. Tal como na identificação dos requisitos, a inclusão do *Product Owner* é vantajosa ao processo de definição do âmbito.

#### **Perguntas-chave:**

- Em que consiste o projeto?
- O que se pretende conseguir com o projeto?
- Qual é o domínio do projeto?

A definição do âmbito envolve as atividades de identificação de requisitos (5.1.1), restrições e pressupostos (5.1.3), definição do não-âmbito (5.1.4) e definição dos *deliverables* (5.1.5) apresentadas nas próximas secções.

#### **Técnicas:**

- Modelação por Diagramas – a construção de diagramas, como por exemplo diagramas de casos de uso, diagramas de contexto, etc. ajudam a definir o âmbito do projeto. Estes diagramas facilitam a compreensão da área de atuação do produto representando possíveis interações dos utilizadores com o sistema que se pretende contruir.
- Análise de Produto – através da definição do produto (conjunto de requisitos e características que o produto possui) pode ser inferido o âmbito associado ao projeto.
- Construção da *Work Breakdown Structure* (WBS) – a WBS define as atividades de trabalho necessárias para obter o resultado final do projeto. A WBS é um esquema cuja raiz corresponde ao resultado final do projeto e as folhas às

atividades (ver Figura 11). Note-se que estas atividades não são de baixo nível, a definição em detalhe é feita segundo um processo *rolling wave* suportado pela metodologia de desenvolvimento. Esta estrutura é construída na definição do âmbito para se perceber o que vai ser abordado ao longo do desenvolvimento do projeto.

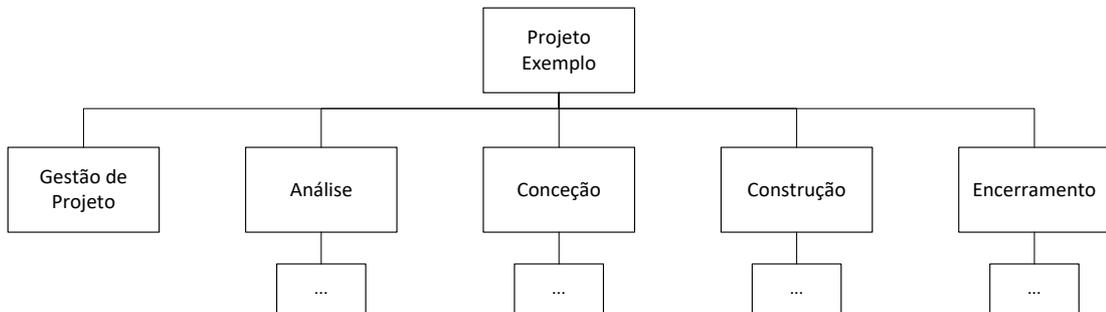


Figura 11 - Exemplo de WBS

### 5.1.3. Detalhar as Restrições e os Pressupostos

As restrições e pressupostos referentes ao projeto devem ser enumerados para completar a definição do âmbito. Tal como nas atividades anteriores, a inclusão do *Product Owner* é vantajosa e recomendada.

**Área de Conhecimento:** (Â) Âmbito.

**Perguntas-chave:**

- Que restrições tem o projeto de respeitar?
- Que fatores podemos assumir?

### 5.1.4. Definir o Não-âmbito

Definir o não-âmbito permite entender o que não vai ser abordado no projeto em causa, isto é, aquilo que fica fora do âmbito do projeto.

**Área de Conhecimento:** (Â) Âmbito.

**Perguntas-chave:**

- Que tarefas poderiam fazer sentido, mas não vão ser realizadas no projeto?

### 5.1.5. Identificar os *Deliverables*

Os *deliverables* identificam o que vai ser entregue ao cliente, clarificando o âmbito do projeto.

**Área de Conhecimento:** (Â) Âmbito.

**Perguntas-chave:**

- O que vai ser entregue ao cliente? (documentação, serviços, produtos, etc.)

**Técnicas:**

- *Product Breakdown Structure* (PBS) – a PBS é uma estrutura hierárquica em que a raiz é o resultado principal do projeto. Esta estrutura fraciona o produto final em componentes de mais baixo nível que correspondem aos *deliverables* a entregar (ver Figura 12).

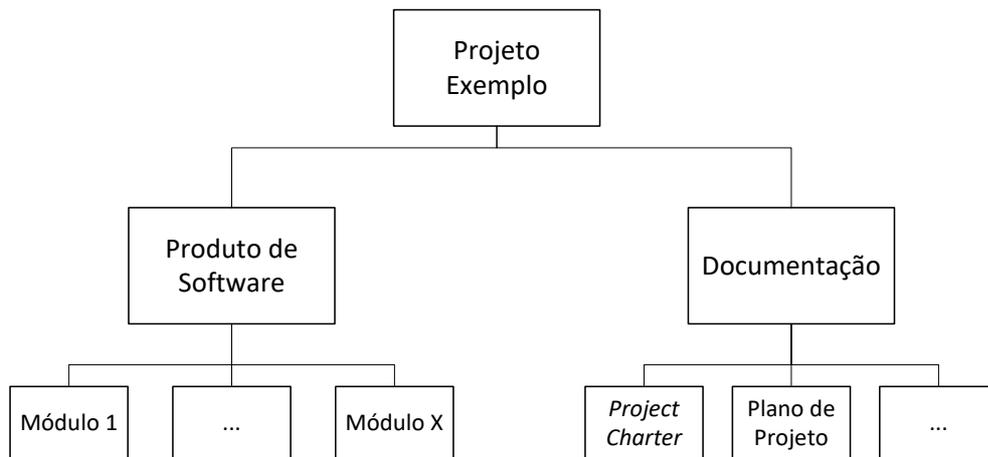


Figura 12 - Exemplo de uma PBS

### 5.1.6. Definir as Atividades

A definição das atividades permite perceber o percurso a tomar ao longo do projeto em termos de cronograma. Para além disto, é um processo crucial a outros processos de planeamento como a determinação do custo, construção do cronograma, etc.

**Área de Conhecimento:** (T) Tempo.

**Perguntas-chave:**

- Que atividades são precisas para levar a cabo o projeto?

**Técnicas:**

- *Work Breakdown Structure* (WBS) – as folhas da WBS correspondem às atividades (ver Figura 11). Apesar da sua construção ser apresentada na definição do âmbito, esta estrutura facilita também a definição das atividades de alto nível.

### 5.1.7. Sequenciar as Atividades

As atividades a sequenciar correspondem às definidas na WBS da secção anterior (5.1.6.). É preciso, para tal, identificar as dependências existentes entre as atividades. As dependências podem ser internas ou externas, e obrigatórias ou preferenciais.

Atividades que possuam dependências externas dependem de fatores externos à empresa para poderem ser executadas. O inverso acontece para as dependências internas, ou seja, existe uma dependência sobre fatores internos à empresa.

Dependências obrigatórias são dependências que não se conseguem quebrar, já dependências preferenciais são dependências que se prefere que se estabeleçam para benefício do projeto, mas podem ser quebradas se necessário.

**Área de Conhecimento:** (T) Tempo.

**Perguntas-chave:**

- Que atividades estão dependentes de outras atividades?
- Que atividades se conseguem realizar em paralelo?
- Que atividades necessitam de iniciar para outras poderem ser concluídas?
- Que atividades precisam de iniciar para outras se iniciarem?
- Que atividades só podem terminar depois outra ser terminada?
- Que atividades é preciso terminar para iniciar outras?

## Técnicas:

- Nomenclatura de Dependências – esta nomenclatura aborda todos os casos de dependências que possam existir entre atividades, facilitando a classificação das mesmas. É estabelecida por:
  - FS (*Finish-to-Start*) define uma relação em que a atividade seguinte só se inicia depois da anterior ser concluída;
  - SS (*Start-to-Start*) define que uma atividade só pode iniciar quando outra for também iniciada;
  - SF (*Start-to-Finish*) define que uma atividade só pode terminar quando outra se iniciar;
  - FF (*Finish-to-Finish*) define que uma atividade só pode terminar quando a outra for terminada.

A notação gráfica representativa desta nomenclatura é apresentada na Figura 13.

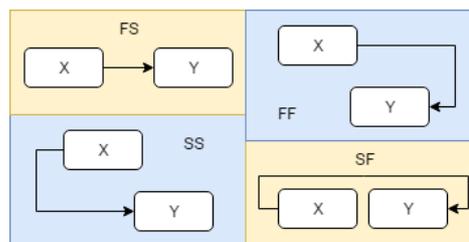


Figura 13 - Notação Gráfica da Nomenclatura de Dependências

### 5.1.8. Estimar os Recursos das Atividades

Com base nas atividades de projeto identificadas, estimam-se os recursos necessários à completude de cada atividade.

Note-se que, no *Project Charter*, a estimativa de recursos era situada na área de conhecimento do custo. Neste caso, como o processo é realizado à custa das atividades identificadas na área de conhecimento do tempo, é a esta área de conhecimento que fica associado o processo.

**Área de Conhecimento:** (T) Tempo.

#### Perguntas-chave:

- Que recursos são necessários?

## Técnicas:

- Construção da *Resource Breakdown Structure* (ReBS) – tal como a WBS e a PBS, a ReBS é uma estrutura composta por uma raiz e várias folhas, mas com um método de decomposição diferente. A ReBS decompõe o resultado final em recursos (ver Figura 14).

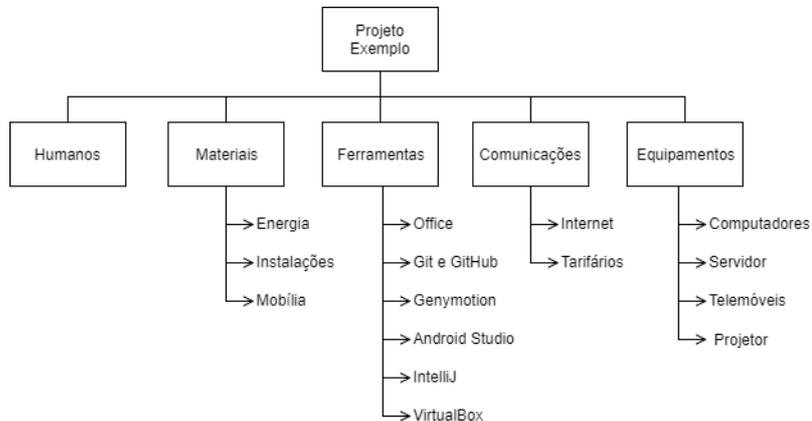


Figura 14 - Exemplo de ReBS

### 5.1.9. Estimar as Durações

Nesta fase são estimadas as durações de cada uma das atividades para que, posteriormente, seja construído o cronograma do projeto. São também calculadas as reservas necessárias.

**Área de Conhecimento:** (T) Tempo.

#### Perguntas-chave:

- Quanto tempo demora a realizar dada atividade?

## Técnicas:

- Analogia – se já existir um histórico de atividades e de projetos (base de conhecimento), pode utilizar-se esse historial para criar as estimativas. Mesmo que as atividades não sejam idênticas, se existirem fatores semelhantes é possível conseguir uma estimativa baseada nos dados dos projetos.
- Paramétrica – esta técnica, tal como a analogia, utiliza o historial obtido de outros projetos para o cálculo das estimativas. Porém, ao contrário da analogia, que é

obtida por julgamento, esta técnica utiliza parâmetros concretos do historial para, através de cálculos, estimar a duração da atividade do projeto.

- Três pontos (PERT) – a estimativa de três pontos utiliza três estimativas diferentes para determinar a duração média que se espera que a atividade tenha. Assim, são realizadas três estimativas, uma otimista ( $T_o$ ), uma pessimista ( $T_p$ ) e uma que se considera a mais provável ( $T_m$ ) e, a partir destes valores, é calculada a duração estimada (DE) através da seguinte expressão:  $DE = \frac{T_o + 4T_m + T_p}{6}$ . Este tipo de técnica de estimação permite diminuir o risco e incerteza da estimativa.
- Quantitativa – tendo em conta a quantidade de trabalho a realizar numa dada atividade ( $w$ ) e a produtividade de um recurso ( $p$ ), é possível calcular o esforço ( $e$ ) associado à realização dessa atividade através da expressão:  $e = \frac{w}{p}$ . Com o esforço ( $e$ ) é possível estimar a duração da atividade (DE):  $DE = \frac{e}{N^{\circ} \text{ Recursos}}$ .

#### 5.1.10. Construir o Cronograma

A construção do cronograma é feita tendo em conta as atividades, a sua sequenciação e as suas durações obtidas nas secções anteriores.

**Área de Conhecimento:** (T) Tempo.

**Perguntas-chave:**

- Qual é o cronograma do projeto?

**Técnicas:**

- Diagrama de *Gantt* – o diagrama de *Gantt* é um gráfico representativo do calendário do projeto. Neste gráfico são expostas as atividades (definindo o seu início, fim e duração associada), as dependências existentes entre as atividades e as *milestones* do projeto. A Figura 15 apresenta um exemplo minimalista de um diagrama de *Gantt*.

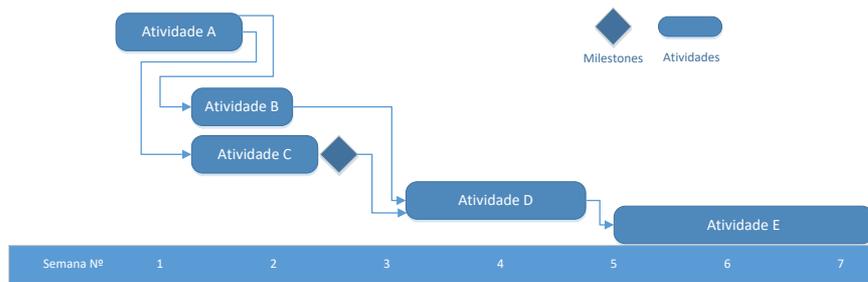


Figura 15 - Diagrama de *Gantt*

- Diagrama de Rede – os diagramas de rede são grafos utilizados para representar o cronograma do projeto. Podem ser diagramas de precedência ou diagramas de setas. A diferença destes dois tipos de diagrama é a sua representação, o primeiro possui as atividades nos nós do gráfico, já o segundo possui as atividades nos arcos ou setas que conectam os nós. Os diagramas de precedência possuem a seguinte notação nos nós:
  - *Earliest Start Time* (ES) – Data de início mais cedo;
  - *Earliest Finish Time* (EF) – Data de término mais cedo;
  - *Latest Start Time* (LS) – Data de início mais tarde;
  - *Latest Finish Time* (LF) – Data de término mais tarde;
  - Folga – diferença entre LF e EF;
  - Identificação (Id);
  - Duração (D).

Os nós do diagrama são apresentados com a estrutura que se observa na Figura 16.



Figura 16 - Nomenclatura do Diagrama de Precedência

A construção do diagrama é feita como descrito na Figura 17. Este diagrama permite obter: o caminho crítico, caminho mais longo da rede formado pelas atividades em que  $ES=LS$ ; as folgas livres, folgas que uma atividade pode usufruir sem atrasar a atividade que lhe sucede; e as folgas totais, folga que uma atividade tem até atrasar o projeto.

O cálculo do caminho crítico permite perceber quais as atividades que se devem monitorizar com mais cuidado. Estas atividades correspondem às atividades a que se devem aplicar reservas (quando necessário), por terem impacto direto no atraso do projeto.

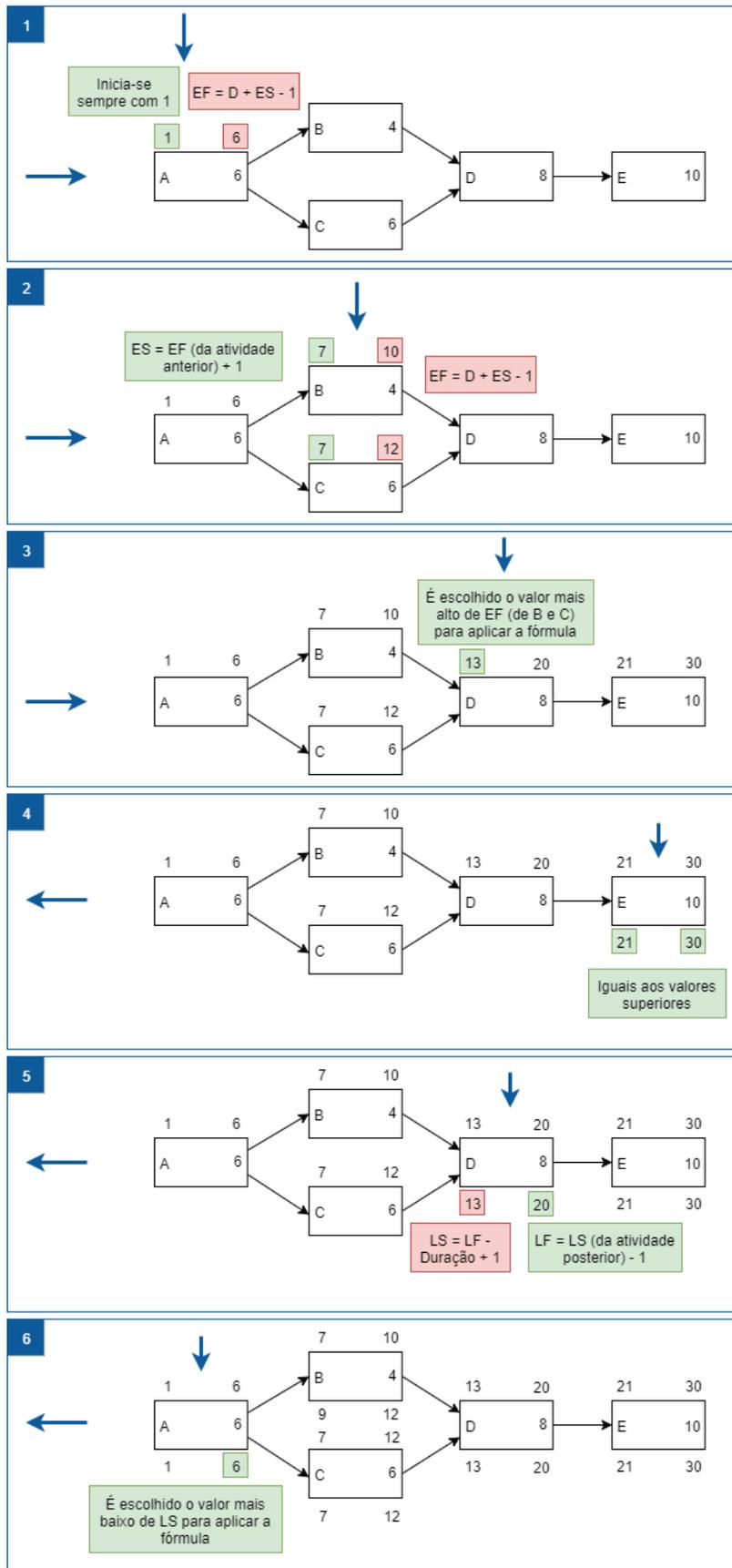


Figura 17 - Construção do Diagrama de Precedências

#### 5.1.11. Estimar as Reservas

As reservas são um plano de contingência para responder a possíveis atrasos nas atividades do projeto. Aplicar reservas às atividades que compõe o caminho crítico permite minimizar o risco do projeto se atrasar.

**Área de Conhecimento:** (T) Tempo.

**Perguntas-chave:**

- Quanto tempo de reserva deve ser atribuído à atividade?

**Técnicas:**

As técnicas de estimativa por analogia, paramétrica e de três pontos aplicáveis na estimativa das durações são também aplicáveis no cálculo das reservas.

- Análise de reservas – a análise de reservas compara a quantidade atribuída de um dado recurso com o risco de esta quantidade não ser suficiente. Aplicada às atividades, permite determinar se deve, ou não, ser aplicada uma reserva a essa atividade. Se já existir uma reserva associada à atividade, permite identificar se a reserva é suficiente.

#### 5.1.12. Determinar o Orçamento

O orçamento do projeto tem em conta os recursos estimados na secção 5.1.8 e as atividades identificadas em 5.1.6.

**Área de Conhecimento:** (C) Custo.

**Perguntas-chave:**

- Qual é o orçamento do projeto?

**Técnicas:**

- Analogia – tal como para as durações, é possível utilizar a analogia para estimar o orçamento do projeto (ver 5.1.9 – Analogia).

- Paramétrica – a análise paramétrica é também utilizada na estimativa do orçamento, utilizando o histórico de projetos para calcular o orçamento (ver 5.1.9 – Paramétrica).
- Três pontos (PERT) – a utilização da técnica de três pontos é também uma opção para a estimativa do orçamento (ver 5.1.9 – Três pontos (PERT)).
- *Bottom-up* – a estimativa *bottom-up* envolve a utilização da WBS construída. É feita a estimativa de recursos individualizada para cada folha da estrutura (*work-package*) e, subindo de nível, são somados os recursos das atividades “filhas” de forma a obter na raiz os recursos necessários ao projeto. Associando aos recursos os seus custos, é possível calcular o orçamento do projeto.
- Análise de reservas – a análise de reservas pode ser utilizada para o cálculo do orçamento, sendo que os recursos em análise passam a ser os recursos necessários.
- Análise de propostas – se existir um ou mais recursos obtidos por uma aquisição, a análise de propostas dos fornecedores permite calcular os custos associados à obtenção do(s) recurso(s).

O orçamento pode também ser apresentado através de uma perspetiva temporal e através de uma perspetiva de *Work-Package*. A perspetiva temporal apresenta a utilização dos recursos dividida em intervalos de tempo (normalmente semanas). A perspetiva de *Work-package* apresenta a utilização dos recursos em função das atividades do projeto. Este tipo de gráficos facilita, posteriormente, o trabalho de monitorização e controlo dos custos do projeto.

#### 5.1.13. Desenvolver o Plano de Recursos Humanos

Nesta secção são estabelecidos horários de trabalho, normas e regras a cumprir, necessidades de formação (caso existam) e sistemas de reconhecimento e/ou recompensa. Para além disto, é definida a organização, detalhada na secção 5.1.14. Em projetos internacionais é necessário ter em conta as diferenças culturais que possam existir [43, 44].

**Área de Conhecimento:** (RH) Recursos Humanos.

**Perguntas-chave:**

- Qual será o horário de trabalho?
- Que regras e normas deve a equipa cumprir?

- A equipa tem todo o conhecimento necessário para desenvolver o projeto?
- Que comportamentos/ações devem ser recompensados e/ou reconhecidos?
- Que recompensas e/ou reconhecimentos vão existir?

**Técnicas:**

- Reuniões – reuniões entre equipa e respetivo gestor, onde o gestor procura que os elementos da equipa respondam às perguntas-chave acima expostas. É vantajoso que a equipa tenha influência na decisão dos sistemas de recompensa e reconhecimento, por incitar à motivação dos elementos, promovendo o seu desempenho e produtividade.
- Avaliações – para identificar se existem necessidades de formação podem ser realizadas avaliações de competências e conhecimento. No entanto, esta técnica pode não ser bem-recebida pela equipa. Cabe ao gestor explicar que o objetivo é perceber se são necessárias formações e que as avaliações não afetam negativamente nenhum dos avaliados.

5.1.14. Definir a Organização

A definição da organização faz parte da conceção do plano de recursos humanos e permite identificar os diversos cargos e responsabilidades de cada um dos elementos da organização do projeto.

**Área de Conhecimento:** (RH) Recursos Humanos.

**Perguntas-chave:**

- Como se estabelece a organização da equipa do projeto?

**Técnicas:**

- *Organizational Breakdown Structure* (OBS) – a OBS identifica os recursos humanos envolvidos no projeto;
- Organograma – o organograma apresenta hierarquicamente a organização do projeto. Os elementos que constituem o organograma correspondem às folhas da OBS.
- Matriz RAM – a matriz RAM detalha as responsabilidades de cada um dos elementos da organização ao longo das diversas atividades do projeto.

### 5.1.15. Criar o Plano de Gestão de Risco

O plano de gestão de risco apresenta a listagem categorizada dos riscos e a respetiva avaliação qualitativa e quantitativa.

A avaliação qualitativa ordena os riscos segundo o seu impacto e probabilidade, a avaliação quantitativa associa a cada risco um valor monetário representativo. Com base nos valores da avaliação quantitativa são também apresentados os valores de melhor (todas as oportunidades se sucedem) e pior caso (todas as ameaças se sucedem) do projeto.

O plano de resposta ao risco que associa a cada risco (ameaça e oportunidade) uma resposta é também exposto nesta secção.

**Área de Conhecimento:** (R) Risco.

#### **Perguntas-chave:**

- Que ameaças existem ao sucesso do projeto?
- Que oportunidades podem ser aproveitadas para benefício do projeto?
- Que tipo de riscos existem no projeto?
- Qual é o impacto dos riscos no projeto?
- Qual a probabilidade de dado risco acontecer?
- Qual é o melhor e pior cenário do projeto em relação aos riscos?
- Que respostas ao risco podem ser utilizadas?

#### **Técnicas:**

- Analogia – podem ser identificados riscos através da análise de projetos similares.
- Análise de falhas – analisar falhas de outros projetos identificando os fatores que levaram ao seu fracasso ajuda a identificar riscos relevantes.
- Análise de Cenários – descrever possíveis cenários que o projeto possa tomar pode auxiliar na identificação dos riscos do projeto.
- *Risk Breakdown Structure* (RiBS) – a construção da RiBS estabelece os riscos do projeto numa estrutura hierarquizada.

Os riscos podem, tal como os custos, ser apresentados por perspetivas temporais e de *Work-package*. Estas perspetivas auxiliam na monitorização e controlo dos riscos do projeto.

#### 5.1.16. Planear as Medidas de Qualidade

Nesta secção do plano de projeto são definidos os padrões de qualidade e testes de conformidade necessários para garantir a qualidade do produto ou serviço desenvolvido no projeto. Pode ser definida a periodicidade de verificação de qualidade do resultado, por tempo ou por eventos, por exemplo: a cada quatro dias (tempo) ou no final de cada iteração (evento). Note-se que, a utilização de uma metodologia iterativa onde são priorizadas as tarefas a desenvolver e se obtém *feedback* constante, facilita o controlo e monitorização da qualidade do produto ou serviço a ser desenvolvido.

**Área de Conhecimento:** (Q) Qualidade.

**Perguntas-chave:**

- Que normas e padrões devem ser seguidos?
- O que define a qualidade do resultado final?
- Que testes podem ser aplicados para validar a qualidade do resultado?

**Técnicas:**

- Análise Custo-Benefício – analisar os custos inerentes a um requisito de qualidade face ao benefício que advém da sua verificação, permite perceber o valor associado à sua obtenção.
- *Benchmarking* – definir medidas de *benchmarking* desejadas e, durante o desenvolvimento, aplicar as medidas como controlo de qualidade.
- Custo de Qualidade – é calculada a soma dos custos associados à qualidade, ou seja, custos de falha, custos de prevenção, custos de avaliação, etc. e, de seguida, é calculado o custo associado à não conformidade da qualidade. Com isto obtém-se o balanço associado a cada uma das medidas de qualidade.
- Ferramentas de Qualidade – como diagramas de causa-efeito, diagramas de fluxo, diagramas de *Pareto*, histogramas, etc. Estas ferramentas são utilizadas genericamente na resolução de problemas e podem, por isto, ser aplicadas à resolução de problemas de qualidade ao longo do projeto.

#### 5.1.17. Definir o Plano de Gestão do Sucesso

Devem ser definidos os fatores de sucesso do projeto e detalhados os procedimentos a abordar para maximizar as hipóteses de sucesso do projeto [45, 46].

**Área de Conhecimento:** (Su) Sucesso.

**Perguntas-chave:**

- Que fatores influenciam o sucesso do projeto?
- O que deve ser feito para maximizar as hipóteses de sucesso do projeto?

**Técnicas:**

- Reuniões e entrevistas – nestas reuniões o gestor de projeto procura identificar que fatores de sucesso são identificados pelos *Stakeholders*.
- Diagrama de Sucesso – construção de um diagrama que para cada fator de sucesso identificado defina as técnicas a utilizar para maximizar o fator de sucesso e as métricas de medição de performance dessas técnicas (para efeitos de monitorização e controlo). Por exemplo, um fator de sucesso “Apoio do cliente”, terá como técnica “Reunião semanal” e, associada à técnica uma métrica de medição “Presenças do cliente na reunião semanal”.

#### 5.1.18. Definir a Gestão dos *Stakeholders*

Aqui são identificados e enunciados os *stakeholders* do projeto detalhando a sua posição e poder.

**Área de Conhecimento:** (St) *Stakeholders*.

**Perguntas-chave:**

- Sobre quem é que o projeto tem impacto?
- Que posição têm os *stakeholders* em relação ao projeto?
- Que influência têm os *stakeholders* no projeto?
- Qual o poder de cada *stakeholder* no projeto?
- Como se distribui a influência e poder de cada *stakeholder* ao longo do desenvolvimento do projeto?

### **Técnicas:**

- Reuniões e entrevistas – a existência de reuniões e entrevistas com os *stakeholders* ajuda o gestor de projeto, não só a perceber o que cada *stakeholder* pretende obter (identificação de requisitos), mas também a definir o seu interesse e posição no projeto.
- Matriz de *stakeholders* – a matriz de *stakeholders* apresenta a posição e poder de cada um dos *stakeholders* em relação ao projeto.

É vantajoso que sejam construídas as perspetivas temporal e de *Work-package*, indicando ao longo do tempo e em relação às atividades que *stakeholders* devem ser considerados.

#### 5.1.19. Desenvolver o Plano de Gestão da Comunicação

É detalhado para cada parte interessada o tipo de informação que esta necessita, a periodicidade com que a deve receber e a forma e condições associadas à entrega da informação. É necessário ter em conta as diferenças culturais que possam existir [43, 44].

Note-se que, nesta secção, devem estar incluídas as reuniões referentes à metodologia de desenvolvimento.

**Área de Conhecimento:** (Co) Comunicação.

### **Perguntas-chave:**

- Como vão ser contactados os *stakeholders*?
- Que tipo de informação é necessário fornecer?
- Que necessidades de informação tem cada *stakeholder*?
- Que tecnologias serão utilizadas para dar suporte à comunicação?
- Que periodicidade é necessária na passagem de informação aos diversos *stakeholders*?

### **Técnicas:**

- Reuniões e entrevistas – nestes encontros deve procurar-se resposta às perguntas-chave acima apresentadas, de forma a entender a necessidade de obtenção de informação do *stakeholder*.

- Análise de Requisitos de Comunicação – este tipo de análise determina a necessidade de informação de cada *stakeholder* através da combinação do tipo e formato das informações com o valor das informações. Para isto são tidos em conta dados como organogramas, esquemas da organização do projeto, necessidades de informação, etc. Um indicador para avaliar a complexidade da comunicação do projeto é obtido pela seguinte expressão *Nº de Canais* =  $\frac{n(n-1)}{2}$ , onde n é o número *stakeholders*.
- Modelos de comunicação – estes modelos são construídos para cada comunicação. Os modelos variam e definem diferentes detalhes desde o regras a seguir para garantir uma comunicação efetiva ou que informações devem ser transmitidas na comunicação, etc.

Um exemplo de um modelo de comunicação é o modelo de Lasswell que se constrói respondendo às seguintes questões:

- Quem? – Quem será responsável pela transmissão da mensagem;
- O quê? – Que informação será fornecida;
- Através de que meio? – Que meio será usado para transportar a informação até ao recetor;
- A quem? – Quem vai receber a informação;
- Com que efeito? – Para que se estabelece a comunicação, o que é esperado que seja conseguido.

#### 5.1.20. Planear Aquisições

Esta secção detalha as diversas aquisições necessárias, apresentado para cada aquisição o seu custo, circunstâncias, dados do fornecedor, etc.

Durante esta fase de planeamento das aquisições são formados os contratos para a obtenção de recursos necessários ao projeto. Existem vários tipos de contratos, cabe ao gestor de projeto, utilizando técnicas de negociação e investigação, encontrar a opção mais lucrativa.

**Área de Conhecimento:** (A) Aquisições.

#### **Perguntas-chave:**

- Qual é a melhor opção para obter dado recurso?
- Que contratos são precisos formar?

- Que alternativas existem para a obtenção do recurso?

### **Técnicas:**

- Análise *Make or Buy* – a análise *make or buy* determina se é mais lucrativo comprar o recurso ou produzi-lo internamente. A escolha deve pesar os custos diretos e os custos indiretos incluindo restrições de produção e de compra (por exemplo: orçamento limite ou contratos já existentes na empresa).
- Análise do Mercado – a análise de mercado recorre à investigação da indústria e de vendedores para determinar os preços associados à compra de um recurso. Esta análise, para além do preço do recurso, tem em conta todos os fatores relacionados com a compra que possam ter impacto no projeto, por exemplo, o preço dos materiais para produção do recurso, a facilidade de obtenção do recurso, riscos associados à compra, etc.

Deve ser construída a perspetiva temporal e de *Work-package* relacionada com as aquisições, que corresponderá à necessidade das aquisições ao longo do tempo e das atividades do projeto, respetivamente.

## **5.2. Definir o *Product Backlog***

A construção do *Product Backlog* é da responsabilidade do *Product Owner*. O PO, tendo por base os requisitos levantados e o âmbito definido no plano de projeto, realiza a construção do *Product Backlog*.

Sendo que o PO é o especialista de negócio e o representante das necessidades do cliente, a sua presença nos processos relacionados com o âmbito do projeto é vantajosa. A sua integração nos processos agiliza a construção do *Product Backlog* e evita alterações desnecessárias ao plano de projeto.

Tal como o plano de projeto, o *Product Backlog* é detalhado e melhorado ao longo do projeto. Inicialmente o *Product Backlog* apenas deve conter objetivos funcionais, ou seja, tarefas que os utilizadores necessitam que o produto realize.

A Tabela 5 apresenta os processos da definição do *Product Backlog* divididos por áreas de conhecimento.

Âmbito (Â)
5.2.1 Identificar as <i>User Stories</i>
5.2.2 Validar as <i>User Stories</i>
5.2.3 Priorizar as <i>User Stories</i>
5.2.4 Aceitação do <i>Product Backlog</i>
Tempo (T)
5.2.3 Priorizar as <i>User Stories</i>
Custo (C)
5.2.3 Priorizar as <i>User Stories</i>
Risco (R)
5.2.3 Priorizar as <i>User Stories</i>

Tabela 5 - Áreas de Conhecimento Abordadas na Construção do *Product Backlog*

### Inputs:

- Plano de projeto – o *Product Backlog* deve ser construído paralelamente ao desenvolvimento do plano de projeto. A integração do *Product Owner* nas atividades relacionadas com o âmbito facilita a construção do *Product Backlog*. Desta forma, *input* pode ser um termo demasiado restritivo, mas reforça-se a necessidade de consistência entre o plano de projeto e o *Product Backlog*;
- Fatores e documentação interna relevantes;
- Fatores e documentação externa relevantes.

### Técnicas:

As técnicas aplicadas à identificação de requisitos são úteis para a construção do *Product Backlog* e, com o PO integrado no processo de definição dos requisitos e âmbito, a construção do *Product Backlog* é agilizada.

- Consulta de peritos;
- Análise de documentação.

### 5.2.1. Identificar as *User Stories*

O primeiro passo para construir o *Product Backlog* consiste em identificar as atividades que os utilizadores realizam, ou seja, perceber quais são as necessidades que o resultado final do projeto deve satisfazer.

**Área de Conhecimento:** (Â) Âmbito.

**Perguntas-chave:**

- Que necessidades existem?
- O que gostaria o utilizador de conseguir ver realizado?
- Que atividades precisa de realizar para cumprir o seu trabalho?
- Que tipo de suporte precisa nas suas atividades?

**Técnicas:**

- Reuniões e entrevistas – a comunicação com os *Stakeholders* e futuros utilizadores do resultado final do projeto, permite entender as necessidades a que se pretende que o resultado responda.
- Técnicas de Levantamento de Requisitos (ver secção de técnicas em 5.1.1) – como já foi referido as técnicas utilizadas no levantamento de requisitos são úteis na identificação das atividades.

As atividades são descritas como *User Stories* constituídas por um título elucidativo e uma descrição que respeita a seguinte estrutura:

“Como [Cargo], preciso/gostaria de [atividade] para [efeito].”

Em que *cargo* corresponde ao cargo do utilizador e *efeito* é o motivo para realizar a atividade, ou seja, o efeito que pretende obter com a realização da atividade.

As *User Stories* são abertas a consulta, isto é, qualquer *stakeholder* as pode visualizar. Isto incita à participação, propaga conhecimento e aumenta a confiança dos *stakeholders* no projeto. Por este motivo as *User Stories* devem ter uma linguagem clara e simples.

As *User Stories* auxiliam assim à definição do âmbito do projeto. Note-se que, no desenvolvimento do plano de qualidade (5.1.16), devem ser incorporadas, sempre que possível, medidas de qualidade específicas a cada uma das *User Stories* identificadas.

### 5.2.2. Validar as *User Stories*

Após terem sido construídas as *User Stories*, o PO deve apresentar ao gestor de projeto a listagem final para validação. Sendo que o gestor de projeto esteve, também, em contacto com os *stakeholders* do projeto, este deve ser capaz de, ao analisar a listagem, identificar *User Stories* em falta, *User Stories* desnecessárias e/ou *User Stories* pouco claras.

**Área de Conhecimento:** (Â) Âmbito.

#### **Perguntas-chave:**

- As *User Stories* agregam todos os requisitos definidos no plano de projeto?
- Existe alguma *User Story* pouco clara ou inadequada?

#### **Técnicas:**

- Mapeamento de Requisitos – analisando os requisitos definidos no plano de projeto é possível mapear as diversas *User Stories* aos requisitos e, com isto, identificar *User Stories* em falta, desnecessárias ou incorretas.

### 5.2.3. Priorizar as *User Stories*

Depois de obtida a listagem completa das *User Stories*, cabe ao *Product Owner* priorizar a lista construindo, assim, o *Product Backlog*.

A priorização das *User Stories* é inserida nas seguintes áreas de conhecimento: âmbito, por identificar que porções do âmbito total são realmente valiosas e que requisitos supérfluos podem ser dispensados ou implementados posteriormente; tempo e custo, por orientar a disponibilidade da equipa às funcionalidades mais importantes, originando um melhor aproveitamento do tempo e orçamento disponível; e risco por minimizar, entre outros, os riscos sobre a alteração de requisitos, (dado que a probabilidade de alterações é mais baixa nas funcionalidades mais importantes e mais alta nas funcionalidades complementares) e por maximizar a probabilidade de satisfação do cliente ao apresentar protótipos iniciais com as funcionalidades que se consideram mais relevantes.

**Área de Conhecimento:** (Â) Âmbito; (T) Tempo; (C) Custo; (R) Risco.

## Perguntas-chave:

- Quais as *User Stories* de maior importância?
- Que *User Stories* trazem mais valor ao resultado final?

## Técnicas:

O *Product Owner* utiliza técnicas de priorização para entender o valor de cada uma das *User Stories*. Estas técnicas aplicam-se a grupos de *stakeholders*, particularmente a potenciais utilizadores, para obter informação mais precisa sobre o valor pertencente a cada *User Story*.

- MoSCoW – a técnica de MoSCoW consiste na divisão das *User Stories* em quatro grupos:
  - *Must have* – requisitos críticos que têm de ser implementados;
  - *Should have* – requisitos importantes que devem ser implementados;
  - *Could have* – requisitos desejáveis, mas não necessários;
  - *Won't have* – requisitos dispensáveis.Esta técnica é utilizada como primeira abordagem, porém, posteriormente, é aplicada uma outra técnica de priorização mais divisiva para obter a priorização total da listagem. Normalmente é apenas priorizado o grupo mais valioso ("Must Have").
- Top 10 – cada *stakeholder* escolhe as 10 *User Stories* que considera mais importantes.
- 100 Unidades – cada *stakeholder* possui 100 unidades para distribuir pelos requisitos, sendo que deverá atribuir mais unidades aos requisitos que considera mais importantes. O número de unidades deve ser ajustado ao número de *User Stories*.
- Método de Kano – pede-se aos *stakeholders* que dividam as *User Stories* em:
  - Obrigatórias – têm de estar presentes para que as necessidades dos *stakeholders* sejam satisfeitas;
  - Lineares – quanto mais destas *User Stories* forem implementadas mais valioso será o resultado;
  - Desejáveis – características que os *stakeholders* não identificam, mas, quando referidas, apoiam a sua implementação.

Com isto é possível construir o gráfico representativo do impacto das *User Stories* na satisfação dos utilizadores, Figura 18 - Gráfico do Método de Kano.

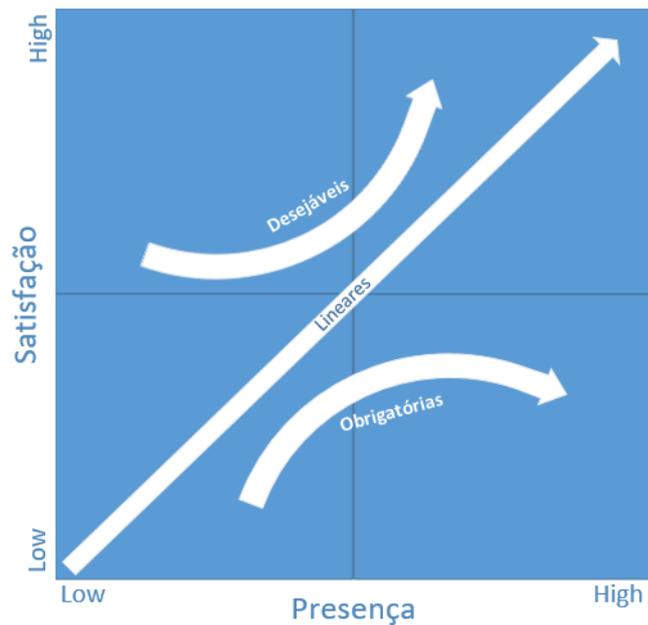


Figura 18 - Gráfico do Método de Kano

- AHP – o método AHP utiliza uma escala numérica de 1 a 9 para classificar a importância de uma *User Story* em relação a outra. Neste caso 1 define “igual importância” e 9 extremamente “mais importante que”. Por exemplo, tenha-se a comparação entre a *User Story* 1 e a *User Story* 2: se for atribuído um 5 à *User Story* 1 significa que esta é mais importante que a *User Story* 2; se for atribuído um 7 significa que a *User Story* 1 é muito mais importante que a *User Story* 2. Com isto, é construída uma matriz de *User Story* x *User Story* e preenchida com os valores das comparações. Posteriormente, é feito o somatório das colunas e são normalizados os valores da matriz. De seguida, é feito o somatório das linhas e, com isto, obtém-se a ordem de importância das *User Stories*. Sendo que as linhas correspondem a uma *User Story*, o valor obtido no somatório dessa linha é o valor numérico de importância da *User Story*. As *User Stories* são ordenadas segundo esses valores para obter a listagem priorizada. Esta técnica deve ser aplicada a um número pequeno de *User Stories*. Uma boa estratégia é aplicar uma técnica de agrupamento, como, por exemplo MoSCoW, antes da aplicação da AHP.

#### 5.2.4. Aceitação do *Product Backlog*

Depois de obtido o *Product Backlog*, é preciso obter a aceitação do mesmo por parte do cliente. Assim, deve-lhe ser apresentada a listagem obtida para discussão da sua validade. Nesta fase, pode ser necessário reformular determinados detalhes em função da opinião do cliente. É relevante alertar o cliente de que a ordenação e detalhes apresentados podem sofrer alterações ao longo do projeto, sem refletir a introdução de custos adicionais.

Esta aceitação é uma primeira validação para perceber a orientação a tomar no desenvolvimento do projeto. A aceitação não deve ser demasiado rígida e não é expectável que todos os detalhes das *User Stories* sejam esclarecidos.

Sendo que a aceitação do *Product Backlog* se enquadra na área de conhecimento do âmbito, é de notar que, esta deve planeada na gestão de risco (5.1.15), na gestão de *stakeholders* (5.1.18) e na gestão de comunicação (5.1.19).

**Área de Conhecimento:** (Â) Âmbito.

---

## FASE DE EXECUÇÃO

---

É na fase de execução que se dá o desenvolvimento do resultado final do projeto. Esta fase é realizada em simultâneo com a fase de monitorização e controlo para garantir que o projeto é bem-sucedido e que o resultado final respeita as expectativas e necessidades do cliente.

Nesta fase, são expectáveis alterações ao plano inicialmente traçado. A utilização de uma metodologia de desenvolvimento ágil promove a flexibilidade do projeto de forma a tirar partido dessas alterações. Obtém-se, por isto, um resultado final mais adequado às necessidades do cliente. A metodologia garante que o trabalho realizado em primeiro é o de maior importância (priorização) e, com as validações existentes no final de cada iteração (*feedback*), que as alterações são detetadas e implementadas o mais cedo possível evitando desperdícios de recursos e um resultado inadequado.

As subfases de planeamento, execução e conclusão da *sprint* reforçam a possibilidade de conciliação de metodologias de gestão de projetos com metodologias ágeis para aproveitamento das vantagens de ambas.

Na Figura 19 é apresentado o esquema de processos associado à fase de execução.

Localização temporal adaptável: - - - -

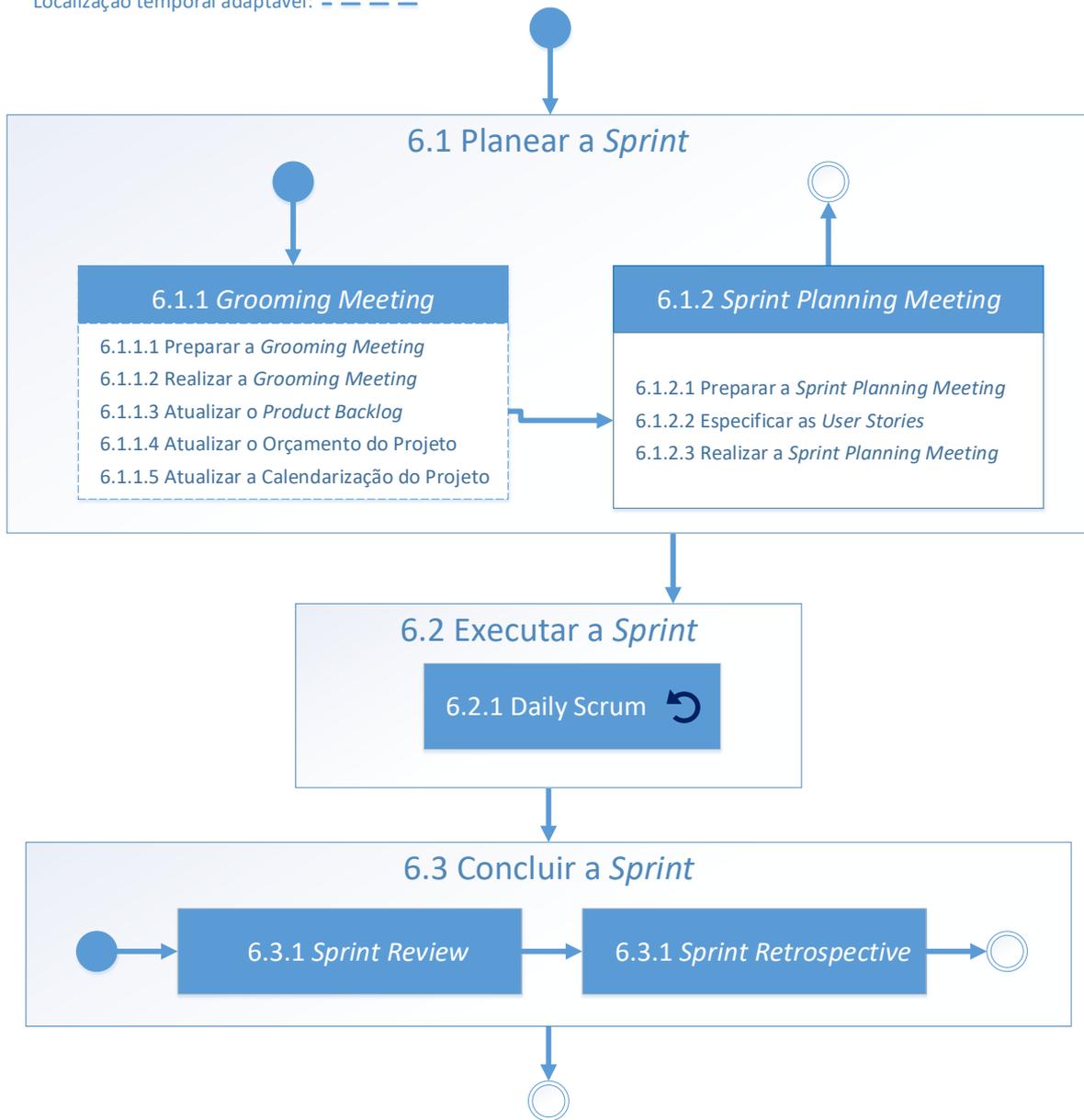


Figura 19 - Esquema da Fase de Execução

## 6.1. Planear a *Sprint*

No início de cada iteração é necessário planear o que será desenvolvido na mesma. Assim é obrigatoriamente realizada uma *Sprint Planning Meeting* e, caso seja necessária, uma *Grooming Meeting*. Nestas reuniões são esclarecidas dúvidas e definidas prioridades para conseguir realizar uma implementação correta e priorizada.

### 6.1.1. *Grooming Meeting*

A *Grooming Meeting* é uma reunião aberta a todas as partes interessadas onde se discute e detalha o *Product Backlog*. Esta reunião não é uma reunião formal do *Scrum*, mas tem vindo a ser defendida e incorporada na metodologia por promover confiança no projeto através da comunicação entre os vários *stakeholders*.

Estas reuniões são especialmente importantes no início da fase de execução por existir um grau superior de incerteza e passam a opcionais face a baixos graus de incerteza. A localização temporal da reunião deve ser adaptada à equipa (note-se o contorno segmentado no processo de *Grooming Meeting* – ver Figura 19). Realizar a *Grooming Meeting* no início de cada iteração auxilia na preparação da *sprint planning meeting* e realizar a reunião durante o *sprint* auxilia na validação do trabalho a decorrer.

Deve existir incentivo para que os participantes se expressem livremente e opinem sobre que *User Stories* pensam ser prioritárias e, conseqüentemente, devam ser implementadas na iteração seguinte. Não é expectável que todos os participantes tenham conhecimento sobre o domínio técnico da solução, por isso, é importante que as *User Stories* utilizem linguagem corrente e simples, para que qualquer *stakeholder* a consiga compreender.

O *Product Owner* é responsável pela marcação e gestão das *Grooming Meetings*. A presença do gestor de projeto é obrigatória, todas as outras presenças são opcionais, mas recomendadas. O gestor de projeto, nesta reunião, analisa a adequação do plano atual, realizando as alterações necessárias.

Na Tabela 6 são listados os processos inerentes à *Grooming Meeting* segundo áreas de conhecimento.

Âmbito (Â)
6.1.1.2 Realizar a <i>Grooming Meeting</i> 6.1.1.3 Atualizar o <i>Product Backlog</i>
Tempo (T)
6.1.1.4 Atualizar a Calendarização do Projeto
Custo (C)
6.1.1.5 Atualizar o Orçamento do Projeto
<i>Stakeholders (St)</i>
6.1.1.1 Preparar a <i>Grooming Meeting</i>
Comunicação (Co)
6.1.1.1 Preparar a <i>Grooming Meeting</i>

Tabela 6 - Áreas de Conhecimento Abordadas na *Grooming Meeting*

**Inputs:**

- *Product Backlog*

6.1.1.1. Preparar a *Grooming Meeting*

Como preparação da *Grooming Meeting*, o *Product Owner* informa os *stakeholders* sobre o estado atual do *Product Backlog* para que o mesmo possa ser discutido na reunião. O PO trata, também, dos detalhes técnicos como, por exemplo, reservar espaço e/ou equipamentos e prepara um guião para os temas a abordar na reunião (dúvidas existentes, detalhes pouco claros, novas propostas, etc.).

Assim sendo é necessário enquadrar este processo na área de *Stakeholders*, por poder envolver a criação de mecanismos de gestão dos participantes da reunião, e na área de Comunicação, por serem definidos processos de passagem de informação e identificadas necessidades de comunicação.

**Área de Conhecimento:** (St) *Stakeholders*; (Co) Comunicação.

### **Perguntas-chave:**

- Quem estará presente na reunião?
- Que informações é necessário discutir na reunião?
- Que recursos são necessários para levar a cabo a reunião?

### **Técnicas:**

- Plano de Reunião – é vantajoso que o PO elabore um plano e o transmita aos participantes, isto torna a reunião mais objetiva e eficiente. O plano define detalhes como hora de início e de término da reunião, local, objetivos, questões a discutir e resultados esperados.
- *Canvas & Boards* – a apresentação do estado do projeto e do produto a desenvolver através da construção de *Canvas* e *Boards* facilita a comunicação. Estes artefactos são diagramas e estruturas construídas para apresentar estratégias e informações referentes ao projeto. Por exemplo, o *Business Model Canvas* descreve, genericamente, a estratégia de negócio para criar, entregar e garantir valor num determinado produto. Estas ferramentas, quando utilizadas, devem ser atualizadas ao longo do desenvolvimento do projeto.

#### 6.1.1.2. Realizar a *Grooming Meeting*

A *Grooming Meeting* tem como objetivo manter o *Product Backlog* atualizado. O PO dirige a reunião com o auxílio do *Scrum Master* e do Gestor de Projeto e incentiva à participação dos *stakeholders* para que exponham as suas dúvidas e alertem para informações incorretas.

### **Área de Conhecimento: (Â) Âmbito.**

#### **Perguntas-chave:**

- Existe alguma informação em falta ou desatualizada?
- Existe alguma *User Story* que não seja compreensível?
- Alguma das *User Stories* deixou de ser relevante?
- A priorização está correta?
- Deve ser inserida alguma *User Story*?

## **Técnicas:**

O *Product Owner* recorre às suas *Soft Skills* para levar a bom termo a reunião. O PO pode, também, incitar a que mencionem o que correu bem e mal em cada reunião, para melhorar este processo.

- Negociação – a presença de *stakeholders* com perspetivas diferentes e conflituosas requer negociação entre ambas as partes para obter consenso.
- Comunicação – o PO introduz aspetos relevantes e promove a comunicação controlada entre os intervenientes para atingir os objetivos.
- Moderação – o grande número de participantes na reunião pode levar a que se gere alguma confusão, assim, cabe ao PO moderar a reunião mantendo o foco nos objetivos e sem dispersar nem despender mais tempo que o necessário nos vários temas.
- Estimulação e Inclusão – é comum que nem todos os participantes se sintam à vontade para participar na reunião potenciando a perda de informação valiosa. O PO deve estimular a participação de todos os *stakeholders* no debate.

### 6.1.1.3. Atualizar o *Product Backlog*

Depois de realizada a *Grooming Meeting* deve ser feita a análise do feedback obtido para identificar que alterações se têm de estabelecer ao *Product Backlog*. O *feedback* obtido pode implicar alterações ao plano de projeto. O PO e o gestor de projeto devem, em conjunto, decidir que modificações são necessárias e que modificações não devem ser incorporadas.

Este processo incide sobre o âmbito e pode, se as alterações assim o exigirem, implicar uma reformulação na gestão de tempo e custo. Deve por isto ser considerado aquando o desenvolvimento do plano de gestão de risco.

**Área de Conhecimento:** (Â) Âmbito.

## **Perguntas-chave:**

- Que *User Stories* têm de ser alteradas?
- Que *User Stories* têm de ser introduzidas?
- Que *User Stories* têm de ser eliminadas?
- Que *User Stories* têm de ser fracionadas?

- A prioridade necessita de ser alterada?

#### 6.1.1.4. Atualizar a Calendarização do Projeto

As alterações realizadas ao âmbito podem exigir uma atualização da calendarização do projeto. É comum existirem reservas a aplicar ao cronograma do projeto. Caso as reservas sejam insuficientes pode ser preciso alterar o cronograma estabelecido. Neste caso é necessária uma atualização ao cronograma do projeto sujeita, posteriormente, a avaliação.

**Área de Conhecimento:** (T) Tempo.

#### **Perguntas-chave:**

- As alterações introduzidas vão exigir mais tempo de desenvolvimento?
- As reservas aplicadas ao cronograma são suficientes face à atualização?

#### 6.1.1.5. Atualizar o Orçamento do Projeto

Tal como na secção anterior, as alterações realizadas sobre o âmbito do projeto podem exigir uma atualização do orçamento definido. E, mais uma vez, tal como no caso do cronograma do projeto, é usual que o plano inicialmente traçado, prevendo estas alterações, inclua reservas de contingência. Porém, pode ser preciso alocar novas verbas reformulando o orçamento inicial. Nestes casos, é, tal como para o cronograma, necessária uma atualização ao orçamento do projeto sujeita, posteriormente, a avaliação.

**Área de Conhecimento:** (C) Custo.

#### **Perguntas-chave:**

- As alterações introduzidas vão envolver novos custos?
- As reservas alocadas são suficientes face a estas novas alterações?

#### 6.1.2. *Sprint Planning Meeting*

Nestas reuniões devem estar presentes *Product Owner*, *Scrum Master*, equipa de desenvolvimento e gestor de projeto, sendo que o responsável pela gestão da reunião é o *Scrum Master*. A Tabela 7 possui os processos associados à *Sprint Planning Meeting* divididos por áreas de conhecimento.

Âmbito (Â)
6.1.2.2 Especificar as <i>User Stories</i> 6.1.2.3 Realizar a <i>Sprint Planning Meeting</i>
Qualidade (Q)
6.1.2.2 Especificar as <i>User Stories</i> 6.1.2.3 Realizar a <i>Sprint Planning Meeting</i>
Stakeholders (St)
6.1.2.1 Preparar da <i>Sprint Planning Meeting</i>

Tabela 7 - Áreas de Conhecimento Abordadas na *Sprint Planning Meeting*

**Inputs:**

- *Product Backlog*

6.1.2.1. Preparar a *Sprint Planning Meeting*

Tal como na preparação da *Grooming Meeting* (6.1.1.1), o *Product Owner* fica encarregue da passagem de informação relevante para a realização da reunião e trata, também, dos detalhes técnicos.

**Área de Conhecimento:** (St) *Stakeholders*; (Co) Comunicação.

**Perguntas-chave:**

- Quem estará presente na reunião?
- Que informações é necessário discutir na reunião?
- Que recursos são necessários para levar a cabo a reunião?

**Técnicas:**

- Plano de Reunião – é vantajoso que o PO elabore um plano e o transmita aos participantes, isto torna a reunião mais objetiva e eficiente. O plano define detalhes como hora de início e de término da reunião, local, objetivos, questões a discutir e resultados esperados.

- *Canvas & Boards* – a apresentação do estado do projeto e do produto a desenvolver através da construção de *Canvas* e *Boards* facilita a comunicação. Estes artefactos são utilizados e atualizados ao longo do desenvolvimento.

#### 6.1.2.2. Especificar as *User Stories*

O PO é responsável por especificar as *User Stories* a implementar na próxima iteração antes da *Sprint Planning Meeting*. É recomendado que sejam especificadas *User Stories* equivalentes a duas iterações para que exista alguma flexibilidade na escolha de funcionalidades a implementar.

A especificação das *User Stories* consiste em detalhar as tarefas de cada *User Story*, apresentando os critérios de aceitação, restrições e toda a informação relevante à implementação da funcionalidade.

A especificação das *User Stories* incide sobre o âmbito por detalhar as *User Stories* e qualidade por serem associados critérios de aceitação às mesmas.

**Área de Conhecimento:** (Â) Âmbito; (Q) Qualidade.

#### **Perguntas-chave:**

- Que tarefas define a *User Story*?
- Que restrições têm de ser tidas em conta?
- Que informação é necessária para conseguir implementar a *User Story*?

Tenha-se o exemplo da seguinte *User Story*: “Como vendedor quero introduzir produtos no catálogo para os clientes os conseguirem encomendar”.

A especificação inclui detalhes como:

- O vendedor possui um código de vendedor;
- O produto possui um nome, um código, um preço, uma categoria, uma descrição e uma pontuação (*rating*);
- O catálogo está dividido em categorias.

Esta informação é crucial à implementação correta da funcionalidade.

### 6.1.2.3. Realizar a *Sprint Planning Meeting*

Após terem sido especificadas as *User Stories* e preparada a reunião é, então, estabelecida a *Sprint Planning Meeting*.

Nesta reunião cabe ao *Product Owner* apresentar a especificação à equipa expondo também a respetiva prioridade. A equipa procura esclarecer todas as dúvidas que existam em relação às *User Stories* apresentadas e, de seguida, estima o esforço inerente a cada uma. Posteriormente, a equipa propõe-se a realizar um determinado valor de trabalho. O PO constrói, a partir das indicações da equipa, o *Sprint Backlog* que deve ser acordado por todos os elementos presentes.

Para além do *Sprint Backlog*, é ainda definido o *Sprint Goal* (descrição do que se pretende obter no final da iteração) e estabelecida a definição de pronto (*checklist* de atividades a realizar para considerar uma *User Story* pronta) que devem ser também acordados por todos os elementos presentes.

O *Scrum Master* é responsável por garantir que as normas da metodologia são respeitadas no decorrer da reunião.

A *Sprint Meeting* incide sobre o âmbito do projeto e, por definir o que se considera “pronto”, é também enquadrada na área de conhecimento da qualidade.

**Área de Conhecimento:** (Â) Âmbito; (Q) Qualidade.

#### **Perguntas-chave:**

- Que dúvidas existem em relação às especificações apresentadas?
- Qual é a proposta de trabalho da equipa?
- Que funcionalidades serão implementadas no próximo *sprint*?
- Quando se considera a funcionalidade pronta?
- Que critérios se vão utilizar para estabelecer o término do desenvolvimento de uma *User Story*?

#### **Técnicas:**

- *Planning Poker* – esta técnica é utilizada para estimar o esforço envolvido na implementação de uma *User Story*. Cada elemento possui um conjunto de cartas numeradas que utiliza para atribuir valor a uma dada *User Story*. Todos os elementos devem apresentar a carta simultaneamente para evitar influências nos

restantes. Depois de viradas as cartas todos devem chegar a um consenso sobre o valor de esforço a atribuir à *User Story*.

- *Definition of Done (DoD)* – A utilização da DoD permite saber quando considerar terminada a implementação da *User Story* através de uma listagem de tarefas. As tarefas devem ser claras e definir o que se precisa de cumprir para terminar a implementação de dada *User Story*, por exemplo, implementar a interface, implementar a lógica de negócio, aplicar testes unitários, documentar as conclusões, etc.

A utilização da DoD foca os elementos da equipa nos passos a tomar para desenvolver a *User Story* e auxilia na avaliação do desempenho e produtividade da equipa.

- *Sprint Goal* – O *Sprint Goal* define o que a equipa pretende conseguir até ao final da iteração, por exemplo, “Implementar a funcionalidade de pesquisa no catálogo”. Este objetivo é partilhado pela equipa e, tal como a DoD, ajuda a focar e concentrar a equipa em tarefas prioritárias e facilita a análise de desempenho no final do sprint.

## 6.2. Executar a *Sprint*

Nesta fase a equipa de desenvolvimento executa o que foi definido na *Sprint Planning Meeting*. De notar que a metodologia enaltece a autonomia e independência dos elementos da equipa e, por isto, cada elemento estipula o seu trabalho, não existindo uma regular atribuição de tarefas. Procura-se que a equipa seja pró-ativa e consciente do trabalho que é necessário realizar em cada iteração.

Ao longo da implementação o *Product Owner* e o *Scrum Master* orientam a equipa nas suas decisões, e auxiliam na resolução de problemas nas suas respetivas áreas de conhecimento. Na Tabela 8 temos o processo associado à execução da *Sprint* na respetiva área de conhecimento.

Recursos Humanos (RH)
6.2.1 Daily Scrum

Tabela 8 - Áreas de Conhecimento Abordadas na *Sprint Execution*

## Inputs:

- *Sprint Backlog*
- *Sprint Goal*
- *Definition of Done*

## Técnicas:

- *Sprint Goal e Definition of Done* – ao longo da iteração é considerado sempre o objetivo da *Sprint* e as definições de completude das tarefas a desenvolver.
- *Task Board* – A utilização de um quadro com as *User Stories* da iteração contribui para a definição mais compreensível e evidente do trabalho a realizar e do trabalho a decorrer. O quadro de trabalho é um mecanismo bastante simples e eficiente que promove a propagação do conhecimento a todas as partes interessadas. O próprio cliente deve conseguir inferir o estado do projeto através da consulta deste quadro. Na Figura 20 é observável um exemplo de um *Task Board* e o processo de uma tarefa exemplo desenvolvida. Assim sendo, a produção e organização da equipa é dinamizada [47]. Note-se que, tanto o *Board* como o processo apresentado em relação a uma tarefa, podem variar de acordo com as necessidades do projeto e/ou equipa de desenvolvimento.

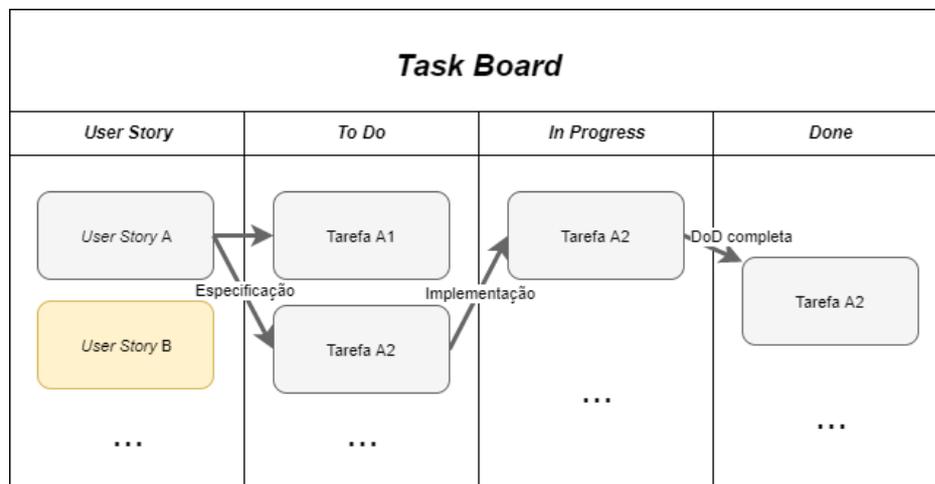


Figura 20 - Exemplo de uma *Task Board*

- *Impediments List* – lista de impedimentos que apresenta os obstáculos que cada elemento encontra ao desenvolver uma determinada tarefa. A análise da lista, tipicamente realizada pelo *Scrum Master* mas aberta a qualquer elemento da equipa, permite detetar problemas impeditivos e encontrar soluções mais

rapidamente. O anonimato no que toca à inserção de elementos impeditivos é opcional.

Estes impedimentos podem ou não ser tecnológicos, alguns exemplos são: falta de conhecimento sobre a linguagem utilizada; computador lento; falta de determinada informação; sem acesso à ferramenta; etc.

- *Spontaneous Grooming Meetings* – durante o desenvolvimento pode ser necessária a realização de *Grooming Meetings* para clarificar dúvidas existentes em relação ao trabalho a desenvolver. O *Scrum Master*, em conjunto com o *Product Owner*, decide se existe ou não necessidade de uma reunião de *Grooming*.
- *Sprint Burndown Chart* – este gráfico apresenta o valor estimado de trabalho face ao realizado até ao momento. Atualizado diariamente, é uma forma de rastrear o esforço da equipa ao longo da iteração.

As unidades utilizadas para o esforço correspondem às utilizadas pela equipa para a estimação das *User Stories*.

É, então, construída a linha ideal de esforço que corresponde à distribuição constante de esforço por parte da equipa ao longo do tempo (ver Figura 21).

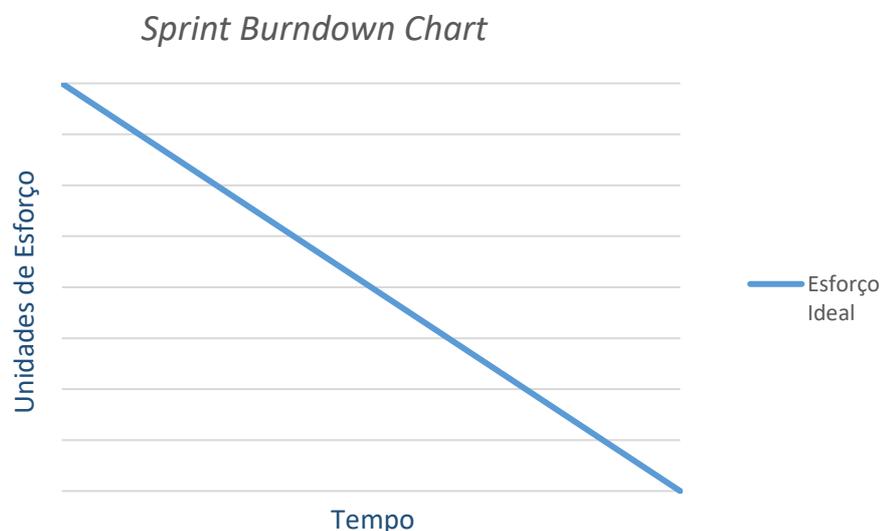


Figura 21 - *Sprint Burndown Chart*, Esforço Ideal

Cada elemento, no final de cada dia de trabalho, atualiza o esforço que acredita faltar para a realização das suas tarefas. Assim, com a informação sobre o “esforço restante” é construída a segunda série do gráfico (Figura 22).

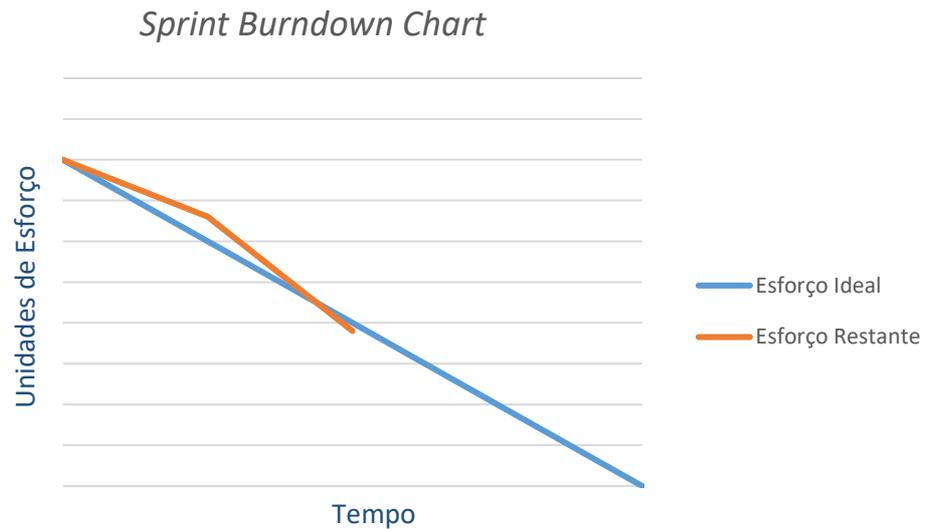


Figura 22 - *Sprint Burndown Chart*

### 6.2.1. *Daily Scrum*

A existência de reuniões diárias com a equipa para promoção de conhecimento e comunicação é crucial para o sucesso do projeto [48]. As reuniões devem ser curtas (15 minutos no máximo) e focadas no essencial: sincronização da equipa e propagação de conhecimento.

Estas reuniões procuram que toda a equipa se torne ciente do trabalho a ser realizado, do progresso do projeto e dos obstáculos existentes.

Este processo deve estar presente no plano de comunicações desenvolvido (5.1.19) e pertence à área de conhecimento dos recursos humanos, onde são definidas responsabilidades, necessidades de formação, etc.

**Área de Conhecimento:** (RH) Recursos Humanos.

#### **Perguntas-chave:**

- O que fiz ontem?
- O que vou fazer no dia de hoje?
- O que me está a impedir de conseguir realizar dada tarefa?

#### **Técnicas:**

- *Task Board* – a *Daily Scrum* tem como objetivo propagar a informação do estado do projeto. Assim, a utilização do *Task Board* é vantajosa. Cada elemento da

equipa, através da análise do *Board*, é capaz de entender o trabalho produzido e o trabalho que se encontra em produção.

- *Impediments List* – com a identificação dos problemas encontrados pode já existir uma solução conhecida, que acelera o processo de resolução, ou, posteriormente, com a descoberta da solução esse conhecimento é transferido à equipa para futura utilização. Ao utilizar a *impediments list* como suporte para discussão sobre os impedimentos ao progresso do projeto, a procura de soluções é dinamizada.
- *Sprint Burndown Chart* – este gráfico situa o desenvolvimento do projeto no que toca ao esforço envolvido. É possível, mediante a análise do *Sprint Burndown Chart*, perceber se a distribuição de esforço está a ser realizada da melhor forma.

### 6.3. Concluir a *Sprint*

No final da iteração é realizada uma entrega fracionada ao cliente para que seja obtido feedback sobre o trabalho realizado. Em função do *feedback* obtido, podem ser identificadas alterações a incorporar. As alterações podem, ou não, significar a alteração do plano de projeto inicialmente definido.

O gestor de projeto decreta quando as alterações não devem ser permitidas. Por exemplo, o *feedback* obtido pode exigir a inclusão de determinadas funcionalidades fora do âmbito definido para o projeto. Neste caso, e justificadamente, as alterações devem ser rejeitadas. O gestor analisa as alterações face ao acordo inicial com o cliente e, utilizando o seu bom senso, pode ou não permitir a introdução das alterações. É preciso ter como prioridade o objetivo principal do projeto e nem sempre as alterações justificam o gasto de recursos.

Note-se que as alterações podem ser remoções ou adições de funcionalidades para além de alterações a funcionalidades já existentes.

Na Tabela 9 são apresentados os processos da conclusão da *Sprint* segundo as respetivas áreas de conhecimento.

Âmbito (Â)
6.3.1 <i>Sprint Review</i>
Risco (R)
6.3.2 <i>Sprint Retrospective</i>
Sucesso (Su)
6.3.2 <i>Sprint Retrospective</i>

Tabela 9 - Áreas de Conhecimento Abordadas no *Sprint Closing*

### Inputs:

- Especificação das *User Stories*
- *Sprint Goal*
- *Definition of Done*
- *Impediments List*

### Técnicas:

- *Product Burndown Chart* – à semelhança do *Sprint Burndown Chart*, no final da *sprint*, pode ser construído o *Product Burndown Chart*. Este gráfico mede o trabalho restante ao longo das *sprints*, ou seja, no eixo vertical tem-se o trabalho existente no início de cada *sprint* e, no eixo horizontal, os *sprints* do projeto.

#### 6.3.1. *Sprint Review*

A *Sprint Review* é realizada para expor aos participantes o trabalho desenvolvido ao longo da *sprint*, esta reunião é aberta a todos os *stakeholders*.

Cabe ao *Product Owner* avaliar a fração de produto construído durante a iteração. O PO estabelece que tarefas considera prontas e que alterações devem ser introduzidas (caso existam). Tarefas não aceites são desenvolvidas novamente numa outra iteração.

Ao longo da *Sprint Review* todos os participantes podem expressar as suas opiniões em relação ao trabalho apresentado. Esta reunião é conduzida pelo PO e/ou por elementos da equipa de desenvolvimento.

**Área de Conhecimento:** (Â) Âmbito.

**Perguntas-chave:**

- O que foi desenvolvido nesta iteração?
- O trabalho apresentado vai de encontro ao pretendido?

**Técnicas:**

As demonstrações devem ser curtas e focadas na tarefa que se pretende apresentar. Podem ser criados cenários para ilustrar o procedimento exposto ou, em vez disto, enumerar e detalhar as tarefas que se pretendem realizar durante a demonstração.

- *Reviews Report* – estes relatórios devem ser construídos pelo *Product Owner* em função da avaliação à fração de produto apresentada. A documentação é disponibilizada aos *stakeholders* para corrigir possíveis erros ou complementar informação e utilizada pela equipa para reformular implementações incorretas.

### 6.3.2. *Sprint Retrospective*

Esta reunião é apenas para a equipa de desenvolvimento e o *Scrum Master*. O objetivo desta reunião é melhorar o processo atual de desenvolvimento [49]. O *Scrum Master* procura detetar falhas e pontos fracos no processo atual e desenvolver soluções para os mesmos.

Note-se que, a existência da *Sprint Retrospective*, não significa que a iteração anterior tenha corrido mal ou que o processo tenha falhas, esta reunião baseia-se no princípio “*Há sempre forma de melhorar*”.

Esta reunião incide na área do sucesso porque procura identificar novos fatores de sucesso e eliminar bloqueadores dos fatores conhecidos. É também associada à área dos riscos por procurar identificar oportunidades e ameaças ao desenvolvimento do projeto.

**Área de Conhecimento:** (R) Risco; (Su) Sucesso.

**Perguntas-chave:**

- O que correu bem?
- Que acontecimentos foram positivos/negativos?
- Que obstáculos apareceram na iteração?

- Que fatores podemos aproveitar para aumentar o sucesso do projeto?
- Onde podemos melhorar?

### **Técnicas:**

Existem vários métodos de execução de retrospectivas, cabe ao *Scrum Master* decidir o mais adequado à equipa. Os métodos podem ainda ser alterados de reunião para reunião se necessário.

- *The Sailboat* – este método recorre a metáforas para conseguir identificar possíveis melhorias do processo. O *Scrum Master* desenha no quadro um barco e explica à equipa as metáforas:
  - Motores/Velas/Tripulantes simbolizam elementos do processo que promovem o sucesso da equipa;
  - Âncoras/Icebergs/Tempestades simbolizam obstáculos ao sucesso da equipa.

Posteriormente, cada elemento coloca *post-its* com as suas sugestões no barco (de acordo com as metáforas apresentadas). Com todos os *post-its* colocados, a equipa discute possíveis soluções e/ou fatores a aproveitar.

- *Mad Sad and Glad* – o *Scrum Master* divide o quadro em 3 áreas:
  - *Mad* – frustrações, fatores que irritaram a equipa ou que a fizeram perder tempo;
  - *Sad* – desilusões, elementos que não correram como se esperava;
  - *Glad* – fatores que influenciaram positivamente a equipa.

À semelhança do método anterior, cada elemento deve colocar *post-its* com as suas sugestões nas secções adequadas do quadro e, com todos os *post-its* colocados, são discutidas possíveis soluções ou fatores a ser aproveitados.

---

## FASE DE MONITORIZAÇÃO E CONTROLO

---

Esta fase decorre em simultâneo com a fase de execução. O gestor de projeto é encarregue de monitorizar e controlar o progresso do projeto. Porém, deve evitar interferir na sequenciação normal da metodologia de desenvolvimento que promove a autonomia e independência dos elementos da equipa de desenvolvimento. Assim, a existência de um controlo demasiado presente e “invasor” poderá afetar negativamente a performance da equipa.

Se o gestor de projeto, através do controlo e monitorização, se aperceber de questões que precisam de ser resolvidas, deverá marcar uma reunião para esse efeito. Se o problema incidir sobre aspetos tecnológicos ou inerentes à metodologia de desenvolvimento, a reunião deverá ser marcada com o *Scrum Master*. Se o problema for em relação ao negócio ou âmbito do projeto, a reunião deverá ser com o *Product Owner*. Nesta reunião deve procurar-se uma solução adequada ao problema.

A monitorização permite identificar a necessidade de desvios do plano traçado e reagir de acordo, promovendo o sucesso do projeto. Através da natureza iterativa e incremental da metodologia de desenvolvimento, aliada à promoção de contacto regular com as várias partes interessadas, a monitorização é facilitada.

### **Inputs:**

- Plano de Projeto – globalmente utilizado em toda a monitorização aplicada ao projeto.

## 7.1. Controlar o Âmbito

Segundo o PMBoK, monitorizar e gerir o âmbito permite manter a base do âmbito do projeto intacta. Este tipo de monitorização pode ser erradamente entendido como recusar qualquer pedido fora do âmbito inicialmente traçado. Ao utilizar o *Scrum*, a incorporação de alterações exteriores ao âmbito definido que não representem *Scope Creep*, ou seja, sem impacto negativo (em termos de custo, tempo, etc.) é permitida. Alterações que não representem maior esforço por parte da equipa ou maior tempo de desenvolvimento podem ser aceites (mas é preciso estar alerta para a existência *Scope Creep*).

Se ainda não tiver sido realizado qualquer trabalho em relação a esse âmbito, se não houver maior esforço de trabalho, maior custo ou mais tempo de desenvolvimento envolvido não existe *Scope Creep*.

Existe *Scope Creep* se for pretendida a inserção de novas tarefas substituindo tarefas já concluídas (não incluindo correções ao trabalho realizado!). Existe *Scope Creep* se o pedido representar um maior esforço de trabalho.

Para alterações de grande escala é necessário gerar um pedido oficial de alteração do projeto que deverá ser avaliado pelas entidades supervisoras do projeto e, caso sejam incorporadas as alterações, o plano deve ser redefinido.

### **Inputs:**

- *Product e Sprint Backlog*
- *Task Board* – a análise do *task board* ao longo do projeto permite perceber a prioridade dada aos diferentes requisitos e se o trabalho realizado está de acordo com o planeado.

### **Técnicas:**

- Análise de Variância – a análise de variância compara o projeto atual com o plano para o projeto, permitindo determinar os desvios do âmbito e classificar o grau desses desvios.
- Realização de *Sprint Reviews* – estas reuniões auxiliam o gestor de projeto a verificar se o trabalho realizado vai de encontro ao esperado pelo(s) cliente(s) ou se são necessárias alterações ao âmbito.

## 7.2. Controlar o Tempo

Monitorizar o estado do projeto estipula se os prazos estão a ser cumpridos e se é, ou não, necessário adotar ações preventivas e/ou reativas referentes à calendarização.

A utilização de uma metodologia ágil como o *Scrum*, que permite produzir *Burndown Charts* e visualizar facilmente o trabalho implementado até ao momento, facilita a monitorização do cronograma. Para além disto, a priorização constante dos requisitos agiliza o desenvolvimento e identifica com maior facilidade possíveis atrasos.

### **Inputs:**

- *Burndown Charts*
- *Impediments List*

### **Técnicas:**

- Realização de *Sprint Retrospectives* – nestas reuniões é feita uma análise ao trabalho realizado e avaliado se o mesmo foi ou não o definido na *Sprint Meeting*. No caso de a iteração não ter decorrido como o planeado são discutidas as razões para tal ter acontecido e possíveis soluções que se possam utilizar no futuro.
- Análise dos *Burndown Charts* – sendo que o *sprint burndown chart* identifica o valor planeado de trabalho face ao valor de trabalho efetivamente realizado e o *product burndown chart* mede o trabalho que falta realizar, ambos os gráficos permitem ao gestor de projeto identificar se o projeto está a decorrer ou não como planeado.
- Ajuste de *Leads e Lags* – aplicar uma *lead* corresponde a iniciar uma atividade B com dependência *Finish to Start* de outra atividade A mais cedo. Por exemplo, apesar de B ter de se iniciar posteriormente ao término A, existe uma folga que permite que B seja iniciada 5 dias antes de A terminar, neste caso tem-se uma *lead* de 5 dias. *Lag* corresponde ao intervalo de tempo que se tem de esperar para iniciar uma atividade B com dependência *Finish to Start* de outra atividade A. O gestor de projeto pode procurar ajustar valores de *Leads* e *Lags* para benefício do projeto.

### 7.3. Monitorizar o Orçamento

O gestor de projeto deve manter-se atento às variações do orçamento em relação ao previsto no plano de projeto e reagir adequadamente por forma a minimizar os riscos relacionados com os custos do projeto. Caso o projeto sofra alterações significativas no que toca a custos do orçamento, o valor estimado no plano de projeto deve ser reformulado para representar as variações no custo.

#### Técnicas:

- *Earned Value Management* (EVM) – esta técnica permite monitorizar o progresso do projeto de forma concreta e objetiva. A EVM analisa a performance no que toca ao âmbito, cronograma e orçamento. Por esta razão, será explicada antes da estruturação da fase de monitorização e controlo do orçamento.

A EVM envolve o cálculo dos seguintes valores:

- *Planned Value* (PV) – orçamento previsto para uma dada atividade num determinado intervalo de tempo. Por exemplo, se é previsto um custo de 10 euros para a atividade A e uma duração de 2 dias, então, em cada um dos dias o custo é de 5 euros.
- *Actual Cost* (AC) – custos totais, diretos ou indiretos, necessários para completar uma dada atividade num dado intervalo de tempo.
- *Earned Value* (EV) – corresponde à percentagem de trabalho completado multiplicado pelo *Planned Value* (PV).
- *Cost Variance* (CV) – valor obtido subtraindo o valor de *Actual Cost* (AC) ao *Earned Value* (EV),  $CV = EV - AC$ . Este valor representa a diferença entre o custo estimado de uma atividade e o custo real. Assim sendo um CV negativo significa que a atividade superou os custos esperados e um número positivo significa que a atividade foi mais económica do que se planeava.
- *Schedule Variance* (SV) – representa a diferença entre a previsão de completude de uma atividade e a calendarização real. Este valor é obtido subtraindo ao *Earned Value* (EV) o *Planned Value* (PV),  $SV = EV - PV$ .
- *Cost Performance Index* (CPI) – indicador para determinar se o projeto se encontra dentro do orçamento. O CPI é calculado pela divisão entre o *Earned Value* (EV) e o *Actual Cost* (AC),  $CPI = EV/AC$ . Se o CPI for igual

- a 1, o orçamento está como o definido no plano de projeto, se o valor do CPI for menor que 1, o orçamento foi ultrapassado e, se o valor for superior a 1, o orçamento real do projeto está mais baixo que o planejado.
- *Schedule Performance Index (SPI)* – indicador para determinar se o progresso do projeto decorre como estimado no plano de projeto. O SPI é calculado pela divisão do *Earned Value* com o *Planned Value*,  $SPI = EV/PV$ . Um SPI de 1 significa que a estimativa está de acordo com o progresso real do projeto, valores inferiores a 1 significam que o projeto se encontra atrasado em relação à previsão e, valores superiores a 1 significam que o projeto se encontra adiantado em relação ao plano.
  - *Forecasting* – com os valores obtidos na EVM o gestor de projeto consegue calcular previsões sobre o *Estimate at Completion (EAC)*, valor estimado do custo do projeto com base nos valores atuais. Esta previsão pode ser feita assumindo:
    - Os valores da EVM como os valores atuais do custo do projeto e que os valores de progresso do projeto vão corresponder à estimativa efetuada no plano de projeto. Assim calcula-se o EAC através da fórmula:  $EAC = AC + (BAC - EV)$ , sendo que AC e EV são os valores obtidos na EVM e BAC representa o orçamento atribuído no plano de projeto.
    - Os valores de progresso do projeto obtidos na EVM vão ser mantidos, ou seja, o histórico atual representa o progresso previsto. Dentro disto podemos ter em conta duas fórmulas:  $EAC = BAC / CPI$  e  $EAC = AC + [(BAC - EV) / (CPI \times SPI)]$ , sendo que a última tem em conta o SPI e o CPI calculados na EVM e a primeira apenas considera o CPI.
  - *To-Complete Performance Index (TCPI)* – se o projeto não estiver a cumprir as métricas de progresso desejadas (atrasado ou acima do orçamento), o gestor de projeto pode recorrer ao TCPI para determinar a performance necessária para completar o progresso dentro dos valores estimados. O TCPI determina a performance necessária para que o projeto cumpra determinadas métricas. O TCPI é calculado através da seguinte fórmula:  $(BAC - EV) / (BAC - AC)$ .
  - *Performance Review* – esta técnica baseia-se na comparação do custo do projeto ao longo do cronograma, atividades ou *work-packages*. Se a EVM for utilizada então o gestor de projeto consegue aplicar:
    - Análise de Variância – (Variância = Valor atual – valor estimado), compara a performance esperada com a atual e identifica as causas da variância, caso exista, entre as duas.

- Análise de Tendência – através dos valores da EVM, é determinado se os valores do custo do projeto estão a melhorar ou a piorar.
- *Earned Value Performance* – compara os valores obtidos na EVM com os valores estimados no plano de projeto.

#### 7.4. Monitorizar a Equipa do Projeto

A gestão da equipa de projeto, segundo o PMBoK, é o processo de rastrear a performance de cada elemento da equipa fornecendo feedback, resolvendo problemas e otimizando a performance. Com a integração do *Scrum* como metodologia de desenvolvimento, esta responsabilidade recai sobre o *Scrum Master*. Assim, o gestor de projeto deve procurar aplicar uma monitorização não invasiva para análise de desempenho da equipa e, caso seja necessário, reunir com o *Scrum Master* para a resolução de eventuais problemas identificados.

É da responsabilidade do *Scrum Master* (no âmbito da gestão da equipa) reportar quaisquer questões/decisões de grande importância ao gestor de projeto, aplicar mecanismos de reconhecimento e apreciação e identificar conflitos e problemas sobre a equipa do projeto de forma a promover a *performance* da equipa [50].

##### **Inputs:**

- *Task Board* – com a análise da *Task Board* são determinados os responsáveis pelas tarefas e respetivo estado de desenvolvimento.
- *Impediments List* – pode conter informação sobre o que a equipa pensa ter impacto negativo no seu trabalho.

##### **Técnicas:**

- Reuniões de Equipa Regulares – o contacto regular com os elementos da equipa permite, por observação, comunicação e análise de comportamentos, identificar eventuais problemas referentes à gestão da equipa.
- Eventos de *Team Building* – este tipo de eventos permite a identificação e/ou resolução de conflitos existentes na equipa e promove a criação de relações sociais entre elementos da equipa.
- Diferenciação dos *Burndown Charts* – é possível especificar os *burndown charts* para um determinado elemento da equipa. Isto permite avaliar o seu desempenho

no projeto e auxilia na identificação de problemas sobre o elemento em questão (através de uma análise de desempenho ao longo do projeto).

- Gestão de Conflitos – entre outros métodos de resolução de conflitos, podem ser criadas normas adicionais de funcionamento da equipa, alteradas e/ou transferidas responsabilidades e promovida comunicação entre os elementos conflituosos.
- Capacidades interpessoais – para gerir de forma eficaz a equipa do projeto, o responsável deve possuir capacidades interpessoais de liderança, influência, comunicação, compreensão, etc.
- Reconhecimentos – um sistema de reconhecimentos motiva a equipa promovendo o seu desempenho ao longo do projeto.

## 7.5. Monitorizar e Controlar os Riscos

O gestor de projeto deve manter-se atento à alteração, introdução e eliminação de riscos no projeto, atualizando o plano do projeto para responder efetivamente a cada um dos riscos.

A gestão do risco determina a capacidade do projeto ser bem-sucedido face a ameaças e/ou oportunidades que surjam ao longo do seu desenvolvimento.

### **Inputs:**

- *Impediments List* – através da análise dos impedimentos do projeto podem ser descobertos novos riscos como, por exemplo, a possibilidade de um conflito entre colegas ou a inadequação das ferramentas disponíveis a um novo item a desenvolver.
- *Burndown Charts* – o progresso do projeto permite perceber quais os riscos que podem ocorrer neste estado e formar previsões sobre a evolução do projeto e seus riscos.

### **Técnicas:**

- Reavaliação dos Riscos – o gestor de projeto reavalia periodicamente a listagem de riscos e respetivas respostas de forma a identificar novos riscos, eliminar riscos ultrapassados ou agora não relevantes, detalhar riscos, etc.

- Reavaliação dos Planos de Resposta – para além de ser possível que os riscos sofram alterações, os seus planos de resposta podem também necessitar de ser alterados. É por isto importante que se reavaliem os planos de resposta de cada risco para determinar planos de resposta desatualizados ou agora inadequados, novos planos de resposta, etc.
- Análise de Variância e Tendência – a análise de variância e tendência já referidas na monitorização do orçamento e âmbito permitem identificar necessidades de alterações ao plano de resposta aos riscos.
- Análise das Reservas – a análise de reservas compara as reservas restantes com o risco do projeto, determinando se as reservas disponíveis são suficientes.
- *Scrum Meetings* – estas reuniões tornam possível a recolha das opiniões das partes interessadas em relação a potenciais obstáculos e/ou riscos do projeto. Mesmo que não exista uma discussão direta sobre os riscos do projeto, a partir das opiniões trocadas nestas reuniões, podem ser descobertos novos riscos, alterações aos riscos correntes, novas técnicas de resposta aos riscos, etc.

## 7.6. Assegurar a Qualidade

A monitorização da qualidade possibilita a obtenção de um resultado de acordo com critérios dos clientes. Este processo analisa a adequação do trabalho e produto obtido face às expectativas do cliente, tendo por base a definição de qualidade apresentada no plano de projeto.

A metodologia de desenvolvimento ágil utilizada visa garantir que o produto final respeita as necessidades do cliente. O que se relaciona diretamente à construção de um produto final de qualidade, assim sendo, a utilização de uma metodologia ágil auxilia à obtenção de qualidade no produto final.

### **Inputs:**

- *Definition of Done* – as definições de pronto desenvolvidas nas iterações de desenvolvimento têm em conta a qualidade esperada do trabalho final.
- Métricas e Medidas de Qualidade – métricas definidas e acordadas sob o projeto e seu resultado.

### **Técnicas:**

- Testes – A utilização de testes para verificar a definição de pronto estabelecida em cada iteração garante, ao longo do desenvolvimento, que o trabalho produzido respeita os padrões de qualidade definidos.
- Realização de *Sprint Reviews* – com a presença do(s) cliente(s) é possível verificar se o resultado fracionado corresponde às suas expectativas. Estas reuniões, para além de auxiliar na monitorização do âmbito, facilitam o controlo de qualidade do produto.

## 7.7. Controlar os Fatores de Sucesso

Controlar os fatores de sucesso envolve perceber se os fatores estão a ser aproveitados, e se as técnicas identificadas são eficazes para conseguir impulsionar os fatores de sucesso.

O gestor de projeto, utilizando as métricas definidas no plano de gestão de sucesso, avalia a eficiência das técnicas de aproveitamento dos fatores e identifica se o projeto usufrui ou não dos fatores de sucesso estabelecidos.

### **Inputs:**

- Métricas dos Fatores de Sucesso – métricas definidas no plano de gestão do sucesso.

Técnicas:

- Realização de *Sprint Retrospective* – nestas reuniões procura-se identificar bloqueios aos fatores de sucesso e caracterizar novos fatores de sucesso. A utilização de uma metodologia ágil promove a frequência das monitorizações e, conseqüentemente, afeta o sucesso do projeto positivamente [51].

## 7.8. Gerir as Expectativas dos *Stakeholders*

O gestor de projeto, com o apoio do *Product Owner*, é responsável por encontrar e utilizar as técnicas de gestão de *stakeholders* mais apropriadas a cada parte interessada. Estas técnicas podem sofrer alterações ao longo do desenvolvimento do projeto.

É da responsabilidade do *Product Owner* aplicar o plano de gestão de expectativas traçado sendo que, se perceber que são necessárias alterações, deve passar essa informação ao gestor para que o plano seja atualizado de acordo.

### **Técnicas:**

- Integração dos *Stakeholders* – a integração dos *stakeholders* como parte ativa do projeto é um mecanismo eficiente para manter os *stakeholders* informados sobre o projeto. Esta técnica promove a segurança e confiança dos *stakeholders* no projeto gerindo ativamente as suas expectativas.
- *Sprint Reviews* – aumentam a proximidade dos *stakeholders* com o projeto, incitando à participação e recolhendo informação valiosa para o projeto.
- Sistemas de Gestão da Informação – estes sistemas agregam informação relevante sobre o projeto de forma a conseguir distribuir o conhecimento aos *stakeholders*. Um exemplo será a utilização da *Task Board* para transmitir o progresso do desenvolvimento do âmbito.

## 7.9. Controlar as Comunicações

A monitorização das comunicações envolve a distribuição e disponibilização de informação do projeto a determinadas partes interessadas consoante as necessidades do projeto e de cada um dos *stakeholders*.

A utilização de uma metodologia de desenvolvimento, que integra reuniões e processos de interação bem definidos e estruturados, auxilia a gerir as comunicações.

### **Inputs:**

- *Task Board* – a *Task Board* transmite de forma simples o estado corrente do projeto, isto facilita a comunicação entre partes interessadas sobre detalhes do projeto.
- *Impediments List* – auxilia na discussão sobre os obstáculos ao projeto.
- *Burndown Charts* – demonstram a performance da equipa e o progresso do projeto facilitando a comunicação relativa à evolução do projeto.
- Modelos de Comunicação – devem seguir-se os modelos de comunicação, caso tenham sido desenvolvidos na fase de planeamento.

### **Técnicas:**

- Ferramentas Tecnológicas – a utilização de ferramentas que facilitem comunicação entre *stakeholders* é importante para que a informação seja propagada para benefício do projeto.
- Reuniões de Equipa Regulares e Entregas Fracionadas (*Scrum*) – o *Scrum* define reuniões estruturadas e com objetivos claros que promovem a comunicação entre as partes interessadas.
- *Product Owner* – o *Product Owner* “faz a ponte” entre os clientes e a equipa para que as partes interessadas com conhecimentos limitados ao domínio do problema/solução consigam comunicar eficazmente com os integrantes da equipa de projeto.
- *Scrum Master* – o *Scrum Master* é o porta-voz da equipa de desenvolvimento e assegura-se que a equipa não é sobrecarregada para que os elementos se foquem no trabalho a desenvolver.
- Acesso Livre – disponibilizar livremente toda a informação aos *stakeholders*, não só simplifica a comunicação (por já existir propagação de conhecimento) como aumenta a confiança das partes interessadas no projeto. O *Scrum* defende a transparência ao longo do projeto.

### 7.10. Levar a Cabo as Aquisições

O gestor de projeto leva a cabo as aquisições necessárias ao desenvolvimento do projeto, previamente estudadas na elaboração do plano de projeto. O gestor de projeto informa-se sobre as ofertas existentes no mercado no que trata à obtenção dos recursos e, com isto, escolhe a opção mais vantajosa.

#### **Inputs:**

- Propostas de fornecedores;
- Documentação da Análise de Mercado.

#### **Técnicas:**

- *Bidder Conferences* – reuniões com todos os possíveis fornecedores para esclarecer os detalhes da aquisição que se pretende realizar.

- Avaliação de Propostas – consiste na exclusão de potenciais fornecedores numa primeira fase avaliando a presença de elementos cruciais associados à aquisição (por exemplo: tecnologia, capacidade, etc.) e, posteriormente, por avaliação de critérios vantajosos (por exemplo: menor custo, maior rapidez, etc.). Esta avaliação é, normalmente, realizada por um júri e documentada formalmente.
- Estimativas Independentes – através da análise de mercado, podem ser associadas estimativas de propostas às aquisições. Estas estimativas devem ser utilizadas como referências face às propostas dos fornecedores.
- Técnicas Analíticas – este tipo de técnica determina a capacidade ou prontidão do fornecedor para levar a cabo a aquisição da forma mais satisfatória possível.
- Negociação – apesar de já se terem planeado os detalhes dos contratos referentes às aquisições, os fornecedores podem não aceitar os termos estabelecidos. Assim, é necessário um processo de negociação para obter um resultado favorável e acordado pelas duas partes.
- Propaganda – publicitar a necessidade de um fornecedor pode recolher outros fornecedores com propostas mais benéficas que as disponíveis anteriormente.

### 7.11. Monitorizar as Aquisições

Para garantir que os contratos das aquisições são cumpridos por ambas as partes, o gestor de projeto deve monitorizar o desenvolvimento dos contratos das aquisições.

#### **Inputs:**

- Documentação dos Contratos estabelecidos.

#### **Técnicas:**

- Inspeções e Auditorias – pedidas pelo comprador e suportadas pelo fornecedor, servem para verificar se o fornecedor está a cumprir o que foi estabelecido no contrato.
- Relatórios de Desempenho – o fornecedor pode disponibilizar documentação referente ao serviço prestado para que o comprador consiga avaliar o seu desempenho.

- Sistema de Controlo de Alterações ao Contrato – este tipo de sistema define um conjunto processos que permite analisar, aprovar e recusar alterações propostas aos contratos.
- Sistema de Gestão de Registos – utilizado para armazenar toda a documentação produzida referente aos contratos e sua gestão, este sistema contém informação relevante para monitorizar e controlar as aquisições.

---

## FASE DE CONCLUSÃO

---

Esta é a última fase do projeto onde se formaliza o seu término. É nesta fase que o cliente aceita formalmente o resultado do projeto, são efetuadas revisões e avaliações ao projeto e são fechados todos os contratos do projeto.

### 8.1. Fecho de Contratos e Aquisições

São terminados oficialmente cada um dos contratos estabelecidos, avaliando os resultados obtidos em cada um e registrando estas avaliações para utilização em projetos futuros.

### 8.2. Consolidação das Lições Aprendidas

O gestor de projeto regista e armazena as lições aprendidas que se retiram do projeto para futura utilização [52]. Para isto, questiona os *stakeholders* sobre as suas opiniões no que toca ao desenvolvimento do projeto e reflete sobre a informação obtida e sobre a sua própria experiência ao longo do projeto, formulando documentação importante a reter para projetos futuros.

#### 8.2.1. *Closing Meeting*

Na *closing meeting* (reunião de encerramento) é feita uma retrospectiva do projeto onde toda a equipa deve estar presente, procurando responder a questões como:

- O que correu bem/mal neste projeto?
- Que impedimentos foram onerosos para o projeto?
- Que oportunidades poderiam ser/foram aproveitadas?
- O que se deveria tentar reproduzir em projetos futuros?

- O que se deveria evitar nos projetos futuros?

### 8.2.2. Recolha de Opiniões dos *Stakeholders*

O gestor de projeto determina as opiniões dos restantes *stakeholders* em relação ao projeto de forma a que consiga perceber o que correu bem e o que correu mal, do ponto de vista exterior à equipa, para o ajudar na redação da documentação e consolidação das lições aprendidas.

---

## CONCLUSÃO

---

“Many of the theories and concepts of project management are not difficult to understand. What is difficult is implementing them in various environments”, Kathy Schwalbe [6]. Esta frase descreve com precisão a realidade: cada projeto é único e a complexidade surge da necessidade de entender quais as melhores soluções de gestão para um determinado ambiente de um projeto.

A primeira fase deste trabalho consistiu no estudo do estado da arte, para identificar e caracterizar as principais metodologias de gestão de projetos e de desenvolvimento ágil. Construiu-se, assim, uma base conceptual que foi, posteriormente, utilizada para desenvolver um novo método de gestão de projetos, articulando as metodologias estruturadas da gestão de projetos, com metodologias ágeis.

De seguida, foram identificadas e caracterizadas em detalhe as fases e os processos do método e, para cada um desses processos, detalhadas as técnicas passíveis de utilização e as respetivas vantagens.

Obteve-se assim uma descrição completa e detalhada do método proposto, que concilia um referencial de gestão de projetos clássico (PMBok) com uma metodologia de desenvolvimento ágil (*Scrum*). Desta forma, considera-se que o principal contributo desta dissertação é a disponibilização deste método de gestão de projetos à eSPap ou a outras entidades para aplicação nos seus projetos.

Dado o tempo disponibilizado para o desenvolvimento desta dissertação, a avaliação do método foi feita apenas com base no estudo sobre os processos e técnicas propostas e não houve possibilidade de aplicação do método a um projeto real. Isto representa uma limitação sobre os resultados obtidos e também uma oportunidade de trabalho futuro, a aplicação do método a projetos reais de diferentes dimensões e complexidades permitirá

identificar oportunidades de melhoria. De notar, no entanto, ter havido um projeto preliminar em que, de forma exploratória, se conseguiram experimentar muitas das ideias definidas e efetivamente comprovar a sua eficácia. Para além disto, considera-se interessante o desenvolvimento futuro de uma ferramenta informática para suporte deste método que facilite a sua aplicação e melhore a performance do projeto.



---

## MATERIAL DE SUPORTE

---

### I. Glossário

***Burndown Chart:*** gráfico construído para monitorizar o projeto.

***Definition of Done:*** *checklist* que se tem de verificar para considerar uma *user story* completa.

***Deliverables:*** produto, resultado ou serviço resultante no final de um processo, fase ou projeto.

***Earned Value Management:*** técnica de monitorização do projeto, podendo aplicar-se ao âmbito, cronograma ou orçamento.

**Gestão de Projeto:** utilização de conhecimentos, técnicas, metodologias e ferramentas nas fases de um projeto em prol do seu sucesso.

**Gestor de Projeto:** responsável pela gestão de um ou mais projetos.

***Grooming Meeting:*** reunião aberta aos vários *stakeholders* onde é discutida a validade do *product backlog*.

***Impediments List:*** lista que detalha os obstáculos ao desenvolvimento do projeto.

***Milestone:*** marco no cronograma relevante ao projeto.

***Persona:*** sujeito fictício que representa um determinado tipo de utilizador do sistema.

***Planning Poker:*** técnica de estimação de esforço e dificuldade de uma dada *user story*.

**Portfólio:** conjunto de programas e de projetos que, em conjunto, procuram atingir um ou mais objetivos de negócio.

**Product Backlog:** conjunto de *user stories* a desenvolver para obter o resultado final do projeto.

**Product Burndown Chart:** gráfico que permite medir o progresso do projeto, analisando a percentagem de completude do resultado final.

**Product Owner:** elemento da equipa com grande conhecimento sobre a área de negócio.

**Programa:** conjunto de projetos que têm um ou mais objetivos de negócio em comum.

**Project Initiator:** elemento que inicia o projeto.

**Project Sponsor:** responsável pela alocação de recursos ao projeto.

**Projeto:** esforço temporário empreendido para criar um produto, serviço ou resultado único.

**Protótipo:** produto incompleto ou secção do produto que permite testar a viabilidade das funcionalidades necessárias.

**Scope Creep:** redefinição exagerada do âmbito, causando impasse no projeto e refletindo-se negativamente no sucesso do projeto.

**Scrum Master:** elemento da equipa com grande conhecimento sobre a metodologia de desenvolvimento.

**Sprint Backlog:** conjunto de *user stories* que se pretendem implementar numa iteração.

**Sprint Burndown Chart:** gráfico de monitorização do progresso do projeto no que toca à produtividade da equipa ao longo das *sprints*.

**Sprint Goal:** objetivo de uma *sprint*.

**Sprint Meeting:** reunião em que o *Product Owner* apresenta à equipa o detalhe das *user stories* a implementar de seguida.

**Sprint Retrospective:** reunião da equipa de desenvolvimento para conseguir otimizar a metodologia utilizada.

**Sprint Review:** reunião onde é realizada uma demonstração dos resultados obtidos numa *sprint*.

***Sprint:*** intervalo de tempo limitado onde é desenvolvido um número definido de funcionalidades, iteração.

***Stakeholders:*** pessoas e/ou organizações sobre a qual o projeto tem impacto.

***Story Board:*** conjunto de ilustrações que representam uma tarefa que o utilizador pretende realizar.

***Task Board:*** quadro que contém o detalhe e caracterização das *user stories* do projeto divididas segundo o seu estado, caracterizando de uma forma simples o progresso do projeto no que toca ao desenvolvimento.

***User Story:*** Descrição de uma tarefa que o produto necessita de cumprir/que o utilizador precisa de conseguir realizar.

## II. Referências/Bibliografia

- [1] Vaishnavi and Kuechler, "Promoting Relevance in IS Research: An Informing System for Design Science Research," *Informing Science: the International Journal of an Emerging Transdiscipline*, vol. 14, 2011.
- [2] PMI, *A guide to the project management body of knowledge (PMBOK guide)*, 5th ed.: Project Management Institute, Inc., 2013.
- [3] Standish Group, "Chaos Report," 2014.
- [4] J. Varajão, J. Cardoso, D. Gonçalves, and J. B. Cruz. (2008) Análise à gestão de projetos de desenvolvimento de software em grandes empresas portuguesas. *Semana Informática, nº 904*.
- [5] PMI, *Project Management Institute Certifications*, 2014.
- [6] K. Schwalbe, *Information Technology Project Management, Revised*, Seventh ed. Boston: Course Technology, 2013.
- [7] A. Murray, N. Bennett, C. Bentley, and C. Great Britain Office of Government. (2009). *Managing successful projects with PRINCE2, 2009 edition manual*.
- [8] I. Group. (2014, 16/11/2016). *PRINCE2*. Available: <https://www.prince2.com/uk>
- [9] P. Dorsey and D. Inc. (2005, 18/11/2016). Top 10 Reasons Why Systems Projects Fail.
- [10] APM, *APM body of knowledge*, Sixth ed.: APM, 2012.
- [11] S. Ohara, *A Guidebook of Project & Program Management for Enterprise Innovation*: Project Management Association of Japan 2005.
- [12] I. P. M. Association, *Individual Competence Baseline for Project, Programme & Portfolio Management*, 4th Version ed.: International Project Management Association, 2015.
- [13] A. Zandhuis and R. Stellingwerf, *ISO 21500 Guidance on Project Management - A Pocket Guide*: Van Haren Publishing, 2013.
- [14] K. Beck, M. Beedle, A. v. Bennekum, A. Cockburn, W. Cunningham, M. Fowler, et al. (2001, 22/11/2016). *Manifesto for Agile Software Development*. Available: <http://agilemanifesto.org/>
- [15] D. Turk, R. France, and B. Rumpe, "Limitations of Agile Software Processes," *Third International Conference on Extreme Programming and Flexible Processes in Software Engineering, XP2002, May 26-30, Alghero, Italy, pg. 43-46, 2002*, 2014.
- [16] K. Schwaber and J. Sutherland, *The Scrum Guide - The Definitive Guide to Scrum: The Rules of the Game*: Scrum.Org and Scrum Inc., 2016.
- [17] J. Wester. (26/11/2016). *What is Kanban?* Available: <http://www.everydaykanban.com/what-is-kanban/>
- [18] D. J. Anderson and A. Carmichael, *Essential Kanban Condensed*, 1st Edition ed.: Lean Kanban University Press, 2016.
- [19] T. Point. (03/12/2016). *Extreme Programming* Available: [http://www.tutorialspoint.com/extreme\\_programming/](http://www.tutorialspoint.com/extreme_programming/)
- [20] D. Wells. (2009, 06/12/2016). *The Values of Extreme Programming*. Available: <http://www.extremeprogramming.org/values.html>
- [21] J. Stapleton, *DSDM, Dynamic Systems Development Method: The Method in Practice*: Addison-Wesley, 1997.
- [22] A. B. Consortium. (07/12/2016). *What is DSDM*. Available: <https://www.agilebusiness.org/what-is-dsdm>

- [23] A. B. Consortium. (2014, 07/12/2016). *The DSDM Agile Project Framework*. Available: <https://www.agilebusiness.org/resources/dsdm-handbooks/the-dsdm-agile-project-framework-2014-onwards>
- [24] M. Poppendieck and T. Poppendieck, *Leading Lean Software Development: Results Are Not the Point*. Addison Wesley Professional, 2009.
- [25] V. Santos and J. Varajão, "PMO as a Key Ingredient of Public Sector Projects' Success – Position Paper," *Procedia Computer Science*, vol. 64, pp. 1190-1199, 2015/01/01/ 2015.
- [26] A. Monteiro, V. Santos, and J. Varajão, "Project Management Office Models – A Review," *Procedia Computer Science*, vol. 100, pp. 1085-1094, 2016/01/01/ 2016.
- [27] M. Henson. (2014, 07/12/2016). *Mature robust and Agile DSDM celebrates being twenty*. Available: <https://www.agilebusiness.org/resources/articles/mature-robust-and-agile-dsdm-celebrates-being-twenty>
- [28] S. Jones. (2016). *Agile Project Management: QuickStart Guide - The Complete Beginners Guide To Mastering Agile Project Management! (Scrum, Project Management, Agile Development)*. Available: <https://www.amazon.com/Agile-Project-Management-QuickStart-Development-ebook/dp/B01FSZI2H4>
- [29] J. C. Goodpasture, *Project Management the Agile Way: Making It Work in the Enterprise*. J. Ross Publishing, Inc. , 2010.
- [30] D. Rawsthorne. (2004, Managing the Work in an Agile Project Available: <http://www.netobjectives.com/files/resources/downloads/ManagingTheWork.pdf>
- [31] J. Carroll and D. Morris, *Agile Project Management in easy steps*, Second ed.: In Easy Steps Limited, 2015.
- [32] J. Nuottila, K. Aaltonen, and J. Kujala, "Challenges of adopting agile methods in a public organization," *International Journal of Information Systems and Project Management*, vol. 4, No 3, 2016.
- [33] E. S. F. Cardozo, J. B. F. A. Neto, A. Barza, A. C. C. França, and F. Q. B. d. Silva, "SCRUM and Productivity in Software Projects: A Systematic Literature Review," *14th International Conference on Evaluation and Assessment in Software Engineering (EASE)*, 2010.
- [34] B. Barton and E. Campbell, "Implementing a Professional Services Organization Using Type C Scrum," *HICSS 2007 - 40th Hawaii International International Conference on Systems Science 3-6 January, 2007, Waikoloa, Big Island, HI, USA*. pp. 275, 2007
- [35] J. Sutherland. (2004), *Agile Development: Lessons Learned from the First Scrum*. Available: <http://www.toolshero.nl/wp-content/uploads/agile-development-lessons-learned-jeff-sutherland.pdf>
- [36] T. Chow and D.-B. Cao, "A survey study of critical success factors in agile software projects," *Journal of Systems and Software*, vol. 81, pp. 961-971, 2008/06/01/ 2008.
- [37] K. Schwaber, *Agile Project Management with Scrum*. Microsoft Press, 2004.
- [38] PMI, "PMI Certifications," 2014.
- [39] L. ILIEȘ, E. CRIȘAN, and I. N. MUREȘAN, "Best Practices in Project Management " *Review of International Comparative Management*, vol. 11, No 11, 2010.
- [40] K. Mar and K. Schwaber, "Scrum with XP," *informIT*, 2002.
- [41] J. Varajão, "Success Management as a PM Knowledge Area – Work-in-Progress," *Procedia Computer Science*, vol. 100, pp. 1095-1102, 2016/01/01 2016.
- [42] J. Varajão, J. Cardoso, D. Gonçalves, and J. B. Cruz. (2008) Eficácia do gestor é crucial na gestão de projetos. *Semana Informática, nº 920*.
- [43] R. Amster and C. Böhm, "Improving intercultural competency in global IT projects through recognition of culture-based behaviors," *International Journal of Information Systems and Project Management*, vol. 4, No 2, 2016.

- [44] O. Stawnicza, "Distributed team cohesion – not an oxymoron. The impact of information and communications technologies on teamness in globally distributed IT projects," *International Journal of Information Systems and Project Management*, vol. 3, No 2, 2015.
- [45] J. Varajão and A. Trigo, "Evaluation of IS project success in InfSysMakers: an exploratory case study," *Thirty Seventh International Conference on Information Systems, Dublin 2016*, 2016.
- [46] A. Paiva, J. Varajão, C. DomÍnguez, and P. Ribeiro, "Principais aspectos na avaliação do sucesso de projectos de desenvolvimento de software. Há alguma relação com o que é considerado noutras indústrias?," vol. 36, No 3, 2011.
- [47] I. C. Moura, P. M. Dias, C. E. Dominguez, and J. E. Varajão, "What Team Members Perceive as Important to Achieve High Performance: An Exploratory Case Study," *Procedia Technology*, vol. 16, pp. 1010-1016, 2014/01/01/ 2014.
- [48] A. Amaral, G. Fernandes, and J. Varajão, "Identifying Useful Actions to Improve Team Resilience in Information Systems Projects," *Procedia Computer Science*, vol. 64, pp. 1182-1189, 2015/01/01/ 2015.
- [49] O. Pankratz and D. Basten, "Ladder to success – eliciting project managers' perceptions of IS project success criteria," *International Journal of Information Systems and Project Management*, vol. 2, No 2, 2014.
- [50] I. Moura and J. Varajão, "Information Systems Students' Perceptions on Leading Teams to High Performance: a Pilot Study," *Twenty-Fourth European Conference on Information Systems (ECIS), İstanbul, Turkey, 2016*, 2016.
- [51] L. Siddique and B. A. Hussein, "A qualitative study of success criteria in Norwegian agile software projects from suppliers' perspective," *International Journal of Information Systems and Project Management*, vol. 4, No 2, 2016.
- [52] M. S. Chaves, C. C. S. d. Araújo, L. R. Teixeira, D. V. Rosa, I. G. Júnior, and C. D. Nogueira, "A new approach to managing Lessons Learned in PMBoK process groups: the Ballistic 2.0 Model," *International Journal of Information Systems and Project Management*, vol. 4, No 1, 2015.