

Mestrado em

Gestão de Projetos

Trabalho Final de Mestrado

Dissertação

**Metodologias Híbridas no Desenvolvimento de Sistemas
Embebidos**

João Luís Boleto Tavares

Outubro - 2019

Mestrado em

Gestão de Projetos

Trabalho Final de Mestrado

Dissertação

**Metodologias Híbridas no Desenvolvimento de Sistemas
Embebidos**

João Luís Boletto Tavares

Orientação:

Professora Doutora Filipa Pires da Silva

Outubro - 2019

Aos meus pais, Ana e Luís Pedro.

Agradecimentos

Este trabalho é o culminar de um longo caminho de crescimento pessoal e profissional. Nasce da vontade insaciável de aprender, descobrir, de fazer a diferença na vida das pessoas e das organizações através do conhecimento e da competência. No decorrer desta caminhada, tive de sacrificar momentos de família e de amizade. A todos, queria pedir desculpa pela ausência e agradecer a força e a confiança que sempre me transmitiram ao longo desta jornada.

Aos meus pais que são os meus pilares e que serão sempre o meu norte e a quem devo quem sou. Pela dedicação e arte que tiveram ao transmitirem-me as ferramentas que precisava para voar, por me plantar um espírito crítico e construtivo em tudo o que faço e vejo, por acreditarem no meu talento e nas minhas escolhas e pela partilha desta viagem incrível que temos percorrido juntos.

À minha Joana, que me apoiou sempre a fazer aquilo que mais gosto, que me mostrou que podia ser mais do que aquilo, que sonhava ser. A ela que sempre foi cor nos dias mais cinzentos, sempre foi luz no breu das noites mais longas, sempre foi e sempre será amor nesta aventura única que é a vida.

À Professora Doutora Filipa Pires da Silva, por ser sempre tão disponível, por todo o apoio, conselhos e sugestões, assim como a todos aqueles que tiveram a amabilidade de participar nas entrevistas e no questionário que permitiram a concretização deste trabalho.

Outubro de 2019

João Tavares

Resumo

Nos últimos 20 anos, as metodologias ágeis têm vindo a afirmar-se como uma referência metodológica nos projetos de desenvolvimento de *software*. Esta realidade alicia muitos gestores a identificarem a necessidade de implementar esta abordagem no desenvolvimento de sistemas embebidos. Todavia, esta tarefa torna-se complexa devido à natureza deste tipo de projetos. Por esta razão têm surgido propostas de utilização de metodologias híbridas, que são uma abordagem de desenvolvimento que combina práticas ágeis e tradicionais. No entanto, a sua utilização atual tem sido relatada como pouco estruturada, não havendo um enquadramento claro dos benefícios para o desenvolvimento destes sistemas específicos. Assim, este estudo pretende aprofundar o conhecimento atual sobre as práticas correntes no âmbito da utilização das metodologias, explorando e mapeando-as, em particular no que toca às metodologias híbridas. Foram realizadas quatro entrevistas semi-estruturadas para ganhar um maior entendimento sobre o contexto que leva à utilização de metodologias híbridas no desenvolvimento de sistemas embebidos. Posteriormente, foi aplicado um questionário a 132 profissionais da área dos quais se 68 forneceram respostas válidas. A análise dos resultados revelou que as metodologias mais utilizadas em Portugal não diferem das anteriormente reportadas na literatura. Porém, foi possível estabelecer uma dependência estatística entre as metodologias *Scrum* e *Waterfall*, confirmando a existência e utilização da metodologia híbrida *Water-Scrum-Fall*.

Palavras-Chave: Metodologias Híbridas, Sistemas Embebidos, Metodologias Ágeis, Metodologias Tradicionais, *System Development Life Cycles*.

Abstract

Across the last 20 years, agile methodologies have substantially increased their influence on software and systems development. This reality encourages many managers to identify the need to implement this approach in embedded systems development projects. However, this task becomes very complex due to the nature of such development projects. For this reason, efforts have been made to use hybrid methodologies, which are a development approach that combines agile and traditional practices. Nevertheless, this use has been reported as unstructured and without a clear framework for the benefits to the development of these specific systems. Thus, this study intends to expand the current knowledge about current practices in the use of methodologies, exploring and mapping them, namely concerning hybrid methodologies. Four semi-structured interviews were conducted to acquire a deeper understanding of the reasons that lead to the use of hybrid methodologies in the development of embedded systems. Subsequently, a questionnaire was applied to 132 professionals in the area of embedded system development, from which 68 provided valid answers were collected. The analysis of the results revealed that the most used methodologies in Portugal do not differ from those previously reported in the literature. Still, it was possible to establish a statistical dependence between Scrum and Waterfall methodologies, confirming the existence and application of the Water-Scrum-Fall hybrid methodology.

Keywords: Hybrid Methodologies, Embedded Systems, Agile, Tradicional Methodologies, System Development Life Cycles.

Índice

Lista de Figuras	xi
Lista de Tabelas	xi
Lista de Acrónimos	xii
1 Introdução	1
2 Revisão da Literatura	5
2.1 Sistemas Embebidos	5
2.2 Sistemas Embebidos <i>vs Software</i>	6
2.3 Projetos e Metodologias de Desenvolvimento	9
2.4 Metodologias <i>System Development Life Cycle</i>	9
2.4.1 <i>Waterfall</i>	10
2.4.2 <i>V-Model</i>	11
2.4.3 <i>Scrum</i>	12
2.5 Metodologias Híbridas	13
2.5.1 <i>Agile Stage-Gate</i>	14
2.5.2 <i>Water-Scrum-Fall</i>	16
2.5.3 <i>Disciplined Agile</i>	17
3 Metodologia de Investigação	19
3.1 Desenho da Investigação	19
3.2 Entrevistas semi-estruturadas	20
3.3 Questionário	21

ÍNDICE

4	Análise e Discussão de Resultados	25
4.1	Análise das Entrevistas	25
4.2	Questionário	26
4.2.1	Metodologias e Práticas Utilizadas	27
4.2.2	Combinações de Metodologias e Abordagens	30
5	Conclusões, Limitações e Trabalho Futuro	33
	Referências Bibliográficas	35
	Anexos	41
	Anexo A	43
	Anexo B	50
	Anexo C	52
	Anexo D	56

Lista de Figuras

1.1	Nova Estrutura Tecnológica (adaptado de Porter e Heppelmann (2015))	2
2.1	Modelo de Sistemas Embebidos (adaptado de Wang, Wang, Huang, Wen, e Zhang (2017))	6
2.2	Modelo <i>Waterfall</i> (adaptado de Kneuper (2018))	11
2.3	Modelo V (adaptado de Kneuper (2018))	11
2.4	Modelo <i>Scrum</i> (Schwaber, 2010)	13
2.5	Modelo <i>Water-Scrum-Fall</i> (adaptado de West (2011))	16
2.6	Modelo <i>Disciplined Agile</i> (Disciplined Agile, 2013)	18
3.1	Visão geral da abordagem de investigação	19
4.1	Utilização de metodologias	28
4.2	Combinações de metodologias	31

Lista de Tabelas

I	Tabela de contingência da utilização de metodologias <i>Scrum</i> e <i>Waterfall</i>	32
I	Protocolo da entrevista semi-estruturada	43
II	Guião da entrevista	44
III	Estrutura do Questionário	45

III	Estrutura do Questionário	46
III	Estrutura do Questionário	47
III	Estrutura do Questionário	48
III	Estrutura do Questionário	49
IV	Metodologias agregadas por categoria/abordagem	49
V	Quadro de análise de respostas das entrevistas semi-estruturadas	51
VI	Dimensão da organização dos respondentes	52
VII	Dimensão dos projetos no qual se os inquiridos basearam as suas respostas	52
VIII	Experiência Profissional no desenvolvimento de sistemas/software	52
IX	Faixa Etária dos inquiridos	52
X	Género dos inquiridos do questionário	52
XI	A sua organização possui uma Metodologia ou Processo Padrão para o desenvolvimento de sistemas?	53
XII	De que forma as abordagens, métodos e práticas foram implementados na sua organização?	53
XIII	Quais são os objetivos gerais que se pretende alcançar com a seleção das abordagens, métodos e práticas de desenvolvimento?	53
XIV	De que forma foi definido o processo de gestão do seu projeto de desenvolvimento?	54
XV	Utilização de Metodologias de Desenvolvimento	54
XVI	Utilização de Práticas de Desenvolvimento	55
XVII	Utilização de Pares de Metodologias Combinadas	56
XVIII	Teste do Qui-Quadrado das variáveis <i>Waterfall</i> e <i>Scrum</i>	56
XIX	Coefficientes de Associação Estatística das Combinação <i>Waterfall</i> * <i>Scrum</i>	56
XX	Resultados da agregação de Metodologias Utilizadas por Categoria	57
XXI	Tabela de Contingência das Categorias Ágil e Tradicional	57
XXII	Teste de Hipóteses da Combinação de Categoria Ágil e Tradicional	57

Lista de Acrónimos

CPS *Cyber-Physical Systems*

HELENA *Hybrid Development Approaches in Software Systems Development*

IoT *Internet of Things*

SDLC *System Development Live Cycle*

Capítulo 1

Introdução

Nos últimos 50 anos, as tecnologias da informação alteraram radicalmente o panorama da competitividade e da estratégia empresarial. A automatização das atividades individuais na cadeia de valor despoletou um aumento dramático da produtividade, permitindo a construção de cadeias de valor globais (Porter e Heppelmann, 2014). Recentemente, as inovações tecnológicas deram origem ao conceito de *Internet of Things* (IoT), no qual uma variedade de dispositivos que são capazes de interagir e cooperar uns com os outros de forma a atingir objetivos comuns (Giusto, Iera, Morabito, e Atzori, 2010). Na base dos produtos enquadrados no conceito de IoT estão tipicamente sistemas embebidos, que permitem uma interface e experiência de utilizador melhorada (Porter e Heppelmann, 2015).

Os sistemas embebidos realizam a integração do processamento de informação com o ambiente físico, em conjugação com um vasto leque de possibilidades de armazenamento de dados e conectividade, criando melhorias na funcionalidade e na *performance* dos produtos (Marwedel, 2018). Segundo Porter e Heppelmann (2015), estas características requerem que as empresas construam uma nova infraestrutura tecnológica de suporte, constituída por várias módulos que devem cooperar entre si (Figura 1.1).

Na nova estrutura tecnológica, os sistemas embebidos desempenham um papel fundamental na transformação digital (Conforto e Amaral, 2015). Como resultado, as equipas de desenvolvimento sentem cada vez mais pressão para entregar produtos de qualidade,

1. INTRODUÇÃO

em ciclos de desenvolvimento cada vez mais curtos e iterativos, de forma a dar a resposta adequada às exigências do mercado (R. Cooper e Sommer, 2016). Cockburn e Highsmith (2001) defendem que o mundo dos negócios e da tecnologia se caracteriza como turbulento, de alta velocidade e com elevado grau de incerteza, o que requer processos que criem mudanças e que respondam rapidamente a essas mesmas mudanças.

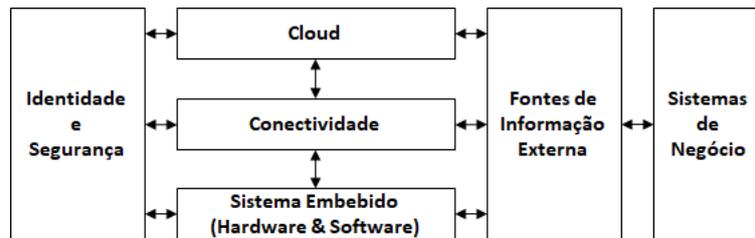


Figura 1.1: Nova Estrutura Tecnológica (adaptado de Porter e Heppelmann (2015))

Como consequência, a natureza dos sistemas embebidos e a sua integração criam uma complexidade acrescida nos projetos de desenvolvimento de produto (Conforto e Amaral, 2015). As empresas estão a mudar a forma como desenvolvem produtos, isto porque necessitam de desenvolver produtos de qualidade, com custo reduzido, numa janela temporal bem definida, de forma a corresponderem ao *time-to-market*. Segundo Papadakis e Tsironis (2018), as alterações nos processos de desenvolvimento são causadas fundamentalmente pela alteração das necessidades dos clientes e pelo aumento da complexidade dos produtos. Isto leva a uma eficiência limitada das metodologias atuais, despoletando a necessidade de incorporação de novas metodologias. Neste sentido, as empresas precisam de implementar metodologias apropriadas que lhes permitam acompanhar o ritmo do mercado (Graaf, Lormans, e Toetenel, 2003).

As metodologias ágeis têm vindo a apresentar resultados positivos no desenvolvimento de *software*, melhorando a qualidade e a velocidade de desenvolvimento de produto (Rigby, Sutherland, e Takeuchi, 2016). Isto leva a que muitos gestores identifiquem a necessidade de as implementar no desenvolvimento de sistemas embebidos (R. Cooper e Sommer, 2016). No entanto, segundo Freitas (2015), a implementação de metodologias ágeis no desenvolvimento de sistemas embebidos tem vindo a revelar-se uma tarefa complexa devido à natureza que este tipo de desenvolvimento de produto encerra em si mesmo. Para colmatar estas dificuldades, alguns autores têm sugerido

diferentes abordagens, nomeadamente a combinação dos diferentes processos de desenvolvimento de *software* e *hardware* (Hendler, 2018). Estas abordagens, designadas habitualmente por metodologias híbridas, procuram equilibrar os benefícios e desafios de dois modelos, criando um número significativo de vantagens (R. Cooper, 2016).

Dada a pouca investigação destas abordagens, a literatura existente tem relatado a sua utilização como pouco estruturada, não existindo ainda um enquadramento claro dos benefícios no desenvolvimento destes sistemas específicos. Assim, este trabalho pretende contribuir para o aprofundar do conhecimento atual da utilização de metodologias e práticas no desenvolvimento de sistemas embebidos. Em particular, pretende-se explorá-las e mapeá-las, nomeadamente no que concerne às metodologias híbridas, permitindo responder aos desafios do desenvolvimento destes sistemas considerando as suas características, as estruturas organizacionais e a cadeia de valor existente. Foram então definidas as seguintes questões de investigação:

- Q1: Que abordagens são utilizadas no desenvolvimento de sistemas embebidos nas prática? Esta questão de investigação pretende desenvolver um catálogo das abordagens de desenvolvimento de sistemas embebidos utilizadas (metodologias e práticas).
- Q2: Que metodologias são utilizadas como base para a conceção de metodologias híbridas? Esta questão pretende analisar as estruturas e os métodos mais abrangentes que formam a base dos métodos híbridos e unem as diferentes práticas.

Através de uma abordagem exploratória e qualitativa, considerou-se a utilização de métodos de recolha de dados mistos. Numa primeira fase, realizaram-se entrevistas semi-estruturadas a gestores de projetos de desenvolvimento de sistemas embebidos. Posteriormente, foi aplicado um questionário cuja recolha e o tratamento dos dados permitiram a construção de um estudo caracterizado por uma janela de tempo definida (*cross-sectional study*).

Este trabalho está organizado em cinco capítulos. O primeiro capítulo, pretende dar ao leitor uma breve contextualização do tema do trabalho realizado. O segundo capítulo

1. INTRODUÇÃO

enquadra o leitor com os conceitos, processos e metodologias mais relevantes no desenvolvimento de sistemas embebidos, de forma a proporcionar o entendimento necessário para a interpretação dos resultados apresentados. O terceiro capítulo pretende descrever de forma detalhada a abordagem de investigação, os instrumentos de recolha de dados utilizados, assim como as metodologias de tratamento de dados e respetivos critérios de aceitação definidos. O quarto capítulo apresenta os resultados obtidos assim como o respetivo tratamento estatístico realizado, por forma a proporcionar respostas às questões de investigação. O quinto e último capítulo apresenta as principais conclusões do trabalho realizado, expõe as limitações das metodologias e amostra utilizadas e inclui sugestões para estudos futuros.

Capítulo 2

Revisão da Literatura

2.1 Sistemas Embebidos

Marwedel (2018) define os sistemas embebidos como sistemas de processamento de informação encapsulados num produto ou num contexto físico específico. Os sistemas embebidos são definidos como sistemas baseados num processador com finalidades específicas e baseados em tecnologias (*hardware* e *software*), que são personalizados de forma a alcançar a fiabilidade, custo, volume e capacidade energética que é estritamente necessária para a execução das funcionalidades pretendidas (Wang et al., 2017). A importância da ligação dos sistemas embebidos com os sistemas físicos é realçada através do conceito de *Cyber-Physical Systems* (CPS) que Lee (2012) define como sendo sistemas que integram a componente computacional com a de processos físicos, formando esta relação entre os sistemas embebidos e os sistemas físicos.

Os sistemas embebidos são constituídos essencialmente por dois blocos: *hardware* e *software*, podendo ter apenas uma camada de *hardware* ou ser constituído por várias camadas, nomeadamente *hardware*, *software* de sistema e *software* aplicacional (Wang et al., 2017) (Figura 2.1).

2. REVISÃO DA LITERATURA

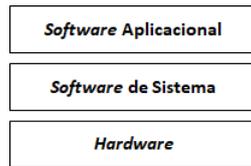


Figura 2.1: Modelo de Sistemas Embebidos (adaptado de Wang et al. (2017))

Desenvolver sistemas com estas características requer conhecimentos em diferentes áreas, pelo que as equipas que os desenvolvem são habitualmente multidisciplinares (Forrai, 2013). É frequente o envolvimento de *designers*, engenheiros de *hardware*, engenheiros de *software*, engenheiros de sistemas, engenheiros mecânicos, especialistas em qualidade, entre outros.

A principal característica dos sistemas embebidos é a sua orientação para uma aplicação específica e para a interação com o utilizador, o que lhe confere requisitos funcionais muito claros (Kaisti et al., 2013). Do ponto de vista físico, os sistemas embebidos tendem a apresentar dimensões reduzidas, baixos consumos energéticos, custos de produção baixos e um alto grau de integração com o ambiente de aplicação envolvente (Wang et al., 2017). Um dos requisitos fundamentais dos sistemas embebidos é a utilização de um sistema operativo em tempo real, de forma a potenciar a velocidade e a eficiência do sistema (Marwedel, 2018). Não existe uma diferenciação vincada entre o sistema operativo e as aplicações de *software* como no caso dos computadores pessoais, isto porque todo o *software* desenvolvido é otimizado e integrado para obter as funcionalidades pretendidas com o mínimo de recursos possíveis (Wang et al., 2017). Os sistemas embebidos são frequentemente parte de um todo mais vasto constituído por outros sistemas, embebidos ou de outra natureza, habitualmente designado como Sistema de Sistemas (Maier, 1999).

2.2 Sistemas Embebidos vs *Software*

Os sistemas embebidos diferem das aplicações de *software* convencionais em vários aspetos. O desenvolvimento de sistemas embebidos consiste no desenvolvimento de *hardware* e *software*, que são partes comuns de um todo muito mais abrangente e cuja

2.2 Sistemas Embebidos vs Software

complexidade tem vindo a aumentar (Kaisti et al., 2013).

Uma das principais diferenças entre o desenvolvimento de *software* aplicacional e de sistemas embebidos é o impacto significativo que uma alteração num componente pode ter no sistema (Könnölä et al., 2016). Greene (2004) realça que a ligação entre a camada de *hardware* e de *software* (operativo e/ou aplicacional) é especialmente sensível a alterações nos sistemas embebidos. Desta forma, Vishwas e Satao (2012) dividem os projetos de desenvolvimento de *software* e *sistemas* em 3 classes em função do esforço de desenvolvimento (em linhas de código): projetos orgânicos, caracterizado por equipas experientes e de tamanho reduzido cujo desenvolvimento é baseado em requisitos pouco rígidos; projetos mistos que são caracterizados por equipas de tamanho médio com um nível de experiência mista e cujo desenvolvimento é baseado em requisitos mistos (rígidos e flexíveis) e; projetos embebidos cujo desenvolvimento é limitado por constrangimentos técnicos apertados, onde o projeto apresenta uma combinação das características dos projetos orgânicos e mistos supramencionadas.

A natureza do *software* aplicacional permite ciclos de desenvolvimento pequenos, onde são implementados, testados e lançados em produção novas funcionalidades ou melhorias de forma célere e simples (Lwakatare et al., 2016). A natureza do *hardware* leva à necessidade de um ciclo de desenvolvimento mais longo e com tarefas sistemáticas, tais como desenho do esquema eléctrico, *design* da placa de circuito impresso, produção de protótipos e testes de validação (Könnölä et al., 2016). Para além disto, o *software* aplicacional pode ser desenvolvido orientado ao objeto, o que permite criar uma estrutura modular, e que desbloqueia o impacto das alterações no sistema no seu todo. Por sua vez, nos sistemas embebidos existe uma limitação substancial da subdivisão do sistema e da execução de uma arquitectura modular e flexível (Selnes, Visser, Sinha, e Elverum, 2017).

Os ambientes e ferramentas nos quais se enquadram o desenvolvimento de *software* e de sistemas embebidos são também distintos (Xie, Shen, Rong, e Shao, 2012). O desenvolvimento de *software* é, na sua grande maioria, aplicacional, seja no domínio *Web*, seja em sistemas operativos altamente padronizados. As suas ferramentas de

2. REVISÃO DA LITERATURA

desenvolvimento oferecem a possibilidade de testar e simular de forma rápida, fiável e acessível o ambiente de produção (Lwakatare et al., 2016). O desenvolvimento de sistemas embebidos aplica-se a domínios e sistemas operativos com um elevado grau de personalização e cujo ambiente de produção é extremamente específico e difícil de testar e simular antecipadamente (Xie et al., 2012).

A planificação do trabalho a executar também apresenta diferenças no desenvolvimento de *software* e sistemas embebidos. Existe uma necessidade constante de considerar os requisitos previamente definidos e que podem sofrer alterações. No caso do *software* aplicacional, a planificação do trabalho tem como base requisitos funcionais que são convertidos em tarefas de implementação de funcionalidades (Selnes et al., 2017). Por outro lado, no caso dos sistemas embebidos, além dos requisitos funcionais, torna-se incontornável considerar requisitos não funcionais e as limitações tecnológicas do *hardware*.

O aumento do número de requisitos funcionais e da sua complexidade em sistemas embebidos, como é exemplo a ligação à *internet* ou a integração de *Displays* sofisticados, implica um aumento substancial de *software* que é adicionado à estrutura simples de um sistema embebido. No entanto, a integração de vários componentes de *software* a um sistema embebido é uma tarefa de grande complexidade e que pode levar a deficiências funcionais e de segurança (Marwedel, 2018).

Também os custos de ciclos de desenvolvimento em *software* e em *hardware* são diferentes (Selnes et al., 2017). Os custos associados a um ciclo de desenvolvimento de *software* são reduzidos e incidem nos honorários da equipa de desenvolvimento e nas plataformas de desenvolvimento digital que estes utilizam. No caso do *hardware*, os custos de produção de protótipos, aquisição de instrumentação ou a contratação de serviços especializados de testes de validação são elevados e com janelas temporais próprias, o que limita a possibilidade de executar muitas iterações de desenvolvimento (Selnes et al., 2017).

2.3 Projetos e Metodologias de Desenvolvimento

Um projeto é definido como um esforço único e transitório realizado para alcançar um resultado desejado (Association for Project Management, 2012). No contexto da execução do projeto destacam-se tarefas que podem ser agrupadas em fases de execução, habitualmente designadas por ciclo de vida dos projetos (Project Management Institute, 2008). O ciclo de vida do projeto é um conjunto de fases que de uma forma generalizada são sequenciais, cujo nome e o número são determinados pelas necessidades das organizações, a natureza do projeto e a sua área de aplicação Project Management Institute (2008). Os projetos de desenvolvimento de sistemas embecidos têm características e necessidades particulares que levaram à existência de vários modelos de ciclos de vida específicos para este tipo de projetos, denominados frequentemente como *System Development Live Cycle* (SDLC) (Kaisti et al., 2013).

A natureza complexa dos sistemas embecidos necessita de uma metodologia que configure, defina e monitorize o progresso do desenvolvimento ao longo de todas as etapas do ciclo de vida do sistema (Isaias e Issa, 2015). Apesar da escolha da metodologia a utilizar depender das características peculiares de cada projeto, existem aspetos chave que todas as metodologias devem integrar, sendo uma das mais notáveis a segmentação do processo de desenvolvimento em fases distintas (Isaias e Issa, 2015). Os mesmos princípios são aplicáveis não apenas a processos de desenvolvimento interno de sistemas, como também a processos de desenvolvimento de sistemas contratualizados a fornecedores externos (Cohen, 2010).

2.4 Metodologias *System Development Life Cycle*

Leau, Loo, Tham, e Tan (2012) definem um SDLC como uma metodologia que descreve todo o processo de desenvolvimento, no qual uma organização de desenvolvimento de sistemas deve utilizar para garantir o sucesso do projeto. Segundo Whitten e Bentley (2007) um SDLC é um processo que define as necessidades, desenha uma solução e implementa essa solução. Geralmente, um SDLC é constituído por 5 fases distintas: identificação do problema e planeamento, análise, *design*, implementação e manutenção

2. REVISÃO DA LITERATURA

(Hedman e Lind, 2009).

O objetivo nuclear do desenvolvimento de sistemas embebidos é a sua eficiência de integração em situações da vida real, pelo que o levantamento das necessidades das diferentes pessoas envolvidas na utilização do sistema e do contexto em que o sistema irá operar, são dois fatores críticos de sucesso em projetos desta natureza (Tetlay e John, 2009). A desinformação ou a falta de qualidade da mesma tem repercussões no desenvolvimento do sistema e geralmente resulta na implementação de produtos que ficam aquém das expectativas, tanto do ponto de vista dos utilizadores como do ponto de vista da produtividade (Isaias e Issa, 2015).

Existem diversas metodologias de desenvolvimento de sistemas propostas na literatura que pretendem minimizar os erros e falhas que levam frequentemente ao insucesso dos projetos (Isaias e Issa, 2015). A utilização de um modelo SDLC menos adequado às características e contexto do projeto pode atrasar e afetar a satisfação do cliente e, em última instância, levar ao cancelamento do desenvolvimento do sistema (Executive Brief, 2008). Assim, torna-se crucial estudar as principais características, vantagens e desvantagens das metodologias propostas na literatura. De seguida serão apresentadas as metodologias mais representativas da diversidade existente para o desenvolvimento de sistemas tecnológicos, onde se enquadram os sistemas embebidos.

2.4.1 *Waterfall*

A Metodologia *Waterfall*, citada pela primeira por Royce (1970), é caracterizada por um processo sequencial composto por fases distintas. Estas fases do processo são habitualmente designadas como análise de requisitos, *design*, implementação, teste, lançamento e manutenção. Esta abordagem é também referida como a maior representante das metodologias tradicionais onde também está representado o modelo *Stage-Gate* (Petersen, Wohlin, e Baca, 2009).

Cada fase da metodologia é caracterizada por um conjunto de entregáveis com critérios de aceitação definidos e que servem como base para a decisão se o projeto

2.4 Metodologias *System Development Life Cycle*

deverá avançar para a fase seguinte (Figura 2.2). O gestor de projeto é responsável por garantir a satisfação desses critérios (R. Cooper e Sommer, 2016).



Figura 2.2: Modelo *Waterfall* (adaptado de Kneuper (2018))

Na fase de requisitos, as necessidades dos clientes são identificadas e documentadas com alto nível de abstração. Após esta tarefa, os requisitos são refinados de forma a que possam ser utilizados na fase de seguinte. Na fase de desenho do sistema e implementação, a arquitetura do sistema é projetada, documentada e serve de base para as atividades de desenvolvimento por parte das equipas. No final desta fase realizam-se verificações sobre se o desenvolvimento realizado respeita a arquitetura previamente definida. Na fase de testes realiza-se a avaliação da integração do sistema, tendo em conta aspetos funcionais e de qualidade. Nesta fase, são recolhidos dados de forma a criar ferramentas de apoio à decisão para o lançamento do produto (Petersen et al., 2009).

2.4.2 *V-Model*

O *V-Model* foi apresentado pela primeira vez em 1991 no Simpósio NCOSE em Chattanooga por Forsberg e Mooz (1991) e representa uma variação da metodologia *Waterfall*. A sua representação gráfica toma a forma da letra V (Figura 2.3) com o objetivo de aumentar a eficiência de integração entre as atividades de desenvolvimento e de teste (Mathur e Malik, 2010).

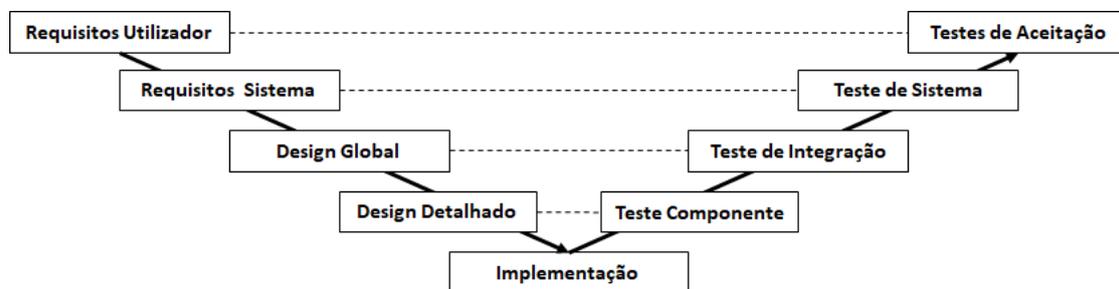


Figura 2.3: Modelo V (adaptado de Kneuper (2018))

O fluxo esquerdo do modelo representa a evolução dos requisitos dos componentes do sistema, através de um processo de decomposição e definição. O fluxo direito

2. REVISÃO DA LITERATURA

representa a integração e a verificação dos componentes do sistema em sucessivos níveis de implementação e integração (Ruparelia, 2010). Com o decorrer do projeto, existe uma análise sistemática dos requisitos e identificação e redução dos riscos. Este processo é representado através das atividades apresentadas ao longo do eixo vertical do modelo no sentido descendente (Forsberg e Mooz, 1991). A estrutura do modelo leva a que quanto mais complexo for o sistema a desenvolver, maior a profundidade que a forma V assume (Ruparelia, 2010). O fluxo temporal das atividades segue a sequência que inicia no topo do fluxo da esquerda e que finaliza no topo do fluxo da direita (Clark, 2009). Segundo Mathur e Malik (2010), as atividades de testes, nomeadamente o *design* dos testes, podem desenrolar-se no início do projeto de forma a otimizar o tempo necessário para a sua conclusão.

A simetria do processo permite que os procedimentos de verificação e de garantia de qualidade sejam definidos para cada atividade de desenvolvimento, assegurando desta forma que os requisitos e o *design* são verificados de uma forma SMART¹ (Doran, 1981). Segundo Sabale e Dani (2013), o modelo V oferece maiores possibilidades de sucesso comparativamente ao Modelo *Waterfall*, devido à definição e realização de módulos de testes durante todo o ciclo de vida de desenvolvimento do sistema. A validação do sistema é demasiado importante para que seja apenas realizada nas fases finais dos projetos (Mathur e Malik, 2010).

2.4.3 *Scrum*

A metodologia *Scrum* baseia o seu processo no Manifesto Ágil (Beedle et al., 2001). Através de uma abordagem empírica, o processo recolhe dados que permitem a tomada de decisões de forma iterativa. Além de iterativo, o *Scrum* é fundamentalmente incremental, definindo ciclos de desenvolvimento curtos, entre dois a quatro semanas, designados por *sprints* (Paasivaara, Durasiewicz, e Lassenius, 2009). Cada *sprint* procura adicionar valor com base nas aprendizagens acumuladas dos *sprints* anteriores (Schwaber e Sutherland, 2017) (Figura 2.4). Em cada *sprint* a equipa seleciona uma parte desses itens e colocam-nos no *sprint backlog* comprometendo-se a entregá-los no final do *sprint*. An-

¹*Specific, Measurable, Achievable, Realistic e Time-Bound*

2.5 Metodologias Híbridas

tes de terminar cada *sprint*, existe uma reunião de retrospectiva e de revisão do que foi implementado e dos resultados obtidos (Cohen, 2010).

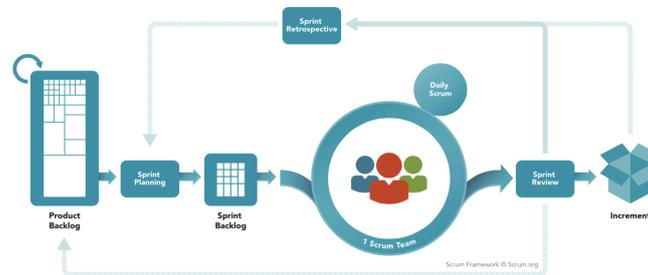


Figura 2.4: Modelo *Scrum* (Schwaber, 2010)

A equipa de *Scrum* é constituída por um *product owner*, uma equipa de desenvolvimento e um *scrum master*. O *scrum master* é responsável por desbloquear problemas e manter a equipa focada nos objetivos. A equipa de desenvolvimento é responsável por desenvolver e entregar o âmbito acordado para cada *sprint*. O *product owner* é o responsável por gerir e priorizar a lista de funcionalidades, requisitos, correções ou alterações que devem ser realizadas no produto (Cohen, 2010). Esta lista designa-se por *product backlog* (Schwaber e Sutherland, 2017).

2.5 Metodologias Híbridas

Kuhrmann et al. (2017) define uma metodologia híbrida como uma abordagem de desenvolvimento de sistemas ou de *software* que utilize qualquer combinação de práticas ágeis e tradicionais adotada por uma unidade organizacional ao seu contexto de trabalho, através da personalização da sua cultura, processos, estrutura organizacional e tecnologia. Kneuper (2018) considera que são metodologias que resultam da combinação de diferentes modelos ágeis e tradicionais.

Jones (2003) analisou aproximadamente 12 mil projetos, em que apesar da diversidade de métodos utilizados, identificou um padrão recorrente na utilização de metodologias híbridas. Kuhrmann e Fernandez (2015) apresentaram a utilização de uma quantidade substancial de abordagens de desenvolvimento aplicadas e combinadas entre si. Garousi, Coskunçay, Betin-Can, e Demirörs (2015) evidenciaram o

2. REVISÃO DA LITERATURA

crescimento da utilização de metodologias tradicionais combinadas com abordagens de desenvolvimento ágeis. Vijayarathy e Butler (2016) reforçaram que as abordagens tradicionais e ágeis coexistem e constituem a maioria das metodologias híbridas em prática. Kneuper (2018) defende que a principal razão para a necessidade da utilização de metodologias híbridas passa pela existência de muitos casos práticos em que nem as metodologias ágeis, nem as metodologias tradicionais endereçam por si só todas as questões.

Segundo Hendler (2018), as metodologias de desenvolvimento híbridas equilibram os benefícios e desafios das duas abordagens, criando um número significativo de vantagens. Estas permitem aos gestores garantir o planeamento do trabalho de forma a realizar estimativas, análise e controlo das atividades e às equipas de desenvolvimento garantir uma metodologia de trabalho ágil, flexível e autónoma (Theocharis, Kuhrmann, Münch, e Diebold, 2015).

Segundo (Bohmer et al., 2018), as metodologias ágeis têm um impacto positivo no desenvolvimento de sistemas embebidos, apesar da sua utilização ainda não se ter propagado na indústria de forma evidente. Todavia, Könnölä et al. (2016) defendem que estas metodologias necessitam de ser personalizadas aos sistemas embebidos, dadas as características singulares. Contudo, as metodologias híbridas carecem de amadurecimento e maior rigor na sua definição (Cohen, 2010). Seguidamente irão ser apresentadas as metodologias híbridas com maior relevância na literatura.

2.5.1 *Agile Stage-Gate*

R. Cooper e Sommer (2016) defendem que as metodologias ágeis e *Stage-Gate* podem ser utilizadas, em combinação, no desenvolvimento de sistemas e de *software*. O modelo *Stage-Gate* tem uma natureza de macroplaneamento e as metodologias ágeis são processos de microplaneamento de gestão de projetos. Esta abordagem híbrida pretende entregar o melhor de cada metodologia, desbloqueando aquilo a que designaram como a nova geração do modelo *Stage-Gate* (Sommer, Hedegaard, Dukovska-Popovska, e Steger-Jensen, 2015).

2.5 Metodologias Híbridas

Boehm e Turner (2003) realçam que uma das principais razões para adotar esta metodologia é alcançar agilidade e disciplina no processo de desenvolvimento. Esta combinação permite melhorias na resposta às alterações das necessidades do mercado na forma de gerir os recursos disponíveis, e no tempo do ciclo de desenvolvimento (Hendler, 2018).

No momento em que um projeto é aprovado e os requisitos iniciais são definidos, as metodologias ágeis garantem o foco na execução dos pacotes de trabalho necessários através do desenvolvimento de vários *sprints* (R. Cooper, 2016). As ferramentas de microplaneamento que as metodologias ágeis possuem, conferem ao processo controlo diário do trabalho e relatórios de progresso (Taivalsaari e Mikkonen, 2018). Dackhammar e Ek (2017) realçam que um dos principais benefícios de incluir metodologias ágeis passa pela substituição de documentos escritos por reuniões presenciais, que melhoram a comunicação e a receção de *feedback* dos clientes.

O desenvolvimento de sistemas insere-se invariavelmente num ambiente complexo que deve ser gerido e coordenado de forma concorrente (Dackhammar e Ek, 2017). Neste contexto, o modelo *Stage-Gate* desempenha um papel fundamental na comunicação entre as equipas de projeto e as várias partes interessadas (Karlström e Runeson, 2006). No mesmo sentido, este possibilita a apresentação de um plano com uma visão temporal mais alargada que as habituais duas a quatro semanas dos *sprints* ágeis (Taivalsaari e Mikkonen, 2018).

Através da metodologia *Agile-Stage-Gate* os requisitos para uma solução estão, na sua maioria, definidos antes de entrar na fase de desenvolvimento. Isto porque as fases iniciais do processo abordam os aspetos de mercado, técnicos e de negócio de forma a definir o sistema que se pretende desenvolver e a justificar financeiramente o projeto. No caso dos requisitos de produto sofrerem alterações, o *design* pode ser modificado num estado inicial quando o custo de alteração no desenvolvimento de sistemas embebidos é ainda reduzido (Hendler, 2018). De forma a consolidar esta flexibilidade, a reunião de planeamento de desenvolvimento deverá concentrar-se principalmente na priorização,

2. REVISÃO DA LITERATURA

considerando a interdependência entre tarefas. Esta abordagem torna possível não só a gestão dos requisitos, bem como a alteração de prioridades, com base numa visão holística do sistema (Könnölä et al., 2016).

A aceleração do desenvolvimento é conseguida através de *time-boxed sprints*, que conferem uma sensação de urgência no desenvolvimento do projeto. As equipas de projeto comprometem-se na entrega de certos entregáveis numa determinada janela temporal. Isto força a equipa a focar-se no essencial e a entregar resultados (Hendler, 2018). A implementação de equipas dedicadas melhora os níveis de cooperação e de comunicação, um fator chave na diminuição do *time-to-market* e no aumento do sucesso de desenvolvimento do produto (Hendler, 2018).

2.5.2 *Water-Scrum-Fall*

Em 2011, West (2011) definiu o termo *Water-Scrum-Fall*, apresentando a hipótese de que as metodologias híbridas tornar-se-iam o novo modelo padrão do desenvolvimento de sistemas. Este termo foi apresentado não como uma metodologia desenhada de forma deliberada, mas como resultado da falha na implementação de abordagens ágeis genuínas. No entanto, atualmente a metodologia *Water-Scrum-Fall* é considerada frequentemente como uma combinação útil de vantagens entre as metodologias ágeis e tradicionais, bem como a mais utilizada (Kneuper, 2018).

Segundo Kneuper (2018), na maioria dos casos, o desenvolvimento de *software* e sistemas é realizado num contexto onde é necessário um acordo inicial sobre os parâmetros básicos do projeto, tais como os requisitos de alto nível do sistema, custos e o calendário das atividades. Porém, as equipas de desenvolvimento tendem a seguir a metodologia *Scrum*, de forma a endereçarem ao longo do tempo os riscos envolvidos.



Figura 2.5: Modelo *Water-Scrum-Fall* (adaptado de West (2011))

2.5 Metodologias Híbridas

A metodologia *Water-Scrum-fall* é uma metodologia híbrida constituída por três fases, onde a fase inicial tem como base a metodologia *Waterfall*, alternando de seguida para a utilização da metodologia *Scrum* e voltando novamente a utilizar a metodologia *Waterfall* (Bannink 2014; Dalcher 2015; Jinzenji et al. 2013; West 2011) (Figura 2.5). Na primeira fase, designada por *Water*, são tomadas as decisões estratégicas básicas, formulando a visão holística do projeto de desenvolvimento suportada na definição de requisitos essenciais e da respetiva estruturação do trabalho, da janela temporal e dos recursos necessários (Schlauderer, Overhage, e Fehrenbach, 2015). Na fase seguinte, as equipas utilizam o *Scrum* para suportar o desenvolvimento de *software* e do sistema, através da realização de vários *sprints*, até o nível de implementação da solução ser aceitável (West, 2011). Na fase final, volta a utilizar-se como modelo de referência o *Waterfall*, existindo formalmente uma entrega que será testada e avaliada, tanto ao nível dos requisitos funcionais, como também ao nível dos requisitos não funcionais constituindo a aceitação do sistema desenvolvido como um todo, concluindo o processo (Schlauderer et al., 2015).

2.5.3 *Disciplined Agile*

O processo *Disciplined Agile* é uma metodologia híbrida orientada para a aprendizagem das equipas e para a entrega de soluções tecnológicas assente num ciclo de vida com metas predefinidas e numa consciência organizacional holística (Ambler e Lines, 2011). Segundo Adelakun, Garcia, Tabaka, e Ismail (2017), esta abordagem é única no que concerne à integração de elementos da modelo *Waterfall* com o modelo ágil. O ciclo de vida do sistema que se pretende desenvolver vai desde o conceito inicial do sistema, passando pela entrega, até às operações e o suporte (Ambler e Lines, 2016). Ambler e Lines (2011) descrevem o ciclo de vida do modelo *Disciplined Agile* como uma extensão do processo de construção do modelo *Scrum*, criando, desta forma, um ciclo de vida de entrega completo desde o início do projeto até ao lançamento do sistema em produção. Nesta metodologia existem cinco fases explícitas no decorrer do processo: conceptualização, iniciação, construção, transição e produção (Figura 2.6).

2. REVISÃO DA LITERATURA

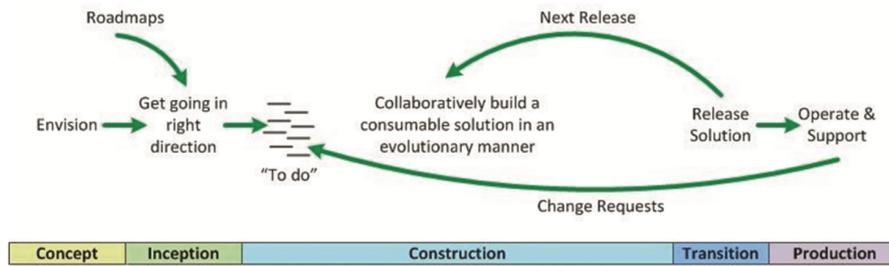


Figura 2.6: Modelo *Disciplined Agile* (Disciplined Agile, 2013)

Na fase de conceptualização existem aspectos contextuais do projeto que são definidos como: a composição do portfólio de sistemas a desenvolver, identificação dos projetos, priorização e financiamento para que seja possível passar para a fase de iniciação. A fase de iniciação do *Discipline Agile* permite a definição da visão que vai enquadrar o projeto corretamente. Segue-se a fase de construção, onde a equipa desenvolve uma solução de forma incremental. Este desenvolvimento poderá ser realizado baseado num conjunto de *sprints* ou através de uma abordagem de entrega contínua. A fase de transição tem como objetivo assegurar a implementação da solução desenvolvida nas partes interessadas. Após a fase de transição, uma solução é operada e suportada em produção (Disciplined Agile, 2013).

O *Disciplined Agile* não é restritivo e tem como objetivo refletir a realidade da melhor maneira possível e, por isso, suporta quatro versões do seu ciclo de vida. As equipas do *Disciplined Agile* deverão adotar a versão mais apropriada para o seu contexto de projeto (Ambler e Lines, 2016). Segundo Adelakun et al. (2017), esta metodologia permite flexibilidade para albergar constantes alterações de requisitos no decorrer do processo de desenvolvimento do projeto e, em simultâneo, assegurar que as técnicas, ferramentas e documentação necessárias são realizadas.

Capítulo 3

Metodologia de Investigação

Este capítulo tem como objetivo apresentar a abordagem e os métodos utilizados na realização deste trabalho. São descritos, nos seguintes sub-capítulos, as questões de investigação, a abordagem de investigação conduzida, os critérios utilizados, assim como o processo de recolha de dados e respetivo tratamento.

3.1 Desenho da Investigação

Este trabalho enquadra-se num estudo exploratório, com o objetivo de investigar a utilização de metodologias híbridas no desenvolvimento de sistemas embebidos em Portugal. Para o conseguir, procedeu-se à utilização de uma estratégia de investigação qualitativa com recolha de dados mista. A principal vantagem de utilizar fontes diferentes de dados é a possibilidade de criar linhas convergentes de investigação através do método de triangulação dos dados (Yin, 2014). Segundo Saunders, Lewis, e Thornhill (2007), os resultados obtidos são condicionados pelas respetivas técnicas de recolha e tratamento de dados utilizadas, pelo que, a utilização de métodos mistos de recolha permite um maior nível de confiança nas conclusões alcançadas.

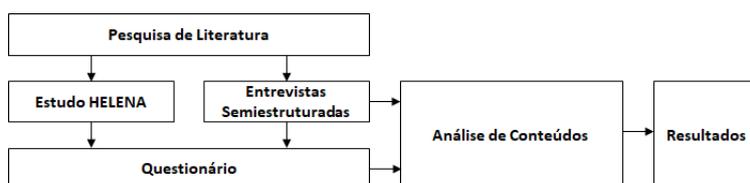


Figura 3.1: Visão geral da abordagem de investigação

3. METODOLOGIA DE INVESTIGAÇÃO

O processo de estudo utilizado neste trabalho, representado na Figura 3.1, apresenta uma estrutura constituída por três fases, procurando desta forma implementar os quatro princípios da recolha de dados de (Yin, 2014). O principal objetivo desta estratégia é reforçar a validade dos resultados obtidos e assegurar a fiabilidade da evidência recolhida. Numa primeira fase, realizou-se uma pesquisa de literatura de forma a providenciar uma extensa compreensão da teoria e do estado de arte. Este entendimento permitiu sustentar as fases seguintes do trabalho. Seguiu-se a recolha de dados através de entrevistas semi-estruturadas e questionário, e sua respetiva análise para culminar nos resultados apresentados.

3.2 Entrevistas semi-estruturadas

As entrevistas semi-estruturadas são habitualmente conhecidas como entrevistas de investigação qualitativa (King, 2004). Segundo Saunders et al. (2007), os dados recolhidos através desta tipologia de entrevista não têm como objetivo apenas revelar e compreender o quê e o como, mas também colocar maior foco em explorar o porquê. D. R. Cooper e Schindler (2008) defendem que nos estudos exploratórios ou que contenham elementos de natureza exploratória, é bastante provável a inclusão de entrevistas não estruturadas (qualitativas) na metodologia de investigação.

Considerando a natureza das questões de investigação, entende-se que a realização de entrevistas semi-estruturadas seja apropriada para entender em profundidade o contexto em estudo. Segundo as recomendações de Yin (2014), foi desenvolvido um protocolo de entrevista que pode ser consultado na Tabela I disponível no Anexo A, constituído por três secções: visão geral do caso de estudo, procedimentos de recolha de dados e estrutura da entrevista. Desta forma, procuraram-se profissionais envolvidos no desenvolvimento de sistemas embebidos, gestores de projetos, ou gestores de equipas, cuja experiência lhes permitisse um maior entendimento sobre a problemática (mais de três anos de experiência no cargo). Através do conhecimento pessoal de vários profissionais enquadrados nas várias indústrias e da sua rede de contactos, foi possível recolher os contactos telefónicos de vários profissionais. Subsequentemente, foram

3.3 Questionário

contactados telefonicamente os profissionais de diferentes empresas que satisfizessem o perfil definido. Neste contacto foi explicado o âmbito do trabalho, assim como as questões de investigação que se pretendiam endereçar com as entrevistas.

Dos profissionais contactados, quatro aceitaram participar no estudo. Estas entrevistas realizaram-se no local de trabalho dos entrevistados, numa sala isolada acusticamente. Segundo North (1983), citado em Healey (1991), os gestores tendencialmente apresentam maior disponibilidade em ser entrevistados do que em completar um questionário, especialmente quando o tema da entrevista é do seu interesse e relevante para o seu trabalho atual. A entrevista seguiu a estrutura apresentada na Tabela II do Anexo A, que é constituída por quatro categorias, abordadas de forma sequencial, sendo que a ordem das questões realizadas dentro de cada categoria não é rígida. A duração estimada de cada entrevista foi de 30 minutos, onde os cinco minutos iniciais foram reservados para a apresentação do tema, dos objetivos do trabalho e da aplicabilidade dos dados da entrevista. No final da introdução foi pedido o consentimento ao entrevistado para a gravação áudio da entrevista. As entrevistas foram posteriormente transcritas e procedeu-se à análise de conteúdos das mesmas, recorrendo aos princípios definidos por Yin (2014).

3.3 Questionário

O questionário foi desenvolvido tendo por base a pesquisa de literatura e as entrevistas semi-estruturadas realizadas. Utilizou-se como base a versão mais recente do questionário do projeto internacional *Hybrid Development Approaches in Software Systems Development* (HELENA), que procurou investigar o estado atual das práticas de desenvolvimento de sistemas e *software*. Em particular, este questionário permite determinar que abordagens são utilizadas, de que forma são combinadas, qual tem sido a evolução dessas combinações ao longo do tempo e se/como as normas externas afetam o processo de desenvolvimento e respetivas metodologias. O questionário do projeto HELENA foi utilizado em estudos semelhantes por Marinho, Richardson, Beecham, e Noll (2019), Theocharis et al. (2015) e Kuhrmann et al. (2019).

3. METODOLOGIA DE INVESTIGAÇÃO

O questionário é constituído por 38 questões, divididas em quatro partes: utilização de metodologias e processos (7 questões), utilização de metodologias e *standards* (3 questões), experiência profissional (3 questões) e informação geral (10 questões). A estrutura detalhada do questionário está disponível na Tabela III do Anexo A. A ferramenta utilizada para o tratamento e respetiva análise estatística foi o IBM SPSS® (versão 22), onde se aplicaram as metodologias adequadas ao tipo de dados recolhidos no questionário, constituído na sua maioria, por variáveis categóricas.

O questionário foi desenvolvido utilizando a aplicação Survey Monkey® e disponibilizado para pré-teste a três pessoas com experiência no *design* e avaliação de questionários com objetivo de aferir a qualidade e procurar pontos de melhoria. Seguidamente, foram discutidas e incorporadas as recomendações e sugestões recolhidas. Na fase seguinte, foi publicado *online* e distribuído, no período compreendido entre dia 2 de setembro de 2019 e o dia 25 de setembro de 2019. A sua promoção foi feita através de contatos pessoais e das redes sociais, nomeadamente Facebook® e LinkedIn®. A população alvo é caracterizada por profissionais que participam ou participaram, no último ano, em projetos de desenvolvimento de sistemas embebidos ou sistemas tecnológicos com um grau de personalização elevado em Portugal, sendo este perfil o mais adequado para obter informações relevantes e devidamente enquadradas que permitam responder às questões de investigação.

Findo este processo, os dados foram tratados e repetivamente analisados. Dada a natureza categórica das variáveis do questionário recorreu-se à construção de tabelas de contingência (*contingency table* ou *crosstab*). As tabelas terão tantas linhas e colunas quantas as categorias em cada uma das variáveis em análise. A construção de uma tabela de contingência permite o cálculo de uma estatística, a partir da qual, é possível realizar um teste de hipóteses designado por Qui-quadrado, para aferir a independência das variáveis. Este teste aplica-se sempre que se pretende verificar a dependência entre duas variáveis do tipo categórico (Everitt, 1992). Rejeita-se a hipótese de independência (hipótese nula) entre as variáveis quando o *p-value* é superior a um certo valor significância e cujos os valores habituais utilizados são 1%, 5% e 10%. Neste trabalho

3.3 Questionário

definiu-se como referência um nível de significância de 5%.

H_0 : as variáveis são independentes *versus* H_1 : as variáveis não são independentes.

Neste trabalho, garantiu-se o cumprimento dos requisitos mínimos do teste do Qui-quadrado, nomeadamente que as frequências esperadas em cada classe não devem ser inferiores a cinco, sempre que o número total de observações é inferior a 20. Nos casos em que o número total de observações é superior a 20, não deverá existir mais do que 20% das células da tabela de contingência com frequências esperadas inferiores a 5, nem deverá existir nenhuma célula com frequência esperada inferior a 1 (Everitt, 1992). Devido à natureza discreta da contagem das frequências, o *p-value* do teste do Qui-quadrado vem acrescido de um erro. No caso de tabelas 2x2 utilizou-se uma correção da continuidade, sendo este o valor de referência nas tabelas com estas características (Everitt, 1992).

Os dez pontos de metadados existentes no questionário permitiram uma caracterização da amostra e uma análise contextual das respostas obtidas. Com o objetivo de identificar que práticas estão associadas às metodologias, procedeu-se à construção de tabelas de contingência e respetiva aplicação do teste do Qui-quadrado para todas as combinações existentes na amostra. Os resultados da utilização das metodologias foram agregadas em três categorias/abordagens: ágil, tradicional e genérica, com base nas definições fornecidas por Kuhrmann et al. (2017). As metodologias organizadas por categoria apresentam-se na Tabela IV do Anexo A. Com base no mapeamento da utilização das metodologias, produziu-se uma análise das combinações híbridas ao nível das metodologias existentes na amostra, segundo o conceito de Kuhrmann et al. (2017).

Capítulo 4

Análise e Discussão de Resultados

Este capítulo tem como objetivo a apresentação e a discussão dos resultados obtidos através dos dois instrumentos de recolha de dados utilizados neste trabalho. Desta forma, começa-se por apresentar os resultados obtidos nas entrevistas semi-estruturadas e de seguida os resultados obtidos no questionário.

4.1 Análise das Entrevistas

No sentido de averiguar o interesse em participar no estudo, foram contactados 15 profissionais da área. Os seus contactos foram obtidos utilizando a rede de contactos profissionais que já possuía, assim como através da realização de uma pesquisa pelo na rede social LinkedIn[®]. A escolha dos profissionais foi alvo de uma cuidada análise para garantir a adequabilidade do perfil face aos critérios definidos. Dos contactos efetuados, seis declinaram o convite, em participar no estudo. Cinco não mostraram disponibilidade para agendar a entrevista dentro de um prazo razoável à execução do trabalho. As quatro entrevistas realizaram-se entre 1 de julho de 2019 e 20 de julho de 2019, na zona metropolitana de Lisboa. Três dos entrevistados eram provenientes da mesma empresa, enquanto que um era de uma empresa distinta. As entrevistas foram transcritas integralmente conforme recomendado por Yin (2014), com vista ao preenchimento do quadro de resultados definido no protocolo da entrevista (Tabela V do Anexo B).

Da análise dos resultados obtidos foi possível entender que relativamente à forma como as equipas estão distribuídas, apenas um dos entrevistados indica a co-localização

4. ANÁLISE E DISCUSSÃO DE RESULTADOS

da equipa de projeto, sendo que todos os restantes trabalham com os recursos distribuídos geograficamente pelo menos no mesmo continente. A forma de alocação dos membros da equipa de projeto, na generalidade, é partilhada com outros projetos, sendo que apenas um dos entrevistados frisou a possibilidade da equipa estar dedicada a um único projeto, na condição desse apresentar uma complexidade elevada. A dimensão temporal dos projetos foi referida de forma transversal pelos entrevistados como sendo de um a dois anos de duração.

Na categoria de processos e metodologias utilizadas, todos os entrevistados indicaram a utilização do processo padrão que é transversal à empresa. De entre os quais foi possível destacar a utilização das metodologias *waterfall*, *Stage-Gate* e num dos casos o *Scrum* era combinado com os dois anteriormente referidos. Importa salientar que três dos entrevistados referiram-se ao processo da sua empresa como um processo próprio da organização. No entanto, através das suas descrições foi possível identificá-los como sendo derivações do modelo *Stage-Gate*. A maioria dos entrevistados referiu que possui liberdade condicionada para se desviar do processo padrão da organização.

Sobre a utilização de normas externas, todos os entrevistados referiram a sua utilização na metodologia de desenvolvimento (IATF, ISO e GSS), sendo que apenas um não identifica incompatibilidades da utilização de metodologias ágeis com normas externas. No caso dos entrevistados que identificam incompatibilidades, estes referem que “as estruturas organizacionais estão construídas com base em metodologias tradicionais” e que “existe uma diretiva de entrega de *software* após validação total do produto, o que não é totalmente compatível com os ciclos de desenvolvimento e de entrega das metodologias ágeis. As próprias normas externas não foram desenhadas a ter em conta a existência de uma abordagem ágil.”

4.2 Questionário

Através da promoção do questionário realizada com base na rede de contactos profissionais e das redes sociais, nomeadamente Facebook[®] e LinkedIn[®], o questionário

4.2 Questionário

foi enviado a 132 profissionais com experiência profissional no desenvolvimento de sistemas embebidos, dos quais 75 forneceram respostas. Consideram-se como critérios de inclusão todas as respostas recolhidas no período de tempo compreendido entre 2 de setembro e 25 de setembro de 2019 e referidas a projetos de desenvolvimento de sistemas embebidos realizados em Portugal. Apenas se consideraram 68 respostas válidas. Importa salientar que neste trabalho não se aplicou qualquer critério de exclusão relativamente à faixa etária, género, habilitações literárias ou tempo de experiência profissional no desenvolvimento de sistemas embebidos.

A maioria dos respondentes (67,6%) estiveram envolvidos em projetos que decorreram em organizações cuja dimensão se caracteriza por “grande” (251-2499 colaboradores) ou, “muito grande” (mais de 2500 colaboradores) (Tabela VI do Anexo C). Relativamente à dimensão dos projetos, tendo como referência o número de pessoas e o tempo alocado, 61,7% das respostas enquadram-se nas categorias “grande” (6 pessoas mês - 1 pessoa ano) ou “muito grande” (mais de 1 pessoa ano) (Tabela VII do Anexo C). No que concerne aos anos de experiência profissional no desenvolvimento de sistemas embebidos, 63,2% dos respondentes apresenta uma experiência até cinco anos, inclusive (Tabela VIII do Anexo C). A faixa etária dos 20 aos 29 anos é a mais representativa da amostra com 42,6% (Tabela IX do Anexo C). Cerca de 26% dos inquiridos eram engenheiros de desenvolvimento ou *developers*, 25% eram gestores de projeto ou equipa, 15% eram engenheiros de sistemas ou analistas de requisitos, 12% eram engenheiros de teste ou *testers*, 3% eram arquitetos, 3% eram *scrum masters* ou gestores de equipa, gestores de topo e, por fim, 16% ocupam outros cargos não discriminados no questionário. Caracterizando a amostra relativamente ao género, 77,9% dos inquiridos eram do sexo masculino e 22,1% do sexo feminino.

4.2.1 Metodologias e Práticas Utilizadas

Os resultados que se seguem permitem uma melhor compreensão das variáveis face às respostas obtidas no questionário. Com o objetivo de alcançar as respostas à primeira pergunta de investigação (Q1), realizou-se uma análise de frequências, de forma a ser possível fornecer tabelas e gráficos que permitam mapear a utilização de metodologias e

4. ANÁLISE E DISCUSSÃO DE RESULTADOS

práticas.

As metodologias mais utilizadas (total ou parcialmente) pelos inquiridos são o *Scrum* (56,1%), o *Waterfall* (54,6%), o *Kanban* (36,4%) e o Modelo *DevOps* (25,8%), evidenciado na Figura 4.1. A Tabela XV que consta do Anexo C apresenta a totalidade dos resultados referentes à utilização das metodologias. Estes resultados vão na mesma linha dos resultados obtidos por Theocharis et al. (2015) que revelou no seu artigo que as metodologias mais utilizadas são o *Scrum*, *Waterfall* e o *Kanban*. Também Marinho et al. (2019) apresentaram o *Scrum*, o *DevOps*, o *Kanban* e *Waterfall* como sendo as metodologias mais utilizadas na prática.

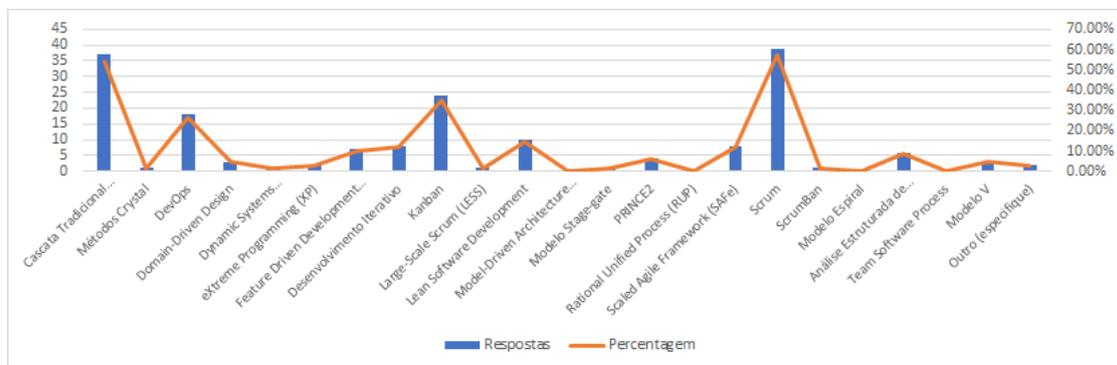


Figura 4.1: Utilização de metodologias

As práticas que se revelaram mais utilizadas pelos respondentes são a gestão de *backlog* (54,4%), revisão de código (42,7%), implementação contínua (39,7%), integração contínua (39,7%) e especificações arquiteturais (38,2%). A Tabela XVI (Anexo C) apresenta a totalidade dos resultados referentes à utilização das práticas. Estes resultados não diferem muito aos resultados obtidos por Kuhrmann et al. (2018). A exceção é a prática de especificações arquiteturais, que se trata de uma atividade muito característica do desenvolvimento de sistemas embebidos e que desempenha um papel fundamental na gestão das limitações técnicas próprias deste tipo de sistemas.

Da análise da associação entre as práticas de desenvolvimento e as quatro metodologias mais utilizadas pelos inquiridos, não foi possível estabelecer uma associação estatística à metodologia *Waterfall* (nível de significância de 5%). No caso da metodologia *Scrum* as práticas que se destacaram pela existência de uma associação estatística são

4.2 Questionário

user stories (como prática de engenharia de requisitos), reuniões *standup* diárias, gestão de *backlog*, teste de sistema integral (*end-to-end system testing*), integração contínua, estimativa baseada no especialista/equipa, retrospectivas, planeamento de entregas e *release planning*. Comprovou-se a existência de uma associação estatística entre a metodologia *Kanban* e as práticas teste de sistema integral, *user stories* e reuniões *standup* diárias. Finalmente, para a metodologia *DevOps* identificou-se uma associação estatística com as práticas de integração contínua e os testes unitários automatizados.

As práticas associadas ao *Scrum*, encontram-se, na sua maioria, mencionadas no *Scrum Guide* (Schwaber e Sutherland, 2017), com exceção dos testes de sistema integral. A prática de integração contínua, realizada a jusante do processo de desenvolvimento, é aqui associada ao *Scrum* e resulta como resposta das entregas frequentes que o *Scrum* potencia. Apesar da raiz conceptual da metodologia *Kanban* surgir da metodologia *Lean* (Ahmad, Dennehy, Conboy, e Oivo, 2018), duas das práticas que lhe foram associadas têm origem na metodologia *Scrum*, revelando desta forma uma tendência para a combinação das práticas destas duas metodologias. A prática de testes de sistema integral, associada nos resultados às metodologias *Scrum* e *Kanban*, trata-se de uma prática genérica no desenvolvimento de sistemas embebidos e cuja múltipla associação resulta da natureza dos sistemas embebidos e das características próprias deste tipo de projetos. As práticas associadas estatisticamente ao *DevOps* enquadram-se no conceito e no processo da própria metodologia. Segundo Ebert, Gallardo, Hernantes, e Serrano (2016), entregas de qualidade com ciclos temporais curtos requerem um alto nível de automatização dos processos, sendo os testes unitários um desses processos. Além disso, a entrega sistemática e bastante frequente implica uma prática de integração contínua que está inerente à natureza e ao objetivo do próprio processo.

Com base nos resultados obtidos após a agregação das metodologias por categoria/abordagem, 80,9% dos entrevistados reportaram utilizar (total ou parcialmente) uma abordagem ágil, 64,7% uma abordagem tradicional e 16,2% uma abordagem genérica. Estes valores corroboram a predominância de utilização das metodologias da categoria ágil evidenciada por Tell et al. (2019).

4. ANÁLISE E DISCUSSÃO DE RESULTADOS

A utilização de tabelas de contingência entre as metodologias utilizadas e os domínios de aplicação/indústria das respostas obtidas tornou possível identificar associações estatísticas (nível de significância de 0,05) dos quais se destacam: uma associação positiva entre o *Scrum* e a indústria das telecomunicações; uma associação negativa entre a indústria automóvel e as metodologias *Scrum*, *Kanban* e *Waterfall*. A associação negativa da indústria automóvel com as metodologias enumeradas deve-se à relação histórica ligada ao modelo *Stage-Gate* e à sua personalização e evolução no seio de cada organização, originando muitas vezes processos organizacionais próprios, conforme foi identificado nas entrevistas realizadas neste trabalho.

4.2.2 Combinações de Metodologias e Abordagens

Os resultados que se seguem têm como objetivo alcançar as respostas à segunda questão de investigação (Q2). Para tal, realizou-se uma análise de frequências dos processos de definição e decisão das metodologias utilizadas, assim como das combinações de metodologias ágeis e tradicionais existentes na amostra. Esta análise permitiu fornecer tabelas, gráficos e, por conseguinte, realizar testes de hipóteses que evidenciem associação estatística entre metodologias que sejam utilizadas como base para a conceção de metodologias híbridas.

Sobre a forma como as metodologias, práticas e combinações aplicadas na organização foram desenvolvidas, 73,5% dos inquiridos indicam que evoluíram da aprendizagem de projetos desenvolvidos ao longo do tempo e 41,2% relatam que foram planeados como parte de um programa de melhoria de processos (Tabela XII do Anexo C). Estes resultados corroboram as evidências recolhidas nas entrevistas semi-estruturadas. No que toca à forma como o processo de gestão do projeto era definido, 42,6% dos respondentes indicam que as práticas e os métodos são selecionados conforme a necessidade e 27,9% que as práticas e métodos específicos são selecionados de acordo com os requisitos do clientes (Tabela XIV do Anexo C). Relativamente aos objetivos gerais que se pretendem alcançar com as abordagens, métodos e práticas definidas, destacam-se o melhor planeamento e estimativa do tempo e dos custos (66,2%), melhor

4.2 Questionário

qualidade do produto (52,9%) e melhores níveis de produtividade (51,5%) (Tabela XIII do Anexo C).

Referente à segunda questão de investigação deste trabalho, procurou-se aferir que pares de combinações de metodologias são utilizadas como base para a constituição de metodologias híbridas. Através da construção de uma tabela de contingência constituída por metodologias tradicionais e ágeis, evidenciaram-se três pares de combinações dos resultados obtidos: *Waterfall-Scrum* (41,2%), *Waterfall-Kanban* (17,6%) e *Waterfall-DevOps* (14,7%) (Figura 4.2).

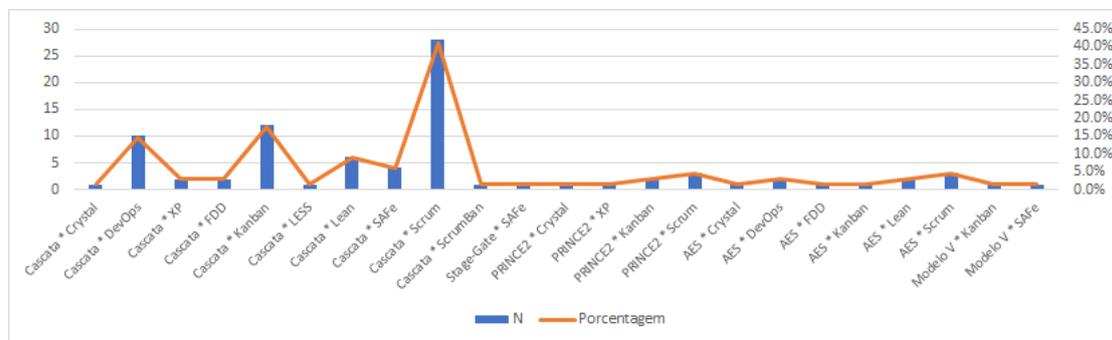


Figura 4.2: Combinações de metodologias

Destes três casos, apenas a combinação *Waterfall-Scrum*, revela a existência de uma associação estatística entre as duas variáveis para um nível de significância de 0,05 (Tabela XVIII do Anexo C). A associação foi analisada através dos coeficientes de *Phi*, *V Cramer* e de Contingência, onde todos revelaram uma associação significativa (valores superiores a 0,32 (Statistics Solutions, 2019)) como é possível verificar na Tabela XIX do Anexo C. Este resultado permite evidenciar que estas duas metodologias são a base para a conceção de metodologias híbridas no desenvolvimento de sistemas embebidos em Portugal. Similarmente, Theocharis et al. (2015) revelaram que a combinação das metodologias *Waterfall* e *Scrum* era o padrão nos Estados Unidos da América, confirmando a hipótese lançada por West (2011) de que a metodologia híbrida *Water-Scrum-Fall* é a realidade para a maioria das organizações.

4. ANÁLISE E DISCUSSÃO DE RESULTADOS

Tabela I: Tabela de contingência da utilização de metodologias *Scrum* e *Waterfall*

		<i>Scrum</i>		Total
		Não	Sim	
<i>Waterfall</i>	Não	20	11	31
	Sim	9	28	37
Total		29	39	68

De forma a realizar uma análise das abordagens de desenvolvimento do conjunto de resultados obtidos, agregaram-se as metodologias do questionário pelas categorias ágil, tradicional ou genérica (Marinho et al., 2019). Nos resultados obtidos após a agregação, destaca-se que 80,9% dos respondentes afirmaram utilizar, pelo menos, uma abordagem ágil, 64,5% uma abordagem tradicional e apenas 16,2% uma metodologia genérica (Tabela XX do Anexo C).

Realizou-se, por fim, a análise de associação estatística entre as categorias/abordagens ágil e tradicional. Após a construção da tabela de contingência não foi possível realizar o teste de independência do Qui-quadrado face ao não cumprimento dos critérios de validação (Tabela XXII do Anexo C).

Capítulo 5

Conclusões, Limitações e Trabalho Futuro

No contexto da utilização de metodologias e práticas de desenvolvimento, os resultados alcançados sugerem que, de forma geral, os projetos de desenvolvimento de sistemas embebidos, em Portugal, utilizam combinações de metodologias ágeis e tradicionais, ou seja, metodologias híbridas. As metodologias mais utilizadas e, simultaneamente, as que com maior frequência são combinadas entre si são o *Scrum* e o *Waterfall*. Tendo-se verificado uma associação estatística relevante entre estas, corrobora-se a teoria de West (2011) de que o *Water-Scrum-Fall* seria uma realidade para a maioria das organizações. Além disso, os resultados permitem verificar que, tanto o *Scrum*, como o *Waterfall* formam a base para a conceção de metodologias híbridas em Portugal.

Este trabalho revela que a maioria das metodologias de desenvolvimento de sistemas embebidos surgem de um processo de evolução. Cerca de 73,9% dos inquiridos afirmou que a abordagem de desenvolvimento emergiu de uma aprendizagem dos projetos desenvolvidos ao passado. Consequentemente, as organizações que utilizam uma combinação de metodologias ágeis e tradicionais procuram otimizar o *trade-off* entre potenciar os benefícios de uma abordagem iterativa, flexível e colaborativa com a manutenção de um certo nível de planeamento e estrutura.

Em suma, este trabalho corrobora as conclusões de Aitken e Ilango (2013), de que não existe uma incompatibilidade com a aplicação de todos os princípios e valores da

5. CONCLUSÕES, LIMITAÇÕES E TRABALHO FUTURO

abordagem ágil com a maioria das práticas tradicionais do desenvolvimento de sistemas e *software*. Adicionalmente confirma a teoria de West (2011) e valida os resultados de Theocharis et al. (2015) de que as metodologias híbridas, em particular a metodologia *Water-Scrum-Fall* são a nova realidade da maior parte das organizações que desenvolvem sistemas embebidos.

No que concerne às limitações do trabalho, sublinham-se duas como as mais relevantes. A primeira limitação prende-se com o facto do trabalho estar circunscrito do ponto de vista geográfico aos projetos de desenvolvimento de sistemas embebidos em Portugal. Isto vem limitar à partida o tamanho da amostra, ao considerar como válidas apenas as respostas que cumprissem este critério de inclusão. A segunda limitação é que a pequena dimensão amostral, uma vez que se considera a população que desenvolve sistemas embebidos em Portugal. Um maior número de inquiridos com respostas válidas, de acordo com os critérios de inclusão definidos, providenciariam maior robustez e precisão nos cálculos e nas análises estatísticas realizadas. Em particular, os dados disponíveis não permitiram aferir a existência de uma associação estatística entre as categorias/abordagens ágeis e tradicionais de uma forma mais transversal.

Face aos resultados e conclusões alcançados neste trabalho, a realidade das metodologias híbridas no desenvolvimento de sistemas embebidos deverá ser considerada como uma nova linha base para pesquisas e estudos futuros, nomeadamente no desenvolvimento de estratégias adequadas para a definição e utilização de metodologias híbridas por parte das organizações, quando confrontadas com desafios e necessidades de integrar diferentes metodologias e práticas de desenvolvimento nos projetos desta natureza.

Referências Bibliográficas

- Adelakun, O., Garcia, R., Tabaka, T., e Ismail, R. (2017). Hybrid Project Management: Agile with Discipline. In *Con-firm 2017 proceedings* (p. 13).
- Ahmad, M. O., Dennehy, D., Conboy, K., e Oivo, M. (2018). Kanban in software engineering: A systematic mapping study. *Journal of Systems and Software*, 137, 96–113.
- Aitken, A., e Ilango, V. (2013). A comparative analysis of traditional software engineering and agile software development. In *Proceedings of the annual hawaii international conference on system sciences* (pp. 4751–4760).
- Ambler, S. W., e Lines, M. (2011). *Disciplined Agile Delivery: An introduction (white paper)*, pg 11.
- Ambler, S. W., e Lines, M. (2016). The Disciplined Agile Process Decision Framework , SWQD 2016 Vienna, Austria, January 18–21, 2016 Proceedings. In *Software quality: The future of systems- and software development: 8th international conference* (Vol. 238, pp. 3–14). Viena: Springer, Cham.
- Association for Project Management. (2012). *APM Body of Knowledge* (6th ed.).
- Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., e Highsmith, J. (2001). *Manifesto for Agile Software Development*. Retrieved from <https://agilemanifesto.org/>
- Boehm, B., e Turner, R. (2003). Observations on balancing discipline and agility. In *Proceedings of the agile development conference, adc 2003* (pp. 32–39).
- Bohmer, A. I., Meinzinger, M., Hostettler, R., Knoll, A., Lindemann, U., Böhmer, A. I., ... Lindemann, U. (2018). *Towards a framework for agile development of physical products* (Vol. 2018-Jan).
- Clark, J. O. (2009). System of systems engineering and family of systems engineering from a standards, v-model, and dual-v model perspective. In *2009 iee international systems conference proceedings* (pp. 381–387).
- Cockburn, A., e Highsmith, J. (2001). Agile Software Development: The People Factor. *COMPUTER*, 131–133.
- Cohen, M. (2010). *Succeeding With Agile: Software Development Using Scrum*, by

Referências Bibliográficas

- Mike Cohn. In *The journal of object technology* (7th ed., Vol. 9). Addison-Wesley.
- Conforto, E. C., e Amaral, D. C. (2015). Agile project management and stage-gate model—A hybrid framework for technology-based companies. *Journal of Engineering and Technology Management - JET-M*, 40(2015), 1–14.
- Cooper, D. R., e Schindler, P. S. (2008). *Business research methods* (10th ed.). McGraw-Hill Higher Education.
- Cooper, R. (2016, jan). Agile–Stage-Gate Hybrids. *Research-Technology Management*, 59(1), 21–29.
- Cooper, R., e Sommer, A. (2016, sep). The Agile-Stage-Gate Hybrid Model: A Promising New Approach and a New Research Opportunity. *Journal of Product Innovation Management*, 33(5), 513–526.
- Dackhammar, N., e Ek, J. (2017, sep). Combining Agile methods and a Stage-Gate model within the product development process; A multiple case study.
- Disciplined Agile. (2013). *Full Delivery Lifecycles*. Retrieved 2019-07-29, from <http://disciplinedagiledelivery.com/lifecycle/>
- Ebert, C., Gallardo, G., Hernantes, J., e Serrano, N. (2016). DevOps. *IEEE Software*, 33(3), 94–100.
- Everitt, B. (1992). *The Analysis of Contingency Tables* (2nd ed.). London: CHAPMAN & HALL/CRC.
- Executive Brief. (2008). *Which Life Cycle is best for your project?* Retrieved 2019-10-12, from <https://www.projectsmart.co.uk/which-life-cycle-is-best-for-your-project.php>
- Forrai, A. (2013). Embedded Systems - Basic Concepts. In *Embedded control system design* (pp. 1–19). Heidelberg: Springer.
- Forsberg, K., e Mooz, H. (1991). The Relationship of System Engineering to the Project Cycle. *INCOSE International Symposium*, 1, 57–65.
- Freitas, T. (2015). Agile Practices in Embedded Systems Development.
- Garousi, V., Coskunçay, A., Betin-Can, A., e Demirörs, O. (2015). A survey of software engineering practices in Turkey. *Journal of Systems and Software*, 108, 148–177.
- Giusto, D., Iera, A., Morabito, G., e Atzori, L. (2010). *The Internet of Things, 20th Tyrrhenian Workshop on Digital Communications*. New York, NY: Springer New York.
- Graaf, B., Lormans, M., e Toetenel, H. (2003, nov). Embedded software engineering: The state of the practice. *IEEE Software*, 20(6), 61–69.
- Greene, B. (2004). Agile methods applied to embedded firmware development. In *Proceedings of the agile development conference, adc 2004* (pp. 71–77).
- Hedman, J., e Lind, M. (2009). Is There Only One Systems Development Life Cycle?

Referências Bibliográficas

- In W. W. hris Barry, Kieran Conboy, Michael Lang, W. Gregory Wojtkowski (Ed.), *Information systems deveelopment: Challenges in practice, theory and education* (pp. 105–116). Springer Science+Business Media.
- Hendler, S. (2018, sep). Digital-physical product development: a qualitative analysis. *European Journal of Innovation Management*, EJIM–01–2018–0026.
- Isaias, P., e Issa, T. (2015). High level models and methodologies for information systems. *High Level Models and Methodologies for Information Systems*, 1–145.
- Jones, C. (2003). Variations in Software Development Practices. *IEEE Software*, 20(6), 22–27.
- Kaisti, M., Rantala, V., Mujunen, T., Hyrynsalmi, S., Könnölä, K., Mäkilä, T., e Lehtonen, T. (2013, dec). Agile methods for embedded systems development - a literature review and a mapping study. *EURASIP Journal on Embedded Systems*, 2013(1), 15.
- Karlström, D., e Runeson, P. (2006). Integrating agile software development into stage-gate managed product development. *Empirical Software Engineering*, 11(2), 203–225.
- King, N. (2004). Using interviews in qualitative research. In *Essential guide to qualitative methods in organizational research* (pp. 11–22). Sage.
- Kneuper, R. (2018). *Software Processes and Life Cycle Models: An Introduction to Modelling, Using and Managing Agile, Plan-Driven and Hybrid Processes*.
- Könnölä, K., Suomi, S., Mäkilä, T., Jokela, T., Rantala, V., e Lehtonen, T. (2016, aug). Agile methods in embedded system development: Multiple-case study of three industrial cases. *Journal of Systems and Software*, 118, 134–150.
- Kuhrmann, M., Diebold, P., Munch, J., Tell, P., Trektere, K., McCaffery, F., ... Prause, C. R. (2019). Hybrid Software Development Approaches in Practice: A European Perspective. *IEEE Software*, 36(4), 20–31.
- Kuhrmann, M., e Fernandez, D. M. (2015). Systematic Software Development: A State of the Practice Report from Germany. *Proceedings - 2015 IEEE 10th International Conference on Global Software Engineering, ICGSE 2015*, 51–60.
- Kuhrmann, M., Hanser, E., Prause, C. R., Diebold, P., Münch, J., Tell, P., ... Linssen, O. (2017). Hybrid software and system development in practice: waterfall, scrum, and beyond. In *Proceedings of 2017 international conference on software and systems process* (pp. 30–39).
- Kuhrmann, M., Tell, P., Klünder, J., Hebig, R., Licorish, S., e MacDonell, S. (2018). *HELENA Stage 2 Results*.
- Leau, Y., Loo, W., Tham, W., e Tan, S. (2012). Software Development Life Cycle AGILE vs Traditional Approaches. In *International conference on information and network*

Referências Bibliográficas

- technology (icint 2012)* (Vol. 37, pp. 162–167).
- Lee, E. A. (2012). Introducing Embedded Systems : A Cyber- Physical Systems Approach Background : Five Years of Experience with “ Introduction to Embedded Systems ”. In (p. 27).
- Lwakatare, L. E., Karvonen, T., Sauvola, T., Kuvaja, P., Olsson, H. H., Bosch, J., e Oivo, M. (2016, jan). Towards DevOps in the Embedded Systems Domain: Why is It So Hard? In *Towards devops in the embedded systems domain: Why is it so hard?* (pp. 5437–5446). IEEE.
- Maier, M. W. (1999). Architecting Principles for. *The Aerospace Corporation; John Wiley & Sons, Inc. Syst Eng, 1*, 267–284.
- Marinho, M., Richardson, I., Beecham, S., e Noll, J. (2019). Plan-Driven approaches are alive and kicking in agile Global Software Development. (June).
- Marwedel, P. (2018). *Embedded System Design* (3rd ed.; Springer, Ed.). Dortmund.
- Mathur, S., e Malik, S. (2010). Advancements in the V-Model. *International Journal of Computer Applications, 1*(12), 30–35.
- Paasivaara, M., Durasiewicz, S., e Lassenius, C. (2009). Using scrum in distributed agile development: a multiple case study. In *Proceedings - 2009 4th iee international conference on global software engineering, icgse 2009* (pp. 195–204).
- Papadakis, E., e Tsironis, L. (2018). Hybrid methods and practices associated with agile methods, method tailoring and delivery of projects in a non-software context. *Procedia Computer Science, 138*, 739–746.
- Petersen, K., Wohlin, C., e Baca, D. (2009). The Waterfall Model in Large-Scale. *10th International Conference, PROFES 2009, 32*(8), 386–400.
- Porter, M. E., e Heppelmann, J. E. (2014). *How Smart, Connected Products Are Transforming Competition* (Tech. Rep. No. November).
- Porter, M. E., e Heppelmann, J. E. (2015). *How smart, connected products are transforming companies* (Vol. 2015) (No. October).
- Project Management Institute. (2008). *Project Management Body of Knowledge* (4th ed.).
- Rigby, D. K., Sutherland, J., e Takeuchi, H. (2016). Embracing Agile. *Havard Business Review*(May), 41–50.
- Royce, W. W. (1970). *Managing The Development of Large Software Systems*.
- Ruparelia, N. B. (2010). Software development lifecycle models. *ACM SIGSOFT Software Engineering Notes, 35*(3), 8.
- Sabale, R., e Dani, A. (2013). Comparative Study of Prototype Model For Software Engineering With System Development Life Cycle. *IOSR Journal of Engineering, 02*(07), 21–24.

Referências Bibliográficas

- Saunders, M., Lewis, P., e Thornhill, A. (2007). *Research Method for Business Students fifth edition*. Peason Education.
- Schlauderer, S., Overhage, S., e Fehrenbach, B. (2015). Widely used but also highly valued? Acceptance factors and their perceptions in water-scrum-fall projects. *2015 International Conference on Information Systems: Exploring the Information Frontier, ICIS 2015*, 1–19.
- Schwaber, K. (2010). *Scrum.org*. Retrieved 2019-05-15, from <https://www.scrum.org/>
- Schwaber, K., e Sutherland, J. (2017). The Scrum Guide: The Definitive The Rules of the Game. *Scrum.Org and ScrumInc*(November), 19.
- Selnes, J., Visser, S., Sinha, V., e Elverum, C. (2017). Agile Frameworks for Physical Product Development. , 126.
- Sommer, A. F., Hedegaard, C., Dukovska-Popovska, I., e Steger-Jensen, K. (2015, jan). Improved Product Development Performance through Agile/Stage-Gate Hybrids: The Next-Generation Stage-Gate Process? *Research-Technology Management*, 58(1), 34–45.
- Statistics Solutions. (2019). *Nominal Variable Association - Statistics Solutions*. Retrieved 2019-10-12, from <https://www.statisticssolutions.com/nominal-variable-association/>
- Taivalsaari, A., e Mikkonen, T. (2018). On the development of IoT systems. *2018 3rd International Conference on Fog and Mobile Edge Computing, FMEC 2018*, 13–19.
- Tell, P., Klunder, J., Kupper, S., Raffo, D., MacDonell, S. G., Munch, J., ... Kuhrmann, M. (2019). What are Hybrid Development Methods Made Of? An Evidence-Based Characterization. In *International conference on software and systems process, at montreal, canada* (pp. 105–114).
- Tetlay, A., e John, P. (2009). Determining the Lines of System Maturity, System Readiness and Capability Readiness in the System Development Lifecycle. In *Annual conference on systems engineering research 2009* (pp. 1–8).
- Theocharis, G., Kuhrmann, M., Münch, J., e Diebold, P. (2015). Is water-scrum-fall reality? On the use of agile and traditional development practices. In *Lecture notes in computer science* (Vol. 9459, pp. 149–166).
- Vijayasathy, L. R., e Butler, C. W. (2016). Choice of Software Development Methodologies: Do Organizational, Project, and Team Characteristics Matter? *IEEE Software*, 33(5), 86–94.
- Vishwas, M., e Satao, K. (2012). Comparing Various SDLC Models And The New Proposed Model On The Basis Of Available Methodology. *International Journal*

Referências Bibliográficas

- of Advanced Research in Computer Science and Software Engineering*, 2(4), 170–177.
- Wang, X., Wang, Z., Huang, W., Wen, G. Q., e Zhang, S. L. (2017, apr). Summary of Embedded Systems Development. In *Proceedings of the 2017 5th international conference on frontiers of manufacturing science and measuring technology (fmsmt 2017)*. Paris, France: Atlantis Press.
- West, D. (2011). Water-Scrum-Fall Is The Reality Of Agile For Most Organizations Today. *For Application Development & Delivery Professionals*, 1–15.
- Whitten, J., e Bentley, L. (2007). *Systems Analysis and Design Methods* (7th ed.). McGraw-Hill Irwin.
- Xie, M., Shen, M., Rong, G., e Shao, D. (2012). Empirical studies of embedded software development using agile methods. *Proceedings of the 2nd international workshop on Evidential assessment of software technologies*, 2, 21.
- Yin, R. K. (2014). *Case Study Research: Design and Methods* (5th ed.). SAGE.

Anexos

Anexo A

Tabela I: Protocolo da entrevista semi-estruturada

Visão geral das entrevistas semi-estruturadas	
Objetivo	Aprofundar o conhecimento sobre as principais metodologias de desenvolvimento utilizadas nos projetos de sistemas embebidos, assim como explorar os principais desafios existentes e as principais razões que levam à utilização desses mesmos processos neste contexto.
Proposições a verificar	A implementação de metodologias ágeis no desenvolvimento de sistemas embebidos revela-se uma tarefa complexa devido à sua natureza (Freitas, 2015); Na prática, as organizações utilizam metodologias híbridas no desenvolvimento de sistemas embebidos (West, 2011);
Enquadramento teórico das entrevistas	As metodologias ágeis têm vindo a apresentar resultados positivos no desenvolvimento de <i>software</i> , melhorando a qualidade e a velocidade de desenvolvimento de produto (Rigby et al., 2016). Isto leva a que muitos gestores identifiquem a necessidade de as implementar no desenvolvimento de sistemas embebidos (R. Cooper e Sommer, 2016). No entanto, segundo Freitas (2015), a implementação de metodologias ágeis no desenvolvimento de sistemas embebidos tem vindo a revelar-se uma tarefa complexa devido à natureza que este tipo de desenvolvimento de produto encerra em si mesmo. Para colmatar estas dificuldades, alguns autores têm sugerido diferentes abordagens, nomeadamente a combinação dos diferentes processos de desenvolvimento de <i>software</i> e <i>hardware</i> (Hendler, 2018). Estas abordagens, designadas habitualmente por metodologias híbridas, procuram equilibrar os benefícios e desafios de dois modelos, criando um número significativo de vantagens (R. Cooper, 2016).
Papel do protocolo na orientação da entrevista	O protocolo é a principal linha condutora da realização das entrevistas de forma a garantir uma maior fiabilidade de todo processo de preparação, execução, recolha e análise de conteúdos.
Procedimentos de recolha de dados	
Perfil das pessoas para realizar as entrevistas	Gestores de projeto ou de equipa no desenvolvimento de sistemas embebidos que apresentam uma experiência superior a 3 anos no cargo.
Plano de recolha de dados	Investigação das principais empresas em Portugal que realizam atividades de desenvolvimento de sistemas embebidos; Identificação dos profissionais nessas empresas que possam corresponder ao perfil de entrevistas definido; Contactar os profissionais identificados, preferencialmente por contacto telefónico e caso não seja possível por email. No primeiro contacto pretende-se aferir se o perfil da pessoa é o adequado para participar no estudo, verificar o possível interesse em participar no trabalho e agendar a entrevista no dia e hora mais favorável para o entrevistado. Adicionalmente no decorrer deste contacto é explicado, de forma sucinta, o tema da entrevista, a sua aplicação e os objetivos da investigação.
Preparação prevista antes da entrevista	Antecipadamente obtém-se um consentimento informado do entrevistado relativamente à gravação em formato audio da entrevista que será utilizada para posterior transcrição e análise dos conteúdos.
Localização da Entrevista	Preferencialmente no local de trabalho dos entrevistados, numa sala de reuniões isolada acústicamente.
Duração estimada entrevista	30 minutos

Tabela II: Guião da entrevista

Categorias	Variável/Tópico	Tema/Pergunta
Metadata	Dimensão da Empresa	Qual é a dimensão da empresa em número de colaboradores em tempo integral?
	Core Business	Qual é o Core Business da empresa?
	Dimensão dos projetos de Desenvolvimento	Qual é a dimensão temporal dos projetos de desenvolvimento?
	Criticidade dos Sistemas	Qual é a criticidade dos sistemas desenvolvidos? Dar o exemplo se o sistema quando falha pode por a vida humana em risco ou não, etc.
	Distribuição/Localização de Equipas	As equipas de projeto estão Co-Localizadas ou estão distribuídas geograficamente?
	Recursos Dedicados/Partilhados	Os recursos alocados ao projeto são dedicados a esse projeto ou partilhados com outros projetos concorrentes?
	Posição na Empresa	Qual é a sua posição atual na empresa?
	Anos de Experiência Profissional	Quantos anos de Experiência Profissional na área do desenvolvimento de sistemas?
Processos e Frameworks	Existência de Processo Padrão na Empresa	Existe um Processo Padrão transversal a toda a empresa para os projetos de desenvolvimento de produto?
	Descrição do Processo	Será que pode descrever de forma breve o processo de desenvolvimento da Empresa?
	Grau de liberdade para desviar do processo	Existe grau de liberdade para realizar desvios ao processo padrão? Se sim como costuma executar esses desvios e que condições tem de gerir?
	Frameworks Utilizados	Quais são os frameworks ou metodologias utilizadas no processo de desenvolvimento?
Standards Externos	Standard Externos	A empresa segue standards externos? Dar o exemplo das Normas ISO se necessário.
	Compatibilidade dos Standard Externos e das Metodologias Ágeis	Na sua opinião acha que existe incompatibilidade entre os Standard Externos e as Metodologias Ágeis?
	Processo de Avaliação de Conformidade	Qual é o processo de avaliação da conformidade da empresa com os Standard Externos?
Experiência Pessoal	Principais desafios nos Projetos de Desenvolvimento	Quais são, do seu ponto de vista, os principais desafios nos projetos de desenvolvimento de sistemas nos dias de hoje?
	Resposta das Metodologias Tradicionais face aos desafios	Na sua opinião as Metodologias Tradicionais respondem de forma adequada aos desafios atuais?
	Processo de Qualidade	Qual é a importância da qualidade no processo atual, e de que forma esta é assegurada?
	Ciclo de Vida dos Produtos	Acha que o ciclo de vida e de desenvolvimento dos produtos está a diminuir, estável ou a aumentar? Este é um aspeto que tem levantado desafios?

Tabela III: Estrutura do Questionário

Categoria	Questão	Opção de Resposta
Utilização de Metodologias/Processos	A sua organização possui uma Metodologia ou Processo Padrão para o desenvolvimento de <i>software</i> /sistemas?	<p>Sim, todos os projetos da organização seguem o mesmo processo padrão ou portfólio de processos.</p> <p>Cada unidade organizacional tem a suas próprias abordagens/padrões, que todos os projetos daquela unidade devem seguir. Cada projeto pode selecionar o processo a ser seguido individualmente.</p>
	De que forma as abordagens, métodos e práticas foram implementados na sua organização?	<p>Planeado como parte de um programa de melhoria de processos</p> <p>Evoluído com a aprendizagem de projetos históricos ao longo do tempo</p> <p>Específico da situação, por favor explique:</p>
	Quais são os objetivos gerais que se pretende alcançar com a seleção das abordagens, métodos e práticas de desenvolvimento? Selecione todas as opções que se aplicam.	<p>Melhor envolvimento do cliente</p> <p>Melhor planejamento e estimativa (custo, tempo, etc.)</p> <p>Melhor monitorização e controlo do projeto (estatísticas de tempo real, tomada de decisão eficiente, etc.)</p> <p>Melhor reutilização de artefatos de projeto (modelos, código, testes, etc.)</p> <p>Melhor suporte de ferramentas</p> <p>Melhor frequência de entregas para clientes (entregas rápidas)</p> <p>Melhor qualidade externa do produto (percebida pelos clientes)</p> <p>Melhor qualidade interna dos artefatos (requisitos, testes, qualidade do código, etc.)</p> <p>Melhor transferência de conhecimento e aprendizagem</p> <p>Melhor satisfação dos colaboradores</p> <p>Melhor adaptabilidade e flexibilidade do processo para reagir a mudanças</p> <p>Melhor produtividade (redução de esforço, custo, etc.)</p> <p>Melhor educação e desenvolvimento da equipa (desenvolvimento das habilidades pessoais)</p> <p>Melhor capacidade organizacional de desenvolver sistemas críticos (habilidade de certificar um produto)</p> <p>Melhor maturidade organizacional (habilidade de atender a padrões, conformidade)</p> <p>Melhor gestão de riscos (ou seja, mitigar, eliminar ou transferir riscos)</p> <p>Melhor retorno de investimento - ROI (ou seja, ROI com dimensão temporal reduzida)</p> <p>Melhor time-to-market</p>
	De que forma foi definido o processo de gestão do seu projeto de desenvolvimento?	<p>Práticas e métodos específicos são selecionados no projeto conforme a necessidade</p> <p>Práticas e métodos específicos são selecionados de acordo com os requisitos dos clientes</p> <p>Um gestor de projetos define (personaliza) o processo no início do projeto</p> <p>O processo não é definido (personalizado) para o projeto (utilização direta do processo padrão da organização)</p> <p>Outra</p>

Tabela III: Estrutura do Questionário

Categoria	Questão	Opção de Resposta
	Indique qual(is) das seguintes metodologias ou processos utilizou (total ou parcialmente)?	<p>Cascata Tradicional</p> <p>Métodos Crystal</p> <p>DevOps</p> <p>Domain-Driven Design</p> <p>Dynamic Systems Development Method (DSDM)</p> <p>eXtreme Programming (XP)</p> <p>Feature Driven Development (FDD)</p> <p>Desenvolvimento Iterativo</p> <p>Kanban</p> <p>Large-Scale Scrum (LESS)</p> <p>Lean <i>software</i> Development</p> <p>Model-Driven Architecture (MDA)</p> <p>Nexus</p> <p>Personal <i>software</i> Process</p> <p>Modelo Stage-gate</p> <p>PRINCE2</p> <p>Rational Unified Process (RUP)</p> <p>Scaled Agile Framework (SAFe)</p> <p>Scrum</p> <p>ScrumBan</p> <p>Modelo Espiral</p> <p>Análise Estruturada de Sistemas</p> <p>Team Software Process</p> <p>Modelo V</p>
	Qual(is) das seguintes práticas utilizou? Deve selecionar todas as opções utilizadas.	<p>Especificações Arquiteturais</p> <p>Geração Automatizada de Código</p> <p>Prova Automatizada de Teoremas</p> <p>Testes Unitários Automatizados</p> <p>Gestão de Backlog</p> <p>Burn-Down Charts (como Prática de Monitoração do Progresso)</p> <p>Revisão de Código</p> <p>Padrões de Programação</p> <p>Posse Coletiva de Código</p> <p>Implementação Contínua</p> <p>Integração Contínua</p> <p>Reuniões Standup Diárias</p> <p>Definição de Pronto (Definition of Done / Ready)</p> <p>Revisões de Projeto (Desenho)</p> <p>Teste Destrutivo</p> <p>Projeto Detalhado/Especificações de Projeto (Desenho)</p> <p>Teste de Sistema Integral (End-to-End System Testing)</p> <p>Estimativa Baseada no Especialista/Equipa (por exemplo, Planning Poker)</p> <p>Estimativa Formal (por exemplo, COCOMO, Pontos de Função)</p> <p>Especificação Formal</p> <p>Planejamento de Iterações</p> <p>Revisões de Iteração/Sprint</p> <p>Limitar Trabalho em Andamento (por exemplo, utilizando quadros Kanban)</p> <p>Verificação de Modelos</p> <p>Cliente Internalizado</p>

Tabela III: Estrutura do Questionário

Categoria	Questão	Opção de Resposta
		Programação em Pares Prototipação Refatoração Planejamento de Entregas (Release Planning) Retrospectivas Scrum-de-Scrums Testes de Segurança Desenvolvimento Orientado a Testes (Test-driven Development) Modelagem de Casos de Uso (como Prática de Engenharia de Requisitos) User Stories (como Prática de Engenharia de Requisitos) Planejamento Baseado na Velocidade (Velocity-based planning)
	Por favor, indique qual o tipo de abordagem utilizada nas seguintes atividades de projeto: Gestão de Projetos Gestão de Riscos Análise/Engenharia de Requisitos Arquitetura/Desenho Técnico Implementação/Codificação Integração e Testes Lançamento e Operação Manutenção e Evolução	Totalmente Tradicional Híbrido Totalmente Ágil
Utilização de Metodologias e Standards	Implementa padrões ou normas externas na sua organização? (Ex: ISO 9001)	Não são implementados padrões externos Sim, implementamos padrões externos, em particular:
	O projeto ou produto a que sua resposta se refere também está sujeito a certificação?	Não, o referido projeto ou produto não está sujeito a certificação Sim, o referido projeto ou produto está sujeito a certificação, em particular:
	Considera que uma abordagem mais ágil representa um desafio para a implementação de padrões externos?	Não, a agilidade é fácil de ser implementada no nosso contexto Sim, a agilidade representa desafios para a implementação
Experiência Profissional	Baseado na sua experiência pessoal, por favor julgue as seguintes sentenças: A abordagem de desenvolvimento atual é adequado para apoiar o negócio do nosso projeto (por exemplo, referente a objetivos internos e do cliente) A abordagem de desenvolvimento atual ajuda a alcançar boa qualidade do produto A abordagem de desenvolvimento atual ajuda a alcançar um rápido time-to-market para o produto A abordagem de desenvolvimento atual fornece apoio suficiente no meu trabalho A abordagem de desenvolvimento atual trata de forma adequada os requisitos externos (por exemplo, padrões) que devemos implementar O nível de agilidade da abordagem de desenvolvimento atual é suficiente	1 (Discordo Totalmente) 2 3 (Neutro) 4 5 (Concordo Totalmente)

Tabela III: Estrutura do Questionário

Categoria	Questão	Opção de Resposta
	Em geral, estou satisfeito com a abordagem de desenvolvimento atual Se eu pudesse modificaria/melhoraria a abordagem de desenvolvimento atual	
	Por favor especifique os problemas que tenham surgido no decorrer do Projeto.	Texto Livre
	O projeto foi concretizado com sucesso (tempo, recursos e requisitos)?	Sim Não
Informação Geral e Metadata	Por favor, indique o Género	Masculino Feminino
	Indique a sua Formação Académica	Ensino Secundário Licenciatura Mestrado Doutoramento
	Indique o seu Grupo Etário	<20 anos 20-29 anos 30-39 anos 40-49 anos 50-60 anos
	Qual é a dimensão da sua organização (em número de funcionários de tempo integral)?	Micro (<10 funcionários) Pequena (11-50 funcionários) Média (51-250 funcionários) Grande (251-2499 funcionários) Muito Grande (>2500 funcionários)
	Qual é a dimensão do projeto no qual a sua resposta se refere?	Muito Pequeno (<2 pessoas semana) Pequeno (2 pessoas semana - 2 pessoas mês) Médio (2 pessoas mês - 6 pessoas mês) Grande (6 pessoas mês - 1 pessoa ano) Muito Grande (mais de 1 pessoa ano)
	Qual o domínio(s) de aplicação do projeto a que sua resposta se refere?	Automóvel/Automotive Aeronautia/Aeroespacial Aplicações/Serviços na Cloud Defesa/Militar Energia/Eficiência Energética Internet of Things Dispositivos Médicos e de Saúde Aplicações Móveis Robótica (por exemplo, robôs autónomos, UAVs/drones) Sistemas aeroespaciais Telecomunicações Outro (Texto Livre)
	No projeto a que se refere, indique a que consequências uma falha do sistema poderá levar. Selecione todas as respostas que considere que se aplicam.	... ameaçar a saúde ou vida humana ... levar à perda completa do sistema ... levar à degradação do sistema ... ter consequências legais (lei civil)

Tabela III: Estrutura do Questionário

Categoria	Questão	Opção de Resposta
		... ter consequências legais (lei criminal) ... afetar o meio ambiente ... afetar a reputação da organização ... after os negócios da organização Outra (Texto Livre)
	O projeto a que sua resposta se refere está a ser conduzido/desenvolvido de forma geograficamente distribuída?	Não Sim, nacionalmente (mesmo país) Sim, regionalmente (mesmo continente) Sim, globalmente
	Qual o seu principal papel no projeto a que as respostas se referem?	Analista/Engenheiro de Requisitos Arquiteto Diretor (por exemplo, CIO, CTO, etc.) Engenheiro de Desenvolvimento/Developer Gestor de Projetos/Equipa Scrum Master/Agile Coach Tester Outro (Texto Livre)
	Quantos anos de experiência possui no desenvolvimento de /sistemas?	<3 anos 3-5 anos 6-10 anos >10 anos

Tabela IV: Metodologias agregadas por categoria/abordagem

Metodologia ou Processo	Categoria
Métodos Crystal	Ágil
DevOps	
Dynamic Systems Development Method (DSDM)	
eXtreme Programming (XP)	
Feature Driven Development (FDD)	
Kanban	
Large-Scale Scrum (LESS)	
Lean Development	
Scaled Agile Framework (SAFe)	
Scrum	
ScrumBan	
Domain-Driven Design	Genérica
Desenvolvimento Iterativo	
Model-Driven Architecture (MDA)	
Team Process	
Cascata Tradicional (<i>Waterfall</i> Model)	Tradicional
Modelo Stage-gate	
PRINCE2	
Rational Unified Process (RUP)	
Modelo Espiral	
Análise Estruturada de Sistemas	
Modelo V	

Anexo B

Tabela V: Quadro de análise de respostas das entrevistas semi-estruturadas

	Entrevistado 1	Entrevistado 2	Entrevistado 3	Entrevistado 4
Metadada	<p>Dimensão da Empresa > Muito Grande (;>10k)</p> <p>Core Business Electrónica Automóvel</p> <p>Dimensão dos projetos de 1 a 2 anos</p> <p>Desenvolvimento Falha sem impacto nos restantes sistemas adjacentes.</p> <p>Criticidade dos Sistemas Distribuídas Regionalmente (Continente)</p>	<p>>Muito Grande (;>10k)</p> <p>Electrónica Automóvel 1 a 2 anos</p> <p>Pode por em causa a vida humana.</p> <p>Distribuídas Globalmente</p>	<p>>Muito Grande (;>10k)</p> <p>Electrónica Automóvel 1 a 2 anos</p> <p>Pode por em causa a vida humana.</p> <p>Distribuídas Globalmente</p>	<p>>Muito Grande (;>10k)</p> <p>Infotainment Cerca de 1 ano</p> <p>Falha sem impacto nos restantes sistemas adjacentes.</p> <p>Co-Localizadas</p>
Processos e Frameworks	<p>Equipas Partilhados</p> <p>Estratégia de Recursos (Dedicados/Partilhados) Sim</p> <p>Existência de Processo Padrão na Empresa</p> <p>Condição: Grau de liberdade para desviar do processo Padrão</p>	<p>Partilhados</p> <p>Sim</p> <p>Condição: Condicionado - Sujeito a adendas ou subprojectos para adicionar vertentes novas ao processo original</p>	<p>Partilhados</p> <p>Sim</p> <p>Condição: Condicionado - Existe sempre um certo grau de liberdade normalmente despoletado pelo cliente devido a novos requisitos.</p>	<p>Depende da complexidade do Projeto: Complexos-Dedicados; Simples-Partilhados Sim</p> <p>Não existe liberdade para desviar do processo.</p>
Standards Externos	<p>Frameworks Utilizados Scrum e Waterfall</p> <p>Standard Externos IATF</p> <p>Compatibilidade dos Standard Externos e das Metodologias Ágeis Não identifica incompatibilidade das Metodologias Ágeis com os Standard Externos existentes.</p>	<p>Waterfall e Stage Gate ISO e GSS</p> <p>Auditoria do Cliente (Externa)</p>	<p>Waterfall e Stage Gate IATF</p> <p>Incompatibilidades nas entregas de Software que ocorre após validação total do produto o que não é compatível com os ciclos de desenvolvimento curtos das metodologias ágeis; as normas externas utilizadas não foram desenhadas considerando metodologias ágeis.</p>	<p>Waterfall e Stage Gate ISO</p> <p>Identifica incompatibilidades porque as estruturas estão adaptadas a metodologias tradicionais. Será necessário um esforço adicional por parte das organizações para transformar os processos dada a dimensão da estrutura existente.</p>
Experiência Pessoal	<p>Processo de Avaliação de Conformidade Auditoria do Cliente (Externa)</p> <p>Principais desafios nos projetos de Desenvolvimento Sincronização das janelas temporais/ciclos dos diferentes frameworks utilizados (Waterfall vs Scrum)</p>	<p>Janela de desenvolvimento definida pelo cliente; existe maior dificuldade em abranger atividades de validação dentro das janelas temporais requisitadas; alteração de requisitos frequentes; ciclos de desenvolvimento mais curtos; maior complexidade dos sistemas;</p>	<p>Pouca flexibilidade a alterações de requisitos com impacto no sistema reduzido; alteração de requisitos frequentes;</p>	<p>Auditoria Externa</p> <p>Time-to-market mais exigente; Alteração de requisitos frequente;</p>
Resposta das Metodologias Tradicionais face aos desafios	<p>Não dão uma resposta adequada.</p>	<p>Não dão uma resposta adequada.</p>	<p>Não dão uma resposta adequada. A indústria está integralmente baseada em metodologias tradicionais</p>	<p>Não dão uma resposta adequada. "As pessoas têm de tomar consciencia que é necessário fazer diferente para ter um resultado diferente."</p>
Processo de Qualidade	<p>Processo de validação integral assente numa bateria de testes previamente especificada</p>	<p>Existe sempre a necessidade de passar por esse processo de validação/bateria de testes predefinidos.</p>	<p>Nada é considerado para entrega sem estar totalmente validado.</p>	<p>-</p>

Anexo C

Tabela VI: Dimensão da organização dos respondentes

Dimensão da Organização dos Inquiridos	Frequência	Percentagem
Micro (<10 colaboradores)	1	1.5%
Pequena (11-50 colaboradores)	6	8.8%
Média (51-250 funcionários)	15	22.1%
Grande (251-2499 funcionários)	21	30.9%
Muito Grande (>2500 funcionários)	25	36.8%
Total	68	100%

Tabela VII: Dimensão dos projetos no qual se os inquiridos basearam as suas respostas

Dimensão do Projeto	Frequência	Percentagem
Muito Pequeno (<2 pessoas semana)	5	7.4%
Pequeno (2 pessoas semana - 2 pessoas mês)	5	7.4%
Médio (2 pessoas mês - 6 pessoas mês)	16	23.5%
Grande (6 pessoas mês - 1 pessoa ano)	17	25.0%
Muito Grande (mais de 1 pessoa ano)	25	36.8%
Total	68	1

Tabela VIII: Experiência Profissional no desenvolvimento de sistemas/software

Experiência Profissional	Frequência	Percentagem
<3 anos	24	35.3%
3-5 anos	19	27.9%
6-10 anos	15	22.1%
>10 anos	10	14.7%
Total	68	100.0%

Tabela IX: Faixa Etária dos inquiridos

Faixa Etária	Frequência	Percentagem
<20 anos	0	0.0%
20-29 anos	29	42.6%
30-39 anos	25	36.8%
40-49 anos	8	11.8%
50-60 anos	5	7.4%
>60 anos	1	1.5%

Tabela X: Género dos inquiridos do questionário

Género	Frequência	Percentagem
Masculino	53	77.9%
Feminino	15	22.1%
Total	68	100.0%

Tabela XI: A sua organização possui uma Metodologia ou Processo Padrão para o desenvolvimento de sistemas?

Opções	Respostas	Percentagem
Cada projeto/equipa pode selecionar o processo a ser seguido individualmente.	17	25.0
Cada unidade organizacional tem a suas próprias abordagens/padrões, que todos os projetos daquela unidade devem seguir.	29	42.6
Sim, todos os projetos da organização seguem o mesmo processo padrão ou portefólio de processos.	22	32.4
Total	68	100.0

Tabela XII: De que forma as abordagens, métodos e práticas foram implementados na sua organização?

Resposta	Frequência	Percentagem
Outras das opções apresentadas	40	58.8
Planeado como parte de um programa de melhoria de processos	28	41.2
Total	68	100.0

Resposta	Frequência	Percentagem
Outras das opções apresentadas	18	26.5
Evoluido com a aprendizagem de projetos históricos ao longo do tempo.	50	73.5
Total	68	100

Tabela XIII: Quais são os objetivos gerais que se pretende alcançar com a seleção das abordagens, métodos e práticas de desenvolvimento?

Resposta	Frequência	Percentagem
Outras das opções apresentadas	23	33.8
Melhor planeamento e estimativa (custo, tempo, etc.)	45	66.2
Total	68	100.0

Resposta	Frequência	Percentagem
Outras das opções apresentadas	32	47.1
Melhor qualidade externa do produto (percecionada pelos clientes)	36	52.9
Total	68	100

Resposta	Frequência	Percentagem
Outras das opções apresentadas	33	48.5
Melhor produtividade (redução de esforço, custo, etc.)	35	51.5
Total	68	100

Tabela XIV: De que forma foi definido o processo de gestão do seu projeto de desenvolvimento?

Resposta	Frequência	Porcentagem
O processo não é definido (personalizado) para o projeto (utilização direta do processo padrão da organização)	9	13.2
Outro (especifique)	1	1.5
Práticas e métodos específicos são selecionados de acordo com os requisitos dos clientes	19	27.9
Práticas e métodos específicos são selecionados no projeto conforme a necessidade	29	42.6
Um gestor de projetos define (personaliza) o processo no início do projeto	10	14.7
Total	68	100.0

Tabela XV: Utilização de Metodologias de Desenvolvimento

Metodologia(s) utilizada(s)	Respostas	Porcentagem	Categoria
Cascata Tradicional (<i>Waterfall Model</i>)	37	54.41%	Tradicional
Métodos Crystal	1	1.47%	Ágil
DevOps	18	26.47%	Ágil
Domain-Driven Design	3	4.41%	Genérica
Dynamic Systems Development Method (DSDM)	1	1.47%	Ágil
eXtreme Programming (XP)	2	2.94%	Ágil
Feature Driven Development (FDD)	7	10.29%	Ágil
Desenvolvimento Iterativo	8	11.76%	Genérica
Kanban	24	35.29%	Ágil
Large-Scale Scrum (LESS)	1	1.47%	Ágil
Lean Development	10	14.71%	Ágil
Model-Driven Architecture (MDA)	0	0.00%	Genérica
Modelo Stage-gate	1	1.47%	Tradicional
PRINCE2	4	5.88%	Tradicional
Rational Unified Process (RUP)	0	0.00%	Tradicional
Scaled Agile Framework (SAFe)	8	11.76%	Ágil
Scrum	39	57.35%	Ágil
ScrumBan	1	1.47%	Ágil
Modelo Espiral	0	0.00%	Tradicional
Análise Estruturada de Sistemas	6	8.82%	Tradicional
Team Process	0	0.00%	Genérica
Modelo V	3	4.41%	Tradicional
Outro (especifique)	2	2.94%	-

Tabela XVI: Utilização de Práticas de Desenvolvimento

Prática Utilizada	Respostas	Porcentagem
Especificações Arquiteturais	26	38.24%
Geração Automatizada de Código	4	5.88%
Prova Automatizada de Teoremas	0	0.00%
Testes Unitários Automatizados	24	35.29%
Gestão de Backlog	37	54.41%
Burn-Down Charts	12	17.65%
Revisão de Código	29	42.65%
Padrões de Programação	14	20.59%
Posse Coletiva de Código	7	10.29%
Implementação Contínua	27	39.71%
Integração Contínua	27	39.71%
Reuniões Standup Diárias	23	33.82%
Definição de Pronto (Definition of Done/Ready)	11	16.18%
Revisões de Projeto (Desenho)	17	25.00%
Teste Destrutivo	5	7.35%
Projeto Detalhado/Especificações de Projeto (Desenho)	16	23.53%
Teste de Sistema Integral (End-to-End System Testing)	27	39.71%
Estimativa Baseada no Especialista/Equipa	15	22.06%
Estimativa Formal	2	2.94%
Especificação Formal	9	13.24%
Planeamento de Iterações	13	19.12%
Revisões de Iteração/Sprint	22	32.35%
Limitar Trabalho em Andamento	11	16.18%
Verificação de Modelos	2	2.94%
Cliente Internalizado	1	1.47%
Programação em Pares	4	5.88%
Prototipação	20	29.41%
Refactoring	8	11.76%
Planeamento de Entregas (Release Planning)	23	33.82%
Retrospectivas	12	17.65%
Scrum-de-Scrums	9	13.24%
Testes de Segurança	12	17.65%
Desenvolvimento Orientado a Testes (Test-driven Development)	16	23.53%
Modelação de Casos de Uso (como Prática de Engenharia de Requisitos)	8	11.76%
User Stories (como Prática de Engenharia de Requisitos)	22	32.35%
Planeamento Baseado na Velocidade (Velocity-based planning)	9	13.24%

Anexo D

Tabela XVII: Utilização de Pares de Metodologias Combinadas

Combinação de Metodologias	Válido		Casos Ausente		Total	
	N	Porcentagem	N	Porcentagem	N	Porcentagem
Cascata * Crystal	1	1.5%	67	98.5%	68	100.0%
Cascata * DevOps	10	14.7%	58	85.3%	68	100.0%
Cascata * XP	2	2.9%	66	97.1%	68	100.0%
Cascata * FDD	2	2.9%	66	97.1%	68	100.0%
Cascata * Kanban	12	17.6%	56	82.4%	68	100.0%
Cascata * LESS	1	1.5%	67	98.5%	68	100.0%
Cascata * Lean	6	8.8%	62	91.2%	68	100.0%
Cascata * SAFe	4	5.9%	64	94.1%	68	100.0%
Cascata * Scrum	28	41.2%	40	58.8%	68	100.0%
Cascata * ScrumBan	1	1.5%	67	98.5%	68	100.0%
Stage-Gate * SAFe	1	1.5%	67	98.5%	68	100.0%
PRINCE2 * Crystal	1	1.5%	67	98.5%	68	100.0%
PRINCE2 * XP	1	1.5%	67	98.5%	68	100.0%
PRINCE2 * Kanban	2	2.9%	66	97.1%	68	100.0%
PRINCE2 * Scrum	3	4.4%	65	95.6%	68	100.0%
AES * Crystal	1	1.5%	67	98.5%	68	100.0%
AES * DevOps	2	2.9%	66	97.1%	68	100.0%
AES * FDD	1	1.5%	67	98.5%	68	100.0%
AES * Kanban	1	1.5%	67	98.5%	68	100.0%
AES * Lean	2	2.9%	66	97.1%	68	100.0%
AES * Scrum	3	4.4%	65	95.6%	68	100.0%
Modelo V * Kanban	1	1.5%	67	98.5%	68	100.0%
Modelo V * SAFe	1	1.5%	67	98.5%	68	100.0%

Tabela XVIII: Teste do Qui-Quadrado das variáveis *Waterfall* e *Scrum*

Cascata*Scrum	Valor	Graus de liberdade	Significância
Qui-quadrado de Pearson	11,140a	1	.001
Correção de continuidadeb	9.557	1	.002
N de Casos Válidos	68		

a. 0 células (.0%) esperavam uma contagem menor que 5. A contagem mínima esperada é 13.22.
b. Computado apenas para uma tabela 2x2

Tabela XIX: Coeficientes de Associação Estatística das Combinação *Waterfall* * *Scrum*

Medidas Simétricas	Valor	Aprox. Sig.
Coefficiente de Phi	.405	.001
V de Cramer	.405	.001
Coefficiente de contingência	.375	.001
Número de Casos Válidos	68	

Tabela XX: Resultados da agregação de Metodologias Utilizadas por Categoria

Categoria	Frequência	Porcentagem
Outra	24	35.3
Tradicional	44	64.7
Total	68	100.0

Categoria	Frequência	Porcentagem
Outra	13	19.1
Ágil	55	80.9
Total	68	100.0

Categoria	Frequência	Porcentagem
Outra	57	83.8
Genérico	11	16.2
Total	68	100.0

Tabela XXI: Tabela de Contingência das Categorias Ágil e Tradicional

Ágil * Tradicional		Ágil		Total
		Não	Sim	
Tradicional	Não	2	22	24
	Sim	11	33	44
Total		13	55	68

Tabela XXII: Teste de Hipóteses da Combinação de Categoria Ágil e Tradicional

Ágil * Tradicional	Valor	df	Significância
Qui-quadrado de Pearson	2.790a	1	.095
Correção de continuidade ^b	1.816	1	.178
N de Casos Válidos	68		

a. 1 células (25.0%) esperavam uma contagem menor que 5. A contagem mínima esperada é 4.59.
b. Computado apenas para uma tabela 2x2