

Departamento de Ciências e Tecnologias da Informação

**Análise**  
**de metodologias de desenvolvimento de *software***  
**utilizadas na gestão de Projetos**  
**no sector das TIC**

Rui Miguel Maia Bernardo

Dissertação submetida como requisito parcial para obtenção do grau de Mestre em  
Informática de Gestão

Orientadora:

Doutora Luísa Domingues, Professora Auxiliar,  
ISCTE-IUL

Coorientadora:

Doutora Teresa Calapez, Professora Auxiliar,  
ISCTE-IUL

Outubro de 2018

Análise de metodologias de desenvolvimento de software  
utilizadas na gestão de Projetos no sector das TIC

Rui Bernardo

Outubro,  
2018

Análise de metodologias de desenvolvimento de software utilizadas na gestão  
de Projetos no sector das TIC

Rui Bernardo

Outubro, 2018

*Esta página foi intencionalmente deixada em branco*

## Resumo

O sucesso de um projeto de desenvolvimento de *software* não se mede somente pelo resultado final, mas também pelo cumprimento de variáveis que compõem o ciclo de vida do projeto.

A escolha de boas práticas, a utilização ou não de metodologias de desenvolvimento de *software* e a gestão de fatores críticos de sucesso são pontos a que não nos podemos alhear e que permitem obter os resultados que pretendemos.

Com esta investigação apurou-se quais os fatores críticos de sucesso com maior influência no resultado final de um projeto, bem como a relação existente entre a adoção de determinadas metodologias e o resultado de um projeto. Para tal, optámos por uma abordagem metodológica qualitativa realizando um inquérito vocacionado para membros de equipas de projetos de desenvolvimento de *software*.

Os resultados obtidos permitiram comparar a situação da gestão de projetos de desenvolvimento de *software* em Portugal com a realidade que se passa noutros países, que, muitas vezes, apenas é conhecida através de estudos como, por exemplo, o realizado pelo 'The Standish Group'.

Da informação recolhida, foi possível verificar que os fatores críticos que maior influência têm no resultado de um projeto são “Infraestrutura e ferramentas de apoio”, “Comprometimento e motivação da equipa do projeto” e “Qualificações da equipa”.

Existe um longo caminho a percorrer na gestão de projetos em Portugal, mas aqui fica uma breve análise, em jeito de contributo, dos principais fatores a ter em conta, para que se possam implementar estratégias que resultem, cada vez mais, no sucesso dos projetos.

**Palavras-Chave:** Gestão de Projetos, Fatores Críticos de Sucesso, Projetos de software, Metodologias de desenvolvimento de software

## **Abstract**

The success of a software development project depends not only on the outcome, but on the many variables composing its life cycle. Good practice choices, using (or not) software development methodologies and managing critical success factors must not be ignored to achieve the intended results.

This research establishes which factors have major influence in final result relates analyzing the relationships between adopted methodologies and project's outcome. We have therefore chosen a qualitative methodological approach by holding a survey destined to software development project team members.

The results obtained allowed to compare the current situation of the management of software development projects in Portugal with the reality in other countries which is often only known through studies such as the one made by 'The Standish Group'.

From the information collected, it can be concluded that the critical success factors with major influence in the project's outcome are “Infrastructure and supporting tools”, “Commitment and motivation of project teams” and “Team qualifications”.

There is still a long way ahead regarding project management in Portugal, but here is a brief study, as a contribution, of the main factors which must be taken into account to implement strategies which will increasingly result in a success projects.

**Keywords:** Project Management, Critical Success Factors, Software Projects; Software Development Methodologies.

## **Agradecimentos**

A realização desta dissertação de mestrado contou com importantes apoios e incentivos sem os quais não seria possível terminar e aos quais estarei eternamente grato.

Às professoras Luísa Domingues e Teresa Calapez, pela sua orientação, apoio, disponibilidade, pelo saber que me transmitiram, pelas opiniões e críticas e pela total colaboração no solucionar de dúvidas e problemas que foram surgindo ao longo da realização deste trabalho.

Aos meus pilares basilares que em todas as fases da minha vida permitem encarar cada desafio, por mais difícil que seja, com um sorriso na cara e com o coração cheio: A minha namorada, pelo carinho, compreensão e alegria contagiante que permitiu que nunca descolasse do principal objetivo.

Os meus pais e irmã, pelos valores que sempre me transmitiram, por me ensinarem a lutar pelos meus ideais e a não desistir até alcança-los, sempre com humildade.

*"Ninguém escapa ao sonho de voar, de ultrapassar os limites do espaço onde nasceu, de ver novos lugares e novas gentes. Mas saber ver em cada coisa, em cada pessoa, aquele algo que a define como especial, um objeto singular, um amigo, é fundamental. Navegar é preciso, reconhecer o valor das coisas e das pessoas, é mais preciso ainda."*

Antoine de Saint-Exupery

## **Abreviaturas**

APOGEP – Associação Portuguesa de Gestores de Projetos

DSDM - Dynamic Systems Development Method

FDD - Feature-driven development

FCS – Fatores Críticos de Sucesso

PMI - Project Management Institute

ROI - Return on Investment

RUP - Rational Unified Process

SDLC - Software Development Life Cycle

SDM - Strategic Design & Management

SI - Sistemas de Informação

TDD - Test-Driven Development

TI - Tecnologia da informação

WIP - work-in-progress

XP - Extreme Programming

## Índice

<b>Capítulo 1 – Introdução</b> .....	<b>1</b>
<b>1.1. Enquadramento do tema e motivação</b> .....	<b>1</b>
<b>1.2. Questões e objetivos de investigação</b> .....	<b>2</b>
<b>Capítulo 2 – Revisão da Literatura</b> .....	<b>3</b>
<b>2.1. Metodologias de desenvolvimento de software</b> .....	<b>3</b>
<b>2.2. Metodologias Agile</b> .....	<b>5</b>
2.2.1. Scrum .....	8
2.2.2. Feature-Driven Development .....	9
2.2.3. Extreme Programming (XP).....	11
2.2.4. Lean .....	14
2.2.5. DSDM .....	16
<b>2.3. Metodologias Iterativas</b> .....	<b>17</b>
<b>2.4. RUP</b> .....	<b>18</b>
<b>2.5. Metodologias Tradicionais</b> .....	<b>19</b>
2.5.1. Waterfall.....	19
2.5.2. Agile vs Waterfall .....	21
<b>2.6. Fatores críticos de sucesso e insucesso na gestão de projetos de TIC</b> .....	<b>23</b>
2.6.1. The Standish Group – CHAOS Report .....	23
2.6.2. Outros estudos .....	27
2.6.3. Síntese dos Fatores Críticos de Sucesso.....	31
<b>2.7. Sumário</b> .....	<b>34</b>
<b>Capítulo 3 – Metodologia</b> .....	<b>35</b>
<b>3.1. Desenho de investigação</b> .....	<b>35</b>
<b>3.2. Hipóteses de investigação</b> .....	<b>35</b>
<b>3.3. Elaboração do questionário</b> .....	<b>37</b>
<b>3.4. Divulgação e Processo de amostragem</b> .....	<b>38</b>
<b>Capítulo 4 – Análise e discussão dos resultados</b> .....	<b>39</b>
<b>4.1. Perfil e dimensão da amostra</b> .....	<b>39</b>
<b>4.2. Pré-tratamento dos dados</b> .....	<b>40</b>
4.2.1. Q2 - Como classifica a dimensão da equipa no projeto?.....	40
4.2.2. Q5 - Em termos do resultado final como classifica o projeto?.....	41
4.2.3. Q8 - Em que medida os seguintes fatores afetaram o resultado final do projeto? ..	42
4.2.4. Q10 - Foram adotadas algumas metodologias/princípios? .....	45

<b>4.3. Teste das hipóteses em estudo.....</b>	<b>47</b>
4.3.1. Hipótese 1 - Que influência tem a dimensão da equipa no sucesso de um projeto .	47
4.3.2. Hipótese 2 - Que influência tem a dimensão do projeto no sucesso de um projeto	49
4.3.3. Hipótese 3 - Quais os fatores críticos de sucesso que têm maior influência num projeto	50
4.3.4. Hipótese 4 - Quais as principais razões que são apontadas como insucesso num projeto	52
4.3.5. Hipótese 5 - Quais as principais fases com maior relevância para o sucesso/insucesso do projeto .....	52
4.3.6. Hipótese 6 - Quais as metodologias mais utilizadas que resultaram em sucesso/insucesso.....	54
4.3.7. Hipótese 7 - Quais os principais impactos da escolha dessas metodologias .....	56
<b>4.4. Discussão dos resultados .....</b>	<b>58</b>
<b>Capítulo 5 - Conclusões, limitações e sugestões .....</b>	<b>59</b>
<b>5.1. Conclusões .....</b>	<b>59</b>
5.1.1. Chaos Report .....	59
5.1.2. Fatores Críticos de Sucesso .....	60
5.1.3. Metodologias de Desenvolvimento de Software .....	61
5.1.4. Conclusão final .....	61
<b>5.2. Limitações do trabalho .....</b>	<b>63</b>
<b>5.3. Sugestões para pesquisa futura.....</b>	<b>63</b>
<b>Anexo A .....</b>	<b>66</b>
<b>Anexo B .....</b>	<b>67</b>
<b>Anexo C .....</b>	<b>69</b>
<b>Anexo E .....</b>	<b>75</b>
<b>Anexo F.....</b>	<b>76</b>
<b>Anexo G.....</b>	<b>80</b>

## Índice de Tabelas

Tabela 1 : Práticas de eXtreme Programming Fonte:Agile Practice Guide by Project Management Institute .....	13
Tabela 2 : Comparação Waterfall vs Agile. Adaptação de Harden 2011 .....	22
Tabela 3 : Modern resolution for all projects Fonte: The Standish Group (2015).....	24
Tabela 4 : Chaos resolution by project size Fonte: The Standish Group (2015) .....	24
Tabela 5 : Chaos resolution by Agile vs Waterfall Fonte: The Standish Group (2015) .....	25
Tabela 6 : Chaos factors of success Fonte: The Standish Group (2015).....	26
Tabela 7 : CSF categories and rankings based on methodologies Fonte: Ahimbisibwe et al. (2015) .....	27
Tabela 8 : CSF categories and rankings based on methodologies Fonte :Ahimbisibwe et al. (2015) .....	29
Tabela 9 : Critical success factors identified across 63 publications Fonte: Fortune e White (2006) .....	30
Tabela 10 : Lista de Fatores Críticos Utilizados no estudo .....	31
Tabela 11 : Frequência da variável Q2 - Como classifica a dimensão da equipa no projeto?. 40	
Tabela 12 : Frequência da nova variável “Dimensao_Equipa” com base na pergunta Q2 .....	41
Tabela 13 : Frequência da variável Q5 - Em termos do resultado final como classifica o projeto.....	41
Tabela 14 : Frequência da nova variável “Project_Result” com base na pergunta Q5 .....	41
Tabela 15 : Matriz de Pesos loading, após rotação, numa solução com 4 componentes e correspondente % de var explicada .....	43
Tabela 16 : Frequência da variável correspondente à pergunta Q10 .....	45
Tabela 17 : Frequência da nova variável “Tipo de Metodologias” com base na pergunta Q10 .....	46
Tabela 18 : Tabela de referência cruzada entre "Dimensao_Equipa" e "Projec_Result".....	48
Tabela 19 : Tabela de referência cruzada entre " Qual a dimensao do projecto" e "Project_Result" .....	49
Tabela 20 : Resumo do modelo .....	50
Tabela 21 : Teste de Hosmer e Lemeshow .....	51
Tabela 22 : Variáveis na Equação .....	51
Tabela 23 : Análise descritiva da variável “Porque classifica como insucesso o Projeto.....	52
Tabela 24 : Teste de Qui-Quadrado para a hipótese 5 .....	53

## Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

Tabela 25 : Tabela de referencia cruzada entre as variáveis “Fase do projecto que teve maior relevância para os resultados do projeto” e “Project_Result” .....	54
Tabela 26 : Teste Qui-Quadrado para a Hipótese 6 .....	55
Tabela 27 : Tabela de referência cruzada entre as variáveis "Tipo de Metodologias" e "Project_Result".....	56
Tabela 28 : Analise descritiva da variável “Quais os maiores impactos da adoção das metodologias escolhidas anteriormente?” .....	57
Tabela 29 : Tabela de referência cruzada entre as variáveis "Tipo de Metodologias" e “Quais os maiores impactos da adoção das metodologias escolhidas anteriormente?” .....	57
Tabela 30 : Resultados dos Teste das Hipóteses .....	58
Tabela 31 : Resumo das conclusões finais .....	61

## Índice de Figuras

Figura 1 : “Adoção de Agile continua a crescer”. Fonte: Forrester (2009). .....	4
Figura 2 : “Scrum começou a ser muito popular”. Fonte: ”. Fonte: Forrester (2009).....	6
Figura 3 : Os ciclos de vida Agile Iteration-Base e Flow-Base Adaptado de:Agile Praticce Guide by Project Management Institute .....	7
Figura 4 : Metodologia Scrum Fonte: (Sutherland, 2012) .....	9
Figura 6 : Tabela Kanban demonstração dos limites do trabalho em progresso, de forma a otimizar o fluxo de trabalho Adaptado de:Agile Praticce Guide by Project Management Institute. Fonte: Project Management Institute, (2017).....	15
Figura 7 : DSDM aproxima-se do conceito Constraint-Driven Agility Adaptado de:Agile Praticce Guide by Project Management Institute. Fonte: Project Management Institute, (2017) .....	16
Figura 8 : Ciclo de vida Iterativo Adaptado de:Agile Praticce Guide by Project Management Institute. Fonte: Project Management Institute, (2017).....	18
Figura 9 : Arquitetura do RUP Fonte: MARTINS (2007) .....	19
Figura 10 : Ciclo de Vida Waterfall Adaptado de:Agile Praticce Guide by Project Management Institute. Fonte: Project Management Institute, (2017).....	20
Figura 11 : Fases de Investigação .....	35

## Capítulo 1 – Introdução

### 1.1. Enquadramento do tema e motivação

A realização de um projeto pode ser bastante complexa. Contudo, existem determinados elementos que não podem falhar e que potenciam o sucesso do projeto. A estes elementos chave dá-se o nome de ‘Fatores Críticos de Sucesso’. Acontece que estes fatores não são estanques e variam de área de negócio para área de negócio, de empresa para empresa e até de projeto para projeto. A correta escolha e adoção dos mesmos influencia o resultado final do projeto.

Alguns autores identificaram quais os fatores críticos com maior referência bibliográfica em estudos anteriormente feitos, de forma a poderem criar uma compilação de quais os fatores de sucesso mais predominantes e, como tal, indispensáveis na condução de um projeto.

O *The Standish Group* tem lançado, ao longo dos anos, o *CHAOS Report*, um relatório que estuda o sucesso e insucesso de 50.000 projetos num total de 1.000 empresas de desenvolvimento de *software*. Em todos esses relatórios é perceptível o longo trabalho que ainda é necessário fazer para aumentar as probabilidades de sucesso dos projetos.

Tendo isto em conta, o presente estudo passa pela análise da situação atual em Portugal na área de TI, nomeadamente, quais os fatores críticos mais importantes para o sucesso de um projeto, quais as metodologias de desenvolvimento de software mais utilizadas, como é que as mesmas são implementadas, bem como qual o impacto que têm no resultado do projeto, com vista à confrontação final destes resultados com os obtidos pelo estudo realizado, em 2015, pelo *The Standish Group* que analisa projetos à escala mundial.

Tal permitir-nos-á ter uma perceção global do caminho que a gestão de projetos está a trilhar em Portugal, bem como quais os fatores críticos de sucesso que devemos ter em conta de forma a obtermos melhores resultados nos projetos de desenvolvimento de *software*.

## 1.2. Questões e objetivos de investigação

Os principais objetivos da presente investigação serão:

- Identificar quais os Fatores Críticos de Sucesso (FCS) que maior influência têm no sucesso e insucesso dos projetos de desenvolvimento de software,
- Verificar se existe relação entre escolha das metodologias de desenvolvimento de software e o resultado final de um projeto.

Tal permitirá perceber quais as melhores práticas a adotar na condução de um projeto com vista ao seu sucesso, bem como identificar quais os principais motivos do insucesso dos projetos.

Assim sendo, com a realização desta dissertação, pretende-se responder às seguintes questões:

**P1:** Quais são as principais metodologias adotadas na realização de projetos de *software* em Portugal?

**P2:** Quais são os fatores com maior influência no sucesso e insucesso dos projetos de *software* em Portugal?

**P3:** Existe uma relação do resultado do projeto com a escolha da metodologia utilizada no desenvolvimento de software?

## Capítulo 2 – Revisão da Literatura

### 2.1. Metodologias de desenvolvimento de software

Entende-se por ‘metodologias’ um conjunto estruturado de práticas, que visa atingir um objetivo, e que poderá estar relacionado com aspetos técnicos como a programação ou não técnicos como a gestão (Avison e Fitzgerald, 2003), (Marks, 2012), (Aitken e Ilango, 2013), (Snyder, 2014).

O objetivo maior da adoção de uma metodologia é aumentar as probabilidades de sucesso de um projeto, permitindo um maior controlo do desenvolvimento do mesmo, reduzindo riscos, garantindo prazos de entrega, etc..

À semelhança do que acontece com os fatores críticos de sucesso, é necessário adotar uma metodologia que se enquadre na realidade do negócio/empresa/projeto (Paulson, 2001), (Kerzner, 2002), (Charvat, 2003), (Chin e Spowage, 2010).

As empresas de desenvolvimento de software, muitas delas em ambientes de múltiplos projetos, têm adotado metodologias com vista à melhoria da gestão dos projetos e ao uso mais eficiente dos seus recursos (Dai e Wells, 2004).

Segundo Forrester (2009), num estudo que realizou e no âmbito do qual inquiriu 1.023 profissionais de TI, as metodologias de desenvolvimento dividem-se em três grandes grupos: metodologias tradicionais, metodologias ágeis e iterativas. *Waterfall* é um exemplo de metodologia tradicional, com os seus processos simples. No que tange às metodologias ágeis menciona SCRUM, Extreme Programming, Lean, DSDM e FDD e como metodologia iterativa indica a metodologia RUP.

Apesar de Forrester (2009) definir RUP como uma metodologia iterativa é importante referir que esta é uma framework de processos de desenvolvimento de Software que fornece uma abordagem metodológica, no que diz respeito a tarefas e responsabilidades no desenvolvimento de Software organizacional.

No estudo realizado por Forrester (2009), concluiu-se que 38,6% dos profissionais utilizam metodologias Ágeis, 19,5% utilizam metodologias iterativas, 13% utilizam metodologias tradicionais e por fim 28,8% não utilizam qualquer metodologia no desenvolvimento de software, tal como podemos observar na Figura 1.

## Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

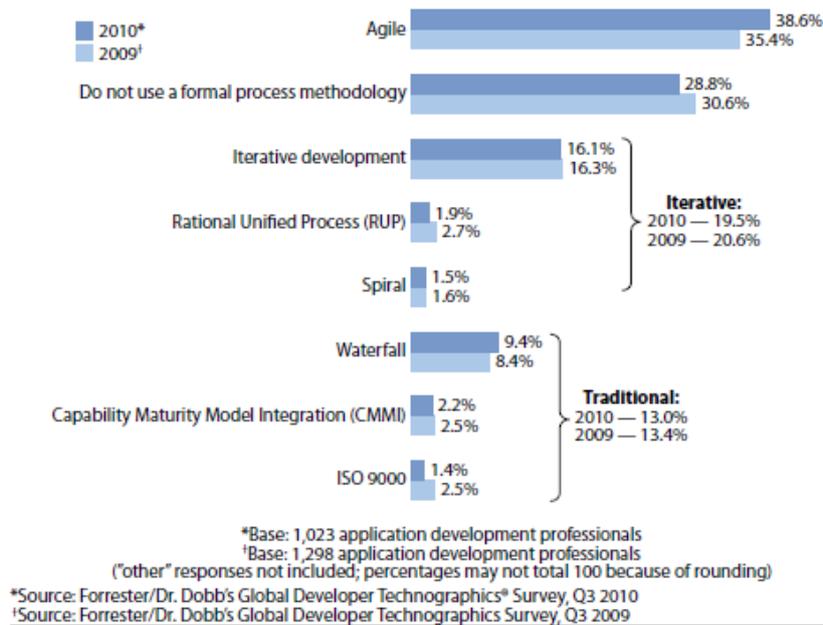


Figura 1 : “Adoção de Agile continua a crescer”. Fonte: Forrester (2009).

Existem inúmeras diferenças entre as metodologias ágeis e as metodologias tradicionais, sendo que a primeira grande diferença encontra-se na antecipação do planeamento. Isto é, nas metodologias tradicionais, o planeamento é feito, detalhadamente, antes de se iniciar o desenvolvimento, enquanto que nas metodologias ágeis o planeamento é feito de forma iterativa e incremental, vai-se descobrindo o percurso no próprio desenrolar do projeto.

Por outro lado, no entendimento de Cockburn (2001), as metodologias ágeis têm como principal foco a resolução de determinado problema com um orçamento e prazo fixo. Já nas metodologias tradicionais, as especificações são mais importantes que o prazo e custo. Todos os detalhes são definidos com antecedência, o prazo e o custo necessário para atingir essas especificações serão uma consequência.

Em termos técnicos, os dois tipos de abordagem diferenciam-se na forma como tratam as questões de prazo, custo, especificações e qualidade.

Com base em tudo o *supra* exposto, estão descritas nas secções seguintes algumas das metodologias ágeis, tradicionais e iterativas mais comuns no sector das TI.

## 2.2. Metodologias Agile

Nos anos 80, um grupo de profissionais da área de desenvolvimento informático, decidiu reunir-se para discutir formas de melhorar o desempenho dos seus projetos fazendo face às metodologias ditas “tradicionais”.

Embora cada um dos envolvidos tivesse as suas próprias práticas e teorias acerca de como um projeto de *software* deveria ser realizado com sucesso, todos chegaram a acordo quanto ao principal objetivo: cumprir os prazos e requisitos do cliente. Com base nisso criaram o Manifesto para o Desenvolvimento Agile de *software*, frequentemente chamado apenas de Manifesto Agile. O termo “Desenvolvimento Agile” passou a descrever abordagens de desenvolvimento que seguissem estes princípios/objetivos.

As equipas ágeis preferem *software* funcional à documentação exaustiva, pois isso conduz a uma versão mais estável e melhorada do produto que está a ser desenvolvido. Estas equipas valorizam a resposta à mudança em detrimento do seguimento “cego” de planos estipulados, pois o objetivo principal é criar o maior valor possível para os clientes e utilizadores do projeto.

“Desenvolvimento Agile” é o termo utilizado por diferentes metodologias e *frameworks* que desenvolvem *software* de forma iterativa e incremental. Segundo Rachel Burger(2018) algumas destas são mais indicadas para determinados projetos, mas as metodologias Agile mais comuns, como podemos ver na Figura 2, são: *Extreme Programming (XP)*, *Scrum*, *Lean Development*, *Feature-Driven Development (FDD)*, *DSDM*.

Destas, o *Scrum* é, de longe, o *framework* mais utilizado com 12,3%, no questionário realizado por “Forrester / Dr. Dobbs Global Developer Technographics® Survey, Q3 2010”.

## Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

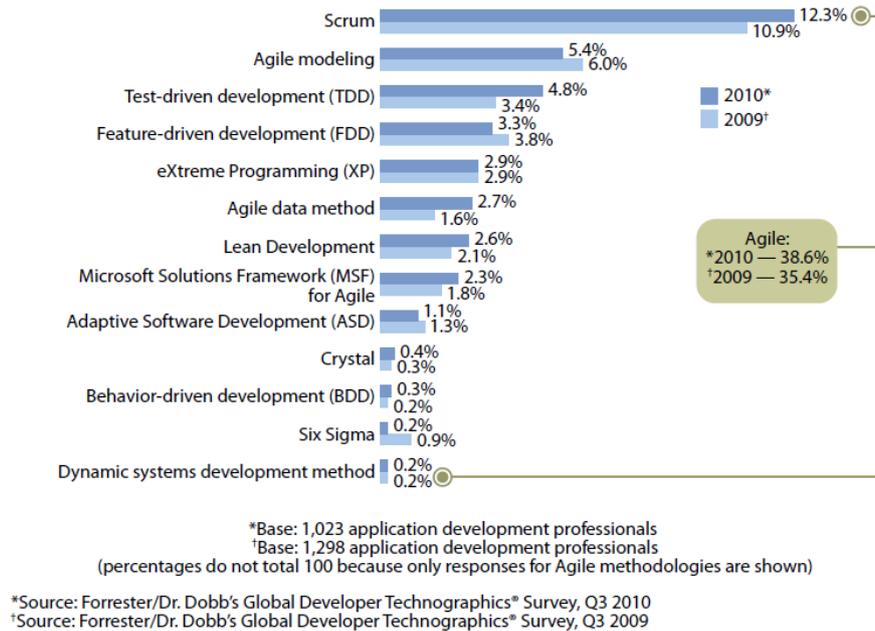


Figura 2 : “Scrum começou a ser muito popular”. Fonte: Forrester (2009).

O ciclo de vida de desenvolvimento de *software* Agile baseia-se nos modelos de processos iterativos e incrementais. Centra-se na adaptabilidade à mudança de requisitos dos produtos e na melhoria da satisfação do cliente através da participação deste e na entrega rápida das principais funcionalidades dos produtos. Os métodos ágeis concentram-se principalmente na divisão de entregas do produto, facilmente desenvolvíveis através de ciclos incrementais conhecidos como "*sprints*". Project Management Institute, (2017)

### ❖ Características

Num ambiente Agile, a equipa espera pedidos de alterações. As abordagens iterativas e incrementais fornecem ‘feedbacks’ que permitem planear melhor a próxima parte do projeto. No entanto, em projetos ágeis, a entrega faseada costuma conter pedidos mal interpretados/incompreendidos.

A Figura 3 ilustra duas formas possíveis de obter uma entrega faseada para que o projeto se alinhe às necessidades do cliente e possa ser adaptado conforme as necessidades.

## Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC



Figura 3 : Os ciclos de vida Agile Iteration-Base e Flow-Base Adaptado de: Agile Practice Guide by Project Management Institute

Na primeira forma (Iteração com base em Agile), a equipa trabalha em iterações, sendo o tempo despendido em cada tarefa igual. A equipa foca-se na tarefa mais importante e trabalha toda, em colaboração, para completar a mesma, só mudando para a seguinte depois de esta estar concluída.

Na segunda forma (Fluxo baseado em Agile) a equipa define o trabalho a fazer através de colunas num quadro de tarefas e vai atualizando as diversas fases. Cada tarefa, ao contrário do primeiro exemplo, pode demorar períodos de tempo diferentes a concluir. A equipa pode ainda decidir manter o trabalho em progresso com pequenas tarefas de forma a poder identificar com antecedência os problemas e, dessa forma, reduzir o tempo para efetuar as alterações que possam ser necessárias.

Os ciclos da vida ágeis são aqueles que cumprem os princípios do Manifesto Agile. Em particular, a satisfação do cliente aumenta com a entrega antecipada e contínua do projeto. Além disso, um produto incremental que é funcional e fornece valor é a principal medida de progresso. Project Management Institute, (2017)

### 2.2.1. Scrum

Scrum consiste num processo de desenvolvimento iterativo e incremental para gestão de projetos de desenvolvimento Agile.

Não é um processo prescritivo, mas antes um processo que descreve tópicos de como o processo deve decorrer. (Lei et al., 2017)

#### **Características principais:**

- Equipas pequenas, multidisciplinares, altamente treinadas e focadas;
- Prazos muito agressivos;

#### **Hayata e Han (2011), descrevem os seguintes conceitos:**

- Product Owner: A quem se destina o produto final e pode ser externo ou interno, sendo da sua responsabilidade a conceção das especificações do produto;
- Backlog: Lista de todas as funcionalidades a serem desenvolvidas durante o projeto, sendo bem definida e detalhada no início do trabalho, deve ser listada e ordenada por prioridade de execução.
- Sprint: Período não superior a 30 dias, onde o projeto (ou apenas algumas funcionalidades) é desenvolvido.
- Sprint Backlog: Trabalho a ser desenvolvido num Sprint de modo a criar um produto a apresentar ao cliente. Deve ser desenvolvido de forma incremental, relativa ao *Backlog* anterior (se existir).
- Scrum meeting: Reunião diária onde são avaliados os progressos do projeto e os problemas encontrados durante o desenvolvimento.
- Scrum Meeting Rules: Protocolo a seguir de modo a realizar uma reunião Scrum.
- Scrum Team: A equipa de desenvolvimento de um Sprint.
- Scrum Master: Elemento da equipa responsável pela gestão do projeto e liderar as Scrum Meetings. Apesar de ser gestor não tem propriamente autoridade sobre os demais membros da equipa de forma a incentivar a autogestão.

#### **Processo:**

Na sua essência, este método Agile é composto por ciclos de trabalho (Sprint), geralmente com a duração de 30 dias de calendário e ciclos diários (*Daily Scrum*). No início, são definidos os

Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

objetivos (*Sprint Backlog*) que se pretendem atingir no final de cada ciclo de trabalho tendo sempre presente que nessa altura o projeto deverá ter avançado mais um degrau na escada que conduz à conclusão/implementação deste, conforme se pode verificar na Figura 4 A intenção é, no final de cada ciclo de trabalho, serem notórios os progressos do projeto em curso, aproximando-o cada vez mais da sua finalização. (Schwaber e Sutherland, 2013)

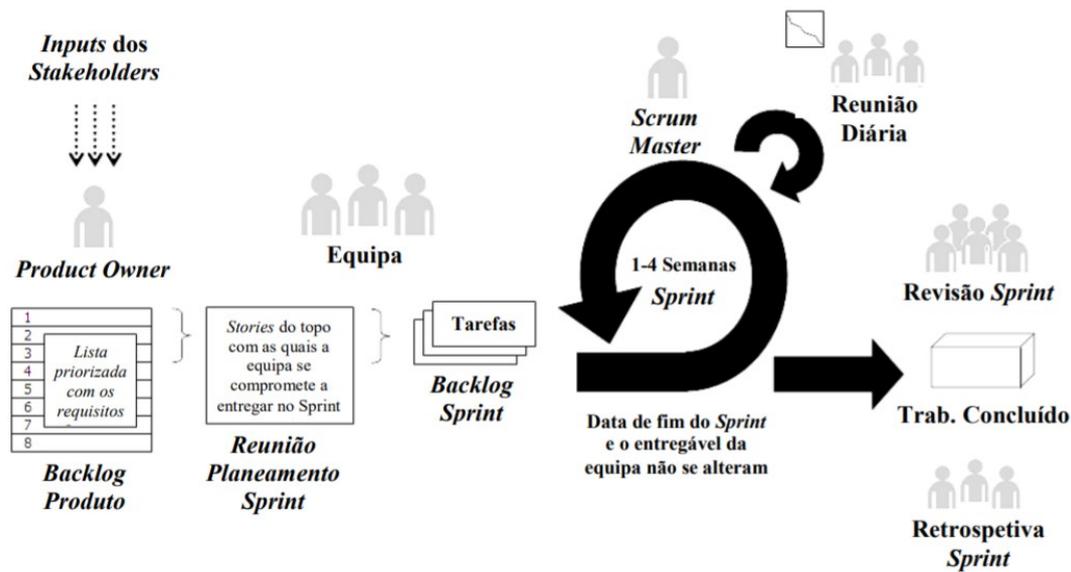


Figura 4 : Metodologia Scrum Fonte: (Sutherland, 2012)

## 2.2.2. Feature-Driven Development

*Feature-Driven Development* (FDD) foi desenvolvido para atender às necessidades específicas do desenvolvimento de *software* num projeto.

No FDD os membros da equipa comportam-se como uma pequena empresa, existindo vários papéis que os mesmos podem assumir. Mais, o mesmo papel pode ser assumido por vários membros e cada membro pode assumir vários papéis simultaneamente. (Pressman, 2005)

**Existem três tipos de papéis: os principais, secundários e os adicionais.**

- Os papéis principais são: Gestor de Projeto, Chefe de *Design*, Gestor de Desenvolvimento, Programador Chefe, Programador e Especialista da Área.
- Os papéis secundários são: Gestor de Atividade, Guru da Linguagem, Engenheiro de *Builds* e Administrador de Sistema.

## Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

- Os papéis adicionais são: *Tester*, Suporte, Documentador.

Um projeto de desenvolvimento orientado a recursos é organizado em torno de cinco processos ou atividades, que são realizadas iterativamente:

- Desenvolvimento de um modelo geral;
- Construção de uma lista de características;
- Planeamento por funcionalidade;
- Desenho por funcionalidade;
- Desenvolvimento por funcionalidades;

O fluxo do ciclo de vida e a interação desses cinco processos estão ilustrados na Figura .

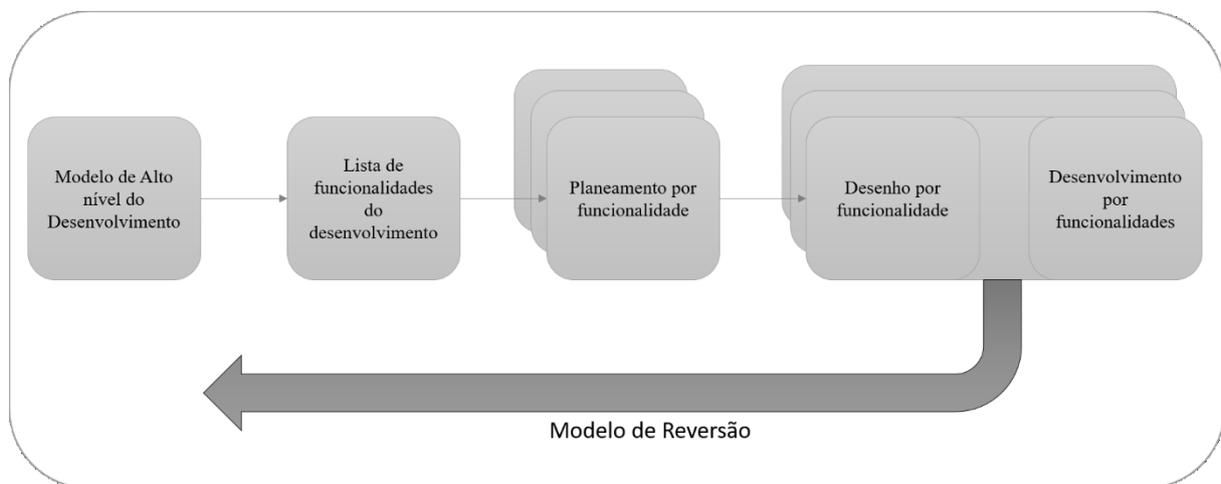


Figura 5 : *Featura-Driven Development* Ciclo de vida do projeto. Adaptado de: *Agile Practice Guide* by Project Management Institute. Fonte: Project Management Institute, (2017)

As atividades de desenvolvimento orientadas por recurso são suportadas por um conjunto básico de melhores práticas de engenharia de *software* (Rico, Sayani e Sone, 2009).:

- Modelagem de objetos;
- Desenvolvimento por funcionalidade;
- Autoria individual;
- Equipas por característica;
- Inspeções;
- Integração regular;
- Gestão de configurações;
- Reportar o progresso e resultados.

Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

A utilização do FDD num projeto, torna recomendável o uso de uma ferramenta que permita organizar todas as implementações que se deseje criar, sendo possível a inclusão e discriminação de todas as componentes necessárias para as novas funcionalidades. Uma ferramenta deste tipo deve ter alguns requisitos necessários para facilitar a utilização de FDD (exemplo: FDD Manager, MagicDraw, OptimalJ, Poseidon).

### 2.2.3. Extreme Programming (XP)

*Extreme Programming* (XP) é uma metodologia de desenvolvimento de *software* baseado em ciclos frequentes.

Apresenta como valores fundamentais a simplicidade, comunicação, feedback, coragem e respeito (SCFCR), valores que se manifestam nas práticas desenvolvidas ao longo do ciclo de vida do XP. (Anand e Dinakaran, 2016)

#### Características principais:

- Equipas pequenas, altamente treinadas e focadas.
- Trabalho em áreas comuns.
- Multidisciplinaridade.
- Prazos muito agressivos.

#### De acordo com Beck (2000), os papéis são:

##### • Customer:

- Cria e prioriza as *Stories* a desenvolver.
- Ao contrário de outras metodologias, o *Customer* pode alterar as datas da *release* e adicionar ou remover *Stories* ao *Backlog*.

##### • Programmer:

- Estima as *Stories* e é responsável pelas tarefas.
- Desenvolve código e testes.
- Normalmente trabalham em pares.

##### • Coach:

- É opcional, mas importante pois monitoriza todo o processo e a equipa.
- Ajuda a equipa a identificar e focar-se nos riscos e oportunidades de otimização.

## Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

### •Tracker:

- É opcional, mas importante pois monitoriza todo o progresso e alerta quando é necessário ajustar as tarefas ao calendário.
- Pode ser exercido pelo “segundo” programador.

### **Práticas Core são definidas por (Beck, 2000) da seguinte forma:**

- Whole Team: Todos os *stakeholders* de um projeto XP devem sentar-se juntos no mesmo local, como membros de uma única equipa. Uma equipa inclui programadores, responsáveis da qualidade e um "cliente" de um representante da unidade de negócio, que estabelece os requisitos e prioridades do projeto.
- Planning Game: Técnica usada para o processo de levantamento (*elicitation*) de novos requisitos e respetiva estimativa de esforço necessária por parte da equipa para o desenvolvimento.
- Customer Test: O cliente XP define um ou mais testes de aceitação automatizados para mostrar que a funcionalidade está a funcionar. A equipa desenvolve estes testes e usa os mesmos para provar a si e ao cliente que a funcionalidade está implementada corretamente.
- Small Releases: Começar com o menor conjunto de requisitos e entregas em intervalos curtos.
- Simple Design: Garantir a simplicidade do trabalho, o código deve ser baseado apenas nas necessidades atuais.
- Pair Programming: Técnica em que todo o *software* é construído por conjuntos de dois programadores sentados lado a lado na mesma máquina, assegurando avaliação dupla e garantindo melhor conceção, melhores testes e um melhor código.
- Test-Driven Development (TDD): Conceito de que, no desenvolvimento de *software*, um bom feedback exige um bom teste. Nunca se deve passar para a funcionalidade seguinte sem se testar primeiro e garantir que está a funcionar a 100%.
- Design Improvement (refactoring): Melhorar constantemente o produto sem alterar o seu comportamento de forma a garantir maior fiabilidade e adaptação às necessidades (*Business Value*).
- Continuous Integration: Manter o sistema integrado em todos os momentos e em todas as *releases*. É necessário garantir em cada funcionalidade testes integrados.

Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

- Collective Code Ownership: O código é da responsabilidade de todos e é permitido todas alterações necessárias ao mesmo.
- Coding Standard: Todos programam com os mesmos *standards* e convenções.
- Metaphor: Uso de nomes comuns para descrever como o programa funciona e ter a certeza de que todos entendem.
- Sustainable Pace: O *standard* de esforço de desenvolvimento é de 40h por semana. É permitido exceder este tempo em uma semana no máximo. Se for necessário mais, a metodologia não está a ser corretamente implementada.

XP é conhecida como um conjunto de práticas desenvolvidas para promover resultados nos projetos de desenvolvimento de *software* (Anand e Dinakaran, 2016). A metodologia formaliza primeiro o que entende como práticas primárias, mas tende gradualmente a adotar outras práticas como as listadas na Tabela 1.

Tabela 1 : Práticas de eXtreme Programming Fonte: Agile Practice Guide by Project Management Institute

<b>Área Prática de Xp</b>	<b>Primário</b>	<b>Secundário</b>
<b>Organizacional</b>	<ul style="list-style-type: none"> <li>• Reuniões constantes</li> <li>• Equipa como um todo</li> <li>• Espaço de trabalho Informativo</li> </ul>	<ul style="list-style-type: none"> <li>• Envolvência do cliente</li> <li>• Estabilidade dos membros da equipa</li> <li>• Ritmo constante</li> </ul>
<b>Técnico</b>	<ul style="list-style-type: none"> <li>• Desenvolvimento em paralelo</li> <li>• Testar antes de desenvolver</li> <li>• Desenho incremental</li> </ul>	<ul style="list-style-type: none"> <li>• Partilha do conhecimento técnico</li> <li>• Documentação do código e dos testes</li> <li>• Refazer</li> </ul>
<b>Planeamento</b>	<ul style="list-style-type: none"> <li>• Ciclos semanais</li> </ul>	<ul style="list-style-type: none"> <li>• Analisar a causa principal do problema</li> <li>• Equipas pequenas</li> <li>• Pagamento por desenvolvimento</li> <li>• Espaço de negociação no contrato</li> <li>• Briefing diário</li> </ul>
<b>Integração</b>	<ul style="list-style-type: none"> <li>• Desenvolvimentos em 10 minutos</li> <li>• Integração continua</li> <li>• Testes Iniciais</li> </ul>	<ul style="list-style-type: none"> <li>• Base simples de desenvolvimento</li> <li>• Entregas incrementais</li> <li>• Desenvolvimento diário</li> </ul>

Em suma, a metodologia Xp é a adoção de técnicas através da filtragem dos valores *core* da metodologia (comunicação, simplicidade, feedback, coragem, respeito) e dos seus princípios (humanidade, económicos, mútuos benefícios, auto semelhança, melhoria, diversidade,

Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

reflecção, desenho, oportunidade, redundância, falhas, qualidade, paços de bebé, aceitação das responsabilidades).

#### 2.2.4. Lean

Em 1950, surgiu a filosofia de gestão de *Lean*, tendo sido a ‘Toyota’ a primeira a aplicar essa filosofia.

Na década de 90, a Toyota já estava num patamar bem elevado no que à gestão dos seus negócios dizia respeito e, por isso, impulsionou outras empresas a adotarem a mesma filosofia. *Lean* normalmente é considerada como uma metodologia de negócio, que resulta do aumento do crescimento rentável, do ritmo de inovação acelerado e do aumento da satisfação dos clientes. (Teixeira, 2013)

Esta metodologia foca-se na melhoria da qualidade dos produtos, reduzindo os custos e, dessa forma, melhorando a capacidade de resposta aos clientes.

Saleem (Sall,2008) definiu a filosofia de *Lean* como sendo uma filosofia de produção que assenta na minimização do montante de todos os recursos (incluindo o tempo) utilizados nas várias atividades da empresa.

Muitas empresas utilizam *Lean* como a filosofia principal para os seus negócios como, por exemplo: FEDEX, MICROSOFT, DELL, WIPRO, BOEING, etc.

Segundo Poppendieck (Poppendieck, 99) existem sete princípios do *Lean*:

- 1) Eliminar desperdício;
- 2) Ampliar a aprendizagem;
- 3) Atrasar o compromisso: adiar as suas decisões pelo maior tempo possível, para que se possa dar uma resposta com base em factos;
- 4) Entregar rapidamente;
- 5) Dar poder à equipa;
- 6) Construção com Integridade;
- 7) Visão completa: o produto deve ser visto como um todo, não apenas em parte ou como um subsistema.

O método *kanban* é derivado dos princípios de pensamento *Lean*.

*Kanban* em *Lean manufacturing* é um método para controlo de *stock* e reabastecimento. Este processo de inventário “*just-in-time*” foi originariamente adotado nas mercearias quando as prateleiras eram reabastecidas com base nas lacunas existentes nas mesmas e não no inventário

## Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

dos fornecedores. Inspirado por este sistema de inventário “*just-in-time*”, Taiichi Ohno desenvolveu *Kanban*, que acabou por ser aplicado na principal fábrica da Toyota em 1953.

*Kanban* é hoje utilizado não apenas no ambiente fabril como na década de 50, mas também em muitas outras realidades. Nos projetos de *software*, o *Kanban* é utilizado através de um quadro físico que promove a visualização do fluxo de trabalho de forma acessível a todos. Este quadro físico é constituído por colunas que representam as diversas fases do trabalho e onde são colocadas indicações do desenvolvimento das mesmas. O mais simplista dos quadros poderia ser constituído por apenas três colunas (ex.: por fazer, em execução, feito), mas todos os quadros são adaptáveis a cada situação concreta, dependendo das necessidades de cada equipa. O método *Kanban* é utilizado e aplicável em muitas configurações e permite um fluxo contínuo de trabalho e acrescenta valor para o consumidor. O método *Kanban* é menos prescritivo do que algumas abordagens ágeis e, portanto, mais fiável para começar.

O método de *Kanban* é, em suma, uma estrutura holística para mudanças incrementais, evolutivas e de sistemas para organizações, uma vez que através deste método, se pode mover o trabalho ao longo de todo o processo, isto é, quando a equipa completa uma tarefa, pode assinalar essa finalização e transferir a atenção para o próximo passo.

O quadro de *Kanban*, como mostra a Figura 5, pode parecer excessivamente simplista no início, mas aqueles que o utilizam percebem logo a sua mais-valia, pois fornece uma visão clara do fluxo de trabalho, contratempos, bloqueadores e *status* geral do projeto.

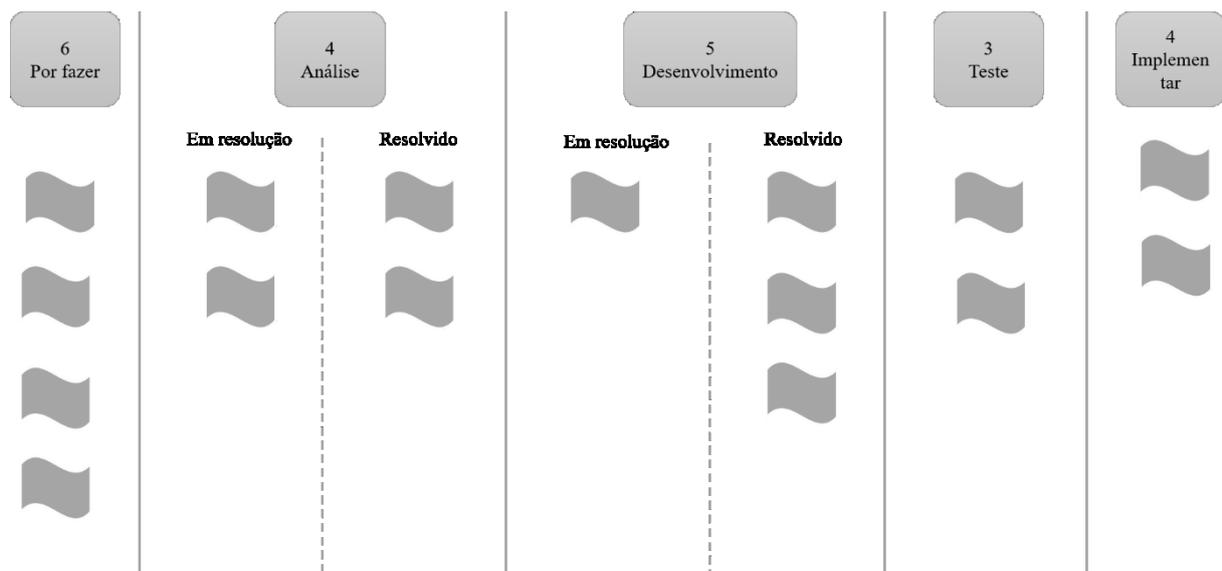


Figura 5 : Tabela Kanban demonstração dos limites do trabalho em progresso, de forma a otimizar o fluxo de trabalho Adaptado de: Agile Practice Guide by Project Management Institute. Fonte: Project Management Institute, (2017)

### 2.2.5. DSDM

*Dynamic Systems Development Method* (DSDM) é uma estrutura de entrega de projetos Agile projetada inicialmente para adicionar mais rigor aos métodos iterativos existentes que eram muito populares na década de 1990. Foi desenvolvido como uma colaboração não comercial entre líderes da indústria. (Pressman, 2005)

O DSDM é conhecido pelo seu foco na entrega direcionada nas condicionantes.

A Figura 6 demonstra o princípio fundamental do DSDM, como estabelece custo, qualidade e tempo em primeiro plano e, em seguida, usa a priorização formalizada do âmbito para atender a essas restrições. Diferente da metodologia Tradicional (*Waterfall*) onde tempo e recursos podem variar, no DSDM tempo e recursos são fixos. (AgileBusiness, 2017)

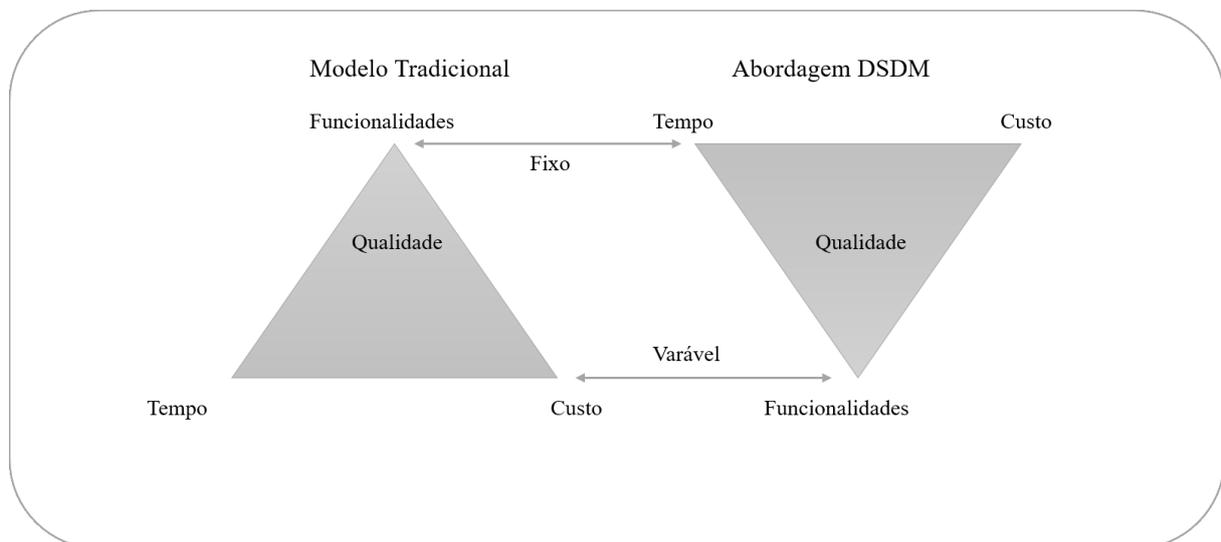


Figura 6 : DSDM aproxima-se do conceito Constraint-Driven Agility Adaptado de: Agile Practice Guide by Project Management Institute. Fonte: Project Management Institute, (2017)

O DSDM consiste em três fases sequenciais (AgileBusiness, 2017):

- **Pré-projecto:** são identificados os projetos candidatos, onde são definidos o orçamento e a assinatura do contrato. Estes critérios devem ser controlados antecipadamente para evitar problemas futuros e em estágios mais críticos.
- **Projeto:** o Estudo de viabilidade e o Estudo de Negócio são subfases sequenciais que se complementam entre si. Após a conclusão destas subfases, o sistema é desenvolvido iterativamente e de forma incremental nos níveis de Análise Funcional, Desenho e Implementação.

Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

- **Pós-Projeto:** Esta fase garante a eficiência e eficácia do projeto. Através de manutenções, melhorias e ajustes de acordo com os princípios do DSDM. A manutenção pode ser vista como um contínuo desenvolvimento. Ao invés de finalizar o ciclo de vida de apenas uma vez, normalmente o projeto pode retomar fases anteriores a fim de refinar ainda mais o passo concluído.

#### **Oito princípios orientam o uso da estrutura DSDM:**

- 1) Foco na necessidade do negócio;
- 2) Entregas dentro do prazo;
- 3) Colaboração;
- 4) Nunca comprometer a qualidade;
- 5) Desenvolver gradualmente soluções firmes;
- 6) Desenvolver iterativamente;
- 7) Comunicar de forma contínua e clara;
- 8) Demonstrar controlo (através de técnicas apropriadas).

### **2.3. Metodologias Iterativas**

Os ciclos de vida iterativos melhoram o produto ou o resultado através de protótipos sucessivos ou provas de conceito.

#### **❖ Características:**

O protótipo produz novos comentários das partes interessadas e insights da equipa. A equipa incorpora a informação fornecida e dessa forma repete uma ou mais atividades do projeto no próximo ciclo. As equipas reúnem informações e, em seguida, reformulam as atividades com base nessas mesmas informações. Desta forma, as iterações ajudam a identificar e reduzir as incertezas no projeto.

Os projetos beneficiam de ciclos de vida iterativos quando a complexidade é alta. Quando o projeto incorre em mudanças frequentes ou quando o âmbito está sujeito a diferentes pontos de vista dos interessados sobre o produto final desejado. Os ciclos de vida iterativos podem levar mais tempo porque são otimizados para aprender em vez de darem ênfase à velocidade de entrega. (Project Management Institute, 2017)

## Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

A Figura 7 ilustra alguns elementos de um ciclo de vida do projeto iterativo para uma entrega de produto único.

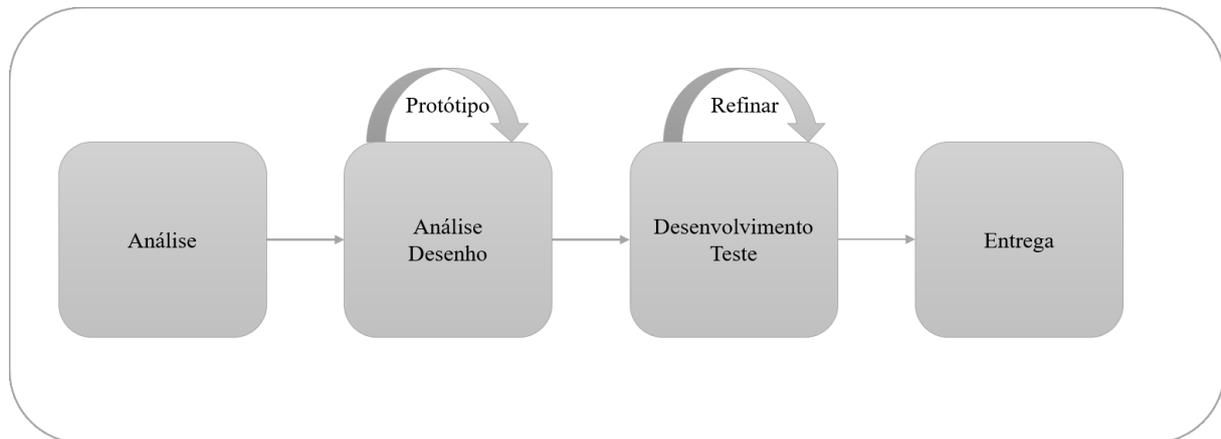


Figura 7 : Ciclo de vida Iterativo Adaptado de: Agile Practice Guide by Project Management Institute. Fonte: Project Management Institute, (2017)

### 2.4. RUP

O RUP foi criado pela *Rational Software Corporation* que mais tarde foi comprada pela IBM, onde passou a ser chamada de IRUP, tornando-se numa marca na área de *software*, que fornece técnicas a serem seguidas pelos membros da equipa de desenvolvimento de software com o objetivo de aumentar a produtividade no processo de desenvolvimento.

É uma metodologia que visa viabilizar o sucesso de grandes projetos de *software*. Sendo uma metodologia de desenvolvimento iterativo, concentra-se na redução de riscos num projeto.

#### ❖ *Características:*

- Desenvolvimento iterativo e incremental, orientado a objetos, com foco na criação de uma arquitetura robusta e análise de risco, segundo Kruchten (KRUCHTEN, 1996).
- É um processo de construção de sistemas de *software* feito em pequenos passos e é apresentado como um modelo mais detalhado.

#### ❖ *Arquitetura do RUP:*

A metodologia RUP identifica cada ciclo de desenvolvimento do projeto em quatro fases, que são (KRUCHTEN 2000): Conceção; Elaboração; Construção; Transição.

Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

O RUP tem também nove disciplinas que servem para organizar logicamente as atividades da engenharia de *software* de acordo com a sua natureza, ou seja, elas descrevem o que deve ser feito em cada iteração a respeito de atividades responsáveis e documentos. Essas disciplinas são chamadas de *workflows*. Dessas disciplinas existem seis que são da engenharia de *software* e três que são de suporte. As disciplinas tal como observamos na Figura 8 são as seguintes: Análise e Design; Implementação; Testes; Implantação; Gestão de Projeto; Gestão de Configuração e Mudanças; Ambiente;

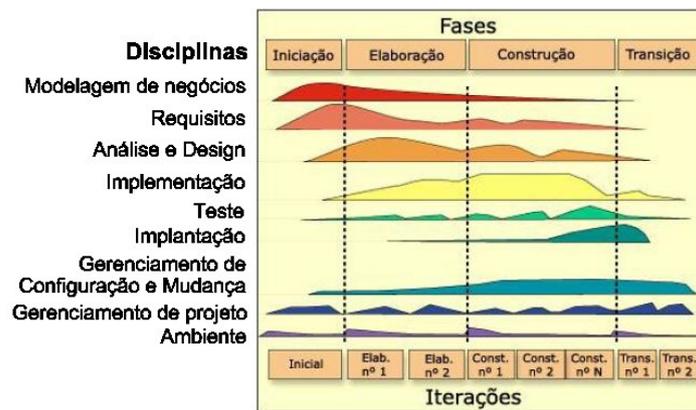


Figura 8 : Arquitetura do RUP Fonte: MARTINS (2007)

## 2.5. Metodologias Tradicionais

As metodologias tradicionais são também chamadas de pesadas ou orientadas a documentação. Estas metodologias foram muito utilizadas no passado num contexto de desenvolvimento de *software* muito diferente do atual. Numa época em que o custo de fazer alterações e correções era muito alto, uma vez que o acesso aos computadores era limitado e não existiam ferramentas modernas de apoio ao desenvolvimento do *software*. Por isso o *software* era todo planeado e documentado antes de ser implementado. Uma das metodologias tradicionais mais utilizadas até hoje é o modelo Clássico/Cascata/*Waterfall* (SOARES, 2004).

### 2.5.1. Waterfall

O modelo *Waterfall*, até aos meados dos anos 80, foi o único modelo com uma grande aceitação por partes dos *developers*.

## Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

Este modelo propõe uma abordagem sistemática, linear e sequencial no que tange ao desenvolvimento de *software*.

*Waterfall*, muitas vezes referido como SDLC é um modelo de metodologia de gestão de projetos com uma abordagem muito simples que valoriza o planeamento sólido, fazendo-o uma única vez e de forma correta, ao contrário da abordagem Agile com a entrega incremental e iterativa.

O gestor do projeto tende a ser o grande responsável, e o trabalho é planeado, exhaustivamente, em primeiro lugar e de seguida é executado aderindo aos requisitos para entrega do projeto num único ciclo, geralmente muito longo.

A Figura 9 indica o desenho das cinco atividades do ciclo de vida do modelo *Waterfall* que atualmente é a metodologia tradicional a mais utilizada.

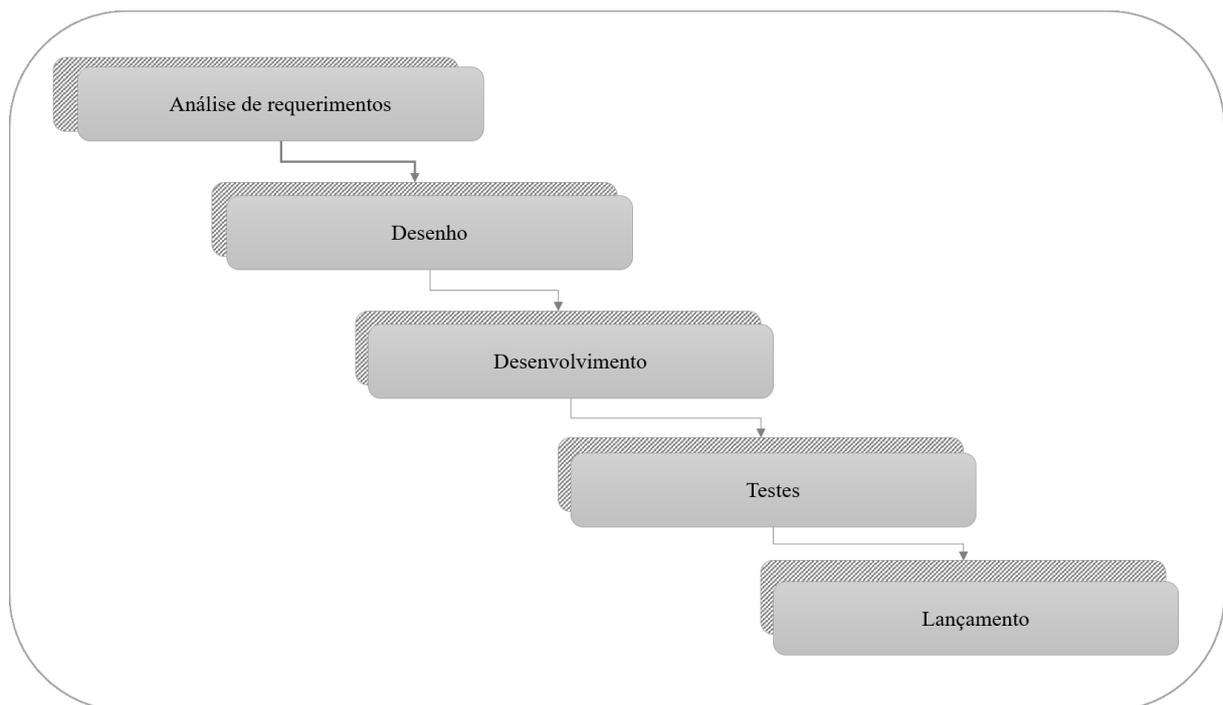


Figura 9 : Ciclo de Vida Waterfall Adaptado de: Agile Practice Guide by Project Management Institute. Fonte: Project Management Institute, (2017)

No modelo *Waterfall*, cada fase deve ser completada antes da próxima fase começar e não há sobreposição nas fases. Normalmente, numa abordagem *Waterfall*, o resultado de uma fase atua como entrada para a próxima fase.

Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

- Os requisitos são definidos na íntegra no início, antes de qualquer trabalho começar.
- O desenho lógico do produto, visa criar uma solução de um modo independente da tecnologia
- Após a realização do desenho da solução, o mesmo vai ser traduzido em linguagem própria da máquina que consiste na fase de desenvolvimento.

Após a codificação, são realizados os testes. Os testes concentram-se na lógica interna do *software* e nas funções externas. Estes testes são realizados para garantir que os requisitos do *software* foram atendidos. Após estes testes, o *software* é entregue ao cliente.

Após o produto ser entregue, instalado e colocado em operação para que possa ser utilizado, podem ser detetados alguns erros que não foram encontrados ou que passaram despercebidos durante a fase de teste, também pode haver algumas alterações que o cliente pretende fazer no produto

Depois de aprovado o plano, há pouco espaço para adaptar o plano, a menos que seja absolutamente necessário. As mudanças que são necessárias, normalmente, requerem solicitações de mudança. O projeto, então, passa pelo processo de requisitos, por meio de *design*, implementação, teste e manutenção.

### **2.5.2. Agile vs Waterfall**

O método *Waterfall* e Agile são duas abordagens distintas para o desenvolvimento de *software* e, segundo Harden (2011), cada uma tem as suas vantagens e desvantagens e a sua escolha deverá ser baseada no tipo de projeto a desenvolver. Os parâmetros de comparação estão presentes no **Anexo A** e permitem perceber onde são comparadas essas metodologias, e quais as principais diferenças.

De seguida na Tabela 2 serão apresentados os pontos-chave da metodologia *Waterfall* e a metodologia Agile.

## Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

Tabela 2 : Comparação *Waterfall* vs *Agile*. Adaptação de Harden 2011

	<i>Waterfall</i>	<i>Agile</i>
<b><i>Entregas</i></b>	Com uma sequência de entregas e um caminho bem definido	Baseada em pequenos ciclos de entregas
<b><i>Estimativas</i></b>	As estimativas são feitas baseadas no trabalho necessário para satisfazer os requisitos	As estimativas são baseadas no trabalho que a equipa consegue alcançar num determinado período de tempo
<b><i>Requisitos</i></b>	Requer requisitos completamente definidos no início	Os requisitos evoluem ao longo do projeto
<b><i>Medição do sucesso</i></b>	O sucesso é medido pela organização	O sucesso é medido pela qualidade do projeto entregue

No entanto, tendo em consideração tudo o que foi exposto anteriormente e como explica Harden (2011), a metodologia *Agile* surgiu como uma alternativa às limitações identificadas no método *Waterfall*, como por exemplo a sua inflexibilidade e rigidez.

## 2.6. Fatores críticos de sucesso e insucesso na gestão de projetos de TIC

Segundo Miguel (2013), os fatores críticos de sucesso são como uma metodologia de identificação dos fatores de negócio críticos que deverão ter a prioridade mais elevada no desenvolvimento de sistemas com o objetivo de atingir o sucesso. É o que é necessário para satisfazer os entregáveis que o cliente quer. As organizações consideradas excelentes na gestão de projetos medem o sucesso, quer interna e externamente, recorrendo a fatores críticos de sucesso.

### 2.6.1. The Standish Group – CHAOS Report

The Standish Group foi criado em 1985 com o objetivo de recolher informações sobre o sucesso e insucesso de projetos em ambientes de TI. Esta recolha de informações visa o desenvolvimento de análises, mais concretamente relatórios, cujas conclusões potenciam a adoção de determinadas vias/comportamentos que, por sua vez, potenciam o sucesso dos projetos e que, ao mesmo tempo, previnem as causas mais comuns de falhas.

Ora, estes relatórios, entretanto denominados CHAOS, passaram a ser um ponto de referência da indústria de desenvolvimento de *software*.

A primeira grande recolha de informação realizada em 1994, através do envio de e-mails com questionários, resultou na avaliação de mais de 8.000 projetos de desenvolvimento de software. Nos dias de hoje, a metodologia adotada pela empresa para recolha de informações passa pela utilização de uma base de profissionais e executivos de empresas que recebem uma quantia monetária por cada questionário respondido e que conta, atualmente, com aproximadamente 3.000 membros ativos.

Os requisitos para que esses membros possam responder aos questionários são os seguintes:

- Devem disponibilizar acesso a alguns dados dos projetos;
- Já devem ter aplicações em operação;
- Devem possuir plataformas particulares em operação.

## Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

Ao longo dos anos, foram publicados diversos relatórios que envolveram a análise de mais de 50 mil projetos de todo o mundo e onde foram abordadas questões que variaram desde as pequenas possíveis melhorias até às implementações maciças de reengenharia de sistemas.

No estudo presente em The Standish Group. (2015). foram usadas as seguintes definições:

- **Sucesso:** o projeto é considerado um sucesso quando a solução é entregue e a empresa reconheceu os critérios de sucesso dentro de um intervalo aceitável.
- **Alterado:** o projeto é considerado como alterado se a solução foi entregue mas a equipa não cumpriu com todos os requisitos do projeto (qualidade está boa, o projeto ultrapassou o tempo estipulado e o ROI é muito baixo).
- **Falhou:** a equipa de projeto não entregou a solução.

Os resultados da Tabela 3 indicam que ainda há trabalho a ser feito para obtenção de melhores resultados nos projetos de desenvolvimento de software:

Tabela 3 : Modern resolution for all projects Fonte: The Standish Group (2015)

	<b>2011</b>	<b>2012</b>	<b>2013</b>	<b>2014</b>	<b>2015</b>
<i>Sucesso</i>	29%	27%	31%	28%	29%
<i>Sofreu Alterações</i>	49%	56%	50%	55%	52%
<i>Falhou</i>	22%	17%	19%	17%	19%

Os relatórios têm tentado demonstrar a influência que tem a dimensão de um projeto no seu sucesso.

Tabela 4 : Chaos resolution by project size Fonte: The Standish Group (2015)

<b>Dimensão</b>	<b>Sucesso</b>	<b>Sofreu Alterações</b>	<b>Falhou</b>
<i>Grande</i>	2%	7%	17%
<i>Largo</i>	6%	17%	24%
<i>Médio</i>	9%	28%	31%
<i>Moderado</i>	21%	32%	17%
<i>Pequeno</i>	62%	16%	11%
<i>Total:</i>	100%	100%	100%

Ora, atentemos nos resultados do relatório Chaos de 2015 presentes na Tabela 4, pode concluir-se que os projetos de menor dimensão têm uma maior probabilidade de sucesso quando comparados com os de maior dimensão, como se pode verificar na tabela anterior.

## Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

No que diz respeito ao cruzamento das metodologias utilizadas, dimensão do projeto e o respetivo sucesso os resultados foram os seguintes:

Tabela 5 : Chaos resolution by Agile vs Waterfall Fonte: The Standish Group (2015)

<i>Dimensão</i>	<i>Método</i>	<i>Sucesso</i>	<i>Sofreu Alterações</i>	<i>Falhou</i>
<i>Largo</i>	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
<i>Médio</i>	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
<i>Pequeno</i>	Agile	58%	38%	4%
	Waterfall	44%	45%	11%
<i>Todas as dimensões</i>	Agile	39%	52%	9%
	Waterfall	11%	60%	29%

Os resultados apresentados na Tabela 5 permitem concluir que o uso de metodologias tradicionais (*waterfall*) no desenvolvimento de *software* de projetos de TI proporcionam uma maior probabilidade de falha do que o uso de metodologias Ágeis, independentemente da dimensão do projeto.

Uma parte fundamental da análise feita pelo *The Standish Group* nos últimos 21 anos é a identificação e classificação constante dos fatores que, em conjunto, permitem tornar os projetos bem-sucedidos. Entre eles:

- **Apoio da gestão superior:** quando um executivo ou grupo de executivos concorda em fornecer apoio financeiro e emocional. O executivo ou os executivos incentivarão e ajudarão na conclusão bem-sucedida do projeto.
- **Maturidade Emocional:** é o conjunto de comportamentos básicos que as pessoas adotam quando trabalham juntas. Em qualquer grupo, organização ou empresa, o nível de maturidade emocional é determinado pela capacidade de lidar com situações menos agradáveis.
- **Envolvência do cliente:** ocorre quando os clientes estão envolvidos no processo de tomada de decisão e no processo de recolha de informações para o projeto. Isso também inclui comentários dos clientes, revisão de requisitos, pesquisa básica, prototipagem e outras ferramentas de construção de consenso.
- **Otimização:** é um meio estruturado que melhora a eficácia do negócio. A otimização começa com a capacidade de gerir os recursos com base no valor comercial relativo.

## Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

- **Qualificação da equipa:** são pessoas que entendem o negócio e a tecnologia em questão. Uma equipa qualificada é altamente proficiente na execução dos requisitos do projeto e entrega do projeto ou produto.
- **Infraestruturas e ferramentas de apoio:** é o Ambiente Padrão de Gestão Arquitetónica: O Grupo *Standish* define o mesmo como um grupo consistente de práticas, serviços e produtos integrados para desenvolver, implementar e gerir aplicativos de *software*.
- **Processos Agile:** significa que a equipa Agile e o proprietário do produto são habilidosos no processo Agile. A proficiência Agile é a diferença entre bons resultados ágeis e os maus resultados ágeis.
- **Automatização das tarefas:** é ter um processo com poucas peças móveis, e essas peças serem automatizadas e simplificadas. A execução modesta também significa o uso de ferramentas de gestão de projetos com moderação e apenas alguns recursos.
- **Experiência do gestor de projetos:** é a aplicação do conhecimento, das habilidades e das técnicas para atividades do projeto, com vista a atender ou exceder as expectativas dos interessados e produzir valor para a organização.
- **Objetivos claros e realistas:** é o consenso de todas as partes interessadas e participantes no objetivo comercial para a execução do projeto. Objetivos de negócios claros também podem significar que o projeto está alinhado com os objetivos e a estratégia da organização.

Em termos da importância que têm, os fatores definidos para o sucesso dos projetos são os seguintes:

Tabela 6 : *Chaos factors of success* Fonte: *The Standish Group (2015)*

<b>Fator de sucesso</b>	<b>Percentagem</b>
<i>Apoio da gestão superior</i>	15%
<i>Maturidade Emocional</i>	15%
<i>Envolvência do Cliente</i>	15%
<i>Otimização</i>	15%
<i>Qualificação da equipa</i>	10%
<i>Infraestrutura e ferramentas de apoio</i>	8%
<i>Processos Agile</i>	7%
<i>Automatização das tarefas</i>	6%
<i>Experiencia do Gestor de projetos</i>	5%
<i>Objetivos claros e realistas</i>	4%

## 2.6.2. Outros estudos

- **Arthur Ahimbisibwe *et al.***

Ahimbisibwe *et al.* (2015) argumentam que as pesquisas realizadas em gestão de projetos têm contribuído, principalmente, para identificar os fatores críticos de sucesso que influenciam o sucesso ou o fracasso dos projetos de *software*. Porém admitem que é claro que não existe um amplo consenso sobre estes fatores críticos. Mais, para estes autores, os esforços destas pesquisas têm incidido, essencialmente, sobre os resultados de projetos de desenvolvimento de *software*, e não no processo de desenvolvimento de software em si.

Por estas razões, Ahimbisibwe *et al.* (2015) seleccionaram 148 estudos que envolviam pesquisas sobre projetos de *software* de diferentes dimensões, em vários domínios, em diferentes países e que utilizavam metodologias tradicionais ou metodologias Agile e tornaram possível a identificação inicial de 37 fatores críticos de sucesso.

Tabela 7 : CSF categories and rankings based on methodologies Fonte: Ahimbisibwe *et al.* (2015)

<i>Fatores de sucesso</i>	<i>Total citações na literatura (n = 148)</i>		<i>Metodologias Agile (n=43)</i>		<i>Metodologias Tradicionais (n=45)</i>	
	<b>Freq.</b>	<b>%</b>	<b>Freq.</b>	<b>%</b>	<b>Freq.</b>	<b>%</b>
<i>Apoio da gestão superior</i>	104	70.2	33	76.7	71	67.6
<i>Envolvência do Cliente</i>	102	68.9	38	88.4	64	61.0
<i>Envolvência e motivação da equipa de projeto</i>	98	66.2	35	81.4	63	60.0
<i>Cultura organizacional</i>	96	64.8	39	90.7	57	54.3
<i>Nível de planeamento do projeto</i>	93	62.8	6	14.0	87	82.9
<i>Liderança</i>	92	62.2	34	79.1	58	55.2
<i>Missão e Visão</i>	90	60.8	6	14.0	84	80.0
<i>Monitorização e controlo do projeto</i>	88	59.5	4	9.3	84	80.0
<i>Conhecimentos técnicos na gestão das mudanças</i>	87	58.7	36	83.7	51	48.6
<i>Comunicação no projeto</i>	85	57.4	37	86.0	48	45.7
<i>Suporte ao usuário</i>	84	56.8	29	67.4	55	52.4
<i>Incertezas tecnológicas</i>	82	55.4	41	95.3	41	39.0

Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

<i>Metodologia de desenvolvimento</i>	81	54.7	31	72.1	50	47.6
<i>Complexidade do projeto</i>	79	53.4	38	88.4	41	39.0
<i>Capacidade e competência da equipa</i>	78	52.7	35	81.4	43	41.0
<i>Composição da equipa de projeto</i>	78	52.7	36	83.7	42	40.0
<i>Formação do cliente</i>	78	52.7	32	74.4	46	43.8
<i>Experiência do cliente</i>	78	52.7	36	83.7	42	40.0
<i>Experiência da equipa de projeto com a tarefa atribuída</i>	77	52.0	31	72.1	46	43.8
<i>Experiência geral da equipa de projeto</i>	77	52.0	8	18.6	69	65.7
<i>Falta de experiência técnica da equipa de desenvolvimento</i>	75	50.6	28	65.1	47	44.8
<i>Urgência</i>	73	49.3	24	55.8	49	46.7
<i>Dimensão da equipa de projeto</i>	73	49.2	27	62.8	46	43.8
<i>Alterações nos requerimentos</i>	71	47.7	34	79.1	37	35.2
<i>Experiência da equipa de projeto com o SDM</i>	69	46.6	9	20.9	60	57.1
<i>Criticidade do projeto</i>	68	45.9	11	25.6	57	54.3
<i>Falta de experiência do usuário</i>	67	45.2	31	72.1	36	34.3
<i>Requerimentos e especificações</i>	65	43.9	36	83.7	29	27.6
<i>Bom trabalho dos vendedores</i>	48	32.4	4	9.3	44	41.9
<i>Infraestrutura e ferramentas de apoio</i>	45	30.4	27	62.8	18	17.1
<i>Cronograma realista</i>	43	29.1	4	9.3	39	37.1
<i>Número de recursos suficientes</i>	39	26.4	8	18.6	31	29.5
<i>Riscos identificados e gerenciados</i>	37	25.0	22	51.2	15	14.3
<i>Orçamento adequado</i>	35	23.6	6	14.0	29	27.6
<i>Boa qualidade de gestão</i>	32	21.6	13	30.2	19	18.1
<i>Processo de reports atualizado</i>	29	19.6	9	20.9	20	19.0
<i>Regras e responsabilidades claras e atribuídas</i>	15	10.1	7	16.3	8	7.6

Posteriormente e após uma análise minuciosa desses fatores, Ahimbisibwe *et al.* (2015) retiraram os fatores entendidos como “menos influentes”, restando, assim, 27 fatores críticos

Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

de sucesso. No que diz respeito à categorização desses mesmos fatores, Ahimbisibwe *et al.* (2015) salientam a inexistência de um amplo consenso entre os pesquisadores e profissionais, sugerindo uma estrutura alternativa com quatro temas-chave: fatores organizacionais, fatores de equipa, fatores de clientes e fatores de projetos.

Tabela 8 : CSF categories and rankings based on methodologies Fonte :Ahimbisibwe et al. (2015)

<i>Categoria</i>	<i>Fator. Crítico</i>	<i>No Geral</i>	<i>Projetos Agile</i>	<i>Projetos Tradicionais</i>
<i>Cultura Organizacional</i>	Apoio da gestão superior	1	13	4
	Cultura organizacional	4	2	10
	Nível de planeamento do projeto	5	25	1
	Liderança	6	12	9
	Missão e Visão	7	27	2
	Monitorização e controlo do projeto	8	26	3
<i>Fatores internos</i>	Conhecimentos técnicos na gestão das mudanças	9	6	13
	Envolvência e motivação da equipa de projeto	3	10	7
	Comunicação no projeto	10	5	16
	Capacidade e competência da equipa	15	11	21
	Composição da equipa de projeto	16	7	22
	Experiencia da equipa de projeto com a tarefa atribuída	19	16	19
	Experiencia geral da equipa de projeto	20	24	5
	Falta de experiencia técnica da equipa de desenvolvimento	21	19	17
	Experiencia da equipa de projeto com o SDM	25	23	8
	<i>Fatores externos</i>	Envolvência do Cliente	2	3
Suporte ao usuário		11	18	12
Formação do cliente		17	14	18
Experiencia do cliente		18	8	23
Falta de experiência do Cliente		27	17	27
<i>Fatores do Projeto</i>	Incertezas tecnológicas	12	1	24
	Metodologia de desenvolvimento	13	15	14
	Complexidade do projeto	14	4	25
	Urgência	22	21	15
	Dimensão da equipa de projeto	23	20	20
	Alterações nos requerimentos	24	9	26
	Criticidade do projeto	26	22	11

Onde leva a concluir que nesse agrupamento o grupo de Fatores com maior referência é os Fatores do Projeto.

- **Joyce Fortune & Diana White**

Fortune e White (2006) apresentaram uma revisão bibliográfica abrangente, na qual selecionaram os principais fatores críticos de sucesso em projetos. A conclusão deveu-se ao facto de estes serem os mais citados numa pesquisa que envolveu cerca de 63 artigos. Relacionaram os fatores de acordo com a frequência que os mesmos aparecem nos artigos, conforme listado na Tabela 9:

Tabela 9 : Critical success factors identified across 63 publications Fonte: Fortune e White (2006)

<b>Fator Crítico</b>	<b>Contagem de citações</b>
<i>Apoio da gerência sénior</i>	39
<i>Objetivos claros e realistas</i>	31
<i>Planeamento firme e detalhado mantido atualizado</i>	29
<i>Boa comunicação e feedback</i>	27
<i>Envolvimento do cliente/usuário</i>	24
<i>Equipa qualificada / suficiente staff</i>	20
<i>Gestão da mudança eficaz</i>	19
<i>Gerente de projeto competente</i>	19
<i>Boa base em projetos</i>	16
<i>Recursos suficientes e bem alocados</i>	16
<i>Boa liderança</i>	15
<i>Tecnologia comprovada e familiar</i>	14
<i>Cronograma realista</i>	14
<i>Riscos identificados e gerenciados</i>	13
<i>Patrocinador de projeto</i>	12
<i>Controlo/monitorização eficaz</i>	12
<i>Orçamento adequado</i>	11
<i>Cultura organizacional</i>	10
<i>Bom desempenho de consultores externos</i>	10
<i>Planeamento para encerramento/revisão</i>	9
<i>Disponibilização de formação</i>	7
<i>Estabilidade política</i>	6
<i>Seleção adequada e experiência com metodologia e ferramentas de gestão de projetos</i>	6
<i>Influências do ambiente</i>	6
<i>Experiência dos recursos</i>	5
<i>Dimensão do projeto/Nível de complexidade/Número de pessoas envolvidas/Duração</i>	4
<i>Diferentes pontos de vista</i>	3

É importante salientar que na análise referida foi identificado que pelo menos um dos três primeiros fatores críticos de sucesso (apoio da gerência sénior, objetivos claros e realistas e

## Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

planeamento firme e detalhado mantido atualizado) apareceu em 87% dos artigos avaliados, enquanto os três juntos apareceram em apenas 17% deles. Isso indica a grande heterogeneidade presente não apenas nos projetos, mas também nas organizações e nos ambientes em que os mesmos estão inseridos.

### 2.6.3. Síntese dos Fatores Críticos de Sucesso

Os autores referidos nos pontos anteriores, referem diferentes fatores críticos de sucesso no desenvolvimento de projetos de software. O cruzamento dos diferentes estudos permite obter com maior probabilidade os fatores críticos de sucesso que maior influência têm no resultado final de um projeto.

Tal como Ahimbisibwe *et al.* (2015), os fatores críticos de sucesso foram agrupados em 4 (quatro) categorias distintas (Cultura Organizacional, Fatores Externos, Fatores Internos e Fatores do projeto) e dentro dessas 4 (quatro) categorias foram limitados os fatores críticos que melhor se adequam aos projetos de *software* realizados em Portugal, tendo-se tido atenção ao número de fatores críticos a avaliar, de forma a ser obtida uma amostra realista.

Tabela 10 : Lista de Fatores Críticos Utilizados no estudo

	<b>Fatores Críticos (Ahimbisibwe et al. (2015))</b>	<b>Fatores Críticos (Fortune &amp; White (2006))</b>	<b>Fatores Críticos (Chaos Report (2015))</b>	<b>Lista final de Factores Críticos</b>
<b>Cultura Organizacional</b>	Apoio da gestão superior	Apoio da gestão superior	Apoio da gestão superior	Apoio da gestão superior
	Cultura organizacional/ Missão e Visão	Cultura organizacional/ Estabilidade política	Processos Agile	
	Nível de planeamento do projeto/ Especificações/ Alteração dos requisitos	Planeamento firme e detalhado mantido atualizado/ Planeamento para encerramento/revisão	Automatização das tarefas	Planeamento
	Boa qualidade de gestão/ Conhecimentos técnicos na gestão das mudanças/ Liderança/ Riscos	Gestor de projeto competente/ Gestão da mudança eficaz/ Boa liderança/ Riscos identificados e gerenciados/	Experiência do gestor de projetos	Competência do gestor de projetos

Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

	identificados e gerenciados	Patrocinador de projeto		
<i>Fatores internos</i>	Envolvência e motivação da equipa de projeto/ Regras e responsabilidades claras e atribuídas/ Experiência da equipa de projeto com a tarefa atribuída/ Bom trabalho dos vendedores	Bom desempenho de consultores externos	Maturidade Emocional	Comprometimento e motivação da equipa do projeto
	Composição da equipa de projeto/ Experiência da equipa de projeto com o SDM/ Falta de experiência técnica da equipa de desenvolvimento/ Experiência geral da equipa de projeto/ Capacidade e competência da equipa	Equipa qualificada / suficiente staff/ Diferentes pontos de vista	Qualificação da equipa	Qualificações da equipa
	Comunicação no projeto	Boa comunicação e feedback		Comunicação no projeto
<i>Fatores externos</i>	Experiência do cliente/ Formação do cliente/ Participação do usuário/ Suporte ao usuário/ Falta de experiência do usuário	Envolvimento do cliente/usuário	Envolvência do cliente	Envolvimento do cliente
<i>Fatore</i>	Cronograma realista/ Criticidade do	Cronograma realista		

Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

projeto/ Urgência			
Infraestrutura e ferramentas de apoio/ Incertezas tecnológicas/ Metodologia de desenvolvimento	Seleção adequada e experiência com metodologia e ferramentas de gestão de projetos/ Formações e certificações/ Tecnologia comprovada e familiar	Infraestruturas e ferramentas de apoio	Infraestrutura e ferramentas de apoio
Orçamento adequado	Orçamento adequado		Orçamento adequado
Requerimentos e especificações	Objetivos claros e realistas	Objetivos claros e realistas	Objetivos claros e realistas
Monitorização e controlo do projeto	Controlo/monotorização eficaz		Controlo / Monitorização
Complexidade do projeto/Dimensão do projeto	Dimensão do projeto/Nível de complexidade/Número de pessoas envolvidas/Duração		Complexidade do projeto
Número de recursos suficientes	Recursos suficientes e bem alocados	Otimização	Número de recursos suficientes

Como podemos observar na Tabela 10, os fatores críticos de sucesso foram alinhados de forma a perceber-se quais os referidos em ambos os estudos, sendo que se escolheram para utilização no presente estudo aqueles com maior correspondência.

**Do cruzamento referido resultaram os seguintes fatores críticos de sucesso:**

- Apoio da gestão superior;
- Planeamento;
- Competência do gestor de projetos;
- Comprometimento e motivação da equipa do projeto;
- Qualificações da equipa;
- Comunicação no projeto;
- Envolvimento do cliente;
- Infraestrutura e ferramentas de apoio;
- Orçamento adequado;

## Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

- Objetivos claros e realistas;
- Controlo / Monitorização;
- Complexidade do projeto;
- Número de recursos suficientes.

### 2.7. Sumário

Neste capítulo, com base no estudo e categorização dos tipos de metodologias de desenvolvimento de software realizado por Forrester (2009), foi feita a análise das diferentes metodologias de desenvolvimento de software mais utilizadas, identificando quais as principais características, vantagens e desvantagens tal como podemos observar no **Anexo B**.

Em relação aos fatores críticos de sucesso, foi feito o levantamento de diferentes estudos como Ahimbisibwe *et al.* (2015), Fortune e White (2006) e The Standish Group. (2015), todos com relevância sobre o tema. Com base nesses estudos, foi realizado um cruzamento que permitiu perceber e elencar quais os fatores críticos de sucesso referidos por todos os estudos.

O estudo que requereu maior análise, foi o realizado pelo The Standish Group. (2015). Este estudo tem em conta o panorama dos projetos de desenvolvimento de software em diferentes países.

Da análise dos diferentes estudos, surgiram várias questões, às quais se tentará responder e que serviram de guia à realização deste trabalho. No final tentar-se-á perceber se os resultados de The Standish Group. (2015) se observam em Portugal, se nesses projetos são aplicadas metodologias de desenvolvimento de software bem como quais os fatores críticos de sucesso que maior impacto tem no resultado de um projeto.

## Capítulo 3 – Metodologia

### 3.1. Desenho de investigação

Segundo Malhotra (2006), o questionário é o principal instrumento na recolha de dados quantitativos primários. Esta técnica possibilita uma rápida recolha dos dados, aumenta a precisão e facilita o processamento dos mesmos. Ao permitir uma recolha *standartizada* dos dados, é garantida a consistência e a coerência interna da análise.

A Figura 10 representa as diferentes fases percorridas pela investigação. De acordo com os objetivos desta investigação e com base nos estudos descritos no estado da arte, desenvolveu-se um questionário que visa responder às hipóteses que serão formuladas seguidamente.

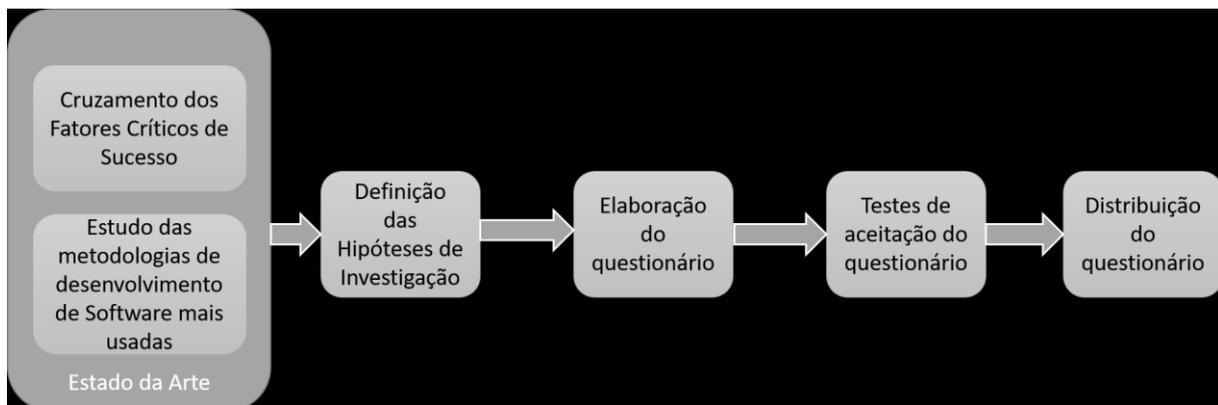


Figura 10 : Fases de Investigação

### 3.2. Hipóteses de investigação

Com base no estudo realizado no Capítulo 2, foram criadas várias hipóteses de investigação que visam:

- 1) Responder a dúvidas que surgiram no estudo de outros autores, nomeadamente:
  - a. Porque razão consideram o projeto realizado como insucesso?
  - b. Que fase do projeto têm maior influência no resultado do mesmo.
  - c. Que metodologias são mais utilizadas
- 2) Comparar os resultados obtidos por outros autores com a realidade existente nos projetos de software em Portugal.

## Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

- a. Influência da dimensão do projeto no sucesso do mesmo;
- b. Influência da dimensão do projeto no sucesso do mesmo quando aplicada determinada metodologia de desenvolvimento de software.
- c. Quais os fatores críticos de sucesso com maior influência no resultado do mesmo.
- d. Quais os fatores críticos de sucesso com maior influência no resultado do mesmo.

### **As hipóteses criadas e que irão ser estudadas nos capítulos seguintes são:**

H1: Que influência tem a dimensão da equipa no sucesso de um projeto?

H2: Que influência tem a dimensão do projeto no sucesso de um projeto?

H3: Quais os fatores críticos de sucesso que têm maior influência num projeto?

H4: Quais as principais razões que são apontadas como insucesso num projeto?

H5: Quais as principais fases com maior relevância para o sucesso/insucesso do projeto?

H6: Quais as metodologias mais utilizadas que resultaram em sucesso/insucesso?

H7: Quais os principais impactos da escolha dessas metodologias?

A elaboração das Hipótese 1, 2 e 3 visa responder à questão P2: Quais são fatores com maior influência no sucesso e insucesso dos projetos de software em Portugal? Formada no ponto 1.2. Questões e objetivos de investigação, enquanto que a Hipótese 6 visa responder à questão P3: Existe uma relação do resultado do projeto com a escolha da metodologia utilizada no desenvolvimento de software?.

### 3.3. Elaboração do questionário

Com o objetivo de responder às hipóteses descritas anteriormente, foi realizado um questionário.

Foi realizada uma fase inicial de pré testes, de modo a compreender se as perguntas eram claras e diretas. Essa validação foi feita por participantes em projetos de desenvolvimento de *software*.

Com a aprovação do questionário por parte da comunidade, foi realizada uma versão final presente no **Anexo C** e que é composto pelas seguintes partes:

- A primeira onde se abordavam algumas características do projeto realizado, bem como qual a classificação atribuída ao projeto em termos de resultado final (sucesso/insucesso/sofreu alterações);
- A segunda onde se referiam quais os principais fatores críticos de sucesso nos projetos (em que a influência dos mesmos era medida a partir da escala do tipo *Likert*., sendo que a escala *Likert* é uma escala de mensuração amplamente utilizada, que permite aos inquiridos indicar o seu grau de concordância ou discordância com cada uma das declarações expostas: 0-4 Não afetou, 5 foi indiferente, 5-10 afetou consideravelmente), fatores críticos esses que foram obtidos pela combinação de diferentes estudos como foi referido no capítulo anterior.
- Por último, a terceira parte composta pela utilização - ou não - de metodologias de desenvolvimento de software, bem como quais os principais impactos na escolha dessa mesma metodologia.

Existem vários sítios na Internet fornecedores de ferramentas para a criação de questionários, de entre essas ferramentas destaca-se a plataforma *Qualtrics*<sup>1</sup>, que acabou por ser a plataforma escolhida para a divulgação deste estudo. Esta plataforma tem uma interface bastante “*user friendly*”, proporcionando assim uma fácil criação e gestão de questionários.

O ISCTE disponibiliza uma versão do *Qualtrics*, com a possibilidade de criar, distribuir, controlar e tratar a informação, com uma variedade bastante grande de funções, como por

---

<sup>1</sup> Sítio institucional: <http://www.qualtrics.com>

Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

exemplo, ter dois inquéritos ativos simultaneamente, exportar relatórios, entre outras (Machado 2011).

### **3.4. Divulgação e Processo de amostragem**

Dado que o que se pretende estudar é os projetos de software, torna-se relevante salientar que a população é constituída por todos os participantes em projetos de software e que para levar a cabo este estudo, recorreu-se a uma amostra que apresenta limitações quanto à generalização dos resultados, uma vez que se trata de uma amostra de conveniência. Deste modo, o método de amostragem não probabilístico e de conveniência foi o escolhido.

A divulgação do questionário foi feita maioritariamente na rede social LinkedIn, devido à facilidade em filtrar a profissão e o sector empregador. Foi também utilizado o método de *snowball* (“Bola de Neve”), sendo pedido a cada um dos participantes a partilha do questionário a profissionais com participação em projetos de software.

De modo obter mais respostas, foi realizado o contacto com APOGEP, que se disponibilizou a partilhar na sua newsletter e nas suas redes sociais como se pode observar no **Anexo D**.

## Capítulo 4 – Análise e discussão dos resultados

Para proceder à análise de dados recorreu-se a várias técnicas estatísticas.

Ora, numa fase inicial, procedeu-se à caracterização da amostra através da utilização de técnicas de análise descritiva.

Os fatores críticos de sucesso foram analisados conjuntamente, recorrendo a técnicas de redução de dimensionalidade – no caso, recorrendo a Análise em Componentes Principais – de forma a identificar as macro dimensões estruturantes, as quais foram utilizadas em análises subsequentes.

As hipóteses de investigação foram estudadas recorrendo quer a testes de hipóteses ajustados aos diferentes casos (testes de  $\chi^2$ , teste exatos de Fisher) quer a ajustamento de modelos logísticos, quer ainda a análises descritivas dos resultados. É de referir que foram aferidos os pressupostos de aplicação dos testes efetuados, e optado por alternativas adequadas aquando da sua não validade. O software utilizado para o tratamento de dados foi o SPSS Statistics (versão 25).

### 4.1. Perfil e dimensão da amostra

O método de divulgação de dados escolhido permitiu obter um total de 151 respondentes, sendo que todos eles têm em comum o mesmo critério de inclusão: ‘Participação em projetos de desenvolvimento de *software* realizados em Portugal’.

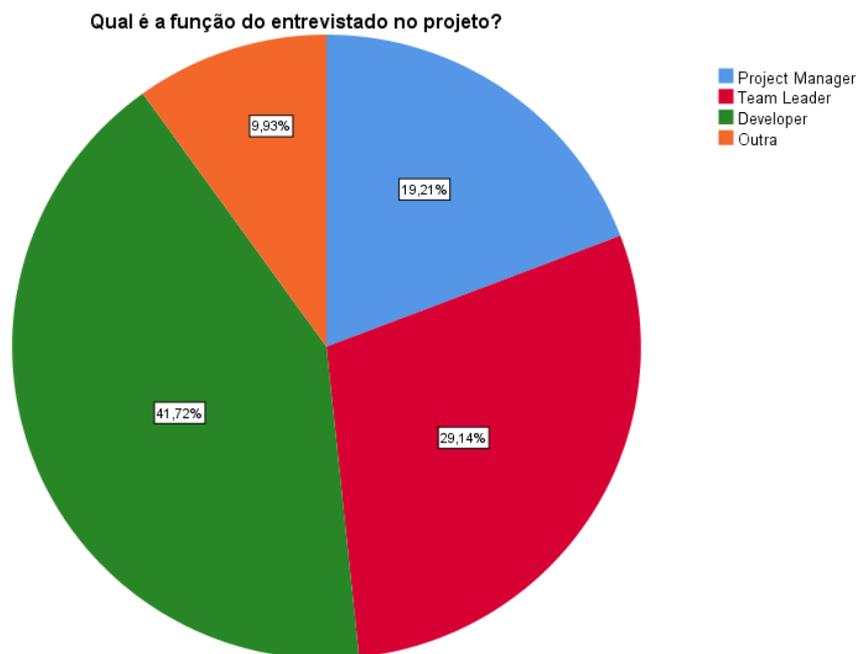


Gráfico 1 : Frequência das funções dos entrevistados nos projetos

## Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

Os 151 participantes estão divididos em: Developers (63), Team Leaders (44), Project Managers (29) e outras funções (15).

Como se observa no Gráfico 1, os participantes com funções de “Developer” representam a maior percentagem dos inquiridos (41,72%). A segunda e a terceira maior fatia pertence aos “Team Leaders” (29,14%) e “Project Managers” (19,21%), o que permite afirmar que quase metade dos inquiridos tinham funções de gestão nos projetos (Team Leader e Project Manager). Por último 9,93% tinham/tiveram outras funções no decorrer dos projetos, entre as quais se destacam as funções de “Gestão de Recursos Humanos”.

### 4.2. Pré-tratamento dos dados

No decorrer da análise dos resultados obtidos pelo questionário, foram analisadas as frequências das respostas para encontrar desvios de frequência e anomalias nas respostas.

Após essa análise inicial conclui-se que seria necessário redefinir algumas variáveis, como será explicada nos capítulos seguintes.

#### 4.2.1. Q2 - Como classifica a dimensão da equipa no projeto?

Na análise dos resultados obtidos em Q2 “Como classifica a dimensão da equipa no projeto?”, apenas foi escolhida a opção “Superior à necessária” em duas ocasiões como se pode observar na Tabela 11, pelo que se entendeu ser preferível criar uma nova variável em que as opções “Adequada” e “Superior à necessária” ficassem ambas contempladas como “Suficiente” e a resposta “Inferior à necessária” passou a ser tida em conta como “Não Suficiente”.

Podemos assim observar na Tabela 12 as frequências da nova variável descrita.

Tabela 11 : Frequência da variável Q2 - Como classifica a dimensão da equipa no projeto?

	<i>Frequência</i>	<i>Percentagem</i>
<i>Inferior à necessária</i>	63	41,7%
<i>Adequada</i>	86	57,0%
<i>Superior à necessária</i>	2	1,3%
<i>Total</i>	151	100,0%

Tabela 12 : Freqüência da nova variável “Dimensao\_Equipa” com base na pergunta Q2

	<b>Freqüência</b>	<b>Percentagem</b>
<b>Não Suficiente</b>	63	41,7%
<b>Suficiente</b>	88	58,3%
<b>Total</b>	151	100,0%

#### 4.2.2. Q5 - Em termos do resultado final como classifica o projeto?

Na análise dos resultados obtidos à questão Q5 do questionário presente no **Anexo C**: “Em termos do resultado final como classifica o projeto?” que era composta por três hipóteses de escolha: “Sucesso”, ”Insucesso” e “Sofreu alterações”, concluiu-se que em face do número reduzido de respostas que referiram “Insucesso” (8 respostas) como podemos observar na Tabela 13, faria mais sentido criar uma nova variável em que as respostas “Sofreu Alterações” e “Insucesso” passassem a estar ambas numa variável de “Não Sucesso”, o que permite uma melhor abordagem na realização dos testes de Hipóteses.

Podemos assim observar na Tabela 14 que os projetos que resultaram em “Sucesso” constituem um total de 86 e os projetos que resultaram em “Não Sucesso” representam um total de 65.

Tabela 13 : Freqüência da variável Q5 - Em termos do resultado final como classifica o projeto

	<b>Freqüência</b>	<b>Percentagem</b>
<b>Sucesso</b>	86	57,0%
<b>Insucesso</b>	8	5,3%
<b>Sofreu alterações</b>	57	37,7%
<b>Total</b>	151	100,0%

Tabela 14 : Freqüência da nova variável “Project\_Result” com base na pergunta Q5

	<b>Freqüência</b>	<b>Percentagem</b>
<b>Não Sucesso</b>	65	43,0%
<b>Sucesso</b>	86	57,0%
<b>Total</b>	151	100,0%

#### **4.2.3. Q8 - Em que medida os seguintes fatores afetaram o resultado final do projeto?**

Na implementação do questionário, por lapso, na questão Q8 “Em que medida os seguintes fatores afetaram o resultado final do projeto?” foi repetida uma das opções “Complexidade do Projeto”. De modo a corrigir tal situação, procedeu-se à medição do grau de correlação entre as duas variáveis, para que se pudesse compreender a relação entre as mesmas.

Como se pode observar nos resultados obtidos na tabela de correlação presente no **Anexo E**, o coeficiente de correlação de *Pearson* ( $r$ ) é próximo de 1 (0,706), o que significa que se nota um aumento no valor de uma variável quando a outra também aumenta, ou seja, há uma relação linear positiva.

Deste modo, procedemos à criação de uma nova variável ‘complexidade composta’, que consiste na média das duas variáveis.

Ainda em relação à questão Q8 “Em que medida os seguintes fatores afetaram o resultado final do projeto?”, devido ao elevado número de variáveis e para comprovar o agrupamento feito por outros autores (cfr. referido no ponto 2.6. Fatores críticos de sucesso e insucesso na gestão de projetos de TIC) foi realizado uma Análise em Componentes Principais de forma a aferir da existência de macro dimensões estruturantes (cfr. **Anexo F**) tendo-se obtido a seguinte matriz de Pesos (*Loadings*):

Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

Tabela 15 : Matriz de Pesos loading, após rotação, numa solução com 4 componentes e correspondente % de var explicada

	<b>Cultura Organizacional</b>	<b>Fatores internos</b>	<b>Fatores do projeto</b>	<b>Fatores externos</b>	<b>%Variância explicada com 4 componentes</b>
<b>Comunicação no projeto</b>	,751			,257	<b>63,6%</b>
<b>Competência do gestor de projetos</b>	,710	,460			<b>71,9%</b>
<b>Apoio da gestão superior</b>	,668		,139		<b>47,6%</b>
<b>Planeamento</b>	,635	,180		,155	<b>46,7%</b>
<b>Infraestrutura e ferramentas de apoio</b>	,159	,826			<b>70,8%</b>
<b>Comprometimento e motivação da equipa do projeto</b>		,666	,364	,330	<b>68,4%</b>
<b>Qualificações da equipa</b>		,647	,393	,370	<b>71,0%</b>
<b>complexidade_composta</b>		,204	,685		<b>52,8%</b>
<b>Número de recursos suficientes</b>	,251	,101	,665		<b>51,8%</b>
<b>Objetivos claros e realistas</b>	,421	-,203	,553	,312	<b>62,2%</b>
<b>Orçamento adequado</b>	,317	,367	,477	-,241	<b>52,1%</b>
<b>Controlo / Monitorização</b>	,101		,169	,767	<b>62,8%</b>
<b>Envolvimento do cliente</b>	,175	,411		,636	<b>60,8%</b>
<b>%Variância explicada</b>	<b>17,946%</b>	<b>16,861%</b>	<b>13,843%</b>	<b>11,539%</b>	<b>60,189%</b>

Como se pode observar na Tabela 15, as variáveis foram agrupadas em 4 componentes o que vai ao encontro do agrupamento definido no Capítulo 2. Foram atribuídas as seguintes designações às componentes : Cultura Organizacional, Fatores Internos, Fatores do Projeto e Fatores Externos, tal como definido no Estado da Arte no ponto 2.6. Fatores críticos de sucesso e insucesso na gestão de projetos de TIC

Essas componentes estão organizadas da seguinte forma:

- **Cultura Organizacional:**

Componente estruturada pelos seguintes FCS: Comunicação no projeto, Competência do gestor de projetos, Apoio da gestão superior, Planeamento.

Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

- **Fatores internos:**

Componente estruturada pelos seguintes FCS: Infraestrutura e ferramentas de apoio, Comprometimento e motivação da equipa do projeto, Qualificações da equipa.

- **Fatores do projeto:**

Componente estruturada pelos seguintes FCS: complexidade\_composta, Número de recursos suficientes, objetivos claros e realistas, Orçamento adequado

- **Fatores externos:**

Componente estruturada pelos seguintes FCS: Controlo / Monitorização, Envolvimento do cliente.

O agrupamento descrito anteriormente, apresenta três fatores críticos de sucesso inseridos em grupos diferentes comparativamente com o levantamento e categorização feita com base em outros autores no ponto 2.6. Fatores críticos de sucesso e insucesso na gestão de projetos de TIC.

- O fator crítico de sucesso “Infraestrutura e ferramentas de apoio” foi classificado como ‘Fatores do Projeto’ enquanto que na Análise em Componentes Principais foi classificado como ‘Fatores Internos’.
- O fator crítico de sucesso “Comunicação no projeto” foi classificado como ‘Fatores Internos’ enquanto que Análise em Componentes Principais foi classificado como ‘Cultura Organizacional’.
- Por fim, o fator crítico de sucesso “Controlo / Monitorização” foi classificado como ‘Fatores do Projeto’ e na Análise em Componentes Principais foi classificado como ‘Fatores Externos’.

Apesar destes três fatores críticos de sucesso terem sido classificados de forma diferente, os resultados obtidos pela Análise em Componentes Principais foram francamente positivos, uma vez que a maioria dos fatores críticos foram agrupados de acordo com o feito no estado da arte. A não existência de uma definição concreta do significado de cada um dos fatores críticos de sucesso, pode levar a que cada pessoa interprete da sua forma o significado dos mesmos, o que poderá justificar as pequenas diferenças existentes.

Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

Nos próximos capítulos será tido em conta a classificação que resultou da Análise em Componentes Principais de modo a facilitar a interpretação e visualização dos resultados obtidos.

#### 4.2.4. Q10 - Foram adotadas algumas metodologias/princípios?

Na pergunta Q10 “Foram adotadas algumas metodologias/princípios?” era dada a possibilidade de escolha de mais do que uma metodologia. Ora, tal possibilidade resultou num total de 181 respostas (superior às 151 do questionário), o que também significa que maioritariamente apenas uma foi escolhida.

Observa-se na Tabela 16 que a maioria dos projetos foram realizados seguindo os princípios de Waterfall e Scrum.

Tabela 16 : Frequência da variável correspondente à pergunta Q10

		<i>Frequência</i>	<i>Percentagem</i>
<i>metodologias</i>	<b>Waterfall</b>	55	30,4%
	<b>Scrum</b>	49	27,1%
	<b>Lean (Kanban)</b>	12	6,6%
	<b>eXtremme Programming (XP)</b>	2	1,1%
	<b>Feature Driven Development (FDD)</b>	4	2,2%
	<b>Dynamic Systems Development Method (DSDM)</b>	2	1,1%
	<b>Prototipagem</b>	11	6,1%
	<b>Outra</b>	5	2,8%
	<b>Nenhuma</b>	41	22,7%
		<b>Total</b>	181

Foi criada uma nova variável: “Tipo de Metodologias”, que permitiu agrupar as metodologias escolhidas na questão 10 do questionário, cfr. representado na Tabela 17.

Com base no estado da arte como indicado no ponto 2.1. Metodologias de desenvolvimento de software, o agrupamento foi feito da seguinte forma:

- Metodologias tradicionais: Waterfall e Prototipagem;
- Metodologias Agile: SCRUM, FDD, XP, Lean. DSDM;
- Agile e Tradicional: uso de ambos os grupos de metodologias;

Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

- Outras metodologias: metodologia não incluída nas opções;
- Não utilização de qualquer tipo de metodologia.

Tabela 17 : Frequência da nova variável “Tipo de Metodologias” com base na pergunta Q10

	<i>Frequência</i>	<i>Porcentagem</i>
<i>Tradicional</i>	47	31,1%
<i>Agile</i>	44	29,1%
<i>Agile e Tradicional</i>	14	9,3%
<i>Outras Metodologias</i>	5	3,3%
<i>Nenhuma Metodologia</i>	41	27,2%
<i>Total</i>	151	100,0%

Como se pode observar na Tabela 17 apesar de idênticas as percentagens, as Metodologias tradicionais obtiveram uma percentagem total de 31,1% seguidas das metodologias Agile com 29,1%.

### 4.3. Teste das hipóteses em estudo

Os critérios de definição dos testes a aplicar para cada uma das hipóteses foram diferentes. Nas seguintes hipóteses:

- H1 - Que influência tem a dimensão da equipa no sucesso de um projeto
- H2 - Que influência tem a dimensão do projeto no sucesso de um projeto
- H5 - Quais as principais fases com maior relevância para o sucesso/insucesso do projeto
- H6 - Quais as metodologias mais utilizadas que resultaram em sucesso/insucesso

, foi realizado o teste qui-quadrado ( $\chi^2$ ) de *Pearson* de modo a perceber a dependência entre as duas variáveis testadas.

Esta análise é feita através da construção de Tabelas de Contingência de dupla entrada e da verificação do nível da significância de prova (*p-value* ou *sig*), calculada pelo teste Qui-Quadrado, considerando um nível de significância de 5%.

Para uma correta interpretação do teste de Qui-quadrado de *Pearson*, é necessário que:

- I. Nenhuma célula tenha frequência esperada inferior a 1
- II. O número de células com frequência esperada  $< 5$  não exceda 20% do número total de células

Alternativamente nas condições acima não estarem comprovadas utilizou-se o teste exato correspondente.

Na hipótese H4 - Quais as principais razões que são apontadas como insucesso num projeto e H7 - Quais os principais impactos da escolha dessas metodologias, foi realizado a análise descritiva das variáveis qualitativas com o objetivo de interpretar as frequências obtidas.

Já no que se refere à hipótese H3 - Quais os fatores críticos de sucesso que têm maior influência num projeto, de modo a compreender qual o grupo de fatores críticos de sucesso que maior influencia tem no resultado do projeto, foi aplicado o modelo de regressão logística e a sua respetiva análise.

#### 4.3.1. Hipótese 1 - Que influência tem a dimensão da equipa no sucesso de um projeto

Partindo da revisão da literatura efetuada e de modo a aferir qual a influência da dimensão da equipa no sucesso de um projeto, foram cruzadas as variáveis “Dimensao\_Equipa” que corresponde à questão Q2 - Como classifica a dimensão da equipa no projeto? e

Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

“Project\_Result” que corresponde a Q5 - Em termos do resultado final como classifica o projeto?.

Para a realização desta análise, procedeu-se a um teste de hipóteses, onde a definição das hipóteses é dada por:

- Hipótese nula, H<sub>0</sub>: A dimensão da equipa de projeto não está relacionada com o sucesso de um projeto (ou a dimensão da equipa é independente do sucesso do projeto).
- Hipótese alternativa, H<sub>a</sub>: A dimensão da equipa de projeto está relacionada com o sucesso de um projeto (ou a dimensão da equipa não é independente do sucesso do projeto).

Aplicado o teste de Fisher, cujo resultado pode ser consultado no **Anexo G**, a significância de prova calculada (sig = 0.000) é inferior a qualquer nível de referência  $\alpha$  usualmente utilizado, permitindo a rejeição da hipótese nula H<sub>0</sub>, pelo que se conclui que a dimensão da equipa de projeto está relacionada com o sucesso de um projeto.

Tabela 18 : Tabela de referência cruzada entre "Dimensao\_Equipa" e "Projec\_Result"

		<i>Não Sucesso</i>	<i>Sucesso</i>	<i>Total</i>
<i>Não Suficiente</i>	<b>Contagem</b>	43	20	63
	<b>% em Coluna</b>	66,2%	23,3%	41,7%
	<b>Suficiente</b>			
<i>Suficiente</i>	<b>Contagem</b>	22	66	88
	<b>% em Coluna</b>	33,8%	76,7%	58,3%
	<b>Total</b>			
<i>Total</i>	<b>Contagem</b>	65	86	151
	<b>% em Coluna</b>	100%	100%	100%

Dos resultados apresentados na Tabela 18, observa-se uma relação relevante entre ter equipa com dimensão não suficiente e projetos em “Não Sucesso”- 66,2% e entre ter equipa com dimensão suficiente e projetos em “Sucesso” - 76,7%.

### 4.3.2. Hipótese 2 - Que influência tem a dimensão do projeto no sucesso de um projeto

De modo a aferir qual a influência da dimensão do projeto no sucesso do mesmo, cruzaram-se as seguintes variáveis Q4 - Qual a dimensão do projeto? (presente no **Anexo C**) e "Project\_Result" que corresponde a Q5 - Em termos do resultado final como classifica o projeto?

Para a realização desta análise, procedeu-se a um teste de hipóteses, onde a definição das hipóteses é dada por:

- Hipótese nula, H0: A dimensão do projeto não está relacionada com o sucesso de um projeto (ou a dimensão do projeto é independente do sucesso do projeto).
- Hipótese alternativa, Ha: A dimensão do projeto está relacionada com o sucesso de um projeto (ou a dimensão do projeto não é independente do sucesso do projeto).

Aplicado o teste de Fisher, como se pode verificar no **Anexo G**, a significância de prova calculada (sig = 0,610) é superior a qualquer nível de referência  $\alpha$  usualmente utilizado, não permitindo a rejeição da hipótese nula H0, pelo que é de assumir que a dimensão do projeto não esteja relacionada com o sucesso de um projeto.

Tabela 19 : Tabela de referência cruzada entre " Qual a dimensao do projecto" e "Project\_Result"

		<i>Não Sucesso</i>	<i>Sucesso</i>	<i>Total</i>
<i>Pequeno</i>	<b>Contagem</b>	13	13	26
	<b>% em Coluna</b>	20%	23,3%	41,7%
	<b>% em Linha</b>	50,0%	50,0%	
<i>Médio</i>	<b>Contagem</b>	27	42	69
	<b>% em Coluna</b>	41,5%	48,8%	45,7%
	<b>% em Linha</b>	39,1%	60,9%	
<i>Grande</i>	<b>Contagem</b>	25	31	56
	<b>% em Coluna</b>	38,5%	36,0%	37,1%
	<b>% em Linha</b>	44,6%	55,4%	
<i>Total</i>	<b>Contagem</b>	65	86	151
	<b>% em Coluna</b>	100%	100%	100%

Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

Da Tabela 19 e tal como era expectável, percebe-se que é pouco relevante a influência da dimensão do projeto no sucesso do mesmo, uma vez que apenas nos projetos de dimensão média existe uma maior diferença, observando-se um total de 42 projetos que resultaram em ‘Sucesso’ contra 27 que resultaram em ‘Não Sucesso’, enquanto que no caso do projeto com dimensão pequena ou grande, os casos de sucesso chegaram a um valor de 50% e 58% respetivamente.

### 4.3.3. Hipótese 3 - Quais os fatores críticos de sucesso que têm maior influência num projeto

No que tange à hipótese 3, o principal objetivo da mesma consiste em perceber quais os fatores críticos de sucesso que têm maior influência no resultado de um projeto.

Para tal, foi utilizado o Método de Regressão Logística (Logit), o qual estima, a partir de um conjunto de variáveis independentes, a probabilidade de ocorrer um determinado evento, neste caso, o Sucesso/Não sucesso dos projetos.

As variáveis independentes utilizadas foram as variáveis que resultaram da ‘Análise de Componentes Principais’ descritas no ponto 4.2.3. Q8 - Em que medida os seguintes fatores afetaram o resultado final do projeto?, ou seja, os Fatores Internos, Fatores Externos, Cultura Organizacional e Fatores do Projeto.

Existem testes estatísticos para analisar o significado do modelo final, nomeadamente o teste Qui-quadrado (tabela Variáveis na equação) e o teste Hosmer e Lemeshow presente na Tabela 21. Estas duas medidas, combinadas, fornecem suporte para que se aceite o modelo de regressão logística como significativo. São estes testes que asseguram a evidência de significância estatística das variáveis.

Na Tabela 20, estão presentes as seguintes colunas:

- “-2 Log da Verosimilhança” que indica a precisão da variável desfecho;
- “R quadrado de Nagelkerke” que indica a aderência do modelo, ou seja, o quanto o modelo explica a ocorrência do desfecho.

Tabela 20 : Resumo do modelo

<i>-2 Log da Verosimilhança</i>	<i>R quadrado Cox &amp; Snell</i>	<i>R quadrado Nagelkerke</i>
196,020 <sup>a</sup>	,066	,089

Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

Tabela 21 : Teste de Hosmer e Lemeshow

<i>Qui-quadrado</i>	<i>gl</i>	<i>Sig.</i>
10,589	8	,226

Analisando os resultados obtidos pela Tabela 20e os resultados presentes em Nagelkerke R Square conclui-se que a aderência do modelo é significativa, ou seja, o modelo obtido encontra-se ajustado. Aceita-se, assim, este modelo de Regressão Logístico, como significativo. Embora caso a capacidade explicativa do modelo seja baixa, no contexto tal apenas significa que o desfecho de um projeto depende de outros fatores que não foram objeto de análise deste trabalho.

A Tabela 22 apresenta vários parâmetros presentes na equação de regressão logística, onde a coluna ‘B’ representa a constante da equação, a coluna ‘E.P.’ o erro padrão associado à constante, a coluna de estatística ‘Wald’ identifica o quanto cada variável independente participa, individualmente, para a explicação da variável dependente e em Sig. o valor de “p” associado a estatística Wald., e a coluna ‘Exp(B)’ a variação da oportunidade do desfecho ocorrer de facto.

Tabela 22 : Variáveis na Equação

	<i>B</i>	<i>E.P.</i>	<i>Wald</i>	<i>gl</i>	<i>Sig.</i>	<i>Exp(B)</i>	<i>95% C.I. para EXP(B)</i>	
							<i>Inf.</i>	<i>Sup.</i>
<i>Cultura Organizacional</i>	-,112	,174	,419	1	,518	,894	,636	1,256
<i>Fatores internos</i>	,477	,180	6,994	1	<b>,008</b>	1,611	1,131	2,294
<i>Fatores do projeto</i>	,090	,169	,283	1	,595	1,094	,786	1,523
<i>Fatores externos</i>	,247	,174	2,025	1	,155	1,280	,911	1,799
<i>Constante</i>	,290	,170	2,908	1	,088	1,337		

Da análise da Tabela 22, observa-se pelo teste Wald que a significância do parâmetro ‘Fatores Internos’ destaca-se dos outros parâmetros. Para além do teste Wald, podemos observar que o coeficiente estimado é significativamente não nulo ( $\text{sig}=0,008 < 0,05=\alpha$ ), pelo que se conclui que entre os fatores analisados, os ‘Fatores Internos’ é o único com influência significativa no *outcome* do projeto.

#### 4.3.4. Hipótese 4 - Quais as principais razões que são apontadas como insucesso num projeto

O principal objetivo desta hipótese consiste em perceber quais foram as principais razões apontadas para o insucesso num projeto.

Desse modo, foi analisada a questão Q7 “Porque classifica como insucesso o projeto?” presente no **Anexo D**.

Tabela 23 : Análise descritiva da variável “Porque classifica como insucesso o Projeto

	<i>Respostas (N)</i>	<i>Percentagem</i>
<i>Retorno sobre o investimento nulo/negativo</i>	4	44,4%
<i>Qualidade inferior à proposta</i>	1	11,1%
<i>Insatisfação do cliente</i>	1	11,1%
<i>Prazo de entrega ultrapassado</i>	3	33,3%
<i>Total</i>	9	100,0%

A variável analisada está associada ao resultado do projeto “insucesso”. Devido ao número reduzido de projetos que resultaram em ‘insucesso’ no questionário, os resultados apresentados têm de ser interpretados com muitas reservas.

De qualquer forma, analisando os resultados obtidos conclui-se que as razões “Retorno sobre o investimento nulo/negativo” apontado em 4 dos 9 casos e “Prazo de entrega ultrapassado” apontado em 3 dos 9 casos são as razões mais apontadas, como se observa na Tabela 23.

#### 4.3.5. Hipótese 5 - Quais as principais fases com maior relevância para o sucesso/insucesso do projeto

De modo a aferir quais as principais fases com maior relevância no sucesso de um projeto, cruzaram-se as variáveis Q9 “Qual foi a fase do projeto que teve maior relevância para os resultados do projeto?” presente no **Anexo C** e “Project\_Result” que corresponde a Q5 - Em termos do resultado final como classifica o projeto?.

Para a realização desta análise, procedeu-se a um teste de hipóteses, onde a definição das hipóteses é dada por:

Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

- Hipótese nula, H0: As fases do projeto não estão relacionadas com o sucesso de um projeto (ou as fases do projeto são independente do sucesso do projeto).
- Hipótese alternativa, Ha: As fases do projeto estão relacionadas com o sucesso de um projeto (ou as fases do projeto não são independentes do sucesso do projeto).

Tabela 24 : Teste de Qui-Quadrado para a hipótese 5

	<i>Valor</i>	<i>gl</i>	<i>Significância Assintótica (Bilateral)</i>	<i>Sig exata (2 lados)</i>	<i>Sig exata (1 lado)</i>	<i>Probabilidade de ponto</i>
<b>Qui-quadrado de Pearson</b>	4,976 <sup>a</sup>	4	,290	,298		
<b>Razão de verossimilhança</b>	6,508	4	,164	,207		
<b>Teste Exato de Fisher</b>	4,587			,328		
<b>Associação Linear por Linear</b>	3,838 <sup>b</sup>	1	,050	,051	,029	,010
<b>Nº de Casos Válidos</b>	151					

a. 3 células (30,0%) esperavam uma contagem menor que 5. A contagem mínima esperada é 1,72.

b. A estatística padronizada é 1,959.

Dos resultados obtidos no âmbito da Tabela 24, resulta que:

- 3 células têm frequência esperada inferior a 5, o que corresponde a 30,0%, - tal não permite preencher o requisito ii), pois excede 20% no número total de células.
- A frequência esperada mínima é de 1,72, portanto superior a 1 – o que permite preencher também o requisito referido em i)

Desta forma, as condições de aplicabilidade do teste  $\chi^2$  de Pearson não estão verificadas, devido ao não preenchimento do requisito ii). Pelo que se optou pelo teste exato, observando-se um valor de  $p$  de 0,328, não permitindo a rejeição de H0. Conclui-se assim que as fases do projeto não estão relacionadas com o sucesso de um projeto.

## Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

Tabela 25 : Tabela de referencia cruzada entre as variáveis “Fase do projecto que teve maior relevância para os resultados do projeto” e “Project\_Result”

		<i>Não Sucesso</i>	<i>Sucesso</i>	<i>Total</i>
<b>Definição dos requisitos</b>	Contagem	23	24	47
	% em Coluna	35,4%	27,9%	31,1%
<b>Desenvolvimento</b>	Contagem	28	34	62
	% em Coluna	43,1%	39,5%	41,1%
<b>Implementação</b>	Contagem	11	16	27
	% em Coluna	16,9%	18,6%	17,9%
<b>Testes</b>	Contagem	3	8	11
	% em Coluna	4,6%	9,3%	7,3%
<b>Manutenção</b>	Contagem	0	4	4
	% em Coluna	0,0%	4,7%	2,6%
<b>Total</b>	Contagem	65	86	151
	% em Coluna	100,0%	100,0%	100,0%

Da Tabela 25, resulta como era expectável atendendo aos resultados do teste anterior que é pouco relevante a influência das fases do projeto no sucesso do mesmo. Podemos verificar que as fases que apresentam um maior número de respostas são a fase de “Definição dos Requisitos” apontando cerca de 31% dos respondentes, mas para cerca de 35% indicaram “Não Sucesso” como *outcome*. A fase de Desenvolvimento, recolheu cerca de 41% do total, mas com influência ligeiramente superior, quando o resultado do projeto é de Não sucesso (43%). Com maior impacto no sucesso dos projetos apesar de a frequência ser mais reduzida para o “Não Sucesso” e a fase de “Desenvolvimento” que apresenta um maior número de resposta para o “Sucesso” com uma frequência e 34 contra 28 do “Não Sucesso”.

### 4.3.6. Hipótese 6 - Quais as metodologias mais utilizadas que resultaram em sucesso/insucesso

De modo a aferir quais as metodologias com maior relevância no sucesso de um projeto, cruzaram-se as variáveis “Metodologias” e “Project\_Result” (descritas no capítulo anterior). Para a realização desta análise, procedeu-se a um teste de hipóteses, onde a definição das hipóteses é dada por:

- Hipótese nula, H0: As metodologias utilizadas no projeto não estão relacionadas com o sucesso de um projeto (ou as metodologias utilizadas no projeto são independentes do sucesso do projeto).

Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

- Hipótese alternativa, Ha: As metodologias utilizadas no projeto estão relacionadas com o sucesso de um projeto (ou as metodologias utilizadas no projeto são independentes do sucesso do projeto).

Tabela 26 : Teste Qui-Quadrado para a Hipótese 6

	<i>Valor</i>	<i>gl</i>	<i>Significância Assintótica (Bilateral)</i>	<i>Sig exata (2 lados)</i>	<i>Sig exata (1 lado)</i>	<i>Probabilidade de ponto</i>
<i>Qui-quadrado de Pearson</i>	2,745a	4	,601	,610		
<i>Razão de verossimilhança</i>	2,736	4	,603	,621		
<i>Teste Exato de Fisher</i>	2,782			,604		
<i>Associação Linear por Linear</i>	,684b	1	,408	,412	,220	,029
<i>Nº de Casos Válidos</i>	151					

a. 2 células (20,0%) esperavam uma contagem menor que 5. A contagem mínima esperada é 2,15.

b. A estatística padronizada é ,827.

Dos resultados obtidos no âmbito da Tabela 26, resulta que:

- 2 células têm frequência esperada inferior a 5, o que corresponde a 20,0%, - tal permite preencher o requisito ii), pois não excede 20% no número total de células.
- A frequência esperada mínima é de 2,15, portanto superior a 1 – o que permite preencher também o requisito referido em i)

Tendo-se verificado condições de aplicabilidade do teste do Qui-Quadrado de Pearson, não se verifica a rejeição da hipótese nula H0 (pois sig é superior a qualquer valor de referência usualmente utilizado) pelo que se conclui que as metodologias utilizadas no projeto não estão relacionadas com o *outcome* do mesmo.

## Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

Tabela 27 : Tabela de referência cruzada entre as variáveis "Tipo de Metodologias" e "Project\_Result

		<i>Não Sucesso</i>	<i>Sucesso</i>	<i>Total</i>
<b>Tradicional</b>	Contagem	24	23	47
	% em colona	36,9%	26,7%	31,1%
<b>Agile</b>	Contagem	17	27	44
	% em colona	26,2%	31,4%	29,1%
<b>Agile e Tradicional</b>	Contagem	5	9	14
	% em colona	7,7%	10,5%	9,3%
<b>Outras Metodologias</b>	Contagem	3	2	5
	% em colona	4,6%	2,3%	3,3%
<b>Nenhuma Metodologia</b>	Contagem	16	25	41
	% em colona	24,6%	29,1%	27,2%
<b>Total</b>	Contagem	65	86	151
	% em colona	100,0%	100,0%	100,0%

Os resultados presentes na Tabela 27, demonstram que o uso de metodologias tradicionais e metodologias Agile apresentam frequências idênticas, sendo que as metodologias tradicionais apresentam um total de 47 respostas contra 44 respostas das metodologias Agile.

No âmbito das metodologias tradicionais, a diferença de projetos que resultaram em ‘Sucesso’ comparativamente com os que resultam em ‘Não sucesso’ não é relevante, reportando-se apenas um projeto a mais que resulta em ‘Não sucesso’.

No caso das metodologias Agile existe uma diferença um pouco mais expressiva, uma vez que a taxa de ‘Sucesso’ é superior, com mais 10 projetos do que os 17 que resultam em ‘Não sucesso’.

Os projetos que não utilizam nenhuma metodologia representam um total de 41 respostas, sendo 25 respostas em projetos cujo o resultado é ‘Sucesso’ contra 16 de ‘Não Sucesso’.

### 4.3.7. Hipótese 7 - Quais os principais impactos da escolha dessas metodologias

O principal objetivo desta hipótese consiste em perceber quais os principais impactos das metodologias escolhidas no sucesso/insucesso de um projeto.

Desse modo, foi analisada a questão Q11 “Quais os maiores impactos da adoção das metodologias escolhidas anteriormente?”, presente no **Anexo C**.

Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

Tabela 28 : Análise descritiva da variável “Quais os maiores impactos da adoção das metodologias escolhidas anteriormente?”

	<i>Respostas</i>		<i>Percentagem de casos</i>
	<b>N</b>	<b>Percentagem</b>	
<i>Relação entre equipas</i>	42	16,3%	<b>39,3%</b>
<i>Qualidade do software</i>	33	12,8%	30,8%
<i>Relação com o cliente</i>	33	12,8%	30,8%
<i>Resposta às mudanças</i>	32	12,5%	29,9%
<i>Dinâmica de equipa</i>	35	13,6%	32,7%
<i>Objetivos</i>	39	15,2%	36,4%
<i>Planeamento</i>	43	16,7%	<b>40,2%</b>
<i>Total</i>	257	100,0%	240,2%

Como é possível observar na Tabela 28, de entre as possíveis respostas de adoção das metodologias apresentadas, é de destacar a “Relação entre as equipas” e o “Planeamento” ambos escolhidos em 40% das vezes e seguidamente os “Objetivos” com 36% das escolhas. As restantes opções foram escolhidas sensivelmente por 30% dos respondentes.

Adicionalmente, foram analisadas as frequências dos impactos no grupo de metodologias escolhidas e foi cruzada a variável anterior e a variável “metodologias” (descrita no capítulo anterior conforme se observa na Tabela 29:

Tabela 29 : Tabela de referência cruzada entre as variáveis “Tipo de Metodologias” e “Quais os maiores impactos da adoção das metodologias escolhidas anteriormente?”

	<i>Relação entre equipas</i>	<i>Qualidade do software</i>	<i>Relação com o cliente</i>	<i>Resposta às mudanças</i>	<i>Dinâmica de equipa</i>	<i>Objetivos</i>	<i>Planeamento</i>
<i>Tradicional</i>	<b>18</b>	11	12	6	10	<b>17</b>	<b>16</b>
<i>Agile</i>	9	<b>20</b>	11	16	<b>18</b>	12	16
<i>Agile e Tradicional</i>	5	8	8	9	5	6	8
<i>Outras Metodologias</i>	1	3	2	1	2	4	3
<i>Total</i>	42	33	33	32	35	39	43

A partir da Tabela 29, observa-se que nas Metodologias tradicionais os maiores impactos observados foram na “Relação entre equipas”, “Objetivos” e o “Planeamento” enquanto que

Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

nas metodologias Agile os maiores impactos foram na “Qualidade do software” e na “Dinâmica de equipa”.

#### 4.4. Discussão dos resultados

A Tabela 30 resume os resultados analisados no ponto anterior. Estatisticamente podemos concluir que as Hipóteses: H1, H3, H4, H5, H7 foram corroboradas enquanto que as Hipóteses H2 e H6 não foram corroboradas.

Tabela 30 : Resultados dos Teste das Hipóteses

<b>Hipóteses</b>	<b>Resultado</b>
<b>H1: Que influência tem a dimensão da equipa no sucesso de um projeto</b>	A dimensão da equipa de projeto está relacionada com o sucesso de um projeto
<b>H2: Que influência tem a dimensão do projeto no sucesso de um projeto</b>	A dimensão do projeto não está relacionada com o sucesso de um projeto
<b>H3: Quais os fatores críticos de sucesso que têm maior influência num projeto</b>	A dimensão Fatores Internos é a que tem maior influência no <i>outcome</i> do projeto
<b>H4: Quais as principais razões que são apontadas como insucesso num projeto</b>	“Retorno sobre o investimento nulo/negativo” e “Prazo de entrega ultrapassado” são as razões mais representativas
<b>H5: Quais as principais fases com maior relevância para o sucesso/insucesso do projeto</b>	As fases do projeto não estão relacionadas com o sucesso de um projeto
<b>H6: Quais as metodologias mais utilizadas que resultaram em sucesso/insucesso</b>	As metodologias utilizadas no projeto não estão relacionadas com o sucesso do mesmo
<b>H7: Quais os principais impactos da escolha dessas metodologias</b>	‘Planeamento’, ‘Relação entre as equipas’ e os ‘Objetivos’

## Capítulo 5 - Conclusões, limitações e sugestões

### 5.1. Conclusões

#### 5.1.1. Chaos Report

Do estudo realizado pelo *The Standish Group* resulta que, desde 2011, no que tange ao resultado final dos projetos, 50% dos projetos sofreram alterações, 30% dos projetos terminaram com sucesso e 20% dos projetos terminaram com insucesso.

Ora, os resultados obtidos nestes relatórios Chaos Report têm apresentado valores preocupantes devido às grandes percentagens de insucesso e de projetos que sofreram alterações. Face a isso mesmo, o The Standish Group procurou, através do cruzamento com diferentes variáveis, perceber o que influenciava o resultado final de um projeto.

A primeira variável que analisaram foi a *influência da dimensão de um projeto no seu resultado*, concluindo que um projeto grande e de média dimensão tem maior probabilidade de insucesso do que um projeto pequeno que, por sua vez, tem maior probabilidade de sucesso.

Outra variável que analisaram foi a *influência da utilização de metodologias de desenvolvimento de software (Ágeis e Tradicionais) no resultado de um projeto* tendo cruzado o tipo de metodologias com o resultado do mesmo, chegando à conclusão que o uso de metodologias tradicionais (*waterfall*) no desenvolvimento de *software* de projetos de TI proporcionam uma maior probabilidade de falha do que o uso de metodologias Ágeis, independentemente da dimensão do projeto.

No âmbito do estudo realizado para a presente investigação, os resultados divergiram bastante dos obtidos pelo Chaos Report, nomeadamente a maior percentagem (57%) pertence aos projetos que resultaram em sucesso, sendo que 37,7% sofreram alterações e apenas 5,3% resultarem em insucesso.

Relativamente à variável *influência da dimensão de um projeto no seu resultado*, o resultado da presente investigação foi que a dimensão do projeto não tem qualquer influência direta no resultado do projeto.

Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

Por outro lado, no que respeita à *influência da utilização de metodologias de desenvolvimento de software (Ágeis e Tradicionais) no resultado de um projeto*, os resultados da presente investigação demonstraram que uso de metodologias tradicionais e metodologias Ágeis apresentam frequências idênticas, sendo que as metodologias tradicionais apresentam um total de 47 respostas contra 44 respostas das metodologias Ágeis.

No âmbito das metodologias tradicionais, a diferença de projetos que resultaram em ‘Sucesso’ comparativamente com os que resultam em ‘Não sucesso’ não é relevante, reportando-se apenas um projeto a mais que resultou em ‘Não sucesso’.

### **5.1.2. Fatores Críticos de Sucesso**

Para além de procurar cruzar diferentes variáveis que pudessem influir no resultado do projeto, o *The Standish Group* identificou diferentes fatores de sucesso que, segundo os chaos report, influenciam diretamente o resultado do projeto.

Entre os fatores críticos de sucesso que apresenta, *The Standish Group* destaca o “Apoio da gestão superior”, “Maturidade Emocional da equipa”, “Envolvência do Cliente” e a “Otimização de processos” como os principais responsáveis pelo sucesso/insucesso representando cada um destes uma fatia de 15%.

Para além do *The Standish Group*, outros autores estudaram os fatores críticos de sucesso que maior influência tinham no resultado final de um projeto de desenvolvimento de software, é o caso de *Ahimbisibwe et al. (2015)* que a partir de recolha bibliográfica que fez identificou, num total de 148 estudos, “Apoio da gestão superior” que é referido em 104 estudos e “Envolvência do Cliente” em 102. *Fortune e White (2006)*, por sua vez, num total de 63 artigos identificou “Apoio da gerência sénior”, que é referido em 39 estudos e “Objetivos claros e realistas” em 31.

Nesta matéria, os resultados da investigação levada a cabo mostram outros fatores críticos de sucesso como os que maior influência têm no resultado final do projeto, nomeadamente, os Fatores Internos como “Infraestrutura e ferramentas de apoio”, “Comprometimento e motivação da equipa do projeto” e “Qualificações da equipa”.

### 5.1.3. Metodologias de Desenvolvimento de Software

A utilização de metodologias de desenvolvimento de software permite gerir melhor o ciclo de vida de um projeto, bem como detetar mais rapidamente as falhas que possam ocorrer no desenvolvimento do mesmo. Segundo um estudo presente em “*Forrester / Dr. Dobbs Global Developer Technographics® Survey, Q3 2010*”, 38,6% dos profissionais utilizam metodologias Ágeis, 19,5% utilizam metodologias iterativas, 13% utilizam metodologias tradicionais, e, por fim, 28,8% não utilizam qualquer metodologia no desenvolvimento de *software*.

Os resultados da presente investigação diferem dos *supra* referidos, uma vez que as metodologias tradicionais surgem como as mais utilizadas com 31,1%, seguidas das metodologias ágeis com 29,1%, 27,2% dos inquiridos não utilizam metodologias ou não têm conhecimento que é utilizado, 9,3% utilizam metodologias ágeis em conjunto com metodologias tradicionais e, por fim, 3,3% utilizam outras metodologias.

### 5.1.4. Conclusão final

Este trabalho teve como objetivos identificar quais os Fatores Críticos de Sucesso (FCS) que maior influência têm no sucesso e insucesso dos projetos de desenvolvimento de software e aferir se existe relação entre a escolha das metodologias de desenvolvimento de software e o resultado final de um projeto.

As principais conclusões estão sintetizadas na Tabela 31:

Tabela 31 : Resumo das conclusões finais

<b>Conclusões</b>	
<b>Chaos Report</b>	Os resultados obtidos neste trabalho divergem do estudo realizado pelo Standish Group no relatório Chaos Report. Segundo o Chaos Report, a maioria dos projetos sofreram alterações e a dimensão do projeto, bem como a utilização de metodologias de desenvolvimento de software influenciam o resultado final do projeto. Já os resultados obtidos nesta investigação demonstram que a maioria dos projetos termina com sucesso e que existem diversos fatores que influenciam o resultado do projeto, sendo que a dimensão do projeto e a utilização de metodologias não afetam, diretamente, o resultado final de um projeto.
<b>Fatores Críticos de Sucesso</b>	A escolha dos fatores críticos de sucesso que maior influência têm no resultado de um projeto de

**Metodologias de Desenvolvimento de Software**

desenvolvimento de software diverge de estudo para estudo, dependendo sempre de diferentes condicionantes, como o tipo de projeto, características do mercado, cultura etc..

O The Standish Group no Chaos Report destaca o “Apoio da gestão superior”, “Maturidade Emocional da equipa”, “Envolvência do Cliente” e a “Otimização de processos” como sendo os fatores críticos de sucesso que maior influência têm no resultado final de um projeto.

Por seu turno, *Ahimbisibwe et al. (2015)*, a partir da análise de diferentes estudos, definiu “Apoio da gestão superior” e “Envolvência do Cliente”.

Por fim, Fortune e White (2006) definiu “Apoio da gerência sénior” e “Objetivos claros e realistas” como os principais fatores críticos de sucesso.

Os resultados obtidos nesta investigação apresentaram conclusões diferentes dos autores referidos, sendo que os fatores críticos de sucesso com maior influência no resultado de um projeto são “Infraestrutura e ferramentas de apoio”, “Comprometimento e motivação da equipa do projeto” e “Qualificações da equipa”.

Segundo *“Forrester / Dr. Dobbs Global Developer Technographics® Survey, Q3 2010”* o número de projetos que utilizam metodologias Ágeis é superior, seguido de projetos que utilizam metodologias Iterativas, depois metodologias Tradicionais e, por fim, outras metodologias ou nenhuma.

Os resultados obtidos nesta investigação concluem que os projetos que usam metodologias tradicionais são mais representativos, seguidos dos projetos com utilização de metodologias Ágeis e, por fim, projetos sem utilização de qualquer tipo de metodologia de desenvolvimento de software.

Os resultados apresentados na Tabela 31, mostram que existe uma divergência nos resultados dos diferentes estudos em comparação com os resultados obtidos na presente investigação.

Um projeto de desenvolvimento de software pode ser influenciado por diferentes fatores para além dos referidos neste estudo o que leva a concluir que os FCS não são universais, ou seja, não existe um conjunto padrão de fatores aplicável a todas as situações. A existência de diferentes culturas, situações financeiras, negócios, entre outros são fatores que influenciam na definição dos fatores críticos, bem como na escolha da metodologia de desenvolvimento de software.

Os resultados obtidos na presente investigação dizem respeito a projetos de desenvolvimento de software exclusivamente realizados em Portugal e, comparativamente ao estudo realizado

Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

pelo The Standish Report em Chaos Report, os valores são, significativamente, mais positivos, mas também demonstram que o setor ainda necessita de evoluir na utilização e divulgação das metodologias de desenvolvimento.

## **5.2. Limitações do trabalho**

A maior limitação à realização da presente investigação foi a dificuldade em conseguir arranjar uma amostra de estudo maior, principalmente, por se pretender que os projetos analisados tivessem sido realizados exclusivamente em Portugal e no sector das TIC.

Não obstante a amostra não ter sido tão grande como se pretendia, foi a suficiente para chegar aos resultados apresentados na presente investigação.

## **5.3. Sugestões para pesquisa futura**

Sugere-se, para a realização de uma pesquisa futura, que se tenha em conta, aquando da realização do questionário inicial:

- Fatores críticos de resposta aberta
- A possibilidade de diferentes intervenientes responderem em relação ao mesmo projeto de modo a perceber se ambos têm a mesma perceção de quais os fatores críticos que maior influência tiveram no resultado final desse mesmo projeto.
- Existir uma definição do significado de “fator crítico de sucesso”, de modo a evitar diferentes interpretações.

## Bibliografia

- AGILEBUSINESS, Consortium - Home page [Em linha], atual. 2017. Disponível em: <https://www.agilebusiness.org/>, acessado em 23 Out 2018
- AHIMBISIBWE, A.; CAVANA, R. Y.; DAELLENBACH, U. (2015) A contingency fit model of critical success factors for software development projects. *Journal of Enterprise Information Management*, Vol. 28, Iss 1, pp. 7-33.
- AITKEN, Ashley; ILANGO, Vishnu - A comparative analysis of traditional software engineering and agile software development. Em *System Sciences (HICSS)*, 2013 46th Hawaii International Conference on. [S.l.] : IEEE, 2013
- ANAND, R. Vijay; DINAKARAN, M. - Popular Agile Methods in Software Development: Review and Analysis. *International Journal of Applied Engineering Research*. 11:5 (2016) 3433–3437
- BECK, Kent - *Extreme programming explained: embrace change*. [S.l.] : addison-wesley professional, 2000
- CHARVAT, Jason - *Project management methodologies: selecting, implementing, and supporting methodologies and processes for projects*. [S.l.] : John Wiley & Sons, 2003
- CHIN, CMM; SPOWAGE, AC - Defining & classifying project management methodologies. *PM World Today*. 12:5 (2010) 1–9.
- Cockburn, A., and Williams. "The Costs and Benefits of Pair Programming," in: *Extreme Programming Examined*, G. Succi and M. Marchesi (eds.). Boston, MA: Addison Wesley, 2001
- DAI, Christine Xiaoyi; WELLS, William G. - An exploration of project management office features and their relationship to project performance. *International Journal of Project Management*. 22:7 (2004) 523–532.
- Forrester / Dr. Dobbs Global Developer Technographics® Survey, Q3 2009
- Fortune, J. and White, D. (2006), “Framing of project critical success factors by a systems model”, *International Journal of Project Management*, Vol. 24 No. 1, pp. 53-65.
- Harden, Benjamim, 2011. Making the Right Choice – Agile vs. Waterfall. [online] Disponível em: <http://www.smartdatacollective.com/bharden/38031/making-right-choice-agile-vs-waterfall> acessado em: 02 Jan. 2017.
- Imreh, R. and Raisinghani, M. (2011), “Impact of agile software development on quality within information technology organisations”, *Journal of Emerging Trends in Computing and Information Science*, Vol. 10 No. 10, pp. 460-475
- KERZNER, Harold - *Strategic planning for project management using a project management maturity model*. [S.l.] : John Wiley & Sons, 2002
- Kruchten 1996, Philippe Kruchten. *A Rational Development Process*, in *CrossTalk*.

Análise de metodologias de desenvolvimento de software utilizadas na gestão de projetos no sector das TIC

- Kruchten 2000, Philippe Kruchten. The Rational Unified Process: An Introduction, 2nd ed.
- LEI, Howard et al. - A statistical analysis of the effects of Scrum and Kanban on software development projects. Robotics and Computer-Integrated Manufacturing. 43:2017) 59– 67
- MALHOTRA, N. Pesquisa de marketing: uma orientação aplicada. 4ª Ed. Porto Alegre: Bookman, 2006.
- MARKS, Adam - Information Systems Development Methodologies in Developing Higher Education. Creative Education. 3:01 (2012) 114.
- MARTINS, José C. C. Gerenciando projetos de desenvolvimento de software com PMI, RUP e UML. Quarta Edição. Rio de Janeiro: Brasport, 2007
- Miguel, A., 2013, Gestão Moderna de Projectos: melhores técnicas e práticas, Editora FCA.
- Nayab, N, 2012. Agile vs. Waterfall - Is There a Real Winner?. [online] Disponível em <http://www.brighthubpm.com/agile/50473-agile-vs-waterfall-is-there-a-real-winner/> acedido em: 02 Jan. 2017.
- PAULSON, Linda Dailey - Adapting methodologies for doing software right. IT Professional. 3:4 (2001) 13–15
- PRESSMAN, Roger S. - Software engineering: a practitioner's approach. [S.l.] : Palgrave Macmillan, 2005
- Project Management Institute, (2017), “Guide to the Project Management Body of Knowledge (pmbok Guide) & Agile Practice”
- Rachel Burger, 15 Incredible Agile Project Management Statistics for 2018 Disponível em: <https://blog.capterra.com/agile-project-management-statistics-for-2018/> acedido em 15 Oct 2018
- RICO, David F.; SAYANI, Hasan H.; SONE, Saya - The business value of Agile software methods. Maximizing ROI with just-in-time processes and documentation, Fort Lauderdale, FL: J. Ross Pub. 10:2009).
- SNYDER, Cynthia Stackpole - A guide to the project management body of knowledge: PMBOK (®) guide. Project Management Institute: Newtown Square, PA, USA. 2014).
- SOARES, Michel dos Santos. Comparação entre Metodologias Agile e Tradicionais para o Desenvolvimento de Software. 2004. Disponível em: <http://www.dcc.ufla.br/infocomp/index.php/INFOCOMP/article/view/68> acedido em: 02 Jan. 2017
- SUTHERLAND, Jeff - The scrum papers: nut, bolts, and origins of an Agile framework. Scrum inc. 2012).
- TEIXEIRA, Bruno Afonso - Agile and traditional project management: bridge between two worlds to manage IT projects PhD Thesis.
- The Standish Group. (2015). The Chaos Report.

## Anexo A

### Traditional plan-based vs agile software development methodology , Adapted from Imreh and Raisinghani (2011)

<b>Parâmetro de projeto</b>	<b>Desenvolvimento tradicional de <i>software</i></b>	<b>Desenvolvimento Ágile de <i>software</i></b>
Equipa de desenvolvimento	Orientado ao planeamento, equipas pre-estruturadas	Equipas Ágeis, partilha de conhecimento, colaborativas e organizadas entre si.
Relação com o cliente	Interações mínimas	Privilegiam a interação e partilha de conhecimento, discussão de soluções em conjunto
Requerimentos	Conhecidos cedo, em grande parte são estáveis e inalteráveis	Adaptáveis e de rápida alteração
Arquitetura	Desenho feito com base nos requerimentos existentes	Desenho feito com base nos requerimentos previsíveis e existentes
Dimensão das equipas	Equipas grandes	Equipas pequenas
Objetivos	Alta garantia	Valor rápido
Principais suposições	Sistema é todo especificado e detalhado na documentação, planeamento extensivo.	Software facilmente adaptável a qualquer solução, desenvolvido com base em princípios de entregas e testes contínuos, baseado no rápido feedback e em alterações rápidas se necessárias
Estilo de gestão	Comandar e controlar	Liderar e colaborar
Conhecimentos de gestão	Explicito	Tático
Modelo de desenvolvimento	Linear ou incremental	Iterativo ou adaptativo
Comunicação	Formal	Informal
Estrutura organizacional	Mecânica (burocrático com alta formalização) destinada a grandes organizações	Orgânico (flexível e participativa, com incentivo à ação cooperativa) destinado a pequenas e médias organizações
Controlo de qualidade	Planeamento e controlo rigoroso, tardio, testes pesados	Controlo contínuo dos requerimentos, desenho das soluções, testes contínuos
Cultura Organizacional	Comando e controlo	Liderança e colaboração
Mercado	Maduro, estável	Dinâmico
Medida de sucesso	Conforme o planeado	Valor comercial entregue

## Anexo B

Metodologia	Vantagens	Desvantagens
Scrum	<ul style="list-style-type: none"> <li>• Os trabalhadores sentem-se mais motivados devido ao interesse de entregar o Sprint dentro do prazo;</li> <li>• Dentro da organização o projeto pode ser observado por todos;</li> <li>• Como a qualidade é mais importante do que o prazo de entrega, o produto apresenta uma diminuição significativa de erros;</li> <li>• Os trabalhadores podem definir as prioridades, garantindo assim que <i>sprints</i> que ainda não foram finalizados possam ser alterados</li> </ul>	<ul style="list-style-type: none"> <li>• Como a qualidade é mais importante do que o resultado, os prazos podem diferir dos estipulados, levando a um atraso no resultado final;</li> <li>• A presença de papéis indefinidos nas funções presentes do projeto, podem originar alguns problemas relacionados com a comunicação interna;</li> <li>• A falta de documentações sobre o desenrolar do projeto pode ser um problema. Por isso é importante documentar aspetos que sejam verdadeiramente importantes para o projeto;</li> </ul>
FDD	<ul style="list-style-type: none"> <li>• Como cada <i>feature</i> é pequena, agrupar os requisitos torna-se mais fácil;</li> <li>• <i>Features</i> podem ser organizadas de forma hierárquica;</li> <li>• Menor custo humano, dado que cada <i>feature</i> pode ser desenvolvida de maneira independente;</li> <li>• Devido ao tamanho da <i>feature</i> é mais fácil detetar erros no <i>design</i>;</li> </ul>	<ul style="list-style-type: none"> <li>• Põe em questão a efetividade/aplicabilidade do FDD;</li> <li>• Não existe um consenso do tamanho que cada <i>feature</i> deve ter;</li> <li>• Manutenção.</li> </ul>
XP	<ul style="list-style-type: none"> <li>• O Cliente está sempre por perto, resultando num produto final muito próximo daquilo que ele deseja;</li> <li>• A programação em dupla reduz o número de erros e a aumenta a legibilidade do código, o que facilita nas futuras manutenções;</li> <li>• Organização das tarefas;</li> <li>• Erros são encontrados inicialmente, pois são realizados vários testes de diversas formas</li> </ul>	<ul style="list-style-type: none"> <li>• O Cliente deve disponibilizar uma equipa para participar em todo o processo de desenvolvimento;</li> <li>• Existe uma perda de produtividade ao se programar em grupos de dois elementos;</li> <li>• Todos os envolvidos no projeto devem conhecer as técnicas e regras;</li> <li>• Exige maturidade dos participantes;</li> </ul>
Lean	<ul style="list-style-type: none"> <li>• Melhoria da qualidade;</li> <li>• Mudanças positivas na organização;</li> <li>• Produção flexível;</li> <li>• Maior velocidade de produção;</li> <li>• Melhor deteção de falhas;</li> </ul>	<ul style="list-style-type: none"> <li>• O <i>developer</i> deve ser capaz de promover manutenção de primeiro nível e estar apto a mudanças;</li> <li>• Alto custo de implantação;</li> </ul>

	<ul style="list-style-type: none"> <li>• Redução do tempo de <i>Setup</i>;</li> <li>• Corrige erros nas suas causas fundamentais;</li> <li>• Cultura de Melhoria Contínua.</li> </ul>	
DSDM	<ul style="list-style-type: none"> <li>• Produz resultados com rapidez;</li> <li>• Pode ser utilizado em projetos que necessitem de avaliação constante dos clientes;</li> <li>• O planeamento pode ser adaptado em qualquer fase do projeto;</li> <li>• Criação e reutilização de componentes;</li> <li>• Enfatiza um ciclo de desenvolvimento extremamente curto (entre 60 e 90 dias);</li> <li>• Visibilidade mais cedo (protótipos);</li> </ul>	<ul style="list-style-type: none"> <li>• O envolvimento com o cliente tem que ser ativo;</li> <li>• Funções desnecessárias (re-uso de componentes);</li> <li>• Custo do conjunto de ferramentas e <i>hardware</i> para correr a aplicação;</li> <li>• Sucessos anteriores são difíceis de se reproduzir;</li> <li>• Para projetos grandes exige recursos humanos suficientes para criar o número correto de equipas, isso implica um alto custo com a equipa;</li> <li>• Se uma aplicação não puder ser modularizada de modo que cada função principal seja completada em menos de 3 meses, não é aconselhável o uso desta metodologia.</li> </ul>
RUP	<ul style="list-style-type: none"> <li>• Após se iniciar o desenvolvimento, é comum que os requisitos se alterem, fazendo com que haja alterações na documentação.</li> <li>• Desenvolvendo iterativamente, essas alterações são fáceis de gerir;</li> <li>• Quando cada módulo é desenvolvido este é integrado ao sistema como um todo, evitando que este seja um problema no final do projeto;</li> <li>• A cada iteração é possível analisar pontos críticos e planear estratégias para não perder tempo durante o desenvolvimento;</li> <li>• Os testes são realizados no final de cada módulo, permitindo que erros e não-conformidades possam ser tratados ainda dentro do mesmo ciclo.</li> </ul>	<ul style="list-style-type: none"> <li>• Investimento em ferramentas de suporte;</li> <li>• Aumento dos requisitos da documentação;</li> <li>• Experiência da equipa;</li> <li>• É necessário adquirir formações adequadas, adaptação da metodologia no contexto ao qual ela será utilizada, apoio especializado para as equipas de desenvolvimento e tempo para a absorção da metodologia.</li> </ul>
Waterfall	<ul style="list-style-type: none"> <li>• Minimiza o tempo de planeamento;</li> <li>• Funciona bem para equipas tecnicamente mais fracas;</li> </ul>	<ul style="list-style-type: none"> <li>• Inflexível;</li> <li>• Apenas a fase final produz um <i>deliverable</i>;</li> <li>• A não deteção de um erro no início do projeto pode colocar em questão que o mesmo seja terminado com sucesso.</li> </ul>

## Anexo C

# Análise de metodologias de desenvolvimento de *software* utilizadas na gestão de projetos no sector

---

### Start of Block

Este questionário serve fins meramente académicos e está inserido no âmbito de uma tese de mestrado na área de Gestão de Projectos. Procura-se estudar o sucesso/insucesso dos projetos, quais os fatores críticos de sucesso/insucesso e quais as metodologias de gestão de software mais utilizadas. O resultado deste estudo será, posteriormente, comparado com o famoso estudo do *CHAOS report do Standish Group*. Afinal, o principal objetivo deste estudo passa por perceber o que falha, porque falha e o que pode futuramente ser feito na gestão de projetos, o que poderá trazer mais valias para todas as empresas e gestores de projetos do nosso país. As respostas são anónimas e os dados serão tratados de forma totalmente confidencial. O seu preenchimento demora cerca de 2 minutos. Por favor responda a todas as questões, pois só assim estará a contribuir para o sucesso desta investigação. Muito Obrigado pela sua colaboração, Rui Bernardo.

---

Q1 Qual é a função do entrevistado no projeto?

- Project Manager (1)
  - Team Leader (2)
  - Developer (3)
  - Outra (4) \_\_\_\_\_
- 

Q2 Como classifica a dimensão da equipa no projeto?

- Inferior à necessária (1)
  - Adequada (2)
  - Superior à necessária (3)
-

Q3 Como classifica os prazos dados no âmbito do projeto?

- Prazos apertados (1)
  - Prazos adequados (2)
  - Prazos demasiado longos (3)
- 

Q4 Qual a dimensão do projeto?

- Pequeno (1)
  - Médio (2)
  - Grande (3)
- 

Q5 Em termos do resultado final como classifica o projeto?

- Sucesso (solução é entregue e a empresa reconheceu os critérios de sucesso dentro de um intervalo aceitável) (1)
- Insucesso (a equipa de projeto não entregou a solução) (2)
- Sofreu alterações (solução foi entregue mas a equipa não cumpriu com todos os requisitos do projeto [qualidade é razoável mas o projeto ultrapassou o tempo estipulado e o ROI - 'return of investment' - é muito baixo]) (3)

*Skip To: Q8 If Q5 = Sucesso (solução é entregue e a empresa reconheceu os critérios de sucesso dentro de um intervalo aceitável)*

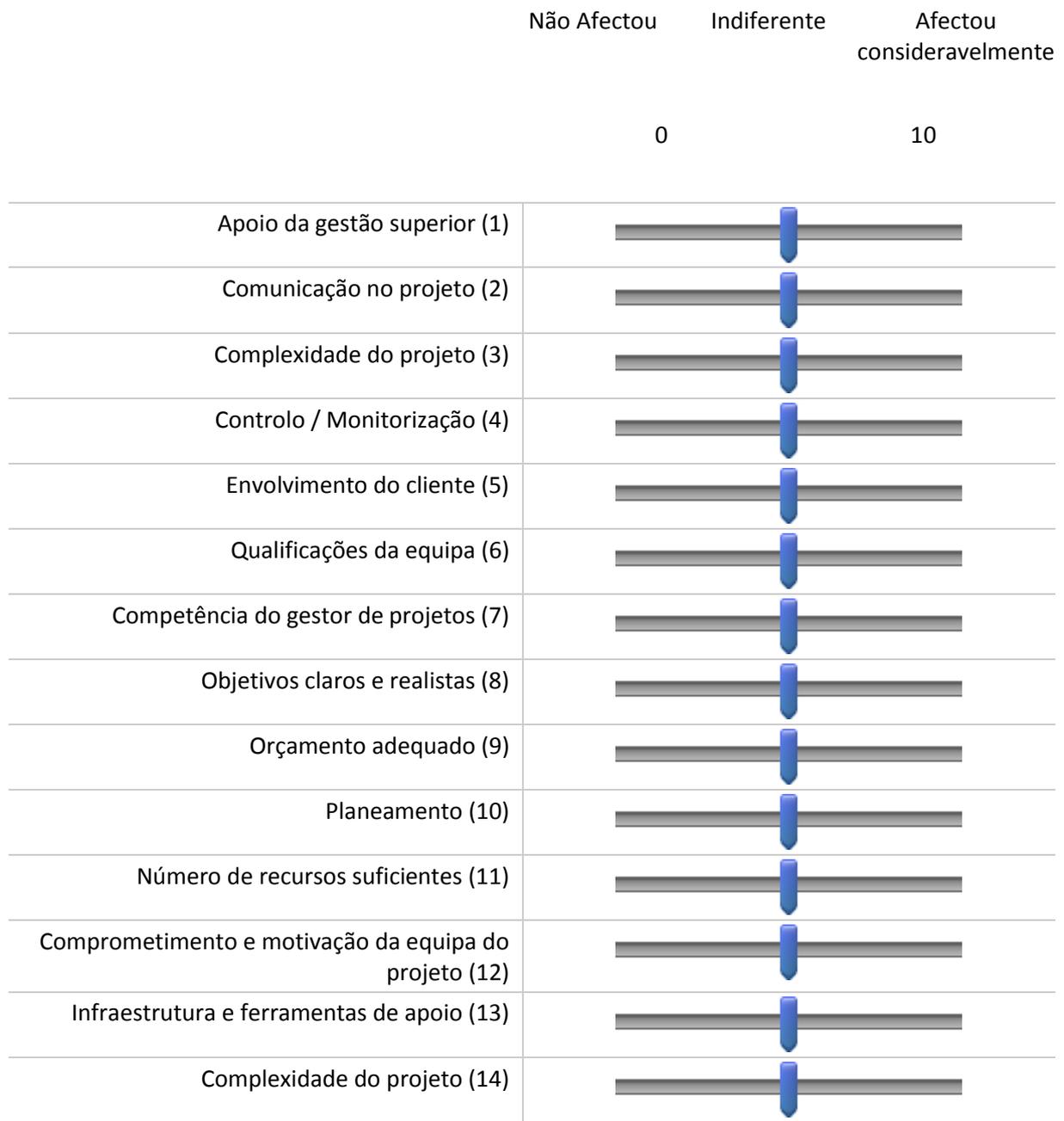
*Skip To: Q8 If Q5 = Sofreu alterações (solução foi entregue mas a equipa não cumpriu com todos os requisitos do projeto [qualidade é razoável mas o projeto ultrapassou o tempo estipulado e o ROI - 'return of investment' - é muito baixo])*

---

Q7 Porque classifica como insucesso o projeto?

- Retorno sobre o investimento nulo/negativo (1)
  - Qualidade inferior à proposta (2)
  - Insatisfação do cliente (3)
  - Prazo de entrega ultrapassado (4)
-

Q8 Em que medida os seguintes factores afectaram o resultado final do projeto?



Q9 Qual foi a fase do projecto que teve maior relevância para os resultados do projeto?

- Definição dos requisitos (1)
  - Desenvolvimento (2)
  - Implementação (3)
  - Testes (4)
  - Manutenção (5)
- 

Q10 Foram adotadas algumas metodologias/princípios?

- Waterfall (1)
- Scrum (2)
- Lean (Kanban) (3)
- eXtremme Programming (XP) (4)
- Feature Driven Development (FDD) (5)
- Dynamic Systems Development Method (DSDM) (6)
- Rational Software Corporation (RUP) (7)
- Prototipagem (8)
- Outra (9) \_\_\_\_\_
- Nenhuma (10)

*Skip To: End of Survey If Q10 = Nenhuma*

---

Q11 Quais os maiores impactos da adoção das metodologias escolhidas anteriormente?

- Relação entre equipas (Indivíduos e interações / processos e ferramentas) (1)
- Qualidade do software (Software operacional / documentação completa) (2)
- Relação com o cliente (Colaboração dos clientes / negociação contratual) (3)
- Resposta às mudanças (Resposta às mudanças / seguir um plano de acções) (4)
- Dinâmica de equipa (A comunicação e ajuda entre as pessoas é encorajada) (8)
- Objetivos (Bem definidos numa fase inicial/Definidos em alto nível e os requisitos são priorizados e definidos de forma iterativa) (10)
- Planeamento (Plano de projecto detalhado/plano de projecto evolutivo) (11)

End of Block: Default Question Block

---

## Anexo D

<https://us8.campaign-archive.com/?u=0d7dfb2b1450f9fe26a10eedf&id=f9c28048eb>



Past Issues Translate ▼



### INVESTIGAÇÃO E DESENVOLVIMENTO

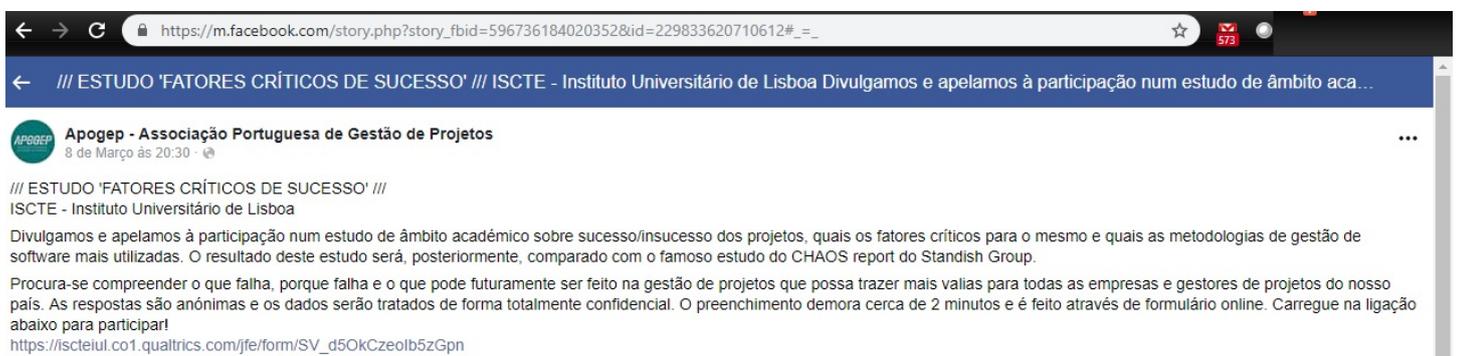
#### ESTUDO 'FATORES CRÍTICOS DE SUCESSO'

**ISCTE - Instituto Universitário de Lisboa**

Divulgamos e apelamos à participação num estudo de âmbito académico sobre sucesso/insucesso dos projetos, quais os fatores críticos para o mesmo e quais as metodologias de gestão de software mais utilizadas. O resultado deste estudo será, posteriormente, comparado com o famoso estudo do [CHAOS report do Standish Group](#).

Procura-se compreender o que falha, porque falha e o que pode futuramente ser feito na gestão de projetos que possa trazer mais valias para todas as empresas e gestores de projetos do nosso país. As respostas são anónimas e os dados serão tratados de forma totalmente confidencial. O preenchimento demora cerca de 2 minutos e é feito através de [formulário online](#). Carregue no link para aceder.

[https://m.facebook.com/story.php?story\\_fbid=596736184020352&id=229833620710612#\\_](https://m.facebook.com/story.php?story_fbid=596736184020352&id=229833620710612#_)



← → ↻ 🔒 [https://m.facebook.com/story.php?story\\_fbid=596736184020352&id=229833620710612#\\_](https://m.facebook.com/story.php?story_fbid=596736184020352&id=229833620710612#_) ☆ 573

← /// ESTUDO 'FATORES CRÍTICOS DE SUCESSO' /// ISCTE - Instituto Universitário de Lisboa Divulgamos e apelamos à participação num estudo de âmbito aca...

**APGEP** Apogep - Associação Portuguesa de Gestão de Projetos  
8 de Março às 20:30 · 🌐

/// ESTUDO 'FATORES CRÍTICOS DE SUCESSO' ///

ISCTE - Instituto Universitário de Lisboa

Divulgamos e apelamos à participação num estudo de âmbito académico sobre sucesso/insucesso dos projetos, quais os fatores críticos para o mesmo e quais as metodologias de gestão de software mais utilizadas. O resultado deste estudo será, posteriormente, comparado com o famoso estudo do CHAOS report do Standish Group.

Procura-se compreender o que falha, porque falha e o que pode futuramente ser feito na gestão de projetos que possa trazer mais valias para todas as empresas e gestores de projetos do nosso país. As respostas são anónimas e os dados serão tratados de forma totalmente confidencial. O preenchimento demora cerca de 2 minutos e é feito através de formulário online. Carregue na ligação abaixo para participar!

[https://iscteiu1.co1.qualtrics.com/jfe/form/SV\\_d5OKCzeolb5zGpn](https://iscteiu1.co1.qualtrics.com/jfe/form/SV_d5OKCzeolb5zGpn)

## Anexo E

### Correlação entre as variáveis duplicadas "Complexidade do Projeto"

		Complexidade do projeto 1	Complexidade do projeto 2
Complexidade do projeto 1	Correlação de Pearson	1	,706**
	Sig. (bilateral)		0,000
	N	151	151
Complexidade do projeto 2	Correlação de Pearson	,706**	1
	Sig. (bilateral)	0,000	
	N	151	151

\*\* . A correlação é significativa no nível 0,01 (bilateral).

## Anexo F

### Análise em Componentes Principais

As questões a serem analisadas correspondem a em que medida os 13 fatores críticos de sucesso afetaram o resultado final do projeto.

Estes itens foram recolhidos segundo uma escala tipo Likert com 10 pontos, variando entre 0 – Não afetou – a 5 Indiferente –a 10 Afetou consideravelmente. Não sendo de facto variáveis quantitativas, este tipo de variáveis são em geral tratadas como se o fossem. Verifica-se ainda que foram considerados 151 casos válidos, sendo assim em quantidade suficiente para a realização da ACP (valor de referência: 10 x número de variáveis = 10 x 13 = 130).

#### Estatísticas Descritivas

	N	Mínimo	Máximo	Média	Desvio Padrão
Apoio da gestão superior	151	,00	10,00	6,4106	2,77914
Comunicação no projeto	151	,00	10,00	7,1854	2,55970
Controlo / Monitorização	151	,00	10,00	6,1523	2,72213
Envolvimento do cliente	151	,00	10,00	6,2848	2,76738
Qualificações da equipa	151	,00	10,00	6,1060	3,08145
Competência do gestor de projetos	151	,00	10,00	6,1987	3,02218
Objetivos claros e realistas	151	,00	10,00	6,6623	2,85864
Orçamento adequado	151	,00	10,00	5,6424	2,90366
Planeamento	151	,00	10,00	6,8742	2,85728
Número de recursos suficientes	151	,00	10,00	6,5762	3,04070
Comprometimento e motivação da equipa do projeto	151	,00	10,00	5,9536	3,23592
Infraestrutura e ferramentas de apoio	151	,00	10,00	5,3974	2,91909
complexidade_composta	151	,00	10,00	6,1755	2,58663
N válido (listwise)	151				

## Teste de KMO e Bartlett

Medida Kaiser-Meyer-Olkin de adequação de amostragem.		,723
Teste de esfericidade de Bartlett	Aprox. Qui-quadrado	782,368
	gl	78
	Sig.	,000

O teste de Bartlett tem como hipóteses

H0: R = I13x13

H1: R ≠ I13x13

Como se pode constatar da Tabela anterior, sendo sig = 0.000 <  $\alpha$  = 0.05, a H0 é rejeitada, assumindo-se então que a matriz de correlação não é a matriz identidade. Por outro lado, a medida KMO é igual a 0.723, o que corresponde a um valor bastante elevado, e a uma muito boa adequabilidade dos dados à análise. Conclui-se assim que os dados são adequados à análise realizada.

### Variância total explicada

Com pone nte	Autovalores iniciais			Somadas de extração de carregamentos ao quadrado			Somadas de rotação de carregamentos ao quadrado		
	Total	% de variância	% cumulativa	Total	% de variância	% cumulativa	Total	% de variância	% cumulativa
1	3,938	30,289	30,289	3,938	30,289	30,289	2,333	17,946	17,946
2	1,640	12,614	42,903	1,640	12,614	42,903	2,192	16,861	34,807
3	1,153	8,871	51,774	1,153	8,871	51,774	1,800	13,843	48,649
4	1,094	8,415	60,189	1,094	8,415	60,189	1,500	11,539	60,189
5	,971	7,468	67,657						
6	,733	5,636	73,293						
7	,719	5,528	78,821						
8	,651	5,008	83,829						
9	,558	4,290	88,119						
10	,511	3,931	92,050						
11	,405	3,115	95,165						
12	,366	2,816	97,981						
13	,262	2,019	100,000						

Método de Extração: Análise de Componente Principal.

A solução com 4 componentes, o que estaria de acordo com o critério de Kaiser (valores próprios superiores a 1, tabela anterior). Nesta solução de quatro componentes, as variáveis originais estão bem representadas pelas novas variáveis, já que, em 11 das 13 variáveis, a percentagem de variância explicada está acima dos 50%. Apenas dois casos, (Planeamento e Apoio da Gestão Superior) apresentam valores inferiores, mas ainda assim relativamente próximos dos 50%.

### Comunalidades

	Inicial	Extração
Apoio da gestão superior	1,000	,476
Comunicação no projeto	1,000	,636
Controlo / Monitorização	1,000	,628
Envolvimento do cliente	1,000	,608
Qualificações da equipa	1,000	,710
Competência do gestor de projetos	1,000	,719
Objetivos claros e realistas	1,000	,622
Orçamento adequado	1,000	,521
Planeamento	1,000	,467
Número de recursos suficientes	1,000	,518
Comprometimento e motivação da equipa do projeto	1,000	,684
Infraestrutura e ferramentas de apoio	1,000	,708
complexidade_composta	1,000	,528
Método de Extração: Análise de Componente Principal.		

### Matriz de componente rotativa

	Componente			
	Cultura Organizacional	Fatores internos	Fatores do projeto	Fatores externos
Comunicação no projeto	,751			,257
Competência do gestor de projetos	,710	,460		
Apoio da gestão superior	,668		,139	
Planeamento	,635	,180		,155
Infraestrutura e ferramentas de apoio	,159	,826		
Comprometimento e motivação da equipa do projeto		,666	,364	,330
Qualificações da equipa		,647	,393	,370
complexidade_composta		,204	,685	
Número de recursos suficientes	,251	,101	,665	
Objetivos claros e realistas	,421	-,203	,553	,312
Orçamento adequado	,317	,367	,477	-,241
Controlo / Monitorização	,101		,169	,767
Envolvimento do cliente	,175	,411		,636
Método de Extração: Análise de Componente Principal. Método de Rotação: Varimax com Normalização de Kaiser.				
a. Rotação convergida em 6 iterações.				

## Anexo G

### Teste Qui-Quadrado para a Hipótese 1

	Valor	gl	Significância Assintótica (Bilateral)	Sig exata (2 lados)	Sig exata (1 lado)	Probabilidade de ponto
Qui-quadrado de Pearson	28,018a	1	,000	,000	,000	
Correção de continuidade <sup>b</sup>	26,282	1	,000			
Razão de verossimilhança	28,687	1	,000	,000	,000	
Teste Exato de Fisher				,000	,000	
Associação Linear por Linear	27,833c	1	,000	,000	,000	,000
Nº de Casos Válidos	151					
a. 0 células (0,0%) esperavam uma contagem menor que 5. A contagem mínima esperada é 27,12.						
b. Computado apenas para uma tabela 2x2						
c. A estatística padronizada é 5,276.						

Dos resultados obtidos no âmbito da tabela supra, resulta que:

- Nenhuma célula tem frequência esperada inferior a 5, o que corresponde a 0,0% - tal permite preencher o requisito ii);
- A frequência esperada mínima é 27,12, portanto superior a 1 – o que permite preencher também o requisito referido em i);

Desta forma, as condições de aplicabilidade do teste estão verificadas.

Por fim,  $\text{Sig} < 0,001 < \alpha = 0,05$

## Teste Qui-Quadrado para a Hipótese 2

	Valor	gl	Significância Assintótica (Bilateral)	Sig exata (2 lados)	Sig exata (1 lado)	Probabilidad e de ponto
Qui-quadrado de Pearson	1,003a	2	,606	,640		
Razão de verossimilhança	1,001	2	,606	,640		
Teste Exato de Fisher	1,032			,610		
Associação Linear por Linear	,045b	1	,833	,908	,462	,090
Nº de Casos Válidos	151					
a. 0 células (0,0%) esperavam uma contagem menor que 5. A contagem mínima esperada é 11,19.						
b. A estatística padronizada é ,211.						

Dos resultados obtidos no âmbito da tabela supra, resulta que:

- Nenhuma célula tem frequência esperada inferior a 5, o que corresponde a 0,0% - tal permite preencher o requisito ii);
- A frequência esperada mínima é 11,19, portanto superior a 1 – o que permite preencher também o requisito referido em i);

Desta forma, as condições de aplicabilidade do teste estão verificadas.

Por fim,  $\text{Sig} < 0,001 < \alpha = 0,05$