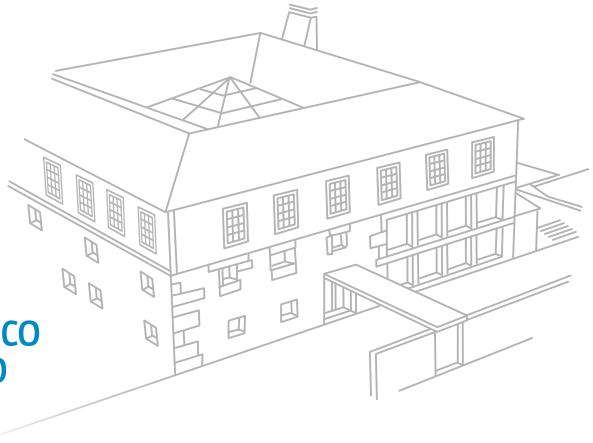


ESTGF | POLITÉCNICO DO PORTO

ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

DESIGNAÇÃO DO MESTRADO

AUTOR

ORIENTADOR(ES)

ANO

www.estgf.ipp.pt

Ao Diogo e à Fabiana,

Agradecimentos

Este espaço é dedicado àqueles que contribuíram e que me acompanharam ao longo desta árdua etapa. Sendo este um espaço limitado que, seguramente, não me permite agradecer como devia a todos, deixo apenas algumas palavras de agradecimento e profundo reconhecimento.

Aos meus orientadores Doutora Carla Sofia Pereira e Dr. José Ângelo Pinto pela orientação científica, em especial à Doutora Carla Sofia Pereira pelo apoio, disponibilidade, incentivo, paciência e preocupação contínua com a evolução deste trabalho, que foi crucial para a sua conclusão.

À Doutora Dorabela Gamboa, coordenadora do Mestrado em Engenharia Informática e diretora do CIICESI por todo o incentivo e disponibilidade manifestados.

À Strongstep, em especial ao Engenheiro Pedro Castro Henriques e ao Engenheiro Bruno Sousa Martins pela confiança, orientação, disponibilidade e principalmente pela possibilidade de aprender com profissionais de excelência ao longo destes meses.

À minha família, em especial aos meus pais e à minha irmã, por toda a confiança e compreensão que me deram alento para avançar.

A todos os meus amigos, pelo incentivo, preocupação, apoio e amizade manifestadas nesta etapa final.

Obrigada!

Resumo

Vivemos numa sociedade onde impera a mudança, como tal as organizações têm de se adaptar e adotar medidas que sirvam de suporte às novas exigências que surgem. O desenvolvimento ágil de *software* surgiu no sentido de dar resposta às constantes alterações dos requisitos de negócio e face à sua popularidade, a gestão do risco numa vertente ágil tornou-se fulcral, visto que as metodologias ágeis por si só não dão resposta aos riscos que possam surgir em projetos ágeis de *software*. Os riscos são inerentes a todos os projetos de *software* e a gestão e controlo dos mesmos representam uma mais-valia para o desenvolvimento e sucesso de projetos. A gestão do risco tem como objetivo controlar, de forma contínua, os riscos que surgem em todas as fases dos projetos, sendo considerada determinante no sucesso destes. Os processos de gestão do risco são definidos através de modelos, que especificam as atividades a realizar durante o projeto, com o objetivo de identificar, analisar, qualificar, eliminar ou minimizar os riscos ou os seus impactos. Tendo esta ideia em mente, este trabalho visa propor um modelo de gestão dos riscos que se revele adequado para projetos ágeis de *software*. Este modelo encontra-se implementado numa ferramenta de gestão de projetos de *software*.

Palavras-chave: Gestão de projetos, gestão do risco, desenvolvimento ágil de *software*.

Abstract

We live in a society dominated by the change, as such organizations have to adapt and adopt measures which support the new demands that are arising. Agile development ascended in order to respond to changing business requirements and given its popularity, agile risk management has become central, because agile methodologies by themselves don't give an answer to the risks that might arise in a software project. Risk is inherent in all software projects. To manage and control them represents gains for the development and success of projects. The risk management goal is to control, in a continue way, the risks that arise in all the phases of projects, and it's considered determinant to the projects' success. The risk management processes are defined through models that specify the activities to accomplish during the project, with the aim to identify, analyze, qualify, eliminate or minimize risks the risk impacts. With this in mind, this work aims to propose a risk management model, suited to agile development, in order to improve the existing risk management. This model is implemented in a project management software tool.

Keywords: Project management, risk management, agile development.

Índice

Índice de Figuras

Índice de tabelas

Lista de Abreviaturas

SEI	**Software Engineering Institute**
XP	**eXtreme Programming**
PMI®	**Project Management Institute**
ROI	**Return on Investment**
GQM	**Goal-Metric-Question**

Capítulo 1

Introdução

1.1 Contextualização

"If you don't actively attack the risks, they will actively attack you." [1]

Risco é uma constante nos projetos de *software* e representa uma variável importante no sucesso dos mesmos. Os impactos positivos frequentemente observados, devido à adoção de estratégias de gestão do risco, levaram a que algumas organizações de desenvolvimento de *software* dessem especial importância ao papel do risco face aos objetivos das mesmas, nomeadamente, a redução de custos, de atrasos e melhorias no desempenho [2]. Todavia, uma grande parte das organizações de desenvolvimento de *software* ainda tem como ideia base que o risco contribui para a instabilidade e ineficiência no desenvolvimento de projetos.

Kward e *Ibbs* [3][4] em 2000 identificaram a gestão do risco como a área menos aplicada entre oito áreas de conhecimento da gestão de projetos. Além disso, uma investigação conduzida por *Adams* e *Pinto* [5] levou à conclusão que ainda não tinha sido dada à gestão do risco atenção especial, sendo que esta também não tinha sido totalmente aceite pela comunidade de engenharia de *software*.

Em 2003, *Dedolph* [6] refere que a gestão do risco é de certo modo negligenciada, uma vez que as organizações, por norma, resistem à mudança. Esta resistência deve-se a várias razões, nomeadamente a gestão do risco parecer complicada, os recursos serem escassos face ao trabalho extra, as equipas serem recompensadas pela resolução de problemas e não pela sua prevenção e as ações de mitigação poderem exigir mudanças organizacionais ou de processo. O mesmo autor refere ainda que a gestão do risco se enquadra numa gestão proativa, em oposição à gestão reativa e requer esforço contínuo e boa comunicação de forma a ser eficaz.

Charrette em 2005 [7], referiu alguns fatores que representam causas para os projetos falharem repetidamente, entre os quais:

- objetivos do projeto irrealistas;
- estimativas incorretas de recursos necessários;
- pobre documentação do estado do projeto;
- requisitos do sistema mal definidos;
- falhas na comunicação entre os interessados no projeto;
- a falta de capacidade para lidar com a complexidade do projeto;
- o descuido relativo a boas práticas de desenvolvimento;
- falhas na gestão do risco;
- políticas dos *stakeholders*;

- pressões comerciais.

É necessário analisar o ambiente de negócio, a gestão técnica, a gestão do projeto e a cultura organizacional para chegar à origem das falhas de *software* que, por sua vez, conduzem ao insucesso dos projetos.

Em 2001, segundo um artigo publicado no *International Journal of Project Management* [8] a gestão do risco foi considerada um dos maiores temas de interesse na área de gestão de projetos, além disso foi designada uma das oito áreas principais do *Project Management Body of Knowledge* (PMBOK) pelo *Project Management Institute* (*PMI®*), a maior organização profissional dedicada à área de gestão de projetos.

"Risk management is a systematic approach for minimizing exposure to potential losses. It provides a disciplined environment for

• continuously assessing what could go wrong (i.e., assessing risks)

• determining which risks to address (i.e., setting mitigation priorities)

• implementing actions to address high-priority risks and bring those risks within tolerance."[9]

A gestão do risco preocupa-se não só com a identificação dos riscos, mas também em perceber os mesmos e delinear planos que minimizem o seu impacto nos projetos. Pode ser vista como um conjunto de passos que servem de suporte à equipa de *software* na gestão da incerteza [10]. É um processo que acompanha todo o projeto de desenvolvimento de *software* desde a fase de definição, planeamento, implementação e controlo.

Não podemos excluir o risco dos projetos, podemos apenas gerir e controlar o impacto que as suas consequências vão ter nos mesmos. Existem boas razões para os riscos serem geridos, a principal é o sucesso dos projetos, pois de acordo com a literatura, a gestão de riscos pode trazer inúmeros benefícios para as organizações, incluindo o aumento da confiança em alcançar os objetivos do projeto, melhorias nas hipóteses de sucesso, redução de surpresas, estimativas precisas e minimização dos esforços [11].

As constantes mudanças ao nível do negócio exigem o uso de metodologias de desenvolvimento de *software* que se adaptem facilmente. No desenvolvimento de projetos de *software* surgem riscos recorrentes e novos riscos, nomeadamente falhas no desempenho de tarefas, prazos e orçamentos elevados, falhas no desenvolvimento de soluções e mudanças frequentes nos requisitos [12].

O uso de abordagens ágeis tem vindo a crescer na última década e provou ser o processo mais usado na indústria de desenvolvimento de *software*. Desde 1990 [13], foram desenvolvidas metodologias ágeis, nomeadamente *eXtreme Programming* (XP), *Scrum*, *Crystal*, *Feature-Driven Development* (FDD), *Adaptative Software Development* e têm evoluído desde então. Apesar de serem ligeiramente diferentes, uma vez que são aplicadas mediante o contexto de cada projeto e darem mais ênfase a determinados valores, estas metodologias têm por base os mesmos princípios. Os métodos de desenvolvimento ágil de *software* surgiram como resposta à insatisfação das abordagens tradicionais e permitiram às equipas de desenvolvimento concentrarem-se mais no

software. Servem de suporte ao desenvolvimento de projetos, nos quais os requisitos mudam rapidamente e focam-se em entregas rápidas do produto. O facto de serem iterativas e darem ênfase ao envolvimento do cliente permite no fim de cada *sprint* a introdução de novos requisitos e alterações relevantes para a próxima iteração [14].

Ao longo dos vários anos foram propostos modelos que servem de suporte a todo o processo de gestão de riscos, porém são poucos os que se baseiam em abordagens ágeis.

Na secção seguinte são apresentados os principais objetivos deste trabalho.

1.2 Principais objetivos do trabalho

Dada a importância da gestão do risco e o crescente uso de metodologias ágeis em projetos de *software*, pretende-se com o presente trabalho definir um modelo de gestão do risco específico para projetos ágeis de software. Deste modo, foram definidos os seguintes objetivos principais:

- Conhecer os valores e princípios ágeis;
- Conhecer as principais metodologias ágeis (foco na SCRUM e XP);
- Conhecer os riscos existentes em projetos de desenvolvimento de *software* e identificar os que são mais relevantes em projetos de desenvolvimento ágil;
- Conhecer os modelos de gestão do risco existentes (revisão da literatura);
- Definir um modelo de suporte à gestão do risco em projetos ágeis;
- Validar a aplicação prática do modelo proposto, com base na realização de um estudo de caso.

Os resultados esperados são:

- Análise das principais metodologias ágeis e perceber o enquadramento com a gestão do risco (secção 2.1).
- Definição da lista de riscos presentes em projetos de desenvolvimento ágil (secção 2.2).
- Análise dos modelos de gestão do risco, considerando o desenvolvimento ágil (secção 3).
- Identificação das lacunas dos diferentes modelos na resposta ao desenvolvimento ágil (secção 3.4.2).
- Definição de um modelo de gestão do risco a utilizar em cenários de desenvolvimento ágil de *software* (secção 4).
- Análise dos resultados após o teste e validação do modelo (secção 5).

Na secção seguinte encontra-se a estrutura do presente documento.

1.3 Estrutura do documento

A presente dissertação encontra-se organizada em 6 capítulos:

- ✓ **Capítulo 1 – Introdução**

  Neste capítulo que agora se encerra, faz-se uma breve contextualização teórica do tema, sendo explicada a sua relevância no âmbito de gestão de projetos. Seguidamente são descritos os principais objetivos a atingir com este trabalho e é apresentada a estrutura do presente documento.

- ✓ **Capítulo 2 - Gestão de projetos ágeis: riscos e metodologias**

  No capítulo 2, abordam-se os principais pontos referentes ao desenvolvimento ágil de *software*, nomeadamente os principais valores e princípios ágeis. Seguidamente, apresentam-se algumas listas de riscos mais comuns em projetos de *software* referidas na literatura.

- ✓ **Capítulo 3 – Modelos para gestão dos riscos**

  No capítulo 3 faz-se uma revisão da literatura relativamente aos principais modelos que permitem gerir os riscos em projetos de *software*. No final do capítulo, faz-se uma análise comparativa dos modelos apresentados.

- ✓ **Capítulo 4 - Modelo de gestão do risco em projetos ágeis**

  No capítulo 4, apresenta-se uma proposta de um modelo para gerir os riscos em projetos ágeis de *software.* Inicialmente é feita uma breve introdução, seguindo-se a apresentação do processo proposto para gestão dos riscos, dados de entrada e saída do processo, papéis e responsabilidades, ferramentas e boas práticas. No final, faz-se o mapeamento entre o processo e as principais metodologias ágeis de *software*.

- ✓ **Capítulo 5 - Análise da aplicabilidade do modelo**

  No capítulo 5 faz-se um enquadramento da componente prática desenvolvida em colaboração com uma empresa. Seguidamente, é feito um mapeamento entre o processo e a componente prática e no final são definidas métricas qualitativas para medir a aplicabilidade do modelo.

- ✓ **Capítulo 6 - Conclusões e trabalho futuro**

  No capítulo 6 apresentam-se as principais conclusões da presente dissertação, bem como possibilidades de trabalho futuro.

- ✓ **Anexos**

  Em anexo encontra-se um esquema geral relativo à componente prática desenvolvida, bem como um questionário definido para a validação do modelo no contexto da componente prática desenvolvida.

Capítulo 2

Gestão de projetos ágeis: riscos e metodologias

2.1 Desenvolvimento ágil de *software*

O desenvolvimento ágil tornou-se bastante popular na indústria do *software*. Valores como rapidez, flexibilidade e a capacidade de resposta são as principais razões desta popularidade. Estes valores são fundamentais e definem a cultura da organização. Ao contrário dos métodos de desenvolvimento tradicionais, cujo foco é nos processos e ferramentas, esta abordagem dá especial ênfase às equipas, ao *software* funcional, à colaboração com o cliente e à resposta às mudanças, tornando-se uma mais-valia face a outras abordagens [15].

"(…) agile methods mean its "ability to endure in an environment of constant change and emerge with success" (…)"[15]

A definição oficial do desenvolvimento ágil de *software* foi proposta sob a forma de um manifesto, em Fevereiro de 2001, por um grupo de 17 peritos em desenvolvimento de *software*, que se reuniram e definiram um conjunto de boas práticas de desenvolvimento de *software*, dando origem ao Manifesto Ágil [15]. Os valores principais do manifesto são quatro [16]:

Indivíduos *e* ***interações*** *mais do que processos e ferramentas.*
Software funcional *mais do que documentação abrangente.*
Colaboração *com o cliente mais do que negociação contratual.*
Responder à mudança *mais do que seguir um plano.*

O Manifesto Ágil define explicitamente quatro valores (acima apresentados) que devem ser considerados ao longo do desenvolvimento ágil de *software*. Contudo, os quatro valores são suportados por doze princípios [16]:

1. *A nossa maior prioridade é, desde as primeiras etapas do projeto satisfazer o cliente através da entrega rápida e contínua de software com valor.*
2. *Aceitar alterações de requisitos, mesmo numa fase tardia do ciclo de desenvolvimento. Os processos ágeis potenciam a mudança em benefício da vantagem competitiva do cliente.*
3. *Fornecer frequentemente software funcional. Os períodos de entrega devem ser de poucas semanas a poucos meses, dando preferência a períodos mais curtos.*
4. *O cliente e a equipa de desenvolvimento devem trabalhar juntos, diariamente, durante o decorrer do projeto.*
5. *Desenvolver projetos com base em indivíduos motivados, dando-lhes o ambiente e o apoio de que necessitam, confiando que irão cumprir os objetivos.*

6 *O método mais eficiente e eficaz de passar informação para e dentro de uma equipa de desenvolvimento é através de conversa pessoal e direta.*

7 *A principal medida de progresso é a entrega de software funcional.*

8 *Os processos ágeis promovem o desenvolvimento sustentável. Os promotores, a equipa e os utilizadores deverão ser capazes de manter, indefinidamente, um ritmo constante.*

9 *A atenção permanente à excelência técnica e um bom desenho da solução aumentam a agilidade.*

10 *Simplicidade – a arte de maximizar a quantidade de trabalho que não é feito – é essencial.*

11 *As melhores arquiteturas, requisitos e desenhos surgem de equipas auto-organizadas.*

12 *A equipa reflete regularmente sobre o modo de se tornar mais eficaz, fazendo os ajustes e adaptações necessárias.*

Na Tabela 1 encontra-se um resumo dos princípios ágeis comuns a todas as metodologias ágeis.

Tabela 1 - Princípios ágeis: resumo [14].

Princípio	**Descrição**
Envolvimento do cliente	Os clientes devem ser envolvidos no processo de desenvolvimento, ajudando no fornecimento e priorização de novos requisitos do sistema e avaliação nas iterações do sistema.
Entrega incremental	O *software* é desenvolvido em incrementos e o cliente especifica os requisitos a serem incluídos em cada incremento.
Foco nas pessoas, e não no processo	Deve ser reconhecida e explorada a forma de trabalhar da equipa e não devem ser usados processos prescritivos.
Mudanças	Os requisitos do sistema podem mudar, desta forma deve projetar-se o sistema para facilitar a adaptação a mudanças.
Simplicidade	Deve haver uma maior concentração na simplicidade do *software* e deve ser eliminada a complexidade do sistema.

Os objetivos dos princípios do Manifesto Ágil centram-se em promover uma correta percepção acerca do que são os métodos ágeis e guiar as equipas de desenvolvimento do projeto, no sentido de perceberem se realmente estão a usar os métodos ágeis [17]. Este manifesto e todos os valores e princípios inerentes, representam a filosofia dos métodos ágeis que idealmente deve estar presente em todas as práticas propostas pelos métodos ágeis.

Os valores ágeis são a base das práticas e princípios ágeis. Existem diversas opiniões acerca do grau de importância destes na determinação das práticas e princípios considerados em certos projetos [15]. Além disso, as metodologias ágeis apesar de terem como base os mesmos valores e princípios atribuem uma importância diferente aos diversos valores.

Em 2000, *Beck* [18] referiu que os valores pelos quais a metodologia XP se rege são a comunicação, simplicidade, *feedback* e coragem. Este autor defende que a **comunicação** da equipa

de forma a cooperar eficientemente é essencial, uma vez que a aprendizagem entre os vários elementos pode evitar alguns erros outrora cometidos; a remoção da complexidade e a produção de soluções baseadas na **simplicidade**; o ***feedback*** regular da equipa pode ser propício a melhorias no *software* a desenvolver; e a **coragem** dos elementos da equipa, no sentido de incentivar a comunicação e a partilha de conhecimento sobre formas de melhorar determinadas tarefas. *Beck* e *Fowler* [19], em 2001, referiram valores como **foco**, **humildade**, **responsabilidade** e **motivação**. Em 2004, *Beck* [20] adicionou um novo valor, o **respeito** entre todos os elementos da equipa de desenvolvimento.

Schwaber & Beedle em 2002 [21] afirmaram que a metodologia Scrum considera cinco valores, nomeadamente **compromisso** por parte de todos os intervenientes do projeto para com as suas responsabilidades; **foco** da equipa nos objetivos de cada iteração; **acessibilidade** à lista de tarefas e prioridades estabelecidas; **respeito** por todos os intervenientes do projeto; e **coragem**, promovendo a confiança nos membros da equipa.

Tamer Madi, Zulkhari Dahalin e *Fauziah Baharom* em 2011 [15], fizeram uma análise ao conteúdo do manifesto ágil e aos comentários feitos por defensores do mesmo entre 2005 e 2011 e enumeraram alguns valores fulcrais a considerar nas abordagens ágeis. Os resultados mostram que os valores relacionados com as pessoas ocupam a primeira posição, seguindo-se os valores referentes ao processo. Analisando o manifesto ágil, os autores extraíram os valores apresentados na Figura 1, representados segundo três dimensões: **produto** (artefactos produzidos), **processo** (linhas de orientação usadas durante o desenvolvimento) e **pessoas** (todos os intervenientes no projeto de desenvolvimento) [12]. Software funcional diz respeito à dimensão produto. Foco no cliente, colaboração, aprendizagem e comunicação são valores relacionados com as pessoas. Flexibilidade, simplicidade, adaptação, sentido prático e naturalidade são valores relacionados com o processo.

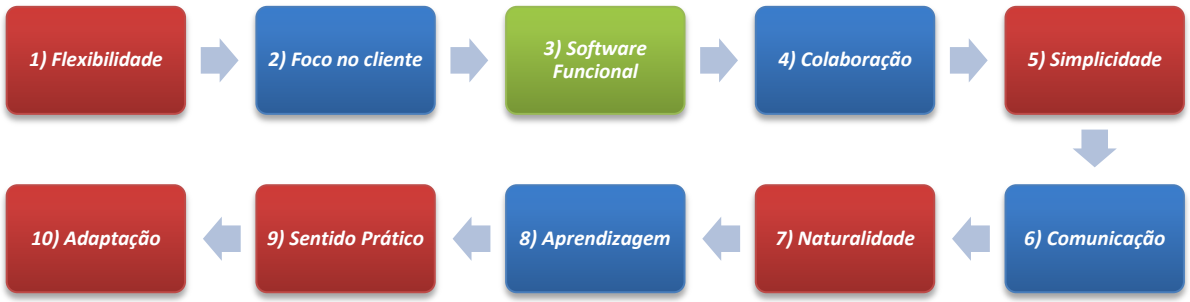


Figura 1 - Principais valores ágeis.

1. **Flexibilidade**: resposta rápida à mudança dos requisitos, o que encaixa com a incerteza e com as rápidas mudanças do ambiente de negócio, aumentando a satisfação do cliente.

2. **Foco no cliente**: o cliente é uma prioridade em qualquer ação do processo de desenvolvimento, deve ser envolvido no processo.

3. ***Software* Funcional**: o objetivo da equipa passa por entregas periódicas de artefactos que representam valor para o cliente.

4. **Colaboração**: a abordagem ágil dá especial importância às pessoas, às relações entre as mesmas e à partilha, comunicação entre as equipas de desenvolvimento.

5. **Simplicidade**: facilidade de implementação de metodologias ágeis, minimizando a complexidade e a burocracia dos métodos tradicionais.

6. **Comunicação**: Boa comunicação entre os membros da equipa que desenvolve o projeto, uns com os outros e para com o cliente é propícia à eliminação de erros.

7. **"Naturalidade"**: Mesmo sem saber muitos já usavam princípios ágeis para o desenvolvimento do projeto. A abordagem ágil é por isso, uma abordagem natural, espontânea.

8. **Aprendizagem**: abordagem ágil é vantajosa ao nível da aquisição de conhecimento, melhoria de capacidades, relações sociais através da aprendizagem diária e contínua entre os membros da equipa.

9. **Sentido prático**: desenvolvimento de soluções de acordo com as condições atuais e com as mudanças ao nível do negócio.

10. **Adaptação**: fácil adaptação às necessidades quer da equipa, quer do cliente.

Os opositores das metodologias ágeis afirmam que o foco destas é única e exclusivamente nas pessoas, ignorando o processo, contudo uma análise feita por *Tamer Madi, Zulkhari Dahalin* e *Fauziah Baharom* em 2011, mostra que existe um equilíbrio entre o foco nas pessoas e no processo [15].

Na secção seguinte, são apresentadas as metodologias ágeis mais populares no desenvolvimento ágil de *software*.

2.1.1 Principais metodologias ágeis de *software*: XP e Scrum

Como fora referido anteriormente, desde 1990 [13] foram desenvolvidas diversas metodologias ágeis e entre as mais populares encontram-se as metodologias *eXtreme Programming* (XP) e *Scrum*.

A metodologia XP é predominantemente focada nas fases referentes à implementação e testes de um projeto de *software*, enquanto a *Scrum* foi criada para suportar essencialmente a gestão

dos projetos [17]. Na Figura 2, encontra-se um esquema do ciclo de vida da metodologia XP, constituída por seis fases [22] :

- ***Exploration*:** o cliente define os *story cards* que devem ser incluídos na primeira *release* e ao mesmo tempo a equipa do projeto começa a familiarizar-se com as ferramentas, tecnologias e práticas que serão usadas no decorrer do projeto. Por norma, esta fase tem a duração de poucas semanas a poucos meses, tendo em conta fatores relacionados com a tecnologia a usar.

- ***Planning:*** são definidas as prioridades e é estabelecido um acordo com o cliente, tendo por base o esforço estimado para as funcionalidades a desenvolver. Esta fase tem a duração máxima de alguns dias apenas.

- ***Iterations to release:*** esta fase é composta por diversas iterações antes da primeira *release*. São criados testes funcionais pelo cliente, que são executados no final da cada iteração. Na última iteração o sistema está pronto para produção.

- ***Productionizing:*** nesta fase são necessários testes extra, bem como uma avaliação da performance do sistema antes de este ser entregue ao cliente. Aqui, podem ser definidas alterações que poderão ou não fazer parte da *release* atual.

- ***Maintenance:*** esta fase requer esforço para tarefas de suporte ao cliente, sendo que podem entrar novas pessoas para a equipa ou até mesmo a estrutura da equipa ser alterada.

- ***Death:*** esta fase é referente ao fecho, ou seja, quando o sistema satisfaz as necessidades do cliente, quando o sistema não corresponde ao resultado esperado ou até mesmo quando o custo do sistema toma proporções excessivas que não haviam sido planeadas.

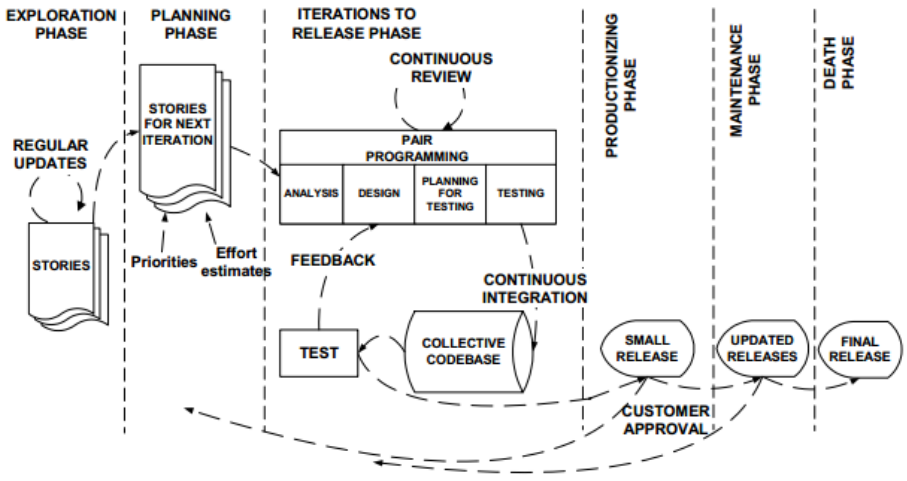


Figura 2 - Metodologia XP [22].

Na metodologia XP são definidos os seguintes papéis: *programmer*, *customer*, *tester*, *tracker*, *coach*, *consultant* e *manager* [22]. O *programmer* é responsável por manter o código simples e claro e pela escrita dos testes. O *customer* deve escrever as *stories* e os testes funcionais, decidir acerca do cumprimento dos requisitos e definir as suas prioridades em termos de implementação. O *tester* é responsável por ajudar o *customer* a escrever os testes funcionais e pela sua execução. O *tracker* traça estimativas, o progresso da equipa em cada iteração e analisa formas de melhorar as estimativas. O *coach* é responsável pelo processo como um todo e por guiar a equipa no processo. O *consultant* guia a equipa na resolução de problemas específicos e o *manager* é responsável por tomar as decisões.

A metodologia Scrum foi proposta em 1999, com o objetivo de mitigar os problemas associados a longos períodos de desenvolvimento sem feedback do cliente, algo que até à data era comum em diversos projetos de *software* [23]. Na Figura 3, encontra-se um esquema referente ao processo desta metodologia composta por três fases [22]*:*

- ***Pre-game:*** esta fase é dividida em duas subfases, nomeadamente *planning* e a*rchitecture/high level design*.

  Na subfase *planning* consta a definição do sistema a ser desenvolvido; é criada uma *product backlog list* que contém os requisitos, oriundos do cliente, dos programadores, entre outros; os requisitos são priorizados e é estimado o esforço necessário para a sua implementação; a *product backlog list* é constantemente atualizada, bem como as prioridades definidas e as estimativas de esforço; esta fase inclui a definição da equipa do projeto, ferramentas e recursos, avaliação dos riscos e controlo das questões críticas.

  Na subfase a*rchitecture/high level design* é planeada a arquitetura do sistema, tendo em conta os itens definidos no *Product Backlog*; é realizada uma reunião de revisão do *design*, sendo avaliadas e tomadas novas decisões com base nessa mesma revisão.

- ***Development:*** esta fase é também chamada de "*game phase*", é a parte ágil da abordagem Scrum, na qual o imprevisível é esperado. As variáveis ambientais e técnicas como tempo, qualidade, requisitos, recursos e até mesmo métodos de desenvolvimento são controladas durante o *Sprint*. Durante esta fase o sistema é desenvolvido em *Sprints* que representam ciclos iterativos, nos quais são desenvolvidas as funcionalidades e produzidos novos incrementos. Cada *Sprint* inclui as fases tradicionais de desenvolvimento de *software* e é planeado para que a sua duração seja entre uma semana a um mês.

- ***Post-game:*** esta fase é referente ao fecho de uma *release* e inclui tarefas relacionadas com a integração, testes ao sistema e documentação.

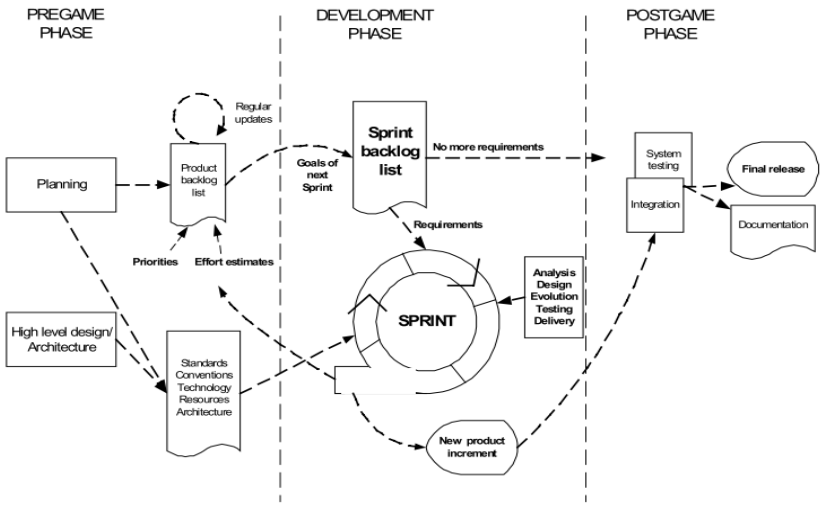


Figura 3 - Metodologia Scrum [22].

Na metodologia *Scrum* existem três papéis principais: *Product Owner, Team* e *Scrum Master* [24]. O *Product Owner* é responsável por maximizar o retorno de investimento (ROI), através da identificação e priorização das características do produto. A equipa (*Team*) é responsável por desenvolver o produto e segundo esta metodologia deve ser organizada, com um elevado grau de autonomia e responsabilidade. O *Scrum Master* é responsável por eliminar impedimentos organizacionais, moderar reuniões e garantir que a equipa é bem-sucedida.

Na secção seguinte são abordados os principais riscos em projetos ágeis de *software*.

2.2 Riscos em projetos ágeis de *software*

"Identifying and dealing with risks early in development lessens long-term costs and helps prevent software disasters. It is easy to begin managing risks in your environment." [25]

A gestão do risco em projetos de *software* é uma atividade complexa, uma vez que lida com eventos incertos e com as causas e impacto dos mesmos. Fazer uma previsão dos riscos associados a um projeto nem sempre é uma tarefa trivial. Posto isto, vários investigadores sugeriram listas de riscos de projetos de *software* como medida para facilitar a gestão do risco [26].

Desde 1981 que são estudados os riscos em projetos de *software* baseados em diferentes dimensões. *McFarlan* [27], em 1981, introduziu três dessas dimensões, nomeadamente o tamanho do projeto, a experiência tecnológica e a estrutura do projeto.

Foi em 2004 que *L. Wallace* e *M. Keil* [28] categorizaram os riscos em seis dimensões, que serão consideradas ao longo da presente dissertação: **utilizador**, **requisitos**, **complexidade do projeto**, **planeamento e controlo**, **equipa** e **ambiente organizacional**. Na Tabela 2 encontram-se riscos identificados pelos autores referidos, bem como a respetiva dimensão a que correspondem.

Note-se que ao longo dos anos vários foram os esforços, no sentido de facilitar a gestão do risco em projetos, sendo que uma das medidas foi a criação de listas de riscos de *software*. Na

secção 2.2.1 são apresentadas algumas das listas produzidas ao longo dos anos. Todavia, é importante ter presente que cada projeto é único, como tal os riscos podem variar de projeto para projeto e a criação das referidas listas serve apenas para se perceber quais os riscos mais comuns detetados ao longo de vários projetos.

Tabela 2 - Seis dimensões dos riscos de software [21].

Dimensão do Risco	**Risco de Software**
Utilizador (U)	Resistência dos utilizadores à mudança
	Conflitos entre utilizadores
	Utilizadores com atitudes negativas perante o projeto
	Utilizadores não comprometidos com o projeto
	Falhas de cooperação entre utilizadores
Requisitos (R)	Mudanças frequentes dos requisitos
	Falhas na identificação dos requisitos do sistema
	Requisitos do sistema pouco claros
	Requisitos do sistema incorretos
Complexidade do projeto (C)	Uso de novas tecnologias
	Elevado nível de complexidade
	Tecnologia "imatura"
	Uso de tecnologias que ainda não foram usadas anteriormente
Planeamento e controlo (PC)	Falhas na gestão da tecnologia do projeto
	Falhas na monitorização do projeto
	Estimativas incorretas dos recursos necessários
	Fraco planeamento do projeto
	Definição dos pontos de controlo pouco clara
	Inexperiência dos gestores do projeto
	Comunicações pouco eficazes
Equipa (E)	Inexperiência de membros da equipa
	Formação inadequada de membros da equipa
	Falhas nas capacidades exigidas para o projeto
Ambiente organizacional (AO)	Mudanças na gestão organizacional durante o projeto
	Políticas organizacionais com efeitos negativos no projeto
	Ambiente organizacional instável
	Reestruturação da organização durante o projeto

O risco existe em todos os projetos de *software*, e felizmente na maioria dos casos pode e deve ser minimizado. Em 2009, *Ellen Gottesdiener* [29], referiu que um dos maiores riscos em

qualquer tipo de projetos está relacionado com os requisitos do sistema, nomeadamente, requisitos pouco claros, falhas na sua implementação, falhas no envolvimento do cliente, entre outros. Este autor apontou também alguns riscos, que segundo o mesmo existem e podem ser minimizados recorrendo às metodologias ágeis. Esses riscos são [29]:

- expectativas do cliente irrealistas e adição de componentes caros e desnecessários ao sistema;
- falhas no envolvimento do cliente;
- lacunas ao nível da análise do impacto;
- alterações no âmbito do projeto;
- falhas na implementação dos requisitos;
- novos processos e ferramentas.

O mesmo autor revela ainda que o facto de serem feitas entregas em cada iteração, permite perceber quais as expectativas do cliente e obter um feedback do decorrer do projeto. Além disso, um dos maiores erros citados é a falta de compromisso por parte do cliente, o que em abordagens ágeis não é frequente, uma vez que uma das pré-condições do desenvolvimento ágil é a participação do cliente em todas as iterações.

Numa outra perspectiva, em 2012, *Haneen Hijazi, Thair Khdour* e *Abdulsalam Alarabeyyat* [30], afirmaram que o desenvolvimento ágil carece de sugestões detalhadas para a gestão do risco. Além disso, apresentaram algumas fontes, que consideram ser as maiores fontes de risco no desenvolvimento ágil de *software*, tais como[30]:

- sistemas muito grandes;
- grandes equipas de desenvolvimento;
- elevada confiança no fator humano;
- representação inapropriada do cliente;
- ambientes de desenvolvimento distribuídos;
- deslizamento do âmbito.

O sucesso do desenvolvimento de projetos é o interesse principal da organização, uma vez que as falhas e o insucesso implicam gastos acrescidos. É essencial para os gestores de projetos a aquisição de conhecimento no âmbito dos riscos, bem como formas para minimizar os mesmos, levando a bom porto a tarefa de concluir o projeto com sucesso. O objetivo da gestão do risco é identificar os riscos e tomar medidas para minimizar os seus efeitos no projeto. Note-se que a gestão eficiente dos riscos é uma das áreas mais importantes de gestão de projetos [31].

2.2.1 Listas de riscos de *software*

Durante o seu percurso profissional, *Barry Boehm* teve oportunidade de observar diversos gestores, inclusive perceber quais as principais características que distinguem os gestores de projetos com mais e menos sucesso. Chegou à conclusão que um dos padrões mais comum era o

facto de os gestores com elevado grau de sucesso serem bons gestores do risco. Estes, apesar de não usarem conceitos inerentes à gestão do risco bastante comuns, nomeadamente a identificação, o planeamento, a avaliação ou até mesmo a monitorização do risco, já recorriam ao conceito de exposição do risco (*risk exposure*) - **RE** -, sendo que este era uma forma de orientação nas suas prioridades e ações. Exposição do risco é definida pela relação entre a probabilidade de resultados pouco favoráveis - **P(UO)** - e a perda das partes afetadas quando o resultado não é o pretendido - **L(UO)** [25].

$$\boldsymbol{RE} = \boldsymbol{P}(\boldsymbol{UO}) \times \boldsymbol{L}(\boldsymbol{UO})$$

Um aspeto importante é a definição de resultado pouco favorável, visto que os projetos envolvem diversas classes de participantes, todos diferentes uns dos outros, todavia com critérios de satisfação diversificados. Para os clientes e para os que desenvolvem o projeto, ultrapassar o orçamento e prazo previstos são critérios de insatisfação. Já para os utilizadores os critérios serão um produto com falhas de funcionalidades, segurança ou desempenho. Enquanto que para os que terão de fazer a manutenção do produto desenvolvido a baixa qualidade do *software* será um sinal de insatisfação. Estes componentes de insatisfação fornecem como resultado, uma lista para identificação e avaliação de pontos de risco.

Em 1991, *Barry Boehm* [25][26], apresentou as dez maiores fontes de risco em projetos de *software* listadas na Tabela 3.

Tabela 3 - Lista dos 10 riscos de *Software* mais importantes, *Barry Boehm* [25] [26]

Posição	Risco de Software	Dimensão
1	Falhas no pessoal	E
2	Prazos e orçamentos irrealistas	PC
3	Desenvolvimento de funcionalidades e propriedades erradas	R
4	Desenvolvimento da interface de utilizador errada	R
5	Adição de componentes caros e desnecessários ao sistema	PC
6	Fluxo contínuo de alterações aos requisitos	R
7	Falhas em componentes fornecidos externamente	PC
8	Falhas em tarefas realizadas externamente	PC
9	Falhas no desempenho em tempo real	C
10	Falta de aptidões em ciência de computadores	C

Note-se que da lista apresentada, e tendo em conta as dimensões anteriormente referidas, pode verificar-se que os riscos associados ao planeamento e controlo (PC) são mais frequentes. O mesmo não acontece com os riscos referentes à equipa.

Outras listas de riscos de *software* surgiram, nomeadamente em 2002 por *Addison* e *Vallabh*, [26] em 2007, por *Han* e *Huang* [7] e em 2008 por *Pare et Al* [26]. Nas tabelas abaixo encontram-se listados os riscos de *software* apresentados pelos autores acima referenciados.

Tabela 4 - Lista dos 10 riscos de *Software* mais importantes, segundo *Addison* e *Vallabh*[26].

Posição	**Risco de Software**	**Dimensão**
1	Objetivo/âmbito mal entendidos	PC
2	Falhas na interpretação dos requisitos	R
3	Incapacidade de envolver o utilizador	U
4	Falta de um grupo de gestão sénior	AO
5	Desenvolvimento de funcionalidades erradas	R
6	Prazos e orçamentos irrealistas	PC
7	Constante mudança dos requisitos	R
8	Falta de conhecimento e /ou capacidades	E
9	Falta de eficácia na metodologia de gestão do projeto	PC
10	Adição de componentes caros e desnecessários ao sistema	PC

Tabela 5 - Lista dos 10 riscos de *Software* mais importantes, segundo *Han* e *Huang* [7].

Posição	**Risco de Software**	**Dimensão**
1	Mudanças nos requisitos do sistema	R
2	Falhas na identificação dos requisitos do sistema	R
3	Requisitos do sistema pouco claros	R
4	Falta de eficácia na metodologia de gestão do projeto	PC
5	Requisitos do sistema errados	R
6	Falhas no planeamento do projeto	PC
7	Estimativas incorretas dos recursos necessários	PC
8	Projeto envolve o uso de novas tecnologias	C
9	Falhas na correta monitorização do projeto	PC
10	Políticas organizacionais com efeitos negativos no projeto	AO

Tabela 6 - Lista dos 10 riscos de *Software* mais importantes, segundo *Pare et Al*.[26].

Posição	**Risco de Software**	**Dimensão**
1	Falhas nas atitudes positivas para com o projeto	E
2	Falta de compromisso por parte da gestão de topo	U
3	Falhas na percepção da utilidade do sistema	U
4	Ambiguidades no projeto	PC
5	Desalinhamento do sistema face às práticas e processos locais	PC

6	Conflitos ou jogos políticos	AO
7	Falta de conhecimento e/ou capacidades	E
8	Mudanças nos membros da equipa	E
9	Instabilidade organizacional	AO
10	Recursos insuficientes	PC

Em 2008, António Miguel no seu livro "*Gestão de Projetos de Software*" apresentou uma lista de riscos de projetos de *software* portugueses [32]:

- falha na obtenção de compromisso do utilizador;
- âmbito/objetivos mal compreendidos/pouco claros;
- falta de compromisso da gestão de topo para com o projeto;
- falhas no envolvimento do utilizador;
- planeamento inadequado ou inexistente;
- alterações no âmbito/objetivos do projeto;
- indefinição de papéis e responsabilidades;
- recursos insuficientes / inadequados;
- conflitos entre departamentos;
- falha na gestão das expectativas dos utilizadores;
- dependências complicadas em projetos multifornecedor;
- alterações nos utilizadores ou na gestão de topo;
- falha dos parceiros externos;
- ausência de uma metodologia de desenvolvimento eficaz;
- não congelamento dos requisitos.

Esta lista encontra-se por ordem decrescente de importância e é o resultado de um trabalho de investigação efetuado em Portugal pelo autor do livro, tendo por objetivo obter uma lista de riscos na óptica dos chefes do projeto [32].

2.2.2 Quais os riscos mais frequentes?

Após a revisão da literatura, pode concluir-se acerca dos riscos mais comuns em projetos de *software*. Abaixo são apresentados os riscos mais frequentes mencionados em todas as listas apresentadas [26]:

- Falhas na interpretação dos requisitos;
- Falta de gestão do compromisso e suporte;
- Falta de envolvimento do utilizador;
- Falha no compromisso com o utilizador;
- Falha na gestão das expectativas do utilizador final;

- Mudanças nos requisitos;
- Falta de uma metodologia eficaz na gestão do projeto.

Apesar de existirem riscos comuns entre todas as listas apresentadas, pode verificar-se que fatores como a cultura, a dimensão temporal, os métodos de pesquisa e as áreas de aplicação exercem bastante influência na definição destas listas, como tal devem ser considerados.

Neste capítulo foram apresentadas algumas das listas mais comuns em projetos de *software*, no capítulo seguinte são analisados os principais modelos que dão resposta à gestão de riscos em projetos de *software*.

Capítulo 3

Modelos para gestão dos riscos em projetos de *software*

3.1 Introdução

Os modelos relativos ao processo de gestão do risco especificam o conjunto de passos a seguir. Tipicamente especificam atividades individuais e a sequência destas para uma gestão eficaz. Além disso, sugerem também técnicas e ferramentas de suporte que são usadas em cada passo do processo. O processo é executado iterativamente ao longo do projeto, sendo feita a gestão dos riscos recorrentes e dos novos que surgem à medida que o projeto avança e as circunstâncias se alteram [33].

Na literatura são vários os modelos que especificam o processo de gestão do risco, sendo que os dois mais importantes em engenharia de *software* estão associados a *Barry Boehm* e ao PMBOK Guide® [33]. Existem outros modelos bastante influentes que podem ser encontrados no *Software Engineering Institute*, em *standards* e na literatura dita académica. Contudo, apesar de existirem vários modelos a descrever todo este processo, existem atividades dentro do mesmo que são consideradas as atividades principais. A Figura 4 apresenta três atividades base a considerar [34]:

- **Avaliar o risco** – transformar as preocupações em riscos distintos que são documentados e analisados corretamente;

- **Planear o risco –** determinar a abordagem a seguir na resolução dos riscos e criar um plano para implementação da mesma;

- **Controlar o risco –** tratar de cada risco implementando e acompanhando o plano definido para o seu controlo.

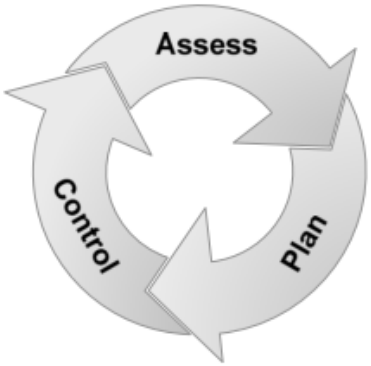


Figura 4 - Principais atividades da gestão do risco. [34]

Uma das maiores contribuições destes modelos é o facto de orientarem o curso das ações relativas à gestão do risco ao longo de todo o projeto. Contudo, são necessárias caraterísticas como

competência, bom senso e persistência para aplicar eficientemente o modelo, bem como as ferramentas e técnicas associadas [33].

Nos anos 80 *Barry Boehm* [35], introduziu o conceito de gestão do risco na indústria de *software* e definiu um modelo para a gestão do risco, apresentado na Figura 5.

Em 1994, *Richard Fairley* [36] sugere sete passos para a gestão do risco:

1) identificar os fatores de risco;
2) avaliar a probabilidade e o impacto dos riscos;
3) desenvolver estratégias para minimizar os riscos identificados;
4) monitorizar os fatores de risco;
5) invocar planos de contingência;
6) gerir a "crise";
7) recuperar da "crise".

Em 1997, *Kliem* e *Ludin* [37], no seu livro "*Reducing Project Risk*", descrevem o processo em quatro fases: identificação, análise, controlo e *reporting*. Ainda em 1997, *Chapman* e *Ward* [38], definem um processo genérico em nove fases, nomeadamente:

1) definição dos aspetos chave do projeto;
2) foco numa abordagem estratégica para a gestão do risco;
3) identificar as fontes de risco;
4) estruturar a informação sobre os pressupostos do risco e relações;
5) atribuir responsabilidades;
6) estimar a extensão da incerteza;
7) avaliar a magnitude dos vários riscos;
8) planear estratégias de resposta;
9) gerir através da monotorização e controlo.

Apesar das diversas propostas de modelos para gestão do risco, tal como referido acima, existem três modelos bastante importantes, que foram considerados ao longo do presente trabalho, nomeadamente, o modelo proposto por *Barry Boehm*, pelo *Software Engineering Institute* (*SEI*) e pelo *Project Management Institute* (*PMI®*). Todos estes modelos foram desenvolvidos por autores e entidades bastante conceituadas na área. Assim, nas subsecções seguintes será feita uma descrição de cada um deles.

3.1.1 Modelo proposto por Barry Boehm

Segundo *Barry Boehm* [39], a gestão do risco é subdividida em dois passos, a avaliação e o controlo.

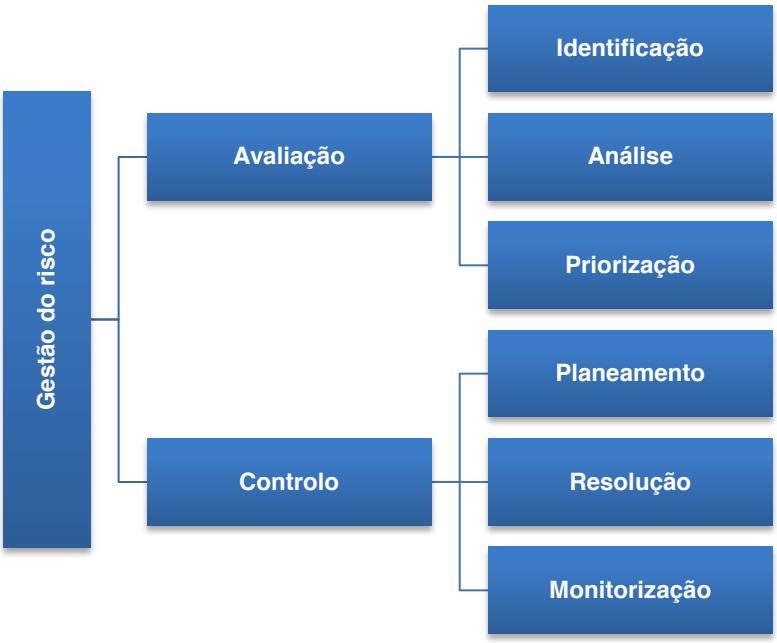


Figura 5 - Passos para a gestão do risco de *software,* segundo *Barry Boehm* [39].

.

A avaliação do risco envolve atividades como a identificação, a análise e a priorização dos riscos. A **identificação** dos riscos produz uma lista de itens de risco que poderão comprometer o sucesso do projeto. Tipicamente, as técnicas associadas à identificação dos riscos incluem *checklists*, comparação com experiências anteriores e decomposição. A **análise** avalia a probabilidade de perda e o impacto da mesma para cada item da lista, além disso avalia a interação dos itens da lista. As técnicas nesta atividade incluem desempenho e custo de modelos, análises de decisão estatísticas e análise do factor qualidade (critérios como confiança, eficácia e segurança são avaliados). Na **priorização** os itens de risco identificados e analisados são priorizados, sendo o resultado uma lista ordenada. Nesta atividade são usadas técnicas como análise de exposição ao risco, análise custo-benefício e técnica de *Delphi* [39].

No controlo dos riscos as atividades passam por planear, delinear estratégias de resolução e monitorizar os riscos. O **planeamento** serve para a preparação da resolução e/ou minimização de cada risco. Frequentemente, as técnicas usadas nesta atividade incluem *checklists* de técnicas de resolução dos riscos, análise custo-benefício e definição de planos *standards* de gestão do risco. Na **resolução**, os riscos são resolvidos e/ou eliminados. As técnicas mais comuns nesta atividade passam pela criação de protótipos, simulações e desenvolvimento incremental. A **monitorização** envolve o acompanhamento do progresso do projeto na resolução dos diversos riscos. Nesta última atividade as técnicas usadas passam pela criação de pontos de controlo, pela definição de listas de riscos revistas a cada semana, mês ou até mesmo nos pontos de controlo referidos, seguidas de meios para a avaliação do item de risco e ações de correção [39].

3.1.2 SEI Continuous Risk Management (CRM) model

Desde janeiro de 1990, que o *Software Engineering Institute* (*SEI*) tem vindo a estudar formalmente a gestão do risco, como forma eficiente de melhorar o sucesso do *software* [40]. Com base em estudos desenvolvidos, o *SEI* criou um modelo designado *Continuous Risk Management (CRM)* [35]. Este fornece um ambiente propício à tomada de decisão, mais precisamente avaliação contínua, priorização e implementação de estratégias de resposta aos riscos [40]. Divide-se em cinco fases distintas: identificação, análise, planeamento, monitorização e controlo.

Quando aplicado corretamente o CRM pode ser benéfico, na medida em que ajuda a prevenir os problemas mesmo antes destes surgirem, através da identificação dos potenciais problemas e tratamento dos mesmos; melhora a qualidade do produto, focando-se nos objetivos do projeto e em aspetos prejudiciais ao mesmo; permite um melhor uso dos recursos; e promove o trabalho em equipa, envolvendo os membros da equipa em todos os níveis do projeto [40].

Este modelo tem por base sete princípios, como visão global, estratégia ativa, ambiente propício à comunicação, processo contínuo, perspetiva do produto centralizada, coordenação da equipa e cooperação [35].

Na Figura 6 é apresentado o paradigma relativo à gestão do risco definido pelo SEI. Cada função neste paradigma é composta por um conjunto de atividades apoiadas por processos, métodos e ferramentas que valorizam e melhoram a comunicação e o trabalho de equipa.

Figura 6 - *Continuous Risk Management (CRM), SEI* [40].

Na **identificação** procura-se perceber quais são os riscos antes mesmo de estes se tornarem num grave problema. Na **análise** é avaliado o impacto e a probabilidade dos riscos, e estes são depois classificados e priorizados. Com o **planeamento** são definidas estratégias de minimização dos riscos, quer individualmente, quer de grupos de riscos. A fase que se segue é a **monitorização**, e tal como o próprio nome indica, os indicadores de possíveis riscos e os planos de minimização são monitorizados. De seguida o **controlo**, onde são analisados os dados e são tomadas decisões face aos mesmos. Por fim, a **comunicação**, que é um fator chave neste modelo, um elo de ligação entre todas as outras fases [40].

Na Tabela 7 é apresentado um breve resumo de cada fase, ou seja, são indicados os dados de entrada, os dados de saída e exemplos de técnicas e/ou ferramentas usadas.

Tabela 7- Caraterização das fases do CRM.

Fases	Dados de entrada	Dados de saída	Técnicas/Ferramentas
Identificação	Incertezas individuais; Incertezas de grupo; Dados do projeto.	Informação sobre cada risco, mediante o contexto; Lista de riscos.	*Brainstorming*, relatórios sobre os riscos periódicos, entrevistas e questionários.
Análise	Output da fase anterior.	Informações sobre o risco atualizadas (contexto, impacto, probabilidade, prazo, classificação e categoria). Lista de riscos priorizada.	Folhas de cálculo com dados sobre os riscos.
Planeamento	Output da fase anterior. Recursos. Objectivos e restrições do projeto.	Informações sobre o risco atualizadas (…, plano de abordagem). Planos de ação.	Folhas de cálculo com dados sobre os riscos.
Monitorização	Output da fase anterior. Recursos. Riscos e planos de minimização. Dados do projeto.	Informação sobre o risco atualizada (…,estado). Relatórios de estado, contendo os riscos e os respetivos planos de minimização.	Diagramas de fluxo de dados sobre o planeamento de decisões.
Controlo	Output da fase anterior. Dados do projeto.	Informação sobre o risco atualizada (…,decisão de controlo). Decisões de controlo.	Relatórios sobre o estado, gráficos, folhas de cálculo com dados sobre os riscos.
Comunicação	Dados sobre os riscos. Planos de minimização	(não aplicável)	(não aplicável)

Este processo apresenta um ponto interessante relacionado com abordagens ágeis, nomeadamente os princípios que tem por base. Isto significa que tal como em abordagens ágeis, este processo dá ênfase à comunicação, à cooperação e ao trabalho em equipa, o que é favorável à aprendizagem e melhoria contínua. Por outro lado, ao longo do desenvolvimento de um projeto, pode

revelar ser complexo e com bastante documentação associada, o que se torna um ponto desfavorável face ao desenvolvimento ágil.

3.1.3 Modelo proposto no PMBOK Guide®

Um dos modelos mais populares para a gestão do risco foi definido no *PMBOK Guide*®, em 2004, e define seis fases distintas [32]:

1) **Planeamento da gestão dos riscos** – decidir o modo como abordar, planear e executar as atividades de gestão dos riscos para o projeto;
2) **Identificação dos riscos** – determinação dos riscos que poderão afetar o projeto, e documentação das respetivas características;
3) **Avaliação qualitativa dos riscos** – priorização dos riscos para posterior análise ou ação, através da avaliação e combinação das respetivas probabilidades de ocorrência e impacto;
4) **Avaliação quantitativa dos riscos** – análise numérica do efeito dos riscos identificados sobre os objetos globais do projeto;
5) **Planeamento de resposta aos riscos** – desenvolvimento das opções e ações destinadas a aproveitar as oportunidades e a reduzir as ameaças aos objetivos do projeto;
6) **Monitorização e controlo dos riscos** – monitorizar os riscos identificados, controlar o risco residual, identificar novos riscos, executar planos de resposta aos riscos e avaliar a sua eficácia ao longo do ciclo de vida do projeto.

Na Figura 7 encontra-se um esquema relativo ao processo de gestão do risco definido no *PMBOK Guide*®.

Na fase de planeamento da gestão dos riscos, o objetivo é definir uma estratégia para o projeto e decidir de que forma é que os processos para gestão dos riscos serão executados e serão integrados com outras atividades de gestão do projeto. Este planeamento deve ser feito o mais cedo possível e as atividades de gestão do risco devem ser integradas no planeamento do projeto. Além disso, deve estabelecer o tipo e nível de detalhe do risco e deve também indicar o esforço necessário para aplicar os processos de gestão do risco. O planeamento encontra-se sujeito a algumas alterações no decorrer do projeto e deve conter elementos como [41]:

- descrição do projeto;
- o âmbito e os objetivos da gestão do risco;
- a descrição da metodologia usada;
- de que forma é que se encontra organizada a gestão do risco, por exemplo, os papéis e as responsabilidades atribuídas aos mesmos;
- ferramentas a serem usadas;
- planos de comunicação.

Após o plano estar concluído segue-se a fase de identificação dos riscos que poderão interferir com os objetivos delineados para o projeto. O facto de alguns riscos não serem explícitos, faz com que esta fase seja iterativa. A identificação dos riscos deve ser feita o mais cedo possível, deve ser iterativa ao longo de todo o ciclo de vida do projeto, pode ser repetida a qualquer momento, não deve ficar limitada a revisões regulares e deve ser objetiva [41].

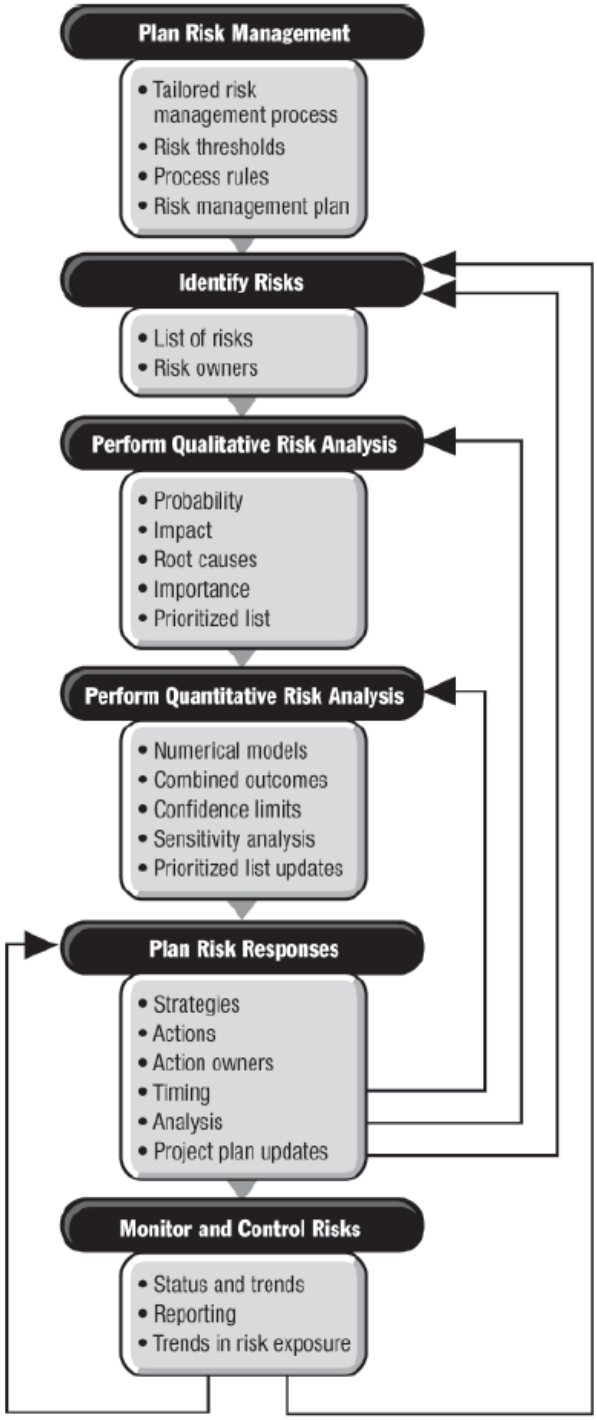


Figura 7 - Fluxo do processo da gestão do risco [41].

A fase que se segue é a avaliação qualitativa dos riscos, nesta é analisado e avaliado cada risco, de seguida os riscos são priorizados tendo por base a probabilidade e o impacto do mesmo nos objetivos do projeto. Um dos passos na análise passa pela categorização dos riscos de acordo com as suas fontes e/ou causas. A identificação de consequências comuns de grupos de riscos permite a identificação de áreas de maior exposição ao risco, facilitando as estratégias de resposta face a essas mesmas áreas. Após a aplicação de métodos de análise qualitativa, consegue perceber-se quais os riscos mais influentes nos objetivos do projeto. Os riscos marcados com elevada prioridade terão especial atenção na fase de planeamento de resposta aos riscos [41].

Após ser obtida uma lista de riscos priorizados, é feita uma análise quantitativa, que fornece uma estimativa numérica do efeito dos riscos nos objetivos do projeto [41]. Note-se que esta fase nem sempre é necessária em todos os projetos, uma vez que pode representar excesso de informação, mais concretamente quando se trata de projetos de pequena dimensão. Desta forma, no plano que é feito inicialmente, deve ser justificada esta fase, tendo por base o esforço necessário para a mesma.

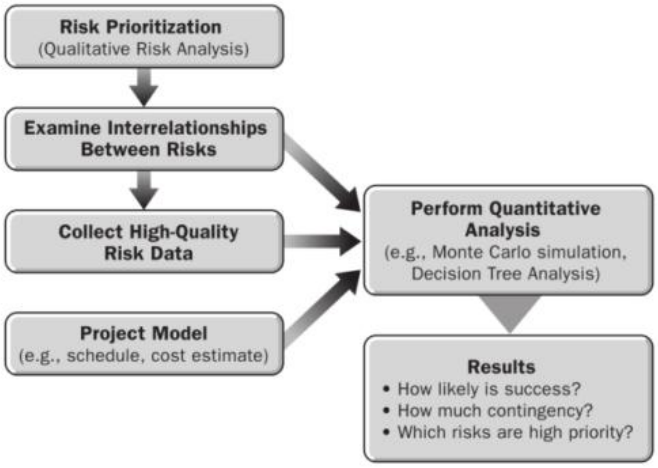


Figura 8 - Estrutura da análise quantitativa [41].

Segue-se a fase de planeamento de resposta aos riscos cujo objetivo é delinear ações de resposta aos riscos que aumentarão a probabilidade de sucesso do projeto, cumprindo com as restrições organizacionais e do projeto. Deve haver concordância entre os participantes do projeto mediante ações que alterem o orçamento, prazos, recursos e âmbito do projeto. Cada risco deve ser atribuído a um *risk owner*, ficando este responsável por gerir a resposta ao risco e planear, se necessário, respostas adicionais. Na Figura 9 são apresentados os fatores importantes para o sucesso nesta fase e a Figura 10 apresenta o conjunto de passos que a constituem [41].

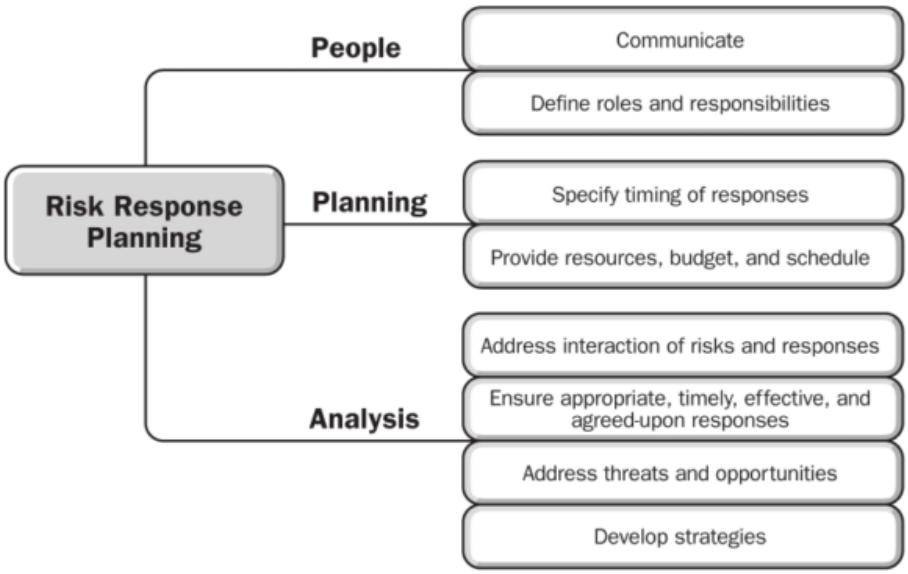


Figura 9 - Fatores de sucesso do planeamento de resposta aos riscos [41].

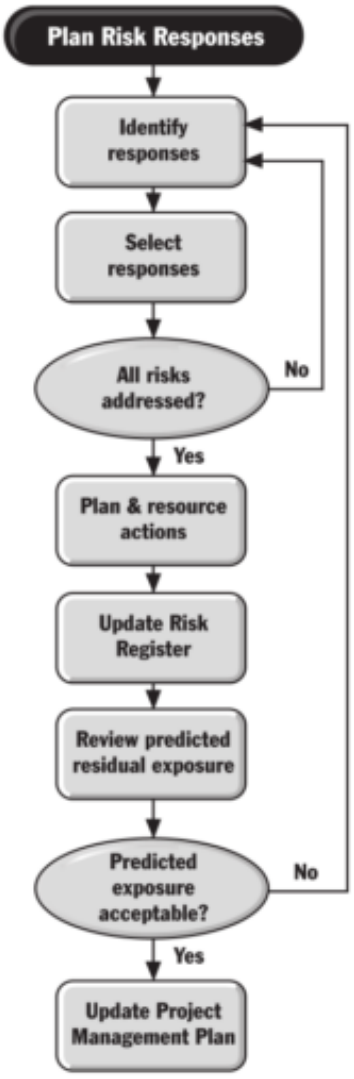


Figura 10 - Conjuntos de passos envolvidos na fase de planeamento de resposta aos riscos [41].

Por último, a fase de monitorização e controlo, que tem como objetivos principais monitorizar os riscos identificados, identificar novos riscos, assegurar que os planos de resposta são executados no tempo apropriado e avaliar a sua eficiência ao longo do ciclo de vida do projeto [41].

Na Tabela 8 é feita uma breve descrição de cada uma das fases deste modelo, nomeadamente, os dados de entrada e de saída, bem como algumas técnicas e/ou ferramentas usadas.

Tabela 8 - Caraterização das fases do modelo definido no *PMBOK Guide*® [41].

Fases	**Dados de Entrada**	**Dados de Saída**	**Técnicas/Ferramentas**
Planeamento	Informação sobre o âmbito do projeto. Plano de gestão de custos, prazos e comunicações. Fatores sobre o ambiente empresarial. Avaliação dos processos organizacionais.	Plano de gestão dos riscos.	Reuniões de planeamento e análise.
Identificação	Plano de gestão dos riscos. Estimativa do custo e duração das atividades. Base do âmbito. Informação dos *stakeholders*. Plano de gestão dos custos, prazos, qualidade. Documentos do projeto. Fatores sobre o ambiente empresarial. Avaliação dos processos organizacionais.	Informações sobre o risco (*Risk register*).	Análise dos pressupostos e restrições, b*rainstorming,* diagramas de causa-efeito, c*hecklist*, técnica de *Delphi*, entrevistas, técnica do grupo nominal, questionários, *Risk Breakdown Structure* (RBS), análise SWOT.
Análise qualitativa	*Risk register*. Plano de gestão dos riscos. Informações sobre o âmbito do projeto. Avaliação dos processos organizacionais.	*Risk register* atualizado.	Matriz de probabilidade e impacto, *Analytic Hierarchy Process* (AHP).

Análise quantitativa	*Risk register*. Plano de gestão dos riscos, custos, prazos. Avaliação dos processos organizacionais.	*Risk register* actualizado.	Árvores de decisão, valor monetário esperado (VME), simulação de Monte Carlo.
Estratégias de resposta	*Risk register*. Plano de gestão dos riscos. Riscos relacionados com decisões contractuais.	*Risk register* atualizado. Atualizações do plano do projeto. Atualizações do documento do projeto.	*Brainstorming*, *Checklists*, planos de contingência, árvores de decisão, técnica de *Delphi*, VME, entrevistas, técnica do grupo nominal, análise do cenário.
Monitorização e Controlo	*Risk register*. Plano de gestão do projeto. Informação sobre o desempenho do trabalho. Relatórios de desempenho.	*Risk register* atualizado. Avaliação dos processos organizacionais atualizada. Pedidos de alterações. Plano de gestão do projeto atualizado. Documento do projeto atualizado.	Avaliação do risco, reuniões, análise da variação dos riscos.

Este processo é bastante completo quando comparado com os anteriores, uma vez que inclui como primeira fase o planeamento detalhado e completo que servirá de base para as fases que se seguem; a análise dos riscos é subdividida em dois tipos, qualitativa e quantitativa, o que fornece mais informação dos riscos que serão tratados; e define papéis de utilizadores, por exemplo, o *risk owner*. Este último aspeto pode ser interessante para o desenvolvimento ágil, que também delega em cada metodologia, responsabilidades através da definição de papéis de utilizadores.

Contudo, tal como fora referido no modelo proposto pelo SEI, este também exige bastante documentação e ao longo do ciclo de vida de um projeto poderá tornar-se complexo, o que se pode tornar numa desvantagem considerando a abordagens ágeis.

3.2 Modelo de integração entre a gestão do risco e o desenvolvimento ágil

O controlo dos riscos pode melhorar caraterísticas essenciais relativas ao desenvolvimento de *software*, nomeadamente a qualidade e a precisão do planeamento. Em 2008, *Jaana Nyfjord* e *Mira Kajko-Mattsson* [42], propuseram um modelo que visa integrar a gestão do risco com o

desenvolvimento ágil de *software*. Dado não existir nenhum método ou processo geral para a integração dos processos ágeis com os processos de gestão do risco, surgiu a necessidade por parte dos autores de criarem o seu próprio modelo de integração. A metodologia em que se basearam foi a Scrum, combinada com algumas técnicas como por exemplo a programação em pares e desenvolvimento baseado em testes [42].

Com o estudo de diversos cenários de processos ágeis, identificaram quatro pontos de integração [42]:

1) **Níveis organizacionais** – identificação da fonte do risco no processo de desenvolvimento. O processo de desenvolvimento ágil consiste em várias fases que se espalham ao longo dos níveis organizacionais. Encontrar a fonte do risco facilitará a integração entre estes dois processos.
2) **Papéis e responsabilidades** – de acordo com algumas práticas da gestão do risco, os riscos devem ser geridos por vários papéis de forma a garantir que são efetivamente geridos. Surge a necessidade de estabelecer papéis e respetivas responsabilidades.
3) **Canais de comunicação** – o desenvolvimento ágil e a gestão do risco são processos intensivos de comunicação. Se os riscos forem geridos através de toda a organização, através de vários processos e fases, surge a necessidade de criar canais de comunicação apropriados que tornem a comunicação o mais eficiente possível.
4) **Aspetos do processo** – neste modelo são considerados aspetos como definição e avaliação do risco; ciclo de vida do *software*; *stakeholders* papéis e responsabilidades; repositórios e ferramentas de suporte; *templates*; estado do produto; ambiente; organização; e medidas de controlo.

Na Figura 11 é apresentado o modelo integrado. O processo ágil abrange as três fases principais de desenvolvimento, ***product vision planning***, ***product roadmap & release planning*** *e* ***implementation.*** A maioria dos riscos são submetidos a um processo de gestão dentro de cada fase. Os riscos que não são minimizados podem ser transferidos para a fase seguinte e/ou são relatados ao *Risk Management Forum*, cuja função é coordenar a gestão do risco ao nível da organização [42].

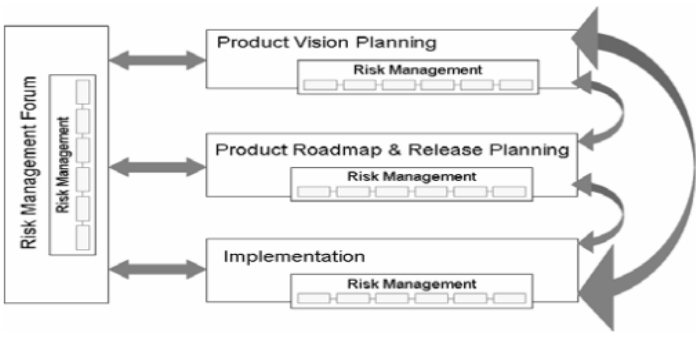


Figura 11 - Visão geral do modelo integrado [42].

A primeira fase, *Product Vision Planning*, envolve a criação de planos que irão servir de guia na tomada de decisão e no desenvolvimento. Na segunda fase, *Product Roadmap & Release*

Planning, são criados planos de alto nível relativos às versões a entregar com frequência, sendo estes revistos antes de começar uma nova entrega. Esta fase envolve riscos de negócio e técnicos. Por fim, a fase de implementação, onde são planeados os próximos passos e o plano é executado de forma a entregar um novo incremento ou funcionalidade do produto. Nesta fase são identificados novos riscos e é feita uma análise e um planeamento da mesma [42].

Este modelo representa um ponto de partida para a integração de abordagens ágeis com modelos de gestão do risco.

3.3 Análise comparativa entre os modelos de gestão dos riscos

A tabela 3 apresenta as fases dos principais modelos de gestão do risco referidas nas secções anteriores.

Tabela 9 - Principais modelos de gestão do risco.

Modelos	**Fases do processo de gestão do risco**					
Barry Boehm	Identificação	Análise	Priorização	Planeamento	Resolução	Monitorização
CRM (SEI)	Identificação	Análise	Planeamento	Monitorização	Controlo	Comunicação
PMBOK Guide®	Planeamento	Identificação	Análise qualitativa	Análise quantitativa	Estratégias de resposta	Monitorização e controlo

Analisando estas fases, pode concluir-se que os modelos apresentados possuem fases em comum, nomeadamente, identificação, análise, planeamento, monitorização e controlo. As técnicas e/ou ferramentas usadas são também semelhantes. A documentação exigida e produzida é uma característica presente em todos os modelos.

Contudo, existem algumas diferenças que se destacam nestes modelos. O modelo do SEI define uma fase para a comunicação, que tal como fora referido é o elo de ligação entre as restantes fases. De facto é o único modelo que destaca a comunicação, incentivando a colaboração e o trabalho em equipa. Esta caraterística aproxima este modelo das abordagens ágeis.

O modelo apresentado no PMBOK *Guide*® é mais completo que os restantes, visto incluir como primeira fase o planeamento de todo o processo de gestão do risco. Esta fase servirá de apoio a todas as fases que se seguem e poderá tornar todo o processo mais eficiente e eficaz. Além disso, inclui análises quantitativas e qualitativas que fornecem mais informação sobre os riscos a serem tratados, o que não acontece com os restantes modelos apresentados. É o único modelo que no

planeamento inicial delega responsabilidades aos utilizadores para todo o processo. Outro aspeto interessante é o facto de ser um modelo geral, não só orientado para o *software* como acontece com o modelo definido pelo SEI.

3.4 Análise dos modelos face aos valores ágeis

Na Tabela 10 são analisados os modelos de gestão do risco, tendo por base os princípios ágeis definidos por *Tamer Madi, Zulkhari Dahalin* e *Fauziah Baharom* em 2011 [13] .

Tabela 10 - Análise dos modelos face aos valores ágeis.

		Valores ágeis									
		Flexibilidade	Foco no cliente	*Software funcional	Colaboração	Simplicidade	Comunicação	*"Naturalidade"	Aprendizagem	Sentido prático	Adaptação
Modelos	Barry Boehm			-				-	✓	✓	✓
	CRM (SEI)			-	✓		✓	-	✓	✓	✓
	PMBOK Guide®			-			✓	-	✓	✓	✓

*Não é passível de ser analisada no contexto em questão.

Todos os modelos dão resposta a três dos princípios ágeis:

- a **aprendizagem** (aquisição de conhecimento e melhoria de capacidades), uma vez que a gestão do risco, segundo estes modelos, tem por base a experiência que é adquirida ao longo do tempo;

- o **sentido prático** (desenvolvimento de soluções de acordo com as condições atuais e com as mudanças ao nível do negócio), visto os modelos serem iterativos e permitirem a inserção e avaliação de novos riscos que possam surgir;

- e, **adaptação** às necessidades do projeto.

A comunicação é propícia à eliminação de erros, e esta é considerada no modelo definido pelo SEI e pelo *PMBOK Guide®*, porém o primeiro atribui-lhe mais importância dado considerar a mesma como uma fase do processo. O modelo do SEI considera também a colaboração, ou seja, existe partilha de informação entre os membros da equipa de desenvolvimento.

Nenhum dos modelos apresentados responde aos seguintes valores ágeis:

- a **flexibilidade** (capacidade de responder rápido à mudança dos requisitos): não se focam única e exclusivamente nos requisitos do sistema, mas em todo o ciclo de vida do projeto;

- **foco no cliente** (o cliente é prioridade em qualquer ação do processo): não envolvem o cliente em todas as fases do processo;

- **simplicidade** (eliminação de toda a burocracia): elevada quantidade de documentação ao longo de todo o processo.

3.4.1 Ligação entre os tipos de riscos e os modelos abordado

Os riscos podem ser agrupados em categorias facilitando todo o processo. Como tal, e recorrendo a informação proveniente da literatura, serão consideradas as dimensões referidas por *L. Wallace* e *M. Keil*, em 2004 [28] : utilizador, requisitos, complexidade do projeto, planeamento e controlo, equipa e ambiente organizacional.

Em nenhum dos modelos surge informação que indique que os mesmos cobrem apenas determinado tipo de riscos. Posto isto, pode afirmar-se que todos os modelos poderão dar resposta a todas as dimensões que serão consideradas.

3.4.2 Lacunas dos modelos abordados

Ao longo do presente documento foram referidas algumas desvantagens existentes nos diversos modelos. A Tabela 11 apresenta as principais lacunas predominantes em todos os modelos de gestão do risco analisados.

Tabela 11 - Principais lacunas dos modelos de gestão do risco.

Modelos	Principais lacunas
Barry Boehm	- Elevada quantidade de documentação (burocracia); - Falhas na definição de papéis de utilizadores em todas as fases do processo; - Podem revelar-se complexos quando aplicados num projeto; - Falhas em algumas práticas/princípios de desenvolvimento ágil, nomeadamente o envolvimento do cliente nas fases do processo;
Continuous Risk Management, SEI	
PMBOK Guide®	

A definição de um modelo de gestão do risco que integre as boas práticas do desenvolvimento ágil será uma medida de minimização das lacunas apresentadas. Este modelo deverá:

- reduzir a burocracia exigida pelos modelos tradicionais;
- envolver o cliente no processo, tal como no desenvolvimento ágil;
- responder às mudanças no ambiente organizacional e de negócio de forma mais eficaz e eficiente;
- ser passível de integrar com metodologias ágeis;
- definir papéis de utilizadores e delegar responsabilidades aos mesmos em cada fase;
- ter por base os princípios ágeis, tornando o processo mais rápido e eficiente.

Capítulo 4

Modelo de gestão do risco em projetos ágeis

4.1 Introdução

A gestão do risco tem como principal objetivo a identificação dos potenciais problemas antes mesmo de estes ocorrerem, de modo a que as atividades relacionadas com o risco possam ser planeadas e invocadas quando necessárias ao longo do ciclo de vida de um produto ou projeto [43] . O sucesso de um projeto depende de vários fatores, sendo que a gestão do risco assume aqui um papel importante. Todos os projetos de *software* se encontram sujeitos a riscos, os projetos cujo desenvolvimento tem por base abordagens ágeis não são exceção. Assim, torna-se importante a definição de um modelo que forneça linhas de orientação específicas relativamente à gestão do risco neste tipo de projetos e que seja um suporte tanto para os gestores de projeto, como para as equipas de desenvolvimento. Este modelo deve permitir às organizações aumentar a percentagem de sucesso nos seus projetos e a implementação de boas práticas, garantindo uma gestão do risco eficaz e eficiente.

4.2 Processo de gestão do risco

Os riscos devem ser considerados quer ao nível organizacional, quer ao nível do projeto, uma vez que um determinado risco da organização pode representar consequências para um determinado projeto. Assim, o modelo poderá definir dois níveis para a gestão do risco, o nível organizacional e o nível do projeto.

a) **Nível organizacional –** neste nível são definidos os riscos mais gerais e relevantes da organização, que podem ter um impacto negativo quer na organização, quer nos projetos da própria organização. Neste nível para representar o processo apenas se encontra definida uma atividade, a análise organizacional, que tem como principal objetivo a definição de um plano de riscos da organização, que servirá de *input* para o nível do projeto. Na Figura 12 está representado o processo relativo a este nível.

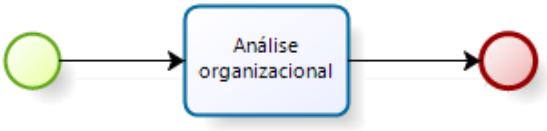


Figura 12 - Proposta nível organizacional.

b) **Nível do projeto –** riscos mais específicos que podem ter um impacto negativo no projeto, conduzindo ao insucesso do mesmo. Na Figura 13 estão representadas as fases do processo para cada iteração.

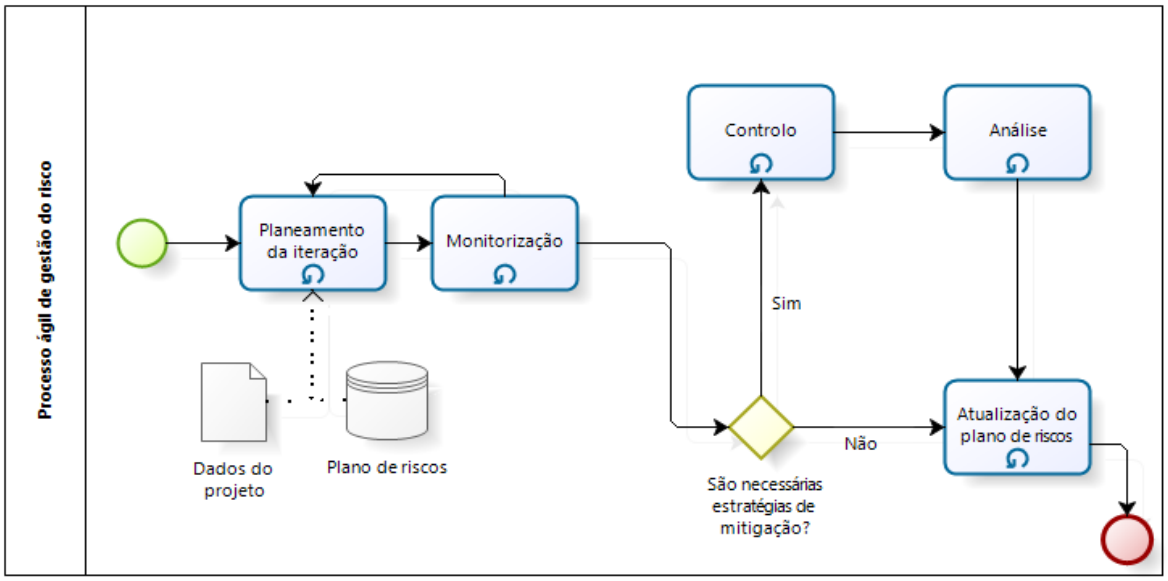


Figura 13 - Proposta do processo para cada iteração.

O processo representado na Figura 13 diz respeito a cada iteração num projeto ágil, isto significa que em cada iteração as atividades se repetem. Inicialmente é feito o planeamento da iteração, tendo por base um plano de riscos previamente definido e os dados referentes ao projeto. O planeamento contém os riscos identificados, as fontes, categorias, estratégias de mitigação, responsáveis e critérios de avaliação dos mesmos, bem como outra informação pertinente. A atividade que se segue é a monitorização, na qual os responsáveis pelos diversos riscos verificam o estado dos mesmos e no caso de novos riscos surgirem essa informação deve ser comunicada ao gestor do projeto para este atualizar o planeamento da iteração. No caso de ser necessária a aplicação de uma estratégia de mitigação para um determinado risco, este passa para a atividade de controlo. Nesta é feito o controlo do risco e são aplicadas as estratégias definidas no planeamento. De seguida, o risco passa para a atividade de análise, na qual é analisado o impacto do risco no projeto. Por fim, é feita uma atualização do plano de riscos base.

Na Tabela 12 encontra-se uma descrição de cada atividade, por nível. Na Tabela 13 encontram-se definidos os dados de entrada e saída para cada atividade e na Tabela 14 estão definidas as responsabilidades em cada atividade.

Tabela 12 – Atividades do modelo proposto para gestão do risco.

	Atividade	Descrição
Nível organizacional	**Análise organizacional**	Fazer uma análise dos processos da organização no sentido de perceber quais são os pontos fracos que no futuro se poderão revelar como sendo riscos ao nível organizacional. Os riscos são priorizados e são apenas considerados os mais relevantes, sendo feita a seleção dos riscos mais preocupantes para a organização. Como *output* é definido um plano de riscos ao nível organizacional que contém os riscos mais importantes (numa vertente organizacional) segundo a análise efetuada, bem como as estratégias de mitigação. Os riscos são categorizados e é definida a periodicidade desta análise.
Nível do projeto	**Planeamento da iteração**	Definir a forma como será conduzida a gestão do risco na iteração. É definido o plano de riscos para a iteração e são definidas as responsabilidades relativamente aos riscos. Os dados de entrada são os dados referentes ao projeto e um plano de riscos base, tendo em conta a análise organizacional.
	Monitorização	Fazer a monitorização dos riscos. Se existir a necessidade de aplicar estratégias de mitigação a um determinado risco, este passa para a atividade de Controlo. No caso de surgirem novos riscos o planeamento da iteração, mais especificamente, o plano de riscos deve ser atualizado.
	Controlo	É feito o controlo dos riscos e são aplicadas estratégias de mitigação e/ou contingência.
	Análise	Após o controlo, é feita uma análise do impacto dos riscos no projeto e são tiradas as conclusões relevantes para a próxima iteração.
	Atualização do plano de riscos	O plano de riscos deve ser atualizado no final de cada iteração, visto este modelo ser baseado no conceito de aprendizagem contínua.

Tabela 13 - Dados de entrada e saída das diferentes atividades do modelo.

	Atividade	Dados de entrada	Dados de saída
Nível organizacional	**Análise organizacional**	1. Dados relativos aos processos organizacionais. 2. Dados recolhidos na organização (incertezas de elementos chave da organização)	**1. Dados sobre riscos organizacionais:** a) Fontes dos riscos b) Categorias dos riscos c) Plano de riscos o Lista de riscos priorizada o Estratégia de gestão dos riscos o Requisitos para a gestão dos riscos (intervalos de reavaliação)

			d) Análise do impacto dos riscos para a organização.
Nível do projeto	**Planeamento da iteração**	1. Dados do projeto (restrições, recursos, requisitos, orçamento, âmbito, prazos, entre outros). 2. Plano de riscos base. 3. Dados sobre riscos organizacionais.	**1. Planeamento da iteração** a) Lista de riscos b) Fontes dos riscos c) Categorias dos riscos d) Contexto, condições e consequências dos riscos ocorrerem. e) Planos de mitigação/contingência f) Lista de responsabilidades g) Critérios de avaliação e quantificação do risco
	Monitorização	1. Dados de saída da fase anterior	**1. Planeamento da iteração.** a) Informação do risco atualizada.
	Controlo	1. Dados de saída da fase anterior.	**1. Planeamento da iteração.** a) Informação do risco atualizada. b) Resultado da aplicação do plano de mitigação.
	Análise	1. Dados de saída da fase anterior	**1. Planeamento da iteração** **2. Análise detalhada do impacto do risco no projeto.** a) Avaliação da probabilidade e consequências dos riscos.
	Atualização do plano de riscos	1. Dados de saída das fases anteriores (monitorização e análise)	-

Tabela 14 - Matriz RACI.[1]

	Gestor de topo	Gestor de projeto	Membro da equipa	Descrição
Planeamento da iteração	I	R/A	C	O planeamento da iteração será feito pela equipa em colaboração com o gestor do projeto, de forma a garantir que a maior parte dos riscos são identificados numa fase inicial. Contudo, a aprovação final do planeamento é da responsabilidade do gestor do projeto.
Monitorização	I	I	R/A	Na monitorização, o gestor do projeto seleciona um membro da equipa para ser responsável pela monitorização de um ou mais riscos. Contudo, o gestor do projeto deve ser informado sobre o progresso da atividade. No caso de surgirem novos riscos o gestor deverá ser informado, para atualizar o planeamento da iteração.
Controlo	I	R/A (I)	I (R/A)	Na fase de controlo existem duas possibilidades: 1. **R/A**: Gestor do projeto; **I**:Membro da equipa. O gestor do projeto é responsável pela aplicação de um plano de mitigação/contingência, sendo que os membros da equipa são informados sobre o progresso da aplicação dessas mesmas estratégias. 2. **I**: Gestor do projeto; **R/A**: Membro da equipa. Um membro da equipa é responsável pela aplicação de um plano de mitigação/contingência, contudo o gestor do projeto é informado sobre o progresso da aplicação dessas mesmas estratégias.
Análise	I	R/A	C	O gestor do projeto é responsável pela análise do impacto dos riscos no projeto,

[1] http://www.managementstudyguide.com/raci-matrix.htm

				assim nesta fase é o responsável pela execução da tarefa e pelos resultados da mesma. A equipa poderá colaborar nesta atividade.
Atualização do plano de riscos	I	R/A	-	O gestor do projeto atualiza o plano de riscos base, com os novos dados que surgiram na iteração.

Nota: O gestor de topo é informado sobre o progresso de cada atividade.

Legenda:

R – Responsável pela execução da tarefa.

A – Responsável pela atividade, quem responde pelos resultados da mesma.

C – Colabora na atividade.

I – É informado sobre o progresso da atividade.

4.3 Ferramentas e boas práticas

O modelo é definido pelas atividades do processo, por um conjunto de boas práticas e por um conjunto de ferramentas. Neste caso em particular, para este modelo foram definidas algumas boas práticas que devem ser consideradas durante a gestão dos riscos, apresentadas na Tabela 15. Note-se que as práticas consideradas ao longo do desenvolvimento ágil de *software*, apesar de não constarem na Tabela 15, devem também ser consideradas, uma vez que este modelo é orientado para projetos ágeis de *software.*

Relativamente a ferramentas que sirvam de suporte à gestão do risco, considerando o modelo apresentado, o módulo desenvolvido na componente prática, em colaboração com uma empresa é uma boa solução. Na secção 5 será apresentado com detalhe o módulo e as suas potencialidades face a uma gestão do risco completa e precisa.

Tabela 15 - Conjunto de boas práticas do modelo de gestão dos riscos organizadas por atividades do processo.

	Atividade	**Boas Práticas**
Nível organizacional	**Análise organizacional**	1. Identificar os riscos organizacionais. 2. Avaliar, categorizar e prioritizar os riscos.

Nível do projeto	**Planeamento da iteração**	1. Definir uma estratégia para a gestão do risco para cada iteração. 2. Identificar os riscos. 3. Avaliar, categorizar e priorizar os riscos. a. Definir escalas de impacto para os objetivos principais do projeto. 4. Definir planos de mitigação e contingência.
	Monitorização	1. Monitorizar e analisar o estado dos riscos.
	Controlo	1. Monitorizar o estado dos riscos. 2. Invocar opções de tratamento dos riscos mediante o resultado da monitorização. 3. Estabelecer um período para cada atividade de tratamento dos riscos que inclui uma data de início e uma data prevista para conclusão.
	Análise	1. Avaliar o impacto dos riscos no projeto. 2. Recolher indicadores de desempenho em cada atividade de tratamento do risco.
	Atualização do plano de riscos	1. Atualizar o plano de riscos com a informação relevante a considerar na próxima iteração.

Além das boas práticas acima apresentadas, no final do projeto devem ser sempre registadas as lições aprendidas que permitem perceber o que correu bem e mal durante o projeto. As lições aprendidas são um fator chave para a melhoria contínua em projetos de *software*.

4.4 Mapeamento entre o processo e as principais metodologias ágeis

Nesta secção pretende-se mapear o processo apresentado anteriormente com as principais metodologias ágeis, nomeadamente Scrum e XP. Na Figura 12 encontra-se representado o

mapeamento entre as atividades do processo e as metodologias ágeis. A fase *Pre-game* (metodologia *Scrum*) e a fase de *Exploration* e *Planning* (metodologia XP) podem ser mapeadas com a atividade de Planeamento da iteração. As atividades de Monitorização, Controlo, Análise e Atualização do plano de riscos podem ser mapeadas com as fases *Development* e *Iterations to release*, das metodologias *Scrum* e XP, respetivamente.

A atividade análise organizacional não é passível de ser mapeada com nenhuma das metodologias uma vez que não é feita ao nível do projeto, mas ao nível da organização. Relativamente às fases que não se encontram mapeadas com as atividades do processo, nomeadamente as fases *Post-game* (metodologia Scrum) e *Death* (metodologia XP), também não são passíveis de ser mapeadas com as fases do processo, uma vez que são fases referentes ao fecho do projeto e não são iterativas.

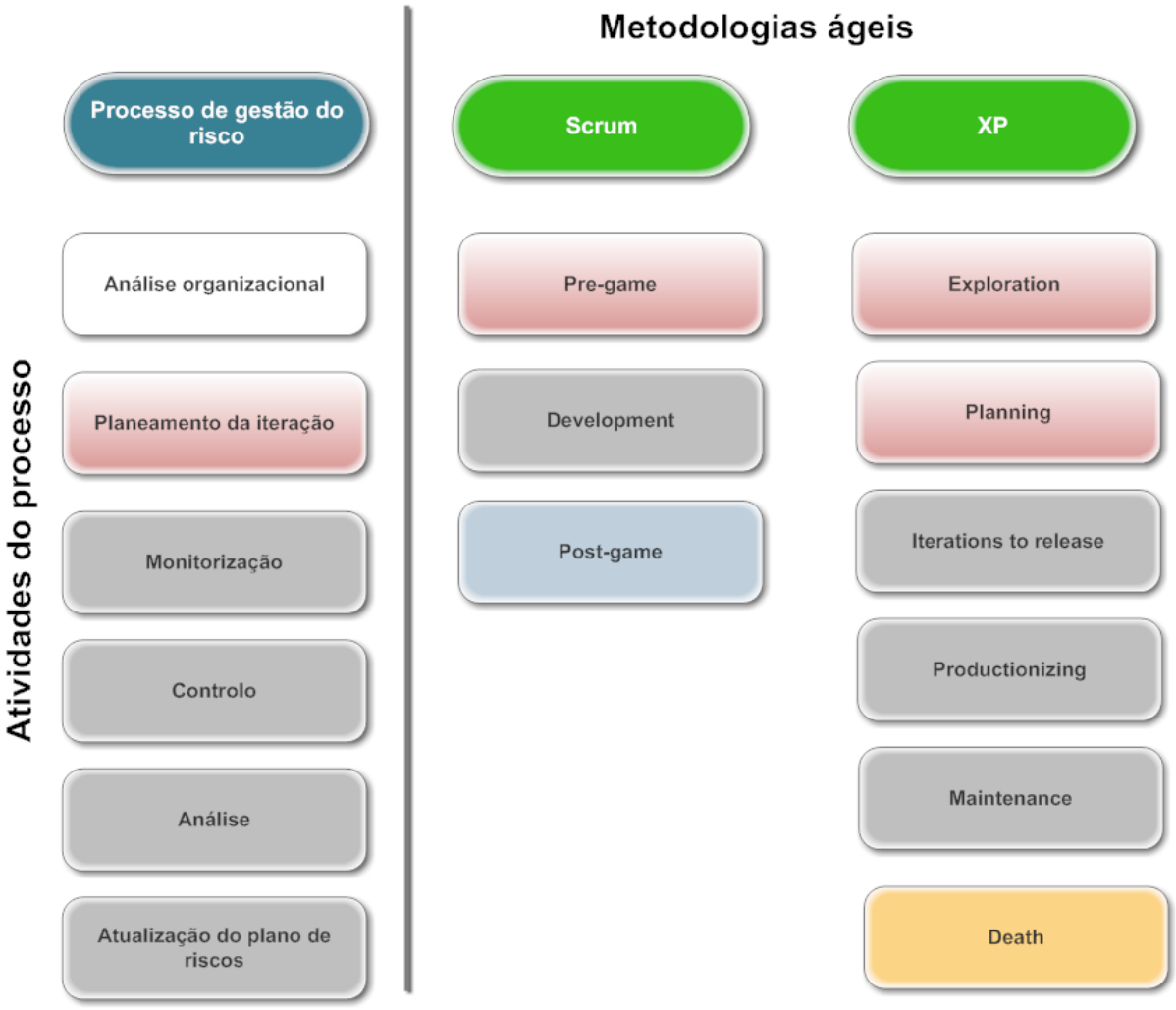


Figura 14- Mapeamento entre as atividades do processo com as metodologias Scrum e XP.

Capítulo 5

Análise da aplicabilidade do modelo

5.1 Módulo de gestão dos riscos

Para efeitos de validação prática do modelo, foi desenvolvido um módulo de gestão dos riscos para uma ferramenta de gestão de projetos. Este desenvolvimento foi realizado em colaboração com uma empresa portuguesa da área da Engenharia de Software. Em termos gerais, a ferramenta da qual o módulo de gestão do risco faz parte, tem como principal objetivo melhorar e facilitar a forma como os projetos são geridos nas organizações, permitir ao gestor de projeto gerir facilmente os projetos pelos quais está responsável e ajudar a própria equipa de desenvolvimento do projeto a gerir as suas tarefas. É composta por diversos módulos que servem de suporte à gestão de projetos de *software*. Foi pensada por peritos em Engenharia de *Software* e pretende ser uma ferramenta de gestão de projetos que tem por base as melhores práticas de Engenharia de *Software*.

O módulo desenvolvido no âmbito desta dissertação foi o módulo de gestão dos riscos. Neste podem ser geridos os riscos, impedimentos e ações de um determinado projeto.

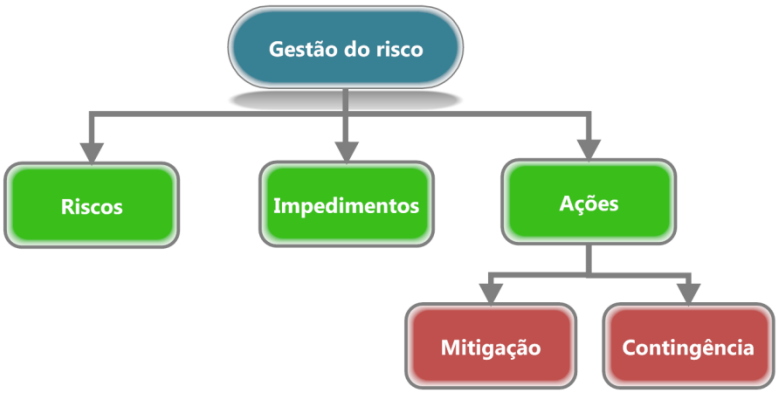


Figura 15 - Gestão do risco: visão geral.

Este módulo permite definir, ao nível organizacional, quais os valores de impacto e probabilidade da organização e a periodicidade de revisão dos riscos (diária, semanal, quinzenal, mensal). Os valores de impacto e probabilidade definidos para a organização serão usados na criação de riscos de um projeto. Note-se que a periodicidade de revisão dos riscos é definida ao nível organizacional, todavia poderá ser alterada na criação de um projeto, uma vez que alguns projetos são mais críticos e complexos do que outros e a periodicidade de revisão dos riscos irá variar tendo por base esse fator.

Ao nível do projeto, este módulo permite, como foi referido, a alteração da periodicidade de revisão dos riscos e a definição de uma estratégia de gestão dos riscos do projeto, nomeadamente a partir de que nível de risco serão obrigatórias ações de mitigação e ações de contingência. Note-se que ações de mitigação representam ações que serão executadas para mitigar o risco, para que este

não se torne um problema. Ações de contingência são executadas quando um risco se tornou um problema. O nível de risco é o calculado pela multiplicação entre o impacto e a probabilidade do risco.

Ainda ao nível do projeto, nomeadamente quando são criados riscos, é possível definir um plano de contingência e ações para mitigar o risco criado. Quando o risco se torna um problema é gerado um impedimento para o projeto em questão, bem como ações de contingência para resolver o impedimento criado. Os riscos podem ser privados ou públicos, em termos de estado. Um risco pode ter apenas dois estados, aberto e fechado.

Na Figura 14 encontra-se um diagrama representativo das funcionalidades do módulo de gestão do risco, mais ao nível do projeto.

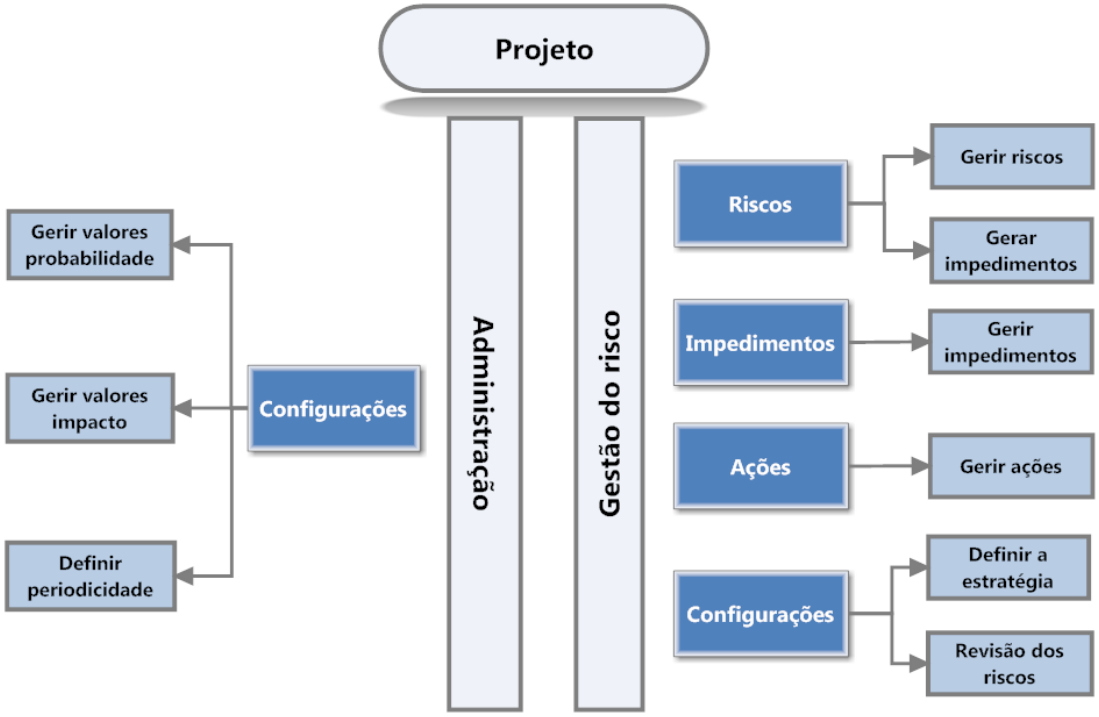


Figura 16 - Funcionalidades do módulo de gestão dos riscos.

Para monitorização e análise dos riscos do projeto surge a matriz de risco, a qual apresenta a quantidade de riscos mediante os valores de impacto e probabilidade e valores dos indicadores. No que diz respeito aos indicadores temos o nível de risco total, o nível de risco atual, o somatório do nível dos riscos abertos e dos riscos fechados e o total de riscos abertos e fechados.

Seja **X** o número de riscos abertos, **X1** o número total de riscos, **Y** o somatório do nível de risco referente aos riscos abertos e **Z** o nível de risco. O nível de risco total é dado pela fórmula:

$$\frac{Y}{Max(Z) \times X1}$$

e o nível de risco atual é dado pela fórmula:

$$\frac{Y}{Max(Z) \times X}$$

É importante ainda salientar, que a ferramenta permite definir permissões por papéis de utilizadores.

Em suma, o módulo de gestão de riscos implementado permite gerir os riscos por projeto, gerir ações de mitigação e contingência que deverão ser executadas para minimizar o impacto de um risco no projeto, gerir os impedimentos associados a um projeto oriundos ou não de riscos, definir uma estratégia para a gestão do risco e fazer uma análise e monitorização dos riscos por projeto. Permite também ao utilizador ter uma vista dos riscos que são passíveis de ser fechados, nomeadamente riscos que não tenham ações associadas, ou riscos cujas ações já estejam fechadas.

Em anexo (ver anexo A) encontra-se um diagrama geral do módulo de gestão do risco implementado para validação prática do modelo definido.

5.2 Mapeamento entre o processo e o módulo de gestão dos riscos

Na Tabela 16 faz-se o mapeamento entre algumas das funcionalidades desenvolvidas com as atividades do processo do modelo de gestão dos riscos definido.

Como se pode verificar, a criação de ações de mitigação e de contingência de um risco e a geração de impedimentos são formas de controlar o risco, ou seja, as ações de mitigação são aplicadas para mitigar o risco e evitar que este se torne um problema, enquanto as ações de contingência são aplicadas após o risco se tornar um problema. Note-se que quando um risco se torna um problema, é gerado um impedimento. A definição da estratégia de gestão do risco, bem como a criação de riscos de um projeto, assentam mais na atividade de planeamento da iteração, uma vez que em cada iteração devem ser definidos os riscos que irão constar e haver uma estratégia que defina a forma como serão geridos. A matriz de risco e o nível de risco permitem monitorizar os riscos do projeto.

Tabela 16 - Mapeamento entre o módulo de gestão de riscos e as atividades do processo.

		Atividades do processo				
		Planeamento da iteração	Monitorização	Controlo	Análise	Atualização do plano de riscos
Funcionalidades	Criação de ações de mitigação e contingência de um risco.			✓		
	Definição da estratégia de gestão do risco.	✓				
	Matriz de risco		✓		✓	
	Atualizar plano					✓

	contingência					
	Nível de risco		✓		✓	
	Criação de riscos de um projeto	✓				
	Geração de impedimentos			✓		

5.3 Avaliação qualitativa do modelo

Para validação do modelo de gestão do risco foram definidas métricas qualitativas, no sentido de avaliar a aplicabilidade prática do mesmo. A abordagem adotada para a definição das métricas foi a abordagem ***Goal-Metric-Question (GQM)*** [44], uma das abordagens mais populares em estudos empíricos em Engenharia de Software que define três níveis: conceptual (***Goal***), operacional (***Metric***) e quantitativo (***Question***).

Na Tabela 17 são apresentadas as métricas, o objetivo de cada métrica e a questão que avalia o objetivo da métrica (ver anexo B questionário), segundo a abordagem GQM.

Tabela 17 - Métricas qualitativas para validação do modelo definido.

Métrica	Objetivo	Questão
Eficácia	Avaliar se a abordagem definida permite gerir os riscos em projetos ágeis de *software*.	Q5
Eficiência	Avaliar se a abordagem definida permite gerir os riscos corretamente, sem erros e com qualidade em projetos ágeis de *software*.	Q6
Relação com valores/princípios ágeis	Avaliar se o modelo se enquadra com as abordagens ágeis.	Q7
Complexidade	Medir as dificuldades do modelo face a outros modelos.	Q8
Usabilidade	Avaliar a abordagem definida para gerir os riscos em termos de capacidade de satisfação das necessidades do utilizador, nomeadamente simplicidade e eficiência.	Q9
Relação com o modelo	Avaliar e validar se a componente prática dá suporte ao modelo definido.	Q10

Para avaliação das métricas qualitativas definidas, foi criado um questionário (ver Anexo B) que pretende avaliar gestores de projeto, a correlação entre o modelo e o módulo desenvolvido. Note-

se que as primeiras questões do questionário garantem que a resposta aos questionários é válida e permitem perceber o perfil da amostra que participou no processo de validação.

A ferramenta onde se insere o módulo desenvolvido no âmbito deste trabalho ainda não se encontra em comercialização, como tal não foi possível a disponibilização do módulo para validação por gestores de projeto. Porém, o processo de validação encontra-se definido e está prevista a sua aplicação aquando da divulgação da ferramenta por parte da empresa, ou seja, após estar em produção e numa fase em que quem usa já tem uma opinião formada sobre a mesma. Assim, procurar-se-á efetuar a validação tendo por base várias equipas de projeto e gestores de projeto e após um tempo de utilização da mesma, de pelo menos 12 meses.

Capítulo 6

Conclusões e trabalho futuro

A competitividade organizacional leva a que as organizações se preocupem com fatores como a confiança, reputação, oportunidades e ameaças com que se deparam nos mercados em que estão inseridas. Cada vez mais se verifica a adoção de novas estratégias que se traduzem em processos de melhoria contínua para as organizações.

Quando se menciona o termo risco, rapidamente surgem preocupações, uma vez que, para muitos o risco é apenas um evento que tem impacto negativo. Porém, o risco pode representar uma ameaça ou uma oportunidade. Desta forma, torna-se fulcral para as organizações de desenvolvimento de *software* abordar o risco e geri-lo. A gestão do risco em projetos de *software,* atualmente, possui um papel importante nas organizações, apesar de até há algum tempo ser negligenciada pelas mesmas. A resistência à mudança surge como uma das principais razões apontadas para esta negligência.

Lidar com o risco nem sempre é tarefa fácil, por isso existem técnicas que servem de suporte para cada passo do processo de gestão do risco. Apesar de existirem modelos de suporte à gestão do risco, é importante referir que as pessoas envolvidas são uma peça fundamental para o sucesso do desenvolvimento dos projetos. É fundamental que as pessoas envolvidas estejam predispostas a fazer com que o projeto tenha sucesso.

Um aspeto adicional a salientar é que, durante o processo de desenvolvimento de um projeto, os gestores têm de tomar decisões que serão importantes e determinantes no decorrer do mesmo. A tomada de decisão, por vezes, não é um processo simples, pois a imprevisibilidade dos projetos a tal conduz. Torna-se necessária a conjugação de ferramentas de apoio à decisão com a gestão do risco, de forma a garantir que, ao longo do processo de gestão do risco, as decisões corretas são tomadas e que são bem justificadas. Por último, o facto de se lidar com o risco não é sucesso garantido nos projetos, mas é um fator muito importante no caminho do mesmo.

A transição de modelos tradicionais para metodologias ágeis criou novos desafios no âmbito da gestão do risco. Quando se fala em combinar a gestão do risco com metodologias ágeis, é importante ter em conta que os modelos existentes para a gestão do risco devem ser reajustados [45]. As estatísticas revelam que a implementação de uma gestão do risco eficiente é importante na redução de perdas e no sucesso dos projetos de *software* [35]. Os riscos encontram-se presentes em todos os projetos de desenvolvimento de *software*, desta forma torna-se importante identificar os mesmos, estimar a probabilidade de ocorrerem, bem como o impacto que poderão causar no projeto. Vários são os projetos que face a uma incorreta gestão dos riscos são interrompidos e acabam por causar perdas significativas para a organização. As abordagens ágeis por si só não são uma medida de minimização dos riscos, uma vez que os riscos não podem ser considerados apenas numa vertente relacionada com os requisitos.

Em suma, o modelo definido na presente dissertação surgiu devido ao facto de não existir ainda um modelo de gestão do risco orientado para projetos ágeis de *software* que forneça linhas de orientação específicas e seja passível de integrar com as metodologias de desenvolvimento ágil. O interesse do modelo definido suscitou interesse numa empresa e foi feita a especificação de uma solução. Assim, este trabalho contribuiu para alguns avanços nesta área ao propor um novo modelo de gestão do risco que considera princípios e valores ágeis de *software* e um módulo de gestão do risco numa ferramenta comercial.

Ao longo do presente trabalho surgiram pequenas restrições ao nível da validação prática do modelo, uma vez que, tal como já fora referido, a ferramenta na qual foi desenvolvido o módulo de gestão dos riscos, ainda não está a ser comercializada e como tal existem algumas questões de confidencialidade associadas que não permitiram a avaliação prática do módulo desenvolvido. Contudo, no anexo B encontra-se um questionário que poderá ser aplicado para recolha de dados para futura validação.

No que diz respeito a possibilidades de trabalho futuro, o modelo definido na presente dissertação poderá ser refinado e avaliado após o processo de validação, tendo em conta outras metodologias ágeis, como a XP, de forma a haver uma comparação de resultados mais completa. Além disso, no futuro, poderão ser aplicadas para avaliação do modelo para além das métricas qualitativas, mencionadas na secção anterior, algumas métricas quantitativas. De salientar que o módulo desenvolvido se encontra ainda numa fase de testes, como tal serão implementadas algumas melhorias no mesmo.

Referências Bibliográficas

[1] T. Gilb, *Principles of Software Engineering Management*. Addison-Wesley, 1988, p. 72.

[2] L. Sarigiannidis, “Software Development Project Risk Management: A New Conceptual Framework,” *J. Softw. Eng. Appl.*, vol. 04, no. 05, pp. 293–305, 2011.

[3] C. Ibbs and Y. Kwak, “Assessing project management maturity,” *Proj. Manag. J.*, no. March, 2000.

[4] Y. Kwak and C. Ibbs, “Calculating project management’s return on investment,” *Proj. Manag. J.*, 2000.

[5] K. M. Adams and C. A. Pinto, “Software Development Project Risk Management: A Literature Review,” in *Pro- ceedings of the 26th National Conference, Organizational Transformation: Opportunities and Challenges, American Society for Engineering Management, Rolla*, 2005, pp. 635–641.

[6] F. M. Dedolph, “The neglected management activity: Software risk management,” *Bell Labs Tech. J.*, vol. 8, no. 3, pp. 91–95, Dec. 2003.

[7] R. Charette, “Why software fails,” *IEEE Spectr.*, 2005.

[8] T. Raz and E. Michael, “Use and benefits of tools for project risk management,” *Int. J. Proj. Manag.*, vol. 19, no. 1, pp. 9–17, Jan. 2001.

[9] C. Alberts and A. Dorofee, “Mission Risk Diagnostic (MRD) Method Description,” 2012.

[10] S. Alhawari and F. Sciences, “A Risk Management Model for Project Execution,” pp. 887–893.

[11] P. L. Bannerman, “Risk and risk management in software projects: A reassessment,” *J. Syst. Softw.*, vol. 81, no. 12, pp. 2118–2133, Dec. 2008.

[12] P. Meso and R. Jain, “AGILE SOFTWARE DEVELOPMENT : ADAPTIVE SYSTEMS PRINCIPLES AND SOFTWARE DEVELOPMENT,” *Inf. Syst. Manag.*, pp. 19–30, 2006.

[13] T. Madi, Z. Dahalin, and F. Baharom, “Content analysis on agile values: A perception from software practitioners,” *2011 Malaysian Conf. Softw. Eng.*, pp. 423–428, Dec. 2011.

[14] I. Sommerville, *Software Engineering, eighth edition*. England: Pearson Education, 2007.

[15] T. Madi, Z. Dahalin, and F. Baharom, “Content analysis on agile values: A perception from software practitioners,” *2011 Malaysian Conf. Softw. Eng.*, pp. 423–428, Dec. 2011.

[16] “Manifesto for Agile Software Development.” [Online]. Available: http://agilemanifesto.org/.

[17] J. M. Fernandes and M. Almeida, “Classification and Comparison of Agile Methods,” *2010 Seventh Int. Conf. Qual. Inf. Commun. Technol.*, pp. 391–396, Sep. 2010.

[18] K. Beck, *Extreme Programming Explained: Embrace Change*. Addison- Wesley Professional, 2000.

[19] K. Beck and M. Fowler, *Planning Extreme Programming*. Addison- Wesley Professional, 2001.

[20] K. Beck and C. Andres, *Extreme Programming Explained: Embrace Change*, 2 ed. Boston, MA, USA: Addison-Wesley, 2004.

[21] K. Schwaber and M. Beedle, *Agile Software Development with Scrum*. Prentice Hall, 2002.

[22] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, “Agile software development methods,” *VTT Publ. 478*, 2002.

[23] J. M. Fernandes and M. Almeida, “Classification and Comparison of Agile Methods,” *2010 Seventh Int. Conf. Qual. Inf. Commun. Technol.*, pp. 391–396, Sep. 2010.

[24] J. Sutherland, *Scrum Handbook*. 2010.

[25] B. Boehm, “Software risk management: principles and practices,” *Software, IEEE*, no. January, 1991.

[26] T. Arnuphaptrairong, “Top Ten Lists of Software Project Risks: Evidence from the Literature Survey,” *Proc. Int. MultiConference Eng. Comput. Sci.*, vol. I, 2011.

[27] F. MCFarlan., “Portfolio approach to information systems,” *Harv. Bus. Rev.*, vol. 59, no. 5, pp. 142–150, 1981.

[28] L. Wallace and M. Keil, “Software project risks and their effect on outcomes,” *Commun. ACM*, vol. 47, no. 4, pp. 68–73, Apr. 2004.

[29] E. Gottesdiener, “How agile practices reduce requirements risk,” *Better Softw.*, no. August, pp. 14–15, 2009.

[30] H. Hijazi, T. Khdour, and A. Alarabeyyat, “A Review of Risk Management in Different Software Development Methodologies,” *Int. J. Comput. Appl.*, vol. 45, no. 7, pp. 8–12, 2012.

[31] G. Babu, “Increasing Success of Software Projects through Minimizing Risks,” *Int. J. Res. Rev. Softw. Eng.*, vol. 1, no. 1, 2011.

[32] M. António, *Gestão de Projectos de Software – 3ª Edição*. FCA – Editora de Informática, 2008.

[33] P. L. Bannerman, “Risk and risk management in software projects: A reassessment,” *J. Syst. Softw.*, vol. 81, no. 12, pp. 2118–2133, Dec. 2008.

[34] C. Alberts and A. Dorofee, “Mission Risk Diagnostic (MRD) Method Description,” 2012.

[35] P. Tianyin, “Development of software project risk management model review,” in *2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC)*, 2011, pp. 2979–2982.

[36] R. Fairley, “Risk management for software projects,” *IEEE Softw.*, vol. 11, no. 3, pp. 57–67, May 1994.

[37] R. L. Kliem and I. S. Ludin, *Reducing Project Risk*. Grower, 1997.

[38] C. Chapman and S. Ward, *Project risk management: processes, techniques and insights*. John Wiley, 2007.

[39] B. W. Boehm, “Software risk management: principles and practices,” *IEEE Softw.*, vol. 8, no. 1, pp. 32–41, Jan. 1991.

[40] A. J. Dorofee, J. A. Walker, C. J. Alberts, R. P. Higuera, R. L. Murphy, and R. C. Williams, “Continuous Risk Management Guidebook,” 1996.

[41] Project Management Institute, *Practice Standard for Project Risk Management*. 2009.

[42] J. Nyfjord and M. Kajko-Mattsson, “Outlining a Model Integrating Risk Management and Agile Software Development,” in *2008 34th Euromicro Conference Software Engineering and Advanced Applications*, 2008, pp. 476–483.

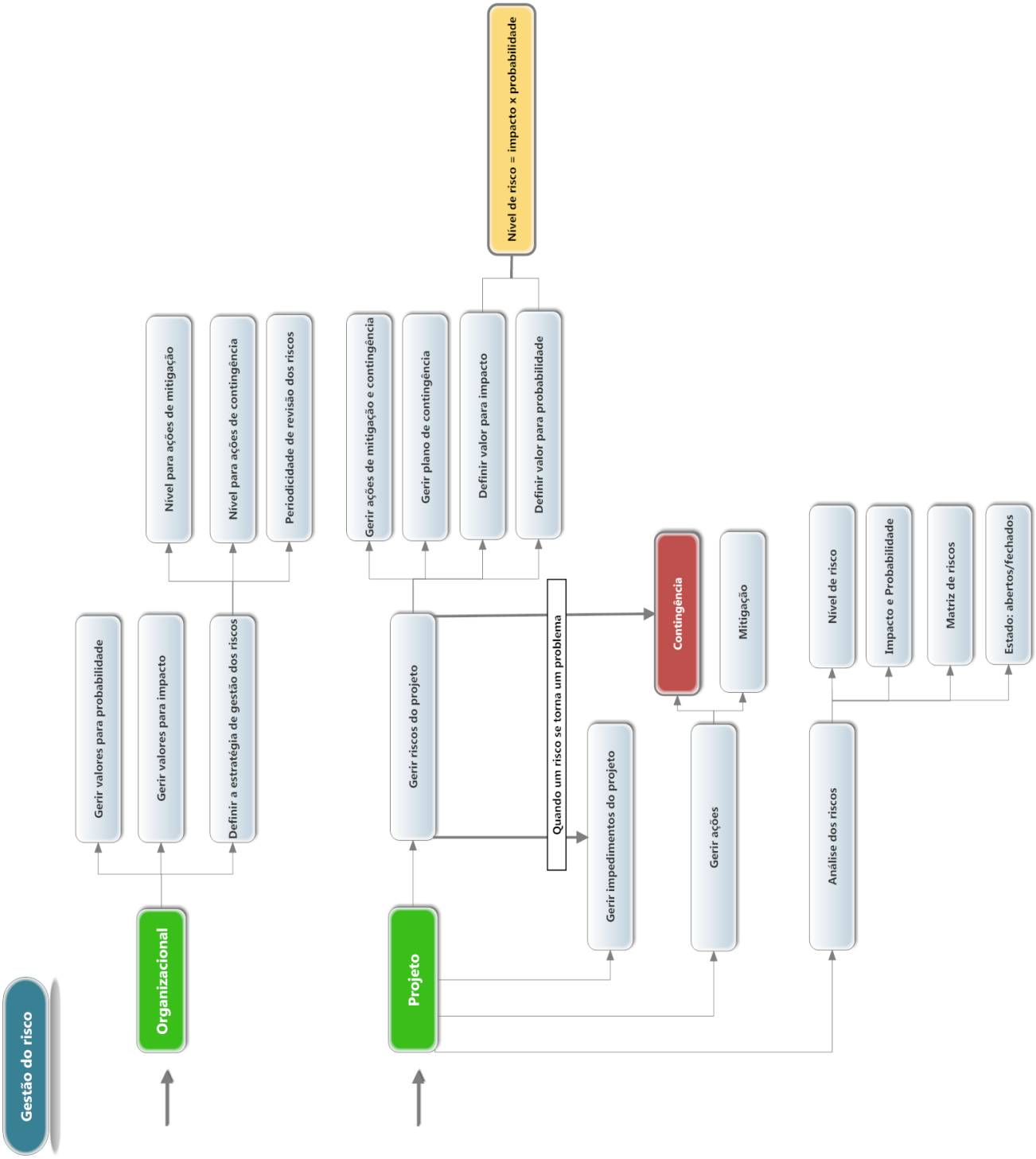
[43] S. E. Institute, “CMMI® for Development, Version 1.3 CMMI-DEV, V1.3,” 2010.

[44] V. R. Basili, G. Caldiera, and H. D. Rombach, “The goal question metric approach,” *Encyclopedia of Software Engineering*. pp. 528–532, 1996.

[45] V. Ylimannela, “A model for risk management in agile software development.”

Anexo A

Diagrama geral do módulo de gestão de riscos



Anexo B

Questionário para avaliação da aplicabilidade do modelo ágil de gestão do risco

Este questionário procura avaliar a aplicabilidade do modelo de gestão do risco definido e a sua ligação com o módulo de gestão do risco implementado na ferramenta em análise. Para tal é fornecida informação sobre o modelo construído.

1. Qual a função que exerce normalmente nos projetos em que está inserido?
   1 – Gestor do projeto **2 – Membro da equipa de desenvolvimento** **3 – Responsável técnico** **4 – Outro_____**

2. Na função mencionada na questão anterior, quantos anos de experiência tem?

3. Quais os modelos de gestão do risco que conhece?

4. Tem conhecimento de ferramentas de gestão de projetos que permitam fazer a gestão dos riscos de projetos de *software*? Se sim, quais?

5. Em termos de eficácia (permitir gerir os riscos), como classifica a gestão dos riscos de um projeto, segundo a abordagem apresentada:
   1 – Insuficiente **2 – Suficiente** **3 – Bom** **4 - Muito Bom**

6. Em termos de eficiência, como classifica a gestão dos riscos permitida pela ferramenta:
   1 – Insuficiente **2 – Suficiente** **3 – Bom** **4 - Muito Bom**

7. Identifique quais os valores ágeis que considera que estejam presentes no módulo de gestão dos riscos apresentado:

- Flexibilidade
- Foco no cliente
- Colaboração
- Simplicidade
- Comunicação
- Aprendizagem
- Sentido prático
- Adaptação
- Nenhum dos anteriores

8. Como classifica esta gestão dos riscos face a outros modelos existentes, em termos de complexidade:
   1 – Complexa 2 – Equiparável 3 – Simples 4 - Bastante simples 5 - Sem opinião

9. Como classifica, em termos de usabilidade a abordagem usada para gerir os riscos do projeto:
   1 – Bastante complexa 2 – Complexa 3 – Simples 4 - Bastante simples

10. Relacione as atividades do processo do modelo de gestão do risco com as respetivas funcionalidades do módulo desenvolvido.

		Atividades do processo				
		Planeamento da iteração	Monitorização	Controlo	Análise	Atualização do plano de riscos
Funcionalidades	Criação de ações de mitigação e contingência de um risco.					
	Definição da estratégia de gestão do risco.					
	Matriz de risco					
	Atualizar plano contingência					
	Nível de risco					
	Criação de riscos de um projeto					
	Geração de impedimentos					

11. Comentários gerais (Se pretender poderá deixar o seu feedback):