

Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Programa de Pós-Graduação em Ciência da Computação

Um Processo baseado em Redes Bayesianas para
Avaliação da Aplicação do Scrum em Projetos de
Software

Mirko Barbosa Perkusich

Tese submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande - Campus I como parte dos requisitos necessários para obtenção do grau de Doutor em Ciência da Computação.

Área de Concentração: Ciência da Computação

Linha de Pesquisa: Engenharia de Software

Hyggo Oliveira de Almeida e Kyller Costa Gorgonio
(Orientadores)

Campina Grande, Paraíba, Brasil

©Mirko Barbosa Perkusich, 05/03/2018

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

P451p

Perkusich, Mirko Barbosa.

Um processo baseado em redes bayesianas para avaliação da aplicação do scrum em projetos de software / Mirko Barbosa Perkusich. – Campina Grande, 2018.
170 f. : il. color.

Tese (Doutorado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2018.

"Orientação: Prof. Dr. Hyggo Oliveira de Almeida, Prof. Dr. Kyller Costa Gorgonio".

Referências.

1. Desenvolvimento Ágil de Software. 2. Redes Bayesianas. 3. Inteligência Artificial. I. Almeida, Hyggo Oliveira de. II. Gorgonio, Kyller Costa. III. Título.

CDU 004.8(043)

**"UM PROCESSO BASEADO EM REDES BAYESIANAS PARA AVALIAÇÃO DA
APLICAÇÃO DO SCRUM EM PROJETOS DE SOFTWARE"**

MIRKO BARBOSA PERKUSICH

TESE APROVADA EM 05/03/2018

**HYGGO OLIVEIRA DE ALMEIDA, Dr., UFCG
Orientador(a)**

**KYLLER COSTA GORGÔNIO, Dr., UFCG
Orientador(a)**

**JOSEANA MACÊDO FECHINE RÉGIS DE ARAÚJO, Dra., UFCG
Examinador(a)**

**EVANDRO DE BARROS COSTA, Dr., UFAL
Examinador(a)**

**MARÍA DEL ROSARIO GIRARDI GUTIERREZ, Dra., UFMA
Examinador(a)**

**ROSSANA MARIA DE CASTRO ANDRADE, Dra., UFC
Examinador(a)**

CAMPINA GRANDE - PB

Resumo

O aumento na utilização de métodos ágeis tem sido motivado pela necessidade de respostas rápidas a demandas de um mercado volátil na área de software. Em contraste com os tradicionais processos dirigidos a planos, métodos ágeis são focados nas pessoas, orientados à comunicação, flexíveis, rápidos, leves, responsivos e dirigidos à aprendizagem e melhoria contínua. Como consequência, fatores subjetivos tais como colaboração, comunicação e auto-organização são chaves para avaliar a maturidade do desenvolvimento de software ágil. O Scrum, focado no gerenciamento de projetos, é o método ágil mais popular. Ao ser adotado por uma equipe, a aplicação do Scrum deve ser melhorada continuamente sendo complementado com práticas e processos de desenvolvimento e gerenciamento ágeis. Apesar da Reunião de Retrospectiva, evento do Scrum, ser um período reservado ao final de cada sprint para a equipe refletir sobre a melhoria do método de desenvolvimento, não há procedimentos claros e específicos para a realização da mesma. Na literatura, há diversas propostas de soluções, embora nenhuma consolidada, para tal. Desta forma, o problema em questão é: como instrumentar o Scrum para auxiliar na melhoria contínua do método de desenvolvimento com foco na avaliação do processo de engenharia de requisitos, equipe de desenvolvimento e incrementos do produto? Nesta tese, propõe-se um processo sistemático baseado em redes bayesianas para auxiliar na avaliação da aplicação do Scrum em projetos de software, instrumentando o método para auxiliar na sua melhoria contínua com foco na avaliação do processo de engenharia de requisitos, equipe de desenvolvimento e incrementos do produto. A rede bayesiana foi construída por meio de um processo de Engenharia de Conhecimento de Redes Bayesianas. Uma base de dados, elicitada de dezoito projetos reais de uma empresa, foi coletada por meio de um questionário. Essa base de dados foi utilizada para avaliar a acurácia da predição da Rede Bayesiana. Como resultado, a previsão foi correta para quatorze projetos (acurácia de 78%). Dessa forma, conclui-se que o modelo é capaz de realizar previsões com acurácia satisfatória e, dessa forma, é útil para auxiliar nas tomadas de decisões de projetos Scrum.

Abstract

The use of Agile Software Development (ASD) is increasing to satisfy the need to respond to fast moving market demand and gain market share. In contrast with traditional plan-driven processes, ASD are people and communication-oriented, flexible, fast, lightweight, responsive, driven for learning and continuous improvement. As consequence, subjective factors such as collaboration, communication and self-management are key to evaluate the maturity of agile adoption. Scrum, which is focused on project management, is the most popular agile method. Whenever adopted, the usage of Scrum must be continuously improved by complementing it with development and management practices and processes. Even though the Retrospective Meeting, a Scrum event, is a period at the end of each sprint for the team to assess the development method, there are no clear and specific procedures to conduct it. In literature, there are several, but no consolidated, proposed solutions to assist on ASD adoption and assessment. Therefore, the research problem is: how to instrument Scrum to assist on the continuous improvement of the development method focusing on the requirements engineering process, development team and product increment? In this thesis, we propose a Bayesian networks-based process to assist on the assessment of Scrum-based projects, instrumenting the software development method to assist on its continuous improvement focusing on the requirements engineering process, development team and product increments. We have built the Bayesian network using a Knowledge Engineering Bayesian Network (KEBN) process that calculates the customer satisfaction given factors of the software development method. To evaluate its prediction accuracy, we have collected data from 18 industry projects from one organization through a questionnaire. As a result, the prediction was correct for fourteen projects (78% accuracy). Therefore, we conclude that the model is capable of accurately predicting the customer satisfaction and is useful to assist on decision-support on Scrum projects.

Conteúdo

1	Introdução	1
1.1	Problemática	4
1.2	Objetivo	7
1.3	Hipóteses	8
1.4	Metodologia	9
1.5	Sumário das Contribuições	15
1.6	Estrutura do Documento	16
2	Fundamentação Teórica	17
2.1	Desenvolvimento Ágil de Software	17
2.2	Scrum	18
2.3	Métricas de software no contexto de ASD	20
2.4	Redes Bayesianas	23
3	Rede Bayesiana Proposta	35
3.1	Construção do DAG da rede Bayesiana base	38
3.1.1	Backlog do produto	40
3.1.2	Reunião de Planejamento	41
3.1.3	Qualidade do incremento	44
3.1.4	Qualidade do trabalho em equipe	44
3.2	Definição das TPN	47
3.3	Complementação da rede Bayesiana base	50
3.4	Validação conceitual	59
3.5	Validação com Cenários de Simulação	67

3.5.1	Cenário 1	67
3.5.2	Cenário 2	68
3.5.3	Cenário 3	69
3.5.4	Cenário 4	72
3.5.5	Cenário 5	74
3.5.6	Cenário 6	74
3.5.7	Cenário 7	77
3.5.8	Cenário 8	78
3.5.9	Cenário 9	81
3.5.10	Cenário 10	82
3.5.11	Cenário 11	84
3.5.12	Cenário 12	86
3.6	Ameaças à validade	88
4	Processo Proposto	90
5	Avaliação da solução	97
5.1	Avaliação da utilidade prática do processo	97
5.1.1	Dados da Execução do Estudo de Caso	99
5.1.2	Resultados e Discussão do Estudo de Caso	102
5.1.3	Ameaças à validade	104
5.2	Avaliação da acurácia da predição	104
5.2.1	Ameaças à validade	107
6	Trabalhos Relacionados	109
7	Considerações finais	115
7.1	Trabalhos futuros	117
A	Perguntas do questionário	145
B	Respostas do questionário	164

Lista de Símbolos

ASD - Desenvolvimento Ágil de Software

CMMI - Modelo de Maturidade em Capacitação - Integração

DSDM - Dynamic Systems Development Model

GAD - Grafo Acíclico Direcionado

GQM - Goal/Question/Metric

ISD - Internet-speed Development

JIT - Just in Time

MUSE - Maxim of Uncertainty in Software Engineering

FDD - Feature-Driven Development

RUP - Rational Unified Process

SAMI - Sidky Agile Measurement Index

TDD - Test-Driven Development

TPN - Tabela de Probabilidade dos Nós

TI - Tecnologia da Informação

XP - Extreme Programming

XPMM - eXtreme Programming Maturity Model

Lista de Figuras

1.1	Visão geral do processo de melhoria continua no Scrum.	7
1.2	Ciclo de engenharia seguido.	9
1.3	Engenharia de Conhecimento de rede Bayesiana adaptado de Mendes [131].	11
2.1	Visão geral do Scrum	19
2.2	Exemplo de rede Bayesiana	24
2.3	Exemplos de expressões idiomáticas.	29
2.4	Exemplos de TNormal.	32
2.5	Exemplos de WMEAN, WMIN, WMAX e MIXMINMAX.	33
3.1	Visão geral da rede Bayesiana.	37
3.2	Fragmento <i>Backlog do produto</i>	42
3.3	Fragmento <i>Reunião de Planejamento</i>	43
3.4	Fragmento <i>Qualidade do incremento</i>	45
3.5	Fragmento <i>Qualidade do trabalho em equipe</i>	48
3.6	DAG completo.	48
3.7	Exemplo de mapeamento da rede Bayesiana base para processo ou prática complementar.	52
3.8	Exemplo de decomposição de nó auxiliar.	55
3.9	Decomposição do nó Product Owner.	56
3.10	Exemplo de adição de métricas na rede Bayesiana base.	56
3.11	Exemplo de previsão de indicador quantitativo. Imagem extraída do Agena-Risk.	58
3.12	Exemplo de definição de evidência no indicador quantitativo. Imagem extraída do AgenaRisk.	58

3.13	Exemplo de adição de métrica na rede Bayesiana base.	59
3.14	Resultado da rede Bayesiana para o antipadrão 1 de Eloranta et al. [60]. Imagem extraída do AgenaRisk.	62
3.15	Resultado da rede Bayesiana para o antipadrão 4 de Eloranta et al. [60]. Imagem extraída do AgenaRisk.	63
3.16	Análise de sensibilidade do nó <i>Reunião de Planejamento</i> com relação ao nó <i>Monitoramento de progresso</i> . Imagem extraída do AgenaRisk.	65
3.17	Análise de sensibilidade do nó <i>Qualidade da execução da sprint</i> com relação ao nó <i>Autonomia</i> . Imagem extraída do AgenaRisk.	66
3.18	Resultados calculados para o cenário 1. Imagem extraída do AgenaRisk. . .	69
3.19	Resultados calculados para o cenário 2. Imagem extraída do AgenaRisk. . .	70
3.20	Resultados calculados para o cenário 3. Imagem extraída do AgenaRisk. . .	72
3.21	Resultados calculados para o cenário 4. Imagem extraída do AgenaRisk. . .	73
3.22	Resultados calculados para o cenário 5. Imagem extraída do AgenaRisk. . .	75
3.23	Resultados calculados para o cenário 6. Imagem extraída do AgenaRisk. . .	77
3.24	Resultados calculados para o cenário 7. Imagem extraída do AgenaRisk. . .	79
3.25	Resultados calculados para o cenário 8. Imagem extraída do AgenaRisk. . .	79
3.26	Resultados calculados para o cenário 9. Imagem extraída do AgenaRisk. . .	82
3.27	Resultados calculados para o cenário 10. Imagem extraída do AgenaRisk. .	84
3.28	Resultados calculados para o cenário 11. Imagem extraída do AgenaRisk. .	86
3.29	Resultados calculados para o cenário 12. Imagem extraída do AgenaRisk. .	88
4.1	Visão geral do processo para avaliação da aplicação do Scrum.	91
5.1	Dados coletados no estudo de caso	103
5.2	Quantidade de linhas de código dos projetos.	106
5.3	Linguagens de programação utilizadas nos projetos.	106

Lista de Tabelas

1.1	Quadro de evolução da rede Bayesiana.	15
2.1	Métricas de produto utilizadas no contexto de ASD de acordo com Kupiainen et al. [114].	21
2.2	Métricas de processo utilizadas no contexto de ASD de acordo com Kupiainen et al. [114].	22
2.3	Métricas de recursos utilizadas no contexto de ASD de acordo com Kupiainen et al. [114]	23
3.1	Variância definida para os nós indicadores.	49
3.2	Calibração para os nós calibrados como método de nós ranqueados [69]. . .	51
3.3	Calibração para o nó <i>Backlog do produto</i>	52
3.4	Exemplos de associações de práticas a fatores da rede Bayesiana.	54
3.5	Exemplos de métricas que podem ser associadas com nós da rede Bayesiana base.	60
3.6	Antipadrões do Scrum apresentados em Eloranta et al. [60].	61
3.7	Evidências para o cenário 1	68
3.8	Evidências para o cenário 2	70
3.9	Evidências para o cenário 3	71
3.10	Evidências para o cenário 4	73
3.11	Evidências para o cenário 5	74
3.12	Evidências para o cenário 6	76
3.13	Evidências para o cenário 7	78
3.14	Evidências para o cenário 8	80
3.15	Evidências para o cenário 9	81

3.16	Evidências para o cenário 10	83
3.17	Evidências para o cenário 11	85
3.18	Evidências para o cenário 12	87
4.1	Exemplo de definição de priors para o nó <i>Qualidade do Product Owner</i> . . .	94
5.1	Valores reais e previstos para <i>Satisfação do cliente</i>	108
6.1	Comparação entre os trabalhos relacionados e a solução proposta.	113

Capítulo 1

Introdução

Nos últimos anos, a adoção de Desenvolvimento Ágil de Software (ASD) por empresas de tecnologia tem aumentado [36] e apresentado diversos casos de sucesso [3, 168, 128, 118, 215, 119, 176]. Em West e Grant [208], um estudo a partir de dados coletados de 1.298 profissionais de empresas de Tecnologia da Informação (TI), 39% dos respondentes afirmaram que aplicam ASD nas suas empresas.

A crescente adoção de ASD é uma reação aos problemas enfrentados pelas organizações ao desenvolver software por meio dos tradicionais processos dirigidos a planos. Em Emam e Koru [61], um estudo no qual foram coletados dados de 388 profissionais, concluiu-se que entre 26% e 34% dos projetos de desenvolvimento de software falham ou são cancelados. Neste estudo, os critérios de sucesso utilizados foram: satisfação do usuário; habilidade de alcançar metas de negócio; habilidade de alcançar metas de orçamento; qualidade do produto; e produtividade da equipe. Dentre as causas de falhas, foram identificados problemas no gerenciamento de mudanças de requisitos e escopo e desalinhamento entre negócio e TI. Em Boehm [27], as seis principais causas de falha em projetos de software foram apresentadas: requisitos incompletos; falta de envolvimento do usuário; falta de recursos; expectativas irrealistas; falta de apoio executivo; e mudanças de requisitos e especificações. A maioria desses problemas é causada por falhas de colaboração e alinhamento entre desenvolvedores e clientes. Portanto, para aumentar as chances de sucesso de projetos de software é de fundamental importância buscar soluções que proporcionem melhorias à colaboração entre os atores envolvidos.

Em contraste com os tradicionais processos dirigidos a planos, ASD é focado nas pes-

soas, orientado à comunicação, flexível, rápido, leve, responsivo e dirigido à aprendizagem e melhoria contínua [72]. Ao invés de se basear em uma abordagem linear baseada em fases, em ASD as atividades de desenvolvimento são baseadas em um processo dinâmico caracterizado por ciclos iterativos e envolvimento ativo das partes interessadas [147]. Além disso, em ASD as mudanças nos requisitos são bem-vindas e a ênfase é maior nas pessoas do que no processo. Desta forma, diferentes ciclos de vida, papéis e atividades compõem o ASD, quando comparado com o desenvolvimento tradicional de software.

De acordo com Serrador e Pinto [179], que analisaram dados de 1.002 projetos, há correlação positiva entre o uso de ASD e a eficiência e satisfação das partes interessadas do projeto, independente da experiência da equipe e complexidade do projeto. Por outro lado, na literatura há relatos de experiências negativas na adoção de ASD tais como o estouro de 3 bilhões de dólares no orçamento de um projeto da Administração Federal de Aviação dos Estados Unidos [28]. Portanto, não se pode afirmar que apenas o uso de ASD garante o sucesso do projeto. Por um lado, o uso de ASD pode acarretar em vantagens tais como redução do retrabalho, utilização mais eficiente dos recursos de testes, menor necessidade de documentação e validação precoce do produto. Por outro lado, há desvantagens tais como os desafios relativos à realização contínua de testes, maior custo de manutenção e falta de detalhamento das dependências arquiteturais [160].

Para obter sucesso com ASD, há diversos fatores envolvidos. De acordo com Misra, Kumar e Kumar [134], os principais fatores são: satisfação do cliente; colaboração do cliente; comprometimento do cliente; tempo de decisão; cultura corporativa; controle; características pessoais; cultura social; e treinamento e aprendizagem. De acordo com Chow e Cao [39], os principais fatores são associados à estratégia de entrega, o uso de técnicas ágeis de engenharia de software; e a capacidade da equipe. Além disso, foram identificados fatores auxiliares ao sucesso do projeto: o processo de gerenciamento do mesmo; ambiente da equipe; e envolvimento do cliente. Em Stankovic et al. [189], um estudo realizado apenas em empresas da antiga Iugoslávia utilizando o mesmo método de pesquisa de Chow e Cao [39], todas as conclusões do estudo dos mesmos foram refutadas; concordando apenas que apoio executivo e cultura organizacional não influenciam no sucesso de projetos ágeis.

Apesar de haver algumas contradições nos resultados dos estudos mencionados, pode-se concluir que para obter sucesso na adoção de ASD é necessária uma mudança de paradigma.

Em ASD, o foco é na aprendizagem contínua da necessidade do usuário por meio de entregas constantes. Por causa da estratégia de entrega, uma equipe colaborativa e a utilização de técnicas específicas de entregas contínuas são necessárias. A estratégia só faz sentido se o produto for validado precocemente e constantemente pelo cliente.

Métodos ágeis podem ser considerados como um meio para conseguir implementar ASD com sucesso. Na literatura, há diversos métodos ágeis propostos. Alguns com ênfase em práticas de gerenciamento de projetos tais como Scrum [193] e Internet-speed Development (ISD) [14]; outros em desenvolvimento de software tais como Agile Modeling [7] e Extreme Programming (XP) [17]; e outros com suporte a ambos tais como Crystal Clear [42], Adaptive Software Development [92], Dynamic Systems Development Model (DSDM) [190] e Feature-driven Development (FDD) [40].

Scrum é o método ágil mais popular. De acordo com o resultado de um questionário com 3.880 respondentes publicado em 2016 pela VersionOne [202], 76% dos respondentes utilizam o Scrum ou alguma variante (e.g., Scrumban [115] ou híbrido de Scrum e XP) nas suas empresas.

O Scrum é um arcabouço para gerenciamento de projetos composto de artefatos, processos, práticas de gerenciamento e papéis. O Scrum é iterativo e incremental. As iterações, chamadas de sprints, têm duração máxima de um mês e, ao final delas um incremento funcional do produto deve ser entregue para ser inspecionado e adaptado, se necessário, pelo cliente. Os requisitos são gerenciados por meio do Backlog do Produto, uma lista ordenada descrevendo o produto. Os papéis propostos são o Scrum Master, Product Owner e desenvolvedores. O Scrum Master é responsável por garantir que as teorias, práticas e regras do Scrum sejam seguidas e guiar a equipe e a organização na melhoria contínua dos seus processos [22]. O Product Owner é responsável por gerenciar o Backlog do Produto. A equipe de desenvolvimento é multifuncional e auto-organizável. O Scrum também foca na inspeção e adaptação contínua da equipe e seus processos, por meio da Reunião de Retrospectiva, realizada ao final de cada sprint. Mais detalhes sobre o Scrum são apresentados na Seção 2.2.

Ao aplicá-lo é necessário complementá-lo, de acordo com o contexto do projeto, com práticas e processos de desenvolvimento de software e gerenciamento [193] tais como TDD, Estórias do Usuário e Integração Contínua. Um exemplo de caso de sucesso em comple-

mentar Scrum com práticas XP é apresentado em Fitzgerald et al. [70]. Na literatura, há evidências de que o uso apenas das práticas do Scrum (i.e. “Flaccid Scrum” [75]) para desenvolver software resulta em redução de produtividade e qualidade do produto desenvolvido [214, 212, 216].

É neste contexto de utilização de métodos ágeis, com foco em Scrum, que se insere este trabalho. Mais especificamente, buscam-se mecanismos de monitoramento e avaliação da aplicação de Scrum na implementação de ASD em projetos de software.

1.1 Problemática

A adoção de métodos ágeis envolve um processo de seleção e adaptação de práticas ágeis, seguindo duas abordagens: baseada em contingência e engenharia de método [44]. Na abordagem baseada em contingência, a seleção assume que métodos de desenvolvimento de software não são universalmente aplicáveis, mas equipes e organizações devem selecionar um método inteiramente de acordo com o contexto do projeto. Em Alqudah e Razali [5], um modelo conceitual resultante de uma revisão sistemática na literatura para auxiliar na seleção de um método ágil de acordo com a características do projeto, envolvimento do cliente, cultura organizacional e habilidade da equipe de desenvolvimento é apresentado. Em contraste, na engenharia de método, a equipe desenvolve um novo método adaptando fragmentos de métodos existentes [32]. No contexto de ASD, a engenharia de método é a mais popular [36, 213].

Para adaptar um método de desenvolvimento, é necessário considerar o contexto do projeto. De acordo com Kalus e Kuhrmann [107], há quatro critérios a serem considerados para adaptar métodos ágeis: ambiente interno (e.g., orçamento, duração do projeto e linguagem de programação); ambiente externo (e.g., número de clientes, tipo de contrato e disponibilidade do cliente); equipe (e.g., tamanho, distribuição e conhecimento do domínio); e objetivos (e.g., complexidade, grau de inovação e segurança). Em Campanelli e Parreiras [36], mais dois critérios foram apresentados: (i) nível de maturidade, a definição e adoção de práticas de acordo com o nível de maturidade a ser atingido; e (ii) conhecimento prévio, baseado em projetos e experiências passadas da organização.

No contexto de ASD, há diversas pesquisas propondo processos para guiar a adaptação

contínua do método de desenvolvimento. Algumas propõem combinar métodos ágeis com o CMMI (Modelo de Maturidade em Capacitação - Integração). Em Selleri Silva et al. [184], resultados de uma revisão sistemática na literatura [110] explorando a combinação de métodos ágeis com CMMI são apresentados. Como resultado, foram encontrados 81 estudos e foi concluído que é possível alcançar os níveis 2 e 3 do CMMI aplicando métodos ágeis. Por outro lado, para alcançar os níveis 4 e 5, é necessário adicionar outras práticas.

Em outras pesquisas, abordagens específicas para ASD são propostas. Em Campanelli e Parreiras [36], foram apresentados resultados de uma revisão sistemática que explora a adaptação de métodos ágeis. Foram identificados 56 trabalhos propondo soluções.

Algumas soluções são modelos de maturidade específicos para um método ágil. Em Nawrocki et al. [144] e Lui e Chan [124], são apresentados modelos compostos de 4 níveis para a adoção de práticas XP. Nenhuma foi avaliada empiricamente. Em Yin et al. [223], é apresentado um modelo composto de 5 níveis, alinhado com o CMMI-DEV, para adoção do Scrum.

Modelos de maturidade genéricos para métodos ágeis também são apresentados, tal como Packlick [150], Benefield [20]. A limitação dos modelos de maturidade é a baixa flexibilidade na escolha das práticas ou dos seus níveis de aplicação, que pode não ser adequado com ASD [186]. Além disso, percebe-se o foco maior no processo, não nas pessoas.

Arcabouços para a adoção de métodos ágeis foram propostos por Sidky et al. [181], Gandomani e Nafchi [102], Qumer e Henderson-Sellers [166] e Ayed et al. [10]. O foco de Sidky et al. [181] é o processo de tomada de decisão, com apoio de uma ferramenta para mensurar a agilidade da organização, sobre quais práticas ágeis devem adotadas no nível de projeto e organizacional. Gandomani e Nafchi [102] têm como foco a transição de métodos tradicionais para ágeis, mas não recomenda práticas nem tem apoio ferramental. Qumer e Henderson-Sellers [166] apresentam um arcabouço composto de 6 etapas e com apoio de uma ferramenta para mensurar a agilidade da organização. Em Ayed et al. [10] foi apresentada uma abordagem baseada em métricas para avaliação do método de desenvolvimento.

Em Salo e Abrahamsson [174], um processo sistemático e iterativo composto de 6 passos para melhoria da adoção de ASD foi proposto. Para a definição das ações de melhoria é sugerido utilizar práticas como Análise de Causa Raíz [21], Grupo de Foco [108] e GQM [13] dependendo do conhecimento da equipe e de um facilitador externo.

Questionários para avaliação de métodos ágeis são apresentados em Williams et al. [216], Fontana et al. [73], Pikkarainen e Passoja [162] e da Cunha [48]. Todas as abordagens são baseadas em dados subjetivos.

Desta forma, pode-se concluir que ainda não há uma solução consolidada para a adoção ou avaliação contínua de métodos ágeis. Além disso, as soluções atuais sofrem de uma ou mais das seguintes limitações: tem maior foco no processo, assumem um conjunto de práticas definidas ou dependem apenas de avaliação subjetiva. Neste trabalho, aborda-se esta lacuna do ponto de vista do Scrum.

No caso do Scrum, o mesmo é baseado na teoria do controle empírico de processo e descreve um evento, a Reunião de Retrospectiva, e um papel, o Scrum Master, para promover a melhoria contínua do método de desenvolvimento [193]. A Reunião de Retrospectiva deve ocorrer ao final de cada iteração (i.e., sprint) e trata-se de uma oportunidade para a equipe inspecionar-se e criar planos de ação para tornar-se mais eficiente e eficaz. Esta prática é uma das mais populares em projetos ágeis [214] e está em conformidade com o Manifesto Ágil, o qual lista como um dos seus princípios que “em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e então refina e ajusta seu comportamento de acordo.”

Apesar disto, no Guia do Scrum [193] não há procedimentos claros e específicos para a realização da Reunião de Retrospectiva. Na literatura, há estudos que apresentam técnicas para serem utilizadas durante essas reuniões, tais como o método KJ [178, 54] e histograma de satisfação [52]. Apesar de alcançarem seus objetivos, elas não consideram meios para planejamento e validação sistemática das melhorias implementadas. Desta forma, estudos indicam que essa reunião pode se tornar longa e sem foco [125].

Durante a Reunião de Retrospectiva, diversos pontos podem ser apontados pela equipe, como problemas de relacionamento ou ambiente organizacional, oportunidades de melhoria no método de desenvolvimento, entre outros. Para que a mesma seja eficaz do ponto de vista de melhoria do método de desenvolvimento, é necessário que a equipe realize o diagnóstico das principais entidades do mesmo (e.g, Backlog do produto, Product Owner, reuniões Scrum), identificando problemas, oportunidades de melhoria e riscos. Como resultado, um plano de ação deve ser definido de forma a maximizar o valor do produto entregue e, conseqüentemente, a satisfação do cliente. Para tal, é necessário priorizar as possíveis intervenções (e.g., ações corretivas ou preventivas) no processo e realizar o prognóstico (i.e.,

previsões do impacto) de cada intervenção na satisfação do cliente ou em outro fator chave do método de desenvolvimento. Esse processo de tomada de decisão, apresentado na Figura 1.1, é complexo e, sem o apoio de dados e ferramentas, depende somente da experiência e conhecimento tácito da equipe. Por exemplo, dado que problemas foram identificados na comunicação da equipe, qualidade da inspeção do produto e descrição dos requisitos (i.e., itens de Backlog do produto), qual deles deve ser resolvido primeiro?

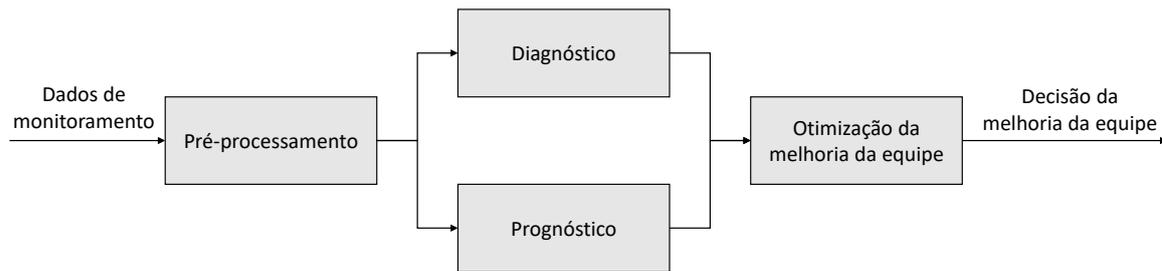


Figura 1.1: Visão geral do processo de melhoria contínua no Scrum.

Pode-se concluir que a melhoria contínua é um dos pilares do Scrum e ASD e que a sua otimização trata-se de um processo complexo, principalmente para equipes inexperientes. Apesar do Scrum descrever a Reunião de Retrospectiva como um evento no qual a equipe deve se inspecionar e adaptar para promover sua melhoria, não é descrito como isso deve ser realizado. Além disso, o processo de melhoria é baseado no conhecimento tácito e intangível da equipe. Desta forma, há a necessidade de uma abordagem sistemática e objetiva para dar suporte à melhoria contínua do método de desenvolvimento no Scrum.

Desta forma, o problema em questão é: como instrumentar o Scrum para auxiliar na melhoria contínua do método de desenvolvimento com foco na avaliação do processo de engenharia de requisitos, equipe de desenvolvimento e incrementos do produto?

1.2 Objetivo

O objetivo da pesquisa é propor um processo para a avaliação da aplicação do Scrum em projetos de software, instrumentando o método para auxiliar sua melhoria contínua.

A instrumentação provê suporte ao diagnóstico e prognóstico do método de desenvolvimento. Além disso, está em conformidade com os princípios Scrum e assume a ausência de

volume de dados históricos, pois os mesmos nem sempre estarão disponíveis.

Para a instrumentação, modelou-se o método de desenvolvimento representando suas principais entidades e a relação causal entre as mesmas considerando incertezas nas medições (i.e., observações) e relacionamentos. Para isso, foram utilizadas redes Bayesianas [152]. As motivações para a escolha de Redes Bayesianas são apresentadas no Capítulo 3.

As seguintes questões de pesquisa foram definidas:

- **RQ1:** Como construir uma rede Bayesiana para modelar o método de desenvolvimento Scrum?
- **RQ2:** Como definir um processo, com apoio da rede Bayesiana, para auxiliar na tomada de decisão para a melhoria contínua do método de desenvolvimento Scrum com foco na avaliação do processo de engenharia de requisitos, equipe de desenvolvimento e incrementos do produto.?
- **RQ3:** Qual a utilidade prática da processo proposto?
- **RQ4:** Qual a acurácia das predições da rede Bayesiana proposta?

1.3 Hipóteses

As hipóteses da pesquisa são apresentadas a seguir.

H1. É possível utilizar a engenharia de conhecimento para construir uma rede Bayesiana para prever a satisfação do cliente e dar apoio ao processo de diagnóstico do método de desenvolvimento Scrum.

H2. É possível definir um processo para utilização da rede Bayesiana para dar apoio à melhoria contínua do método de desenvolvimento Scrum com foco na avaliação do processo de engenharia de requisitos, equipe de desenvolvimento e incrementos do produto.

A prova é por demonstração. Foi construída uma rede Bayesiana para a modelagem das principais entidades do método de desenvolvimento Scrum e suas relações causais por meio de um processo de engenharia de conhecimento. Avaliações foram realizadas para demonstrar a acurácia da predição da rede Bayesiana.

1.4 Metodologia

De acordo com o objetivo desta pesquisa, a mesma classifica-se como de proposta de solução, validação e avaliação [209]. Seguindo o ciclo de engenharia proposto em Wieringa [210], depois de entender o problema em questão, a solução proposta (i.e., design de tratamento) deve ser desenvolvida. Inicialmente, a mesma é validada no laboratório (e.g., simulação). Caso tenha obtido sucesso na etapa de validação, a mesma deve ser transferida para o contexto real e ter a sua interação com o mesmo avaliada (e.g., questionário e estudo de caso). Dessa forma, pode-se concluir que o ciclo de engenharia seguido, apresentado na Figura 1.2, é composto de cinco etapas: (i) Investigação do problema, (ii) Desenvolvimento da solução (i.e., tratamento), (iii) Validação da solução, (iv) Implementação (i.e., transferência) da solução, e (v) Avaliação da implementação.

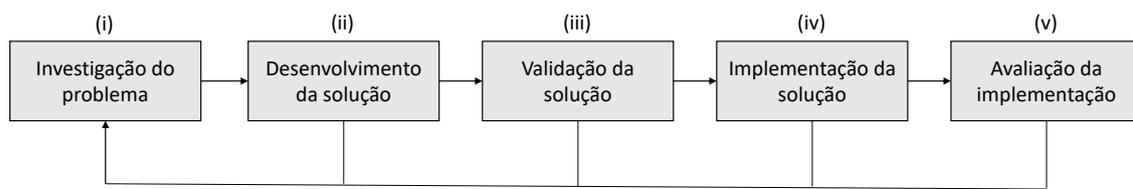


Figura 1.2: Ciclo de engenharia seguido.

A execução desta pesquisa segue um processo em espiral, onde em cada ciclo uma ou mais etapas do ciclo de design seguido foram executadas. Para a realização da mesma, definem-se as seguintes atividades e seus relacionamentos com as questões de pesquisa apresentadas na Seção 1.2:

1. identificar as principais entidades do método de desenvolvimento baseado em Scrum e seus relacionamentos (**RQ1**);
2. construir a rede Bayesiana que modela o relacionamento entre as principais entidades do método de desenvolvimento baseado em Scrum (**RQ1**);
3. definir o processo para utilização da rede Bayesianas no apoio à tomada de decisões de melhoria do método de desenvolvimento em projetos Scrum (**RQ2**);
4. validar a rede Bayesiana proposta (**RQ4**);

5. avaliar a rede Bayesiana proposta (**RQ4**);
6. avaliar o processo proposto (**RQ3**);

As atividades 1, 2, e 3 estão relacionadas com a etapa (ii); a atividade 4, com a etapa (iii); a atividade 5, com as etapas (iv) e (v); a atividade 6 com as etapas (iv) e (v); e a atividade 7 com a etapa (iv). Para a construção, validação e avaliação da rede Bayesiana (atividades 1, 2, 4 e 5), o processo de Engenharia de Conhecimento de Rede Bayesiana (KEBN) apresentado em Mendes [131] foi adotado e adaptado. O mesmo é apresentado na Figura 1.3.

O primeiro passo do KEBN, cujo o objetivo é construir a estrutura da rede Bayesiana, é composto de um subprocesso iterativo para identificar as variáveis, seus estados e o relacionamento entre as mesmas. No contexto dessa pesquisa, todas as variáveis são mensuradas em uma escala ordinal. Como fonte de informação, pode-se eliciar o conhecimento de especialistas de domínio e dados da literatura ou projetos passados. As mesmas fontes de informação podem ser utilizadas para o segundo passo: definição das funções de probabilidade. Nesta pesquisa, todas as funções de probabilidade dos nós filho foram definidas por meio do método de nós ranqueados [69], o qual é detalhado na Seção 2.4. Para os nós folha, foi assumida uma distribuição uniforme.

Para validação do modelo, cenários de simulação foram utilizados. Os mesmos representam cenários de casos reais que são preparados e utilizados pelo especialista de domínio para avaliar se a predição calculada apresenta maior probabilidade correspondente à predição que o especialista de domínio faria baseado no seu conhecimento [131].

Finalmente, a avaliação do modelo é realizada por meio da acurácia da predição de dados de projetos reais. Evidências correspondentes a dados coletados nos projetos são inseridas no modelo e avalia-se se as predições do modelo estão em conformidade com os dados dos resultados dos projetos. Nos casos nos quais a predição de maior probabilidade do modelo não esteja em conformidade com o resultado do projeto, há três explicações possíveis: o modelo está incorreto, o resultado do projeto foi inesperado ou a predição estava fora do escopo do modelo. Para o último caso, por exemplo, pode ser que o cliente esteja insatisfeito devido a problemas contratuais e isso não está no escopo do modelo dessa pesquisa, que se restringe ao método de desenvolvimento.

Inicialmente, foi desenvolvida uma versão da rede Bayesiana (atividades 1, 2 e 4), a

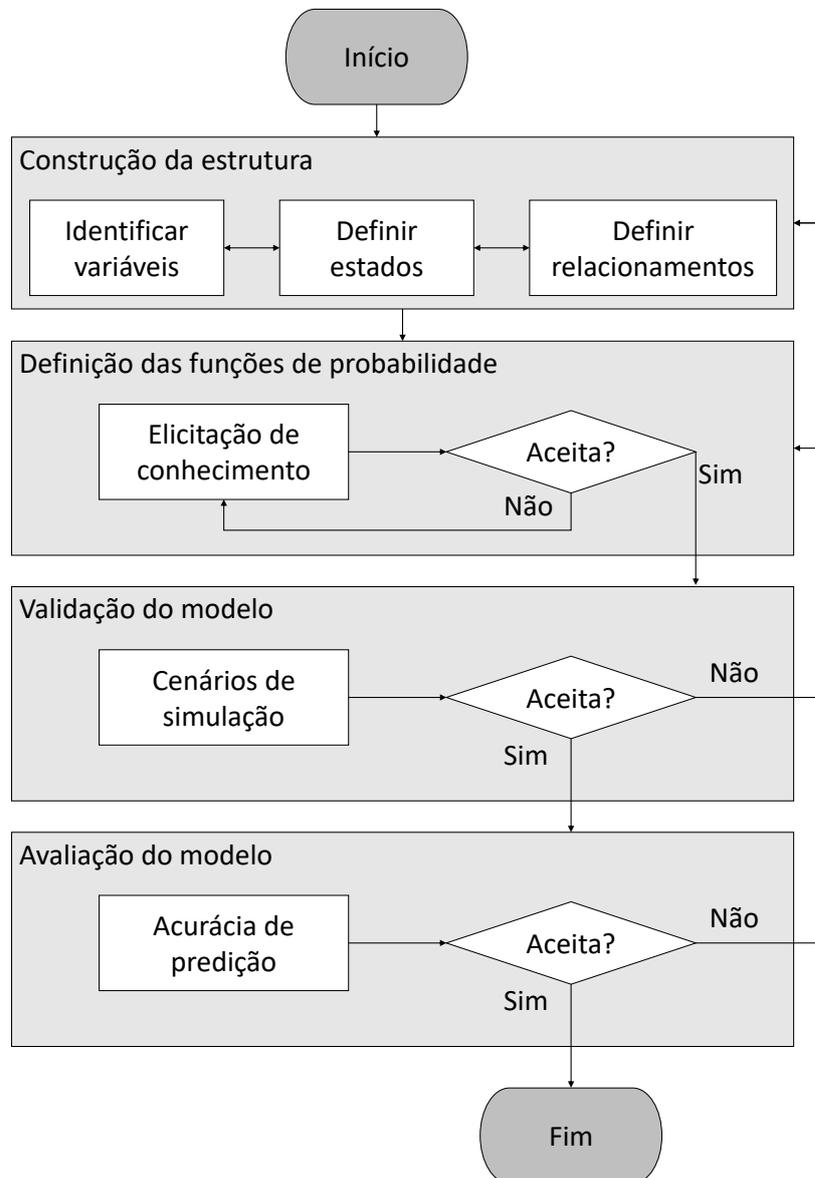


Figura 1.3: Engenharia de Conhecimento de rede Bayesiana adaptado de Mendes [131].

qual foi publicada em Perkusich et al. [155]. A mesma é composta das principais entidades (i.e., artefatos, recursos e atividades) do método de desenvolvimento e algumas práticas recomendadas para projetos Scrum. As mesmas foram identificadas a partir de uma revisão na literatura, elicitaco de conhecimento de especialistas e um questionrio respondido por 39 profissionais na indstria. Desta forma, ela é considerada um modelo genérico que deve ser modificado para se adaptar ao contexto de uma equipe em questo. Para construir a mesma, baseou-se no método de definio de TPN apresentado em Perkusich et al. [158].

A primeira verso do processo (atividade 3), tendo como base a primeira verso da rede Bayesiana, foi publicada em Perkusich et al. [159]. O mesmo é iterativo, contínuo e alinhado com práticas de gerenciamento do Scrum. A utilidade prática do processo foi avaliada em um estudo de caso com 2 equipes (atividade 6) no qual ficou demonstrada sua utilidade prática. A primeira verso, juntamente com sua avaliao, é apresentada em Perkusich [154]. No Capítulo 5.1, os resultados do estudo de caso so apresentados.

Com os resultados do estudo de caso, foram implementadas melhorias na rede Bayesiana e uma nova verso foi apresentada em Perkusich et al. [157]. O foco desta verso foi aumentar a acurcia do modelo. As mudanas foram definidas de acordo com o conhecimento de domnio e de redes Bayesianas do pesquisador. Foram realizadas mudanas no grafo, com o intuito de reduzir a complexidade para definio das funoes de probabilidade dos ns filho (i.e., o nmero mdio de pais por no filho) e recalibrar as funoes de probabilidade. Alm disso, o modelo foi incrementado com a adio de mtricas objetivas (i.e., cobertura de testes de unidade). A mesma foi validada com os mesmos cenrios de simulao utilizadas para a primeira verso. Como resultado, a acurcia do modelo melhorou: o Brier score mdio [31] foi reduzido de 0,558 para 0,002. O nmero de ns no grafo aumentou de 73 para 76. A complexidade para definio das funoes de probabilidade dos ns filho (i.e., o nmero mdio de pais por no filho) foi reduzida de 3,47 para 2,77. Por outro lado, o esforo para definio da rede se manteve elevado, o modelo inclui práticas que no so obrigatrias no Scrum (e.g., a prática *Planejamento da release* era citada no Guia do Scrum [193], mas foi retirada na ltima verso do mesmo) e o fragmento *Qualidade da equipe* foca mais no processo do que nas caractersticas da equipe em si.

Em Freire et al. [185], o fragmento que modela os construtos, indicadores e preditores da qualidade do trabalho em equipe foi redefinido. Em Freire et al. [76], ainda em avaliao,

uma segunda versão avaliada em um estudo de caso com 3 equipes foi apresentada (Atividade 6). O fragmento resultante é composto de 17 nós, dos quais 9 são nós folha e os nós filho tem em média 2,12 de complexidade. As melhorias na rede Bayesiana apresentadas em Freire et al. [185] e Freire et al. [76] foram realizadas em colaboração com Arthur Freire, então aluno de Mestrado no Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande.

De forma a corrigir os problemas apresentados na segunda versão da rede Bayesiana e incorporar os resultados de Freire et al. [76], uma terceira versão foi apresentada em Perkusich et al. [156]. O principal foco foi separar os conceitos de entidade do método de desenvolvimento, métricas e práticas de modo a definir um modelo base com o mínimo de nós representando todos os pilares do Scrum. Métricas e práticas podem ser adicionados para complementar o mesmo. As mudanças foram realizadas a partir do meu conhecimento de domínio e de redes Bayesianas. Dessa forma, nessa versão da solução, o processo também foi alterado para atender às modificações da rede Bayesiana. Essa versão foi validada com relação a sua completude (i.e., capacidade de mapear os principais anti-padrões do Scrum), onde dos 14 antipadrões válidos apresentados em Eloranta et al. [60], 12 foram diretamente detectados pelo modelo, 1 foi indiretamente detectado e 1 foi considerado inválido por estar fora do escopo do mesmo. Como resultado, o número de nós da rede Bayesiana foi reduzido para 45 (onde 26 são nós folha), e a complexidade para 2,42.

Um grupo de foco, composto por três especialistas de domínio e com o pesquisador como mediador, foi formado para avaliar a terceira versão da rede Bayesiana. Como resultado, foram realizadas mudanças na terminologia e na estrutura da mesma. Além disso, com o intuito de melhorar a distinção de métricas e preditores, o conceito de nós indicadores (i.e., Classificador Nãive Bayes) [69] foi aplicado. Como resultado, o modelo base definitivo foi concluído. Vale salientar que, como discutido em Mendes [131], modelos desse tipo não devem se estagnar. Ou seja, ao implantar o modelo em projetos reais, o contexto da empresa pode mudar com o tempo e, por isso, é importante manter um processo de calibração e validação contínua. A acurácia da predição do modelo foi avaliada com dados de 18 projetos de uma empresa, dos quais a previsão foi correta em 78% dos casos. O número de nós da rede Bayesiana foi reduzido para 43 (onde 24 são nós folha e 11 são indicadores) e a complexidade para, 1,70. O modelo definido é apresentado no Capítulo 3; o processo, no

Capítulo 4; os resultados da avaliação empírica, no Capítulo 5.

Além disso, foram executadas atividades para facilitar a implantação do processo na prática. Em da Silva et al. [49], uma solução baseada em regras de produção é apresentada para calibrar a rede Bayesiana sem necessidade de um especialista em nós ranqueados [69] (atividade 2). Com isto, aumenta-se a aplicabilidade da solução proposta por esta pesquisa. Esta atividade foi realizada em colaboração com Raissa Silva, então aluna de Mestrado no Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande.

Em Willamy et al. [211], apresenta-se um método para construir redes bayesianas baseadas em artefatos e métricas para avaliar projetos ágeis. Para tal, o paradigma Goal/Question/Metric (GQM) [13] foi utilizado. Esta solução pode ser aplicada para, ao definir as entidades do modelo desta pesquisa, selecionar as métricas apropriadas e complementar a rede Bayesiana (atividade 2). Esta atividade foi realizada em colaboração com Renan Willamy, então aluno de Iniciação Científica no Programa Institucional de Bolsas de Iniciação Científica e Tecnológica do Instituto Federal da Paraíba.

Finalmente, foi desenvolvido um algoritmo para implementar nós ranqueados [69]. O mesmo é apresentado em Bezerra et al. [25] e Bezerra et al. [24]. Com este algoritmo, possibilita-se o desenvolvimento de uma ferramenta própria para implementar a rede Bayesiana desta pesquisa. Esta atividade foi realizada em colaboração com João Bezerra, então aluno de Iniciação Científica no Programa Institucional de Bolsas de Iniciação Científica e Tecnológica do Instituto Federal da Paraíba.

Na Tabela 1.1, um quadro de evolução da rede Bayesiana é apresentado. Como pode ser observado, a última versão da rede Bayesiana é mais simples (i.e., menos nós e menor complexidade) e mais flexível para a utilização de outros tipos de dados para avaliar o método de desenvolvimento. A evolução da rede Bayesiana se deve ao refinamento progressivo da mesma de acordo com os resultados das validações e avaliações. Além disso, se deve ao refinamento do processo de utilização da mesma, que contém mais instruções sobre como utilizar diferentes fontes de dados, tais como questionários e ferramentas de desenvolvimento de software, apresentado no Capítulo 4.

Tabela 1.1: Quadro de evolução da rede Bayesiana.

Versão	Número de nós	Complexidade	Escala
1 [155, 159]	73	3,47	Ordinal
2 [157]	76	2,77	Ordinal
3 [156]	45	2,42	Ordinal
4	43	1,70	Ordinal Booleano Absoluto Proporção Nominal

1.5 Sumário das Contribuições

As principais contribuições desta tese são apresentadas a seguir.

- É apresentada uma rede Bayesiana para prognóstico da satisfação do cliente em função da medição de atributos das principais entidades do método de desenvolvimento Scrum e para diagnóstico do mesmo [155, 157, 156];
- É apresentado um processo sistemático com apoio da rede Bayesiana para a melhoria contínua do método de desenvolvimento Scrum [159, 156], no qual as melhorias são identificadas e implantadas durante o projeto em andamento;
- São reportados resultados da validação da rede Bayesiana por meio de cenários de simulação [155, 157, 156];
- São reportados resultados da avaliação da utilidade prática do processo proposto por meio de um estudo de caso em dois projetos [159];
- São reportados resultados da avaliação da acurácia da predição da satisfação do cliente utilizando a rede Bayesiana proposta por meio de dados coletados de dezoito projetos;
- É apresentado um caso de como utilizar a engenharia de conhecimento para a construção de uma rede bayesiana [157];

- É apresentado um caso de como utilizar a metodologia Design Science executada por meio de um processo em espiral para o desenvolvimento de uma solução inovadora dentro de um contexto com incerteza;

1.6 Estrutura do Documento

O restante do documento está organizado da seguinte forma:

- No Capítulo 2, a fundamentação teórica da pesquisa é apresentada. Os conceitos de ASD, Scrum, métricas de software no contexto de ASD e redes bayesianas são introduzidos;
- No Capítulo 3, detalhes sobre a rede Bayesiana construída são apresentados, incluindo o seu processo de construção e validação;
- No Capítulo 4, o processo proposto é apresentado, destacando suas etapas e como o mesmo pode ser aplicado para auxiliar na tomada de decisões durante as Reuniões de Retrospectiva;
- No Capítulo 5 são apresentados os resultados da avaliação da utilidade prática do processo e da acurácia das predições da rede Bayesiana;
- No Capítulo 6, os trabalhos relacionados são apresentados; e
- Por fim, no Capítulo 7 são apresentadas as considerações finais, sendo discutidos os resultados e sugestões para trabalhos futuros.

Capítulo 2

Fundamentação Teórica

2.1 Desenvolvimento Ágil de Software

Em 2001, um grupo de engenheiros de software se reuniu para definir uma nova metodologia com princípios e valores que se adaptem à crescente taxa de mudanças de requisitos e expectativas dos clientes [213]. O resultado dessa reunião foi a criação da *Agile Alliance* e do Manifesto Ágil [126]. O Manifesto Ágil expressa os valores da metodologia ágil: indivíduos e interações mais que processos e ferramentas; software em funcionamento mais que documentação abrangente; colaboração com o cliente mais que negociação de contratos; e responder a mudanças mais que seguir um plano. Além disso, o Manifesto Ágil fornece doze princípios para guiar equipes ágeis:

- Satisfazer o cliente a partir da entrega contínua e adiantada de software com valor agregado.
- Mudanças nos requisitos são bem-vindas, mesmo tardiamente no desenvolvimento. Processos ágeis tiram vantagem das mudanças visando vantagem competitiva para o cliente.
- Entregar frequentemente software funcionando, de poucas semanas a poucos meses, com preferência à menor escala de tempo.
- Pessoas de negócio e desenvolvedores devem trabalhar diariamente em conjunto por todo o projeto.

- Construir projetos em torno de indivíduos motivados. Dê a eles o ambiente e o suporte necessário e confie neles para fazer o trabalho.
- O método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é por meio de conversa face a face.
- Software funcionando é a medida primária de progresso.
- Os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente.
- Contínua atenção à excelência técnica e boa arquitetura aumenta a agilidade.
- Simplicidade - a arte de maximizar a quantidade de trabalho não realizado - é essencial.
- As melhores arquiteturas, requisitos e modelos emergem de equipes auto-organizáveis.
- Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e então refina e ajusta seu comportamento de acordo.

2.2 Scrum

O Scrum foi criado no início dos anos 90 e, atualmente, é o método ágil mais utilizado na indústria [202]. De acordo com o resultado de um questionário com 3.880 respondentes publicado em 2016 pela VersionOne [202], 75% dos respondentes utilizam o Scrum ou alguma variante (e.g., Scrumban [115] ou híbrido de Scrum e XP) nas suas empresas. O objetivo do Scrum é prover um arcabouço para construir produtos complexos. O mesmo deve ser complementado com processos e técnicas [193]. O Scrum utiliza uma abordagem iterativa e incremental para otimizar a previsibilidade e controlar riscos. Ao final de cada iteração, chamada de sprint, um incremento funcional do produto é entregue para ser avaliado pelas partes interessadas. Três pilares sustentam o Scrum: transparência, inspeção e adaptação. Ao final de cada sprint, é entregue um incremento funcional do produto para ser avaliado pelos clientes. Aspectos significativos do processo (Definição de Pronto e critérios de aceitação, por exemplo) devem ser visíveis (transparentes) para os envolvidos. Os envolvidos devem

inspecionar os artefatos e progresso relativo ao objetivo frequentemente para detectar variações indesejáveis. Caso algum desvio indesejável seja detectado no processo, o mesmo deve ser modificado para se adaptar às circunstâncias. Uma visão geral do Scrum é apresentada na Figura 2.1.

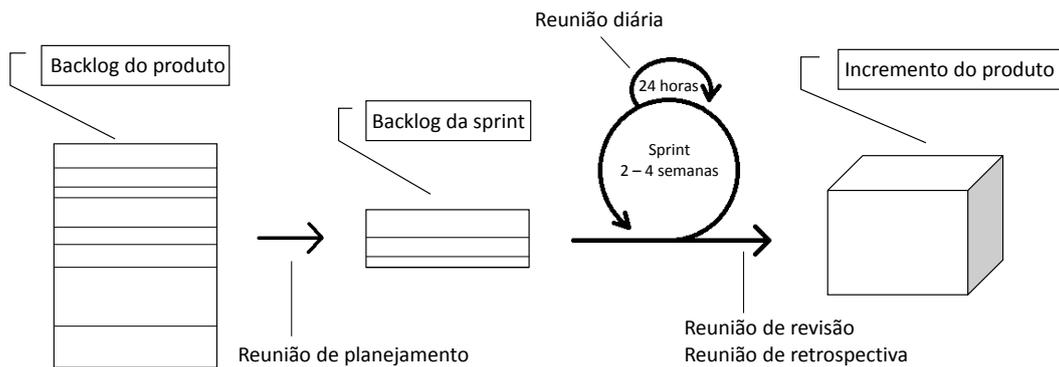


Figura 2.1: Visão geral do Scrum

Os dois artefatos principais são: Backlog do produto e Backlog da sprint. O Backlog do produto é uma lista ordenada que representa todas as funcionalidades, requisitos, melhorias e correções de bugs do produto. A ordenação é baseada em valor, risco, prioridade e necessidade. Além disso, o Backlog do produto é evolutivo. Ou seja, ele se adapta às mudanças requisitadas pelos clientes durante a execução do projeto. O Backlog da sprint é composto por itens do Backlog do produto alocados para a sprint e o planejamento para alcançar o objetivo da sprint. Além dos artefatos já citados, na literatura, artefatos de monitoramento de progresso são apresentados tais como *Sprint burndown*, *Release Burndown graph* e *Release Burndown bar* [161].

Os três papéis essenciais são: Product Owner, Scrum Master e desenvolvedor. O Product Owner é responsável por maximizar o valor do produto e o trabalho dos desenvolvedores. Ele é responsável pela interface entre a equipe técnica e a equipe de negócios envolvida no projeto e por gerenciar o Backlog do produto. O Scrum Master age como um líder-servidor

e é responsável por garantir que as teorias, práticas e regras do Scrum sejam seguidas. Ele é responsável por guiar a equipe e a organização na melhoria contínua dos seus processos [22]. O mesmo auxilia a equipe na tomada de decisões acerca do uso do Scrum e garante que os princípios do Scrum e ASD estão sendo seguidos. Além disso, o Scrum Master facilita reuniões, remove impedimentos, auxilia no planejamento das sprints (e.g., auxiliar na decomposição de tarefas técnicas) e gerenciamento do Backlog do produto [16, 18] e, em um contexto de projeto distribuído, facilita o Scrum of Scrums e dá suporte à integração do código [15].

O desenvolvedor é responsável por realizar qualquer trabalho necessário para entregar o incremento do produto ao final da sprint, tais como: construir arquitetura, programar e testar. Além destes papéis, na prática, há a figura do cliente. O cliente representa os interesses de negócio do projeto. Geralmente, um projeto Scrum é composto por uma equipe de desenvolvedores, um Scrum Master, um Product Owner e diversos clientes.

No Scrum há quatro reuniões essenciais: Planejamento, Diária, Retrospectiva e Revisão. As Reuniões de Planejamento ocorrem no início de cada sprint com o objetivo de planejar o trabalho e objetivo da mesma. As Reuniões Diárias ocorrem diariamente com o objetivo de inspecionar o andamento da sprint e sincronizar o trabalho para diminuir os riscos. As Reuniões de Revisão ocorrem ao final de cada sprint com o objetivo de inspecionar o produto produzido na sprint e, se necessário, adaptar o Backlog do produto às mudanças requisitadas pelos clientes. As Reuniões de Retrospectiva ocorrem após as reuniões de revisão com o objetivo de inspecionar a sprint com relação às pessoas, relacionamentos, processos e ferramentas; identificar problemas e criar um plano para aplicar as melhorias.

2.3 Métricas de software no contexto de ASD

Métricas de software são utilizadas como uma tentativa de avaliar ou prever um atributo de uma entidade (produto, processo ou recurso) [68]. Atributos podem ser internos ou externos. Atributos internos podem ser mensurados diretamente. Exemplos de atributos internos são tamanho do código e experiência do desenvolvedor. Atributos externos, geralmente, são os de interesse e visíveis externamente. Exemplos de atributos externos são qualidade do produto e produtividade da equipe.

Tabela 2.1: Métricas de produto utilizadas no contexto de ASD de acordo com Kupiainen et al. [114].

Entidade	Atributos internos	Atributos externos
Produto	Running Tested Features, situação da build	Satisfação do cliente, progresso como código em funcionamento
Planos de teste	Número de casos de teste	
Código	Débito técnico, violações de análise estática	
Funcionalidades	data de entrega de tarefa, tarefas finalizadas, estimativa de esforço, porcentagem, de story point	Valor de negocio entregue
Requisitos	Custo de requisitos, porcentagem de itens preparados para sprint	
Defeitos		Tendência de defeitos

Tabela 2.2: Métricas de processo utilizadas no contexto de ASD de acordo com Kupiainen et al. [114].

Entidade	Atributos internos	Atributos externos
Testes	Número de defeitos, taxa de sucesso de testes, taxa de falha de testes, cobertura de testes, taxa de crescimento de testes, defeitos adiados	número de avaliação de defeitos, número de defeitos deslizados
Implementação	velocidade, número de testes de unidade, páginas web completadas, índice de custo de performance, índice de cronograma de performance, velocidade planejada, número de check-ins por dia, tempo para correção de build quebrada	porcentagem de fluxo de itens
Engenharia de requisitos	tempo para elaborar requisitos	
O todo		cycle time, lead time, processing time, queue time, esforço de manutenção, requisitos implementados e desperdiçados, pilha, trabalho em andamento

Tabela 2.3: Métricas de recursos utilizadas no contexto de ASD de acordo com Kupiainen et al. [114]

Entidade	Atributos internos	Atributos externos
Equipe		Eficácia
Cliente	Retorno de investimento por cliente	

Em Jones [106], após análise em uma base de dados de milhares de projetos de software, foi concluído que nas empresas mais eficientes (e.g., IBM e Microsoft) usa-se métricas exaustivamente. Por outro lado, equipes ineficientes não utilizam métricas. Tradicionalmente, métricas são utilizadas para: planejar e estimar projetos; controlar e monitorar projetos; avaliar qualidade e objetivos de negócios; e melhorar a comunicação, processos e ferramentas de desenvolvimento de software [80, 165].

Em Kupiainen et al. [114], resultados de uma revisão sistemática na literatura acerca da utilização de métricas no contexto de ASD na indústria são apresentados. Um total de 30 estudos primários foram identificados como relevantes pelos autores.

Há diversos tipos de métricas ágeis. As mesmas são aplicadas em produtos, planos de teste, código, *build*, funcionalidades, requisitos e defeitos. Segundo Kupiainen et al. [114], as mais relevantes em ASD são velocidade, estimativa de esforço e satisfação do cliente. Elas são utilizadas para planejar iterações, monitorar progresso, mensurar a qualidade do produto, corrigir problemas no processo de desenvolvimento e motivar pessoas [114]. As métricas apresentadas em Kupiainen et al. [114] para produtos, processos e recursos são apresentadas, respectivamente, nas Tabelas 2.1, 2.2 e 2.3.

2.4 Redes Bayesianas

Redes bayesianas pertencem à família de modelos gráficos probabilísticos e são usadas para representar incertezas de um domínio [19]. Elas representam variáveis aleatórias e suas dependências condicionais a partir de um grafo acíclico direcionado (GAD). As variáveis aleatórias são representadas pelos nós e as dependências condicionais pelos arcos. Além disto, os nós são associados com funções de probabilidade que têm como entrada os conjun-

tos de valores dos nós pais e calculam a probabilidade da variável representada pelo nó. Os nós são representados por círculos. Os arcos são representados por setas. Uma função de probabilidade, geralmente, é representada com uma Tabela de Probabilidade dos Nós (TPN). Nota-se que apesar dos arcos representarem a direção da conexão causal entre as variáveis, informações podem propagar em qualquer direção no grafo [151]. Um exemplo de rede Bayesiana é apresentado na Figura 2.2.

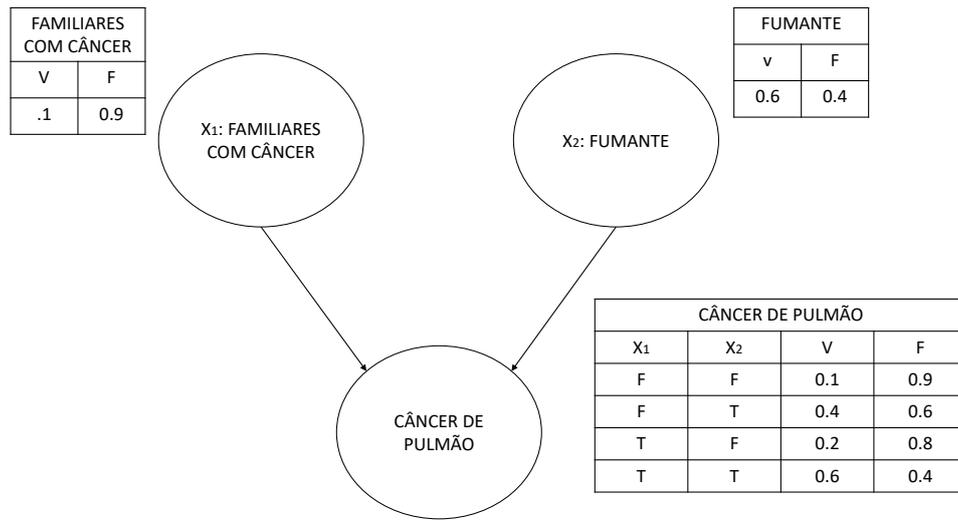


Figura 2.2: Exemplo de rede Bayesiana

Neste caso, deseja-se calcular a probabilidade de uma pessoa ter câncer de pulmão dado o histórico de câncer na sua família e se ela é fumante. As TPN dos nós pai representam o conhecimento a priori com relação a essas variáveis. Por exemplo, para X_1 , assume-se que 10% dos familiares da pessoa tem câncer; para X_2 , que 60% dos pacientes do hospital em questão é fumante. O valor de X_2 pode ser atualizado com uma evidência, caso a informação em questão seja coletada. A NPT do nó filho representa as probabilidades de uma pessoa ter câncer dada cada combinação possível de valores dos nós pai. Os valores das NPT podem ser definidos de acordo com dados históricos de pacientes ou conhecimento de especialistas (i.e., no caso, um oncologista).

Formalmente, uma rede Bayesiana B é um grafo acíclico direcionado que representa a distribuição conjunta de probabilidade sobre o conjunto de variáveis aleatórias V [77]. A rede é definida pelo par $B = \{G, \Theta\}$. G é o grafo acíclico direcionado cujo os nós X_1, \dots, X_n representam variáveis aleatórias, e os arcos representam dependências diretas

entre essas variáveis. O grafo G codifica suposições de independência, nas quais cada variável X_i é independente dos seus *não-dependentes* dados seus *pais* em G . Θ representa o conjunto de parâmetros da rede. Esse conjunto contém o parâmetro $\theta_{x_i|\pi_i} = P_B(x_i|\pi_i)$ para cada x_i em X_i condicionado por π_i , o conjunto de parâmetros de X_i em G . Equação 2.1 apresenta a distribuição conjunta definida por B sobre V .

$$P_B(X_1, \dots, X_n) = \prod_{i=1}^n P_B(x_i|\pi_i) = \prod_{i=1}^n \theta_{X_i|\pi_i} \quad (2.1)$$

A utilização de redes bayesianas tem vários pontos positivos tais como adequação para base de dados pequenas ou incompletas, aprendizagem estrutural, combinação de diferentes fontes de conhecimento, tratamento explícito da incerteza, suporte para análise de decisão e respostas rápidas [199]. Desta forma, as mesmas são aplicadas para dar suporte a sistemas com incerteza [120]. Redes bayesianas vêm sendo utilizadas no contexto de engenharia de software em vários contextos [133] tais como planejamento de projeto [153, 51], gerenciamento de riscos [62, 63, 104, 97] e gerenciamento da qualidade [103]. No contexto de métodos ágeis, essa técnica foi aplicada para detecção de antipadrões no gerenciamento de projetos [180], suporte ao planejamento do projeto [143], previsão de velocidade da equipe (i.e., produtividade) [89], gerenciamento de riscos [8], gerenciamento da qualidade [2] e estimativa de esforço [58].

O problema de construir uma rede Bayesiana pode ser decomposto em dois subproblemas: construir o grafo acíclico direcionado (DAG) e definir as TPN [146]. O DAG pode ser construído a partir apenas de dados utilizando-se de técnicas de aprendizagem de máquina ou busca, tais como as apresentadas em Neapolitan [145], Spirtes et al. [188], van Dijk et al. [200], Tsamardinos et al. [196], Zhang et al. [225], Villanueva e Maciel [203], Julia Flores et al. [71] e Gasse et al. [78].

Para otimizar a definição do DAG, pode-se complementar os dados com o conhecimento de especialistas. Uma abordagem é consultar os especialistas sobre as probabilidades posteriores da estrutura para reduzir o espaço de busca, tais como apresentado em Heckerman [90], Heckerman et al. [91], Buntine [33] e Imoto et al. [101].

Outra abordagem híbrida é utilizar o especialista durante o processo de aprendizagem, na qual o mesmo apenas é consultado quando há incerteza na relação entre as variáveis de acordo com os dados disponíveis como apresentado em Cano et al. [37], Masegosa e

Moral [127], Murphy [142], Meganck et al. [129] e Tong e Koller [195].

Em Constantinou et al. [47] e Constantinou et al. [45] são apresentadas soluções para complementar uma rede Bayesiana com o conhecimento de especialistas de domínio por meio da adição de novos fatores no modelo. Desta forma, possibilita-se a previsão de eventos raros, muitas vezes não representados nas bases de dados disponíveis.

Finalmente, o DAG pode ser definido apenas em função do conhecimento de especialistas, que é o caso desta pesquisa, na qual assume-se que não há dados disponíveis antes do processo de construção do DAG. Neste caso, o DAG pode ser definido de acordo com o conhecimento elicitado de um ou múltiplos especialistas, como apresentado em Xian-Xuan et al. [219] e Richardson e Domingos [170].

Ao definir o DAG, é necessário definir a relação de dependência entre as variáveis aplicando as propriedades de separação direcional (d-separação) [152]. Dado um grafo direcionado, um caminho entre dois conjuntos de nós \mathbf{X} e \mathbf{Y} é qualquer sequência de nós entre um membro de \mathbf{X} e um membro de \mathbf{Y} tal que todos os pares adjacentes de nós são conectados por um arco e cada nó apenas aparece uma vez na sequência. Um caminho é *bloqueado*, dado um conjunto de nós \mathbf{E} , se existir um nó Z no caminho para a qual uma das condições a seguir é verdadeira:

1. Z pertence à \mathbf{E} e Z tem um arco no caminho apontando para Z e um arco apontando de Z (i.e., corrente).
2. Z pertence à \mathbf{E} e Z tem ambos os arcos apontando para fora (i.e., causa comum).
3. Nem Z e nenhum descendente de Z pertence à \mathbf{E} , e ambos os arcos apontam para Z (i.e., efeito comum)

Um conjunto de nós \mathbf{E} d-separa dois conjuntos de nós \mathbf{X} e \mathbf{Y} se todos os caminhos de um nó \mathbf{X} para um nó pertencente à \mathbf{Y} são bloqueados dado \mathbf{E} . Se \mathbf{X} e \mathbf{Y} são d-separados por \mathbf{E} , então \mathbf{X} e \mathbf{Y} são condicionalmente independentes em \mathbf{E} . Em Boneh et al. [30] uma ferramenta chamada Matilda foi apresentada com o intuito de auxiliar não-especialistas em redes Bayesianas a aplicar as propriedades de d-separação ao construir o DAG.

Para exemplificar a propriedade 1, considere a estrutura $A \rightarrow B \rightarrow C$. Não se pode afirmar que $A \perp C$. Por outro lado, pode-se afirmar que $A \perp C | B$ e $C \perp A | B$. Para a propriedade

2, considere a configuração $A \rightarrow B \leftarrow C$. Pode-se afirmar que $A \perp B$, se não houver evidências nos nós. Por outro lado, caso haja evidência em B e um dos nós-pai (e.g., A), o outro pai (e.g., B) é dependente. Para a propriedade 3, considere a configuração $A \leftarrow B \rightarrow C$. Essa configuração é também conhecida como Classificador Nãive Bayes, onde os nós A e C são considerados nós indicadores [69]. Nesse caso, não se pode afirmar que $A \perp C$. Por outro lado, pode-se afirmar que $A \perp C | B$ e $C \perp A | B$.

Além das propriedades de d-separação, para escolher a direção do arco é necessário levar em consideração quais informações relativas às TPN fazem mais sentido serem elicitadas dos especialistas. Por exemplo, é mais fácil o especialista elicitar as TPN para a estrutura $A \rightarrow B \leftarrow C$ ou $A \leftarrow B \rightarrow C$?

Para auxiliar na construção do DAG utilizando apenas especialistas, o mesmo precisa identificar as variáveis que podem causar uma variável a estar em um determinado estado, ou prevenir de estar no estado em questão. Em Korb e Nicholson *korb bayesian*, recomenda-se utilizar os seguintes critérios para definir as variáveis e suas causalidades: (i) identificar as causas, (ii) identificar os efeitos, (iii) definir variáveis que previnem o efeito (e.g., *PERGUNTA: que pode prevenir que HIV cause AIDS? RESPOSTA: Drogas anti-virais.*), e (iv) definir variáveis que interferem na relação causal (e.g., *PERGUNTA: Há algum fator que interfira no efeito da medicação para tratamento de doenças cardíacas? RESPOSTA: Sem uma dieta adequada, o medicamento não é efetivo.*). As variáveis podem ser classificadas em quatro tipos: (i) **Alvo**, consiste do valor que o usuário quer calcular (i.e., saídas); (ii) **Observação**, representa fontes de informação disponíveis ou observáveis (i.e., entradas); (iii) **Controlável**, variável que pode sofrer intervenção; e (iv) **Contexto**, representa condições do ambiente.

Em Neil et al. [146], um conjunto de expressões idiomáticas foi apresentado. Expressões idiomáticas são fragmentos que representam graficamente tipos genéricos de raciocínio com incerteza. Neste documento, apresenta-se apenas as expressões idiomáticas utilizados na pesquisa: causa-consequência, síntese e reconciliação. Na causa-consequência, modela-se o processo causal com consequências observáveis. Na síntese, combina-se vários nós em um com o propósito de melhor organizar o DAG ou modelar definições determinísticas ou incertas entre as variáveis. Na reconciliação, resultados de modelos diferentes de medição e predição são reconciliados. Exemplos dessas expressões idiomáticas são apresentados na

Figura 2.3.

Da mesma forma que para construir o DAG, para definir as TPN é possível utilizar soluções baseadas em dados, especialistas e híbridas. Um exemplo de solução baseada apenas em dados, chamada aprendizagem *batch*, é apresentada em Heckerman [90]. Outra solução popular é o algoritmo Expectation Maximization (algoritmo EM) [117], que é capaz de lidar com a falta de dados para algumas combinações dos nós-pai. O algoritmo EM pode ser utilizado para inferir as probabilidades de variáveis latentes por meio da definição de *clusters* [221]. Por outro lado, essas soluções dependem da disponibilidade de dados, que na prática é muitas vezes inviável.

Para os casos nos quais há uma quantidade pequena de dados, pode-se utilizar soluções híbridas, nas quais é utilizado o conhecimento de especialistas para complementar os dados. Em Zhou et al. [226], uma solução baseada na utilização de restrições, redes Bayesianas auxiliares e aprendizagem multinominal de parâmetros é apresentada. Outros exemplos de soluções híbridas são apresentados em Yet et al. [222], Zhou e Fenton [227], Niculescu et al. [148] e Khan et al. [109]. Outra solução nesse contexto é, ao invés de elicitado dados de especialistas, transferir o conhecimento de redes Bayesianas diferentes, mas relacionadas, como apresentado em Zhou et al. [228]. Por exemplo, essa solução viabiliza a transferência do conhecimento de um modelo de diagnóstico médicos com dados de um país para outro.

No caso de total indisponibilidade de dados, a única opção é elicitado o conhecimento de especialistas. Os riscos associados à elicitado são referentes à quantidade de informação necessária, que pode causar inconsistências nos dados coletados, e a complexidade da informação [207]. Nesse contexto, há estudos para guiar no processo de elicitado e remoção de viés, tais como O'Hagan et al. [149] e Santos Jr et al. [175]. Para auxiliar na elicitado das probabilidades pode-se utilizar escalas verbais [169, 95] e informações visuais [187, 100]. Por outro lado, essas técnicas são insuficientes no contexto de redes bayesianas de larga escala.

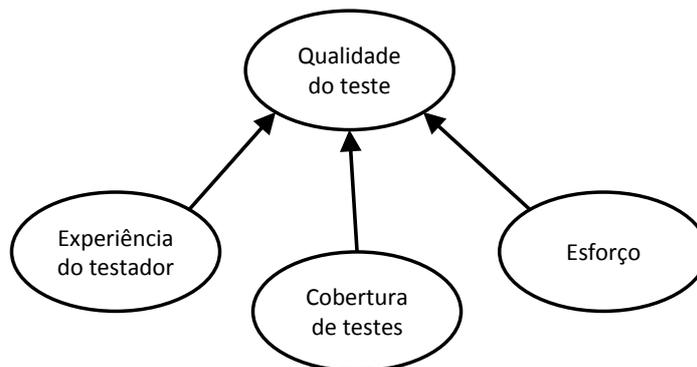
Uma alternativa é, ao invés de utilizar modelos discretos, utilizar distribuições contínuas e elicitado informações de dependências por meio de correlações ranqueadas condicionais [207]. Esses modelos são conhecidos como redes bayesianas não-paramétricas [84, 139].

Para o caso de modelos discretos, pode-se utilizar abordagens semiautomáticas para definição de TPN. No caso, dados para um conjunto de combinações dos nós-pai são eli-

Causa-consequência



Síntese



Reconciliação

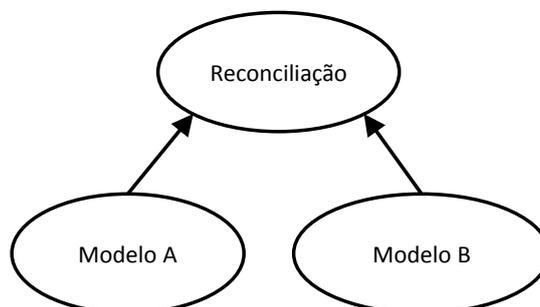


Figura 2.3: Exemplos de expressões idiomáticas.

citados e as demais calculadas. Dessa forma, a complexidade é reduzida de exponencial para linear [135]. A abordagem semiautomática mais popular é baseada na porta Noisy-OR [152, 98, 224]. O modelo Noisy-OR avalia o efeito na saída na presença de um fator por vez, com todos os outros ausentes, e calcula o valor das demais combinações. Modelos Noisy-OR clássicos são baseados em nós binários, mas algumas extensões foram propostas na literatura [218, 121]. Outra solução da mesma família é o método Noisy-MAX [53].

Em Mkrtchyan et al. [135], cinco abordagens semiautomáticas para definição de TPN foram comparadas: Røed et al. [172], Wisse et al. [217], Podofillini e Dang [163], Cain [35] and Fenton et al. [69]. Foi concluído que as soluções apresentadas em Røed et al. [172], Wisse et al. [217] e Cain [35] são limitadas para representar influências entre múltiplos fatores (e.g., $A \rightarrow B \leftarrow C$). Foi concluído que Fenton et al. [69] tem menor flexibilidade de modelagem de incerteza do que Podofillini e Dang [163]. Por outro lado, Podofillini e Dang [163] requer um esforço maior e só é recomendado utilizá-la para redes bayesianas de tamanho moderado. Uma limitação do trabalho, com relação à avaliação da solução apresentada em Fenton et al. [69], é que a mesma necessita, dado as elicitções de um conjunto de combinações de nós-pai, da definição de uma função ponderada. Para tal, é necessário ter conhecimento de como definir os parâmetros da função. De acordo com os dados apresentados em Mkrtchyan et al. [135] não é claro se a limitação da modelagem foi causada por limitação da abordagem em si ou de quem definiu os parâmetros. Em da Silva et al. [49], foi apresentada uma solução para esta limitação. Para tal, regras de produção foram utilizadas para representar o conhecimento de um especialista em nós ranqueados. Com esta solução, as TPNs de nós ranqueados podem ser definidas sem necessitar de conhecimento específico acerca da técnica; é apenas necessário preencher a tabela de verdade.

Outra limitação de Fenton et al. [69] é apenas lidar com escalas ordinais. Em Laitila e Virtanen [116], a solução foi expandida para lidar com escalas intervalares. Além disso, do ponto de vista operacional, outra limitação da abordagem é que a mesma é apenas fornecida no AgenaRisk. Em Bezerra et al. [25] e Bezerra et al. [24], foram apresentados algoritmos para implementar nós ranqueados baseando-se na biblioteca SMILE ¹, uma biblioteca de redes bayesianas livremente distribuída para pesquisas acadêmicas.

Em Das [50] foi apresentado o algoritmo da soma ponderada. Essa solução é baseada

¹<http://www.bayesfusion.com/>

em duas heurísticas: (i) quanto mais acessíveis cognitivamente um evento, maior a probabilidade do mesmo ocorrer, e (ii) mentalmente simular um cenário para avaliar a facilidade um resultado é produzido dado um conjunto de parâmetros e restrições operacionais. Para tal, os especialistas escolhem as combinações de pais que eles facilmente visualizam e simulam para ter como resultado probabilidades mais realistas. Dado isso e a definição dos pesos de cada nó pai, as probabilidades das demais combinações são calculadas. Essa solução foi avaliada por meio de um experimento em Baker e Mendes [12], onde foi comprovada sua eficiência e eficácia no contexto de previsão de estimativas em projetos Web.

Um ponto fraco em comum entre todas as abordagens semiautomáticas mencionadas é que elas necessitam que os especialistas definam as probabilidades para as combinações elicitadas que são suscetíveis à viés e podem resultar em inconsistências. Como alternativa, em Monti e Carenini [138] uma solução baseada em comparações pareadas foi proposta. A limitação da mesma é que só aplica-se para nós com apenas um pai. Em Chin et al. [38], outra solução baseada em comparações pareadas foi proposta com suporte para definição de TPN para nós com múltiplos pais.

Além disso, na literatura há propostas de processos para a construção de redes bayesianas baseadas em conhecimento e, possivelmente, dados. Em Mendes [131], o processo proposto é composto de três passos: construção da estrutura, quantificação da incerteza e validação do modelo. A estrutura pode ser construída por meio do conhecimento dos especialistas e literatura e avaliada por meio de dois estágios: (i) checar se fatores e categorias estão claros e bem nomeados e (ii) revisar se as relações causais estão em conformidade com os modelos mentais dos especialistas do domínio. No segundo estágio, o engenheiro de conhecimento pode auxiliar otimizando a estrutura para reduzir a quantidade de dados a serem elicitados, se mantendo neutro da perspectiva do especialista de domínio. Para quantificar a incerteza, pode-se utilizar o conhecimento dos especialistas, dados históricos, literatura ou uma combinação de todos. Para validação do modelo, executa-se duas atividades: (i) passo a passo do modelo e (ii) acurácia da previsão. O passo (i) trata-se de uma atividade subjetiva na qual os especialistas de domínio utilizam cenários hipotéticos para checar se os resultados calculados fazem sentido. No passo (ii), dados históricos de projetos passados são utilizados para confrontar os dados calculados do modelo. Esse processo foi aplicado para construir modelos preditivos de esforço em projetos Web [130] e para dar suporte à tomada de decisão

no contexto de Engenharia de Software Baseada em Valor [132].

Em Yet et al. [221] um processo para construção de redes Bayesianas no contexto de disponibilidade de poucos dados é proposto. Os passos do processo proposto são similares aos do processo apresentado em Mendes [131]. Por outro lado, destaca-se a utilização de algoritmos de clusterização e regras definidas por especialistas de domínio para a definição de TPN para variáveis latentes e a utilização de validação cruzada para validar o modelo. Além disso, ao final do processo, os resultados são avaliados pelos especialistas de domínio. Para validação da rede Bayesiana, recomenda-se avaliar a performance do modelo baseando-se na discriminação, calibração e acurácia. Para discriminação, recomenda-se *Característica de Operação do Receptor* (ROC) [85]. Para acurácia, Brier score [31] e Brier skill score [205]. Para calibração, o teste de Hosmer-Lemesbow [96]. Caso os resultados da validação não sejam satisfatórios, os especialistas podem concluir que as inconsistências foram causadas devido a erros do modelo ou dados ou escopo do modelo. Por exemplo, no contexto de um modelo para previsão de sobrevivência de pacientes de trauma, um paciente pode sobreviver devido a tratamentos pré-hospitalares, e o modelo poderia não identificar esse caso, pois intervenções pré-hospitalares não estão no escopo do modelo.

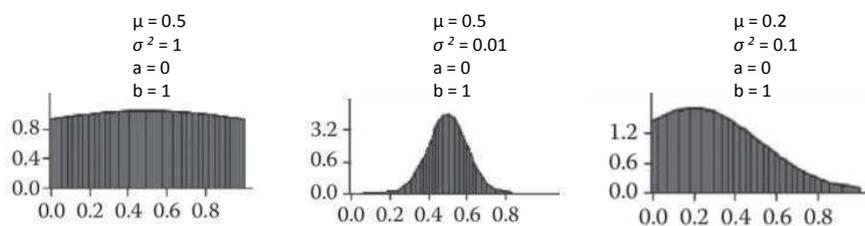


Figura 2.4: Exemplos de TNormal.

Em Constantinou et al. [46], um processo para construção de redes bayesianas a partir de dados coletados por meio de questionários e entrevistas é apresentado. O mesmo contém os mesmos passos presentes em Mendes [131] e Yet et al. [222], mas apresenta mais detalhes de como associar variáveis do modelo com fontes de dados (e.g., uma variável da rede Bayesiana pode estar associada a uma ou mais perguntas de um questionário). Além disso, apresenta instruções de como modelar intervenções na rede Bayesiana. Para validação da estrutura, recomenda-se utilizar análise de sensibilidade.

Nesta pesquisa, para todos os nós, inclusive para os latentes, o método apresentado em

Fenton et al. [69]. A solução é baseada em uma distribuição Normal truncada (TNormal) truncada na região $[0, 1]$. Esta distribuição é composta por 4 parâmetros: μ , média (i.e., tendência central); σ^2 , variância (i.e., confiança no resultado); a , limite inferior (neste caso, 0); e b , limite superior (neste caso, 1). Com esta distribuição, é possível modelar uma variedade de formatos (i.e., relacionamentos) tais como uma distribuição uniforme, quando $\sigma^2 = \infty$, e distribuições, quando $\sigma^2 = 0$. Exemplos de TNormal são apresentados na Figura 2.4.

Na solução apresentada em Fenton et al. [69], μ é definido como uma função ponderada dos nós pais. Há tipos de função: média ponderada (WMEAN), mínimo ponderado (WMIN), máximo ponderado (WMAX) e mistura de WMIN e WMAX (MIXMINMAX). De acordo com os autores, as mesmas são o suficiente para representar os relacionamentos necessários para definir TPN. Um exemplo de cada função é apresentado na Figura 2.5. Nos exemplos apresentados na Figura 2.5, WMEAN e MIXMINMAX, para o mesmo conjunto de entradas, calcula o mesmo valor. A diferença entre as duas funções é que WMEAN calcula a média ponderada dos valores dos nós pai (i.e., pesos são definidos para cada nó pai) e MIXMINMAX mistura as funções WMAX e WMIN (i.e., pesos são definidos para as funções).

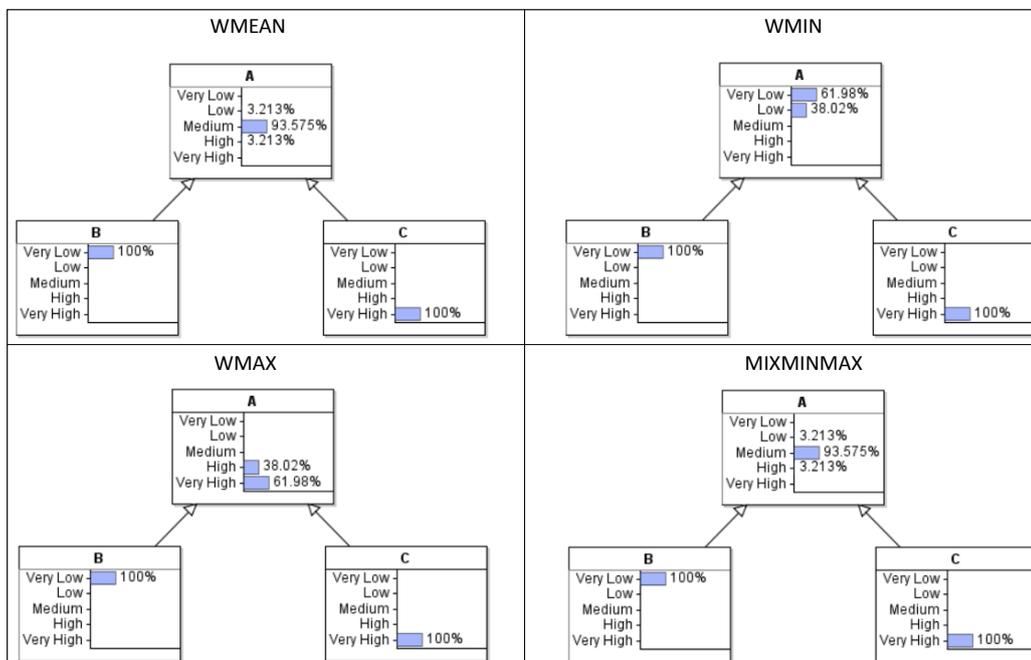


Figura 2.5: Exemplos de WMEAN, WMIN, WMAX e MIXMINMAX.

Para definir qual função utilizar, o desenvolvedor do modelo pode utilizar análise “e se”

com o especialista para coletar respostas para um conjunto de cenários e definir uma tabela da verdade. A tabela deve ser analisada para que a função, com os respectivos pesos, mais apropriada seja definida. A variância é definida empiricamente e deve refletir a confiança do especialista nos resultados.

Capítulo 3

Rede Bayesiana Proposta

Como mencionado anteriormente, para a instrumentação do Scrum, modelou-se o método de desenvolvimento representando suas principais entidades e a relação causal entre as mesmas considerando incertezas nas medições (i.e., observações) e relacionamentos.

Possíveis abordagens para modelar as relações causais são Modelagem de Equações Estruturais (SEM) [197] e redes Bayesianas [152]. SEM é uma técnica muito utilizada para quantificar e avaliar processos causais e já foi aplicada no contexto de ASD [122]. Por outro lado, seu objetivo primário é avaliar se uma rede teórica é uma representação aceitável do processo que gerou os dados em questão e realizar a previsão de variáveis latentes. Além disso, a mesma é incapaz de realizar diagnósticos. Por outro lado, redes Bayesianas, apesar de modelarem variáveis latentes [222, 221, 228], tratam todas as variáveis como observáveis (i.e., é possível definir evidência para qualquer variável). Dessa forma, ambas são capazes de realizar diagnóstico e prognóstico, sendo mais recomendada para dar suporte à tomada de decisão [9].

Com relação ao tratamento das incertezas, há diversas abordagens na literatura tais como filtros de Kalman [26], abordagens “set-membership” [23], redes neurais [88], lógica fuzzy [111], teoria de possibilidade [59], Dempster-Shafer [220] e redes Bayesianas. Filtros de Kalman e abordagens de “set-membership” dependem de uma representação do processo (i.e., sistema) como um modelo, geralmente no espaço de estados linear [201], mas assume-se que o conhecimento necessário para a construção de tal modelo quantitativo não está disponível para o contexto dessa pesquisa. Redes neurais, e outras abordagens baseadas em dados, por definição precisam de volume de dados e não são recomendadas para diagnóstico,

pois a incerteza é modelada de forma implícita e o modelo não é facilmente compreensível para humanos [112, 201].

Com relação à classificação da incerteza no contexto do problema em questão, o mesmo está sujeito à aleatoriedade (i.e., futuro incerto) e incompletude (i.e., informação insuficiente). De acordo com Verbert et al. [201], nesse contexto, Dempster-Shafer, teoria da possibilidade e redes Bayesianas são as possíveis alternativas válidas. Dempster-Shafer e teoria da possibilidade são mais eficazes para o tratamento de incompletude. Rede Bayesiana é mais adequada para raciocínio causal e tomada de decisão, modela com mais clareza as inferências e é mais adaptável. Dado que a problemática desta pesquisa tem foco no diagnóstico e prognóstico para apoiar a tomada de decisão de equipes ágeis, rede Bayesiana foi a técnica de modelagem selecionada.

Neste trabalho, o objetivo da rede Bayesiana é modelar o método de desenvolvimento Scrum para auxiliar no seu diagnóstico e prognóstico. Como premissa, assume-se que a equipe siga princípios ágeis, com equipes multidisciplinares continuamente mantendo o débito técnico baixo. Além disso, assume-se que o tamanho da equipe (i.e., esforço em homem/hora) seja compatível com a produtividade esperada do mesmo.

A mesma deve ser compatível com os indicadores (i.e., métricas) coletados pela equipe. Caso o usuário deseje um diagnóstico mais preciso, a rede Bayesiana deve permitir a adição de fragmentos mais detalhados de alguns processos, por exemplo, permitindo a adição de práticas de desenvolvimento e qualidade (e.g., refatoramento, programação em pares e Desenvolvimento Dirigido por Testes). Na Seção 3.3, o processo de complementação da rede Bayesiana, acompanhado de exemplos, é apresentado.

Como apresentado na Seção 1.4, quatro versões da rede Bayesiana foram desenvolvidas durante esta pesquisa. Este documento restringe-se a apresentar o resultado da última versão. Descrições detalhadas das versões passadas são apresentadas em Perkusich et al. [155], Perkusich et al. [159], Perkusich et al. [157], Freire et al. [185], Freire et al. [76] e Perkusich et al. [156].

Uma visão geral da mesma considerando apenas os principais artefatos, papéis e práticas do Scrum é apresentada na Figura 3.1, onde os retângulos são fragmentos (i.e., conjunto de nós) e as elipses são nós na rede Bayesiana. A rede Bayesiana é construída considerando a perspectiva do Scrum Master e deve ser utilizada para auxiliar nas Reuniões de Retrospectiva.

Por isso, nem o Scrum Master, papel do Scrum, e nem a Reunião de Retrospectiva, prática do Scrum, são representados na rede Bayesiana. Além disso, a Definição de Pronto [182] por ser uma prática guarda-chuva não é representada na rede Bayesiana. Ou seja, a mesma deve ser considerada na avaliação de todos os artefatos. Os fragmentos do DAG são apresentados nas Figuras 3.2, 3.3, 3.4, 3.5. O DAG completo é apresentado na Figura 3.6. Nessa figuras, os nós na cor verde são do tipo *indicador* [69] e os nós na cor laranja, são considerados nós de entrada.

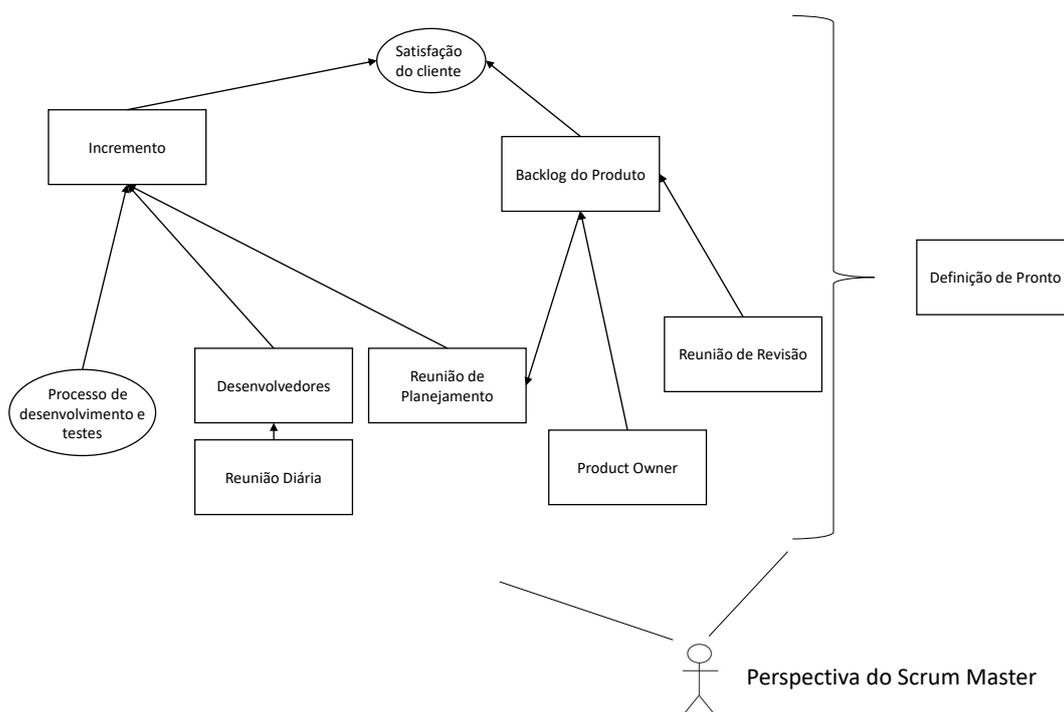


Figura 3.1: Visão geral da rede Bayesiana.

A rede Bayesiana contém dois tipos de fragmentos e nós: *frozen* e *hot*. A parte que modela artefatos, recursos e práticas oficiais do Scrum é a parte considerada *frozen* base. Ou seja, não deve ser modificada a não ser que existam justificativas plausíveis. Este fragmento é considerado a rede Bayesiana base. Por exemplo, não é recomendado retirar os nós que representam o Backlog do produto ou Reunião de Revisão.

Por outro lado, de acordo com o contexto do projeto, é possível complementar a rede Bayesiana base com fragmentos representando práticas ou processos. Além disso, é possível complementar a rede Bayesiana base com fragmentos representando métricas. Estes fragmentos são considerados *hot*. Ou seja, podem ser adaptados de acordo com o projeto. Por

exemplo, pode-se adicionar um nó para representar a prática *Backlog Grooming*, prática popular para refinamento do Backlog do produto, mas que não faz parte do conjunto de práticas oficiais do Scrum.

A rede Bayesiana foi construída seguindo o processo KEBN (ver Figura 1.3). O problema de construir uma rede Bayesiana pode ser subdividido em: (i) construção do DAG e (ii) definição das TPN. Detalhes relativos à construção do DAG são apresentados na Seção 3.1; à definição das TPN são apresentados na Seção 3.2. O processo para complementar a rede Bayesiana base com práticas e métricas é apresentado na Seção 3.3. Resultados de uma validação conceitual a partir de antipadrões na aplicação do Scrum apresentados em Eloranta et al. [60] são apresentados na Seção 3.4. Resultados de uma validação por meio de cenários simulados com especialistas de domínio são apresentados na Seção 3.5.

3.1 Construção do DAG da rede Bayesiana base

No DAG, cada nó representa um fator relevante do método de desenvolvimento Scrum identificado, e há um arco entre os nós sempre que eles se relacionarem. Nos arcos, o fator sendo influenciado é o apontado pela seta. Além disto, cada fator é composto por possíveis estados com probabilidades associadas. Desta forma, cada fator representa um conjunto de tuplas $N = \{(s_1, p_1), \dots, (s_{|N|}, p_{|N|})\}$, em que s_i é um possível estado e p_i é a probabilidade associada ao estado. O conjunto de fatores é representado por $F = \{N_1, \dots, N_{|F|}\}$. O conjunto de arcos é representado por $R = \{(N_j, N_k) \mid N_j \subset F \wedge N_k \subset F\}$, em que N_j é o ponto inicial e N_k é o ponto final do arco. Desta forma, o problema resume-se em identificar todos os elementos de F e R . Para identificar todos os elementos do conjunto F , necessita-se identificar todos os fatores N_a , e para cada fator N_a , identificar todos os possíveis estados s_i e suas probabilidades associadas p_i , em que $a \leq |F|$ e $i \leq |N_a|$. Finalmente, para identificar todos os elementos de R , necessita-se identificar todos f_j e f_k , em que f_j e $f_k \in F$.

Para definir o DAG, o problema foi dividido em dois: identificar os elementos de F e R , e identificar os elementos de N . O primeiro subproblema refere-se a identificar os nomes dos fatores para definir os nós do GAD e identificar os relacionamentos entre os fatores para conectar os nós. O segundo subproblema refere-se a identificar os possíveis estados de cada nó do GAD e as probabilidades associadas a cada estado.

Para resolver o segundo subproblema, foi utilizada uma escala Likert (i.e., ordinal) de 3 pontos para todos os nós. Uma escala de 3 pontos foi escolhida por sua simplicidade e por necessitar de menos dados para calibração e validação [45]. Além disso, como apresentado em Hubbard e Evans [99], utilizar uma escala ordinal com mais pontos não necessariamente aumenta a precisão dos dados coletados.

Desta forma, para cada fator (elemento de F) identificado, foram definidos três possíveis estados. Para o caso de métricas numéricas, limiares são definidos de acordo com o contexto do projeto para mapear a escala numérica para a ordinal. O mapeamento é de acordo com o conhecimento do especialista do projeto em questão ou dados históricos (mais detalhes na Seção 3.3).

Para resolver o primeiro subproblema, utilizou-se um processo incremental. As variáveis e seus relacionamentos foram definidas de acordo com o conhecimento de especialistas de domínio e dados da literatura. Para essa versão, três especialistas de domínio colaboraram. O especialista 1 tem 10 anos de experiência na indústria e a certificação de Project Management Professional (PMP), emitida pelo Project Management Institute (PMI). O especialista 2 tem 15 anos de experiência na indústria. O especialista 3 tem 5 anos de experiência na indústria. Durante o processo de construção do grafo, os especialistas discutiam cada fragmento por meio de grupos de foco. O papel do pesquisador é facilitar a discussão e garantir que cada decisão tomada seja embasada na literatura ou evidências de casos reais.

Independente do paradigma de gerenciamento de projeto (e.g., baseado em planos ou ágil), o principal objetivo de um projeto é satisfazer o cliente. Dessa forma, o primeiro elemento adicionado a F foi: *Satisfação do cliente*. Essa é a variável alvo, ou seja, a variável a ser prevista. Para construir o DAG, uma abordagem *top-down* foi utilizada. Com a mesma, o nó de mais alto nível foi decomposto utilizando o idioma síntese para construção de DAG [66].

A satisfação do cliente, do ponto de vista do processo de desenvolvimento, depende da qualidade da especificação e do produto entregue [67]. No caso do Scrum, a especificação do produto é documentada no Backlog do produto e a entrega é realizada por meio de incrementos. Desta forma, adicionam-se dois elementos a F : *Backlog do produto* e *Qualidade do Incremento*; todos pais (i.e. preditores) do nó *Satisfação do Cliente*.

3.1.1 Backlog do produto

O Backlog do produto é definido no início do projeto e refinado continuamente durante o mesmo atendendo às solicitações de mudanças do cliente. A importância de cada atividade (i.e., definição inicial e refinamento contínuo) depende da volatilidade dos requisitos do projeto. Apesar do Scrum, teoricamente, ser mais recomendado para o desenvolvimento de produtos complexos [193], onde são esperadas mudanças na especificação do produto durante o projeto, na prática, em alguns projetos o mesmo é utilizado mesmo quando a volatilidade dos requisitos é baixa. Por exemplo, foi relatado um caso por um dos especialistas de domínio que colaborou com a pesquisa no qual o projeto consistia em reimplementar um sistema utilizando uma tecnologia nova. Dessa forma, os requisitos de negócio estavam definidos a priori e, dessa perspectiva, não eram esperadas muitas mudanças. Apesar disso, foi decidido utilizar o Scrum, por ser parte da cultura da empresa. Ou seja, apesar de usar o Scrum, nesse caso, a necessidade de ter o Product Owner trabalhando continuamente e próximo à equipe era menor do que o comum.

Dessa forma, é necessário que o modelo leve em consideração o contexto do projeto, com relação à volatilidade dos seus requisitos, e, a partir disso, definir se a definição inicial ou o refinamento contínuo do Backlog do produto é mais importante. De acordo com Fenton et al. [65], *Complexidade dos requisitos* e *Inovação dos requisitos* são preditores para a dificuldade em implementar o projeto. Além disso, dado que em ASD mudanças são esperadas durante o projeto de acordo com a estabilidade do *Ambiente de negócio* (e.g., concorrentes e tecnologia), o mesmo, juntamente com *Complexidade dos requisitos* e *Inovação dos requisitos* foram adicionados a F como preditores de *Volatilidade dos requisitos*. Além disso, *Volatilidade dos requisitos*, *Definição inicial do Backlog do produto* e *Efetividade da definição contínua do Backlog do produto* foram adicionados a F como pais de *Backlog do produto*.

No Scrum, o Product Owner é o único responsável por gerenciar o escopo e todas as partes interessadas devem respeitar as suas decisões [193]. Sua missão é maximizar o valor do produto e do trabalho da equipe. Então, adiciona-se o elemento *Qualidade do Product Owner* a F como pai de *Efetividade da definição contínua do Backlog do produto*.

Além disso, durante o projeto, o Backlog do produto é atualizado como resultado da aprendizagem durante a *Reunião de Revisão* que deve ocorrer ao final de cada sprint. Du-

rante a Reunião de Revisão, o produto deve ser inspecionado e adaptado pela equipe e outras partes interessadas, se necessário. Além disso, o progresso do projeto com relação ao cronograma, orçamento e ambiente de mercado deve ser revisado. A não realização desta reunião ou a falta de participação dos clientes pode causar retrabalho devido à implementação de funcionalidades incorretas [60] que não adicionam valor de mercado ao produto. Desta forma, adiciona-se *Reunião de Revisão* a F como pai de *Efetividade da definição contínua do Backlog do produto*. Além disso, adicionam-se os nós *Inspeção*, *Adaptação* e *Monitoramento do Progresso* como indicadores (i.e., filhos) de *Reunião de Revisão*.

O Backlog do produto deve ser emergente, estimado, ordenado e detalhado corretamente [193, 194]. Os itens do topo do Backlog do produto devem estar com detalhamento o suficiente para que a equipe consiga implementá-los. Os itens de baixo não precisam estar bem detalhados, seguindo os princípios de *Just in time*. A ordenação do mesmo é essencial para maximizar a entrega de valor ao cliente e pode levar em consideração critérios como dependências técnicas, esforço, estimativa e valor de negócio [183]. Ter os requisitos estimados auxilia na tomada de decisão com relação ao escopo do projeto. Desta forma, adicionam-se 3 elementos a F : *Detalhado*, *Estimado* e *Ordenado*, todos indicadores do nó *Backlog do produto*. O fragmento *Backlog do produto* é apresentado na Figura 3.2.

3.1.2 Reunião de Planejamento

O escopo de cada sprint é definido no seu início durante a Reunião de Planejamento. Durante a mesma, itens do Backlog do produto são selecionados e alocados para a sprint em questão, levando em consideração fatores como a capacidade projetada da equipe e o desempenho passado. Como resultado dessa reunião, o Backlog da sprint e o Objetivo da sprint são definidos. O Backlog da sprint trata-se de uma previsão da equipe de quais funcionalidades estarão prontas no próximo incremento do produto entregue e é composto de itens de Backlog do produto e um plano de execução dos mesmos (e.g., cada item é decomposto em tarefas técnicas). O Objetivo da sprint representa o objetivo que a equipe, do ponto de vista estratégico, alcançará por meio da implementação do Backlog da sprint e guia no porquê o incremento está sendo desenvolvido [193]. O mesmo deve ser amplo e realista [161]. O fragmento resultante da Reunião de Planejamento é apresentado na Figura 3.3.

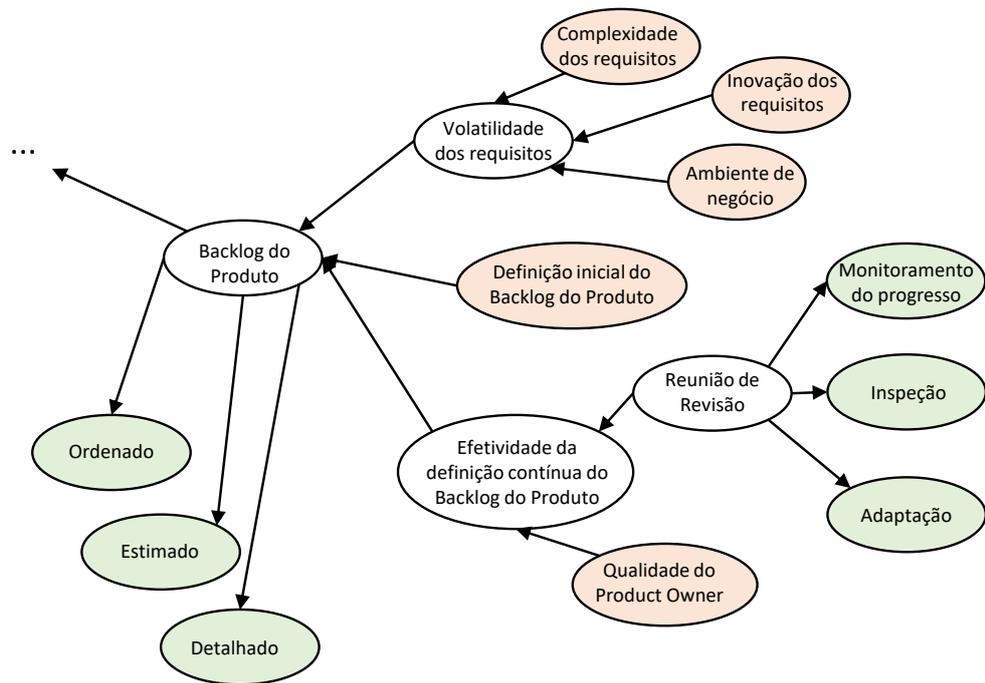


Figura 3.2: Fragmento *Backlog do produto*.

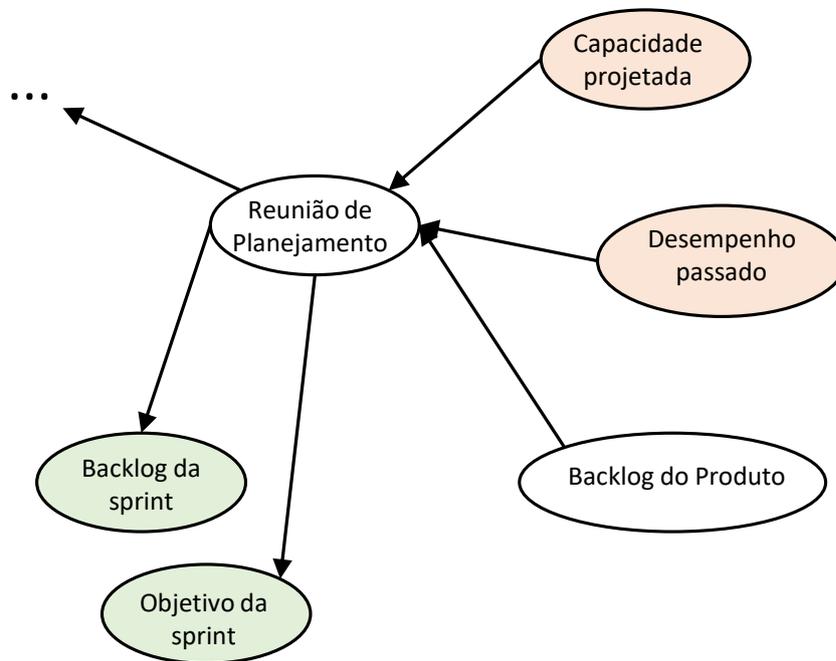


Figura 3.3: Fragmento *Reunião de Planejamento*.

3.1.3 Qualidade do incremento

O desenvolvimento do incremento do produto ocorre durante as sprints. O resultado de uma sprint é resultado da qualidade da execução da mesma. Seguindo o mesmo raciocínio dos modelos apresentados em Fenton et al. [65] e Fenton et al. [67], pode-se concluir que a qualidade da execução depende da Reunião de Planejamento, qualidade da equipe e da qualidade do processo de desenvolvimento e testes. Além disso, o resultado da execução pode ser mensurado em duas dimensões: eficiência e eficácia. Nesse caso, eficiência é a quantidade de funcionalidades entregues (i.e., novos itens de Backlog entregues). A eficácia é a qualidade do incremento em si. De acordo com a ISO 9000, qualidade é “a totalidade das características em satisfazer às necessidades declaradas ou implícitas”. Dessa forma, para, ao final da sprint, o incremento ter uma qualidade satisfatória, o mesmo deve atender aos objetivos definidos na Reunião de Planejamento. Dessa forma, assim como em Fenton e Neil [64], foi considerado que a qualidade do incremento é consequência da qualidade da execução da sprint e da quantidade de funcionalidades entregues. O fragmento resultante da *Qualidade do incremento* é apresentado na Figura 3.4.

3.1.4 Qualidade do trabalho em equipe

Em projetos ágeis, a qualidade do trabalho em equipe é um fator muito importante para melhorar o desempenho do projeto [123]. Para avaliar o trabalho em equipe, um dos principais fatores é a coesão da mesma [123, 137, 171, 83]. De acordo com Mullen et al. [141], a coesão da equipe refere-se à atração interpessoal dos membros da equipe, o comprometimento dos mesmos com as atividades da equipe e o espírito de equipe [93, 206, 177]. Por outro lado, apesar de coesa, o desempenho da equipe pode ser prejudicado caso a equipe sofra interrupções durante a sprint [137, 171, 83], que é fortemente não recomendado no Scrum [193]. Ou seja, a equipe precisa ter autonomia. De acordo com Moe et al. [137], trata-se da habilidade da equipe de controlar as condições de limite, as quais se referem à influência externa que a gestão ou outros indivíduos tem na equipe. Desta forma, adicionam-se 2 elementos a F : *Coesão* e *Autonomia*; ambos pais do nó *Trabalho em equipe*.

Para serem coesas, equipes Scrum devem ser autogerenciadas (i.e., auto-organizáveis) [193]. Equipes autogerenciadas escolhem qual a melhor forma para completarem seu tra-

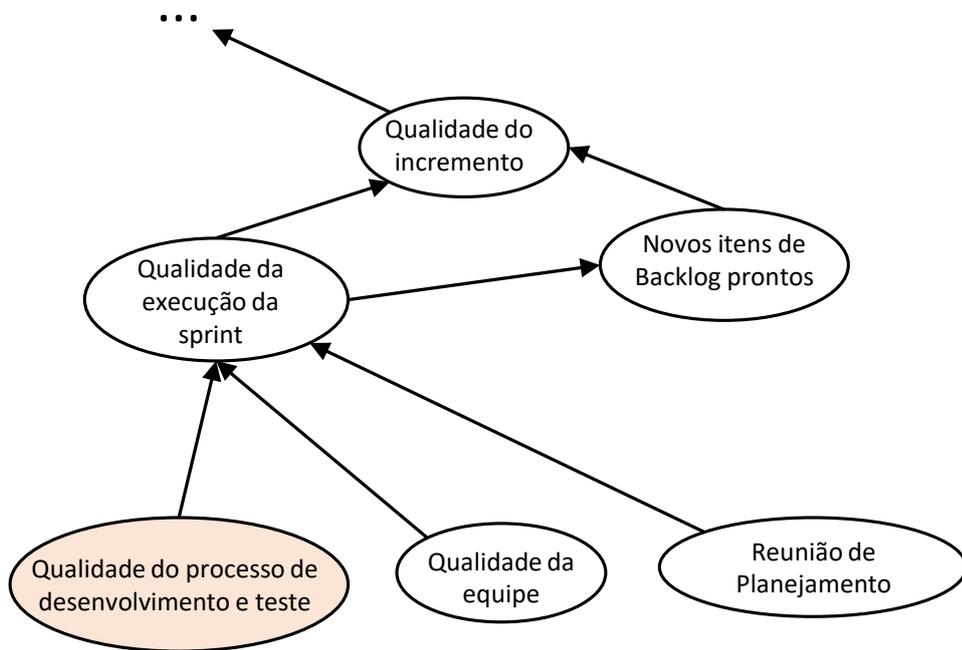


Figura 3.4: Fragmento *Qualidade do incremento*.

balho, em vez de serem dirigidos por outros de fora da equipe. Além disso, é necessário que os membros da equipe colaborem seguindo o conceito de responsabilidade da equipe inteira e se comprometam [93, 206, 41, 105]. Desta forma, adicionam-se 2 elementos a *F*: *Autogerenciamento* e *Colaboração*; ambos pais do nó *Coesão*.

Equipes autogerenciadas devem ser adaptáveis, ter liderança compartilhada e redundância [137, 82, 177, 86, 171, 83, 136]. De acordo com Moe et al. [137], a adaptabilidade da equipe é sua habilidade de identificar mudanças no ambiente e ajustar as estratégias. Liderança compartilhada trata-se de compartilhar a autoridade para tomar decisões e liderança. Redundância trata-se da capacidade dos membros da equipe executarem atividades dos outros e substituí-los, se necessário. No Scrum, a equipe deve ser multifuncional [193]. Equipes multifuncionais possuem todas as competências necessárias para completar o trabalho sem depender de outros que não fazem parte da equipe. Redundância e multifuncionalidade da equipe referem-se às características necessárias do conhecimento da equipe. Desta forma, adicionam-se 3 elementos a *F*: *Adaptabilidade*, *Liderança compartilhada* e *Conhecimento*; todos pais do nó *Auto-gerenciamento*.

Para uma equipe colaborar, a mesma deve ser coordenada [93, 136, 206, 177, 83]. A coordenação da equipe refere-se aos membros da equipe executarem suas atividades de forma sincronizada e integrada [93]. Além disso, ela deve ser orientada [137, 82, 113, 86, 171, 83, 136]. A orientação trata-se do respeito mútuo entre os membros das equipes e priorização dos objetivos da equipe sobre os individuais [173]. Desta forma, adicionam-se 2 elementos a *F*: *Coordenação* e *Orientação*; ambos pais do nó *Colaboração*.

Para uma equipe coordenar suas atividades, é necessário se comunicar. Em equipes ágeis, a comunicação é fundamental, pois o compartilhamento de conhecimento é tácito [29, 93, 136, 206, 113, 177, 41, 83, 105]. No Scrum, há uma prática auxiliar na coordenação da equipe: Reunião Diária. Trata-se de um evento realizado diariamente e limitado em 15 minutos para a equipe sincronizar as atividades, identificar e remover impedimentos, promover tomada rápida de decisão e melhorar o nível de conhecimento [193]. Desta forma, adicionam-se 2 elementos a *F*: *Comunicação* e *Reunião Diária*; ambos pais do nó *Coordenação*.

A qualidade da comunicação da equipe depende dos canais de comunicação e distribuição da equipe. De acordo com Bustamante et al. [34], idealmente, equipes ágeis são

colocalizadas e usam comunicação face-a-face diariamente. Os canais de comunicação são associados ao número de integrantes e os tipos de comunicação (e.g., face-a-face, Voice over Internet Protocol e mensageiros instantâneos). A utilização de tecnologias para comunicação síncrona remota (e.g., mensageiros instantâneos) pode compensar os problemas de comunicação causados pela distribuição da equipe [216]. Desta forma, adicionam-se 2 elementos a *F*: *Canais e Distribuição*; ambos pais do nó *Comunicação*.

Durante as Reuniões Diárias, cada integrante deve responder 3 perguntas: (i) “O que eu fiz ontem que ajudou a equipe a atender o objetivo da sprint?” (ii) “O que eu farei hoje para ajudar a equipe a atender o objetivo da sprint?” (iii) “Eu vejo algum obstáculo que impeça a mim ou a equipe no atendimento do objetivo da sprint?” [193]. De acordo com Stray et al. [191], os tópicos da reunião devem se manter restritos apenas a essas 3 perguntas, com foco maior nas duas últimas. Além disso, deve ser observado se há autojustificação, pois o mesmo é um indicador de escalada de compromisso [192]. Discussões técnicas podem ocorrer, desde que logo após a realização da reunião [191]. Nesse caso, *Monitoramento e Presença* foram adicionados a *F* como indicadores do nó *Reunião Diária*. O fragmento resultante da *Qualidade do incremento* é apresentado na Figura 3.5. O DAG completo é apresentado na Figura 3.6, onde os fragmentos cada fragmento está marcado com uma cor: *Qualidade do incremento*, verde; *Qualidade do trabalho em equipe*, cinza; *Reunião de Planejamento*, amarelo; e, *Backlog do produto*, azul.

3.2 Definição das TPN

Inicialmente, as TPN foram calibradas de acordo com o conhecimento do pesquisador. Posteriormente, elas foram calibradas com o conhecimento dos especialistas com ajustes decorrentes do processo de validação, apresentado na Seção 3.5. Além disso, as mesmas foram calibradas usando as funcionalidades de nós ranqueados da ferramenta AgenaRisk¹. Caso outro algoritmo de nós ranqueados seja utilizado (e.g., [24]), é possível que as mesmas necessitem ser recalibradas devido à possibilidade de utilização de diferentes algoritmos para gerar distribuições normais truncadas.

Para os nós de entrada, foi definida uma distribuição uniforme. Dado que todos os nós

¹www.agenarisk.com

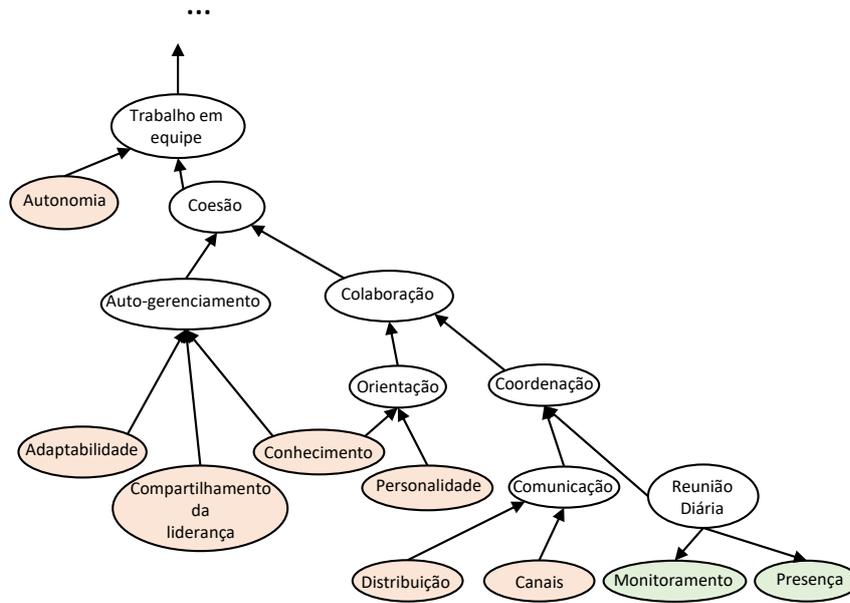


Figura 3.5: Fragmento *Qualidade do trabalho em equipe*.

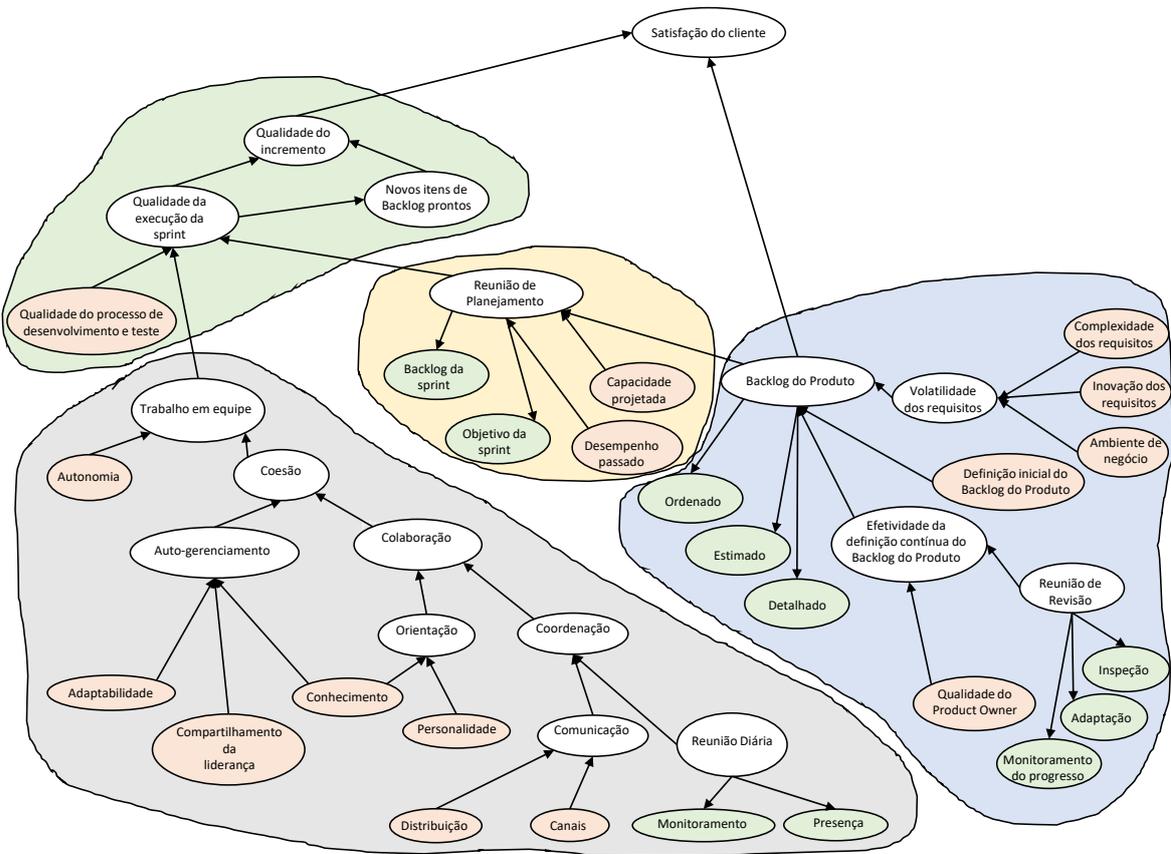


Figura 3.6: DAG completo.

Tabela 3.1: Variância definida para os nós indicadores.

Indicador	Variância
Inspeção	0,003
Adaptação	0,001
Monitoramento do progresso	0,006
Ordenado	0,01
Estimado	0,005
Detalhado	0,0015
Objetivo da sprint	0,004
Backlog da sprint	0,002
Monitoramento	0,0015
Presença	0,01

são compostos de 3 estados, cada estado foi definido com a probabilidade de 1/3. Para os nós indicadores, dado que os mesmos são ranqueados [69], é necessário calibrar a média e a variância da distribuição normal truncada entre os pontos 0 e 1. Para todos os casos, a média foi definida como o nó pai. A variância, que representa o grau de credibilidade do indicador em representar o atributo em questão [69], foi definida por meio de tentativa e erro, como recomendado por Fenton et al. [69] no caso da ausência de dados históricos. Para tal, a menor variância possível no AgenaRisk é inicialmente definida para o indicador em questão (i.e., 0,0005) e a distribuição gerada é analisada pelo usuário. Caso a distribuição não o satisfaça, a variância é aumentada (e.g., de 0,0005 para 0,001) até que o usuário esteja de acordo com a distribuição gerada. Na Tabela 3.1, a variância definida para cada indicador é apresentada.

Para os demais nós, com exceção de *Backlog do produto*, *Qualidade do incremento* e *Novos itens de Backlog prontos*, foi utilizado o método de nós ranqueados [69]. Nesse caso, a média pode ser representada por uma expressão ponderada, representando o tipo e magnitude de influência que os preditores tem no nó filho. Para definir a expressão ponderada, cenários de testes foram definidos para elicitare as probabilidades. Para definir a variância, da mesma forma que com os nós indicadores, foi seguido um processo de tentativa e erro. Na Tabela 3.2, o tipo de função ponderada, pesos dos pais e a variância definida são apresentadas para

cada nó no qual foi utilizado o método de nós ranqueados para definir suas TPN.

Para o nó *Novos itens de Backlog prontos* (NIBP), dado que o mesmo só tem um pai, foi utilizado o mesmo procedimento dos nós indicadores. A variância definida foi 0,01. Para os nós *Backlog do produto*, *Qualidade do incremento* foram utilizadas expressões particionadas. Nesse caso, define-se n TPN para um nó filho, onde n é o número de estados que um dos pais tem. No caso, esse método é utilizado quando um dos nós pai condiciona as probabilidades esperadas para o nó filho de acordo com as combinações dos demais nós pai. Por exemplo, *Backlog do produto* é filho de *Volatilidade dos requisitos* (VR), *Definição inicial do Backlog do produto* (DIBO) e *Efetividade da definição contínua do Backlog do produto* (EDCBP). A DIBO tem impacto maior em *Backlog do produto* quando a VR é baixa. A EDCBP tem impacto maior quando a VR é baixa. Dessa forma, pode-se concluir que o impacto de DIBP e EDCBP em *Backlog do produto* é condicionado (i.e., particionado) em VR. Dessa forma, três TPN foram definidas para *Backlog do produto* e são apresentadas na Tabela 3.3.

De forma similar, a influência de NIBP em *Qualidade do incremento* é condicionada em *Qualidade da execução da sprint* (QES). Por outro lado, ao invés de utilizar as funções apresentadas em Fenton et al. [69] para definir a média da TPN, multiplicadores foram utilizados, pois cada TPN foi definida em função de um nó pai. Para $QES = Baixo$, foi definido o multiplicador 0,5; para $QES = Moderado$, 1; e, para $QES = Alto$, 2. Para todos os casos, a variância definida foi 0,01.

3.3 Complementação da rede Bayesiana base

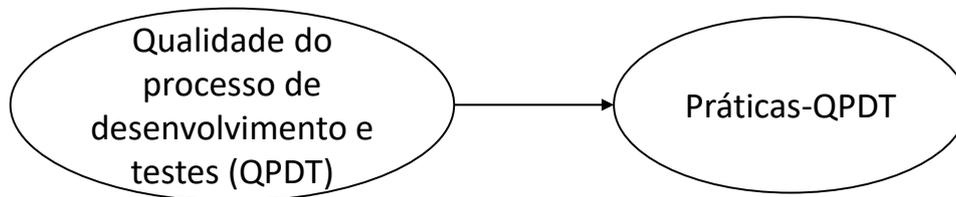
Dado que o Scrum deve ser complementado por práticas e processos, a rede Bayesiana base pode ser adaptada para se adequar às necessidades do projeto em questão. O objetivo da rede Bayesiana é fornecer informações acerca do método de desenvolvimento. Desta forma, cada fator na rede Bayesiana base pode ser mapeado para um conjunto de práticas. Este conjunto de práticas é representado por um nó auxiliar na rede Bayesiana. Este nó é uma cópia do fator em questão. Por exemplo, pode-se mapear o nó *Qualidade do processo de desenvolvimento e testes* para outro nó *Práticas-Código*. Desta forma, as evidências em *Qualidade do processo de desenvolvimento e testes* servirão como preditores para o processo de desenvolvimento e testes do incremento. Na Figura 3.7, um exemplo de mapeamento é apresentado.

Tabela 3.2: Calibração para os nós calibrados como método de nós ranqueados [69].

Nó	Função	Pesos	Variância
Satisfação do cliente	WMEAN	1 ->Backlog do produto 3 ->Qualidade do incremento	0,01
Qualidade da execução da sprint	WMEAN	1,5 ->Qualidade da execução do processo de desenvolvimento e testes 5 ->Trabalho em equipe 1 ->Reunião de Planejamento	0,0005
Trabalho em equipe	WMEAN	5 ->Coesão 1 ->Autonomia	0,001
Coesão	WMIN	3 ->Autogerenciamento 3 ->Colaboração	0,0005
Autogerenciamento	WMIN	1 ->Adaptabilidade 1 ->Compartilhamento da liderança 3 ->Conhecimento	0,0005
Colaboração	WMIN	5 ->Orientação 5 ->Coordenação	0,0005
Orientação	WMIN	5 ->Conhecimento 5 ->Personalidade	0,0005
Coordenação	WMEAN	1 ->Comunicação 1 ->Reuniões diárias	0,0005
Comunicação	WMIN	5 ->Canais 3 ->Distribuição	0,0005
Planejamento da sprint	WMEAN	1 ->Desempenho passado 1 ->Capacidade projetada 3 ->Backlog do produto	0,001
Volatilidade dos requisitos	WMEAN	1 ->Complexidade dos requisitos 1 ->Inovação dos requisitos 1 ->Ambiente de negócio	0,001
Efetividade da definição contínua do Backlog do produto	WMEAN	3 ->Qualidade do PO 1 ->Reunião de revisão	0,005

Tabela 3.3: Calibração para o nó *Backlog do produto*.

Nó	Nó condicional	Estado	Função	Pesos	Variância
Backlog do produto	VR	Baixo	WMEAN	3->DIBP 1->EDCBP	0,001
		Moderado	WMEAN	1->DIBP 1->EDCBP	0,001
		Alto	WMEAN	1->DIBP 5->EDCBP	0,001



Práticas-QPDT/QPDT	Baixo	Moderado	Alto
Baixo	1	0	0
Moderado	0	1	0
Alto	0	0	1

Figura 3.7: Exemplo de mapeamento da rede Bayesiana base para processo ou prática complementar.

Apesar da simplicidade, este procedimento auxilia a equipe na análise da rede Bayesiana durante a Reunião de Retrospectiva. De acordo com evidências coletadas na sprint, indica-se quais práticas ou processos precisam ser analisados para aumentar a probabilidade de construir um produto que satisfaça os clientes. Por outro lado, pode-se utilizar técnicas de construção de rede Bayesiana para decompor o nó auxiliar em um modelo causal representando o processo em questão utilizando a estrutura Estrela Convergente [167]. Assim como na rede Bayesiana base, os nós do fragmento devem ser ranqueados. Apesar de aumentar o custo para construção de rede Bayesiana, pode-se melhorar o diagnóstico. Cabe ao Scrum Master do projeto em questão julgar o custo-benefício de como complementar a rede Bayesiana com as práticas ou processos.

Por exemplo, definir o plano de lançamento e visão do produto é uma boa prática para planejar projetos ágeis [161] e pode influenciar a qualidade do Backlog do produto. O plano de lançamento deve ser composto de data de lançamento, descrição das funcionalidades principais e a visão do produto [161]. A data de lançamento, preferencialmente, deve estar numa escala de meses. A visão do produto deve conter um objetivo amplo e engajante, ser claro e estável, curto e conciso e descrever os atributos críticos para satisfazer as necessidades do usuário. Desta forma, é possível definir um modelo causal adicionando os seguintes nós neste fragmento: *Data de lançamento*, *Descrição das funcionalidades principais* e *Visão do produto*; todos pais do nó *Plano do lançamento*; e *Atributos críticos*, *Objetivo amplo e engajante*, *Curto e conciso* e *Claro e estável*; todos pais do nó *Visão do produto*. Além disto, o *Plano do lançamento* deve ser conectado com o nó *Definição inicial do Backlog do produto*. As TPN do fragmento resultante devem ser calibradas com o método proposto em Fenton et al. [69]. O mesmo é apresentado na Figura 3.8. Na Tabela 3.4, é apresentada uma lista de práticas associadas a nós da rede Bayesiana base.

Um processo similar pode ser utilizado para detalhar melhor os recursos do projeto. No caso do nó *Qualidade do Product Owner*, o mesmo pode ser decomposto em características desejáveis no Product Owner. De acordo com Pichler [161], o Product Owner deve ter habilidades colaborativas, ser empoderado e comprometido e visionário e executor. As habilidades colaborativas referem-se a ser comunicador e negociador, líder e jogador de equipe e disponível e qualificado. O Scrum Master deve observar estas características do Product Owner por meio de suas ações no dia-a-dia e avaliá-las utilizando uma escala Likert de 5

Tabela 3.4: Exemplos de associações de práticas a fatores da rede Bayesiana.

Nó	Práticas
Qualidade do processo de desenvolvimento e testes	Refatoramento, programação em pares, TDD, integração contínua e automação dos testes.
Estimado	<i>Planning Poker</i> [87], <i>big wall</i> , <i>spike</i> [43, 214], <i>story point</i> [43], tempo ideal [43], <i>T-shirt size</i> [43] e redes neurais [4, 198].
Backlog do produto	Reunião de <i>Grooming</i> [140], Plano de lançamento [161] e visão do produto [161].
Ordenado	Modelo <i>Kano</i> [43], Modelo <i>Wieger</i> [194], <i>OR value</i> [194], <i>Innovation games</i> [94], Retorno de Investimento (ROI) e MoSCoW [164, 6].
Detalhamento	<i>User story</i> [140] e Caso de uso.
Reunião de Planejamento	Ritmo sustentável [214], Estabilização [214], Dirigido a velocidade [43].
Backlog da sprint	<i>Interaction Room</i> [81].
Monitoramento do progresso	<i>Burndown</i> e <i>Burnup</i> [193].

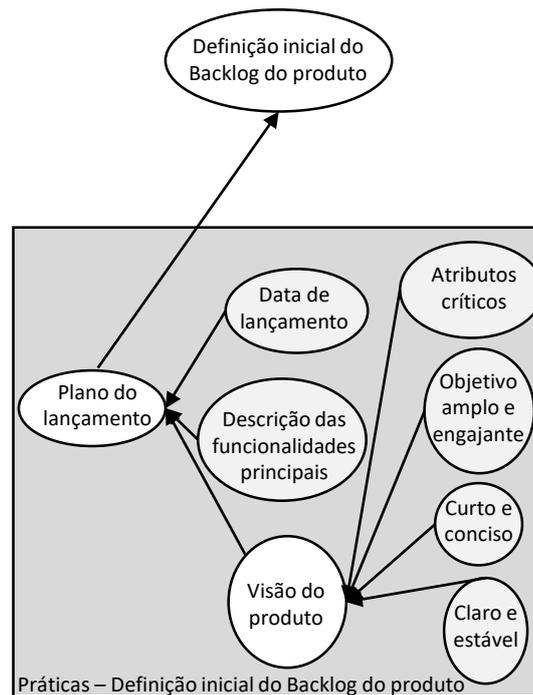


Figura 3.8: Exemplo de decomposição de nó auxiliar.

pontos. Um Product Owner com personalidade e empoderamento inadequados pode causar requisitos confusos e problemas de comunicação com a equipe e clientes. Desta forma, pode-se definir o fragmento apresentado na Figura 3.9.

Além disso, é possível complementar a rede Bayesiana base com métricas. Métricas podem ser utilizadas com dois propósitos: (i) como indicador do valor de um nó de entrada e (ii) alerta. Para o primeiro caso, é necessário conectar as métricas ao nó em questão e calibrar a TPN. Se a métrica for coletada automaticamente (i.e., por meio de ferramentas), o custo de aplicação do processo é reduzido. Por exemplo, para indicar se o Backlog do produto está ordenado corretamente, pode-se avaliar se foram consideradas as dependências técnicas, riscos e valor de negócio [161]. Neste caso, para cada fator, deve ser criado um nó na rede Bayesiana e conectá-lo ao nó *Ordenado* como um nó indicador, como apresentado na Figura 3.10. Caso os nós indicadores sejam ordinais, os mesmos devem ter a mesma quantidade de estados que o nó da entidade em questão, como recomendado em Fenton et al. [69].

Métricas com escalas quantitativas (contínuas ou discretas) também podem ser utilizadas. Nesse caso, expressões particionadas são utilizadas para, para cada estado da enti-

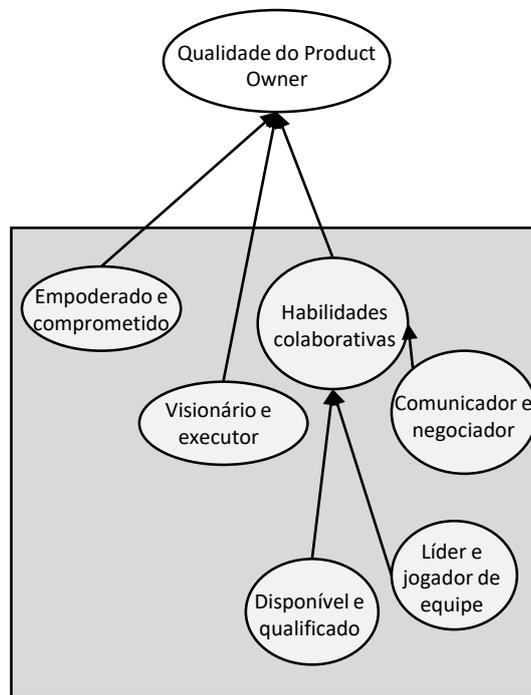


Figura 3.9: Decomposição do nó Product Owner.

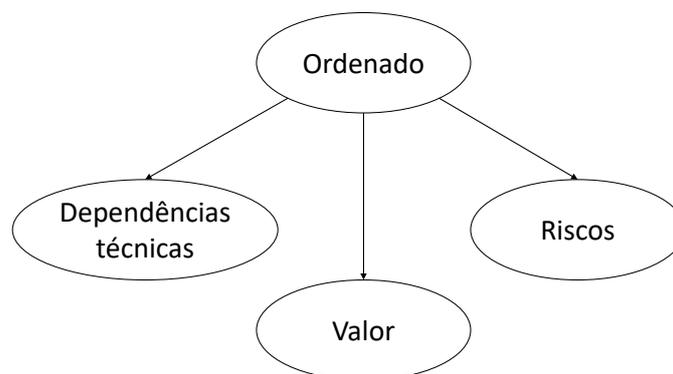


Figura 3.10: Exemplo de adição de métricas na rede Bayesiana base.

dade em questão, definir uma TPN para o nó indicador. No AgenaRisk, há diversos tipos de distribuições de probabilidade tais como Normal, Binomial e Normal Truncada. Por exemplo, pode-se definir a métrica *Defeitos por linhas de código* como indicador para o nó *Qualidade do incremento* usando uma distribuição Normal truncada entre os pontos 0 e 100. Os parâmetros seriam: para *QualidadeDoIncremento = Baixo*, $\mu = 35$ e $\sigma^2 = 30$; para *QualidadeDoIncremento = Moderado*, $\mu = 4$ e $\sigma^2 = 5$, e para *QualidadeDoIncremento = Alto*, $\mu = 0,2$ e $\sigma^2 = 0,5$. Os parâmetros devem ser definidos de acordo com conhecimento de especialistas ou dados históricos. Na Figura 3.11, um cenário calculado para o exemplo em questão é apresentado, onde o valor do indicador foi calculado pelo modelo (e.g., predição). Nesse caso, foi previsto que o valor de *Defeitos por linhas de código* foi, aproximadamente, 28 (i.e., a média da distribuição). Por outro lado, evidências podem ser definidas para o nó indicador, como apresentado na Figura 3.12 (no caso, a evidência definida foi 23).

Como discutido na Seção 2.4, em redes Bayesianas, a premissa de independência entre os nós é válida apenas para os pais de nós, desde que não tenham relação direta entre eles. Além disso, os indicadores não têm efeito no modelo caso não sejam definidas evidências para os mesmos, mas podem ter seus valores previstos pelo modelo. Caso um indicador (sem evidência) seja influenciado por outro (com evidência), esses resultados devem ser ignorados, pois a dependência não necessariamente é real. Uma alternativa é, ao invés de usar a estrutura de Classificador Nãive Bayes para os indicadores, utilizar a estrutura Estrela Convergente, assim como é feito para a adição de práticas. Por outro lado, caso a entidade em questão já tenha preditores (i.e., outros pais), a definição da TPN do mesmo ficará confusa.

Para o segundo caso, a métrica serve como um alerta indicando que pode haver algum problema no modelo. Além disso, é possível identificar possíveis problemas no método de desenvolvimento que a rede Bayesiana falhou em identificar, ou por problema de construção ou utilização ou limitação. Neste caso, a métrica deve ser conectada à rede Bayesiana por meio da expressão idiomática de reconciliação [146].

Por exemplo, pode-se associar a métrica *Número de defeitos deslizados* [114] ao nó *Inspeção*. Esta métrica indica o número de defeitos que deveriam ter sido descobertos em uma sprint anterior, mas não foram. Suponha que, em determinado projeto, as inspeções em todas as Reuniões de Revisão foram satisfatórias de acordo com o Scrum Master. Por outro lado,



Figura 3.11: Exemplo de previsão de indicador quantitativo. Imagem extraída do AgenaRisk.

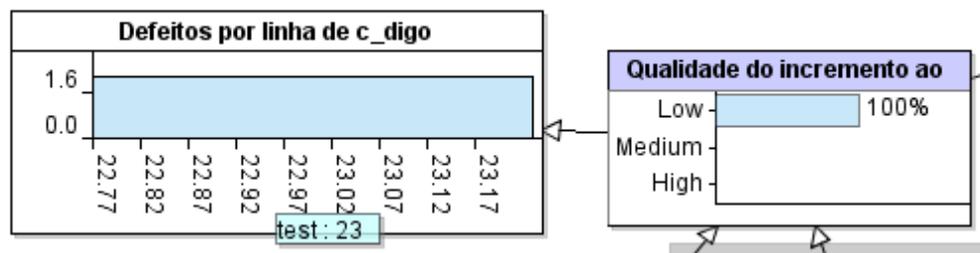


Figura 3.12: Exemplo de definição de evidência no indicador quantitativo. Imagem extraída do AgenaRisk.

de acordo com a métrica em questão, foi gerado um alerta de que há algum problema no nó *Inspeção*. Este alerta pode significar que, na Reunião de Revisão, incrementos foram aprovados mesmo contendo defeitos. Ou seja, pode ser um indicador de que a inspeção durante a reunião não está boa o suficiente. Como resultado, pode-se concluir que, realmente, houve problema na Reunião de Revisão. Por outro lado, é possível que o problema tenha sido na qualidade do teste. Neste caso, a rede Bayesiana seria atualizada e *Número de defeitos deslizados* seria conectado também ao nó *Qualidade do processo de desenvolvimento e testes*. Finalmente, o alerta pode ser falso e ser ignorado. Na Figura 3.13, este exemplo é apresentado. Na Tabela 3.5, é apresentada uma lista de métricas associadas a nós da rede Bayesiana base.

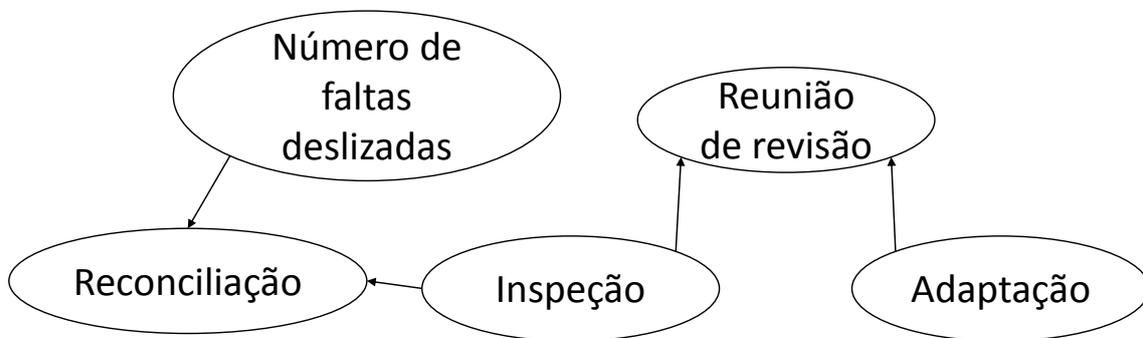


Figura 3.13: Exemplo de adição de métrica na rede Bayesiana base.

3.4 Validação conceitual

Para validar a rede Bayesiana do ponto de vista conceitual, os antipadrões do Scrum apresentados em Eloranta et. [60] foram avaliados. Trata-se de um estudo empírico no qual foram coletados dados de 18 equipes em 11 empresas. Como resultado, foram apresentados 14 antipadrões ao adotar o Scrum e suas possíveis consequências. O objetivo da validação é identificar se, caso a rede Bayesiana seja utilizada, os antipadrões sejam detectados. Além disso, analisa-se se os cálculos da rede Bayesiana estão em conformidade com Eloranta et al. [60] com relação às consequências dos antipadrões. Para tal, assume-se que as evidências são

Tabela 3.5: Exemplos de métricas que podem ser associadas com nós da rede Bayesiana base.

Nó	Métricas
Qualidade do processo de desenvolvimento e testes	Violações de análise estática, débito técnico [56], número de casos de teste, cobertura de código e situação da <i>build</i> [56].
Backlog do produto	Porcentagem de itens preparados.
Reunião de Planejamento	Porcentagem de itens completados.
Detalhado	INVEST [57].
Novos de itens de Backlog prontos	<i>Running Tested Features</i> [1], velocidade [43], pontos de função, linhas de código entregues, quantidade de telas (GUI).
Satisfação do cliente	valor de negócio entregue [1], Índice de Satisfação do Cliente.

alimentadas no modelo corretamente. Os antipadrões e suas consequências são apresentados na Tabela 3.6.

Para o antipadrão *Documento de requisitos extenso*, o nó em questão na rede Bayesiana é *Backlog do produto*, que não é um nó de entrada. Desta forma, analisam-se as consequências do mesmo para definir evidências para os nós: *Detalhado*, *Estimado*, *Emergente* e *Ordenado*. Como apresentado na Tabela 3.5, o Backlog do produto não é ordenado nem emergente. Além disso, dado que a equipe não entende os requisitos, isso significa que o detalhamento dos mesmos não está adequado. Como consequência, as estimativas também estarão inadequadas. Ao analisar os cálculos da rede Bayesiana, apresentados na Figura 3.14, conclui-se que este antipadrão é detectado na rede Bayesiana base.

Para os antipadrões *Product Owner externo* e *Product Owner sem autoridade*, todas as consequências são observadas no processo de alimentar a rede Bayesiana, pois todas são nós de entrada na mesma. Para o antipadrão *Ciclos de aprendizagem longos ou inexistentes*, dado que as principais partes interessadas não participam das Reuniões de Revisão e assumindo que o Product Owner não tem autoridade, pode-se definir os valores dos indicadores *Adap-*

Tabela 3.6: Antipadrões do Scrum apresentados em Eloranta et al. [60].

Nome	Consequências
Documento de requisitos extenso	Equipe não entende requisitos. Requisitos não são ordenados.
Product Owner externo	Product Owner não colabora com a equipe. Equipe não entende os requisitos. Product Owner interrompe a equipe durante as sprints.
Product Owner sem autoridade	Product Owner não entende as necessidades reais dos clientes e não consegue detalhar o Backlog do produto.
Ciclos de aprendizagem longos ou inexistentes	Ciente não participa das Reuniões de Revisão. Produto incorreto pode ser construído. Todas as consequências do antipadrão Product Owner sem autoridade.
<i>Backlog</i> do produto desordenado	Equipe não visualiza riscos e valor do negócio das atividades. É possível que um produto sem valor de negócio seja entregue.
Estimativas do trabalho passadas para as equipes	Estimativas incorretas, pois, provavelmente, a equipe sabe estimar melhor. Resulta em objetivos inalcançáveis. Pode desmotivar a equipe.
Monitoramento de progresso usando horas	Estimativas incorretas.
Equipes semifuncionais	Equipe colabora menos. Afeta negativamente o resultado das sprints.
Cliente causa interrupção	Reduz a eficiência da equipe.
Ausência de Reunião de Retrospectiva	Não há melhoria contínua.
Progresso invisível	Problema no planejamento das sprints.
Duração de sprint variável	Afeta negativamente a velocidade da equipe, pois perde-se o comprometimento com os objetivos da sprint.
sprint longa	Menor eficiência da equipe. Pode causar o antipadrão <i>Cliente causa interrupção</i> .
Testes na sprint seguinte	Incremento com menor qualidade devido ao débito técnico. Menor eficiência devido ao retrabalho.

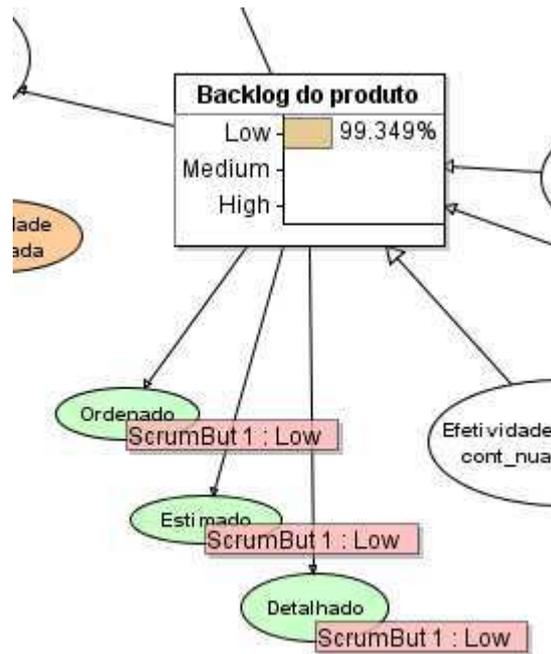


Figura 3.14: Resultado da rede Bayesiana para o antipadrão 1 de Eloranta et al. [60]. Imagem extraída do AgenaRisk.

tação, Inspeção e Qualidade do Product Owner como *Baixo*. Além disso, o mesmo para os indicadores do *Backlog do produto*. Dessa forma, mesmo assumindo que a equipe é a melhor possível, as chances do cliente estar satisfeito ao final do projeto é de 30,773%, como apresentado na Figura 3.15. Desta forma, este antipadrão é detectado na rede Bayesiana base.

Backlog do produto desordenado é relacionado com um indicador: *Ordenado*. De acordo com a relação causal entre o nó em questão e o nó *Incremento*, conclui-se que este antipadrão é detectado na rede Bayesiana base.

Estimativas do trabalho passadas para as equipes é relacionado com dois indicadores: *Estimado* e *Objetivo da sprint*. Por outro lado, não há relação causal entre a qualidade das estimativas e a qualidade do trabalho em equipe (i.e., motivação). Desta forma, este antipadrão é parcialmente detectado na rede Bayesiana base. Por outro lado, caso a equipe esteja desmotivada, isso será detectado ao alimentar a rede Bayesiana. Além disso, a motivação da equipe pode ser adicionada como um indicador de *Trabalho em equipe*.

Monitoramento de progresso usando horas é relacionado com os nós *Monitoramento de progresso* e *Estimado*. Apesar de não haver relação causal direta na rede Bayesiana base

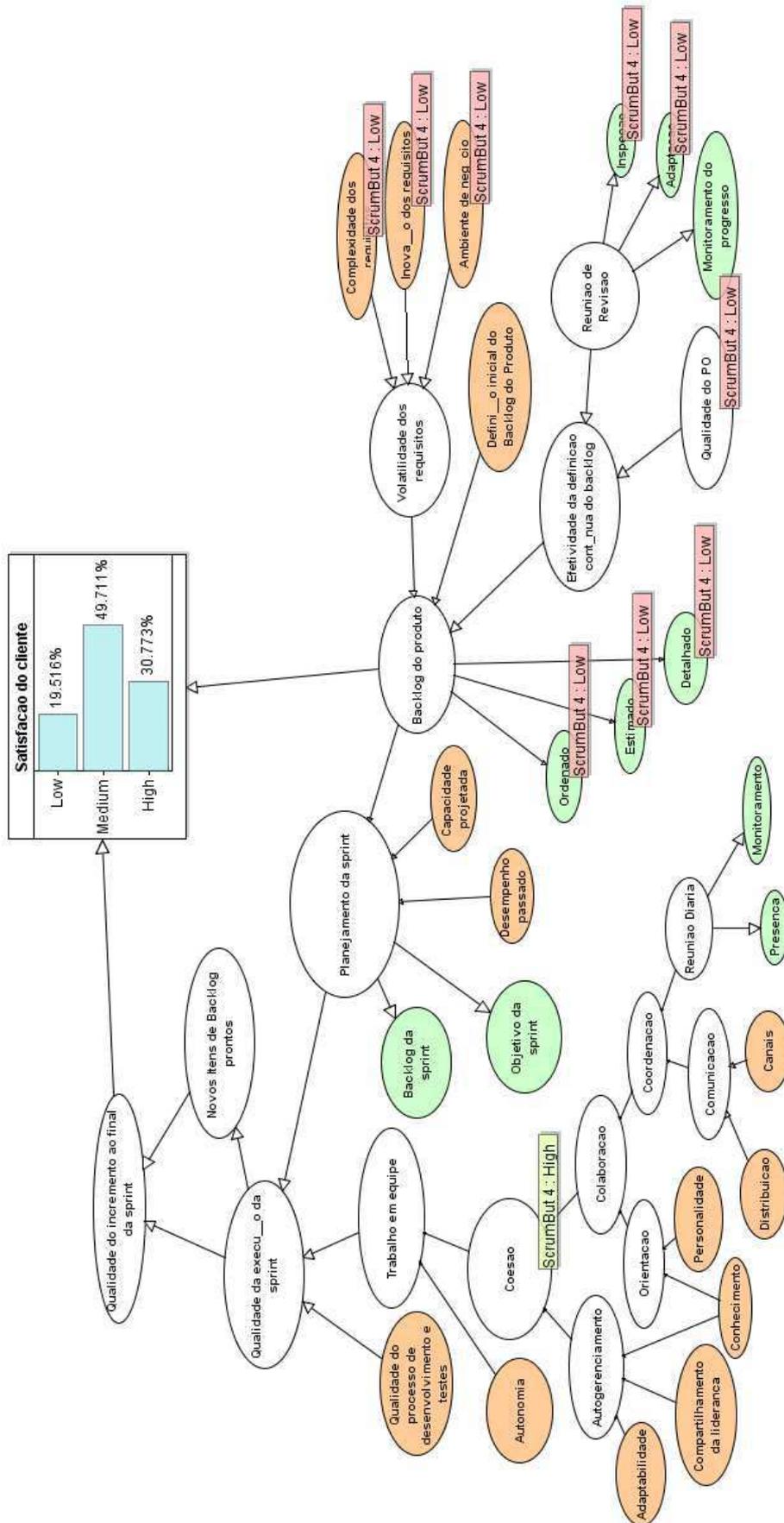


Figura 3.15: Resultado da rede Bayesiana para o antipadrão 4 de Eloranta et al. [60]. Imagem extraída do AgenaRisk.

entre esses dois fatores, conclui-se que este antipadrão é detectado pois tratam-se de dois indicadores.

Equipes semifuncionais é relacionado com o nó *Conhecimento*. De acordo com a relação causal entre o nó em questão e os nós *Colaboração* e *Qualidade da execução da sprint*, conclui-se que este antipadrão é detectado na rede Bayesiana base.

Cliente causa interrupção, o mesmo é relacionado com o nó *Autonomia*. De acordo com a relação causal entre o nó em questão e o nó *Qualidade da execução da sprint*, conclui-se que este antipadrão é detectado na rede Bayesiana base.

Ausência de Reunião de Retrospectiva não é modelado na rede Bayesiana, pois a mesma deve ser utilizada para auxiliar durante Reuniões de Retrospectiva. Ou seja, de acordo com o escopo da rede Bayesiana, este antipadrão é inválido.

Progresso invisível é relacionado com o nó *Monitoramento de progresso*. De acordo com o resultado de uma análise de sensibilidade apresentado na Figura 3.16, a influência do mesmo em *Reunião de Planejamento* é irrelevante. Por outro lado, o nó *Desempenho passado* é diretamente relacionado com este antipadrão, pois é impossível se basear no desempenho passado para realizar o planejamento da sprint se o progresso da equipe não for calculado. Dessa forma, considera-se que este antipadrão é detectado na rede Bayesiana base.

Duração de sprint variável e *Sprint longa* são ambas relacionadas com o nó *Autonomia*, pois como consequência, dado que não há comprometimento da gerência com os objetivos da sprint, tende-se a ter mais interrupções durante a sprint. Como apresentado na Figura 3.17, o nó *Autonomia* tem influência relevante no nó *Qualidade da execução da sprint*. Dessa forma, conclui-se que este antipadrão é detectado na rede Bayesiana base.

Testes na sprint seguinte relaciona-se com o nó *Qualidade do processo de desenvolvimento e testes*. Se os testes não tiverem sido realizados na sprint anterior, a qualidade do código poderá estar ruim e, neste caso, afetar negativamente o resultado da sprint. Devido à relação causal entre o nó *Qualidade do processo de desenvolvimento e testes* e o nó *Qualidade da execução da sprint*, conclui-se que este antipadrão é detectado na rede Bayesiana base.

Ou seja, dos 14 antipadrões apresentados em Eloranta et al. [60], 1 é inválido, 12 são detectados e 1 é parcialmente detectado pela rede Bayesiana base. Com estes resultados,

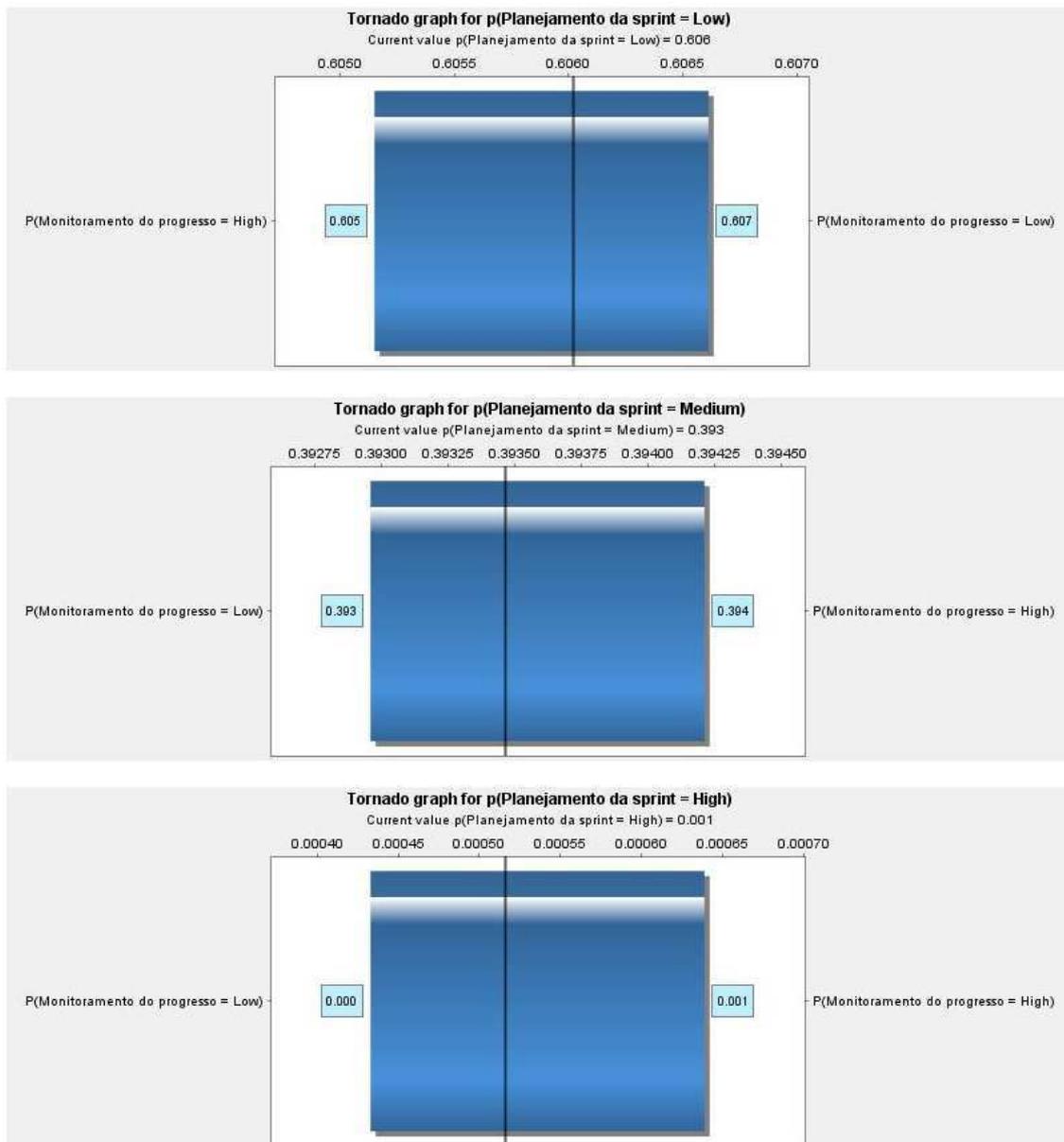


Figura 3.16: Análise de sensibilidade do nó *Reunião de Planejamento* com relação ao nó *Monitoramento de progresso*. Imagem extraída do AgenaRisk.

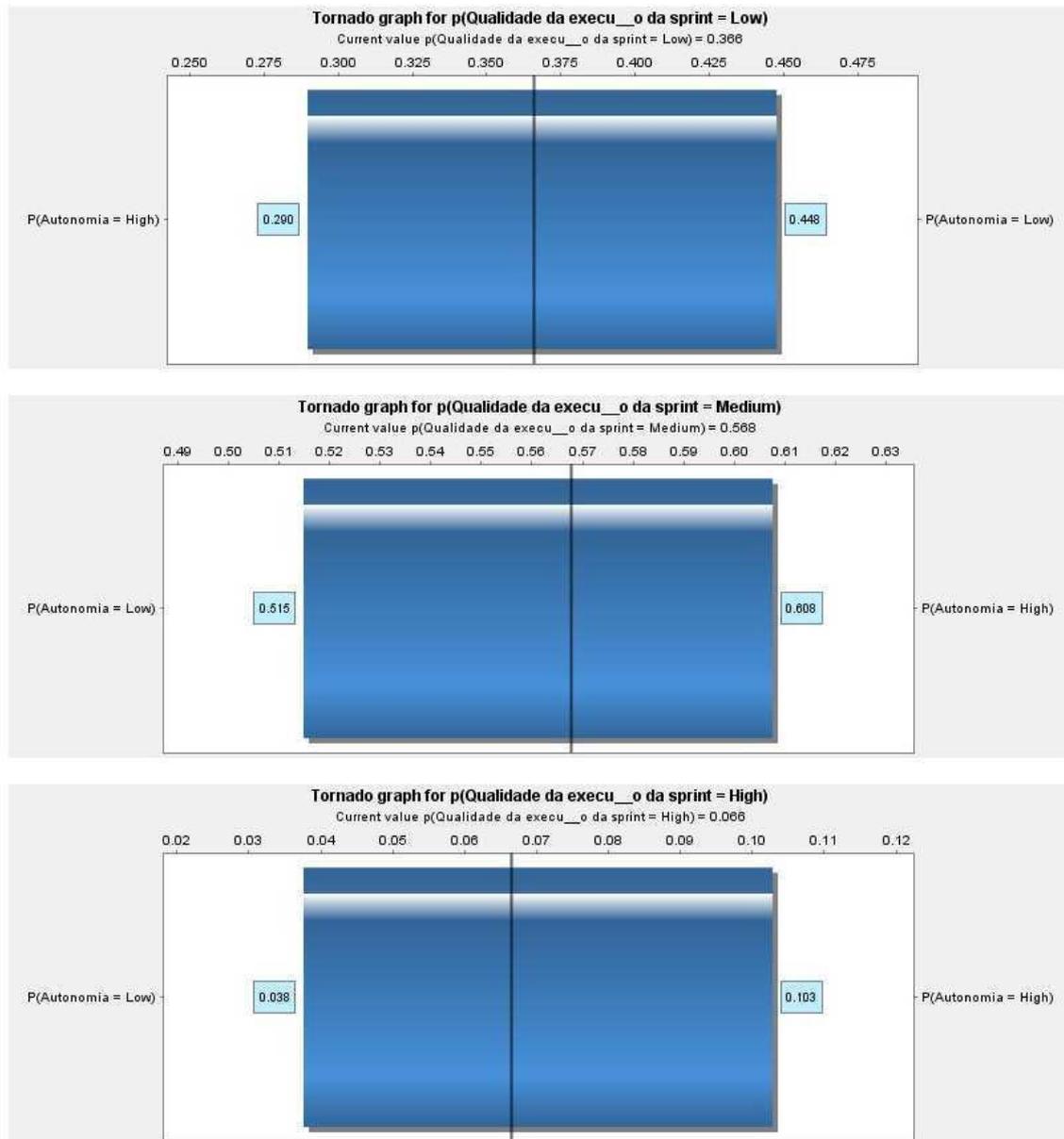


Figura 3.17: Análise de sensibilidade do nó *Qualidade da execução da sprint* com relação ao nó *Autonomia*. Imagem extraída do AgenaRisk.

conclui-se que a rede Bayesiana pode auxiliar na identificação dos principais problemas na aplicação do Scrum.

3.5 Validação com Cenários de Simulação

Para validar modelos de processos de desenvolvimento de software, é possível aplicá-los a cenários e comparar os resultados calculados com os esperados [55, 131]. Desta forma, para validar a rede Bayesiana, os resultados esperados e calculados de doze cenários foram comparados. Cada cenário descreve uma situação que poderia ocorrer em um projeto Scrum de desenvolvimento de software. Os cenários foram descritos e alimentados com base na experiência do pesquisador e dos especialistas de domínio que colaboraram com a pesquisa. Para a definição dos resultados esperados e calculados, desconsiderou-se riscos que não sejam diretamente relacionados ao processo de desenvolvimento, por exemplo: falta de recursos, mudanças dos objetivos organizacionais e catástrofes naturais. Além disso, considera-se que, em todos os cenários, o projeto é adequado para a adoção do Scrum. Para cada cenário, evidências foram definidas para as variáveis as quais os especialistas julgaram estarem relacionadas com a descrição do mesmo. Além disso, foram definidos os valores esperados para o nó alvo (i.e., *Satisfação do cliente*). Os cenários foram utilizados no processo de validação e calibração do modelo. Dessa forma, ao final dessa etapa, os resultados esperados de todos os cenários foram previstos com sucesso.

3.5.1 Cenário 1

Uma empresa, experiente em usar Scrum em projetos de desenvolvimento de software, contrata um grupo de desenvolvedores que estão habituados com o método tradicional de desenvolvimento de software para formar a equipe de desenvolvimento de um projeto. Desta forma, todos os envolvidos no projeto, exceto os desenvolvedores, têm experiência e habilidades necessárias para trabalhar em um projeto Scrum. Os desenvolvedores não estão habituados em desenvolver o produto de forma iterativa, se autogerenciar e trabalhar de forma colaborativa em todos as fases de desenvolvimento do produto. Os dados de entrada para o modelo desse projeto são apresentados na Tabela 3.7.

Para este cenário, espera-se que, por conta da baixa qualidade do código, as chances

Tabela 3.7: Evidências para o cenário 1

Nó	Evidência
Monitoramento do progresso	Alto
Qualidade do Product Owner	Alto
Desempenho passado	Baixo
Capacidade projetada	Baixo
Objetivo da sprint	Baixo
Autonomia	Baixo
Adaptabilidade	Baixo
Compartilhamento da liderança	Baixo
Sincronização	Baixo
Qualidade do processo de desenvolvimento e testes	Baixo

de ter um cliente satisfeito sejam baixas. Na Figura 3.18, os valores calculados para as principais entidades do modelo são apresentados. Como pode-se perceber, a probabilidade de satisfação do cliente é baixa e isso é causado, principalmente, pela má qualidade do código. Ou seja, o resultado calculado está em conformidade com os resultados do cenário.

3.5.2 Cenário 2

Uma empresa, ao perceber que muitas outras empresas estão gerenciando seus projetos de desenvolvimento de software seguindo a metodologia ágil, resolve segui-la. Dado que Scrum é o arcabouço ágil mais popular, a empresa resolve usar Scrum para um projeto teste. Desenvolvedores que estão habituados com o método tradicional de desenvolvimento de software são nomeados para integrar a equipe de desenvolvimento. Para o cargo de Product Owner, nomeou-se um gerente de projetos sênior habituado ao microgerenciamento e comando-e-controle. Além disto, as outras partes envolvidas no projeto não acham necessário ter tantas reuniões com a equipe. Desta forma, nenhum dos envolvidos no projeto tem experiência e habilidades necessárias para trabalhar em um projeto Scrum, ou seja, os desenvolvedores não estão habituados em desenvolver o produto de forma iterativa, se autogerenciar e trabalhar de forma colaborativa em todas as fases de desenvolvimento do produto. Além disto, o Pro-

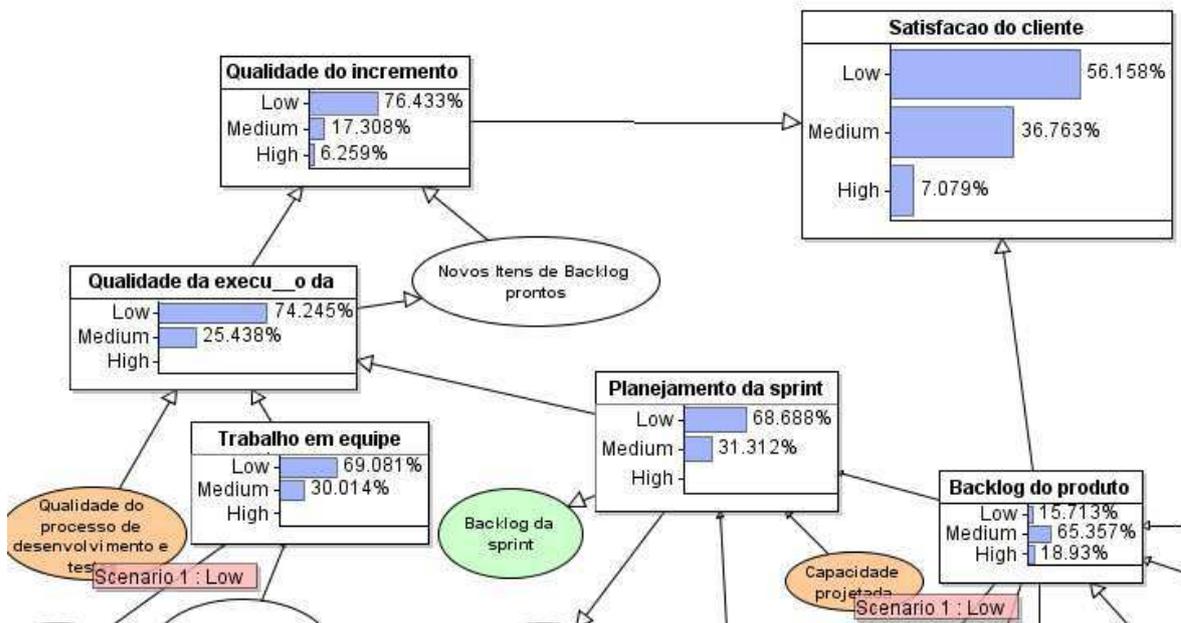


Figura 3.18: Resultados calculados para o cenário 1. Imagem extraída do AgenaRisk.

duct Owner não tem a personalidade e habilidades para liderar uma equipe autogerenciada e não há o apoio necessário para validar o produto. Os dados de entrada para o modelo desse projeto são apresentados na Tabela 3.8.

Para esse projeto, é esperado que o modelo sinalize que o produto não será entregue em sua completude e itens entregues terão baixa qualidade, não atingindo expectativas do cliente. Na Figura 3.19, os valores calculados para as principais entidades do modelo são apresentados. Como pode-se perceber, a probabilidade de satisfação do cliente é baixa e isso é causado pela má qualidade do código e gerenciamento do Backlog do produto. Ou seja, o resultado calculado está em conformidade com os resultados do cenário.

3.5.3 Cenário 3

Uma empresa é contratada para desenvolver um sistema. A contratada fornece os desenvolvedores e o Scrum Master; e, a contratante fornece o Product Owner. As empresas são de países diferentes e a contratada deve enviar dois integrantes para trabalhar na sede da empresa contratante. Apenas o Product Owner representará a contratante nas reuniões do Scrum, e participará apenas duas vezes por semana das Reuniões Diárias. Além disto, há

Tabela 3.8: Evidências para o cenário 2

Nó	Evidência
Qualidade do Product Owner	Baixo
Objetivo da sprint	Baixo
Autonomia	Baixo
Adaptabilidade	Baixo
Compartilhamento da liderança	Baixo
Conhecimento	Baixo
Personalidade	Baixo
Canais	Baixo
Qualidade do processo de desenvolvimento e testes	Baixo

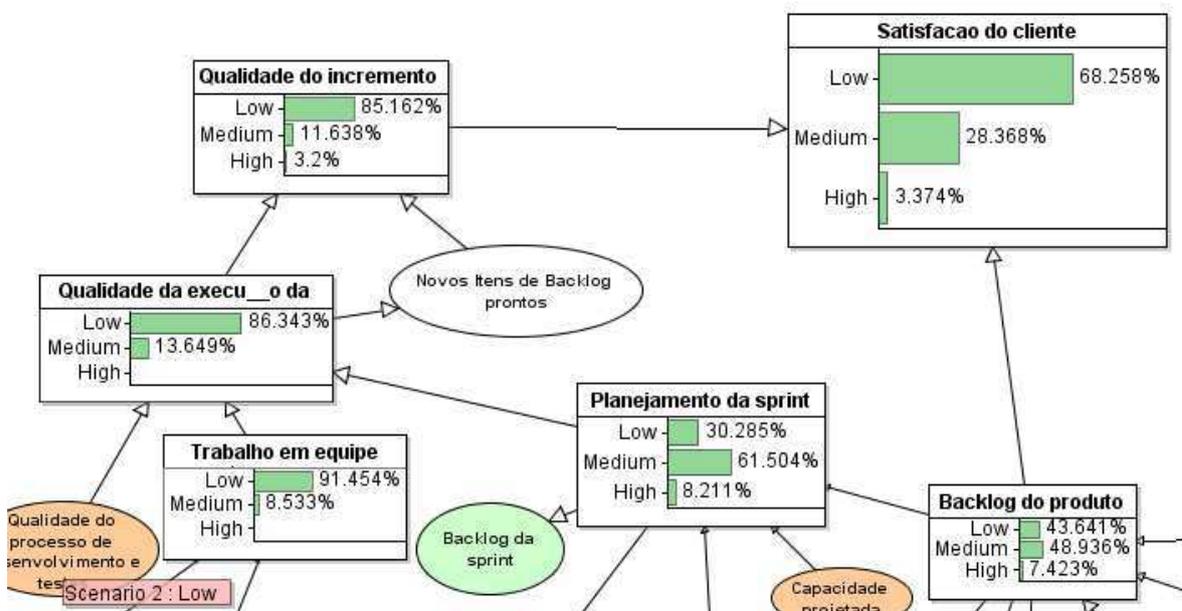


Figura 3.19: Resultados calculados para o cenário 2. Imagem extraída do AgenaRisk.

Tabela 3.9: Evidências para o cenário 3

Nó	Evidência
Qualidade do Product Owner	Baixo
Autonomia	Baixo
Adaptabilidade	Alto
Conhecimento	Alto
Distribuição	Baixo
Canais	Baixo
Presença	Baixo
Definição inicial do Backlog do produto	Baixo
Qualidade do processo de desenvolvimento e testes	Moderado

uma diferença de fuso horário de seis horas entre as sedes das duas empresas, que prejudica a comunicação entre os desenvolvedores, e entre os desenvolvedores e o Product Owner. A empresa contratada tem experiência e competência para trabalhar em projetos Scrum. Por outro lado, a empresa contratante migrou para o gerenciamento ágil de projetos recentemente e, historicamente, não consegue definir a visão do produto pois há muitos clientes e, consequentemente, muitos conflitos de interesses. Historicamente, o escopo da sprint costuma ser alterado frequentemente pois o Product Owner procura atender de imediato as solicitações dos clientes, e, geralmente, as funcionalidades não são escritas com todos os detalhes necessários e as definições ficam a cargo dos desenvolvedores. Os dados de entrada para o modelo desse projeto são apresentados na Tabela 3.9.

Para esse projeto, é esperado que o modelo sinalize que há poucas chances de o projeto não atender às expectativas. As principais causas são o envolvimento dos clientes em momentos inoportunos, problemas de comunicação e falta de visão e definição de objetivos do projeto. As saídas do modelo são apresentadas na Figura 3.20. Analisando as saídas do mesmo, pode-se concluir que os valores calculados estão em conformidade com os esperados.

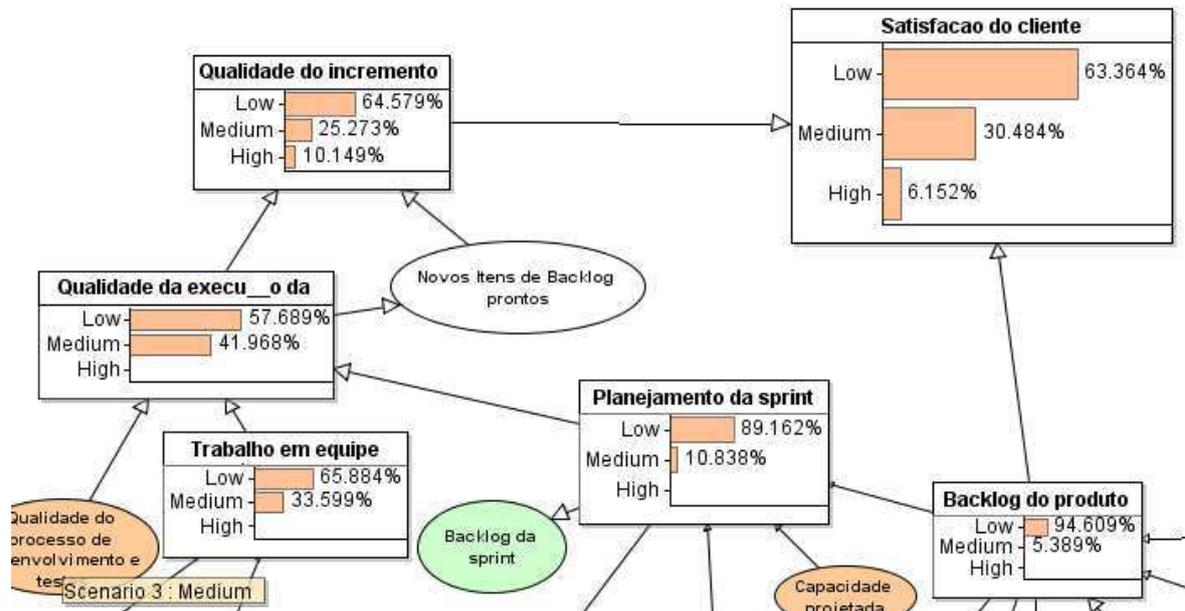


Figura 3.20: Resultados calculados para o cenário 3. Imagem extraída do AgenaRisk.

3.5.4 Cenário 4

Uma empresa com várias sedes no Brasil e madura em gestão ágil indica uma equipe para desenvolver um novo sistema usando Scrum. Os membros dessa equipe trabalham em sedes diferentes da empresa, porém não há diferença de fuso horário. A empresa utiliza as melhores ferramentas para comunicação a longa distância e gerenciamento de projetos. O sistema resultante do projeto está bem definido e tanto os desenvolvedores como os clientes e Product Owner tem experiência e competência para trabalhar com o Scrum. Os dados de entrada para o modelo desse projeto são apresentados na Tabela 3.10.

Para esse projeto, é esperado que o produto seja entregue, podendo haver relativos atrasos. Dessa forma, espera-se que o modelo sinalize que o projeto terá sucesso, pois apesar da equipe ser distribuída, as tecnologias de comunicação a longa distância e as ferramentas de gerenciamento de projetos que existem atualmente devem ajudar a equipe a diminuir o impacto da distância. As saídas do modelo são apresentadas na Figura 3.21. Analisando as saídas do mesmo, pode-se concluir que os valores calculados estão em conformidade com os esperados.

Tabela 3.10: Evidências para o cenário 4

Nó	Evidência
Detalhado	Alto
Ordenado	Alto
Desempenho passado	Alto
Capacidade projetada	Alto
Backlog da sprint	Alto
Autonomia	Alto
Adaptabilidade	Alto
Conhecimento	Alto
Distribuição	Moderado
Canais	Alto

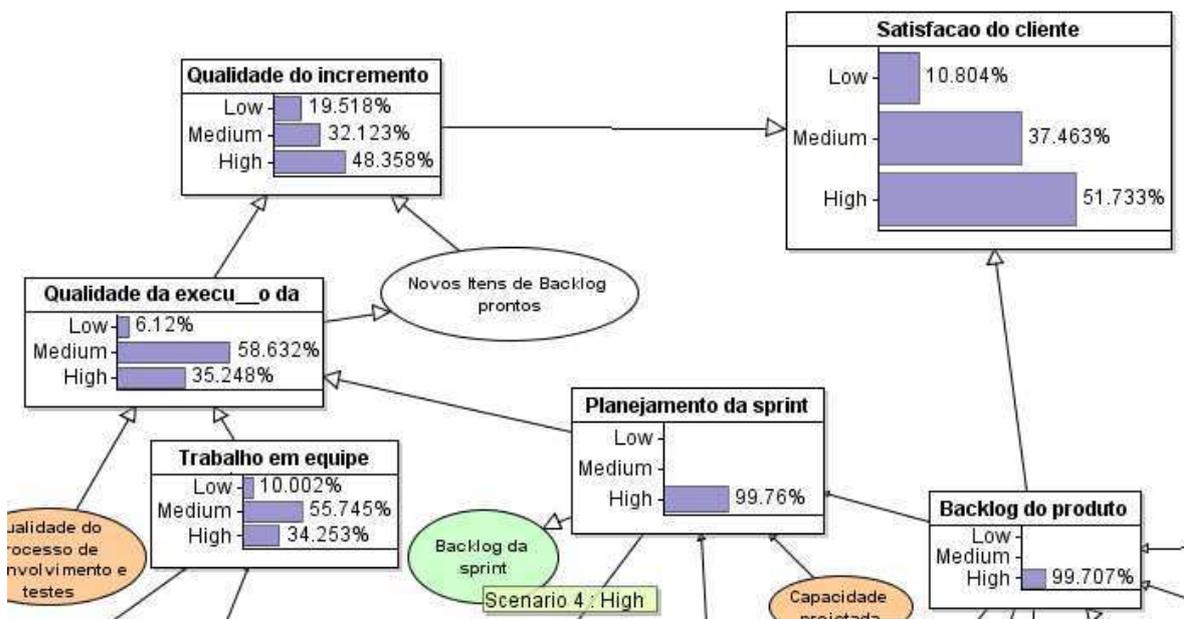


Figura 3.21: Resultados calculados para o cenário 4. Imagem extraída do AgenaRisk.

Tabela 3.11: Evidências para o cenário 5

Nó	Evidência
Detalhado	Alto
Estimado	Alto
Ordenado	Alto
Autonomia	Alto
Adaptabilidade	Alto
Compartilhamento da liderança	Alto
Conhecimento	Alto
Distribuição	Baixo
Canais	Baixo
Qualidade do processo de desenvolvimento e testes	Alto

3.5.5 Cenário 5

Uma empresa nova com recursos humanos competentes indica uma equipe para desenvolver um novo sistema usando Scrum. Os membros dessa equipe trabalham em sedes diferentes da empresa, porém não há diferença de fuso horário. A empresa utiliza ferramentas precárias para comunicação a longa distância e gerenciamento de projetos por não querer aumentar os custos do projeto. O sistema resultante do projeto está bem definido e tanto os desenvolvedores como os clientes e Product Owner tem experiência e competência para trabalhar com o Scrum. Os dados de entrada para o modelo desse projeto são apresentados na Tabela 3.11.

Para esse projeto, é esperado que o projeto seja entregue de acordo com necessidade do cliente. Podem ocorrer atrasos e dificuldade de integração, mas a probabilidade é baixa. As saídas do modelo são apresentadas na Figura 3.22. Analisando as saídas do mesmo, pode-se concluir que os valores calculados estão em conformidade com os esperados.

3.5.6 Cenário 6

Os executivos e gerentes de uma empresa de desenvolvimento de software decidem migrar do método tradicional de gerenciamento de projetos para o ágil. O *waterfall* era o método utilizado para desenvolver software, mas dado que a empresa entrará em um novo nicho de

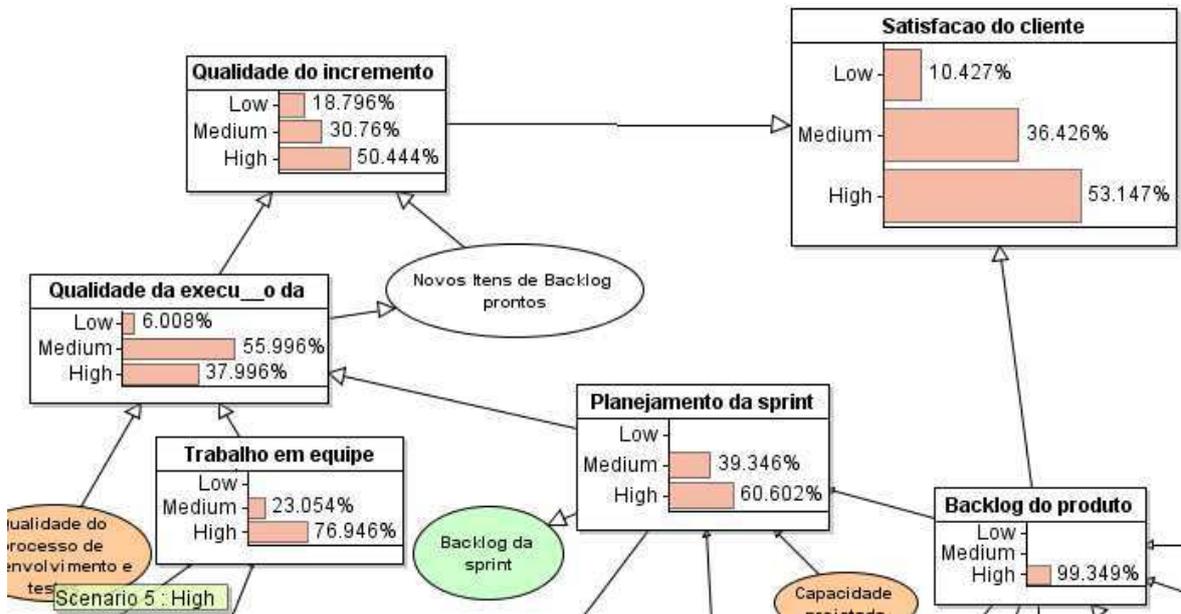


Figura 3.22: Resultados calculados para o cenário 5. Imagem extraída do AgenaRisk.

mercado que utiliza novas tecnologias, decidiu utilizar a metodologia ágil, mais especificamente, o Scrum. Desta forma, uma equipe foi nomeada para trabalhar no primeiro projeto da empresa usando o Scrum. Porém, treinamentos não foram fornecidos para os desenvolvedores e, conseqüentemente, eles continuaram trabalhando com a mesma cultura que estavam habituados. Ou seja, as equipes são formadas por especialistas que trabalham apenas nas suas áreas específicas e o ciclo de desenvolvimento é similar ao *waterfall*, mas em um espaço mais curto de tempo. Por outro lado, os clientes e o Product Owner envolvidos no projeto são competentes e comprometidos com a gestão ágil. Os dados de entrada para o modelo desse projeto são apresentados na Tabela 3.12.

Para esse projeto, é esperado que o modelo sinalize que há mais chances do projeto obter sucesso do que falhar. Dado que os clientes e o Product Owner seguem as regras do Scrum, os problemas com a equipe de desenvolvimento serão detectados. Apesar disto, o modelo deve indicar que os desenvolvedores necessitam de treinamento em práticas ágeis e Scrum para aumentar as chances de sucesso do projeto. As saídas do modelo são apresentadas na Figura 3.23. Analisando as saídas do mesmo, pode-se concluir que os valores calculados estão em conformidade com os esperados.

Tabela 3.12: Evidências para o cenário 6

Nó	Evidência
Qualidade do Product Owner	Alto
Detalhado	Alto
Estimado	Baixo
Ordenado	Alto
Desempenho passado	Moderado
Capacidade projetada	Baixo
Autonomia	Moderado
Adaptabilidade	Baixo
Compartilhamento da liderança	Baixo
Conhecimento	Alto
Distribuição	Alto
Canais	Alto
Sincronização	Baixo
Qualidade do processo de desenvolvimento e testes	Moderado

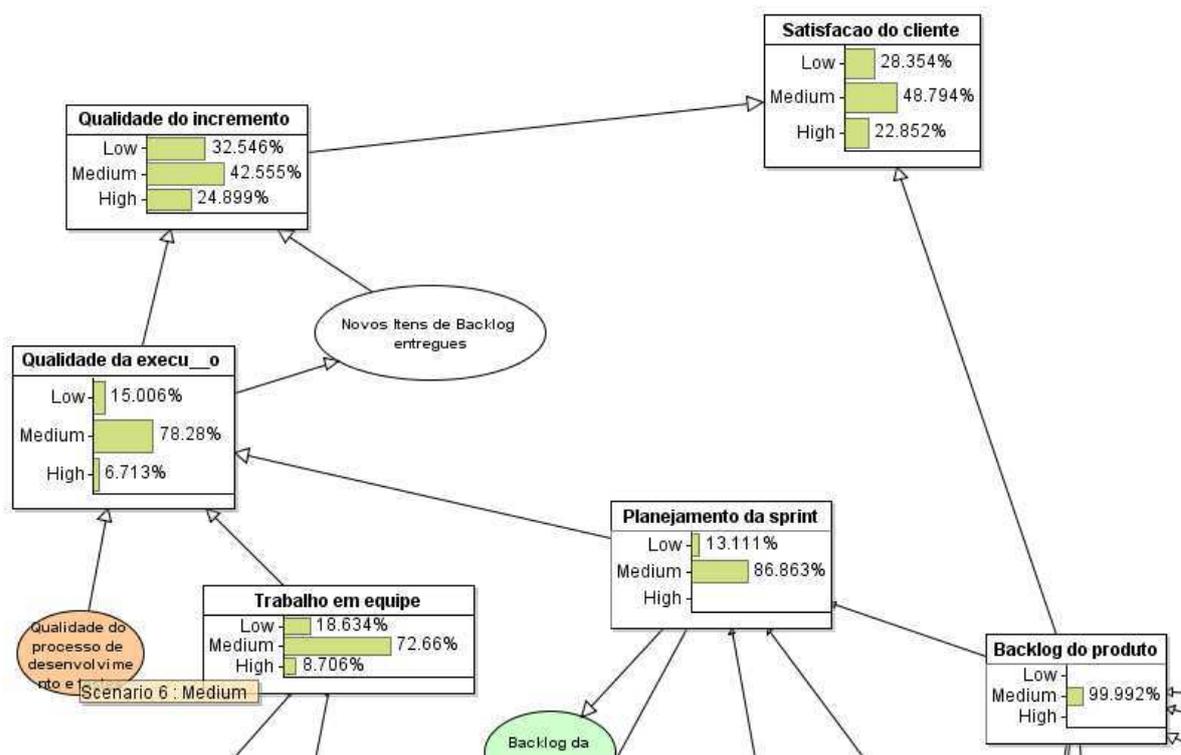


Figura 3.23: Resultados calculados para o cenário 6. Imagem extraída do AgenaRisk.

3.5.7 Cenário 7

Uma empresa é contratada para desenvolver um sistema. A contratada fornece os desenvolvedores e Scrum Master; e, a contratante fornece o Product Owner e um arquiteto de sistemas que deve compor a equipe de desenvolvimento. Pelo contrato firmado entre as duas empresas, o arquiteto da empresa contratante deve ter o poder final de decisões técnicas da equipe. Além disto, apenas o Product Owner representará a contratante nas reuniões do Scrum, participará apenas duas vezes por semana das Reuniões Diárias e apenas se reunirá bimestralmente com os clientes da contratante para discutir o projeto. Os desenvolvedores e o Product Owner são treinados e experientes em utilizar o Scrum. Por outro lado, apenas a empresa contratante tem o poder de tomar decisões sobre o projeto e o Product Owner tem que se adaptar a essas condições do contrato. O arquiteto é o melhor funcionário, em termos técnicos, da empresa contratante. Por outro lado, ele costuma centralizar as decisões técnicas e não tem facilidade de trabalhar em equipe. Os dados de entrada para o modelo desse projeto são apresentados na Tabela 3.13.

Para esse projeto, é esperado retrabalho, uma vez que decisões do projeto são posterga-

Tabela 3.13: Evidências para o cenário 7

Nó	Evidência
Qualidade do Product Owner	Alto
Desempenho passado	Alto
Autonomia	Baixo
Compartilhamento da liderança	Baixo
Conhecimento	Alto
Personalidade	Baixo
Distribuição	Alto
Canais	Alto
Presença	Moderado
Qualidade do processo de desenvolvimento e testes	Moderado

das. Também não teremos muita autonomia na equipe visto que o arquiteto é centralizador. Dessa forma, a satisfação do cliente deve ser moderada. As saídas do modelo são apresentadas na Figura 3.24. Analisando as saídas do mesmo, pode-se concluir que os valores calculados estão em conformidade com os esperados.

3.5.8 Cenário 8

Uma empresa contrata um consultor ágil renomado para capacitar seus executivos, desenvolvedores e Scrum Masters. Depois da consultoria, uma equipe é alocada para trabalhar em um projeto cujo o objetivo não está bem definido. Além disto, a empresa decidiu não seguir algumas recomendações do consultor, e iniciou o projeto sem definir a data final e os atributos básicos da definição da visão do produto. Os dados de entrada para o modelo desse projeto são apresentados na Tabela 3.14.

Para esse projeto, é esperada a possível insatisfação do cliente com relação às restrições de tempo, custo, escopo e qualidade do projeto. As saídas do modelo são apresentadas na Figura 3.25. Analisando as saídas do mesmo, pode-se concluir que os valores calculados estão em conformidade com os esperados.

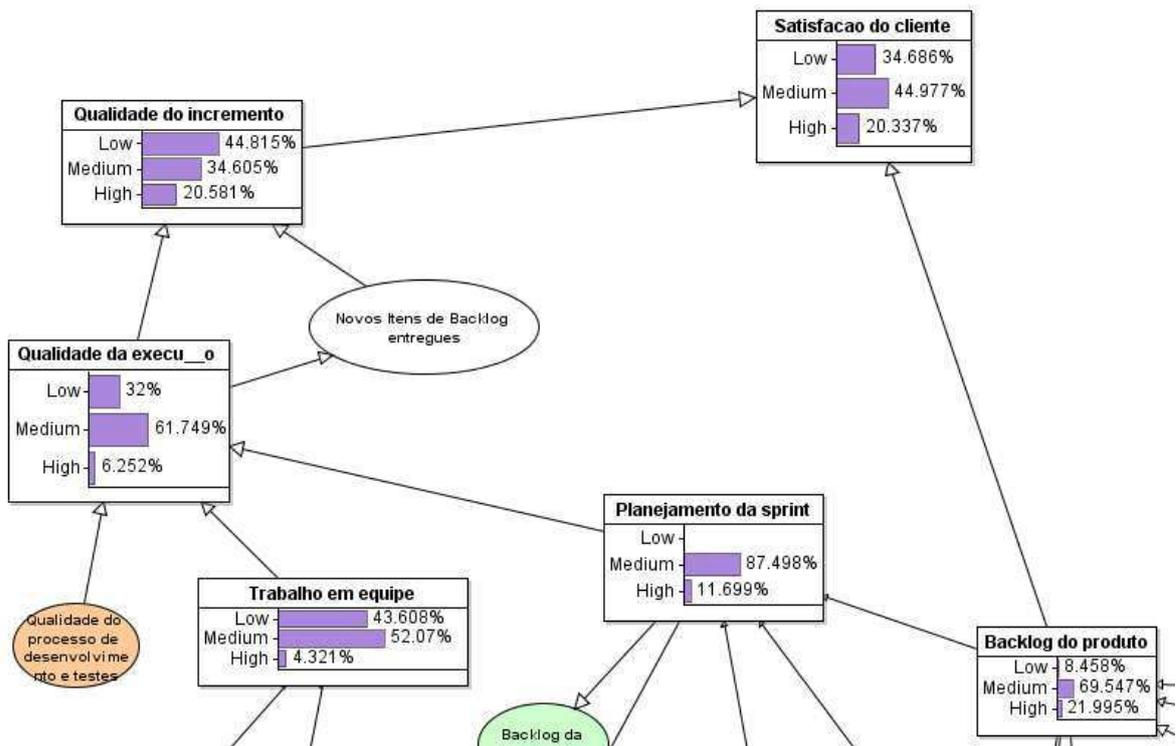


Figura 3.24: Resultados calculados para o cenário 7. Imagem extraída do AgenaRisk.

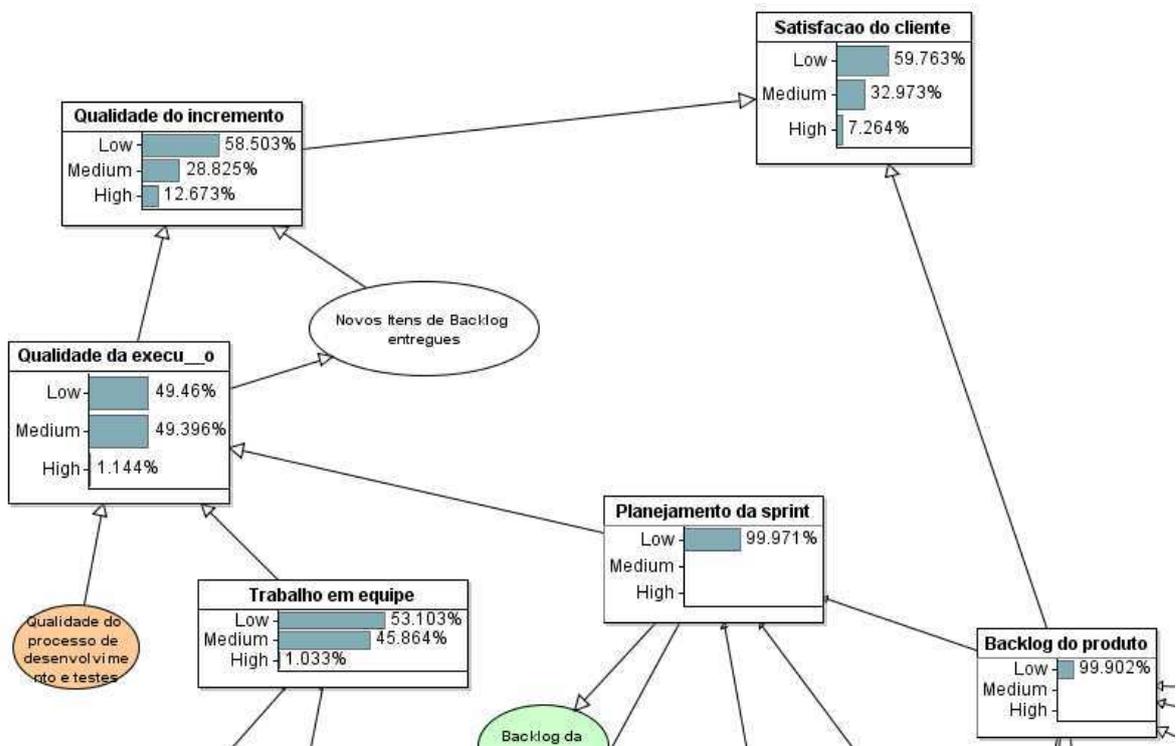


Figura 3.25: Resultados calculados para o cenário 8. Imagem extraída do AgenaRisk.

Tabela 3.14: Evidências para o cenário 8

Nó	Evidência
Detalhado	Baixo
Ordenado	Baixo
Desempenho passado	Baixo
Capacidade projetada	Baixo
Backlog da sprint	Baixo
Objetivo da sprint	Baixo
Autonomia	Baixo
Adaptabilidade	Baixo
Compartilhamento da liderança	Baixo
Conhecimento	Moderado
Distribuição	Alto
Canais	Alto
Sincronização	Baixo
Qualidade do processo de desenvolvimento e testes	Baixo

Tabela 3.15: Evidências para o cenário 9

Nó	Evidência
Adaptação	Alto
Detalhado	Alto
Estimado	Alto
Ordenado	Alto
Backlog da sprint	Alto
Objetivo da sprint	Alto
Autonomia	Alto
Adaptabilidade	Alto
Compartilhamento da liderança	Alto
Conhecimento	Alto
Distribuição	Alto
Canais	Alto
Presença	Alto
Sincronização	Alto
Qualidade do processo de desenvolvimento e testes	Alto

3.5.9 Cenário 9

Uma empresa contrata um consultor ágil renomado para capacitar seus executivos, desenvolvedores e Scrum Masters. Depois da consultoria, os executivos e desenvolvedores trabalham colaborativamente para planejar um projeto seguindo as recomendações do consultor. Desta forma, as melhores práticas para gestão de projetos ágeis foram colocadas em práticas, incluindo as definições do produto a ser construído durante o projeto. Os dados de entrada para o modelo desse projeto são apresentados na Tabela 3.15.

Para esse projeto, é esperado sucesso na entrega. Ou seja, o modelo deve sinalizar que há boas chances do cliente ficar satisfeito com a entrega. As saídas do modelo são apresentadas na Figura 3.26. Analisando as saídas do mesmo, pode-se concluir que os valores calculados estão em conformidade com os esperados.

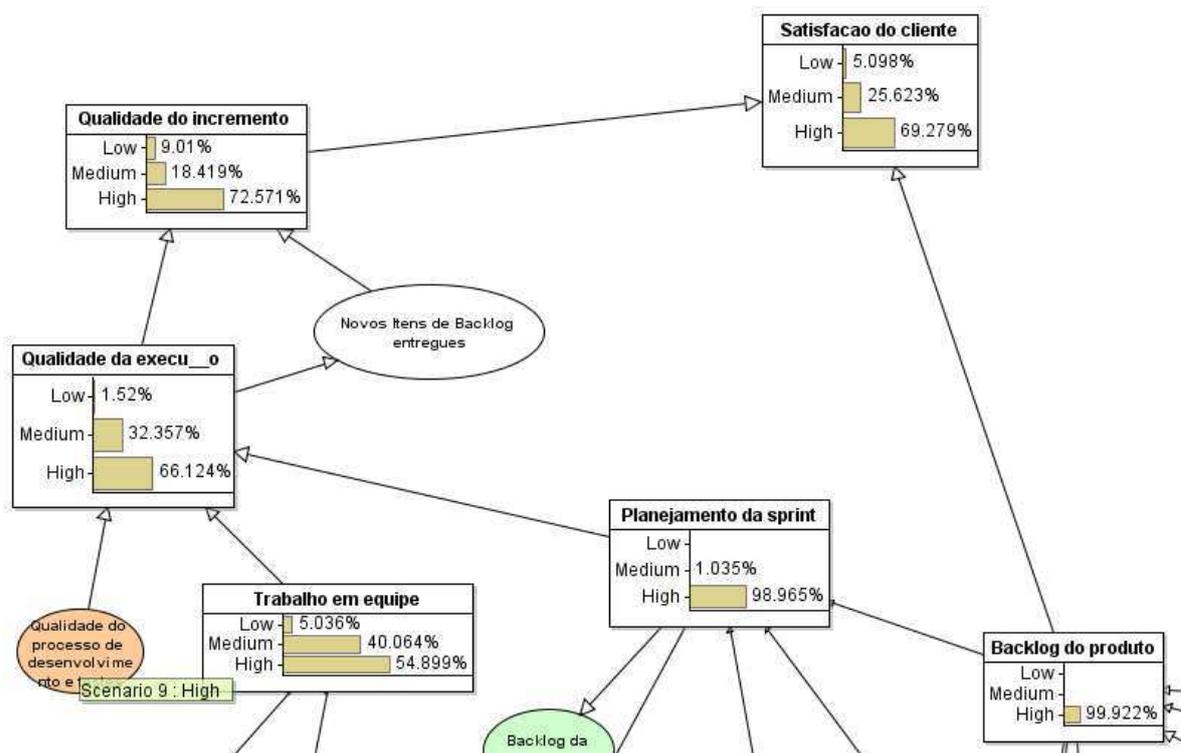


Figura 3.26: Resultados calculados para o cenário 9. Imagem extraída do AgenaRisk.

3.5.10 Cenário 10

Como é feito em muitas empresas, uma empresa promove um dos seus melhores desenvolvedores, sem nenhum treinamento específico, para gerente. Para um projeto cujo objetivo é desenvolver um novo sistema, nomeou-se o recém-promovido gerente para ser o Product Owner. Apesar de ser um ótimo desenvolvedor, o recém-promovido gerente não tem experiência com gerenciamento de projetos e não tem as habilidades necessárias para guiar a equipe de desenvolvimento. Para o projeto, além do recém-promovido gerente, uma equipe experiente de desenvolvedores foi alocada, os clientes se comprometeram a seguir o Scrum e o produto foi definido usando as melhores práticas ágeis. Os dados de entrada para o modelo desse projeto são apresentados na Tabela 3.16.

Para esse projeto, mesmo a equipe de desenvolvimento sendo experiente e utilizando boas práticas de desenvolvimento ágeis, espera-se que o projeto terá o escopo parcialmente entregue e, possivelmente, fora do prazo. Isso é causado pela falta de habilidade do Product Owner em conduzir equipe, mitigar riscos e gerenciar a expectativa do cliente. As saídas do modelo são apresentadas na Figura 3.27. Analisando as saídas do mesmo, pode-se concluir

Tabela 3.16: Evidências para o cenário 10

Nó	Evidência
Qualidade do Product Owner	Baixo
Detalhado	Baixo
Estimado	Baixo
Ordenado	Baixo
Desempenho passado	Alto
Capacidade projetada	Alto
Backlog da sprint	Baixo
Objetivo da sprint	Baixo
Adaptabilidade	Alto
Conhecimento	Alto
Distribuição	Alto
Canais	Alto
Sincronização	Alto
Qualidade do processo de desenvolvimento e testes	Alto

que os valores calculados estão em conformidade com os esperados.

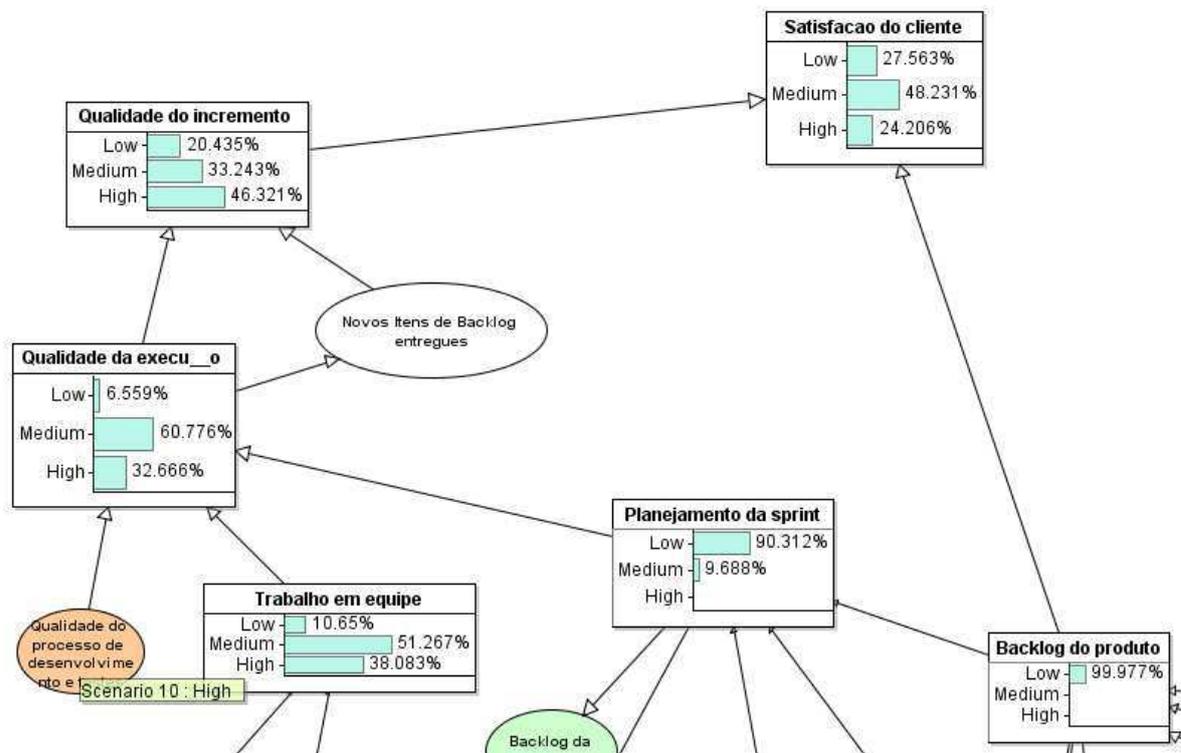


Figura 3.27: Resultados calculados para o cenário 10. Imagem extraída do AgenaRisk.

3.5.11 Cenário 11

Uma empresa especializada em desenvolvimento de software utiliza Scrum para clientes de vários segmentos. Um novo cliente fechou contrato e não tem a cultura de metodologias ágeis. Toda equipe será formada pelos profissionais da contratada, exceto o PO. O Product Owner poderá se reunir pessoalmente apenas uma vez por bimestre com o time de desenvolvimento, informa que dúvidas podem ser enviadas por e-mails, que o mesmo se compromete a responder pelo menos dentro da mesma semana. Embora não faça questão por muita documentação, a empresa contratante solicita que o Scrum Master envie um relatório semanal do progresso das atividades do projeto. Os dados de entrada para o modelo desse projeto são apresentados na Tabela 3.17.

Para esse projeto, espera-se que as entregas não satisfaçam o cliente, pela falta de interação do Product Owner com a equipe, que resulta em um Backlog do produto mal definido. As saídas do modelo são apresentadas na Figura 3.28. Analisando as saídas do mesmo,

Tabela 3.17: Evidências para o cenário 11

Nó	Evidência
Adaptação	Alto
Inspeção	Moderado
Qualidade do Product Owner	Baixo
Detalhado	Baixo
Estimado	Baixo
Ordenado	Baixo
Desempenho passado	Alto
Capacidade projetada	Alto
Backlog da sprint	Baixo
Objetivo da sprint	Baixo
Autonomia	Alto
Conhecimento	Alto
Distribuição	Moderado
Canais	Alto
Presença	Moderado
Sincronização	Alto
Qualidade do processo de desenvolvimento e testes	Moderado

pode-se concluir que os valores calculados estão em conformidade com os esperados.

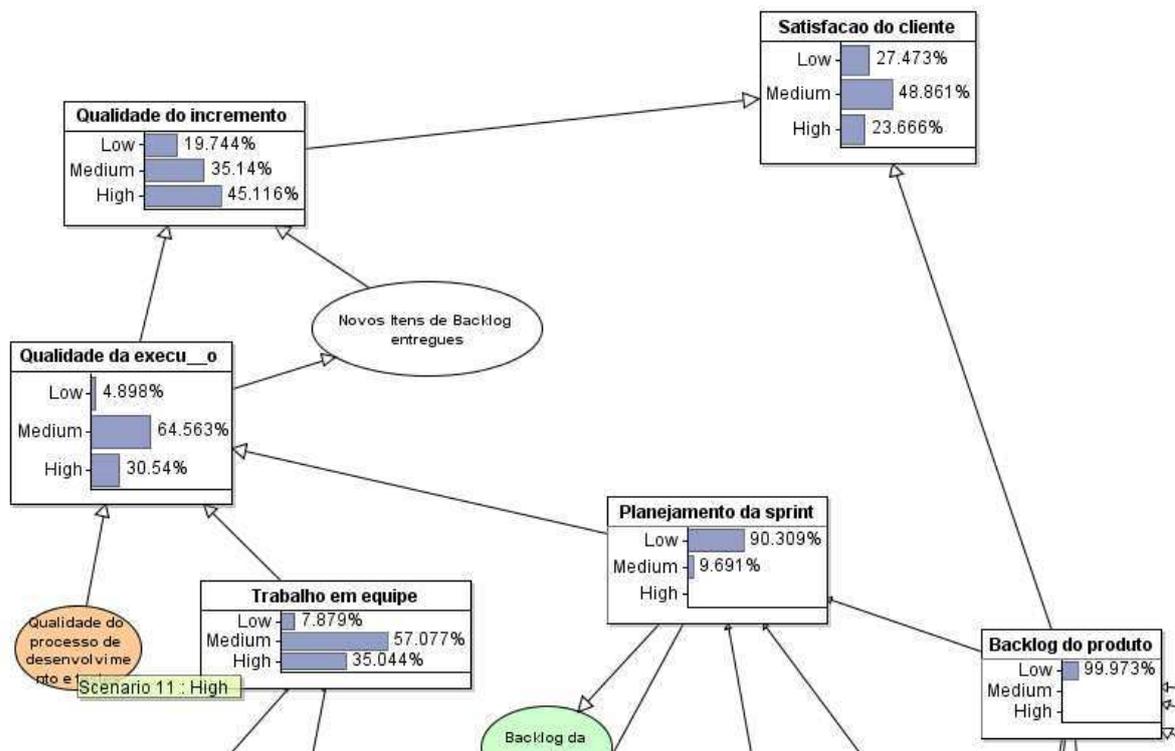


Figura 3.28: Resultados calculados para o cenário 11. Imagem extraída do AgenaRisk.

3.5.12 Cenário 12

Uma empresa especializada em desenvolvimento de software utiliza Scrum para clientes de vários segmentos. Um destes clientes deseja que um de seus principais sistemas seja reimplementado com novas tecnologias que possibilitam recursos de portabilidade e acessibilidade. A base de dados deve ser mantida, e o software antigo deve “morrer” em 01 ano e o novo ser lançado neste momento. Por imaginar ser uma demanda simples, uma equipe apenas com desenvolvedores iniciantes foi alocado para o projeto. O cliente não pretende se reunir com o time frequentemente para discutir requisitos, uma vez que para ele “é só fazer o mesmo que já existe hoje”. Os dados de entrada para o modelo desse projeto são apresentados na Tabela 3.18.

Para esse projeto, espera-se que o escopo seja entregue de forma incompleta. Apesar do projeto não necessitar da presença contínua do cliente, a equipe não tem competência para entregar o escopo no período pretendido. Isso deve gerar insatisfação do cliente. As

Tabela 3.18: Evidências para o cenário 12

Nó	Evidência
Qualidade do Product Owner	Baixo
Detalhado	Alto
Estimado	Alto
Ordenado	Alto
Definição inicial do Backlog do produto	Alto
Complexidade dos requisitos	Alto
Inovação dos requisitos	Alto
Ambiente de negócio	Alto
Desempenho passado	Alto
Capacidade projetada	Alto
Backlog da sprint	Baixo
Objetivo da sprint	Baixo
Autonomia	Baixo
Adaptabilidade	Baixo
Compartilhamento da liderança	Baixo
Conhecimento	Baixo
Distribuição	Baixo
Canais	Baixo
Presença	Moderado
Sincronização	Baixo
Qualidade do processo de desenvolvimento e testes	Moderado

saídas do modelo são apresentadas na Figura 3.29. Analisando as saídas do mesmo, pode-se concluir que os valores calculados estão em conformidade com os esperados.

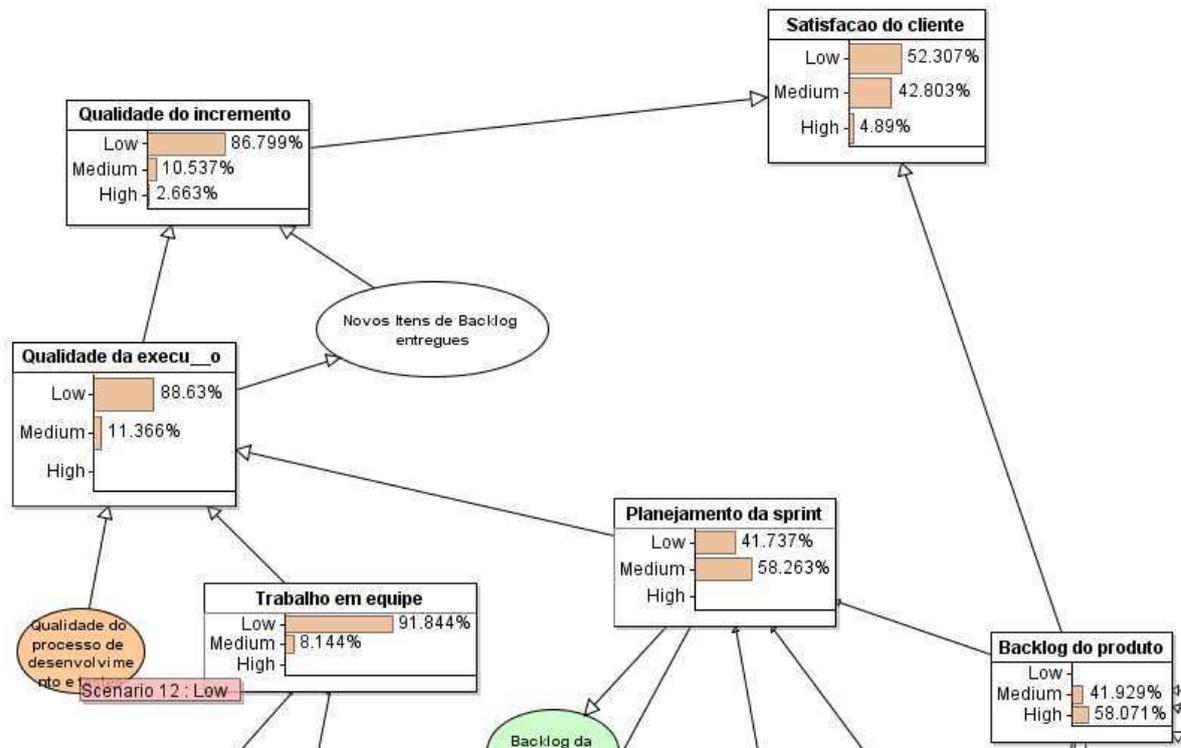


Figura 3.29: Resultados calculados para o cenário 12. Imagem extraída do AgenaRisk.

3.6 Ameaças à validade

Com relação à construção da rede Bayesiana, há três tipos de ameaça à validade: interna, construção e conclusão. Com relação à ameaça interna, destaca-se o problema de *model selection*. Apesar da consulta na literatura e com especialistas, não há como garantir que outros especialistas, ao construir um modelo com o mesmo escopo, selecionasse as mesmas variáveis, estrutura e TPN. Além disso, as premissas com relação aos parâmetros μ e σ^2 da distribuição Normal duplamente truncada utilizada para modelar as causas entre os nós podem não ser ótimas. Como consequência, tem-se a ameaça de conclusão: os cenários de simulação utilizados para validar a rede Bayesiana podem não ser o suficiente para garantir que as previsões do modelo são corretas. Essa ameaça é minimizada com o uso de doze cenários definidos por especialistas com vasta experiência em projetos de software e com a detecção dos anti-padrões apresentados em Eloranta et al. [60].

A ameaça de conclusão é resultado da falta de resultados empíricos comprovando a causalidade entre os fatores da rede Bayesiana. Desta forma, não se pode garantir que as entidades são mensuradas corretamente pelo modelo (i.e., operacionalizações). Para minimizar esta ameaça, o conhecimento de especialistas com vasta experiência em projetos de software foi elicitado para a definição do modelo.

Capítulo 4

Processo Proposto

O processo proposto é baseado na utilização de redes bayesianas para dar suporte ao diagnóstico e prognóstico do método de desenvolvimento de projetos Scrum. O mesmo deve estar de acordo com os princípios do Manifesto Ágil e regras do Scrum. Ou seja, o mesmo deve ser utilizado como um instrumento usado pela equipe para auxiliar na sua melhoria contínua, respeitando o princípio de que “Indivíduos e interações mais que processos e ferramentas”. Além disso, o mesmo deve prever a adaptação para as necessidades de equipes Scrum em diferentes contextos.

O processo é baseado no método apresentado em Perkusich et al. [159], sendo cíclico, contínuo e composto de quatro passos: 1) Adaptação da rede Bayesiana; 2) Observação da equipe; 3) Avaliação do método de desenvolvimento; e 4) Execução de ações corretivas e preventivas. Uma visão geral do processo é apresentada na Figura 4.1. A duração dos ciclos deve ser alinhada com as iterações, mas não precisa ser a mesma. O importante é que o passo 3 seja realizado durante uma Reunião de Retrospectiva. O esforço adicional para a aplicação do método ocorre durante o passo 1, dado que o passo 3 pode ser considerado como um complemento ou técnica adicional utilizada na Reunião de Retrospectiva, e os passos 2 e 4 fazem parte do dia-a-dia do Scrum Master.

Para o passo 1, assume-se que as práticas e processos para complementar o Scrum foram definidas. Além disso, assume-se que as métricas adequadas para o projeto foram definidas. Por exemplo, de acordo com os requisitos de segurança, propósito do software e linguagem de programação, diferentes métricas são adequadas. Inclusive, modelos de métrica propostos na literatura tais como o *Quacomo* [204], para medir a qualidade do produto, podem ser

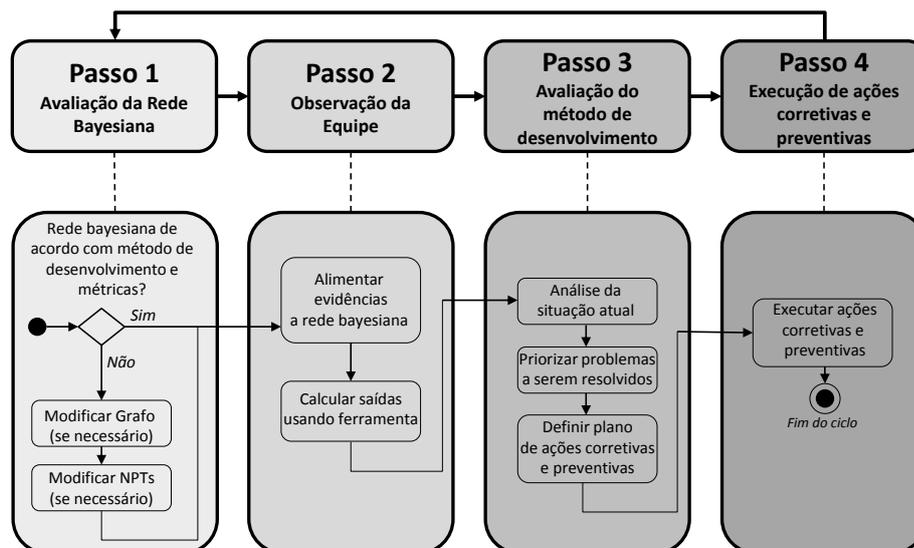


Figura 4.1: Visão geral do processo para avaliação da aplicação do Scrum.

utilizados. As mesmas devem estar de acordo com a Definição de Pronto do projeto. De acordo com estas definições, a rede Bayesiana pode ser adaptada, como apresentado na Seção 3.3. Recomenda-se que as adaptações apenas ocorram nos *hot spots* da rede Bayesiana, como apresentado no Capítulo 3. Além disso, recomenda-se que ao menos um representante de cada papel esteja presente durante a execução das atividades desse passo do processo para ter diferentes perspectivas.

Além disso, no passo 1, os fatores da rede Bayesiana devem ser associados a fontes de dados. As fontes de dados podem ser um questionário ou ferramentas (e.g., Jenkins). Cada variável da fonte de dado em questão pode ser modelada como um indicador na rede Bayesiana (ver Seção 3.3). Por outro lado, múltiplas variáveis das fontes de dados podem ser associadas a um nó da rede Bayesiana. Cada variável (i.e., pergunta) pode ser mapeada para um nó indicador filho do nó em questão. Outra opção é mapear as respostas das perguntas diretamente com o nó em questão por meio de regras de mapeamento. Por exemplo, as perguntas abaixo podem ser mapeadas para o nó *Autonomia*, onde o usuário responde “Sim” ou “Não”.

1. Os membros da equipe precisam trabalhar em atividades que eles acreditem que não adicionem valor?
2. A equipe é interrompida durante a sprint por agentes externos e precisa trabalhar em

atividades que não agregam valor ao objetivo da sprint atual?

Para esse caso, dado que o nó em questão é composto de 3 estados (*Baixo*, *Moderado* e *Alto*), pode-se definir as seguintes regras de mapeamento das respostas:

- Se todas as respostas forem “Não”, mapear para o valor “Baixo”;
- Se apenas uma das respostas for “Não”, mapear para o valor “Moderado”;
- Se todas as respostas forem “Sim”, mapear para o valor “Alto”;

Além disso, recomenda-se definir as perguntas e possíveis respostas de forma objetiva [99] seguindo o contexto da empresa. Por exemplo, para o nó *Conhecimento*, pode-se definir a pergunta: *Qual a média experiência da equipe com projetos Scrum?* e utilizar as opções de respostas abaixo.

- “Menos de 2 anos”, mapear para o valor “Baixo”;
- “Entre 2 anos e 5 anos”, mapear para o valor “Moderado”
- “Mais de 5 anos”, mapear para o valor “Alto”;

Para o caso do uso de questionário, recomenda-se utilizar mais de uma pergunta por nó para aumentar a confiabilidade da consistência interna. Por outro lado, o esforço para coleta de dados é maior. Então, fica a cargo do desenvolvedor do modelo realizar a avaliação da solução adequada, juntamente com a organização em questão.

Para o caso do uso de métricas quantitativas (e.g., cobertura de testes), as mesmas devem ser associadas a um nó indicador e, como apresentado na Seção 3.3, pode-se utilizar expressões particionadas para mapear os valores da mesma com o da entidade do método em questão. Como poucas ferramentas de redes Bayesianas tem suporte a nós quantitativos, uma alternativa é realizar um mapeamento da faixa de valores da métrica em questão com a escala ordinal. Por exemplo, para a métrica (e.g., cobertura de testes) pode-se utilizar o mapeamento abaixo:

- “Abaixo de 50%”, mapear para o valor “Baixo”;
- “Entre 50% e 80%”, mapear para o valor “Moderado”

- “Acima de 80%”, mapear para o valor “Alto”;

Além das possibilidades de complementação da rede Bayesiana com métricas e práticas, como apresentado na Seção 3.3, adaptações podem ser realizadas na mesma para se adequar ao contexto do projeto. Por exemplo, muitas empresas, apesar de usarem o Scrum, mantêm uma equipe externa de qualidade para executar testes independentes e auditoria de processos. Nesse caso, dado que uma das premissas do modelo é que a equipe é multifuncional, para melhorar o diagnóstico e previsão, pode-se adicionar o nó *Qualidade dos testes independentes* como pai do nó *Qualidade do incremento*. Esse nó deve representar o nível de confiança no valor do nó filho e pode ser implementado utilizando expressões particionadas. Preditores, tais como *Experiência dos testadores*, *Qualidade dos casos de teste* e *Qualidade das ferramentas de teste*, podem ser adicionados como pai do nó *Qualidade dos testes independentes*. Nós indicadores, tais como *Vazamento de defeitos em operação*, podem ser associados ao mesmo.

Outro exemplo de adaptação é no caso de equipes que utilizam, além do Backlog do produto, uma ferramenta de *bug tracking* para registrar defeitos que vazaram das sprints passadas [202]. Ou seja, o escopo da sprint pode ser composto de itens de Backlog do produto e/ou defeitos vindos do *bug tracker*. Dado que uma das premissas do modelo é que não há débito técnico relevante no início da iteração, pode-se adicionar o nó *Escopo da sprint* do tipo ranqueado, no qual o usuário pode definir a proporção de itens do Backlog da sprint associados à correção de defeitos e novas funcionalidades. Espera-se que em sprints com escopo focado na correção de defeitos, menos novas funcionalidades serão entregues, mas a qualidade do incremento deve melhorar. Por outro lado, em sprints com escopo focado no desenvolvimento de novas funcionalidades, espera-se que mais funcionalidades sejam adicionadas ao incremento. Esse comportamento pode ser modelado por meio do uso de expressões particionadas.

Além disso, para os nós de entrada (i.e., sem nós pai), ao invés de utilizar uma distribuição uniforme para definir a TPN como apresentado na Seção 3.2, pode-se elicitare dos membros da organização a frequência esperada (i.e., priors) para cada estado. Os priors também podem ser definidos de acordo com o histórico da empresa, caso haja dados disponíveis. Por exemplo, para o nó *Qualidade do Product Owner*, pode-se definir a TPN apresentada na Tabela 4.1. Uma definição de probabilidades relativas realistas pode melhorar a previsão do

Tabela 4.1: Exemplo de definição de priors para o nó *Qualidade do Product Owner*.

Estado	Probabilidade
Baixo	0,2
Moderado	0,5
Alto	0,3

modelo nos casos com indisponibilidade de dados.

Para os demais nós, assume-se que os tipos de funções definidos para as expressões ponderadas são adequados para contextos diferentes. Por outro lado, recomenda-se avaliar se os pesos dos nós pai estão adequados para o contexto da organização (i.e., projeto) em questão.

No passo 2, a equipe Scrum (incluindo o Product Owner) é observada e os artefatos são avaliados pelo Scrum Master. Além disso, as métricas são coletadas. Como resultado, os nós de entrada da rede Bayesiana são alimentados. O ideal é que todos os nós de entrada da rede Bayesiana sejam atualizados com evidências. Para as entradas que não puderem ser verificadas, as mesmas devem ser alimentadas com o mesmo valor para todos os estados possíveis (i.e., uma distribuição uniforme representando incerteza máxima).

O passo 3 deve ser executado durante uma Reunião de Retrospectiva. Neste passo, é necessário calcular as saídas da rede Bayesiana. Para tal, utiliza-se o AgenaRisk¹, pois é a única ferramenta no mercado com suporte a nós ranqueados [69]. As saídas calculadas da rede Bayesiana devem ser analisadas para a equipe se autoavaliar e discutir como melhorar. No contexto do modelo proposto, podem haver modificações nas práticas de desenvolvimento ou gerenciamento escolhidas para complementar o Scrum, correção na execução de práticas existentes ou correção nas métricas selecionadas. Além disso, podem haver atualizações na Definição de Pronto da equipe.

O objetivo desta etapa está alinhado com um dos principais objetivos das Reuniões de Retrospectiva: definir ações corretivas e preventivas que devem ser colocadas em prática para alcançar os objetivos de negócio do projeto. Desta forma, esta etapa pode ser considerada como um complemento ou técnica adicional na execução de Reuniões de Retrospectiva.

¹www.agenarisk.com

Além disto, dado que se trata de um modelo causal, é possível analisar o impacto na qualidade do método de desenvolvimento caso algumas entradas sejam modificadas. Esta característica do modelo pode auxiliar na priorização das ações corretivas e preventivas. Por exemplo, dado que a equipe identifique cinco fatores que precisem de ações, esses fatores podem ter seus valores alterados, isoladamente, no modelo para que o impacto na qualidade do incremento possa ser calculado. Por fim, um plano de execução das ações deve ser criado. Uma alternativa para priorizar as ações é realizar análise de sensibilidade. Como saída da Reunião de Retrospectiva, é esperado ter um Backlog priorizado de ações com um subconjunto do mesmo alocado para a sprint seguinte. De acordo com o Guia do Scrum [193], pelo menos 10% do escopo de uma sprint deve ser de itens definidos na Reunião de Retrospectiva.

Finalmente, no passo 4, o Scrum Master é o responsável por liderar os ajustes (i.e., ações de correção ou melhoria) definidos no passo 3. Um exemplo de atividade realizado neste passo pode ser treinar a equipe em alguma prática ágil (e.g., TDD) ou providenciar alguma ferramenta para automatizar a coleta de métricas (e.g., Findbugs [11]). Ou seja, as ações definidas no passo 3 devem ser executadas ainda durante o projeto, sempre que possível, na sprint seguinte. Desta forma, se como resultado do passo 3 o método de desenvolvimento ou as métricas tiverem sido modificados, o passo 1 deve ser executado em seguida. Caso contrário, o passo 2 é executado. Dado que utilizando ou não o método proposto nesta pesquisa, um dos objetivos das reuniões de retrospectiva é definir um plano de ações corretivas e preventivas e o Scrum Master é responsável por liderar a execução do plano de ações, a execução desse passo não adiciona esforço no dia-a-dia do Scrum Master. Além disso, se as ações corretivas contiverem adaptações no método de desenvolvimento ou métricas, o passo 1 deve ser executado para adaptar o mesmo.

Os custos associados à implantação do método decorrem da necessidade de familiarização dos usuários com o processo e o modelo. Como citado anteriormente, a complexidade técnica para utilizar a rede Bayesiana é abstraída dos usuários, ou seja, não influencia nos custos. Por outro lado, é necessário treinamento para cada etapa do processo. Além disso, é necessário conhecimento suficiente para compreender as entradas e saídas do modelo, para que o mesmo seja alimentado e interpretado corretamente. Dado que o modelo pode ser modificado, é necessário conhecimento suficiente para tal.

Dado que a rede Bayesiana é composta de dados que avaliam as pessoas envolvidas

no projeto, alguns cuidados devem ser tomados ao se utilizar esse processo. Dado que há informações relativas ao desempenho da equipe, o Scrum Master deve ter sensibilidade ao coletar dados para alimentar o modelo e ao apresentar os dados coletados e calculados pelo mesmo para evitar constrangimentos e conflitos. A limitação do processo é que o mesmo apenas considera projetos Scrum compostos por apenas uma equipe de desenvolvimento.

Capítulo 5

Avaliação da solução

A avaliação da solução foi realizada em duas etapas: (i) utilidade prática e (ii) acurácia das predições. A utilidade prática está relacionada ao grau de utilidade do processo para auxiliar na melhoria contínua do método de desenvolvimento. Os resultados do estudo de caso são apresentados na Seção 5.1; os resultados da acurácia das predições, na Seção 5.2.

5.1 Avaliação da utilidade prática do processo

Para avaliar a utilidade prática da solução, a mesma foi aplicada em um contexto real por meio de um estudo de caso em dois projetos de uma empresa brasileira utilizando a primeira versão da rede Bayesiana, apresentada em Perkusich et al. [155]. Outro estudo de caso foi realizado para avaliar a utilidade do processo utilizando-o apenas com o fragmento de *Trabalho em equipe*. Como resultado, foi concluído que o custo-benefício de aplicar o processo é positivo e os resultados são apresentados em Freire et al. [76]. A apresentação dos seus dados está fora do escopo deste documento.

O estudo de caso é do tipo embarcado, no qual cada projeto é considerado uma unidade de análise. A mesma, na época da realização do estudo, tinha mais de 3 anos de experiência aplicando o Scrum na maioria dos seus projetos. A duração do estudo foi 45 dias (3 sprints de 15 dias cada). Essa duração foi definida de acordo com o número mínimo de ciclos julgado como suficiente para responder às questões de pesquisa e a disponibilidade da empresa.

O objetivo do estudo foi analisar a solução proposta com relação a seu custo-benefício do ponto de vista da equipe de desenvolvimento no contexto do auxílio à tomada de decisões

de melhoria contínua da mesma. Para tal, foram definidas as seguintes questões de pesquisa:

1. **QP1** *O processo auxilia na identificação de problemas no método de desenvolvimento na aplicação do Scrum?*
2. **QP2** *O custo para alterar o modelo de forma a melhorar a representação do Scrum para um projeto específico é aceitável?*
3. **QP3** *O processo é útil para auxiliar a equipe na sua busca por excelência?*

As proposições do estudo são respostas positivas para cada questão de pesquisa, indicando que o custo-benefício do processo proposto é positivo.

Cada projeto é composto por uma equipe de desenvolvimento. O escopo de um dos projetos é desenvolver um serviço na nuvem e um cliente web (Projeto A). Esse projeto é composto por seis desenvolvedores. O escopo do outro projeto (Projeto B) é desenvolver clientes em iOS, Android e sistemas embarcados para o serviço web desenvolvido pela equipe do outro projeto. Esse projeto é composto por cinco desenvolvedores.

Antes do estudo de caso, ambos os projetos realizavam Reuniões de Retrospectiva, com duração de 1 hora, ao final de cada sprint com o objetivo das equipes se inspecionarem e se adaptarem. As reuniões eram estruturadas utilizando três etapas: (i) coleta de dados, (ii) autoavaliação, e (iii) definição do plano de melhoria [52]. Diversas técnicas para coleta de dados eram utilizadas tais como: *Triste Bravo Feliz*, *Histograma de satisfação*, e *Radar da equipe*. Para autoavaliação, técnicas como *Espinha de peixe*, *Brainstorming*, e *Os cinco porquês*. Para a definição do plano de melhoria, geralmente, baseado em propostas de ações definidas previamente, cada membro estimava seu valor e os itens de maior valor eram alocados para a próxima sprint.

O método de coleta de dados para responder às questões de pesquisa foi um questionário. No caso, foram definidas quatro perguntas: uma para cada questão de pesquisa e outra perguntando diretamente a opinião do sujeito sobre o custo-benefício do processo proposto. Cada pergunta foi respondida por meio de uma escala Likert de quatro pontos e espaço para o sujeito justificar e evidenciar as suas respostas.

Para cada projeto, os dados foram coletados dos *Scrum Masters* dos mesmos. Ambos os sujeitos tinham 8 anos de experiência em projetos de software; 5 anos com projetos Scrum.

Para executar o estudo, o AgenaRisk foi utilizado para executar os cálculos da rede Bayesiana. De acordo com restrições de licença, todos os modelos foram executados na máquina do pesquisador. O procedimento de execução foi dividido em três fases. O mesmo foi aplicado para cada unidade de análise.

Fase de treinamento

Durante esta fase, os sujeitos foram treinados para aplicar o processo proposto para avaliar a adoção do Scrum. Desafios, dificuldades e primeiras impressões foram coletados para avaliar o entendimento dos conceitos do processo e se o sujeito estava preparado para avançar para a próxima fase. Para tal, as seguintes atividades foram executadas:

Curso introdutório - foi realizado um treinamento por meio de apresentação audiovisual (e.g., demonstrações e *slides*) para ensinar conceitos básicos para aplicar o processo tais como métricas no contexto de ASD e redes bayesianas. Além disso, detalhes do processo foram apresentados. Algumas questões foram abordadas: “Como definir indicadores?”, “Como funciona uma rede Bayesiana?”, “Como alimentar a rede Bayesiana?”, “Como interpretar os dados calculados da rede Bayesiana?” e “Como utilizar os dados para liderar melhorias no método de desenvolvimento?”. O curso teve duração de 1 hora.

Aprender fazendo - com auxílio do treinador (i.e., pesquisador), o sujeito, baseado no contexto do projeto em questão, executou os passos de um ciclo do processo utilizando dados existentes. Foram registrados os questionamentos e dúvidas que surgiram ao executar a atividade e o tempo gasto. Essa atividade teve duração de 1 hora.

Fase de execução do processo

Durante essa fase, os passos do processo proposto apresentado no Capítulo 4 foram executados. Um ciclo completo do processo foi executado a cada sprint, para ambas as unidades de análise.

5.1.1 Dados da Execução do Estudo de Caso

Inicialmente, para executar o passo 1, uma reunião com cada *Scrum Master* foi executada e, para ambas as equipes foi definido que a rede Bayesiana base seria utilizada sem modificações. Além disso, como fonte de dados, foi definido um questionário composto de uma questão por nó indicador e de entrada. Na mesma reunião, o questionário foi respondido com os dados da sprint corrente.

Com o modelo de cada projeto alimentado com os dados da sprint em questão, os resultados foram calculados e apresentados nas Reuniões de Retrospectiva (passo 3), que não sofreram modificações nas suas durações (i.e., tinham duração de 1 hora por sprint). Durante as reuniões, os resultados dos modelos foram analisados para identificar as oportunidades de melhoria dos projetos com o intuito de definir ações corretivas e preventivas para aumentar a probabilidade de sucesso desses. Dessa forma, os resultados do modelo serviram como uma fonte a mais de informação para facilitar na identificação de oportunidades de melhorias, que é um dos objetivos das Reuniões de Retrospectiva. Com as informações coletadas por meio da análise do modelo, oportunidades de melhoria foram identificadas e uma lista com ações corretivas e preventivas foi definida. Em ambos os projetos, alguns pontos de ações já tinham sido identificados no passado, mas por diversos motivos, não tinham sido colocados em prática.

Para o projeto A, foram identificadas oportunidades de melhoria relacionadas às seguintes entidades do método de desenvolvimento: *Monitoramento do progresso*, *Detalhamento do Backlog do produto*, *Desempenho passado*, *Qualidade do Product Owner*, *Conhecimento*, *Processo de desenvolvimento e testes* e *Compartilhamento da liderança*.

Para o projeto B, foram identificadas oportunidades de melhoria relacionadas às seguintes entidades do método de desenvolvimento: *Monitoramento do progresso*, *Detalhamento do Backlog do produto*, *Desempenho passado*, *Qualidade do Product Owner*, *Comunicação*, *Processo de desenvolvimento e testes* e *Compartilhamento da liderança*.

Para ambas as equipes o problema com *Desempenho passado* foi causado por mudanças nas equipes. Como foi previsto que não haveria mais mudanças, esse problema foi descartado. Além disso, foi apontado que alguns dos problemas detectados por meio do uso da rede Bayesiana já tinham sido discutidos antes, tais como a necessidade de melhorias no processo de desenvolvimento e testes por meio de melhor uso de ferramentas de análise estática, aumento da cobertura de testes e automação de testes.

Depois de ter definido a lista de oportunidades de melhoria, foi desenvolvido um plano de execução das ações. Para tal, o modelo foi utilizado para avaliar o impacto no projeto de ter cada ação executada e, dessa forma, priorizar a lista de ações. Para avaliar o impacto no projeto de ter cada ação executada, as informações referentes a cada ação foram modificadas no modelo com o intuito de prever as consequências da execução dessas na probabilidade de

sucesso do projeto. As ações foram priorizadas com relação à magnitude do impacto de suas respectivas execuções na probabilidade de sucesso do projeto. Por exemplo, para o Projeto A, caso o *Monitoramento do progresso* fosse resolvido, a probabilidade de sucesso teria um aumento de 3,8%.

Após ter a lista de ações priorizada, os participantes das reuniões definiram quais ações seriam colocadas em prática durante a execução da próxima sprint. No caso do projeto A, foi definido utilizar a prática de *Burndown* para melhorar o *Monitoramento do progresso*. Para o projeto B, além da prática de *Burndown*, a equipe (incluindo o Product Owner) decidiu melhorar o *Detalhamento do Backlog do produto* melhorando a descrição dos requisitos não-funcionais.

Durante a segunda sprint, os Scrum Masters lideraram suas respectivas equipes para colocarem em prática os planos de ações definidos. Essa atividade faz parte do dia-a-dia de um Scrum Master independente de utilizar o processo proposto neste trabalho. Ao final da sprint, antes das Reuniões de Retrospectiva, os Scrum Masters se reuniram com o Product Owner e um líder da equipe de desenvolvimento de seus respectivos projetos para analisar e alimentar novamente o modelo para refletir as mudanças que ocorreram desde o final sprint anterior (passo 2). Cada reunião foi limitada em quinze minutos, e esse tempo foi o suficiente para tal.

Durante as Reuniões de Retrospectiva da segunda sprint, os modelos atualizados, assim como o resultado da execução dos pontos de ação definidos ao final da sprint passada foi apresentado e discutido. Como resultado da análise do modelo e discussão, novos pontos de ações foram identificados. Um novo problema detectado para ambos os projetos foi com relação à *Autonomia*, pois durante a sprint passada, os clientes requisitaram mudanças no escopo da sprint durante a mesma, que precisaram ser atendidas. Dessa forma, *Autonomia* foi adicionado no Backlog de melhoria de ambas as equipes. A equipe A decidiu melhorar o *Processo de desenvolvimento e testes* por meio do uso de ferramentas de análise estática. Para tal, foi definido um *spike* (i.e., reserva de tempo) para a equipe pesquisar ferramentas. Além disso, decidiu que deveria reservar parte da capacidade da equipe na sprint seguinte para corrigir todos os defeitos legados do sistema. A equipe B também decidiu melhorar o *Processo de desenvolvimento e testes*, mas por meio do aumento da cobertura de testes.

Durante a terceira sprint, o mesmo processo foi repetido. Durante a Reunião de Retros-

pectiva, a equipe A, depois de corrigir os defeitos legados e definir ferramentas de análise estática para as tecnologias utilizadas no projeto, decidiu adicioná-las como parte do processo de revisão de código por pares. Com isso, espera-se aumentar a eficiência da equipe. A equipe B decidiu continuar a trabalhar no aumento da cobertura de testes.

5.1.2 Resultados e Discussão do Estudo de Caso

Ao final do estudo, para o projeto A, a rede Bayesiana calculou um aumento de 7,4% na probabilidade de sucesso do projeto. Para o projeto B, 5,1%. Ao discutir com as equipes durante a última Reunião de Retrospectiva, elas concordaram que os valores calculados representam a realidade. Além disso, dado que o processo foi executado em equipes maduras, concluiu-se que os resultados demonstram a utilidade do processo.

Além disso, como discutido anteriormente, os Scrum Masters de cada equipe responderam um questionário. Os dados coletados são apresentados na Figura 5.1.

Observando os dados coletados, pode-se concluir que todas as proposições do estudo foram validadas, ou seja, o processo proposto é útil com custo-benefício positivo. No caso da robustez do modelo para identificar problemas na aplicação do Scrum, ambos Scrum Masters afirmaram que a aplicação do método os ajudou a identificar pontos de ações que tinham passado despercebidos. Isso confirma que o método auxilia na detecção de problemas. Além disso, um dos Scrum Masters afirmou que a aplicação do método identificou pontos de ações que já tinham sido identificados pela equipe. Isso evidencia que os problemas identificados são válidos.

Os Scrum Masters afirmaram que não precisaram modificar muito o modelo para utilizá-lo nas suas respectivas equipes. Isso segue o objetivo dessa pesquisa de criar um modelo genérico que pudesse ser aplicado para qualquer projeto Scrum de desenvolvimento de software sem necessitar de muitas modificações. Além disso, segundo os Scrum Masters, o esforço necessário para realizar as modificações necessárias e alimentar o modelo é compensado pelos resultados obtidos pela utilização do método.

1. O método é robusto para identificar problemas na aplicação do Scrum?			
Sim	Precisa de algumas melhorias	Precisa de muitas melhorias	Não
2	0	0	0
Comentários			
Foi capaz de identificar pontos que já tinham chamado nossa atenção no passado e alguns que tinham passados despercebidos.			
Conseguimos identificar alguns problemas que tinham passados despercebidos e que eram relevantes.			
2. O custo para alterar o modelo de forma a melhorar a representação do Scrum para um projeto específico é aceitável?			
Sim	Precisa de algumas melhorias	Precisa de muitas melhorias	Não
2	0	0	0
Comentários			
Não precisamos modificar muito o modelo.			
No nosso caso, o modelo não precisou de muitas modificações.			
3. O método é útil para auxiliar o ScrumMaster a guiar a equipe na busca por excelência?			
Sim	Precisa de algumas melhorias	Precisa de muitas melhorias	Não
2	0	0	0
Comentários			
A análise e alimentação do modelo ajudam o ScrumMaster a ter uma visão geral do projeto e prestar atenção em pontos que podem passar despercebidos, principalmente para ScrumMasters inexperientes. Além disso, a utilização do modelo nas reuniões de retrospectiva ajudam na identificação de oportunidades de melhoria.			
É uma ferramenta adicional que acredito que pode auxiliar na identificação de problemas e pontos de melhoria. Acredito que o método pode ser útil para ScrumMasters.			
4. O custo-benefício de usar o método é positivo?			
Sim	Precisa de algumas melhorias	Precisa de muitas melhorias	Não
2	0	0	0
Comentários			
Os únicos custos adicionais referentes à aplicação do método referem-se ao tempo necessário para entender, modificar e editar o modelo. Dado o tempo que o método foi avaliado no projeto que trabalhei, verificou-se que esse tempo é compensado pelo auxílio que o modelo teve para facilitar a identificação de oportunidades de melhoria.			
O esforço adicional necessário pra utilizar o método resume-se a alguns minutos por sprint pra analisar e alimentar o modelo. No nosso caso, esse esforço foi compensado pelos pontos de ações que foram identificados nas reuniões de retrospectiva.			

Figura 5.1: Dados coletados no estudo de caso

5.1.3 Ameaças à validade

Com relação às ameaças à validade desse estudo de caso, destacam-se três: de construção, externa e interna. A ameaça de construção refere-se ao risco das opiniões dos Scrum Masters não serem suficientes para responder às questões de pesquisa. A validade externa refere-se à incapacidade de generalização. Dado que o estudo é realizado em apenas dois projetos de uma empresa, não se deve confiar que o estudo de caso valide o modelo para ser utilizado por qualquer empresa que utiliza Scrum em projetos de desenvolvimento de software. Por outro lado, como discutido anteriormente, em outro estudo [76], o mesmo processo foi utilizado tendo como base o fragmento de *Trabalho em equipe* em outra empresa com resultados satisfatórios. A ameaça interna refere-se ao risco de o processo não ter sido o responsável pela melhora das equipes. Por outro lado, dado que o processo foi avaliado em duas equipes maduras que já realizavam Reuniões de Retrospectiva utilizando boas práticas da indústria e os membros da equipe relataram benefícios na utilização da solução proposta, essa ameaça foi minimizada.

5.2 Avaliação da acurácia da predição

Para avaliar a acurácia da predição da rede Bayesiana, como recomendado em Mendes [131], dados foram coletados de 18 projetos de uma empresa brasileira de desenvolvimento de software. Para cada projeto, foi coletado dado apenas da sprint mais recentemente finalizada. O instrumento para coleta de dados foi um questionário, o qual foi respondido pelo Scrum Master do projeto em questão. A rede Bayesiana base apresentada no Capítulo 3, sem alterações, foi utilizada para realizar as predições. Para cada nó de entrada ou indicador, foi definida uma pergunta no questionário. Além disso, foi adicionada uma questão relativa ao nó alvo do estudo: *Satisfação do cliente*. Apenas uma questão foi definida para cada nó para minimizar o esforço para completar o questionário.

Antes de administrar o questionário, foi realizado um estudo piloto (i.e., pré-teste) com 11 pessoas para avaliá-lo do ponto de vista de clareza. Para realizar o pré-teste, cada pessoa respondeu o questionário separadamente. Depois, grupos de foco foram realizados para discutir as dificuldades encontradas e possíveis soluções. Ao final do processo, a versão definitiva do questionário foi definida.

Para cada nó a ser avaliado, foi criada uma definição de acordo com o contexto da empresa. Para formular a pergunta, foi perguntado se o respondente concorda que a definição em questão é verdadeira para o seu projeto. Por exemplo, para o nó *Autonomia* a definição foi: “A equipe não precisa trabalhar em atividades que não adicionem valor ao projeto em questão e nem sofre influência de agentes externos durante a sprint”. Dessa forma, a pergunta resultante foi: *Você concorda com a seguinte afirmação: “A equipe não precisa trabalhar em atividades que não adicionem valor ao projeto em questão e nem sofre influência de agentes externos durante a sprint”?*. As opções de respostas são *Discordo completamente*, *Concordo parcialmente* e *Concordo completamente*. As mesmas são mapeadas para os estados *Baixo*, *Moderado* e *Alto* no nó em questão da rede Bayesiana. Algumas perguntas são exceção, pois as mesmas foram realizadas de forma mais direta. Por exemplo para o nó *Complexidade dos requisitos*, a pergunta realizada foi “O quão complexos são os requisitos?”, com as opções de respostas *Alta*, *Moderada* e *Baixa*). Além disso, foram realizadas perguntas demográficas para caracterizar os sujeitos e os projetos em questão. O questionário completo é apresentado no Apêndice A. Todas as respostas, mapeadas para os nomes dos nós e estados da rede Bayesiana, são apresentadas no Apêndice B.

Para cada projeto, a rede Bayesiana foi alimentada com as respostas do Scrum Master às perguntas relacionadas aos nós de entrada ou indicador. Depois da execução da rede, o valor calculado para o nó *Satisfação do cliente* foi comparado com a resposta à pergunta relacionada ao mesmo. A predição é considerada correta se o estado de *Satisfação do cliente* calculado com a maior probabilidade for o mesmo da resposta do Scrum Master (e.g., o modelo calcula 60% de probabilidade para o estado *Alto* e o Scrum Master respondeu *Concordo completamente*).

Os respondentes têm em média 9,8 anos de experiência com projetos de software, dos quais 6 com projetos Scrum. Os projetos, em média, são compostos por 6,7 integrantes, incluindo testadores. Dos 18 projetos, 10 já estavam finalizados; 3 com mais de 85% de progresso, 1 com 58%, e os demais com aproximadamente 50%, onde o progresso foi calculado de acordo com a quantidade de sprints planejadas para o projeto. Na Figura 5.2, é ilustrada a distribuição de projetos por tamanho de código, mensurado em linhas de código. Na Figura 5.3, a frequência das linguagens de programação utilizadas nos projetos é apresentada.

Na Tabela 5.1, as respostas de todos os projetos referentes ao nó *Satisfação do cliente*

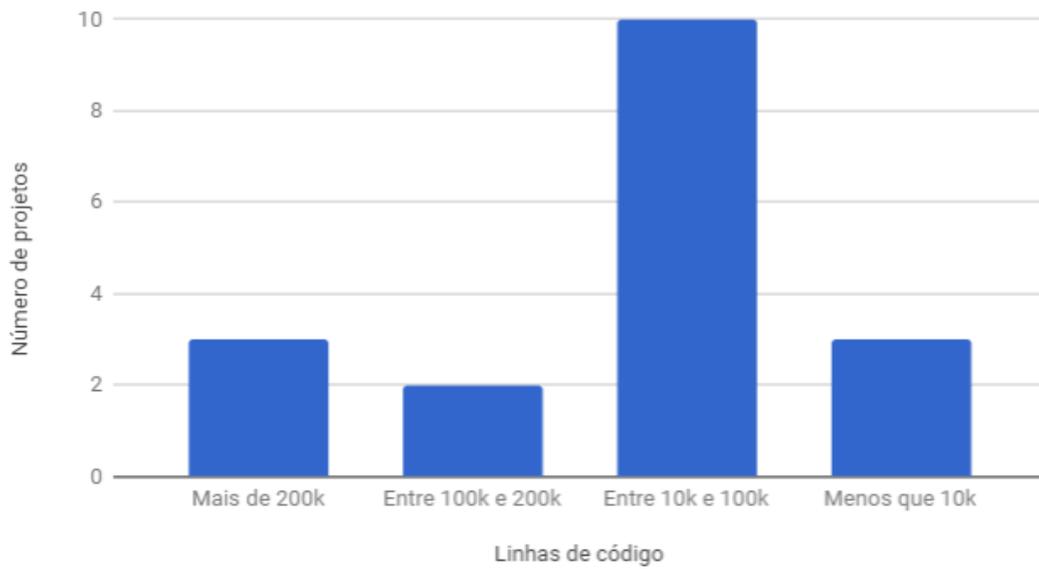


Figura 5.2: Quantidade de linhas de código dos projetos.

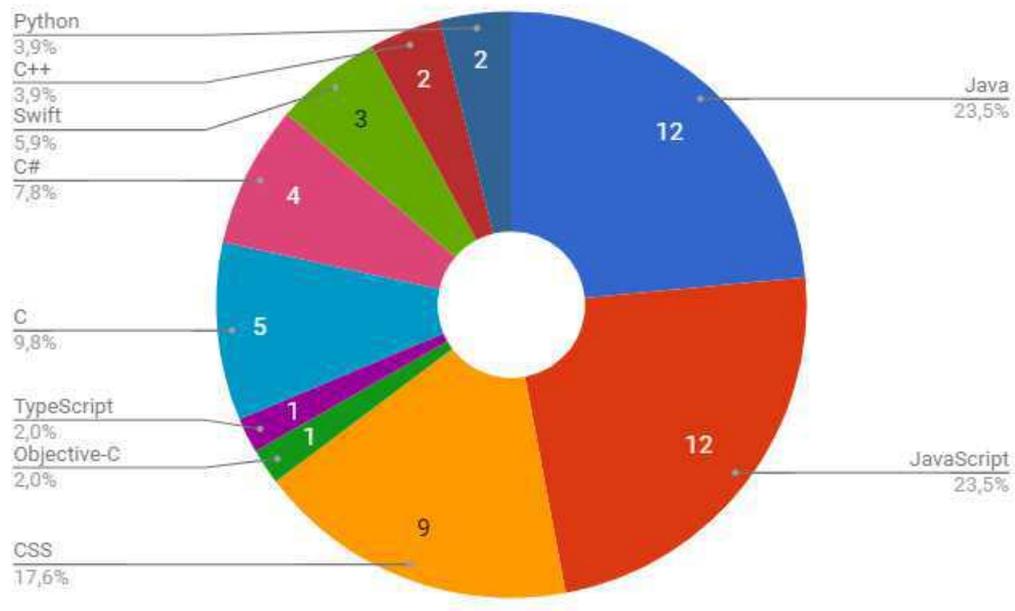


Figura 5.3: Linguagens de programação utilizadas nos projetos.

mapeados para o estado referente ao nó em questão e os resultados calculados do modelo são apresentados. Os valores marcados em negrito representam os de maior probabilidade para o projeto em questão.

Ao analisar os resultados do estudo, pode-se observar que as predições do modelo foram corretas para 14 dos 18 projetos, resultando em 78% de acurácia. Dos projetos com predição incorreta, 2 estão concluídos, 1 com 58% de progresso e 1 com 87%. Dado que o processo de adaptação para o contexto da empresa, devido a limitações de recursos fornecidos pela mesma, não foi o ideal, a acurácia poderia ter sido melhor. Dessa forma, conclui-se que o modelo é capaz de realizar previsões com acurácia satisfatória.

5.2.1 Ameaças à validade

As principais ameaças à validade desse estudo são de confiabilidade, externa, construção e interna. Com relação à confiabilidade, dado que os Scrum Masters responderam, em uma escala ordinal, questões referentes aos seus projetos, as respostas podem estar enviesadas. Em especial, para a avaliação da *Satisfação do cliente*, dado que a empresa não tem um processo formal para tal, os respondentes podem ter sido otimistas (89% dos respondentes afirmaram que os clientes estão satisfeitos). Para minimizar esse risco, foi explicitamente comunicado aos respondentes que todos os dados seriam anonimizados. Como ameaça de construção, há o risco de as perguntas não terem sido adequadas para indicar os valores dos nós da rede Bayesiana. Esse risco foi minimizado por meio do estudo piloto. Finalmente, dado que apenas uma pergunta foi definida para cada nó, não há como afirmar que as respostas foram consistentes. Dessa forma, é possível que não haja fiabilidade nos dados coletados.

A ameaça à validade externa refere-se à incapacidade de generalização. Dado que o estudo é realizado em apenas uma empresa, não se deve confiar que o estudo de caso valide o modelo para ser utilizado por qualquer empresa que utiliza Scrum em projetos de desenvolvimento de software. A ameaça à validade interna refere-se ao risco de outros fatores terem influenciado a satisfação do cliente, tais como contrato e relacionamento entre gestores. Dado que esses fatores não são modelados pela rede Bayesiana, os resultados podem estar imprecisos.

Tabela 5.1: Valores reais e previstos para *Satisfação do cliente*.

Projeto	Resposta	Probabilidades calculadas		
		Baixo	Moderado	Alto
1	Alto	0,130	0,429	0,441
2	Alto	0,212	0,459	0,329
3	Alto	0,137	0,426	0,441
4	Alto	0,121	0,405	0,474
5	Moderado	0,296	0,489	0,215
6	Alto	0,107	0,408	0,484
7	Alto	0,129	0,411	0,460
8	Alto	0,542	0,380	0,080
9	Alto	0,111	0,379	0,510
10	Alto	0,120	0,417	0,463
11	Alto	0,490	0,260	0,691
12	Alto	0,387	0,446	0,167
13	Alto	0,388	0,455	0,156
14	Alto	0,030	0,208	0,762
15	Alto	0,031	0,220	0,749
16	Alto	0,023	0,190	0,787
17	Alto	0,077	0,326	0,597
18	Moderado	0,272	0,479	0,249

Capítulo 6

Trabalhos Relacionados

Neste capítulo, o trabalho realizado nesta pesquisa é relacionado com abordagens apresentadas na literatura para auxiliar na adoção ou melhoria contínua de métodos ágeis.

No contexto de ASD, há diversas pesquisas propondo processos para guiar a adaptação contínua do método de desenvolvimento. Algumas propõem combinar métodos ágeis com o CMMI (Modelo de Maturidade em Capacitação - Integração). Em Selleri Silva et al. [184], resultados de uma revisão sistemática na literatura [110] explorando a combinação de métodos ágeis com CMMI são apresentados. Como resultado, foram encontrados 81 estudos e foi concluído que é possível alcançar os níveis 2 e 3 do CMMI aplicando métodos ágeis. Por outro lado, para alcançar os níveis 4 e 5, é necessário adicionar outras práticas.

Em outras pesquisas, abordagens específicas para ASD são propostas. Em Campanelli e Parreiras [36], foram apresentados resultados de uma revisão sistemática que explora a adaptação de métodos ágeis. Foram identificados 56 trabalhos propondo soluções. A seguir, algumas delas são discutidas.

Algumas soluções são específicas para um método ágil. Em Nawrocki et al. [144], o *eXtreme Programming Maturity Model* (XPMM) é apresentado. O mesmo é composto de 4 níveis para guiar a adoção das práticas XP e não foi avaliado empiricamente.

Em Lui e Chan [124], um guia composto de 4 etapas indicando o sequenciamento ideal para adoção de práticas XP no contexto de equipes inexperientes e estudantes chineses é apresentado. O mesmo não foi avaliado empiricamente.

Em Yin et al. [223], um modelo de maturidade específico para o Scrum é apresentado: *Scrum Maturity Model*. O modelo é composto de 5 níveis alinhados com o CMMI-DEV

e cada nível é composto de objetivos e sugestões de práticas. A avaliação da solução foi restrita a questionários.

Modelos de maturidade genéricos para métodos ágeis também são apresentados. Em Packlick [150], uma abordagem orientada a objetivos e composta de 5 etapas é apresentada para guiar equipes na adoção de práticas ágeis. A abordagem é resultado do sucesso em adotar ASD na Sabre Airline Solutions.

Em Benefield [20], 7 práticas (i.e., dimensões) essenciais para definir um método ágil para equipes globais foram definidas: 1) Automação dos testes de regressão; 2) Métricas de qualidade de código; 3) Transição automatizada; 4) Construção de software automatizada e boas práticas de Gerenciamento da Configuração; 5) Desenvolvimento Dirigido por Testes (TDD); 6) Entregas mapeadas e testes de integração; e 7) Testes de performance e escalabilidade. Cada dimensão é avaliada em cinco níveis: 1) Melhores práticas de engenharia emergentes; 2) Práticas contínuas no nível de componente; 3) Integração Contínua de componentes; 4) Integração Contínua de produto; e 5) Lançamento sob demanda Just in Time (JIT). Este resultado é fruto do sucesso ao adotar ASD na British Telecom.

Arcabouços para a adoção de métodos ágeis foram propostos por Sidky et al. [181], Gandomani e Nafchi [102], Qumer e Henderson-Sellers [166] e Ayed et al. [10]. Em Sidky et al. [181], um arcabouço composto de um processo de 4 etapas e um mecanismo, *Sidky Agile Measurement Index* (SAMI), para avaliação da agilidade é apresentado. O SAMI é composto de cinco níveis: 1) Colaborativo; 2) Evolucionário; 3) Efetivo; 4) Adaptativo; e 5) Abrangente. Os níveis são definidos em termos das práticas ágeis adotadas. O arcabouço foi apenas avaliado por meio de um questionário.

Em Gandomani e Nafchi [102], abordagem com foco na transição de métodos tradicionais para ágeis é apresentado. O mesmo é resultado de um estudo de Teoria Fundamentada em Dados [79] com participação de 49 especialistas em ASD. O arcabouço é baseado no ciclo PDCA, iterativo, contínuo, gradual e baseado em valor de negócio. Diferentemente da maioria dos trabalhos citados acima, o mesmo não recomenda práticas e nem utiliza níveis de maturidade. Ele é composto de 4 passos: 1) Seleção de prática; 2) Adaptação; 3) Avaliação; e 4) Ajuste. Uma limitação do mesmo é a falta de instruções relativas à avaliação do método.

O *Agile Adoption and Improvement Model* (AAIM), apresentado em Qumer e

Henderson-Sellers [166], é composto de 6 etapas agrupados em 3 blocos. O primeiro bloco (Prompt) é composto do nível 1) Infância ágil. O segundo (Crux), dos níveis 2) Iniciação Ágil, 3) Realização ágil e 4) Valor ágil. O último bloco (Apex), dos níveis 5) Inteligência ágil e 6) Progresso ágil. Com o mesmo, uma equipe pode criar, modificar e adaptar um método ágil de acordo com o contexto. Ao subir de nível, o modelo recomenda práticas e artefatos até alcançar um ambiente *lean*, com alta qualidade, recursos mínimos e sustentando a agilidade. Dois estudos de caso foram realizados para avaliar o mesmo. No primeiro, o arcabouço foi aplicado em um contexto balanceado de práticas ágeis e tradicionais. No segundo, foi aplicado em uma equipe de apenas dois desenvolvedores. Em ambos os casos, um número limitado de práticas ágeis foram utilizadas.

O *Agile Methods Quality-Integrated Customisation framework*, apresentado em Ayed et al. [10], é composto de 4 etapas e baseado em métricas para auxiliar na customização do método de desenvolvimento. A solução foi parcialmente avaliada em uma empresa. Apesar de inovar na utilização de métricas, o mesmo foca apenas nas práticas do método de desenvolvimento, ignorando os fatores referentes às pessoas. Desta forma, a contribuição deste trabalho é limitada, pois o sucesso da adoção de métodos ágeis é mais dependente de capacidades subjetivas tais como colaboração, comunicação, comprometimento e auto-organização [72].

Em Salo e Abrahamsson [174], um processo sistemático e iterativo composto de 6 passos para melhoria da adoção de ASD foi proposto. Para a definição das ações de melhoria é sugerido utilizar práticas como Análise de Causa Raíz [21], Grupo de Foco [108] e GQM [13] dependendo do conhecimento da equipe e de um facilitador externo. O processo foi avaliado apenas em projetos compostos por estudantes.

Questionários para avaliação de métodos ágeis são apresentados em Williams et al. [216], Fontana et al. [73], Pikkarainen e Passoja [162] e da Cunha [48]. Em Williams et al. [216], uma ferramenta, *Comparative Agility*, para avaliar a agilidade de projetos ágeis é apresentada. Trata-se de um questionário composto de 7 dimensões: 1) Trabalho em equipe; 2) Requisitos; 3) Planejamento; 4) Práticas técnicas; 5) Qualidade; 6) Cultura; e 7) Geração de conhecimento. Com a ferramenta, uma equipe pode se comparar com outras com relação à agilidade em cada dimensão. A mesma não foi avaliada empiricamente.

Em Fontana et al. [73], o foco de melhoria em ASD é na equipe, ao invés de proces-

tos. Neste estudo, um arcabouço progressivo para a maturidade em 6 quesitos: 1) Práticas; 2) Equipe; 3) Entregas; 4) Requisitos; 5) Produto; 6) Cliente. Este arcabouço, em alinhamento com o Manifesto Ágil, propõe que a melhoria do método deve ser dirigida ao valor de negócio, representado pelos 6 quesitos, e deve ser flexível para que a adaptação ocorra naturalmente de acordo com o contexto do projeto. Em Fontana et al. [74], uma ferramenta baseada em um questionário foi apresentada para avaliar a maturidade da equipe ágil de acordo com esse arcabouço: *Agile Compass*. A limitação da mesma é que se trata de um *checklist*, ou seja, apenas dados subjetivos são coletados.

Em Pikkarainen e Passoja [162], uma ferramenta para a avaliação de quais práticas ágeis são adequadas para um projeto é apresentada. A mesma é baseada apenas em dados subjetivos e não foi avaliada em projetos reais.

Em da Cunha [48], um *checklist* para avaliar o uso do Scrum é apresentado, o qual é utilizado como apoio a um método de avaliação e melhoria do uso e princípios e práticas do Scrum: *Agile DMAIC*. A solução foi avaliada em 15 projetos reais. A limitação da abordagem é apenas coletar dados subjetivos e foco apenas no processo.

Um quadro comparativo entre os trabalhos relacionados e a solução proposta nesta pesquisa é apresentado na Tabela 6.1. Ao analisar a Tabela 6.1, percebe-se que todos os trabalhos relacionados têm ao menos uma das seguintes limitações: não tem flexibilidade na escolha das práticas, não avalia fatores do trabalho em equipe ou não tem apoio ferramental. A exceção é o trabalho apresentado em Fontana et al. [73, 74]. Por outro, a ferramenta *Agile Compass* é um *checklist*, limitando assim o apoio ao diagnóstico e prognóstico do método de desenvolvimento.

Comparado com os trabalhos relacionados, as maiores novidades da abordagem proposta nesta pesquisa são: (1) o uso de fontes complementares tais como questionários e métricas para a avaliação do método de desenvolvimento Scrum; (2) o prognóstico da satisfação do cliente com o incremento do produto de acordo com dados coletados sobre os atributos das principais entidades do processo; (3) um construto de trabalho em equipe para auxiliar na avaliação da equipe de desenvolvimento; e (4) o uso de redes Bayesianas para o apoio ao diagnóstico e prognóstico do método de desenvolvimento.

Tabela 6.1: Comparação entre os trabalhos relacionados e a solução proposta.

Estudo	Tipo de solução	Método ágil alvo	Flexibilidade de práticas	Fatores de trabalho em equipe	Suporte ferramental
Nawrocki et al.	Modelo de maturidade	XP	Não	Não	Não
Lui e Chan	Modelo de maturidade	XP	Não	Não	Não
Yin et al.	Modelo de maturidade	Scrum	Não	Não	Não
Packlick	Modelo de maturidade	Genérico	Sim	Não	Não
Benefield	Modelo de maturidade	Genérico	Não	Não	Não
Sidky et al.	Arcabouço de adoção	Genérico	Sim	Não	Checklist
Qumer e Henderson-Sellers	Arcabouço de adoção	Genérico	Sim	Não	Questionário
Gandomani e Nafchi	Arcabouço de adoção	Genérico	Sim	Não	Não
Ayed et al.	Arcabouço de adoção	Genérico	Sim	Não	Questionário Métricas
Salo e Abrahamsson	Melhoria de processo	Genérico	Sim	Não	Não
Williams et al.	Avaliação da agilidade	Genérico	Não	Não	Questionário
Fontana et al.	Melhoria de processo	Genérico	Sim	Sim	Checklist

Table 6.1 continuada da página passada.

Estudo	Tipo de solução	Método ágil alvo	Flexibilidade de práticas	Fatores de trabalho em equipe	Suporte ferramental
Pikkarainen e Passoja	Avaliação da agilidade	Genérico	Sim	Não	Questionário
da Cunha et al.	Melhoria de processo	Scrum	Sim	Não	Checklist
Perkusich	Melhoria de processo	Scrum	Sim	Sim	Questionário, Métricas e Rede Bayesiana

Capítulo 7

Considerações finais

Nesta pesquisa, foi apresentada uma solução para instrumentar o Scrum e auxiliar na melhoria contínua do método de desenvolvimento. A solução trata-se de um processo sistemático (Capítulo 4) com apoio de rede Bayesiana (Capítulo 3) para auxiliar no diagnóstico e prognóstico do método de desenvolvimento. A solução proposta está em conformidade com os princípios Scrum e assume a ausência de volume de dados históricos, aumentando sua praticabilidade.

A rede Bayesiana foi construída por meio da elicitación do conhecimento de especialistas seguindo o processo de Engenharia de Conhecimento de Redes Bayesianas (KEBN) [131]. Ela foi construída de forma iterativa e incremental. No total, quatro versões foram desenvolvidas e a rede Bayesiana evoluiu para ser minimalista (i.e., genérica) e mais flexível para diferentes tipos de fontes de dados (e.g., questionários ou ferramentas de desenvolvimento de software). Esta tese limita-se a apresentar detalhes apenas da última versão.

A rede Bayesiana realiza a previsão da satisfação do cliente de acordo com métricas calculadas sobre as principais entidades do método de desenvolvimento. A mesma é flexível para ser complementada com indicadores e práticas de desenvolvimento e gerenciamento utilizados no projeto em questão. Para entrada de dados na rede Bayesiana, pode-se utilizar um questionário ou métricas coletadas de ferramentas (e.g., cobertura de código de testes).

A rede Bayesiana foi validada em duas etapas: (i) conceitual (Seção 3.4) e (ii) cenários de simulação (Seção 3.5). Na validação conceitual (i), foi avaliado se a mesma detecta os 14 antipadrões Scrum apresentados em Eloranta et al. [60]. Como resultado, 1 foi considerado inválido (i.e., fora do escopo do modelo), 12 foram detectados e 1 foi parcialmente

detectado pela rede Bayesiana base. Com estes resultados, conclui-se que a rede Bayesiana pode auxiliar na identificação dos principais problemas na aplicação do Scrum. No segundo passo da validação (ii), doze cenários de simulação foram definidos pelo pesquisador e pelos especialistas de domínio que colaboraram na pesquisa. Para cada cenário, foi definida uma descrição, dados de entrada e resultados esperados. Os resultados esperados foram comparados com os resultados calculados pelo modelo. Ao final do processo de validação e recalibração, as previsões calculadas pela rede Bayesiana estavam em conformidade com os resultados esperados de todos os cenários.

A avaliação da solução foi realizada em duas etapas: (i) utilidade prática (Seção 5.1) e (ii) acurácia da predição (Seção 5.2). A utilidade prática está relacionada ao grau de utilidade do processo para auxiliar na melhoria contínua do método de desenvolvimento. A acurácia da predição está relacionada à acurácia dos prognósticos calculados pela rede Bayesiana. Para (i), um estudo de caso foi realizado durante três sprints com duração de 15 dias em dois projetos de uma empresa. Como resultado, foi concluído que o modelo é robusto para auxiliar na identificação de problemas no método de desenvolvimento com custo-benefício positivo.

Para (ii), dados de 18 projetos de uma empresa foram coletados por meio de um questionário. Essa base de dados foi utilizada para avaliar a acurácia da predição da rede Bayesiana. Como resultado, a previsão foi correta para quatorze projetos (acurácia de 78%). Dado que o processo de adaptação para o contexto da empresa, devido a limitações de recursos fornecidos pela mesma, não foi o ideal, a acurácia poderia ter sido melhor. Dessa forma, conclui-se que o modelo é capaz de realizar previsões com acurácia satisfatória e, dessa forma, é útil para auxiliar nas tomadas de decisões de projetos Scrum. Ou seja, com os resultados dos dois estudos, pode-se concluir que a solução proposta é adequada para instrumentar o Scrum e auxiliar na melhoria contínua do método de desenvolvimento.

Os resultados desta tese demonstram que utilizar a engenharia de conhecimento, mais especificamente o processo KEBN, é uma alternativa quando há a necessidade de construir uma rede Bayesiana e ausência de dados históricos. Por outro lado, isso só é possível caso haja a disponibilidade de especialistas de domínio e um especialista de redes Bayesianas capaz de elicitar os dados necessários.

Além disso, os resultados demonstram que utilizar a metodologia Design Science execu-

tado por meio de um processo em espiral é uma alternativa viável quando há a necessidade de se propor uma solução (e.g., processo, método ou algoritmo) inovadora dentro de um contexto com incerteza (e.g., não se sabe se a solução será útil). Ou seja, seguindo a tendência de processos de desenvolvimento de software tal como prototipação, ASD e *Continuous Delivery*, é possível agilizar a avaliação da utilidade prática da solução resultante de projetos de pesquisa a partir da avaliação de versões completas, porém não necessariamente otimizadas.

Desta forma, os resultados da avaliação da acurácia da previsão da rede Bayesiana demonstram que é possível utilizar a engenharia de conhecimento para construir uma rede Bayesiana para prever a satisfação do cliente e dar apoio ao processo de diagnóstico do método de desenvolvimento Scrum, dando suporte à **H1** (ver hipóteses na Seção 1.3). Além disso, os resultados da avaliação da utilidade prática demonstram que é possível definir um processo para utilização da rede Bayesiana para dar apoio à melhoria contínua do método de desenvolvimento Scrum com foco na avaliação do processo de engenharia de requisitos, equipe de desenvolvimento e incrementos do produto, dando suporte à **H2**.

Por outro lado, o modelo foi construído por meio de dados da literatura e o conhecimento de especialistas. Apesar dos resultados positivos nas avaliações realizadas, não há dados suficientes para estatisticamente afirmar que o problema de seleção de atributos para o modelo (i.e., *model selection*) está resolvido. Para tal, é especulado por Anderson e Vastag [9] que uma amostra de 100 é o suficiente para tal. Por outro lado, o fato dos projetos nos quais foram coletados dados terem contextos (e.g., tecnologias e complexidade) diferentes aumenta a credibilidade dessa pesquisa.

7.1 Trabalhos futuros

A construção da rede Bayesiana resultante desta pesquisa partiu do princípio de que não há dados disponíveis, ou seja, utilizou-se, inicialmente, um processo apenas dirigido à elicitación de conhecimento de especialistas. Por outro lado, a partir do momento que o modelo é aplicado em uma empresa, dados com relação às variáveis de interesse são coletados. Dessa forma, pode-se definir um processo para validação e adequação contínua do modelo de acordo com os dados coletados. Além disso, pode-se definir um processo para, a partir do momento que um modelo é calibrado com os dados de uma empresa, transferir o conheci-

mento para o contexto de outra empresa para minimizar o custo de implantação do mesmo, explorando conceitos apresentados em Zhou et al. [228] .

Atualmente, o escopo do modelo é restrito à previsão e diagnóstico da qualidade associado a atributos do processo de entrega de software Scrum. Dessa forma, durante a utilização do mesmo, uma base de conhecimento com relação aos problemas que aconteceram ou poderiam ter acontecido (i.e., riscos) em projetos nos quais o modelo foi aplicado são registrados. Por outro lado, as intervenções (i.e., ações corretivas e preventivas planejadas nas Reuniões de Retrospectiva e executada durante as sprints) e os resultados das mesmas não foram modelados. Pode-se adicionar essas informações ao modelo seguindo o processo apresentado em Constantinou et al. [46]. Isso possibilitará, de acordo com o risco identificado e contexto do projeto, a recomendação de intervenções de acordo com a probabilidade do efeito das mesmas na satisfação do cliente. Desta forma, a empresa terá um processo empírico baseado em riscos para adaptação e melhoria contínua do planejamento e execução dos projetos Scrum.

Nesta pesquisa, devido ao alto custo operacional na coleta de dados na empresa em questão, apenas dados do Scrum Master foram coletados. Como apresentado em Lindsjørn et al. [123], gerentes, líderes e desenvolvedores têm perspectivas diferentes com relação à eficácia (i.e., qualidade dos entregáveis) e eficiência (i.e., custo-benefício) do projeto. Quanto mais alto o nível hierárquico, maior o foco aos atributos externos do produto; quanto mais baixo, maior o foco aos atributos internos. Dessa forma, hipoteticamente, é possível que o mesmo comportamento se repita para outros atributos do processo, tais como a avaliação do Product Owner, reuniões Scrum ou Product Backlog. Desta forma, é possível que membros do mesmo projeto respondam às perguntas do modelo de forma diferente mesmo dentro do mesmo cenário, devido ao seu papel no projeto. Isso pode influenciar negativamente nas previsões do modelo. Então, pode-se investigar, no escopo do nosso modelo, o viés nos dados de acordo com o papel do respondente.

Além disso, dado que o Scrum se trata de um processo iterativo, pode-se utilizar o conceito de rede Bayesiana dinâmica (RBD). Em Hearty et al. [89], foi utilizada RBD para prever a velocidade de uma equipe XP. Dessa forma, planeja-se avaliar o uso de RBD para prever a satisfação do cliente, utilizando os conceitos apresentados nessa pesquisa. Tal abordagem poderia ser utilizada para, de acordo com as observações de uma dada sprint (e.g.,

ausência do Product Owner e falta de utilização de boas práticas de testes) prever o impacto das mesmas em entidades do processo na sprint seguinte (e.g., qualidade do Backlog do produto e qualidade do incremento), melhorando a acurácia das previsões. Finalmente, é necessário coletar mais dados para estatisticamente ter confiança nos atributos selecionados e minimizar a ameaça à validade externa da solução.

Bibliografia

- [1] N. Abbas, A. M. Gravell, and G. B. Wills. The impact of organization, project and governance variables on software quality and project success. In *Agile Conference (AGILE), 2010*, pages 77–86, Aug 2010.
- [2] M. Abouelela and L. Benedicenti. Bayesian network based xp process modelling. *International Journal of Software Engineering and Applications*, 1(3):1–15, 2010.
- [3] P. Abrahamsson. Extreme programming: first results from a controlled case study. In *Euromicro Conference, 2003. Proceedings. 29th*, pages 259–266, Sept 2003.
- [4] P. Abrahamsson, I. Fronza, R. Moser, J. Vlasenko, and W. Pedrycz. Predicting development effort from user stories. In *2011 International Symposium on Empirical Software Engineering and Measurement*, pages 400–403, Sept 2011.
- [5] Mashal Kasem Alqudah and Rozilawati Razali. Key factors for selecting an agile method: A systematic literature review. *International Journal on Advanced Science, Engineering and Information Technology*, 7(2):526–537, 2017.
- [6] Sultan Alshehri and Luigi Benedicenti. Ranking approach for the user story prioritization methods. *Journal of Communication and Computer*, 10:1465 – 1474, 2013.
- [7] Scott Ambler. *Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process*. Wiley, 1 edition, 12 2002.
- [8] I. Ancveire, I. Gailite, M. Gailite, and J. Grabis. Software delivery risk management: Application of bayesian networks in agile software development. *Information Technology and Management Science*, 18(1):62–69, 2009.

-
- [9] Ronald D Anderson and Gyula Vastag. Causal modeling alternatives in operations research: Overview and application. *European Journal of Operational Research*, 156(1):92 – 109, 2004. EURO Excellence in Practice Award 2001.
- [10] H. Ayed, N. Habra, and B. Vanderose. Am-quick: A measurement-based framework for agile methods customisation. In *Software Measurement and the 2013 Eighth International Conference on Software Process and Product Measurement (IWSM-MENSURA), 2013 Joint Conference of the 23rd International Workshop on*, pages 71–80, Oct 2013.
- [11] N. Ayewah, D. Hovemeyer, J. D. Morgenthaler, J. Penix, and W. Pugh. Using static analysis to find bugs. *IEEE Software*, 25(5):22–29, Sept 2008.
- [12] Simon Baker and Emilia Mendes. Evaluating the weighted sum algorithm for estimating conditional probabilities in bayesian networks. In *21st International Conference on Software Engineering and Knowledge Engineering, SEKE 2010*, pages 319–324, 2010.
- [13] Victor R. Basili. Software modeling and measurement: The goal/question/metric paradigm. Technical report, College Park, MD, USA, 1992.
- [14] Richard Baskerville, Balasubramaniam Ramesh, Linda Levine, Jan Pries-Heje, and Sandra Slaughter. Is internet-speed software development different? *IEEE Software*, 20(6):70–77, November 2003.
- [15] Julian M Bass. Scrum master activities: process tailoring in large enterprise projects. In *Global Software Engineering (ICGSE), 2014 IEEE 9th International Conference on*, pages 6–15. IEEE, 2014.
- [16] Julian M Bass. Artefacts and agile method tailoring in large-scale offshore software development programmes. *Information and Software Technology*, 75:1–16, 2016.
- [17] Kent Beck. *Extreme Programming Explained: Embrace Change*. Addison-Wesley, 1 edition, 11 2000.

- [18] Sarah Beecham. A study of the scrum master's role. In *Product-Focused Software Process Improvement: 18th International Conference, PROFES 2017, Innsbruck, Austria, November 29–December 1, 2017, Proceedings*, volume 10611, page 307. Springer, 2018.
- [19] Irad Ben-Gal. *Bayesian Networks*. John Wiley and Sons, 2007.
- [20] R. Benefield. Seven dimensions of agile maturity in the global enterprise: A case study. In *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*, pages 1–7, Jan 2010.
- [21] B. L. S. Bergman, A. P. Fundin, I. C. Gremyr, and P. M. Johansson. Beyond root-cause analysis [product development process improvement]. In *Annual Reliability and Maintainability Symposium. 2002 Proceedings (Cat. No.02CH37318)*, pages 140–146, 2002.
- [22] Joao H.J.A. Bernardo, Jose J.L. Dias Junior, and Jose A.O.G. Cunha. Scrummaster's attributions and competences: An exploratory study. In *Proceedings of the XII Brazilian Symposium on Information Systems on Brazilian Symposium on Information Systems: Information Systems in the Cloud Computing Era - Volume 1*, SBSI 2016, pages 62:470–62:477, Porto Alegre, Brazil, Brazil, 2016. Brazilian Computer Society.
- [23] D Bertsekas and I Rhodes. Recursive state estimation for a set-membership description of uncertainty. *IEEE Transactions on Automatic Control*, 16(2):117–128, 1971.
- [24] João Bezerra, Renan Willamy Bezerra Barbosa, Mirko Perkusich, Renata Saraiva, Kyller Gorgônio, Hyggo Almeida, and Angelo Perkusich. An algorithm to define the node probability functions of bayesian networks based on ranked nodes. *International Journal of Engineering Trends and Technology (IJETT)*, 52(3):151–157, 2017.
- [25] João Batista Nunes Bezerra, Renan Willamy Bezerra Barbosa, and Mirko Perkusich. Um método para a construção de funções de probabilidade de redes bayesianas baseada em nós ranqueado. *Revista Princípia*, 1(1):1–11, 2017.
- [26] Gary Bishop, Greg Welch, et al. An introduction to the kalman filter. *Proc of SIGGRAPH, Course*, 8(27599-23175):41, 2001.

-
- [27] B. Boehm. Software engineering is a value-based contact sport. *IEEE Softw.*, 19(5):95–96, September 2002.
- [28] Barry Boehm. Get ready for agile methods, with care. *Computer*, 35(1):64–69, January 2002.
- [29] Barry Boehm and Richard Turner. *Balancing Agility and Discipline: A Guide for the Perplexed*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2003.
- [30] T. Boneh, A. E. Nicholson, and E. A. Sonenberg. Matilda: A visual tool for modeling with bayesian networks. *International Journal of Intelligent Systems*, 21(11):1127–1150, 2006.
- [31] G. W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, 1950.
- [32] Sjaak Brinkkemper. Method engineering: engineering of information systems development methods and tools. *Information and Software Technology*, 38(4):275 – 280, 1996. Method Engineering and Meta-Modelling.
- [33] Wray Buntine. Theory refinement on bayesian networks. In *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence*, UAI'91, pages 52–60, San Francisco, CA, USA, 1991. Morgan Kaufmann Publishers Inc.
- [34] Sawhney R. Bustamante, A. Agile xxi: Scaling agile for project teams, seapine software, inc. http://downloads.seapine.com/pub/ebooks/AgileScaling_eBook.pdf, 2011. Accessed in 01/10/2017.
- [35] Jeremy Cain. *Planning improvements in natural resource management. guidelines for using bayesian networks to support the planning and management of development programmes in the water sector and beyond*. Centre for Ecology and Hydrology, 2001.
- [36] Amadeu Silveira Campanelli and Fernando Silva Parreiras. Agile methods tailoring - a systematic literature review. *J. Syst. Softw.*, 110(C):85–100, December 2015.

-
- [37] A. Cano, A. R. Masegosa, and S. Moral. A method for integrating expert knowledge when learning bayesian networks from data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 41(5):1382–1394, Oct 2011.
- [38] Kwai-Sang Chin, Da-Wei Tang, Jian-Bo Yang, Shui Yee Wong, and Hongwei Wang. Assessing new product development project risk by bayesian network with a systematic probability generation methodology. *Expert Systems with Applications*, 36(6):9879 – 9890, 2009.
- [39] Tsun Chow and Dac-Buu Cao. A survey study of critical success factors in agile software projects. *Journal of Systems and Software*, 81(6):961 – 971, 2008. Agile Product Line Engineering.
- [40] Peter Coad, Jeff de Luca, and Eric Lefebvre. *Java Modeling In Color With UML: Enterprise Components and Process*. Prentice Hall PTR, 1 edition, 6 1999.
- [41] A. Cockburn and J. Highsmith. Agile software development, the people factor. *Computer*, 34(11):131–133, Nov 2001.
- [42] Alistair Cockburn. *Agile Software Development*. Addison-Wesley Professional, 1 edition, 10 2001.
- [43] Mike Cohn. *Succeeding with Agile: Software Development Using Scrum*. Addison-Wesley Professional, 1 edition, 11 2009.
- [44] Kieran Conboy and Brian Fitzgerald. Method and developer characteristics for effective agile method tailoring: A study of xp expert opinion. *ACM Trans. Softw. Eng. Methodol.*, 20(1):2:1–2:30, July 2010.
- [45] Anthony Constantinou and Norman Fenton. Towards smart-data: Improving predictive accuracy in long-term football team performance. *Knowledge-Based Systems*, 124:93 – 104, 2017.
- [46] Anthony Costa Constantinou, Norman Fenton, William Marsh, and Lukasz Radlinski. From complex questionnaire and interviewing data to intelligent bayesian network models for medical decision support. *Artificial intelligence in medicine*, 67:75–93, 2016.

- [47] Anthony Costa Constantinou, Norman Fenton, and Martin Neil. Integrating expert knowledge with data in bayesian networks: Preserving data-driven expectations when the expert variables remain unobserved. *Expert Systems with Applications*, 56:197 – 208, 2016.
- [48] Thiago FV da Cunha, Ismayle de S Santos, Alysson A de Macedo, Alberto HMT Monteiro, Bruno S Aragão, and Rossana MC Andrade. Cas 2.0: Evolução e automação do checklist de avaliação do scrum para projetos de software. In *Anais do 15º Simpósio Brasileiro de Qualidade de Software, SBQS 2016*, Maceió, Brasil, 2016.
- [49] Raissa da Silva, Mirko Perkusich, Renata Saraiva, Arthur Freire, Hyggo Almeida, and Angelo Perkusich. Improving the applicability of bayesian networks through production rules. In *27th International Conference on Software Engineering and Knowledge Engineering, SEKE 2016*, pages 8 – 13, San Francisco, USA, 2016.
- [50] Balaram Das. Generating conditional probabilities for bayesian networks: Easing the knowledge acquisition problem. *arXiv preprint cs/0411034*, 2004.
- [51] Ana C.V. de Melo and Adilson J. Sanchez. Software maintenance project delays prediction using bayesian networks. *Expert Systems with Applications*, 34(2):908 – 919, 2008.
- [52] Esther Derby and Diana Larson. *Agile Retrospective Making Good Teams Great*. The Pragmatic Programmers, 1 edition, 2006.
- [53] Francisco Javier Diez. Parameter adjustment in bayes networks. the generalized noisy or-gate. In *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence*, pages 99–105. Morgan Kaufmann, 1993.
- [54] Torgeir Dingsøyrr and Geir Kjetil Hanssen. Extending agile methods: postmortem reviews as extended feedback. In *International Workshop on Learning Software Organizations*, pages 4–12. Springer, 2002.
- [55] Shengping Dong and Bin Hu. Multi-agent based simulation of team effectiveness in team’s task process: A member-task interaction perspective. *International Journal of Simulation and Process Modelling*, 4(1):54–68, 2008.

- [56] Paulo Sérgio Medeiros dos Santos, Amanda Varella, Cristine Ribeiro Dantas, and Daniel Beltrão Borges. *Agile Processes in Software Engineering and Extreme Programming: 14th International Conference, XP 2013, Vienna, Austria, June 3-7, 2013. Proceedings*, chapter Visualizing and Managing Technical Debt in Agile Development: An Experience Report, pages 121–134. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
- [57] S. Downey and J. Sutherland. Scrum metrics for hyperproductive teams: How they fly like fighter aircraft. In *System Sciences (HICSS), 2013 46th Hawaii International Conference on*, pages 4870–4878, Jan 2013.
- [58] Srdjana Dragicevic, Stipe Celar, and Mili Turic. Bayesian network model for task effort estimation in agile software development. *Journal of Systems and Software*, 127:109 – 119, 2017.
- [59] Didier Dubois. Possibility theory and statistical reasoning. *Computational statistics & data analysis*, 51(1):47–69, 2006.
- [60] Veli-Pekka Eloranta, Kai Koskimies, and Tommi Mikkonen. Exploring scrumbut—an empirical study of scrum anti-patterns. *Information and Software Technology*, 74:194 – 203, 2016.
- [61] Khaled El Emam and A. Günes Koru. A replicated survey of it software project failures. *IEEE Softw.*, 25(5):84–90, September 2008.
- [62] Chin-Feng Fan and Yuan-Chang Yu. Bbn-based software project risk management. *J. Syst. Softw.*, 73(2):193–203, October 2004.
- [63] N. Fenton, P. Krause, and M. Neil. Software measurement: uncertainty and causal modeling. *Software, IEEE*, 19(4):116–122, 2002.
- [64] NE Fenton and M Neil. Modelling risk in complex software projects using bayesian networks. *RADAR Research Report*, 2003.
- [65] Norman Fenton, Paul Krause, and Martin Neil. Probabilistic modelling for software quality control. *Journal of Applied Non-Classical Logics*, 12(2):173–188, 2002.

- [66] Norman Fenton and Martin Neil. *Risk Assessment and Decision Analysis with Bayesian Networks*. CRC Press, 5 edition, 11 2012.
- [67] Norman Fenton, Martin Neil, William Marsh, Peter Hearty, Łukasz Radliński, and Paul Krause. On the effectiveness of early life cycle defect prediction with bayesian nets. *Empirical Software Engineering*, 13(5):499, 2008.
- [68] Norman E. Fenton and Martin Neil. Software metrics: Roadmap. In *Proceedings of the Conference on The Future of Software Engineering*, ICSE '00, pages 357–370, New York, NY, USA, 2000. ACM.
- [69] Norman E. Fenton, Martin Neil, and Jose Galan Caballero. Using ranked nodes to model qualitative judgments in bayesian networks. *IEEE Trans. on Knowl. and Data Eng.*, 19(10):1420–1432, October 2007.
- [70] Brian Fitzgerald, Gerard Hartnett, and Kieran Conboy. Customising agile methods to software practices at intel shannon. *Eur. J. Inf. Syst.*, 15(2):200–213, April 2006.
- [71] M. Julia Flores, Ann E. Nicholson, Andrew Brunskill, Kevin B. Korb, and Steven Mascaro. Incorporating expert knowledge when learning bayesian network structure: A medical case study. *Artificial Intelligence in Medicine*, 53(3):181 – 204, 2011.
- [72] Rafaela Mantovani Fontana, Isabela Mantovani Fontana, Paula Andrea da Rosa Garbuio, Sheila Reinehr, and Andreia Malucelli. Processes versus people: How should agile software development maturity be defined? *Journal of Systems and Software*, 97:140 – 155, 2014.
- [73] Rafaela Mantovani Fontana, Victor Meyer, Sheila Reinehr, and Andreia Malucelli. Progressive outcomes. *J. Syst. Softw.*, 102(C):88–108, April 2015.
- [74] Rafaela Mantovani Fontana, Sheila S. Reinehr, and Andreia Malucelli. Agile compass: A tool for identifying maturity in agile software-development teams. *IEEE Software*, 32(6):20–23, 2015.
- [75] Martin Fowler. Flaccidscrum. <http://martinfowler.com/bliki/FlaccidScrum.html>, 2009. Acessado em: 02-05-2016.

- [76] Arthur Freire, Mirko Perkusich, Kyller Gorgônio, Hyggo Almeida, and Angelo Perkusich. A bayesian networks-based approach to assess and improve the teamwork quality of agile teams. *Information and Software Technology*, Em avaliação.
- [77] Nir Friedman, Dan Geiger, and Moisés Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2-3):131–163, 1997.
- [78] Maxime Gasse, Alex Aussem, and Haytham Elghazel. A hybrid algorithm for bayesian network structure learning with application to multi-label learning. *Expert Systems with Applications*, 41(15):6755 – 6772, 2014.
- [79] Barney G. Glaser. *Theoretical Sensitivity: Advances in the Methodology of Grounded Theory*. The Sociology Press, 1 edition, 6 1978.
- [80] Robert B. Grady. *Practical Software Metrics for Project Management and Process Improvement*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1992.
- [81] Simon Grapenthin, Steven Poggel, Matthias Book, and Volker Gruhn. Improving task breakdown comprehensiveness in agile projects with an interaction room. *Information and Software Technology*, 67:254 – 264, 2015.
- [82] Viktoria Gulliksen Stray, Nils Brede Moe, and Torgeir Dingsøy. *Agile Processes in Software Engineering and Extreme Programming: 12th International Conference, XP 2011, Madrid, Spain, May 10-13, 2011. Proceedings*, chapter Challenges to Teamwork: A Multiple Case Study of Two Agile Teams, pages 146–161. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
- [83] Chaitanya Gurram and Srinivas Goud Bandi. Teamwork in distributed agile software development. Master’s thesis, Blekinge Institute of Technology, School of Computing, 2013.
- [84] Anca Hanea, Oswaldo Morales Napoles, and Dan Ababei. Non-parametric bayesian networks: Improving theory and reviewing applications. *Reliability Engineering System Safety*, 144:265 – 284, 2015.
- [85] James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982.

-
- [86] Lars Martin Riiser Haraldsen. An investigation of team effectiveness in agile software development. Master's thesis, Norwegian University of Science and Technology, 2012.
- [87] N. C. Haugen. An empirical study of using planning poker for user story estimation. In *AGILE 2006 (AGILE'06)*, pages 9 pp.–34, July 2006.
- [88] Simon Haykin and Neural Network. A comprehensive foundation. *Neural networks*, 2(2004):41, 2004.
- [89] P. Hearty, N. Fenton, D. Marquez, and M. Neil. Predicting project velocity in xp using a learning dynamic bayesian network model. *Software Engineering, IEEE Transactions on*, 35(1):124–137, 2009.
- [90] David Heckerman. Learning in graphical models. chapter A Tutorial on Learning with Bayesian Networks, pages 301–354. MIT Press, Cambridge, MA, USA, 1999.
- [91] David Heckerman, Dan Geiger, and David M. Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, Sep 1995.
- [92] James A. Highsmith. *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems*. Dorset House, 1 edition, 12 1999.
- [93] Martin Hoegl and Hans Georg Gemuenden. Teamwork quality and the success of innovative projects: A theoretical concept and empirical evidence. *Organization science*, 12(4):435–449, 2001.
- [94] Luke Hohmann. *Innovation Games: Creating Breakthrough Products Through Collaborative Play*. Addison-Wesley Professional, 1 edition, 9 2006.
- [95] L Hope, A Nicholson, and K Korb. Knowledge engineering tools for probability elicitation. Technical report, Technical Report, School of Computer Science and Software Engineering, Monash University, 2002.

- [96] David W. Hosmer and Stanley Lemeshow. Goodness of fit tests for the multiple logistic regression model. *Communications in Statistics - Theory and Methods*, 9(10):1043–1069, 1980.
- [97] Yong Hu, Xizhu Mo, Xiangzhou Zhang, Yuran Zeng, Jianfeng Du, and Kang Xie. Intelligent analysis model for outsourced software project risk using constraint-based bayesian network. *Journal of Software*, 7(2), 2012.
- [98] Kurt Huang and Max Henrion. Efficient search-based inference for noisy-or belief networks: Topepsilon. In *Proceedings of the Twelfth International Conference on Uncertainty in Artificial Intelligence, UAI'96*, pages 325–331, San Francisco, CA, USA, 1996. Morgan Kaufmann Publishers Inc.
- [99] Douglas Hubbard and Dylan Evans. Problems with scoring methods and ordinal scales in risk assessment. *IBM Journal of Research and Development*, 54(3):2–1, 2010.
- [100] Maria Hänninen, Osiris A. Valdez Banda, and Pentti Kujala. Bayesian network model of maritime safety management. *Expert Systems with Applications*, 41(17):7837 – 7846, 2014.
- [101] SEIYA IMOTO, TOMOYUKI HIGUCHI, TAKAO GOTO, KOUSUKE TASHIRO, SATORU KUHARA, and SATORU MIYANO. Combining microarrays and biological knowledge for estimating gene networks via bayesian networks. *Journal of Bioinformatics and Computational Biology*, 02(01):77–98, 2004.
- [102] Taghi Javdani Gandomani and Mina Ziaei Nafchi. An empirically-developed framework for agile transition and adoption. *J. Syst. Softw.*, 107(C):204–219, September 2015.
- [103] Kawal Jeet, Nitin Bhatia, and Rajinder Singh Minhas. A bayesian network based approach for software defects prediction. *SIGSOFT Softw. Eng. Notes*, 36(4):1–5, August 2011.
- [104] Kawal Jeet, Nitin Bhatia, and Rajinder Singh Minhas. A model for estimating the impact of low productivity on the schedule of a software development project. *SIGSOFT Softw. Eng. Notes*, 36(4):1–6, August 2011.

-
- [105] Andreas Johansson. Toward improvements of teamwork in globally distributed agile teams. Bachelor of science thesis in software engineering and management, University of Gothenburg, 2013.
- [106] Capers Jones. *Applied Software Measurement: Global Analysis of Productivity and Quality*. McGraw-Hill Education, 3 edition, 5 2008.
- [107] Georg Kalus and Marco Kuhrmann. Criteria for software process tailoring: A systematic review. In *Proceedings of the 2013 International Conference on Software and System Process*, ICSSP 2013, pages 171–180, New York, NY, USA, 2013. ACM.
- [108] F. Kerlinger and H. Lee. *Foundations of Behavioral Research*. Harcourt College Publishers, 1 edition, 2000.
- [109] Omar Z Khan, Pascal Poupart, and M Agosta John-mark. Automated refinement of bayes networks’ parameters based on test ordering constraints. In *Advances in Neural Information Processing Systems*, pages 2591–2599, 2011.
- [110] B. Kitchenham and S Charters. Guidelines for performing systematic literature reviews in software engineering, 2007.
- [111] George Klir and Bo Yuan. *Fuzzy sets and fuzzy logic*, volume 4. Prentice hall New Jersey, 1995.
- [112] KB Korb. Bayesian artificial intelligence/kevin b. korb. *Ann E. Nicholson. p. cm.–Chapman & Hall/CRC computer science and data analysis*, 2003.
- [113] YAVUZ KOZAK. Barriers against better team performance in agile software projects. Master’s thesis, Chalmers University of Technology, Sweden, 2013.
- [114] Eetu Kupiainen, Mika V. Mäntylä, and Juha Itkonen. Using metrics in agile and lean software development – a systematic literature review of industrial studies. *Information and Software Technology*, 62:143 – 163, 2015.
- [115] Corey Ladas. *Scrumban - Essays on Kanban Systems for Lean Software Development*. Modus Cooperandi Press, 1 edition, 1 2009.

- [116] P. Laitila and K. Virtanen. Improving construction of conditional probability tables for ranked nodes in bayesian networks. *IEEE Transactions on Knowledge and Data Engineering*, 28(7):1691–1705, July 2016.
- [117] Steffen L. Lauritzen. The em algorithm for graphical association models with missing data. *Computational Statistics Data Analysis*, 19(2):191 – 201, 1995.
- [118] Lucas Layman, Laurie Williams, and Lynn Cunningham. Exploring extreme programming in context: An industrial case study. In *Proceedings of the Agile Development Conference, ADC '04*, pages 32–41, Washington, DC, USA, 2004. IEEE Computer Society.
- [119] Lucas Layman, Laurie Williams, and Lynn Cunningham. Motivations and measurements in an agile case study. *J. Syst. Archit.*, 52(11):654–667, November 2006.
- [120] Eunchang Lee, Yongtae Park, and Jong Gye Shin. Large engineering project risk management using a bayesian belief network. *Expert Syst. Appl.*, 36(3):5880–5887, April 2009.
- [121] J. F. Lemmer and D. E. Gossink. Recursive noisy or - a rule for estimating complex probabilistic interactions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(6):2252–2261, Dec 2004.
- [122] Yngve Lindsjörn, Dag IK Sjøberg, Torgeir Dingsøy, Gunnar R Bergersen, and Tore Dybå. Teamwork quality and project success in software development: A survey of agile development teams. *Journal of Systems and Software*, 122:274–286, 2016.
- [123] Yngve Lindsjörn, Dag I.K. Sjøberg, Torgeir Dingsøy, Gunnar R. Bergersen, and Tore Dybå. Teamwork quality and project success in software development: A survey of agile development teams. *Journal of Systems and Software*, 122:274 – 286, 2016.
- [124] Kim Man Lui and Keith C. C. Chan. *Unifying the Software Process Spectrum: International Software Process Workshop, SPW 2005, Beijing, China, May 25-27, 2005, Revised Selected Papers*, chapter A Road Map for Implementing eXtreme Programming, pages 474–481. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

- [125] Chris Mann and Frank Maurer. A case study on the impact of scrum on overtime and customer satisfaction. In *Agile Conference, 2005. Proceedings*, pages 70–79. IEEE, 2005.
- [126] Robert C. Martin. *Agile Software Development: Principles, Patterns, and Practices*. Prentice Hall, 2 edition, 10 2002.
- [127] Andrés R. Masegosa and Serafín Moral. An interactive approach for bayesian network learning using domain/expert knowledge. *International Journal of Approximate Reasoning*, 54(8):1168 – 1181, 2013.
- [128] Frank Maurer and Sebastien Martel. Extreme programming: Rapid development for web-based applications. *IEEE Internet Computing*, 6(1):86–90, January 2002.
- [129] Stijn Meganck, Philippe Leray, and Bernard Manderick. Learning causal bayesian networks from observations and experiments: A decision theoretic approach. In Vicenç Torra, Yasuo Narukawa, Aïda Valls, and Josep Domingo-Ferrer, editors, *Modeling Decisions for Artificial Intelligence*, pages 58–69, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
- [130] E. Mendes and N. Mosley. Bayesian network models for web effort prediction: A comparative study. *IEEE Transactions on Software Engineering*, 34(6):723–737, Nov 2008.
- [131] Emilia Mendes. *Practitioner’s knowledge representation: a pathway to improve software effort estimation*. Springer Science & Business, 2014.
- [132] Emilia Mendes, Pilar Rodriguez, Vitor Freitas, Simon Baker, and Mohamed Amine Atoui. Towards improving decision making and estimating the value of decisions in value-based software engineering: the value framework. *Software Quality Journal*, Mar 2017.
- [133] A. T. Misirli and A. B. Bener. Bayesian networks for evidence-based decision-making in software engineering. *IEEE Transactions on Software Engineering*, 40(6):533–554, June 2014.

- [134] Subhas Chandra Misra, Vinod Kumar, and Uma Kumar. Identifying some important success factors in adopting agile software development practices. *J. Syst. Softw.*, 82(11):1869–1890, November 2009.
- [135] L. Mkrtchyan, L. Podofillini, and V.N. Dang. Methods for building conditional probability tables of bayesian belief networks from limited judgment: An evaluation for human reliability application. *Reliability Engineering System Safety*, 151:93 – 112, 2016.
- [136] Nils Brede Moe, Torgeir Dingsøy, and Tore Dybå. A teamwork model for understanding an agile team: A case study of a scrum project. *Information and Software Technology*, 52(5):480 – 491, 2010. TAIC-PART 2008TAIC-PART 2008.
- [137] NilsBrede Moe, Torgeir Dingsøy, and EmilA. Røyrvik. Putting agile teamwork to the test – an preliminary instrument for empirically assessing and improving agile software development. In Pekka Abrahamsson, Michele Marchesi, and Frank Maurer, editors, *Agile Processes in Software Engineering and Extreme Programming*, volume 31 of *Lecture Notes in Business Information Processing*, pages 114–123. Springer Berlin Heidelberg, 2009.
- [138] S. Monti and G. Carenini. Dealing with the expert inconsistency in probability elicitation. *IEEE Transactions on Knowledge and Data Engineering*, 12(4):499–508, Jul 2000.
- [139] O. Morales, D. Kurowicka, and A. Roelen. Eliciting conditional and unconditional rank correlations from conditional probabilities. *Reliability Engineering System Safety*, 93(5):699 – 710, 2008. Expert Judgement.
- [140] Mario E. Moreira. *Being Agile: Your Roadmap to Successful Adoption of Agile*, chapter Writing User Stories and Grooming the Backlog, pages 175–185. Apress, Berkeley, CA, 2013.
- [141] Brian Mullen and Carolyn Copper. The relation between group cohesiveness and performance: An integration. *Psychological bulletin*, 115(2):210, 1994.

- [142] Kevin P. Murphy. Active learning of causal bayes net structure. Technical report, 2001.
- [143] A. Nagy, M. Njima, and L. Mkrtchyan. A bayesian based method for agile software development release planning and project health monitoring. In *2010 International Conference on Intelligent Networking and Collaborative Systems*, pages 192–199, Nov 2010.
- [144] J. Nawrocki, B. Walter, and A. Wojciechowski. Toward maturity model for extreme programming. In *Euromicro Conference, 2001. Proceedings. 27th*, pages 233–239, 2001.
- [145] Richard E. Neapolitan. *Learning Bayesian Networks*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2003.
- [146] Martin Neil, Norman Fenton, and Lars Nielson. Building large-scale bayesian networks. *Knowl. Eng. Rev.*, 15(3):257–284, September 2000.
- [147] Sridhar Nerur and VenuGopal Balijepally. Theoretical reflections on agile development methodologies. *Commun. ACM*, 50(3):79–83, March 2007.
- [148] Radu Stefan Niculescu, Tom M. Mitchell, and R. Bharat Rao. Bayesian network learning with parameter constraints. *J. Mach. Learn. Res.*, 7:1357–1383, December 2006.
- [149] A. O’Hagan, C. E. Buck, A. Daneshkhah, Eiser J. R., Garthwaite P. H., D. J. Jenkinson, Oakley J. E., and Rakow T. *Uncertain Judgements: eliciting experts’ probabilities*. Wiley, 1 edition, 10 2006.
- [150] Jay Packlick. The agile maturity map a goal oriented approach to agile improvement. In *Proceedings of the AGILE 2007, AGILE ’07*, pages 266–271, Washington, DC, USA, 2007. IEEE Computer Society.
- [151] J. Pearl and S. Russell. Bayesian networks. *Handbook of brain theory and neural networks*, 1995.

- [152] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.
- [153] P.C. Pendharkar, G.H. Subramanian, and J.A. Rodger. A probabilistic model for predicting software development effort. *Software Engineering, IEEE Transactions on*, 31(7):615–624, 2005.
- [154] Mirko Perkusich. Um método baseado em modelo probabilístico para auxílio na detecção de problemas na utilização do scrum em projetos de desenvolvimento. Master of science dissertation in computer science, Federal University of Campina Grande, 2013.
- [155] Mirko Perkusich, Hyggo Oliveira de Almeida, and Angelo Perkusich. A model to detect problems on scrum-based software development projects. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing, SAC '13*, pages 1037–1042, New York, NY, USA, 2013. ACM.
- [156] Mirko Perkusich, Kyller Gorgonio, Hyggo Almeida, and Angelo Perkusich. A framework to build bayesian networks to assess scrum-based development methods. In *28th International Conference on Software Engineering and Knowledge Engineering, SEKE 2017*, pages 67 – 73, Pittsburgh, USA, 2017.
- [157] Mirko Perkusich, Kyller Costa Gorgônio, Hyggo Almeida, and Angelo Perkusich. Assisting the continuous improvement of scrum projects using metrics and bayesian networks. *Journal of Software: Evolution and Process*, 29(6):e1835–n/a, 2017. e1835 smr.1835.
- [158] Mirko Perkusich, Angelo Perkusich, and Hyggo Almeida. Using survey and weighted functions to generate node probability tables for *Bayesian* networks. In *Proceedings of BRICS-CCI 2013*, 2013.
- [159] Mirko Perkusich, Gustavo Soares, Hyggo Almeida, and Angelo Perkusich. A procedure to detect problems of processes in software development projects using bayesian networks. *Expert Systems with Applications*, 42(1):437 – 450, 2015.

- [160] Kai Petersen and Claes Wohlin. A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case. *J. Syst. Softw.*, 82(9):1479–1490, September 2009.
- [161] Roman Pichler. *Agile Product Management with Scrum: Creating Products that Customers Love*. Addison-Wesley Professional, 1 edition, 4 2010.
- [162] Minna Pikkarainen and Ulla Passoja. An approach for assessing suitability of agile solutions: A case study. In *Proceedings of the 6th International Conference on Extreme Programming and Agile Processes in Software Engineering, XP'05*, pages 171–179, Berlin, Heidelberg, 2005. Springer-Verlag.
- [163] Luca Podofillini and Vinh N Dang. A bayesian approach to treat expert-elicited probabilities in human reliability analysis model construction. *Reliability Engineering & System Safety*, 117:52–64, 2013.
- [164] R. Popli, N. Chauhan, and H. Sharma. Prioritising user stories in agile environment. In *Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014 International Conference on*, pages 515–519, Feb 2014.
- [165] Kevin Pulford, Annie Kuntzmann-Combelles, and Stephen Shirlaw, editors. *A Quantitative Approach to Software Management: The AMI Handbook*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995.
- [166] A. Qumer and B. Henderson-Sellers. A framework to support the evaluation, adoption and improvement of agile methods in practice. *Journal of Systems and Software*, 81(11):1899 – 1919, 2008.
- [167] Lukasz Radlinski. A survey of bayesian net models for software development effort prediction. *International Journal of Software Engineering and Computing*, 2(2):95–109, 2010.
- [168] Donald J. Reifer. *Extreme Programming and Agile Methods — XP/Agile Universe 2002: Second XP Universe and First Agile Universe Conference Chicago, IL, USA*,

- August 4–7, 2002 Proceedings*, chapter How to Get the Most out of Extreme Programming/Agile Methods, pages 185–196. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.
- [169] Silja Renooij and Cilia Witteman. Talking probabilities: communicating probabilistic information with words and numbers. *International Journal of Approximate Reasoning*, 22(3):169 – 194, 1999.
- [170] Matthew Richardson and Pedro Domingos. Learning with knowledge from multiple experts. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 624–631, 2003.
- [171] Mats Angermo Ringstad, Torgeir Dingsøy, and Nils Brede Moe. *Systems, Software and Service Process Improvement: 18th European Conference, EuroSPI 2011, Roskilde, Denmark, June 27-29, 2011. Proceedings*, chapter Agile Process Improvement: Diagnosis and Planning to Improve Teamwork, pages 167–178. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
- [172] Willy Røed, Ali Mosleh, Jan Erik Vinnem, and Terje Aven. On the use of the hybrid causal logic method in offshore risk analysis. *Reliability Engineering System Safety*, 94(2):445 – 455, 2009.
- [173] Eduardo Salas, Dana E. Sims, and C. Shawn Burke. Is there a “big five” in teamwork? *Small Group Research*, 36(5):555–599, 2005.
- [174] Outi Salo and Pekka Abrahamsson. An iterative improvement process for agile software development. *Software Process: Improvement and Practice*, 12(1):81–100, 2007.
- [175] Eugene Santos, John T. Wilkinson, and Eunice E. Santos. Fusing multiple bayesian knowledge sources. *International Journal of Approximate Reasoning*, 52(7):935 – 947, 2011. Selected Papers - Uncertain Reasoning Track - FLAIRS 2009.
- [176] B. Schatz and I. Abdelshafi. Primavera gets agile: a successful transition to agile development. *IEEE Software*, 22(3):36–42, May 2005.

- [177] Christoph T. Schmidt, Thomas Kude, Armin Heinzl, and Sunil Mithas. How agile practices influence the performance of software development teams: The role of shared mental models and backup. In *ICIS 2014 Proceedings*, page Paper 15, Atlanta, Ga., 2014. AISeL.
- [178] Raymond Scupin. The kj method: A technique for analyzing data derived from japanese ethnology. *Human organization*, 56(2):233–237, 1997.
- [179] Pedro Serrador and Jeffrey K. Pinto. Does agile work? — a quantitative analysis of agile project success. *International Journal of Project Management*, 33(5):1040 – 1051, 2015.
- [180] D. Settas, S. Bibi, P. Sfetsos, I. Stamelos, and V. Gerogiannis. Using bayesian belief networks to model software project management antipatterns. In *Fourth International Conference on Software Engineering Research, Management and Applications (SERA'06)*, pages 117–124, Aug 2006.
- [181] Ahmed Sidky, James Arthur, and Shawn Bohner. A disciplined approach to adopting agile practices: the agile adoption framework. *Innovations in Systems and Software Engineering*, 3(3):203–216, 2007.
- [182] Ana Silva, Thalles Araújo, João Nunes, Mirko Perkusich, Ednaldo Dilorenzo, Hyggo Almeida, and Angelo Perkusich. A systematic review on the use of definition of done on agile software development projects. In *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*, pages 364–373. ACM, 2017.
- [183] Ana Silva, André Silva, Thalles Araújo, Renan Barbosa, Felipe Ramos, Antônio Costa, Mirko Perkusich, and Ednaldo Dilorenzo. Ordering the product backlog in agile software development projects: A systematic literature review. In *28th International Conference on Software Engineering and Knowledge Engineering, SEKE 2017*, pages 1 – 7, Pittsburgh, USA, 2017.
- [184] Fernando Selleri Silva, Felipe Santana Furtado Soares, Angela Lima Peres, Ivanildo Monteiro de Azevedo, Ana Paula L.F. Vasconcelos, Fernando Kenji Kamei,

- and Silvio Romero de Lemos Meira. Using cmmi together with agile software development: A systematic review. *Information and Software Technology*, 58:20 – 43, 2015.
- [185] A. Silva Freire, R.M. Da Silva, M. Perkusich, H. Almeida, and A. Perkusich. A bayesian network model to assess agile teams' teamwork quality. In *Software Engineering (SBES), 2015 29th Brazilian Symposium on*, pages 191–196, Sept 2015.
- [186] S. Soundararajan and J. D. Arthur. A structured framework for assessing the "goodness" of agile methods. In *Engineering of Computer Based Systems (ECBS), 2011 18th IEEE International Conference and Workshops on*, pages 14–23, April 2011.
- [187] Carlo Spaccasassi and Lea Deleris. A web-based tool for expert elicitation in distributed teams. In *Proceedings of the 8th Bayesian Modelling Applications Workshop of UAI*, volume 2011, 2011.
- [188] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. MIT press, 2nd edition, 2000.
- [189] Dragan Stankovic, Vesna Nikolic, Miodrag Djordjevic, and Dac-Buu Cao. A survey study of critical success factors in agile software projects in former yugoslavia {IT} companies. *Journal of Systems and Software*, 86(6):1663 – 1678, 2013.
- [190] Jennifer Stapleton and Peter Constable. *DSDM: Dynamic Systems Development Method: The Method in Practice*. Addison-Wesley Professional, 1 edition, 7 1997.
- [191] Viktoria Stray, Dag I.K. Sjøberg, and Tore Dybå. The daily stand-up meeting: A grounded theory study. *Journal of Systems and Software*, 114:101 – 124, 2016.
- [192] Viktoria Gulliksen Stray, Nils Brede Moe, and Tore Dybå. *Agile Processes in Software Engineering and Extreme Programming: 13th International Conference, XP 2012, Malmö, Sweden, May 21-25, 2012. Proceedings*, chapter Escalation of Commitment: A Longitudinal Case Study of Daily Meetings, pages 153–167. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

- [193] Jeff Sutherland and Ken Schwaber. The scrum guide. <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-US.pdf>, 2013. Acessado em: 01-02-2018.
- [194] Satish Thatte. Product backlog is deep; invest wisely and dive carefully. <https://blogs.versionone.com/agilemanagement/2015/04/14/product-backlog-is-deep-invest-wisely-and-dive-carefully/>, 2015. Acessado em : 02-05-2016.
- [195] Simon Tong and Daphne Koller. Active learning for structure in bayesian networks. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 2, IJCAI'01*, pages 863–869, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.
- [196] Ioannis Tsamardinos, Laura E. Brown, and Constantin F. Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, Oct 2006.
- [197] Jodie B Ullman and Peter M Bentler. *Structural equation modeling*. Wiley Online Library, 2003.
- [198] Muhammad Usman, Emilia Mendes, Francila Weidt, and Ricardo Britto. Effort estimation in agile software development: A systematic literature review. In *Proceedings of the 10th International Conference on Predictive Models in Software Engineering, PROMISE '14*, pages 82–91, New York, NY, USA, 2014. ACM.
- [199] Laura Uusitalo. Advantages and challenges of bayesian networks in environmental modeling. *Ecological Modelling*, 203(3–4):312 – 318, 2007.
- [200] Steven van Dijk, Linda C. van der Gaag, and Dirk Thierens. A skeleton-based approach to learning bayesian networks from data. In Nada Lavrač, Dragan Gamberger, Ljupčo Todorovski, and Hendrik Blockeel, editors, *Knowledge Discovery in Databases: PKDD 2003*, pages 132–143, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.
- [201] K. Verbert, R. Babuška, and B. De Schutter. Bayesian and dempster–shafer reasoning for

- knowledge-based fault diagnosis—a comparative study. *Engineering Applications of Artificial Intelligence*, 60:136 – 150, 2017.
- [202] VersionOne. 11th annual state of agile development survey results. <https://explore.versionone.com/state-of-agile/versionone-11th-annual-state-of-agile-report-2>, 2017. Acessado em: 01-02-2017.
- [203] Edwin Villanueva and Carlos Dias Maciel. Efficient methods for learning bayesian network super-structures. *Neurocomputing*, 123:3 – 12, 2014. Contains Special issue articles: Advances in Pattern Recognition Applications and Methods.
- [204] S. Wagner, K. Lochmann, L. Heinemann, M. Kläs, A. Trendowicz, R. Plösch, A. Seidi, A. Goeb, and J. Streit. The quamoco product quality modelling and assessment approach. In *2012 34th International Conference on Software Engineering (ICSE)*, pages 1133–1142, June 2012.
- [205] Andreas P Weigel, Mark A Liniger, and Christof Appenzeller. The discrete brier and ranked probability skill scores. *Monthly Weather Review*, 135(1):118–124, 2007.
- [206] Emily Weimar, Ariadi Nugroho, Joost Visser, and Aske Plaat. Towards high performance software teamwork. In *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering, EASE '13*, pages 212–215, New York, NY, USA, 2013. ACM.
- [207] Christoph Werner, Tim Bedford, Roger M. Cooke, Anca M. Hanea, and Oswaldo Morales-Nápoles. Expert judgement for dependence in probabilistic modelling: A systematic literature review and future research directions. *European Journal of Operational Research*, 258(3):801 – 819, 2017.
- [208] Dave West and Tom Grant. Agile Development: Mainstream Adoption Has Changed Agility – Trends In Real-World Adoption Of Agile Methods. Technical report, Forrester Research, January 2010.
- [209] Roel Wieringa, Neil Maiden, Nancy Mead, and Colette Rolland. Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. *Requirements Engineering*, 11(1):102–107, 2006.

- [210] Roel J. Wieringa. *Design Science Methodology for Information Systems and Software Engineering*. Springer-Verlag Berlin Heidelberg, 1 edition, 2014.
- [211] Renan Willamy, João Nunes, Mirko Perkusich, Arthur Freire, Renata Saraiva, Hyggo Almeida, and Angelo Perkusich. A method to build bayesian networks based on artifacts and metrics to assess agile project. In *27th International Conference on Software Engineering and Knowledge Engineering, SEKE 2016*, pages 81 – 86, San Francisco, USA, 2016.
- [212] L. Williams, G. Brown, A. Meltzer, and N. Nagappan. Scrum + engineering practices: Experiences of three microsoft teams. In *2011 International Symposium on Empirical Software Engineering and Measurement*, pages 463–471, Sept 2011.
- [213] Laurie Williams. Agile software development methodologies and practices. In Marvin V. Zelkowitz, editor, *Advances in Computers*, volume 80 of *Advances in Computers*, pages 1 – 44. Elsevier, 2010.
- [214] Laurie Williams. What agile teams think of agile principles. *Commun. ACM*, 55(4):71–76, April 2012.
- [215] Laurie Williams, William Krebs, Lucas Layman, Annie I. Antón, and Pekka Abrahamsson. Toward a framework for evaluating extreme programming. In *in Empirical Assessment in Software Eng. (EASE) 2004*, pages 11–20, 2004.
- [216] Laurie Williams, Kenny Rubin, and Mike Cohn. Driving process improvement via comparative agility assessment. In *Proceedings of the 2010 Agile Conference, AGILE '10*, pages 3–10, Washington, DC, USA, 2010. IEEE Computer Society.
- [217] BW Wisse, Sicco Pier van Gosliga, Nicole P van Elst, and Ana Isabel Barros. Relieving the elicitation burden of bayesian belief networks. In *BMA*, 2008.
- [218] Y. Xiang and N. Jia. Modeling causal reinforcement and undermining for efficient cpt elicitation. *IEEE Transactions on Knowledge and Data Engineering*, 19(12):1708–1718, Dec 2007.
- [219] H. Xiao-xuan, W. Hui, and W. Shuo. Using expert’s knowledge to build bayesian networks. In *2007 International Conference on Computational Intelligence and Security Workshops (CISW 2007)*, pages 220–223, Dec 2007.

- [220] Ronald R Yager. On the dempster-shafer framework and new combination rules. *Information sciences*, 41(2):93–137, 1987.
- [221] Barbaros Yet, Zane Perkins, Norman Fenton, Nigel Tai, and William Marsh. Not just data: A method for improving prediction with knowledge. *Journal of Biomedical Informatics*, 48:28 – 37, 2014.
- [222] Barbaros Yet, Zane B. Perkins, Todd E. Rasmussen, Nigel R.M. Tai, and D. William R. Marsh. Combining data and meta-analysis to build bayesian networks for clinical decision support. *Journal of Biomedical Informatics*, 52:373 – 385, 2014. Special Section: Methods in Clinical Research Informatics.
- [223] Alexandre Yin, Soraia Figueiredo, and Miguel Mira da Silva. Scrum maturity model: Validation for it organizations’ roadmap to develop software centered on the client role. In *Proceedings of the Sixth International Conference on Software Engineering Advances, ICSEA 2011*, pages 20–29, Barcelona, Spain, 2011.
- [224] Adam Zagorecki and Marek J Druzdzel. An empirical study of probability elicitation under noisy-or assumption. In *Flairs conference*, pages 880–886, 2004.
- [225] Yinghua Zhang, Wensheng Zhang, and Yuan Xie. Improved heuristic equivalent search algorithm based on maximal information coefficient for bayesian network structure learning. *Neurocomputing*, 117:186 – 195, 2013.
- [226] Yun Zhou, Norman Fenton, and Martin Neil. Bayesian network approach to multinomial parameter learning using data and expert judgments. *International Journal of Approximate Reasoning*, 55(5):1252 – 1268, 2014.
- [227] Yun Zhou, Norman Fenton, and Cheng Zhu. An empirical study of bayesian network parameter learning with monotonic influence constraints. *Decision Support Systems*, 87:69 – 79, 2016.
- [228] Yun Zhou, Timothy M. Hospedales, and Norman Fenton. When and where to transfer for bayesian network parameter learning. *Expert Systems with Applications*, 55:361 – 373, 2016.

Apêndice A

Perguntas do questionário

As perguntas do questionário são apresentadas nas páginas a seguir.

Scrum Assessment

Objetivo:

- Coletar dados para calibração e validação de um modelo de previsão de satisfação do cliente com um projeto Scrum de desenvolvimento de software.
- Além da previsão, o mesmo possibilita o diagnóstico de potenciais falhas no processo de entrega e auxilia na melhoria contínua da equipe.
- Possibilitar a criação de uma base de dados do Virtus para aprendizagem contínua em função de lições aprendidas em projetos passados.

Instruções:

- Algumas questões acompanham um glossário. Utilize-o para se familiarizar com a terminologia utilizada.
- A avaliação refere-se à última sprint finalizada.
- As informações coletadas não serão apresentadas para o cliente, equipe, diretoria ou outra parte interessada do projeto.

*Obrigatório

Dados demográficos

1. Qual seu papel no projeto? (Selecione todos os papéis que condizem com suas responsabilidades) *

Marque todas que se aplicam.

- Product Owner
- Scrum Master
- Gerente do Projeto
- Coordenador do Projeto
- Líder técnico
- Desenvolvedor
- Testador

2. Qual a sua formação acadêmica? (marque todos os títulos obtidos) *

Marque todas que se aplicam.

- Graduação
- Especialização
- Mestrado
- Doutorado

3. Você tem quantos anos de experiência trabalhando em projetos de desenvolvimento de software? *

4. Você tem quantos anos de experiência trabalhando em projetos ágeis de desenvolvimento de software? *

5. Você tem quantos anos de experiência trabalhando em projetos Scrum de desenvolvimento de software? *

Contexto do projeto

6. Qual o tipo de entrega do projeto? *

Marcar apenas uma oval.

- Protótipo
- Produto

7. Qual o tipo de sistema desenvolvido no projeto? *

Marque todas que se aplicam.

- Sistema e software automotivo
- Aviação
- Aplicativos e serviços cloud (e.g., SaaS)
- Sistema de defesa
- Serviços financeiros (e.g., banco, seguros)
- Jogos
- Automação residencial
- Logística e transporte
- Mídia e entretenimento (e.g., imagens e TV)
- Dispositivos de saúde e médico
- Aplicativos móveis
- Outros sistemas e serviços embarcados
- Outros sistemas de informação (e.g., SAP, ERP, etc.)
- Setor público (e.g., licitações)
- Robótica (e.g., robôs autônomos, drones)
- Sistemas espaciais
- Telecomunicações
- Aplicativos e serviços Web (e.g., e-commerce e portais)
- Ferramentas de build
- Geradores de código
- Frameworks
- Outro: _____

8. No caso de falha do sistema, quais as possíveis consequências? *

Marque todas que se aplicam.

- Ameaça à saúde ou vida humana
- Perda total do sistema
- Degradação do sistema (ou serviço)
- Prejuízo financeiro
- Consequências legais (civil)
- Consequências legais (criminal)
- Ameaça ao meio ambiente
- Ameaça à reputação da empresa
- Ameaça aos negócios da empresa
- Outro: _____

9. Qual o tamanho do software produzido em KLOC? *

10. Qual(is) a(s) linguagem(ns) de programação usada(s)? *

Marque todas que se aplicam.

- Java
- C++
- Python
- C
- Swift
- Kotlin
- PHP
- JavaScript
- Objective-C
- Ada
- Scala
- TypeScript
- Go
- C#
- CSS
- Ruby

11. Quantas sprints estão planejadas para o projeto em questão? *

12. Qual o número da sprint em questão? *

13. Qual é o tamanho da equipe (incluindo o líder técnico e testadores)? *

14. **Quais métodos ágeis foram utilizados no projeto? ***

Marque todas que se aplicam.

- Scrum
- Extreme Programming (XP)
- OpenUP
- Kanban
- Scrumban
- Lean
- Crystal
- DSDM
- Outro: _____

15. **O quão complexo são os requisitos do projeto? ***

Marcar apenas uma oval.

- Muito
- Moderado
- Pouco
- Não sei

16. **O quão estável é o ambiente de negócio? (Quanto mais fatores influenciarem os requisitos, como concorrentes e dependência com agências ou sistemas externos, menos estável o ambiente de negócio) ***

Marcar apenas uma oval.

- Instável
- Moderado
- Estável
- Não sei

17. **O quão inovador são os requisitos do projeto? ***

Marcar apenas uma oval.

- Muito
- Moderado
- Pouco
- Não sei

Product Owner

18. **Você concorda com a seguinte afirmação: "O Product Owner do projeto é comprometido em colaborar com a equipe de desenvolvimento para maximizar o valor do produto resultante do projeto?" ***

Marcar apenas uma oval.

- Discordo completamente
- Concordo parcialmente
- Concordo completamente
- Não sei

Reuniões de Revisão

19. **Você concorda com a seguinte afirmação: "Durante a Reunião de Revisão, que ocorrem no máximo a cada um mês, as novas funcionalidades do Incremento são avaliadas de acordo com os critérios de pronto por meio de uma demonstração."? ***

Marcar apenas uma oval.

- Discordo completamente
- Concordo parcialmente
- Concordo completamente
- Não sei

20. **Você concorda com a seguinte afirmação: "Durante a Reunião de Revisão, que ocorrem no máximo a cada um mês, quando necessário, o progresso da equipe é apresentado de forma visual e o mesmo é avaliado e metas projetadas."? ***

Marcar apenas uma oval.

- Discordo completamente
- Concordo parcialmente
- Concordo completamente
- Não sei

21. **Você concorda com a seguinte afirmação: "Durante a Reunião de Revisão, que ocorrem no máximo a cada um mês, o Backlog do produto é adaptado de acordo com a inspeção do incremento, planejamento do projeto e, possivelmente, mudanças no âmbito de negócio."? ***

Marcar apenas uma oval.

- Discordo completamente
- Concordo parcialmente
- Concordo completamente
- Não sei

22. **Você concorda com a seguinte afirmação: "A Reunião de Revisão alcança seu objetivo de inspecionar o produto e adaptá-lo para maximizar o valor entregue do projeto com a frequência necessária."? ***

Marcar apenas uma oval.

- Discordo completamente
- Concordo parcialmente
- Concordo completamente
- Não sei

Backlog do Produto

23. **Qual o nível de detalhamento do Backlog do Produto no início do ciclo de release? ***

Marcar apenas uma oval.

- Poucas ou nenhuma funcionalidade(s) definida(s)
- Funcionalidades principais definidas
- Definição completa dos requisitos e funcionalidades

24. **Você concorda com a seguinte afirmação: "Os requisitos funcionais e não-funcionais contém um conjunto de critérios de completude e são representados em diferentes níveis de detalhamento de acordo com o quão perto estão de serem implementados (itens do topo do backlog são mais detalhados)."? ***

Marcar apenas uma oval.

- Discordo completamente
- Concordo parcialmente
- Concordo completamente
- Não sei

25. **Você concorda com a seguinte afirmação: "Todo o backlog do produto foi estimado de forma colaborativa pelas pessoas que realizarão o trabalho. Os itens de backlog do topo são pequenos o suficiente para serem alocados em uma sprint. Os itens de backlog de menor prioridade, não necessariamente, pois só deverão ser refinados quando necessário."? ***

Marcar apenas uma oval.

- Discordo completamente
- Concordo parcialmente
- Concordo completamente
- Não sei

26. **O Backlog do produto é ordenado continuamente considerando variáveis como valor de negócio, dependências técnicas, esforço/custo e risco? ***

Marcar apenas uma oval.

- O Backlog não é ordenado corretamente
- O Backlog é ordenado continuamente usando algumas das variáveis relevantes para o projeto.
- O Backlog é ordenado continuamente considerando todas as variáveis relevantes para o projeto.
- Não sei

27. **Você concorda com a seguinte afirmação: "O Backlog do Produto é uma lista ordenada que representa todas funcionalidades, funções, requisitos, melhorias e correções a serem realizadas no produto que evolui durante o projeto.?" ***

Marcar apenas uma oval.

- Discordo completamente
- Concordo parcialmente
- Concordo completamente
- Não sei

28. **Você concorda com a seguinte afirmação: "O Backlog do produto é atualizado e refinado continuamente durante a execução do projeto de forma efetiva.?" ***

Marcar apenas uma oval.

- Discordo completamente
- Concordo parcialmente
- Concordo completamente
- Não sei

Reunião de Planejamento

29. **Você concorda com a seguinte afirmação: "A equipe sabe sua velocidade e utiliza-a como parâmetro para planejar a próxima sprint.?" ***

Marcar apenas uma oval.

- Discordo completamente
- Concordo parcialmente
- Concordo completamente
- Não sei

30. **Você concorda com a seguinte afirmação: "A equipe leva em consideração a capacidade (por exemplo: disponibilidade dos membros) para planejar a próxima sprint."? ***

Marcar apenas uma oval.

- Discordo completamente
- Concordo parcialmente
- Concordo completamente
- Não sei

31. **Você concorda com a seguinte afirmação: "O Backlog da sprint é definido por meio de uma negociação entre o Product Owner e a equipe. O mesmo define o que será entregue a final da sprint e como o mesmo será implementado. Inclui, pelo menos, um item de melhoria de processo identificado na Reunião de Retrospectiva "? ***

Marcar apenas uma oval.

- Discordo completamente
- Concordo parcialmente
- Concordo completamente
- Não sei

32. **Você concorda com a seguinte afirmação. "O objetivo da sprint é coerente, amplo e realista. O mesmo representa uma evolução na estratégia de entrega de produto."? ***

Marcar apenas uma oval.

- Discordo completamente
- Concordo parcialmente
- Concordo completamente
- Não sei

33. **Você concorda com a seguinte afirmação: "O planejamento segue os princípios de Just-in-Time (JIT) e a equipe tem confiança de conseguir alcançar a meta definida"? ***

Marcar apenas uma oval.

- Discordo completamente
- Concordo parcialmente
- Concordo totalmente
- Não sei

Trabalho em Equipe

34. **Você concorda com a seguinte afirmação: "A equipe não precisa trabalhar em atividades que não adicionem valor ao projeto em questão e nem sofre influência de agentes externos durante a sprint".? ***

Marcar apenas uma oval.

- Discordo completamente
- Concordo parcialmente
- Concordo completamente
- Não sei

35. **Você concorda com a seguinte afirmação: "A equipe possui membros com boa atração interpessoal entre si, comprometimento com as atividades, e espírito de equipe."? ***

Marcar apenas uma oval.

- Discordo completamente
- Concordo parcialmente
- Concordo completamente
- Não sei

36. **Você concorda com a seguinte afirmação: "A equipe utiliza bons métodos para tomar as decisões e completar seu trabalho, ao invés de ser dirigida por um agente externo."? ***

Marcar apenas uma oval.

- Discordo completamente
- Concordo parcialmente
- Concordo completamente
- Não sei

37. **Você concorda com a seguinte afirmação: "A equipe é capaz de identificar mudanças no ambiente e, rapidamente, ajustar-se as estratégias."? ***

Marcar apenas uma oval.

- Discordo completamente
- Concordo parcialmente
- Concordo completamente
- Não sei

38. **Você concorda que, no âmbito da equipe de desenvolvimento, a autoridade e liderança são compartilhadas? ***

Marcar apenas uma oval.

- Discordo completamente
- Concordo parcialmente
- Concordo completamente
- Não sei

39. **Você concorda com a seguinte afirmação: "A equipe contém experiência trabalhando com projetos ágeis e todo o conhecimento técnico para o desenvolvimento do produto."? ***

Marcar apenas uma oval.

- Discordo completamente
- Concordo parcialmente
- Concordo completamente
- Não sei

40. **Você concorda com a seguinte afirmação: "Os membros da equipe possuem boa colaboração entre si, sendo eles responsáveis por código, projeto e por contribuir com novas ideias."? ***

Marcar apenas uma oval.

- Discordo completamente
- Concordo parcialmente
- Concordo completamente
- Não sei

41. **Você concorda com a seguinte afirmação: "Os membros da equipe possuem respeito mútuo e priorizam os objetivos da equipe sobre os individuais."? ***

Marcar apenas uma oval.

- Discordo completamente
- Concordo parcialmente
- Concordo completamente
- Não sei

42. **Você concorda com a seguinte afirmação: "A combinação de personalidades permite a atitudes positivas com relação à fluidez do processo e evita conflitos."? ***

Marcar apenas uma oval.

- Discordo completamente
- Concordo parcialmente
- Concordo totalmente
- Não sei

43. **Você concorda com a seguinte afirmação: "Os membros da equipe comunicam-se entre si e com agentes externos de forma rápida e informal, utilizando documentação para complementar (não substituir) a comunicação quando necessário"? ***

Marcar apenas uma oval.

- Discordo completamente
- Concordo parcialmente
- Concordo completamente
- Não sei

44. **Você concorda com a seguinte afirmação: "A distribuição da equipe, do ponto de vista de localização, está de acordo com a necessidade de colaboração e interação entre os membros."? ***

Marcar apenas uma oval.

- Discordo completamente
- Concordo parcialmente
- Concordo completamente
- Não sei

45. **Você concorda com a seguinte afirmação: "Os canais de comunicação promovem o compartilhamento de informações entre todos os envolvidos no projeto de forma satisfatória do ponto de vista de completude, clareza e tempo."? ***

Marcar apenas uma oval.

- Discordo completamente
- Concordo parcialmente
- Concordo completamente
- Não sei

46. **Você concorda com as seguintes observações com relação às Reuniões Diárias? ***

Glossário	
Objetivo de Reunião Diária	As reuniões diárias devem sincronizar as atividades restantes do backlog, identificar e remover impedimentos, promover tomada rápida de decisão e melhorar o nível de conhecimento da equipe.
Taxa de Presença	Todos os membros do time Scrum (ScrumMaster, equipe de desenvolvimento e Product Owner) devem estar presentes em todas as reuniões diárias.
Atividades de Monitoramento	O time deve realizar atividades de monitoramento como inspeção do progresso em direção ao objetivo da sprint e verificação da possibilidade de completar o trabalho do backlog da sprint considerando o atual progresso.

Marcar apenas uma oval por linha.

	Discordo completamente	Concordo parcialmente	Concordo completamente	Não sei
Reunião alcançou seus objetivos.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A taxa de presença foi adequada.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
As atividades de monitoramento foram realizadas adequadamente.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

47. **Você concorda com a seguinte afirmação: "A equipe de desenvolvimento é competente e coesa para realizar todas as atividades necessárias para atingir os objetivos do projeto".? ***

Marcar apenas uma oval.

- Discordo completamente
- Concordo parcialmente
- Concordo completamente
- Não sei

Processo de entrega

48. **Você concorda com a seguinte afirmação: "O processo de desenvolvimento é eficiente (i.e., boa produtividade), efetivo (i.e., entrega produtos com boa qualidade) e o custo operacional é baixo devido à automação."? ***

Marcar apenas uma oval.

- Discordo completamente
- Concordo parcialmente
- Concordo completamente

49. **Você concorda com a seguinte afirmação: "A execução da sprint é eficiente e eficaz para alcançar os objetivos da mesma."? ***

Marcar apenas uma oval.

- Discordo completamente
- Concordo parcialmente
- Concordo completamente
- Não sei

50. **Você concorda com a seguinte afirmação: "A equipe tem um software em funcionamento de alta qualidade (pouco ou nenhum débito técnico) e que pode ser confortavelmente testado por pessoas fora da equipe."? ***

Marcar apenas uma oval.

- Discordo completamente
- Concordo parcialmente
- Concordo completamente
- Não sei

51. **Você concorda com a seguinte afirmação: "O objetivo da sprint foi alcançado, do ponto de vista de itens de Backlog entregues." ***

Marcar apenas uma oval.

- Discordo completamente
- Concordo parcialmente
- Concordo completamente
- Não sei

52. **Você concorda com a seguinte afirmação: "O objetivo da sprint foi alcançado, do ponto de vista de quantidade de itens de Backlog entregues do incremento." ***

Marcar apenas uma oval.

- Discordo completamente
- Concordo parcialmente
- Concordo completamente
- Não sei

Satisfação do cliente

53. **Você concorda com a seguinte afirmação: "O cliente está satisfeito com o incremento atual do produto, do ponto de vista de funcionalidades implementadas e qualidade." ***

Marcar apenas uma oval.

- Discordo completamente
- Concordo parcialmente
- Concordo completamente
- Não sei

Powered by



Apêndice B

Respostas do questionário

As respostas do questionário, mapeadas para os nós e estados da rede Bayesiana, são apresentadas nas páginas a seguir.

Projeto	Complexidade dos requisitos	Inovação dos requisitos	Ambiente de negócio
1	Moderado	Moderado	Moderado
2	Alto	Moderado	Alto
3	Moderado	Moderado	Moderado
4	Moderado	Baixo	Moderado
5	Moderado	Moderado	Moderado
6	Moderado	Alto	Moderado
7	Alto	Moderado	Moderado
8	Baixo	Baixo	Baixo
9	Moderado	Moderado	Alto
10	Moderado	Moderado	Baixo
11	Moderado	Baixo	Moderado
12	Alto	Baixo	Moderado
13	Baixo	Moderado	Moderado
14	Baixo	Baixo	Moderado
15	Baixo	Baixo	Baixo
16	Baixo	Baixo	Baixo
17	Moderado	Moderado	Moderado
18	Alto	Alto	Alto

Projeto	Qualidade do Product Owner	Inspeção	Monitoramento do progresso	Adaptação
1	Moderado	Alto	Alto	Alto
2	Alto	Moderado	Alto	Alto
3	Alto	Alto	Moderado	Moderado
4	Alto	Moderado	Moderado	Moderado
5	Moderado	Baixo	Baixo	Baixo
6	Alto	Moderado	Moderado	Moderado
7	Moderado	Moderado	Moderado	Alto
8	Moderado	Alto	Moderado	Alto
9	Alto	Alto	Moderado	Alto
10	Alto	Alto	Alto	Alto
11	Moderado	Alto	Alto	Alto
12	Alto	Alto	Baixo	Alto
13	Alto	Moderado	Moderado	Alto
14	Moderado	Alto	Alto	Moderado
15	Alto	Alto	Moderado	Alto
16	Alto	Alto	Moderado	Alto
17	Alto	Alto	Moderado	Moderado
18	Alto	Alto	Moderado	Alto

Projeto	Detalhamento inicial do Backlog do produto	Detalhamento	Estimado	Ordenado	Desempenho passado
1	Alto	Não sei	Alto	Moderado	Alto
2	Alto	Moderado	Moderado	Alto	Alto
3	Alto	Baixo	Alto	Moderado	Alto
4	Alto	Baixo	Alto	Moderado	Moderado
5	Alto	Moderado	Moderado	Moderado	Alto
6	Alto	Alto	Alto	Moderado	Moderado
7	Alto	Moderado	Baixo	Baixo	Moderado
8	Moderado	Alto	Moderado	Moderado	Moderado
9	Alto	Moderado	Alto	Alto	Alto
10	Alto	Moderado	Alto	Moderado	Alto
11	Alto	Moderado	Alto	Alto	Alto
12	Alto	Alto	Baixo	Alto	Alto
13	Alto	Moderado	Moderado	Baixo	Moderado
14	Alto	Alto	Alto	Alto	Alto
15	Alto	Alto	Moderado	Alto	Alto
16	Alto	Moderado	Alto	Alto	Alto
17	Alto	Alto	Alto	Alto	Alto
18	Moderado	Moderado	Baixo	Alto	Baixo

Projeto	Capacidade projetada	Backlog da sprint	Objetivo da sprint	Autonomia	Adaptabilidade
1	Alto	Moderado	Alto	Alto	Moderado
2	Alto	Moderado	Alto	Alto	Baixo
3	Alto	Moderado	Baixo	Alto	Moderado
4	Alto	Moderado	Moderado	Moderado	Alto
5	Alto	Moderado	Alto	Baixo	Moderado
6	Moderado	Moderado	Alto	Alto	Alto
7	Alto	Moderado	Alto	Baixo	Moderado
8	Moderado	Moderado	Moderado	Baixo	Moderado
9	Alto	Moderado	Alto	Moderado	Moderado
10	Alto	Moderado	Alto	Alto	Alto
11	Alto	Alto	Moderado	Moderado	Alto
12	Moderado	Baixo	Alto	Alto	Moderado
13	Alto	Moderado	Moderado	Baixo	Moderado
14	Moderado	Alto	Alto	Moderado	Alto
15	Alto	Moderado	Moderado	Alto	Alto
16	Alto	Alto	Alto	Alto	Alto
17	Alto	Alto	Alto	Baixo	Alto
18	Baixo	Baixo	Alto	Alto	Moderado

Projeto	Compartilhamento da liderança	Conhecimento	Personalidade	Distribuição	Canais
1	Alto	Moderado	Alto	Alto	Alto
2	Baixo	Moderado	Moderado	Moderado	Moderado
3	Moderado	Alto	Alto	Alto	Alto
4	Alto	Alto	Alto	Alto	Alto
5	Moderado	Moderado	Alto	Moderado	Moderado
6	Alto	Moderado	Alto	Alto	Moderado
7	Alto	Alto	Alto	Alto	Alto
8	Moderado	Baixo	Alto	Alto	Alto
9	Moderado	Alto	Alto	Moderado	Moderado
10	Moderado	Moderado	Alto	Alto	Alto
11	Alto	Alto	Alto	Alto	Moderado
12	Moderado	Baixo	Moderado	Alto	Moderado
13	Moderado	Moderado	Moderado	Alto	Moderado
14	Alto	Alto	Alto	Alto	Alto
15	Alto	Alto	Alto	Alto	Alto
16	Alto	Alto	Alto	Alto	Alto
17	Moderado	Alto	Alto	Alto	Alto
18	Alto	Moderado	Alto	Alto	Alto

Projeto	Presença	Sincronização	Qualidade do processo de desenvolvimento e testes	Satisfação do cliente
1	Moderado	Moderado	Moderado	Alto
2	Moderado	Alto	Moderado	Alto
3	Moderado	Alto	Moderado	Alto
4	Moderado	Alto	Moderado	Alto
5	Moderado	Moderado	Moderado	Moderado
6	Alto	Moderado	Moderado	Alto
7	Moderado	Alto	Alto	Alto
8	Moderado	Alto	Baixo	Alto
9	Baixo	Baixo	Alto	Alto
10	Alto	Alto	Moderado	Alto
11	Alto	Alto	Alto	Alto
12	Baixo	Alto	Moderado	Alto
13	Moderado	Moderado	Baixo	Alto
14	Alto	Alto	Alto	Alto
15	Moderado	Alto	Alto	Alto
16	Moderado	Alto	Alto	Alto
17	Alto	Alto	Moderado	Alto
18	Baixo	Moderado	Baixo	Moderado