Universidade Federal de Pernambuco
Centro de Informática

Graduação em Sistemas de Informação

Estudo de alternativas ao uso de estimativas de esforço na gestão ágil de projetos de desenvolvimento de software.

Wandecleya Martins de Melo

Trabalho de Graduação

Recife
6 de Dezembro de 2018

Universidade Federal de Pernambuco
Centro de Informática

Wandecleya Martins de Melo

Estudo de alternativas ao uso de estimativas de esforço na gestão ágil de projetos de desenvolvimento de software.

Trabalho apresentado ao Programa de Graduação em Sistemas de Informação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial à obtenção do título de Bacharela em Sistemas de Informação.

Orientadora: *Profa. Dra. Bernadette Farias Lóscio*

Recife
6 de Dezembro de 2018

Dedico o presente trabalho a todas que lutaram e contribuíram ao longo da história para permitir que uma mulher, suburbana, parda, lésbica e pobre tivesse acesso a educação de qualidade, abrindo-lhe assim um infinito de oportunidades.

Agradecimentos

Sou grata a minha orientadora, pela confiança, parceria e apoio, que transcendem o presente trabalho; a todas as professoras, pelos desafios encorajados e inspiração; às amigas e familiares, pela paciência; à namorada pelo incentivo incondicional e às companheiras de turma pelo apoio mútuo, que transformaram uma jornada acadêmica em experiência de vida, que gerou amizades perenes como as de Reginaldo Júnior e Maurício Taumaturgo.

"It’s very hard to estimate things that you don’t know you don’t know."
—MAGNE JØRGENSEN (Scientist in Informatics)

Resumo

Estimativa de esforço consiste na tentativa de prever quanto tempo a execução de uma tarefa vai requerer. Ela é a base do planejamento e controle de boa parte dos projetos de software; a despeito de sua baixa acurácia e potenciais impactos na qualidade do projeto, ao serem utilizadas como balizadores de decisões técnicas; e efeitos emocionais na equipe, ao serem encarados como métrica de produtividade. O presente trabalho investiga alternativas às estimativas de esforço nas diferentes granularidades do processo de gestão ágil: do projeto, das suas entregas, iterações e histórias de usuário. Foram conjugadas entrevistas, questionários e estudos de caso como formas de coleta de dados. O que possibilitou a identificação de alternativas ao uso de estimativas de esforço, em tomadas de decisão na gestão de histórias de usuário, considerando impactos de curto e longo prazo das decisões técnicas e gerando intervenções que propiciem colaboração e confiança na equipe; além de permitir o gerenciamento de iterações, entregas e do projeto como um todo.

Palavras-chave: Movimento NoEstimates, Estimativas de Esforço, Desenvolvimento Iterativo e Incremental, Metodologias Ágeis, Planning Poker, Histórias de Usuário.

Abstract

Effort estimation consists in trying to predict how long a task is going to take. Most software projects rely on effort estimation for planning and control. Despite their low accuracy; potential impact on project quality, when estimates are used for making technical decisions; and emotional effects on the team, when estimates are used as productivity measure. This research explores alternatives to the use of effort estimation in different granularities of agile management: project, delivery, iteration and user story. Data was gathered through interviews, questionnaires and case studies. Which made possible to identify other options to support decision making for user story development, considering short and long term impact of technical decision and allowing interventions that bring collaboration and trust to the team and also allows to manage iterations, deliveries and the project as a whole.

Keywords: NoEstimates Movement, Software Effort Estimation, Iterative and Incremental Development, Agile Methodologies, Planning Poker, User Stories.

Sumário

1 Introdução **1**
1.1 Motivação 1
1.2 Objetivos 2
1.3 Organização do Trabalho 3

2 Fundamentação Teórica **4**
2.1 Metodologias Ágeis 4
2.1.1 Modelo Iterativo Incremental e Evolutivo 4
2.1.2 Princípios Ágeis 6
2.2 Planejamento de Projetos 8
2.2.1 Triângulo das Restrições 8
2.2.2 Planejamento Ágil 9
2.3 Estimativas em Projetos Ágeis 12
2.3.1 Planning Poker 15
2.3.2 Velocidade Bruta 15
2.3.3 Cálculo de Estimativas Derivadas 16
2.4 Movimento #NoEstimates 21

3 Metodologia **23**
3.1 Identificação de Alternativas às Estimativas de Esforço na Gestão de Histórias de Usuários 24
3.1.1 Monitorar e Controlar Progresso do Trabalho 24
3.1.2 Aprofundar Entendimento das Histórias 26
3.2 Identificação de Alternativas às Estimativas de Esforço na Gestão de Iterações 26
3.3 Identificação de Alternativas às Estimativas de Esforço na Gestão de Entregas 27
3.4 Identificação de Alternativas às Estimativas de Esforço na Gestão do Projeto 28
3.5 Coleta de Dados 29
3.5.1 Questionário para Pessoas Desenvolvedoras e Analistas de Qualidade 29
3.5.2 Entrevista para Analistas de Negócio 31
3.5.3 Estudo de Caso 32

4 Resultados **33**
4.1 Pergunta de Pesquisa 1 33
4.2 Pergunta de Pesquisa 2 36
4.3 Pergunta de Pesquisa 3 38
4.4 Pergunta de Pesquisa 4 39

4.5 Pergunta de Pesquisa 5 41
4.6 Pergunta de Pesquisa 6 42
4.7 Pergunta de Pesquisa 7 43

5 Considerações Finais 45

Referências Bibliográficas 48

Lista de Figuras

2.1 Triângulo das restrições 9
2.2 Etapas do planning poker 16

3.1 Grau de detalhamento do planejamento 23
3.2 Processo de controle 25
3.3 Dados destinados às perguntas de pesquisa 30

Lista de Tabelas

3.1 Relação estimativa-informação 24

4.1 Decisões técnicas baseadas em estimativas 34
4.2 Métodos alternativos de auxílio a decisões técnicas 34
4.3 Decisões gerenciais baseadas em estimativas 37
4.4 Técnicas alternativas de auxílio a decisões gerenciais 37
4.5 Comparação do entendimento de negócio das histórias pela Equipe A 39
4.6 Comparação do entendimento de negócio das histórias pela Equipe B 39
4.7 Comparação do entendimento técnico das histórias pela Equipe A 40
4.8 Comparação do entendimento técnico das histórias pela Equipe B 41
4.9 Técnicas alternativas de gerenciamento de iterações 42
4.10 Técnicas alternativas de gerenciamento de entregas 43
4.11 Técnicas alternativas de gerenciamento de projeto 43

CAPÍTULO 1

Introdução

1.1 Motivação

A utilidade de se prever o futuro é inquestionável. No âmbito do gerenciamento de projetos de software são muitas as decisões a serem tomadas que se beneficiariam de certa dose de previsibilidade. Buscando tal previsibilidade, os projetos tentam prever quanto tempo levará a realização de cada tarefa, gerando estimativas de esforço, que embasam estimativas de prazo, orçamento e capacidade de entrega de escopo. É questionável, no entanto, quão bem sucedido tem sido o planejamento de projetos baseado em estimativas de esforço e quais os impactos dessa abordagem.

Os relatórios *CHAOS*, publicados desde 1994 pelo *Standish Group International* expõem um retrato alarmante dos projetos de software. De 1994 a 2015, não passaram de 35% o número de projetos que cumpriram as estimativas iniciais de prazo, escopo e orçamento. Um indício de estimativas iniciais otimistas, que não se provaram factíveis. Muitos dos projetos são cancelados sem se quer entregar qualquer valor aos clientes, causando prejuízos da ordem de bilhões de dólares anualmente [1].

As metodologias ágeis ganharam popularidade respondendo a tais desafios de previsibilidade do planejamento de projetos de software privilegiando a capacidade de aprendizagem e adaptação. No ágil, o planejamento permeia toda a duração do projeto, aprofundando a análise do negócio gradualmente. A partir de um entendimento de alto nível do que o projeto se propõe, realiza-se uma atividade de priorização para selecionar o que irá compor a primeira entrega e se concentra a análise inicial nesse subconjunto de funcionalidades; novamente se prioriza no

que focar o trabalho no curto prazo, chamado de iteração, que corresponde ao horizonte de semanas, e se detalha ainda mais as atividades priorizadas. Essas atividades são descritas como histórias de usuário, que são a unidade básica de trabalho.

Em paralelo, em 2012, o movimento *#NoEstimates*, protagonizado por Woody Zuill, Vasco Duarte e Neil Killick, passou a questionar o próprio uso das estimativas; o seu impacto nas equipes, que sentem a pressão emocional e a cobrança por concretizar estimativas, frequentemente não realistas [2]; além do impacto no próprio desenvolvimento do projeto, quando se toma decisões visando cumprir as estimativas de esforço de cada tarefa, um objetivo de curto prazo, e se ignora efeitos de longo prazo, como o comprometimento da qualidade que pode desacelerar o projeto no longo prazo.

Problema

Prover informações para tomadas de decisão da gestão ágil de projetos de desenvolvimento de software prescindindo estimativas de esforço.

1.2 Objetivos

Objetivo Geral

Identificar alternativas ao uso de estimativas de esforço na gestão ágil de projetos de desenvolvimento de software.

Objetivos Específicos

- Identificar alternativas às estimativas de esforço na gestão de histórias de usuário.
- Identificar alternativas às estimativas de esforço na gestão de iterações.
- Identificar alternativas às estimativas de esforço na gestão de entregas.
- Identificar alternativas às estimativas de esforço na gestão do projeto.

1.3 Organização do Trabalho

Discorre-se, no primeiro momento, a respeito das metodologias ágeis, descrevendo suas características iterativas, incrementais e evolutivas e seu princípios. O planejamento dos projetos é então descrito sob o prisma do triângulo das restrições e da abordagem ágil.

Em seguida, o foco se volta às estimativas, partindo do seu papel no gerenciamento de projetos. A análise bibliográfica explora métricas, escalas e técnicas. São detalhadas a técnica de *Planning Poker* para estimativa de esforço, e de velocidade bruta para estimativa inicial da capacidade de entrega de uma equipe. Demonstram-se os cálculos das estimativas de variáveis de controle de projetos: escopo, prazo e orçamento. Por fim, são discutidas os questionamentos oriundos do movimento *#NoEstimates*.

Em seguida é descrita a metodologia de investigação dos objetivos de pesquisa: as perguntas de pesquisa, os métodos de coleta de dados e relação entre eles.

Expõem-se, então, os resultados: os dados coletados e a análise de como levaram a responder às perguntas de pesquisa atingindo o objetivo do trabalho de identificar alternativas ao uso de estimativas de esforço no contexto do gerenciamento de projetos ágeis de software.

CAPÍTULO 2

Fundamentação Teórica

2.1 Metodologias Ágeis

Desenvolvimento de software é uma atividade de difícil previsibilidade, tendo em vista a complexidade de se traduzir as necessidades de negócio em requisitos técnicos e efetivamente implementá-los em um sistema que atenda aos mais variados critérios de qualidade em um contexto de mudanças corriqueiras em aspectos tecnológicos, de negócio e da própria equipe. A dificuldade de se estimar, planejar e controlar projetos de desenvolvimento de software suscitou o surgimento de abordagens que valorizam a capacidade de se adaptar a mudanças mais do que prevê-las ou evitá-las, as metodologias ágeis, cujas técnicas se assentam no modelo iterativo incremental e evolutivo.

2.1.1 Modelo Iterativo Incremental e Evolutivo

De modo geral, podemos citar como características endossadas por modelos iterativos, incrementais e evolutivos:

Iterativo: Definição de períodos de tempo delimitados, as iterações, dedicados a uma parcela do projeto.

Colaborativo: Integração das atividades de análise, design, desenvolvimento e validação.

Testável: Implementação de porções de funcionalidades passíveis de verificação.

Incremental: Implementação simplificada das funcionalidades, continuamente revisitadas e aprimoradas.

Entregável: Entrega recorrente de subconjuntos de funcionalidades.

Evolutivo: Coleta recorrente de feedback dos usuários com aplicação dos aprendizados.

Ao segmentar o trabalho em iterações, é possível focar esforços tanto de análise, quanto de desenvolvimento em um grupo de funcionalidades e obter aprendizados que devem adaptar as abordagens da iteração seguinte. Uma iteração que traga aprendizados tem importância equivalente a iteração que entregue valor de negócio.

Ao organizar o trabalho de forma incremental e evolucionária, busca-se disponibilizar funcionalidades para obter feedback precoce dos usuários reais que oriente a priorização de adição de novas funcionalidades e outras qualidades não-funcionais ao software.

As ideias que caracterizam modelos iterativos e incrementais surgiram de forma independente em diversos projetos de software pelo mundo, com registros que remontam à década de 50[4]. Mesmo o artigo, conhecido como precursor do modelo sequencial, tem uma menção embrionária à iteração, ao sugerir que o processo seja percorrido duas vezes [3].

A etapa de desenvolvimento foi a primeira a ser encarada como uma tarefa que poderia ser cíclica, de modo a obter aprendizados e aplicá-los em seguida. Ainda mantendo tanto o planejamento quanto o teste e a entrega como fases independentes no início e no fim do projeto, respectivamente.

Somente incorporando as atividades de teste e implantação do software a cada iteração de desenvolvimento, contudo, se alcançaria o valioso benefício de *feedback* do usuário. Assim, em vez de ter testes e implementação realizados uma única vez, após finalizado todo escopo, tais atividades também poderiam ser executadas à medida que cada iteração de desenvolvimento ocorria.

Também se tornou evidente como o design de software e o desenvolvimento colaboram mu-

tuamente, com novas funcionalidades influenciando decisões de design, tanto quanto o design determina decisões de desenvolvimento. E que, portanto, poderiam deixar de ser etapas isoladas e dialogar continuamente.

Observou-se então, como mesmo para os clientes, não é fácil articular de forma extensiva suas necessidades e como certos detalhes de requisitos só ficam claros quando o desenvolvimento ocorre. E ainda que fosse possível prever e listar todos os requisitos inicialmente, eles podem mudar, e isso ocorre com frequência. Diante desta realidade, o planejamento também deixou de ser uma atividade a ser finalizada no início do processo e passou a permear todo ciclo de vida do projeto.

Há projetos com as mais variadas definições de aspectos iterativos e incrementais. Por exemplo, iterações com duração de meio dia do projeto Mercury da Nasa a seis meses do projeto de integração de engenharia da IBM [4]. Todos, porém, tem um objetivo comum: evitar um ciclo único de planejamento, execução e entrega, de forma a se beneficiar de múltiplos ciclos, para obter aprendizados e aplicá-los rapidamente, aprimorando as estimativas, gerenciando os riscos e otimizando decisões e resultados.

2.1.2 Princípios Ágeis

As Metodologias Ágeis emergiram seguindo os princípios iterativo, incremental e evolutivo, fixando contudo o tamanho das iterações para o horizonte de semanas e levando a ideia de colaboração ao extremo em todas as suas atividades. Algumas das mais conhecidas e adotadas metodologias ágeis são as seguintes:

- eXtreme Programming (Kent Beck - 1999).
- Feature Driven Development (Jeff De Luca - 1999).
- Scrum (Ken Schwaber e Mike Beedle - 2001).

- Implementação Ágil do Rational Unified Process (Craig Larmann - 2002).
- Crystal – (Alistair Cockburn - 2004)

Cada uma dessas metodologias propõe um conjunto de boas práticas para ajudar projetos de desenvolvimento de software a obter sucesso. Extreme Programming, por exemplo, preconiza a programação em par, o refatoramento de código, o desenvolvimento dirigido a testes e a entrega contínua de software. O Scrum enfatiza recomendações de cerimônias: as retrospectivas, as reuniões diárias de alinhamento da equipe e a apresentação dos resultados ao cliente ao fim de cada iteração [5].

A despeito de variações de nomenclatura e de sistematizarem de forma diferente práticas para otimizar um projeto de desenvolvimento de software, no cerne de cada proposta havia crenças, valores e objetivos muito semelhantes. O que permitiu a convergência de todos em um movimento unificado: o ágil.

Em 2001, Robert C. Martin reuniu proponentes de metodologias iterativas e incrementais "leves", termo usado na época para enfatizar a simplicidade e foco em princípios de tais métodos. O resultado desse encontro foi o Manifesto Ágil, que sintetizou os aprendizados obtidos pela experiência de todos lá presentes, ao buscarem novas e melhores formas de desenvolver software. São 4 sentenças principais, que são aprofundados em mais 12 princípios, dos quais cito alguns por sua relevância ao presente trabalho [6]:

- Nossa maior prioridade é satisfazer o cliente através 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.

- Construa projetos em torno de indivíduos motivados. Dê a eles o ambiente e o suporte necessário e confie neles para fazer o trabalho.
- 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 bom design aumenta a agilidade.
- Simplicidade, a arte de maximizar a quantidade de trabalho não realizado é essencial.

2.2 Planejamento de Projetos

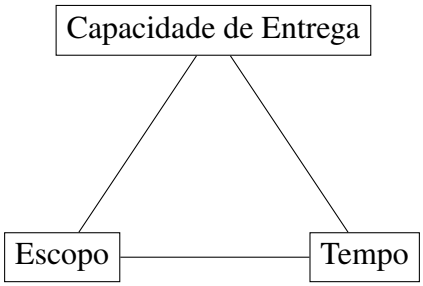
2.2.1 Triângulo das Restrições

Projetos possuem naturezas distintas de acordo com o negócio que lidam e as mais diversas variáveis internas e externas. Alguns têm escopo fechado: o conjunto de funcionalidades que precisam ser entregues não é negociável. Alguns têm um prazo estabelecido: a data de lançamento pode ter razões de negócio para ser determinada. Noutros, o orçamento pode ser fixo.

Orçamento, nesse contexto se refere à capacidade de investimento no projeto. Esse investimento se traduz na contratação ou alocação da equipe. Não se pode equacionar diretamente pessoas a produtividade. Há inclusive efeitos negativos em equipes acima de certo limiar de pessoas, dado o custo não linear de comunicação e organização; ou quando da inclusão de novas pessoas, devido ao processo de integração, aprendizagem e adaptação ao contexto. Embora não exista a relação de proporcionalidade direta entre pessoas e produtividade, é possível influenciar a capacidade de entrega através de investimentos. Assim, vamos considerar o orçamento como proporcional à capacidade de entrega.

O relacionamento entre o tempo, o escopo e o orçamento ou capacidade de entrega de um projeto é representado pelo triângulo das restrições conforme Figura 2.1. Ele nos mostra que todos os aspectos estão profundamente interligados [7]. Não é possível, por exemplo, aumentar o escopo de um projeto sem impactar o prazo de entrega ou suscitar investimentos que viabilizem o desenvolvimento no mesmo tempo, como a criação de uma nova equipe para lidar com o escopo adicional.

Figura 2.1: Triângulo das restrições



Fonte: Própria (2018)

Existe ainda uma quarta dimensão condicionada às três anteriores, a qualidade. É possível, digamos, aumentar a capacidade de entrega, acomodar um escopo maior ou reduzir o prazo às custas da qualidade do produto. No tocante ao código, a qualidade é tratada como premissa no ágil, que preza pela excelência técnica em seus princípios, pois entende que ganhos de curto prazo tem repercussões inversas no longo prazo. Note-se que qualidade é um conceito abstrato, mais difícil de quantificar do que escopo, tempo e capacidade de entrega, mas igualmente negociável.

2.2.2 Planejamento Ágil

O planejamento em metodologias ágeis leva em consideração que o contexto e os requisitos do negócio são dinâmicos e a própria equipe irá adquirir durante o projeto novas habilidades técnicas, maior conhecimento sobre o produto e sobre como trabalhar eficientemente em conjunto [8]. Diante deste cenário de mudanças e aprendizados constantes, faz sentido se beneficiar

das informações e aptidões vindouras, distribuindo o trabalho de planejamento ao longo da execução do projeto.

O planejamento ocorre em diferente níveis. Quanto mais longo o horizonte, mais abrangente e menor o nível de detalhamento. Afinal, por que se dar ao trabalho de detalhar todos os requisitos iniciais, se muitos deles serão modificados no processo ou serão despriorizados e nunca sequer implementados.

O início de um projeto é marcado por um processo de descoberta, idealmente com representantes de múltiplos papéis e partes interessadas. O objetivo é entender os objetivos, as restrições e definir suas características: escopo, prazo e composição da equipe, de onde se pode avaliar sua viabilidade.

Em um horizonte de alguns meses (em geral, entre 3 e 9) há o planejamento de entregas. O termo traz ambiguidade em tempos de difusão da prática de entrega contínua, em que é possível colocar funcionalidades em produção diariamente ou mesmo várias vezes ao dia. A existência de um planejamento de médio prazo, ainda assim, continua sendo relevante. É nele que se discute a priorização de funcionalidades para compor as entregas parciais, de modo que faça sentido para o negócio e para o desenvolvimento [9].

No curto prazo, há o planejamento de iteração, que se concentra no horizonte de uma iteração (em geral, entre 1 e 4 semanas). E discute em mais detalhes os requisitos de negócio e de implementação das funcionalidades, bem como a composição da iteração.

As funcionalidades são descritas como histórias de usuário. O termo expressa bem a proposta: é uma descrição da funcionalidade em termos do valor que será provido ao usuário. Redige-se em primeira pessoa, personificando cada classe de usuário, declarando, do seu ponto de vista, o benefício esperado ao se concluir aquela unidade de trabalho [10].

A história contém informações indispensáveis à implementação e aos testes. Uma das informações são os critérios de aceitação, que geram um entendimento mútuo de quando o trabalho estará concluído e o objetivo de entrega do valor alcançado. Não há pretensão, contudo, de se

tornar uma relação completa e detalhada dos requisitos. A história se propõe a ser mero ponto de partida de conversas, que promovam o compartilhamento de informações dispersas, e alinhamento do entendimento do que deve ser desenvolvido. Seu conteúdo pode ser modificado e complementado a qualquer tempo com a concordância da equipe [10].

As histórias nascem no *backlog*, o qual é uma lista de histórias que ainda não está em desenvolvimento. De acordo com a metodologia, utiliza-se outras nomenclaturas para essas unidades de trabalho de maior granularidade dos planejamentos de médio e longo prazo, como os "épicos", que contém inicialmente apenas uma descrição de alto nível das funcionalidades. À medida que são priorizadas, elas são refinadas, detalhadas e divididas em histórias de usuários propriamente ditas.

O planejamento das histórias busca garantir algumas características que impactam o trabalho da equipe. Ter histórias independentes umas das outras, por exemplo, possibilita que sejam desenvolvidas em paralelo, evitando bloqueios.

Quanto menor uma história for, mais rapidamente ela tende a ser desenvolvida, o que aumenta o fluxo de histórias durante a iteração. Esse aumento no fluxo, evita o acúmulo de histórias na fase de testes, que ocorre quando as histórias são longas e acabam o desenvolvimento simultaneamente no fim da iteração. Histórias pequenas tornam, também, mais gerenciável a identificação de problemas, pois se algo é observado em uma unidade de trabalho menor, será resolvida mais fácil, rapidamente e com menor impacto na entrega do que o retorno de uma porção de funcionalidade maior à fase de desenvolvimento. O fluxo constante de histórias gera ainda uma sensação de progresso motivadora para equipe e clientes.

As histórias precisam ainda ser estimáveis e limitar seu tamanho ajuda também nessa atividade. Unidades de trabalho pequenas e bem definidas facilitam a compreensão do que precisa ser construído e como essa funcionalidade pode ser implementada, reflexões que culminam em uma estimativa do esforço necessário para execução da história.

2.3 Estimativas em Projetos Ágeis

A estimativas formam a base do planejamento e da análise de riscos de projeto. O planejamento descreve uma possível estratégia para alcançar os objetivos de negócio, fundamentado nas estimativas [8], com um risco associado.

O planejamento, portanto, é uma atividade tendenciosa a um resultado positivo, pois demonstra uma forma de alcançar tal resultado. Ao passo que as estimativas são desprovidas de intenção, sem viés por qualquer resultado [11].

Não necessariamente planejamento e estimativas coincidirão. É possível a construção de um planejamento que considera os objetivos de negócio e seja divergente das previsões das estimativas. Tal discrepância é representada por um aumento no fator de risco, representando a redução da probabilidade de concretização do planejamento [11].

A depender das características do projeto, pode se fazer necessário estimar seu escopo, sua duração ou seus custos. Estas são estimativas derivadas, por serem obtidas por cálculos que dependem de outras estimativas, no caso, a estimativa associada a histórias de usuário.

Histórias de usuário são comumente medidas em termos do tempo despendido na execução da tarefa, com dias de trabalho como unidade. Esta, porém, traz consigo grande variabilidade, visto que um dia de trabalho pode ser, por exemplo, repleto de reuniões. Foi proposto então o conceito de dias ideais, para expressar essa diferenciação [8].

Há ainda quem advogue por desassociar as medidas de escalas absolutas. Elas podem expressar a proporção relativa entre uma história e outra. Para isso, adotam-se pontos como unidade agnóstica de medida, que representam os conceitos abstratos de esforço e complexidade que acabam por serem convertidos eventualmente para unidades de tempo [8].

Os processos de estimativa precisam definir uma escala de valores. Uma opção simples é a chamada "tamanhos de camisa", que propõe os rótulos: {pequeno, médio, grande} para clas-

sificar as histórias, às quais valores fixos são atribuídos a priori, a exemplo de: {pequeno=1, médio=2, grande=3}. É comum também o uso de escalas não lineares, inspiradas na sequência de Fibonacci: $\{1,2,3,5,8,...\}$ e em múltiplos de 2: $\{1,2,4,8,...\}$. Essa opção se beneficia do espaçamento crescente entre os valores, que se mostra uma representação apropriada, considerando o aumento de incerteza que acompanha histórias maiores [8]. James Grenning propôs o uso da escala: $\{1,2,3,5,7,10,\infty\}$. O limite de valores foi definido com objetivo de restringir o tamanho das cartas dentro de 2 semanas de trabalho. Se a estimativa de trabalho para uma história extrapola 2 semanas, usa-se o valor infinito. Indicativo de que ela deve ser quebrada em unidades menores de trabalho [12].

Na realidade qualquer escala pode ser utilizada. Mesmo valores maiores e dispersos, como $\{10,20,30,50,100\}$. É válido, no entanto, observar que há estudos que indicam prejuízo na capacidade humana de estimar, quando se percorre mais de uma ordem de grandeza [13].

Pode-se ainda incluir o número 0 para atribuir a tarefas que não constituam estritamente histórias de usuário. A depender da definição das equipes, defeitos, débitos técnicos (que podem ser definidos como problemas técnicos que tiveram sua resolução adiada [14]) e histórias muito pequenas podem ser tratadas como 0 [8].

Os métodos de obtenção das estimativas podem ser classificados em determinísticos e probabilísticos. A abordagem determinística tem uma visão interna do projeto, baseando-se na quebra do trabalho em unidades cada vez menores e mais detalhadas. A abordagem probabilística, por sua vez, toma uma visão externa, fundamentando-se em características de alto nível do projeto para identificar seu comportamento com base no histórico de projetos similares [15].

As estimativas probabilísticas automatizadas baseada em classes de projeto, seguem o seguinte processo [15]:

1. A definição de classes de projetos similares, buscando classes que sejam amplas o bastante para conseguir dados em quantidade que garantam a relevância estatística; e que sejam específicas o suficiente para permitir a diferenciação entre projetos com caracte-

rísticas e comportamentos distintos.

2. Estabelecer uma distribuição de probabilidade para as classes de projetos. Tendo como base dados empíricos com credibilidade de uma quantidade de projetos grande o suficiente para certificar a validade estatística das conclusões.

3. Comparação entre o novo projeto e as classes de referência, para identificação do comportamento mais provável para o projeto.

Os projetos que se enquadram nos perfis comuns obtêm melhores resultados com essa técnica [16]. Ao passo que projetos que lidam com domínios de negócio atípicos, mudanças de tecnologia constantes, variação frequente na composição da equipe e todo tipo de instabilidade e inovação, não são o público-alvo da abordagem probabilística.

A abordagem determinística se baseia em técnicas manuais, que consistem em consultar uma ou mais pessoas desenvolvedoras de software para estimar com base em suas experiências e julgamento pessoal. Chama-se "Opinião do Especialista", quando a participação é individual. E há diversas técnicas que envolvem a participação de múltiplos desenvolvedores, a exemplo de *Wideband Delphi* e *Planning Poker*.

São abundantes as pesquisas com avaliação da acurácia de métodos de estimativa. Eles apresentam, contudo, grande variabilidade de resultados com intervalos que perpassam 5 ordens de grandeza [17]. Como demonstrou BASHA (2010) ao aplicar as técnicas COCOMO, SEER, COSEEKMO, REVIC, SASET, Cost Model, SLIM, FP, Estimac e Cosmic em diversos projetos e obter médias de magnitude do erro relativo MMRE que variaram entre 0.373% e 771.87% [18]. Um indicativo de alta dependência do contexto, com diferentes projetos se beneficiando de diferentes técnicas de estimativa de acordo com suas características.

Existem ainda técnicas híbridas como utilizar algoritmos de aprendizagem de máquina para estimar o esforço de unidades de trabalho baseada na análise de linguagem natural da descrição das tarefas, e utilizar essa informação para auxiliar e comparar com estimativas manuais [17].

2.3.1 Planning Poker

O *planning poker* é uma técnica de estimativa colaborativa, no qual toda equipe se reúne, incluindo desenvolvedores, analistas de qualidade, analistas de negócio, clientes e etc [19].

Equipes de projetos ágeis buscam uma composição que lhes forneça todas as habilidades necessárias ao seu funcionamento. Em geral seus integrantes são "especialistas generalistas", pessoas com conhecimento mais aprofundado em alguma especialidade, porém com conhecimento geral mínimo que lhes torna capazes de contribuir em qualquer área [20].

Essa formação com múltiplos papéis e tendência generalista justifica a participação de todas na elaboração de estimativas, afinal qualquer pessoa poderá atuar na implementação de qualquer história [19].

O *planning poker* foi originalmente proposto em 2002 por James Grenning. Tendo por objetivo solucionar a falta de efetividade nas discussões sobre estimativas [12]. A mecânica do processo está descrita na Figura 2.2 [12]:

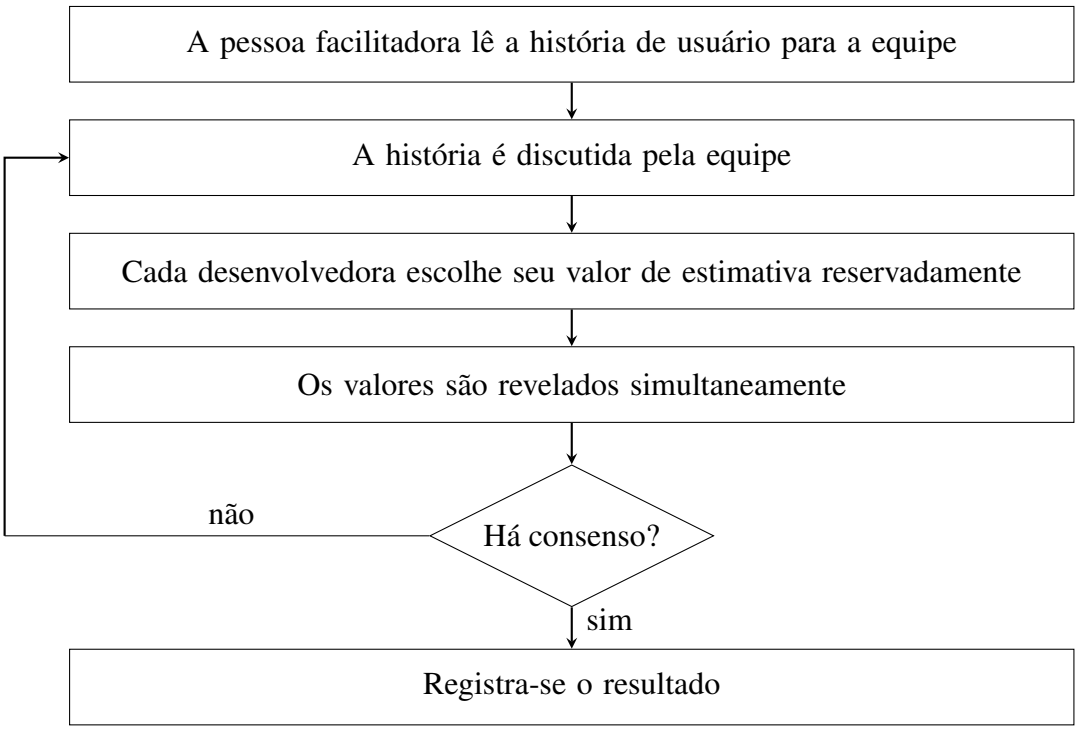
Se não houver consenso, a história pode ainda ser ignorada e posteriormente dividida em histórias menores para facilitar as discussões na próxima sessão [12].

2.3.2 Velocidade Bruta

Para estimar a velocidade de uma equipe no início de um projeto, antes mesmo de iniciar o desenvolvimento, utiliza-se a técnica de velocidade bruta [21]:

1. As histórias de usuário são estimadas.
2. Em seguida, sem observar as estimativas, as histórias são agrupadas em conjuntos que se julga caber em uma iteração.

Figura 2.2: Etapas do planning poker



Fonte: Própria (2018)

3. Contabilizam-se, então, as estimativas das histórias escolhidas e esta é considerada a velocidade bruta.

É razoável também, que se agrupe histórias para montar até três iterações e se tire a média do somatório das estimativas de cada uma para calcular a velocidade bruta.

2.3.3 Cálculo de Estimativas Derivadas

Para estimar a porção de escopo entregue em uma iteração finalizada, utilizam-se as estimativas individuais de cada história de usuário. A soma de pontos estimados das histórias concluídas na iteração constitui a estimativa do escopo entregue [10].

Dados:

i, o número da iteração.

j, o número da história de usuário dentro da iteração.

EH_{ji}, estimativa de uma história de usuário j, pertencente a uma iteração i.

n_i, a quantidade de histórias de usuário concluídas na iteração i.

$EEEI_i$, estimativa de escopo entregue na iteração i.

$$EEEI_i = \sum_{j=1}^{n_i} EH_{ij}$$

Para estimar que parcela do escopo pode ser alocada em iterações futuras, pode-se simplesmente utilizar a estimativa de escopo entregue na iteração anterior, por ser o valor de referência mais recente e atualizado. Essa técnica é chamada de "tempo do dia anterior", aludindo ao fato de que o tempo de um dia é um indicativo de previsão do tempo para o dia seguinte [22].

Dados:

i, o número da iteração.

c, o número da iteração em curso, admitindo $c > 1$

$EEEI_i$, estimativa de escopo entregue na iteração i.

$CEEI$, estimativa de capacidade de entrega de escopo por iteração

$$CEEI = EEEI_{c-1}$$

Com mais tempo de projeto, mais iterações transcorridas e mais dados à disposição, pode-se utilizar a média aritmética das últimas iterações para estimar a capacidade de entrega da equipe. Limita-se o número de iterações consideradas, porém, pois as informações recentes se sobrepõem em peso às passada. Tipicamente, se restringe a um máximo de 3 ou 4 iterações. Note-se que a equação do tempo do dia anterior, nada mais é do que o caso especial, no qual se considera uma única iteração.

Dados:

i, o número da iteração.

c, o número da iteração em curso, admitindo $c > 1$

r, número máximo de iterações a serem contabilizadas, admitindo-se $0 < r < c$

$EEEI_i$, estimativa de escopo entregue na iteração.

$CEEI$, estimativa de capacidade de entrega de escopo por iteração.

$$CEEI = \frac{1}{r} \sum_{j=1}^{r} EEEI_{c-j}$$

A escolha de quantas iterações passadas considerar depende do contexto da equipe. Dados históricos referentes a um conjunto diferente de condições podem ser pouco relevantes para previsões futuras. Se estão ocorrendo muitas mudanças, como entrada e saída de integrantes, mudanças de escopo, de tecnologia e etc. Deve-se limitar o cálculo a última iteração; se a equipe encontra-se em uma conjuntura mais estável. É válido considerar mais de uma iteração. Caso uma iteração específica tenha transcorrido em circunstâncias manifestamente atípicas, ela pode ser desconsiderada dos cálculos intencionalmente. São exemplos de circunstâncias que afetam pontualmente a velocidade da equipe: problemas burocráticos que causem restrições de acesso; época de festividades do ano com recesso compulsório; mudanças drásticas na equipe, como o particionamento em subequipes e etc.

Para estimar o escopo total de um projeto, estimam-se todas as histórias existentes inicialmente, somam-se seus valores, e multiplica-se o resultado do somatório por um fator de crescimento, que representa o acréscimo de escopo devido ao refinamento das histórias.

Dados:

EH, estimativa atribuída a uma história de usuário

EP_i, estimativa de escopo inicial do projeto

EP_f, estimativa de escopo final do projeto

r, fator de crescimento, devido ao refinamento das histórias

$$EP_i = \sum EH$$

$$EP_f = EP_i \cdot r$$

Para estimar o prazo de duração de um projeto, durante sua etapa inicial, calcula-se o quociente da estimativa de escopo final do projeto pela velocidade bruta.

Dados:
IEP, quantidade de iterações estimadas como necessárias para terminar o projeto
EP_f, estimativa de escopo final do projeto
v, velocidade bruta
$CEEI$, estimativa de capacidade de entrega de escopo por iteração.

$$IEP = \frac{EP_f}{v}$$

Se o cálculo for realizado durante o projeto, quando já houver dados históricos, a velocidade bruta é substituída pela capacidade de entrega de escopo por iteração.

$$IEP = \frac{EP_f}{CEEI}$$

Para estimar o tamanho do escopo passível de entrega, dentro de um prazo preestabelecido, calcula-se o produto da capacidade de entrega pelo número de iterações à disposição.

Dados:
I, quantidade de iterações que compõem o período da entrega
$CEEI$, estimativa de capacidade de entrega de escopo por iteração.
EAE, estimativa de escopo alocável em entrega

$$EAE = CEEI \cdot I$$

Para estimar o tempo necessário para realizar uma entrega, primeiro estima-se o escopo da entrega. Estimar o escopo da entrega significa estimar todas as histórias que constituem a entrega individualmente e somar os valores obtidos. Calculamos, então, o quociente do escopo da entrega pela capacidade de entrega por iteração.

Dados:
EH, estimativa atribuída a uma história de usuário
$CEEI$, estimativa de capacidade de entrega de escopo por iteração.
EE, estimativa de escopo da entrega
IE, quantidade de iterações estimadas como necessárias para realizar a entrega

$$EE = \sum EH$$
$$IE = \frac{EE}{CEEI}$$

Para estimar a porção de escopo que precisaria ser alocada em cada iteração para que se viabilize a finalização do escopo de entrega em um período estabelecido, calculamos o quociente do escopo planejado para a entrega pela quantidade de iterações disponíveis.

Dados:
I, quantidade de iterações que compõem o período da entrega
EE, estimativa de escopo da entrega
EI, porção de escopo que precisará ser alocado em cada iteração

$$EI = \frac{EE}{I}$$

2.4 Movimento #NoEstimates

Em 2012, Woody Zuill inaugurou uma onda de questionamentos sobre o uso de estimativas, ao descrever em detalhes uma experiência de projeto em seu blog. Seu cliente tinha uma longa lista de requisitos, todos indispensáveis, e exigia o uso de estimativas no desenvolvimento. Zuill negociou para que o cliente escolhesse uma funcionalidade importante e pequena e permitisse que a equipe trabalhasse nela primeiro e em seguida rediscutissem a questão das estimativas.

A primeira entrega ocorreu em uma semana. Nenhuma menção a estimativas foi feita pelo cliente a partir daí. Tendo contato com o software em produção, clientes e usuários começaram a solicitar aprimoramentos baseados na experiência real de uso. Após ter cerca de 25% dos requisitos iniciais implementados, o cliente estava satisfeito com o produto, tendo observado que os usuários já eram capazes de desempenhar as principais tarefas, e o resto não era mais tão importante no momento [23].

O link para o texto de Zuill foi publicado no Twitter, que se tornou palco de ávidos debates sobre o uso de estimativas. A postagem estava acompanhada da hashtag #NoEstimates, que passou a denominar essa corrente de pensamento. Embora o termo não faça justiça à filosofia do movimento, que não propõe a extinção completa das estimativas e sim substituir as estimativas de esforço por estimativas baseadas em dados históricos com baixo custo de obtenção [2].

Duarte (2012), se uniu ao coro, trazendo uma perspectiva teórica, da física estatística. Admitindo que o processo de desenvolvimento de software seja um sistema complexo, este não admite causalidade. Dado que a capacidade de estimar tem como premissa, que certos fatores impliquem em maior ou menor estimativas. Não havendo relação consistente de causa-efeito, o processo de estimar não tem propósito. E atribui a racionalização de causalidades a estimativas à coerência retrospectiva, um viés cognitivo, que explica como pode ser extremamente fácil explicar fenômenos após sua ocorrência, não obstante, seja impossível predizê-los [24].

Killick (2012) causou repercussão, alegando que mais do que ineficazes, as estimativas podem

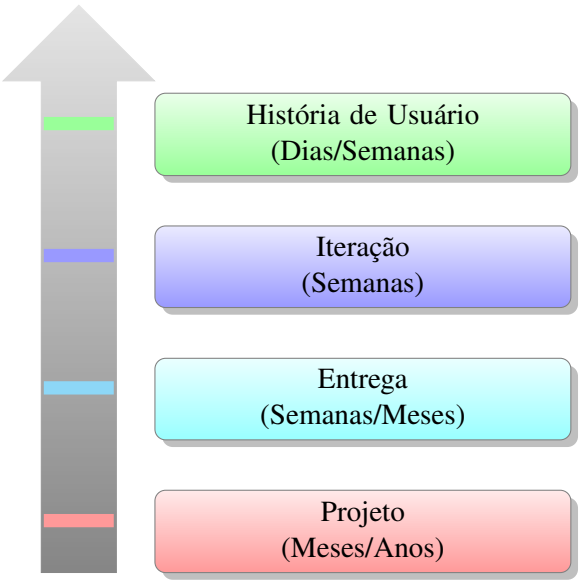
ter impacto negativo nos projetos, ao criarem um ambiente de microgerenciamento, controle e medo, destoantes da filosofia ágil. De fato, não raro, há relatos de equipes sentindo a pressão de não atingir as estimativas de entrega, que refletem bem mais os desejos e metas do negócio, do que uma predição de desempenho que preze pela acurácia e realismo [25].

CAPÍTULO 3

Metodologia

O presente trabalho se propõe a identificar alternativas ao uso de estimativas de esforço na gestão ágil de projetos de desenvolvimento de software. Tendo em vista que o planejamento ágil se dá em diferentes níveis de granularidade, de horizontes de tempo e detalhamento de informações conforme Figura 3.1, investiga-se em separado os usos das estimativas na gestão das histórias de usuário, das iterações, das entregas e do projeto como um todo.

Figura 3.1: Grau de detalhamento do planejamento



Fonte: Própria (2018)

3.1 Identificação de Alternativas às Estimativas de Esforço na Gestão de Histórias de Usuários

As estimativas de esforço, no contexto das histórias de usuário, são utilizadas com o propósito de monitorar e controlar o progresso do trabalho de desenvolvimento e testes. Acredita-se, ainda, que o processo de estimar histórias de usuário utilizando a técnica *planning poker* promove a difusão e aprofundamento do entendimento das histórias pela equipe. Serão avaliadas alternativas para ambos os casos de uso.

3.1.1 Monitorar e Controlar Progresso do Trabalho

Estimar o esforço de uma história de usuário se presta ao propósito de antecipar quanto tempo será necessário para execução da mesma.

Ao comparar o tempo efetivamente transcorrido no desenvolvimento com o esforço estimado, obtém-se a informação se a história está levando mais, menos ou exatamente o tempo previsto originalmente. Caso a estimativa preveja com precisão o tempo de desenvolvimento, tem-se o caso ideal, que não evoca intervenções.

Tabela 3.1: Relação estimativa-informação

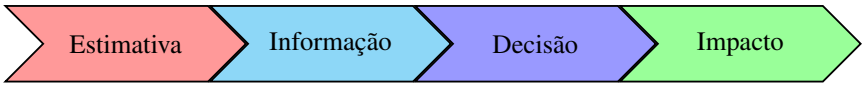
ESTIMATIVA	INFORMAÇÃO
Estimativa de Esforço da História	Desenvolvimento Atrasado
	Desenvolvimento Adiantado

Essa informação pode ser utilizada para auxiliar tomadas de decisões técnicas das pessoas diretamente envolvidas com o desenvolvimento como analistas de qualidade e pessoas desenvolvedoras ou para motivar intervenções gerenciais de pessoas envolvidas indiretamente com o desenvolvimento como analistas de negócio e gerentes de projeto.

Killick (2012) alega que decisões técnicas baseadas exclusivamente em estimativas de esforço impactam a qualidade dos projetos. A prioridade passa a ser cumprir as metas de curto prazo, comprometendo a qualidade e impactando inclusive a velocidade no médio prazo [14]. É válido, por tanto, pesquisar técnicas de decisão que considerem também impactos de longo prazo no projeto.

Há relatos de que o uso de estimativas para o monitoramento de produtividade geram um ambiente de pressão e desconfiança, que pode repercutir na saúde mental das pessoas, no clima e na produtividade das equipes[25]. O manifesto ágil prega que equipes de alto desempenho são compostas por indivíduos motivados e que se deve confiar e apoiar as pessoas. Assim formas de controle devem estimular a colaboração e a confiança mútua. [6]

Figura 3.2: Processo de controle



Fonte: Própria (2018)

Será investigado o processo de controle das histórias de usuário conforme a Figura 3.2, observando quais decisões mais comuns se baseiam nas informações providas pelas estimativas de esforço, visando identificar técnicas alternativas que sejam capazes de auxiliar as mesmas tomadas de decisão.

Pergunta de Pesquisa 1

Que métodos alternativos podem auxiliar decisões técnicas das pessoas envolvidas diretamente com o desenvolvimento considerando impactos de curto e de longo prazo?

Pergunta de Pesquisa 2

Que formas alternativas podem auxiliar decisões de gerenciamento das pessoas envolvidas indiretamente com o desenvolvimento promovendo colaboração e confiança?

3.1.2 Aprofundar Entendimento das Histórias

O processo de estimar histórias tem a pretensão de produzir não só um valor, como também gerar reflexões que aprofundem o entendimento das histórias, levantando perguntas como: Que sistemas estão envolvidos na implementação? Qual será o trabalho necessário? Como essa aplicação deve funcionar?

Em particular, a utilização da técnica de *planning poker* almeja direcionar discussões mais produtivas, que disseminem conhecimentos sobre formas de implementação das histórias de usuário e ampliem o entendimento de seus requisitos entre todas as pessoas da equipe.

O artigo original que propôs o *planning poker*, porém, aponta como preocupação exatamente a possibilidade de que discussões importantes acabem não acontecendo. Embora na experiência do autor, as discussões não ocorram quando as histórias se referem a funcionalidades semelhantes a implementações já realizadas, bem conhecidas e estabelecidas no sistema; ao passo que funcionalidades novas suscitam as discussões e demandam tempos mais longos para entendê-las.[12]

Pergunta de Pesquisa 3

É possível compartilhar o entendimento de negócio das histórias sem estimá-las?

Pergunta de Pesquisa 4

É possível compartilhar o entendimento de técnico das histórias sem estimá-las?

3.2 Identificação de Alternativas às Estimativas de Esforço na Gestão de Iterações

A estimativa da capacidade de entrega de escopo por iteração é resultado do cálculo da média aritmética da estimativa de escopo entregue nas iterações precedentes. Sendo a quantidade de iterações contabilizadas no cálculo um valor arbitrado pelo projeto segundo seu contexto.

A estimativa do escopo entregue nas iterações passadas, por sua vez, convencionalmente decorre da soma das estimativas de esforço das histórias de usuário concluídas na iteração.

Pergunta de Pesquisa 5

É possível gerenciar iterações sem estimar o esforço das histórias?

3.3 Identificação de Alternativas às Estimativas de Esforço na Gestão de Entregas

O monitoramento e controle de uma entrega pode ser realizado acompanhando suas variáveis:

- estimar o escopo alocável na entrega, refazer o cálculo a cada iteração, adaptando o *backlog* das histórias à estimativa do que pode ser entregue.
- estimar o tempo necessário para a entrega, refazer o cálculo a cada iteração, adaptando o prazo à estimativa do tempo necessário.
- estimar o escopo por iteração, comparar o escopo efetivamente entregue na iteração com o que precisaria ser entregue com base no escopo e prazo desejados e realizar intervenções que aumentem a capacidade de entrega.

Dados:
EAE, estimativa de escopo alocável em entrega
IE, quantidade de iterações estimadas como necessárias para realizar a entrega
EI, porção de escopo que precisará ser alocado em cada iteração

I, quantidade de iterações que compõem o período da entrega
$CEEI$, estimativa de capacidade de entrega de escopo por iteração.

EE, estimativa de escopo da entrega

$$EAE = CEEI \cdot I$$

$$IE = \frac{EE}{CEEI}$$

$$EI = \frac{EE}{I}$$

Todas, no entanto, se baseiam indiretamente em estimativas de esforço de histórias de usuário.

Pergunta de Pesquisa 6

É possível gerenciar entregas sem estimar o esforço das histórias?

3.4 Identificação de Alternativas às Estimativas de Esforço na Gestão do Projeto

O planejamento inicial de um projeto baseia suas estimativas de prazo, orçamento e escopo do projeto nas estimativas de esforço de histórias.

Dados:

EP_i, estimativa de escopo inicial do projeto

EP_f, estimativa de escopo final do projeto

r, fator de crescimento, devido ao refinamento das histórias

IEP, quantidade de iterações estimadas como necessárias para terminar o projeto

v, velocidade bruta

$$EP_f = EP_i \cdot r$$

$$IEP = \frac{EP_f}{v}$$

Pergunta de Pesquisa 7

É possível gerenciar projetos sem estimar o esforço das histórias?

3.5 Coleta de Dados

Este trabalho é uma pesquisa primária, o qual para seu processo de investigação foi incluída a coleta de dados [26] de projetos de desenvolvimento de software, de modo a obter informações especificamente curadas para responder as perguntas propostas conforma Figura 3.3. São usados diversos métodos de coleta de dados, visto que apresentam vantagens e desvantagens que os tornam mais apropriados para situações específicas, são eles: entrevistas, questionários e casos de estudo.

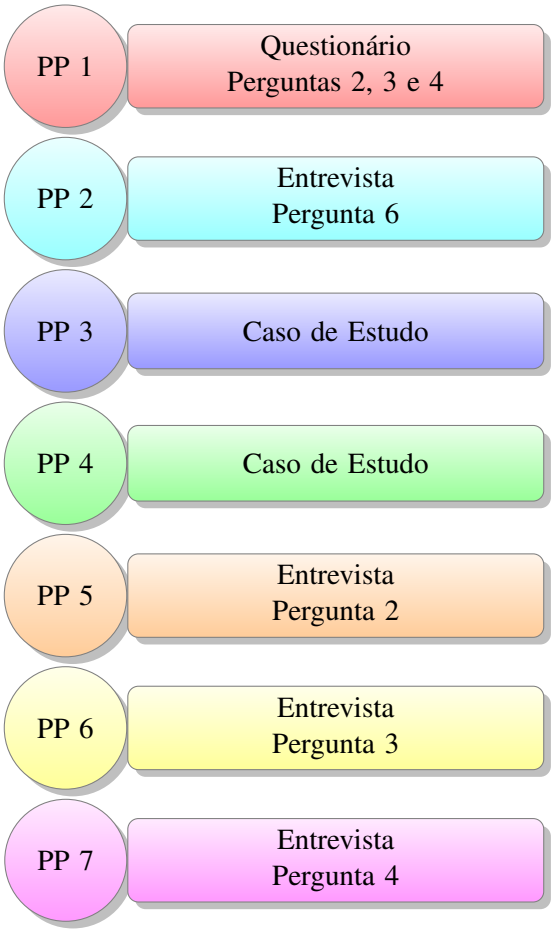
Ao elaborar perguntas, prestou-se atenção especial em utilizar uma linguagem clara, evitar vieses e ter perguntas que inferem uma única informação cada [26].

3.5.1 Questionário para Pessoas Desenvolvedoras e Analistas de Qualidade

Para coletar dados de pessoas desenvolvedoras e analistas de qualidade, optou-se pelo uso de um questionário, privilegiando o volume de respostas. As perguntas abertas foram todas configuradas como opcionais, para conferir flexibilidade, tendo em vista a variabilidade de cenários.

Comparando as estimativas com os dados reais, temos dois cenários de divergência, o tempo de desenvolvimento da história pode ser maior ou menor do que o previsto pelas estimativas, elaboramos perguntas distintas para cada uma das duas situações.

Figura 3.3: Dados destinados às perguntas de pesquisa



Fonte: Própria (2018)

1. Qual o seu papel no projeto?

2. Se a implementação de uma história se mostra mais complexa do que a estimativa previa, isso afeta suas decisões de desenvolvimento e teste?

   Se sim:

   (a) Afetam de que forma? Quais decisões são influenciadas?

3. Se a implementação de uma história se mostra mais simples do que a estimativa previa, isso afeta suas decisões de desenvolvimento e teste?

Se sim:

(a) Afetam de que forma? Quais decisões são influenciadas?

4. Há outros fatores ou técnicas que apoiam suas decisões de desenvolvimento? Quais?

3.5.2 Entrevista para Analistas de Negócio

Optou-se por utilizar entrevistas para obter respostas sobre como as estimativas estão sendo utilizadas e que outras formas de controle têm sido praticadas. A escolha tem a pretensão de estimular o detalhamento das experiências através da interação e dada a imprevisibilidade do que pode ser relatado, se beneficiar da flexibilidade da entrevista para obter esclarecimentos.

As entrevistas são semi-estruturadas, segue-se um roteiro mínimo de perguntas para guiar a conversa para os tópicos de estudo, mas abre-se espaço igualmente para que a entrevistada compartilhe experiências de múltiplos projetos livremente.

1. Qual seu papel no projeto?
2. Como realiza o planejamento e monitoramento das iterações?
3. Como realiza o planejamento e monitoramento das entregas?
4. Como realizou o planejamento inicial do projeto?
5. Se a implementação de uma história se mostra mais demorada do que a estimativa previa, essa informação já lhe levou a decidir por realizar alguma intervenção? Qual?
6. Utiliza alguma outra técnica ou informação para decidir por realizar alguma intervenção? Qual?

3.5.3 Estudo de Caso

Dado um conjunto de histórias de usuário, foi solicitado a um grupo de controle que estimasse o grau de dificuldade de entendimento das histórias. Em seguida, dividiu-se as histórias em dois subconjuntos balanceando a distribuição de modo a equilibrar seu grau de dificuldade. Um deles foi estimado na cerimônia de planejamento da iteração, utilizando a técnica do *planning poker*. O outro subconjunto foi discutido sem realizar estimativas.

Foi solicitado então às pessoas desenvolvedoras que respondessem o quanto entenderam cada história, em uma escala de 1 a 5, em relação aos requisitos técnicos e de negócio separadamente, em formulários distintos para as histórias estimadas e não estimadas:

1. Como avalia seu entendimento dos requisitos de negócio da história?
2. Como avalia seu entendimento dos requisitos técnico das história?

Retira-se a mediana da pontuação de cada pergunta. Em seguida, calcula-se a média de cada método de discussão.

N, número de perguntas.
p_i, pontuação da pergunta i.
M, média do método de discussão

$$M = \frac{1}{N} \sum_{i=1}^{N} p_i$$

CAPÍTULO 4

Resultados

Foram analisados dados de equipes de desenvolvimento de software, em projetos de consultoria a outras organizações, praticantes de híbridos de metodologias ágeis. Os projetos se relacionam com domínios variados. Suas especificidades serão omitidas em respeito a acordos contratuais de confidencialidade.

Foram realizadas 20 entrevistas com analistas de negócio, gerentes de iteração e gerentes de projeto e recebidos 132 respostas ao questionário de pessoas desenvolvedoras e analistas de qualidade.

O estudo de caso se focou em duas equipes composta por: 8 pessoas desenvolvedoras, 1 analista de qualidade e 1 analista de negócio, durante duas iterações com duração de 15 dias cada.

4.1 Pergunta de Pesquisa 1

Que métodos alternativos podem auxiliar decisões técnicas das pessoas envolvidas diretamente com o desenvolvimento considerando impactos de curto e de longo prazo?

Para solucionar essa questão, primeiro foram analisadas quais as decisões mais comuns que se baseiam em estimativas de esforço e em seguida que alternativas são praticadas pelas equipes. Baseando-se nas respostas ao questionário enviado a pessoas desenvolvedoras e analistas de qualidade.

Foram recebidas 48 respostas citando decisões tomadas com base em estimativas de esforço, a

exemplo de:

> "Já deixei de criar mais casos de teste em uma carta, porque ela era de 1 ponto."
>
> "Quando termino a carta, mas ela é de 5 pontos, aproveito para fazer refatorações."
>
> "Não me importo com estimativa, faço o que é preciso pra implementar a carta."
>
> "Me baseio nas estimativas pra decidir entre formas de implementação."

As decisões foram categorizadas em:

Tabela 4.1: Decisões técnicas baseadas em estimativas

OCORRÊNCIAS	DECISÕES
42	Pagar ou Gerar Débito Técnico
15	Opções de Implementação e Arquitetura

Em seguida questionamos que outras técnicas ou informações são utilizadas para auxiliar a tomada das mesmas decisões. Eis algumas das respostas recebidas:

> "Minha equipe decide na *planning* quais débitos técnicos serão pagos na iteração."
>
> "A gente discute na *tech huddle* a necessidade de se fazer ou não grandes refatoramentos, porque, às vezes, começar refatorando facilita todo o desenvolvimento da história."
>
> "A gente tem um quadro de priorização com todos os débitos técnicos."

Os métodos alternativos puderam ser classificados como:

Tabela 4.2: Métodos alternativos de auxílio a decisões técnicas

OCORRÊNCIAS	TÉCNICAS ALTERNATIVAS
22	Matrizes de Priorização de Débito Técnico
16	Discussões Técnicas
8	Alocação de Débitos Técnicos no Planejamento da Iteração

Em relação a débitos técnicos, 22 pessoas relataram o uso de algum tipo de matriz que categorizasse os débitos em relação a dois fatores que representados como eixos que formam os quadrantes da matriz. Um dos eixos utiliza um parâmetro de curto prazo e o outro de longo prazo.

> "Consideramos o benefício esperado ao pagar o débito e o risco de não pagar."
>
> "Avaliamos o valor que pagar esse débito traz para o negócio no momento e em relação às necessidade futuras"
>
> "Para entender a causa dos débitos técnicos nos perguntamos se foram decisões conscientes ou inadvertidas e se pensando prudentemente no longo prazo, ou imprudentemente apenas na entrega corrente"

Oito respostas relataram a priorização de antemão dos débitos a serem pagos na iteração. Observa-se assim uma reflexão intencional de quais dívidas a equipe irá conscientemente postergar ou ignorar e quais considera justificável dispensar atenção imediata.

> "Minha equipe acordou pagar dois débitos técnicos por iteração."

Quando se deparam com a necessidade de tomar decisões arquiteturais, discussões técnicas holísticas foram apresentadas como alternativa ao uso de estimativas de esforço como forma de decisão.

> "Eu convoquei uma reunião com a equipe, apresentei as opções de arquitetura e discutimos vantagens e desvantagens"
>
> "Conversamos com outras pessoas da equipe e avaliamos não só qual seria a arquitetura ideal, mas também o contexto do projeto, por estamos atrasados e buscamos assim uma solução que fosse fácil de ser refatorada para a arquitetura ideal posteriormente"

As equipes avaliam em conjunto as opções, observando os requisitos de negócio, o contexto do projeto e diretrizes de arquitetura evolutiva, que ajudam a identificar soluções apropriadas para o momento do projeto e da equipe.

Nota-se nessa forma de decisão arquitetural uma reflexão sobre o estado ideal, e o estado atual, para obter um balanceamento entre qualidade e custo de implementação. Uma abordagem coletiva e multifacetada com contribuições de argumentos técnicos e de negócio, que considera custos e benefícios.

Obtivemos assim diversos exemplos de alternativas ao uso de estimativas de esforço que auxiliam decisões técnicas considerando impactos de curto e longo prazo.

4.2 Pergunta de Pesquisa 2

Que formas alternativas podem auxiliar decisões de gerenciamento das pessoas envolvidas indiretamente com o desenvolvimento promovendo colaboração e confiança?

O primeiro passo para responder essa pergunta foi identificar quais as decisões gerenciais mais comuns tomadas com base em estimativas de esforço. Eis algumas das respostas obtidas em entrevista com analistas de negócio, gerentes de projeto e de iteração:

> "Se a história ultrapassou a estimativa, vou tentar entender se faz sentido quebrá-la em histórias menores."
>
> "Quando uma história está levando tempo demais, vou procurar saber se existe algum bloqueio que eu possa ajudar a remover."

Assim categorizou-se as decisões desse modo:

Tabela 4.3: Decisões gerenciais baseadas em estimativas

OCORRÊNCIAS	DECISÕES
15	Remoção de Bloqueios
8	Gerenciamento de Escopo

O próximo passo foi entender se haveria formas alternativas de auxiliar tais tomadas de decisão que primem pela colaboração e confiança na equipe. Eis alguns dos relatos obtidos:

> "Na nossa equipe, a gente estabeleceu que toda história deveria durar até 3 dias. Então no segundo dia, a gente conversa, sobre o andamento da história, se ela é muito grande e precisa ser quebrada ou se o par precisa de alguma ajuda."
>
> "Na *daily* os pares contam coisas que estão atrapalhando o desenvolvimento, como falta de clareza nos requisitos e a cliente se dispõe a clarificar as dúvidas, ou eu procuro as respostas que estão precisando."

As respostas foram classificadas do seguinte modo:

Tabela 4.4: Técnicas alternativas de auxílio a decisões gerenciais

OCORRÊNCIAS	TÉCNICAS ALTERNATIVAS
20	Reunião Diária
5	Limite de Dias de Desenvolvimento

Na reuniões diárias, as pessoas desenvolvedoras e analistas de qualidade descrevem os bloqueios que estejam enfrentando em suas histórias. Ao oferecer ajuda com base nas dificuldades informadas voluntariamente a situação caracteriza uma trabalho colaborativo.

Quando a equipe faz um acordo de que cada história seja finalizada em um dado limite de dias e trata isso como uma responsabilidade compartilhada por todos os papéis, que avalia em conjunto que medidas podem ser tomadas para atingir esse objetivo a exemplo de flexibilizar o escopo da história, que pode ser quebrado em histórias menores, mudar a estratégia de implementação ou de testes. Temos assim um trabalho de equipe em ação.

Obtivemos assim técnicas alternativas para auxiliar decisões gerenciais sem o uso de estimativas de esforço e que enfatizam a colaboração e confiança da equipe.

4.3 Pergunta de Pesquisa 3

É possível compartilhar o entendimento de negócio das histórias sem estimá-las?

Obtivemos um conjunto de 20 histórias de usuário relativas à equipe A. Pedimos então à equipe B que atribuísse um valor de 1 a 3 de acordo com a dificuldade de se entender os requisitos de negócio das histórias. A partir desses valores as histórias foram divididas em dois conjuntos com grau de dificuldade similar.

Um dos conjuntos foi discutido pela equipe A tendo a conversa guiada pelo processo de estimativa de esforço, e o outro conjunto foi discutido também pela equipe A livremente sem realizar estimativas.

Cada uma das 8 pessoas desenvolvedoras e a pessoa analista de qualidade da equipe atribuíram, em seguida, um valor de 1 a 5 ao seu entendimento de negócio de cada história.

Em seguida repetimos o processo com a equipe A atribuindo valores de dificuldade de entendimento de negócio às 20 histórias da equipe B. E partir disso dividindo em dois conjuntos de histórias com grau similar de dificuldade de compreensão. E a equipe B de forma similar discutiu um conjunto de histórias estimando e o outro conjunto sem estimar.

Para cada história foi extraída a mediana das respostas. E em seguida calculada a média da pontuação obtida por cada método. O que trouxe resultados consistentemente superiores para discussões sem estimativas.

Notamos assim que é possível compartilhar o entendimento de negócio das histórias sem estimar seu esforço.

Tabela 4.5: Comparação do entendimento de negócio das histórias pela Equipe A

HISTÓRIAS	DISCUSSÃO COM ESTIMATIVA	DISCUSSÃO SEM ESTIMATIVA
1	4	4
2	4	5
3	3	4
4	5	5
5	3	4
6	4	5
7	3	5
8	4	5
9	3	3
10	2	4
MÉDIA	**3,5**	**4,4**

Tabela 4.6: Comparação do entendimento de negócio das histórias pela Equipe B

HISTÓRIAS	DISCUSSÃO COM ESTIMATIVA	DISCUSSÃO SEM ESTIMATIVA
1	5	5
2	2	4
3	3	4
4	4	5
5	4	4
6	3	5
7	2	3
8	4	5
9	4	5
10	5	5
MÉDIA	**3,6**	**4,5**

4.4 Pergunta de Pesquisa 4

É possível compartilhar o entendimento de técnico das histórias sem estimá-las?

Obtivemos outro conjunto de 20 histórias de usuário relativas à equipe A na iteração seguinte. Pedimos então à equipe B que atribuísse um valor de 1 a 3 de acordo com a dificuldade de se entender os requisitos e possível estratégia técnicas das histórias. A partir desses valores as histórias foram divididas em dois conjuntos com grau de dificuldade similar.

Um dos conjuntos foi discutido pela equipe A tendo a conversa guiada pelo processo de estimativa de esforço, e o outro conjunto foi discutido também pela equipe A livremente sem realizar estimativas.

Cada uma das 8 pessoas desenvolvedoras e a pessoa analista de qualidade da equipe atribuíram, em seguida, um valor de 1 a 5 ao seu entendimento de negócio de cada história.

Em seguida repetimos o processo com a equipe A atribuindo valores de dificuldade de entendimento de negócio às 20 histórias da equipe B. E partir disso dividindo em dois conjuntos de histórias com grau similar de dificuldade de compreensão. A equipe B de forma similar discutiu um conjunto de histórias estimando e o outro conjunto sem estimar.

Para cada história foi extraída a mediana das respostas. E em seguida calculada a média da pontuação obtida por cada método. O que trouxe resultados consistentemente superiores para discussões sem estimativas.

Notamos assim que é possível compartilhar o entendimento técnico das histórias sem estimar seu esforço.

Tabela 4.7: Comparação do entendimento técnico das histórias pela Equipe A

HISTÓRIAS	DISCUSSÃO COM ESTIMATIVA	DISCUSSÃO SEM ESTIMATIVA
1	3	5
2	4	5
3	3	5
4	3	5
5	2	4
6	5	5
7	3	5
8	4	5
9	2	4
10	3	4
MÉDIA	**3,2**	**4,7**

Tabela 4.8: Comparação do entendimento técnico das histórias pela Equipe B

HISTÓRIAS	DISCUSSÃO COM ESTIMATIVA	DISCUSSÃO SEM ESTIMATIVA
1	3	5
2	2	4
3	3	5
4	4	5
5	4	5
6	5	5
7	2	4
8	3	5
9	4	5
10	3	5
MÉDIA	**3,3**	**4,8**

4.5 Pergunta de Pesquisa 5

É possível gerenciar iterações sem estimar o esforço das histórias?

Para investigar essa pergunta, foi levantado o questionamento de como analistas de negócio, gerentes de projeto e de iteração realizam o gerenciamento das iterações. Eis uma amostra das respostas:

> "Nós nem temos iterações, a gente utiliza o *kanban*, então a gente tem um *backlog* e um fluxo contínuo de histórias. Então a gente mede quanto tempo é gasto entre o momento que uma história entre em desenvolvimento até o quando ela aprovada pela cliente."
>
> "A gente só conta histórias. Elas têm mais ou menos o mesmo tamanho então dá pra ter uma ideia de quantas histórias a equipe consegue entregar por iteração."

As alternativas citadas podem ser categorizadas em:

O tempo de ciclo é o tempo que leva entre a escolha de um item para implementação até sua entrega. Ao contrário das estimativas de esforço que tentam prever o futuro, essa é uma medida

Tabela 4.9: Técnicas alternativas de gerenciamento de iterações

OCORRÊNCIAS	TÉCNICAS ALTERNATIVAS
3	Tempo de ciclo
6	Contagem de Histórias

do que de fato aconteceu. Do mesmo modo a contagem de histórias é uma medida de quantas histórias foram efetivamente entregues pela equipe.

Comprova-se assim a existência de métricas alternativas para gerenciamento de iterações.

4.6 Pergunta de Pesquisa 6

É possível gerenciar entregas sem estimar o esforço das histórias?

Para investigar essa pergunta, foi levantado o questionamento de como analistas de negócio, gerentes de projeto e de iteração realizam o gerenciamento das entregas. Eis uma amostra das respostas:

> "A gente estabelece um prazo curto, de algumas iterações. Mesmo que o cliente peça uma entrega só de longo prazo com escopo maior, a gente trabalha quebrando o escopo e fazendo sub-entregas."
>
> "Pela velocidade da equipe, a gente tira quantas cartas mais ou menos cabem na entrega e depois é uma questão de controlar o escopo de perto. Mas a gente não estima, só reprioriza o tempo todo."

As medidas citadas podem ser categorizadas em:

Estabelecer um prazo curto para as entregas, se presta ao propósito de ter um horizonte mais previsível de planejamento. A priorização do *backlog* garante que o trabalho está se focando no que for mais importante para o negócio.

Tabela 4.10: Técnicas alternativas de gerenciamento de entregas

OCORRÊNCIAS	TÉCNICAS ALTERNATIVAS
20	Priorizar *backlog*
10	Limitar prazo

Comprova-se assim a existência de formas alternativas para gerenciamento de entregas, prescindo as estimativas de esforço.

4.7 Pergunta de Pesquisa 7

É possível gerenciar projetos sem estimar o esforço das histórias?

Para investigar essa pergunta, foi levantado o questionamento de como analistas de negócio, gerentes de projeto e de iteração realizam o gerenciamento de projetos. Eis uma amostra das respostas:

> "Eu faço simulações de monte-carlo pra estimar orçamento, durante e velocidade de entrega do projeto."
>
> "A parte mais importante é conseguir pensar em um produto mínimo, que a gente consiga entregar algum valor ao cliente rápido, depois a gente vai incrementar em cima desse produto testando hipóteses e colhendo aprendizados."

As medidas citadas podem ser categorizadas em:

Tabela 4.11: Técnicas alternativas de gerenciamento de projeto

OCORRÊNCIAS	TÉCNICAS ALTERNATIVAS
13	Desenvolvimento Incremental a Partir de Produto Mínimo
4	Estimativas Probabilísticas Automatizadas

Projetos que se enquadram em perfis comuns podem utilizar bases de dados se outros projetos

semelhantes para fazer predições de desempenho utilizando algoritmos de aprendizado de máquina, ou caso o projeto funcione em um contexto estável, de equipe e negócio e disponha de dados históricos próprios abundantes é possível modelar seu desempenho e utilizar o modelo para realizar predições.

A definição de um produto mínimo faz o gerenciamento do projeto convergir para um gerenciamento de entrega, que evolui incrementalmente, com implementações que entregam valor no curto prazo e geram aprendizados para os passos seguintes sem a necessidade de estimativas de esforço com baixa confiabilidade dado o horizonte de pouca previsibilidade do longo prazo.

Comprova-se assim a existência de meios alternativos para gerenciamento de projeto sem a necessidade de se estimar o esforço das histórias.

CAPÍTULO 5

Considerações Finais

A pesquisa *Pulse of the Profession*, realizada anualmente desde 2006, comemorou em 2017 a primeira tendência de melhora nos índices de sucesso dos projetos. A própria definição de sucesso, contudo, sofreu reformulações. Pela primeira vez, perguntou-se: "O projeto atingiu a meta original do negócio?". Afinal é possível atingir objetivos de várias maneiras. Em particular, as metodologias ágeis não se prendem ao planejamento inicial. Ao contrário estimulam ponderações, negociações e alterações sobre escopo, prazo e orçamento, mantendo o foco na resolução dos problemas reais do negócio [31].

A mudança de abordagem ocorrida na *Pulse of the Profession* e consequentes resultados, demonstram que há bastante espaço para repensar práticas e entender melhor como adequar métricas às metodologias ágeis.

O presente trabalho identificou alternativas ao uso de estimativas de esforço no gerenciamento de projetos de software nas diferentes granularidades do planejamento ágil: histórias de usuário, iteração, entrega e projeto.

Foram investigados os usos das estimativas para informar tomadas de decisão de pessoas envolvidas no desenvolvimento e de pessoas em posições gerenciais. E como forma de aprofundar o entendimento das histórias pela equipe.

Para todos os casos foi identificada a existência de meios alternativos. As decisões das pessoas envolvidas no desenvolvimento relacionadas à implementação podem ser calcadas em discussões técnicas; as relacionadas a débitos técnicos podem usar matrizes de priorização e a quantidade de débitos técnicos pode ser pré-definida no planejamento da iteração. O controle da iteração pode se basear nos relatos de impedimentos nas reuniões diárias ou em um parâmetro

acordado de dias de desenvolvimento por história. Foi observado ainda que as equipes podem compartilhar o entendimento técnico e de negócio das histórias, discutindo-as sem o processo de estimativa.

No monitoramento de iterações, a métrica de contagem de histórias entregues na iteração e do tempo de ciclo, que calcula o tempo do início do desenvolvimento à entrega de uma história, se apresentaram como alternativas às estimativas de esforço.

No monitoramento de entregas, é possível prescindir o uso de estimativas de esforço limitando o prazo da entrega e priorizando o *backlog*.

No planejamento de projetos, a proposição de um produto mínimo viável no curto prazo torna o planejamento do projeto semelhante ao planejamento de uma entrega e é possível utilizar técnicas probabilísticas automatizadas caso o projeto se enquadre em alguns requisitos: detenha dados históricos e tenha um contexto estável ou pertença a um perfil comum de projetos com dados históricos disponíveis.

Há abundantes pesquisas acadêmicas estudando a precisão de estimativas e propondo aprimoramentos. Não foram encontrados, no entanto, trabalhos com menção a técnicas de gerenciamento de projetos sem o processo formal de estimar. Há, desse modo, inúmeras oportunidades de investigação nesse campo.

Como o impacto das decisões baseadas em estimativa na qualidade do projeto; se há relação entre a síndrome de esgotamento [29] e usos disfuncionais das estimativas; comparativos de acurácia entre as predições baseadas em diferentes métricas de planejamento e monitoramento de iterações, entregas e projetos.

Há novas tendências de métricas de entrega surgindo que podem ser experimentadas: como a quantidade de *deploys* de código em ambiente de produção, alinhadas com a filosofia de entrega contínua, ou mesmo de se substituir métricas genéricas por métricas de valor que retratem especificamente o problema sendo solucionado[26].

De modo geral, há carência de estudos empíricos que testem as alternativas aos métodos de estimativa tradicionais. Muitos profissionais usam seus próprios projetos como laboratórios e esbarram em restrições de confidencialidade ao tentar realizar estudos mais abrangentes. Logo, pesquisas primárias, que coletem dados, reproduzam experimentos e se somem aos resultados, tem grande valor para transcender experiências pessoais e obter generalizações de resultados confiáveis e úteis à comunidade.

Referências Bibliográficas

[1] THE STANDISH GROUP INTERNATIONAL **The CHAOS Report** Disponível em: <https://www.standishgroup.com>. Acesso em: 21 de outubro de 2018.

[2] ISAACS, M. **The #NoEstimates debate: An unbiased look at the origins, arguments, and thought leaders behind the movement.** Disponível em: <https://techbeacon.com/noestimates-debate-unbiased-look-origins-arguments-thought-leaders-behind-movement>. Acesso em: 4 de novembro de 2018.

[3] ROYCE, W. W. Managing the Development of Large Software Systems. **Technical Papers of Western Electronic Show and Convention**. Los Angeles: aug. 1970.

[4] LARMAN, C.; BASILI, V. R. Iterative and Incremental Development: A Brief History. **IEEE Computer Society Press**, v.36, n.6, p.2-11, jun. 2003.

[5] SUTHERLAND, J., SCHWABER, K. **The Scrum Guide.** Disponível em: <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Portuguese-Brazilian.pdf >. Acesso em: 28 de outubro de 2018.

[6] FOWLER, M. et al. **Agile Manifesto.** Disponível em: <http://agilemanifesto.org>. Acesso em: 21 de outubro de 2018.

[7] RICARDO, L. **Entendendo o Triângulo de Ferro, porque não podemos ter tudo.** Disponível em: <http://luizricardo.org/2013/09/entendendo-o-triangulo-de-ferro-porque-nao-podemos-ter-tudo/>. Acesso em: 9 de novembro de 2018.

[8] COHN, M. **Agile Estimating and Planning.** 1. ed. Massachusetts: Pearson Education, 2006. 308p.

[9] COHN, M. **Release Planning: Retiring the Term but not the Technique** Disponível em: <https://www.mountaingoatsoftware.com/blog/release-planning-retiring-term-not-technique>. Acesso em: 7 de novembro de 2018.

[10] COHN, M. **User Stories Applied: for agile software development.** 1. ed. Indiana: Addison-Wesley, 2009, 268p.

[11] MCCONNELL, S. **Software Estimation: Demystifying the Black Art.** 1. ed. Washington: Microsoft Press, 2006. 338p.

[12] GRENNING, J. **Planning Poker or How to avoid analysis paralysis while release planning.** Disponível em: <https://wingman-sw.com/papers/PlanningPoker-v1.1.pdf>. Acesso em: 31 de outubro de 2018.

[13] MIRANDA, E. Improving Subjective Estimates Using Paired Comparisons. **IEEE Software** v.18, n.1, p87-91, jan. 2001.

[14] SANTOS, C. G. **Um Estudo Empírico sobre a Gerência de Dívida Técnica em Projetos de Desenvolvimento de Software que Utilizam SCRUM**. 2015. Dissertação (Mestrado em Ciências da Computação) – Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre.

[15] BAKARDZHIEV, D. **#NoEstimates Project Planning Using Monte Carlo Simulation**. Disponível em: <https://www.infoq.com/articles/noestimates-monte-carlo>. Acesso em: 23 de novembro de 2018.

[16] SAPRE, A. V. **Feasibility of Automated Estimation of Software Development Effort in Agile Environments.** 2012. Dissertação (Graduação em ciências da computação e engenharia) Universidade do estado de Ohio, Ohio.

[17] IONESCU, V. S.; DEMIAN, H.; CZIBULA, I. G. Natural Language Processing and Machine Learning Methods for Software Development Effort Estimation. **Studies in Informatics and Control**, v.26, n.2, p219-228, jun. 2017.

[18] BASHA, M. S.; PONNURANGAM, D. **Analysis of Empirical Software Effort Estimation Models.** Disponível em: <http://arxiv.org/abs/1004.1239>. Acesso em: 4 de novembro de 2018.

[19] ZHANG, Z. **The Benefits and Challenges of Planning Poker in Software Development: Comparison Between Theory and Practice.** 2017. Dissertação (Mestrado em Ciências da Computação e da Informação) – Escola de ciências da computação e matemática, Universidade de Tecnologia de Auckland, Auckland.

[20] ZIAUDDIN, Z., et al. An Effort Estimation Model for Agile Software Development. **Advances in Computer Science and its Applications.** v.2, n.1, p.314-324, jul. 2012.

[21] COHN, M. **How to Estimate Velocity As an Agile Consultant.** Disponível em: <https://www.mountaingoatsoftware.com/blog/how-to-estimate-velocity-as-an-agile-consultant>. Acesso em: 15 de novembro de 2018.

[22] SCRUMINC. **Yesterday's Weather.** Disponível em: <https://www.scruminc.com/yesterdays-weather/>. Acesso em: 15 de novembro de 2018.

[23] ZUILL, W. **No Estimate Programming Series – Intro Post** Disponível em: <http://zuill.us/WoodyZuill/2012/12/10/no-estimate-programming-series-intro-post/>. Acesso em: 5 de novembro de 2018.

[24] DUARTE, V. **Story Points Considered Harmful: Or why the future of estimation is really in our past...** Disponível em: <http://softwaredevelopmenttoday.blogspot.com/2012/01/story-points-considered-harmful-or-why.html>. Acesso em: 29 de outubro de 2018.

[25] KILLICK, N. **Should We Estimate Software Projects... At All?** Disponível em: <https://neilkillick.wordpress.com/2012/04/12/do-not-estimate-software-projects-at-all/>. Acesso em: 28 de outubro de 2018.

[26] KIM, G.; HUMBLE, J.; FORSGREN, N. **Accelerate: Building and Scaling High Performing Technology Organizations**. 1. ed. Portland: IT Revolution, 2018. 257.

[27] MANFREDO, T. **Resumo de fórmulas para propagação de incertezas.** Disponível em: <http://efisica.if.usp.br/mecanica/universitario/incertezas/formulas/>. Acesso em: 13 de novembro de 2018.

[28] THE STANDISH GROUP INTERNATIONAL **CHAOS: A Recepy for Success (1999).** Disponível em: <http://www.dsc.ufcg.edu.br/ garcia/cursos/ger_processos/artigos/chaos 1998.pdf>. Acesso em: 21 de outubro de 2018.

[29] WIKIPEDIA **Síndrome de burnout** Disponível em: <https://pt.wikipedia.org/wiki/S%C3%ADndrome_de_burnout>. Acesso em: 15 de novembro de 2018.

[30] SAURABH, S. et al. A Systematic Review on Software Cost Estimation in Agile Software Development. **Journal of Engineering Science and Technology Review.** v. 10, p. 51-64, aug. 2017.

[31] PROJECT MANAGEMENT INSTITUTE. **Pulse of the profession (2017).** Disponível em: <https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession-2017.pdf?sc_lang_temp=pt-PT>. Acesso em: 30 de outubro de 2018.