UNIVERSIDADE FEDERAL DO CEARÁ

CAMPUS DE QUIXADÁ

CURSO DE GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO

EDIR LUCAS DA SILVA ICETY BRAGA

REVISÃO SISTEMÁTICA DOS FATORES DE IMPACTO NA ESTIMATIVA DE ESFORÇO EM PROJETOS DE SOFTWARE.

QUIXADÁ

2019

EDIR LUCAS DA SILVA ICETY BRAGA

REVISÃO SISTEMÁTICA DOS FATORES DE IMPACTO NA ESTIMATIVA DE ESFORÇO EM PROJETOS DE SOFTWARE.

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Sistemas de Informação do Campus de Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Sistemas de Informação.

Orientadora: Profª. Ma. Antonia Diana Braga Nogueira

QUIXADÁ

2019

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

B793r Braga, Edir Lucas da Silva Icety.
Revisão sistemática dos fatores de impacto na estimativa de esforço em projetos de software / Edir Lucas da Silva Icety Braga. – 2019.
58 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Sistemas de Informação, Quixadá, 2019.
Orientação: Profa. Ma. Antonia Diana Braga Nogueira.

1. Software de sistemas. 2. Métricas de software. 3. Estimativa de software. 4. Projetos de software. I. Título.

CDD 005

EDIR LUCAS DA SILVA ICETY BRAGA

REVISÃO SISTEMÁTICA DOS FATORES DE IMPACTO NA ESTIMATIVA DE ESFORÇO EM PROJETOS DE SOFTWARE.

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Sistemas de Informação do Campus de Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Sistemas de Informação.

Aprovada em: ___/___/___

BANCA EXAMINADORA

Profª. Ma. Antonia Diana Braga Nogueira (Orientadora)
Universidade Federal do Ceará (UFC)

Profª. Drª. Paulyne Matthews Jucá
Universidade Federal do Ceará (UFC)

Prof. Júlio Serafim Martins
Universidade Federal do Ceará (UFC)

À minha vó Meire, seu cuidado e dedicação foi que deram, em alguns momentos, a esperança para seguir. Tio Marcos, sua presença significou segurança e certeza de que não estou sozinho nessa caminhada.

AGRADECIMENTOS

Agradeço a minha orientadora Diana pela confiança, pelos ensinamentos. Foi um grande honra e um grande prazer poder realizar este trabalho com a sua ajuda e paciência. Obrigado pelo ano de convivência, pelo o aprendizado que levarei para o restante da minha vida.

Agradeço aos meus pais, por sempre me apoiarem nas minhas decisões, me direcionarem pelo caminho certo. Aos meus irmãos, a minha família, padrinhos e amigos, colegas e professores.

Aos meus amigos João Paulo, Nathan Lima, Antônio Alves, Lucas Rodrigues, Daniel Lima, por tornarem a caminhada mais fácil. Sempre me recordarei desse tempo com um sorriso no rosto. Sucesso a todos vocês.

A todos que passaram por minha vida e contribuiram de alguma forma para que este sonho pudesse torna-se realidade.

A todos os professores do campus Quixadá. Obrigado por toda experiência passada.

A minha vó Meire e meu tio Marcos, por me moldarem, me direcionarem pelo caminho certo e me ensinarem a seguir sempre os estudos.

Agradeço a minha fé e persistência, por sempre acreditar que o amanhã será melhor, pelo meu esforço e dedicação nesses anos, afinal, Deus é justo.

"Sonhos determinam o que você quer. Ação determina o que você conquista."

(Aldo Novak)

RESUMO

Contexto: a engenharia de *software* surgiu para tornar possível que sistemas complexos possam ser desenvolvidos dentro do prazo e com alta qualidade. A necessidade por estimativas de esforço de *software* cada vez mais precisas e eficientes é evidente no mercado de *software*. Estimar o esforço é uma tarefa muito importante para projetos de *software* e de certa dificuldade de ser realizada com precisão satisfatória. Existem fatores que influenciam essas estimativas de forma direta ou indireta, positiva ou negativamente. Objetivo: este trabalho visa identificar estudos significativos realizados no contexto de estimativa de esforço, verificar fatores que influenciaram projetos de *software* de alguma maneira, e, por fim, demonstrá-los. Método: para alcançar esse objetivo, foi utilizada a metodologia de revisão sistemática de literatura, com as seguintes etapas: 1. Revisão de literatura. 2. Planejamento da revisão sistemática. 3. Realização da revisão sistemática. 4. Responder as perguntas de pesquisa. Ao todo, foram 859 trabalhos de três bases de dados, sendo 53 estudos aceitos para extração de dados. Resultados: os resultados iniciais de concordância calculados da fase de seleção dos estudos, onde foi realizada em paralelo com um colaborador, foram satisfatórios, visto que o índice de menor porcentagem foi de 84,4%. A partir da análise dos estudos relevantes, verificou-se 29 fatores que influenciam a estimativa de esforço. Conclusão: a compreensão das atividades de gerenciamento de projetos desde a sua fase inicial são importantes para evitar falhas de estimativa, e, consequentemente, atrasos no cronograma e custos excedidos. Sendo assim, o conhecimento sobre os fatores que influenciam a estimativa de esforço é relevante para a indústria por possibilitar que os estimadores e gerentes de projetos antecedam os erros ocasionados pelo fatores. Além disso, auxiliam organizações que visam o comprometimento com as estimativas especificadas no início do projeto, beneficando todas as partes interessadas.

Palavras-chave: Software de sistemas. Métricas de software. Estimativa de tamanho. Projetos de software.

ABSTRACT

Context: software engineering has emerged to make it possible for complex systems to be developed on time and with high quality. The need for increasingly accurate and efficient software effort estimates is evident in the software market. Estimating effort is a very important task for software projects and of right difficulty to be performed with satisfactory accuracy. There are factors that influence these estimates directly or indirectly, positively or negatively. Objective: this paper aims to identify significant studies carried out in the context of effort estimation, to verify factors that influence software projects in some way, and finally to demonstrate them. Method: to achieve this goal, we used the systematic literature review methodology, with the following steps: 1. Literature review. 2. Systematic review planning. 3. Conducting the systematic review. 4. Answer the survey questions. Answer the survey questions. In all, there were 859 papers from three databases of which were accepted 53 for data extraction. Results: the initial agreement results calculated from the study selection phase where it was performed, in parallel with a collaborator, were satisfactory, since the lowest percentage index was 84.4%. From the analysis of the relevant studies, 29 factors that influence the effort estimate were verified. Conclusion: understanding project management activities from the outset is important to avoid estimation failures, and consequently schedule delays and cost overruns. Therefore, knowledge about the factors that influence effort estimation is relevant for the industry, as it allows project estimators and managers to anticipate the errors caused by the factors. In addition, they assist organizations that aim to commit to the estimates specified at the beginning of the project, benefiting all stakeholders.

Palavras-chave: Systems software. Software Metrics. Software Estimation. Software projects.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

A engenharia de *software* é uma disciplina com foco na "aplicação de uma abordagem sistemática, disciplinada e quantificável no desenvolvimento, na operação e na manutenção de *software*"(VLIET *et al.*, 2008)(PRESSMAN; MAXIM, 2016). A engenharia de *software* surgiu para tornar possível que sistemas complexos sejam desenvolvidos dentro de um prazo e com alta qualidade. Ela permite que um *software* de computador seja criado com base nos critérios estabelecidos pelo cliente, de acordo com suas necessidades (PRESSMAN; MAXIM, 2016).

De forma geral, “um processo de *software* é definido como um conjunto de atividades relacionadas que levam à execução de um produto de *software*” (SOMMERVILLE, 2011). O processo de *software* fornece a metodologia por meio da qual um plano de projeto abrangente para o desenvolvimento de *software* pode ser especificado. Essa metodologia de processo genérica compreende cinco atividades: comunicação, planejamento,modelagem, construção, entrega . É importante lembrar, que, em um sistema, são descritas as características da organização de desenvolvimento como a experiência, ambiente e o *software* a ser desenvolvido. Com base nessas características é estimado esforço e custo (PRESSMAN; MAXIM, 2016).

Segundo Vliet *et al.* (2008), os problemas surgem quando um *software* excede prazos, orçamentos e possui níveis de qualidade reduzidos, causando a perda de confiabilidade. Portanto, é importante identificar e analisar os fatores que minimizam esses problemas durante o desenvolvimento de *software*, a fim de melhorar a precisão da estimativa de esforço (NGUYEN *et al.*, 2013). Algumas técnicas já foram estabelecidas para tratar e melhorar os desafios da estimativa de esforço, ambas utilizam o julgamento e experiências anteriores do gerente de projetos e o esforço empreendido em atividades relacionadas ao desenvolvimento de *software* (SOMMERVILLE, 2011).

Segundo Pressman, estimativa de esforço é o processo de analisar em quanto tempo um projeto será realizado, de acordo com as métricas estabelecidas, técnicas de decomposição etc. Essa fase é crítica e de suma importância para analisar o grau com que foi estimado adequadamente o tamanho do produto a ser criado, habilidade para traduzir esforço humano, o grau com que o plano do projeto reflete as habilidades da equipe e a estabilidade dos requisitos (PRESSMAN; MAXIM, 2016).

Resultados de pesquisas Molokken e Jorgensen (2003) indicam que o esforço empenhado em projetos de TI exedem as estimativas estabelecidas no processo de definicão do software em 60% a 80%, em média. Sendo assim, o estudo dos fatores que minimizam os

erros em estimativas de *software* se torna relevante para os *stakeholders* e para o crescimento dos números de projetos de *software* concluídos no prazo e dentro dos custos estabelecidos inicialmente.

A estimativa de esforço é uma atividade importante da fase de planejamento de um projeto de *software*. Sua importância é facilmente percebida pelo fato que a estimativa de esforço guiará o planejamento de todas as atividades seguintes no desenvolvimento de *software*. A identificação e análise desses fatores relacionados a estimativa de esforço é determinante para que projetos de *software* possam cumprir seu cronograma (HARMAN *et al.*, 2010). A compreensão desses fatores é essencial para que os stakeholders possam desenvolver medidas para mitigar os fatores negativos e maximizar os efeitos dos fatores positivos.

As informações utilizadas neste trabaho foram obtidas por meio de revisão sistemática da literatura, que trata-se de uma forma de estudo secundário que utiliza uma metodologia para identificar, analisar e interpretar todas as evidências disponíveis relacionadas a uma ou mais perguntas de pesquisa (KITCHENHAM; CHARTERS, 2007). Como o objetivo deste trabalho é encontrar fatores que contribuem de forma positiva ou negativa, foi realizada uma revisão sistemática que consistiu em pesquisar, em base de dados pré-estabelecidas, trabalhos primários relacionados que obtenham conceitos relacionados ao cenário deste trabalho (KITCHENHAM; CHARTERS, 2007).

Nesse contexto, este trabalho especifica todos os fatores relatados a partir da revisão sistemática, sejam fatores positivos ou negativos, e responde as perguntas de pesquisa estabelecidas dentro do contexto do trabalho. As informações utilizadas para responder as perguntas de pesquisa foram obtidas através de revisão sistemática estabelecida por (KITCHENHAM; CHARTERS, 2007), que tem como processos planejar a revisão, conduzir a revisão e, por fim, relatar as conclusões. O objetivo principal foi coletar o máximo possível de informações sobre os fatores que impactam a estimativa de esforço, de acordo com os critérios estabelecidos, e utilizá-los como base para responder a questão de pesquisa principal. Por fim, as perguntas de pesquisa secundárias foram respondidas para melhorar a compreensão do contexto geral dos fatores exemplificados neste estudo.

Este trabalho está organizado da seguinte forma: o capítulo 2 apresenta os trabalhos relacionados. O capítulo 3 a fundamentação teórica necessária para o entendimento deste trabalho. O capítulo 4 mostra a descrição dos procedimentos metodológicos que foram feitos no trabalho, contendo todos os passos. O capítulo 5 possui o protocolo que foi adotado nesta

revisão. O capítulo 6 apresenta os resultados. Por fim, o capítulo 7 apresenta as conclusões e trabalhos futuros.

2 TRABALHOS RELACIONADOS

Nesta seção, serão apresentados alguns trabalhos relacionados destacando as semelhanças e diferenças com o desenvolvido neste trabalho.

A estimativa de esfoço no contexto de desenvolvimento de *software* é um tema investigado em diversos trabalhos (O'KEEFFE, 2012; LAQRICHI *et al.*, 2015; MOLOKKEN; JORGENSEN, 2003; MENDONÇA; ALENCAR, 2019) utiliza diferentes abordagens de gerenciamento de projetos dentro do contexto organizacional para identificar fatores que influenciam o erro de estimativas de esforço.

No trabalho de O'Keeffe (2012), a investigação ocorreu através de um questionário estruturado como instrumento de pesquisa pré-avaliado por especialistas com o objetivo de refinar e alcançar melhorias para a obtenção de uma versão final. O autor também utilizou uma revisão teórica para identificar fatores de impacto no erro de estimativa descritos durante qualquer fase do projeto. Foram encontrados 26 fatores de impacto com a revisão da literatura e mais 5 fatores com a pesquisa feita com especialistas, totalizando 31 fatores que ocasionam o erro de estimativa de *software*. A partir da obtenção dos fatores, foi utilizado uma survey auto administrada como forma de comunicação com os participantes da pesquisa. Nessa abordagem, foi enviado um questionário com questões pré-definidas aos respondentes.

Em Laqrichi *et al.* (2015), são identificados eventos de risco que mais se manifestam em projetos de *software*. Esses fatores são identificados através de entrevistas com especialistas em gerenciamento de *software*. Esse trabalho, propõe uma metodologia para levar em conta a análise da exposição ao risco no modelo de estimativa de esforço. Na metodologia apresentada, a identificação dos eventos de risco de *software* é a primeira tarefa realizada, logo depois sendo realizada a avaliação de risco dentro do processo que consiste em quantificar a importância e criticidade desses eventos de risco.

Dentro da abordagem proposta pelo trabalho de Laqrichi *et al.* (2015), o processo de estimativa de esforço integrando riscos consiste em quatro etapas: a identificação, a análise de risco, avaliação de risco e a estimativa de esforço. Desse modo, métodos e ferramentas são utilizados em cada etapa e o foco é dado na etapa de estimativa de esforço para propor um modelo de estimativa de esforço que integre os riscos, com base no banco de dados de projetos finalizados, que é a meta do estudo. Para a realização dessa estrutura, foram definidas etapas que consistem na seleção de parâmetros para definir os impulsionadores do esforço de desenvolvimento com base em um conjunto de dados. Em seguida, preparar os dados e avaliar os

riscos do projeto de acordo com o banco de dados de projetos concluídos, depois apresentar um modelo de estimativa de esforço integrando os riscos de projetos. E por fim, utilizar indicadores de precisão para validar o modelo proposto.

A pesquisa de Molokken e Jorgensen (2003) procura gerar uma discussão sobre estimativa de esforço, através de revisão de pesquisas relevantes para o campo de projeto de *software*, demonstrando uma visão geral desses estudos a fim de responder quatro perguntas pré-estabelecidas. É exemplificado um resumo de cada projeto de pesquisa e seus focos de pesquisa com o objetivo de descobrir como os projetos são estimados e como seu nível de precisão pode influenciar a indútria de *software*. As questões de pesquisa abordam aspectos centrais da estimativa de *software* para permitir um debate de estimativa mais equilibrado e sugerir novas pesquisas. As principais respostas, de acordo com seu grau de importância, demonstraram que a comparação com projetos semelhantes realizados anteriormente guardados na memória pessoal e em fatos documentados são as duas categorias classificadas com maior notabilidade.

O estudo de Molokken e Jorgensen (2003), apresenta como proposta para pesquisas futuras uma investigação mais detalhada sobre aspectos relacionados a quando, como e porque os métodos de estimação são escolhidos. E desta forma, compreender como a escolha dos métodos, ou combinações de métodos, podem ser influenciados por propriedades como tipo ou tamanho.

No trabalho de Mendonça e Alencar (2019), busca-se responder a pergunta de pesquisa principal que preocupa-se em entender o impacto da medição de tamanho de *software* no emprego de métodos ágeis no setor público. A pesquisa identifica as principais métricas de tamanho de *software* juntamente com os métodos ágeis utilizados no setor público. Como metodologia, o autor identifica essas evidências através de revisão sistemática de litaratura. A partir da definição dessas configurações, do uso de métricas de tamanho e métodos ágeis no setor público, a pesquisa busca direcionar a tomada de decisões para que os projetos de *software* possam cumprir seu cronograma, sejam de tempo ou custo.

A Tabela 1 apresenta as principais semelhanças e diferenças entre este trabalho e os trabalhos analisados.

O trabalho de O'keeffe (O'KEEFFE, 2012) associa-se diretamente com o objetivo deste trabalho, pois o mesmo utiliza fatores e técnicas que contribuem para o erro de estimativa para realizar uma pesquisa estruturada com o objetivo de analisar os principais problemas que causam o erro na estimativa de esforço e na duração de projetos de *software*. A identificação desses fatores é um ponto em comum, porém o objetivo principal deste trabalho é responder a

Tabela 1 – Comparação entre os trabalhos relacionados e o proposto.

Critério	**(O'KEEFEE, 2012)**	**(LAQRICHI; GOURC; MARMIER, 2015)**	**(MOLØKKEN; JØRGENSEN, 2003)**	**(MENDONÇA; ALENCAR, 2019)**	**Trabalho Proposto**
Foco na análise	Fatores e técnicas de estimativa.	Riscos de *software*.	Métodos de uso para estimativa.	Métricas de tamanho de *software* e métodos ágeis.	Fatores de impacto na estimativa de esforço.
Foco no processo de desenvolvimento	Todo o processo.	Todo o processo.	Todo o processo.	Fase inicial da estimativa.	Todo o processo.
Metodologia de coleta de dados	Revisão teórica, entrevista com especialistas e survey.	Utiliza abordagens para propor um novo modelo de estimativa.	Análise de survey's e questões de pesquisa.	Revisão sistemática.	Revisão sistemática.

Fonte: Elaborado pela autor.

questão de pesquisa principal, que objetiva demonstrar todos os fatores que afetam a estimativa de esforço. Além disso, a metodologia utilizada por O'Keeffe (2012) é baseada em uma survey, abordagem que utiliza questionários para obtenção de respostas para as questões estabelecidas anteriormente, a partir da revisão de literatura e entrevistas com especialistas, com a finalidade de demonstrá-los de maneira quantitativa e, neste trabalho, foi realizada uma revisão sistemática da literatura com um protocolo previamente bem definido.

O trabalho de Laqrichi *et al.* (2015) relaciona-se com este estudo de forma indireta, pois o mesmo realiza um levantamento de fatores e eventos de risco de *software* que ocorrem com frequência com o objetivo de melhorar a precisão na estimativa de esforço de *software*. Segundo os autores o risco é um fator importante que impede o projeto de *software* de estar no tempo e dentro do orçamento, sendo preciso levá-lo em conta durante o processo de estimativa de esforço, tempo e custo. Sendo assim, o objetivo do trabalho é propor uma abordagem que integre riscos de *software* ao processo de estimar *software* com a finalidade de ajudar o gerente de projeto e os estimadores em todo o processo de estimativa de esforço fornecendo uma metodologia e suas ferramentas e suportes associados. Portanto, este trabalho realizou um levantamento de fatores, de forma geral, que contribuem, ou não, de alguma forma com o processo de estimativa de esforço enquanto que o trabalho de Laqrichi *et al.* (2015) cria uma metodologia que minimiza

erros de estimativa a partir da priorização de riscos de *software*.

A principal diferença da pesquisa de Molokken e Jorgensen (2003) para este trabalho é que o estudo de pesquisas tem foco em responder perguntas relacionadas ao desvio do plano original do projeto, quais métodos são utilizados e até que ponto a precisão é um problema para a indústria de *software* e quais são suas principais causas. Esse trabalho resume os resultados das pesquisas sobre estimativas de *software*. Além disso, outro objetivo principal é buscar desafiar algumas crenças comuns de estimativa que podem ser imprecisas, discutir aspectos metódicos relacionados à realização de pesquisas sobre estimativa de esforço de *software*. Enquanto que este trabalho elicita todos os fatores de impacto, em todo o processo de estimativa de esforço, obtidos através de revisão sistemática de literatura.

O trabalho de Mendonça e Alencar (2019) correlaciona-se diretamente com este trabalho, pois o mesmo mensura entregas de projetos de *software* baseadas em alguma métrica de tamanho de *software*, atribuindo a escolha dessa métrica ao sucesso de projetos no setor público. A metodologia utilizada na revisão sistemática de literatura é semelhante a utilizada neste trabalho, pois é definido um protocolo que engloba perguntas de pesquisa, *string* de busca, bases de dados, critérios de inclusão/exclusão, critérios de qualidade, objetivos de pesquisa e procedimentos de seleção. Portanto, este trabalho responde a uma pergunta de pesquisa central relacionada a identificação de fatores de impacto na estimativa de esforço e a pesquisa de Mendonça e Alencar (2019) se preocupa em responder qual o impacto das métricas utilizadas na medição de tamanho de *software* no emprego de métodos ágeis no setor público.

3 FUNDAMENTAÇÃO TEÓRICA

Nesta seção, é apresentada a fundamentação teórica necessária para o entendimento e o embasamento deste trabalho.

3.1 Desenvolvimento de software

O processo de desenvolvimento de *software* abrange quatro fases comuns, independente da opção de paradigma de engenharia de *software*, todas elas possuem conceitos básicos em sua abordagem. As fases são: definição, desenvolvimento, operação e manutenção (SOMMERVILLE, 2011).

Na definição, são identificadas as informações a respeito da função e desempenho desejados, assim como qual interface devem ser estabelecidas, as restrições de desenvolvimento e quais critérios de validação serão necessários para que o *software* seja terminado. No desenvolvimento, são feitas as definições de arquitetura de *software* e estrutura de dados. Nessa fase, também são definidos os procedimentos de implementação, e como o projeto será traduzido para uma linguagem de programação. Por último, são realizados os testes para que se possa identificar defeitos de função, lógica e implementação (PRESSMAN; MAXIM, 2016).

Na fase operacional, os usuários poderão utilizar efetivamente o sistema desenvolvido para que se obtenha um *feedback* em relação aos resultados especificados pelo cliente na fase inicial de definição dos requisitos do software. Na manutenção, são feitas correções de código, adaptações exigidas à medida que o *software* evolui e precisa se enquadrar, de acordo com novas exigências que o ambiente de *software* exige (PRESSMAN; MAXIM, 2016).

Existem dois enfoques no desenvolvimento de *software*. O primeiro é o desenvolvimento artesanal, que se baseia em habilidades exclusivas de acordo com cada desenvolvedor para construir um *software*. É um processo simplificado que depende de uma simples conversa do desenvolvedor com o cliente, sem definição de critérios como análise, projeto e testes. Essa técnica é conhecida como “tentativa e erro”. Essa técnica pode funcionar em alguns casos, mas acaba sendo arriscada para o ambiente evoluído e amplo que as indústrias necessitam atualmente (GARCIA, 2013).

No desenvolvimento baseado na Engenharia de *Software*, a abordagem é feita de forma organizada, sistemática, produtiva e econômica. Toda a abordagem se preocupa com a aplicação de suas etapas bem definidas, com documentação e uma equipe especializada

responsável pelo projeto (GARCIA, 2013).

A utilização de aspectos técnicos da qualidade de *software* se torna cada vez mais necessária. De modo que, é preciso definir e empregar princípios consistentes de engenharia para obter *software* econômico, confiável e eficiente já que a engenharia de *software* aborda métodos, ferramentas e procedimentos para o desenvolvimento de *software* (PRESSMAN; MAXIM, 2016).

O Guia PMBOK (INC, 2017), define um projeto como um esforço temporário que será empenhado na criação de um produto, serviço ou resultado exclusivo, contendo uma data para início e término pré-estabelecidos. Segundo Sommerville (2011), durante um projeto, o plajamento é divido em três etapas que definem o ciclo de vida do projeto. Estas etapas se resumem em quando uma pessoa se propõe a desenvolver ou fornecer um *software*, quando é preciso arquitetar quem realizará os esforços necessários; dividivir as tarefas em incrementos; e quais recursos seram disponibilizados; e por fim, adquirir experiência e informações com as modificações periódicas no planejamento.

3.2 Técnicas de estimativa de esforço

Diversas técnicas de estimativas de esforço são utilizadas em projetos de *software*. As duas categorias que abrangem todas essas técnicas são baseadas em modelos e de opinião especializada (JØRGENSEN, 2004).

Vários fatores acompanham e dificultam as estimativas de esforço como complexidade, grau da estrutura, tamanho do projeto, o escopo mal compreendido e requisitos que quase sempre são alterados durante o decorrer do projeto (PRESSMAN; MAXIM, 2016). Portanto, é necessário utilizar técnicas adequadas de acordo com cada projeto para tentar obter o máximo de precisão na tarefa de estimar o esforço.

Para tentar alcançar estimativas precisas, existem algumas alternativas, que são (i) Usar técnicas de estimativa; (ii) Usar um modelo de estimativa; (iii) Utilizar uma ou mais métricas de estimativa (O'KEEFFE, 2012).

Dentre as técnicas de estimativa, uma das alternativas é usar a de técnicas de decomposição, que é usada a abordagem "dividir para conquistar" para concluir com as estimativas através do processo de decompor o projeto nas funções e em atividades menores, assim fica mais fácil a tarefa de estimar fragmentos de atividades do que todo o projeto (VALENTE; FALBO, 2002). Outra alternativa, é utilizar técnicas baseadas em experiência, que se baseia na experiência

de gerentes em projetos de *software* concluídos anteriormente e em seu domínio de aplicação para estimar futuros esforços.

Vários modelos similares surgiram com o intuito de ajudar nas estimativas de esforço, cronograma e custos. Para a estimativa, existem vários modelos de estimativa e os mais conhecidos são COCOMO II, SLIM, COSMIC, Checkpoint, dentre outros.

O COCOMO II é um modelo de estimativa utilizado para medir o esforço, prazo e tamanho da equipe em projetos de software. Este modelo utiliza equações matemáticas baseadas em pesquisas, dados históricos e em coleta de dados de uma grande quantidade de projetos de *software*. Para com isso, analisar essas informações para buscar as fórmulas que mais se ajustam a situação (PRESSMAN; MAXIM, 2016). O modelo se baseia em pontos de função e em medidas de tamanho como: linhas de código e tem como característica fundamental o modelo aberto que permite que o estimador compreenda o porque da estimativa fornecida pelo modelo (PRESSMAN; MAXIM, 2016) (BOEHM *et al.*, 2000).

Já com modelo SLIM, usa-se da curva *Rayleigh* para estimar que pode ser influenciada por fatores como: índice de força de trabalho, produtividade, dentre outros. O modelo pode salvar e analisar dados de projetos realizados anteriormente para ajustar o modelo e como medida de segurança, caso os dados não estiverem disponíveis utilizar de questionários para obter esses valores no banco de dados. Com o modelo é possível estimar tempo, esforço, e custos de um conjunto de requisitos (BOEHM *et al.*, 2000).

Finalmente, o modelo *Checkpoint* é um modelo proprietário e vendido em ferramentas. O modelo possui bancos de dados extensos e se baseia em pontos de função, como dado de entrada para estimar o tamanho do projeto, para analisar milhares de projetos dentro do banco de dados da empresa proprietária (BOEHM *et al.*, 2000). O modelo é capaz de estimar esforço em quatro níveis: projeto, fase, atividade e tarefa.

Em relação as métricas de estimativa, são utilizadas como medição de um atributo de uma determinada entidade como: produtos, processos ou recursos. É necessário medir um *software* para entender e aperfeiçoar processos de desenvolvimento. Como também, para melhorar a gerência de projetos e o relacionamento com os clientes. Para isso é importante que se faça uma avaliação efetiva e coerente da extensão de tempo que o projeto exigirá, para realizar a programação de um sistema (O'KEEFFE, 2012).

Uma medida que pode ser utilizada para realizar estimativas do tamanho do produto de *software* é o número de linhas de código (SLOC – *Source lines of code*). Essa medida tem por

objetivo medir o tamanho do *software* através da contagem do número de linhas em um texto do código fonte. Esta medida é normalmente utilizada para prever o esforço que será necessário para desenvolver um determinado sistema e a produtividade do esforço, para logo após o *software* ser produzido (PUTNAM; MYERS, 2013).

Por fim, outra medida bastante utilizada em projetos de desenvolvimento de *software* é a análise de pontos de função (APF), que é utilizada para medir o tamanho com base em pontos de função. A métrica qualifica quantitativamente as funções que existem em determinado *software* e as atribui um índice numérico em razão do seu tipo e complexidade. Deste modo, determina um valor único em pontos de função que permita avaliar o tamanho do produto (LOPES, 2011).

Os conceitos e exemplos das três alternativas sugeridas são apenas algumas das possibilidades existentes nos processos que envolvem a estimativa de esforço. Portanto, o objetivo principal deste estudo é buscar fatores que abrangem essas alternativas e demonstrá-las no estudo. Alguns desses possíveis fatores são descritos na próxima seção.

3.3 Fatores que impactam a estimativa de desenvolvimento de *software*

A melhoria na precisão das estimativas de esforço, contribuem em larga escala para uma organização que busca direcionar suas atividades e cumprir com seus compromissos. Com a necessidade de melhorar as estimativas no desenvolvimento de projetos de *software*, alguns dos fatores que dificultam a tarefa de estimativa no desenvolvimento de *software* são detalhados a seguir:

- Clareza dos objetivos: isso ocorre quando as metas estão bem definidas e repassadas aos envolvidos com nitidez (BERKUN, 2005).
- Cruzamento de estimativas: neste caso, é importante priorizar o uso de mais de uma técnica de estimativa. Assim, obter duas estimativas com o objetivo de compará-las e buscar convergências. Se essas diferenças se apresentarem muito significativas, buscar seus motivos e corrigi-los (MCCONNELL, 2006).
- Pressão externa para redução das estimativas: esse é um dos fatores mais discutidos e expostos por muitos autores que causam complicações nas estimativas de *software*. Pois influências externas como, por exemplo, uma estratégia de venda pode afetar essas estimativas. Outra razão é quando clientes que não possuem conhecimentos de gerenciamento de

projetos de *software* não aceitam as estimativas válidas e usam de sua autoridade para rejeitá-las (JORGENSEN; MOLOKKEN-OSTVOLD, 2004).

- Experiência em projetos de *software*: segundo (MORGENSHTERN *et al.*, 2007), a experiência em projetos realizados anteriormente está associada ao erro de estimativas, porém profissionais que tenham tido maior envolvimento na área do projeto atual são capazes de estimar com maiores chances de acerto.

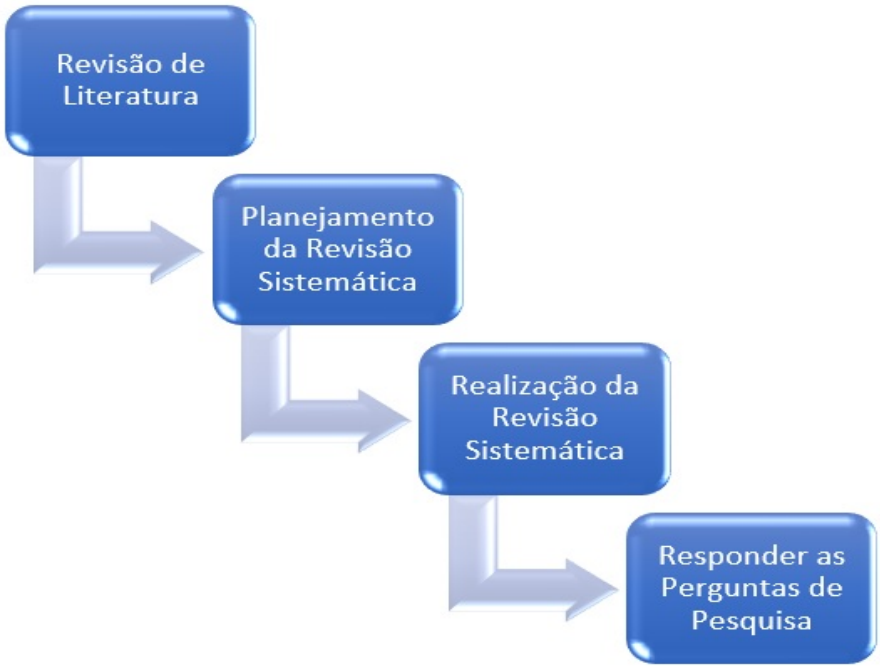
Esses são alguns fatores que dificultam a consistência no processo de estimativa em projetos de *software*. Este trabalho, encontrou por meio de revisão sistemática fatores que influênciam diretamente, ou de maneira indireta na estimativa de esforço em todo o processo do desenvolvimento de *software* e explaná-los na resposta da pergunta de pesquisa principal exemplificada na metodologia deste trabalho.

No próximo capítulo, a metodologia deste trabalho é descrita.

4 PROCEDIMENTOS METODOLÓGICOS

A metodologia do trabalho adotatada foi a revisão sistemática da literatura (KITCHENHAM; CHARTERS, 2007). Os trabalhos e dados relacionados aos fatores que influenciam a estimativa de esforço na engenharia de *software* foram coletados. Os passos com a estratégia de pesquisa são ilustrados na Figura 1.

Figura 1 – Metodologia da pesquisa.



Fonte: Elaborado pela autor.

4.1 Revisão sistemática de literatura

Como o objetivo do trabalho é encontrar fatores que contribuem de maneira positiva ou negativamente, foi realizada uma revisão sistemática de literatura que consiste em pesquisar, em base de dados pré-estabelecidas, trabalhos primários relacionados que obtenham conceitos relacionados ao contexto do trabalho (KITCHENHAM; CHARTERS, 2007).

4.2 Planejamento da revisão sistemática

Para a realização da revisão sistemática de literatura, é estabelecido um protocolo para a realização da pesquisa. O protocolo segue um esquema de execução neste trabalho que

fornece detalhes do planejamento da revisão (KITCHENHAM; CHARTERS, 2007). Onde são estabelecidas a definição do objetivo, perguntas de pesquisa, *string* de busca, bases de dados, critérios de seleção, formulário de extração de dados, procedimento para seleção e ameaças à validade. No capítulo 5, esses processos foram descritos com mais detalhes.

4.3 Realização da revisão sistemática

A partir do protocolo de revisão, a condução deste trabalho foi realizada com o objetivo de obter estudos que possuam fatores de impacto na estimativa de esforço em todo o processo de desenvolvimento de *software*. Esses estudos firmaram a fonte de dados para a obtenção dos fatores que influenciam a estimativa de esforço. Após a extração de dados, foi realizada a comunicação da revisão, apresentando as respostas a cada pergunta de pesquisa para que pudessem ser avaliadas.

4.4 Responder as perguntas de pesquisa

Considerando o objetivo desta revisão, foi possível responder a seguinte pergunta de pesquisa principal:

(QPP). Quais são os fatores de impacto na estimativa de esforço?

Existe a necessidade de especificar perguntas secundárias para entender o contexto do trabalho. As perguntas secundárias que foram respondidas são:

(QPS1). Quais são os *stakeholders* envolvidos?

(QPS2). Quais os métodos/técnicas utilizadas nos estudos?

(QPS3). Quais são os problemas de estimativa apontados no estudo?

Como resultado, foi demonstrado os fatores alcançados na revisão (positivo ou negativo) e sua definição. E também, foram respondidas as perguntas de pesquisa.

5 PROTOCOLO DA REVISÃO SISTEMÁTICA

O protocolo de revisão sistemática da literatura estabeleceu procedimentos para que pudesse ser viável realizar a revisão. A definição do protocolo que foi utilizado nesta pesquisa seguiu as *guidelines* necessárias de (KITCHENHAM; CHARTERS, 2007).

Sendo assim, são muitas as razões para realizar uma revisão de literatura. Em primeiro lugar, pode-se citar a importância de resumir evidências relacionadas a um determinado assunto ou conceito. Outra razão é identificar novas possibilidades na pesquisa, a fim de sugerir novas áreas de investigação. De acordo com Kitchenham e Charters (2007), para se estabelecer um protocolo bem definido é preciso seguir alguns passos, estes com seus critérios bem definidos:

- Obter uma justificativa de pesquisa consistente;
- Definir questões de pesquisa que a revisão pretende responder;
- Estabelecer estratégias que serão utilizadas para pesquisar estudos primários, incluindo termos de pesquisa e recursos a serem utilizados;
- Definir uma *string* de busca eficiente para que seja feito pesquisa automática nas bases escolhidas;
- Estabelecer as bases de dados que serão utilizadas na revisão;
- Definir os critérios de seleção, para determinar quais estudos serão incluídos e excluídos da revisão sistemática;
- Criar um formulário de extração de dados;
- Apresentar soluções às ameaças de validade do trabalho;

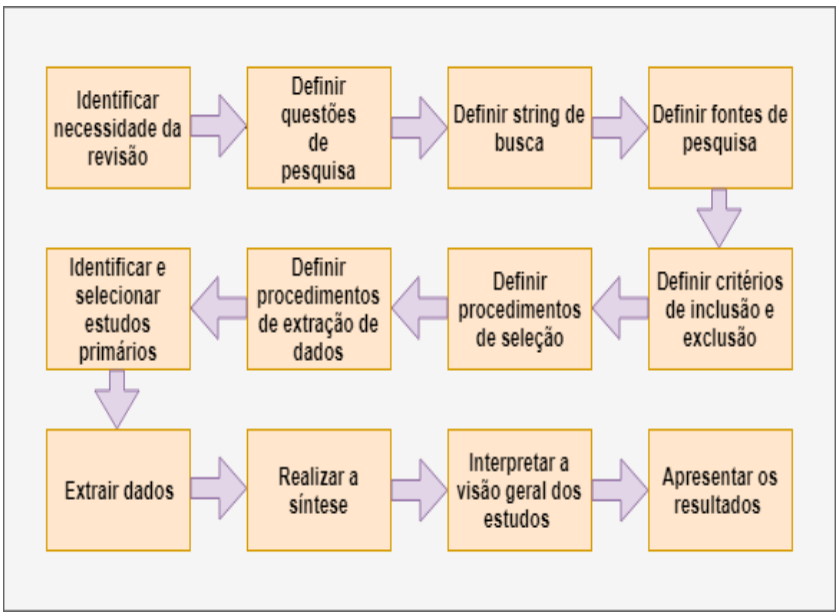
A vantagem de realizar uma revisão sistemática de literatura é, que a partir de uma metodologia bem definida ocorrem menos chances de obter resultados tendenciosos.

A partir de um protocolo bem definido, dar-se início a revisão com o objetivo de encontrar o maior número de trabalhos primários relacionados as perguntas de pesquisa estabelecendo uma *string* de busca que realize as pesquisas automáticas de maneira imparcial. Com isso, um resumo dos dados é elaborado e, por fim, um relatório principal é feito de acordo com os dados extraídos da pesquisa. Todos os processos necessários para a condução da revisão sistemática são ilustrados na Figura 2.

O protocolo, que foi definido utilizando a ferramenta *Parsifal* [1], é detalhado nas próximas seções.

[1] <https://parsif.al/>

Figura 2 – Processos da revisão sistemática.



Fonte: Adaptado de (KITCHENHAM; CHARTERS, 2007)

5.1 Definição do objetivo

O objetivo desta revisão é identificar estudos que descrevam fatores que influenciam a estimativa de esforço em todo o processo de desenvolvimento de software.

Considerando o objetivo dessa revisão, foram respondidas as perguntas de pesquisa apresentadas na Tabela 2.

Tabela 2 – Perguntas de pesquisa.

QPP. Quais são os fatores de impacto na estimativa de esforço?
QP1. Quais são os stakeholders envolvidos?
QP2. Qual o método/técnica utilizada no estudo?
QP3. Quais são os problemas de estimativa apontados no estudo?

Fonte: Elaborado pela autor.

5.2 *String* de busca

A *string* de busca foi definida considerando os principais termos dos conceitos sob investigação. As palavras-chave são listadas na Tabela 3.

Testes com pesquisas foram realizadas a fim de refinar a *string* de busca. Foram excluídas palavras-chave cuja inclusão não retornou estudos adicionais nas pesquisas automáticas.

Com isso, a seguinte *string* foi utilizada para pesquisar as palavras chave, título, resumo e texto completo das publicações:

("project management" OR "project manager") AND ("software engineering") AND ("software development" OR "software projects") AND ("metrics" OR "technical" OR "method" OR "model") AND ("duration" OR "cost" OR "time" OR "effort") AND ("software sizing" OR "size estimation") AND ("estimates" OR "estimative" OR "estimation").

Tabela 3 – Palavras-chave.

Palavras-chave
Effort estimation
Impact
Factors
Size estimation

Fonte: Elaborado pela autor.

5.3 Base de dados

As bases de dados que foram utilizadas para a seleção dos artigos são descritas na Tabela 12.

Tabela 4 – Bases utilizadas na pesquisa.

Nome	**URL**
ACM Digital Library	https://dl.acm.org
IEEE Digital Library	https://ieeexplore.ieee.org
Scopus	https://www.scopus.com
Springer Link	https://link.springer.com

Fonte: Elaborado pela autor.

5.4 Critérios de seleção

Os critérios de seleção foram utilizados para determinar se um trabalho deve ou não ser incluído na revisão. Esses critérios estão descritos na Tabela 5.

Tabela 5 – Critérios de inclusão e exclusão.

Critérios de inclusão	**Critérios de exclusão**
Estudos primários.	Estudos duplicados.
Estudos publicados em qualquer ano até maio de 2019.	Estudos incompletos.
Estudos que abordam nos objetivos estimativa de esforço.	Estudos secundários.
Estudos que relacionam estimativa e esforço.	Documento redundante de mesma autoria.
Estudos que possuam relatos de experiência em estimativa de esforço.	Estudos claramente irrelevantes para a pesquisa, levando em consideração as questões de pesquisa.
Estudos que possuam fatores de impacto na estimativa de esforço(critério de inclusão principal).	Estudos cujo foco não relaciona-se a estimativa de esforço.
	Estudos cujo texto completo não esteja disponível.
	Literatura cinza (teses, dissertações, monografias, etc).
	Estudos não escritos em inglês.

Fonte: Elaborado pela autor.

5.5 Formulário de extração de dados

Para guardar todas as informações necessárias para responder as perguntas da pesquisa foi preparado o formulário de extração de dados apresentado na Tabela 6.

Tabela 6 – Formulário de extração dos dados.

Dado	**Descrição**	**Pergunta de Pesquisa**
Autores, ano, título		Visão Geral dos estudos
Origem do trabalho	IEEE, ACM, Springer, Scopus, Science Direct.	Visão Geral dos estudos
Tipo do trabalho	Jornal, conferência, simpósio, workshop, capítulo de livro.	Visão Geral dos estudos
Contexto de aplicação	Indústria, Academia, Ambos.	Visão Geral dos estudos
Tipo de Pesquisa (baseado em (WIERINGA, 2010))	Pesquisa de validação, pesquisa de avaliação, proposta de solução, Artigos filosóficos, Artigos de experiência.	Visão Geral dos estudos
Fator de estimativa		QPP
Stakeholders envolvidos		QP1
Método/Técnica de estimativa		QP2
Problemas de estimativa		QP3

Fonte: Elaborado pela autor.

5.6 Procedimentos para seleção de estudos

O procedimento de seleção foi composto por três etapas. Na primeira etapa, a *string* de busca definida e apresentada na seção 5.2 é utilizada nas bases de dados descritas na seção 5.3. Todos os estudos retornados na pesquisa são armazenados na ferramenta Parsifal [2].

A segunda etapa consistirá na exclusão de artigos duplicados e, a partir da leitura do título e *abstract* dos estudos primários, usando os critérios de selção descritos na seção 5.4, selecionar ou rejeitar os estudos que avançaram para a próxima etapa. Para mitigar ameaças à validade interna descrita com detalhes na seção 5.7, qualquer dúvida no julgamento, de acordo com os critérios de seleção, foi adotado o critério de dar continuidade com o estudo para a próxima etapa. Mitigando do contexto, a subjetividade, que acaba sendo inerente ao julgamento humano.

A terceira e última etapa consistiu na realização, de fato, da revisão sistemática de literatura que resume-se na leitura completa dos estudos restantes para classificar os estudos relevantes e os fatores que influenciam a estimativa de esfoço possam ser exemplificados. Por fim, foram respondidas as questões de pesquisa secundárias definidas na seção 5.1 para facilitar o entendimento a respeito do contexto deste trabalho.

5.7 Ameaças à validade

É fundamental considerar as ameaças à validade deste trabalho. Desde o planejamento do trabalho, é preciso ponderar a validade dos resultados obtidos para que estes possam ser considerados válidos para o contexto analisado. Portanto, a classificação de ameaças descritas por (WOHLIN *et al.*, 2012) foi utilizada para discutir as ameaças deste trabalho. Esta classificação é dividida em quatro tipos de ameaças:

• Ameaças à validade de construção: Onde pode ocorrer uma má definição da base teórica ou da definição do processo de experimentação. Desse modo, pode ocorrer generalização do conceito ou teoria na execução do estudo. Para minimizar que palavras-chave específicas possam limitar a abrangência das pesquisas, foi utilizado sinônimos para essas palavras.

• Ameaças à validade interna: A validade interna, para Travassos *et al.* (2002), “define se o relacionamento observado entre o tratamento e o resultado é causal, e não é resultado da influência de outro fator – não controlado ou medido". Pode-se ocorrer decisões tendenciosas

[2] <https://parsif.al/>

ou subjetivas onde não se pode confiar nos resultados primários obtidos por não fornecerem uma descrição clara e objetiva. Tornando a aplicação dos critérios de inclusão e exclusão mais difíceis. Em busca de corrigir o problema de extração e seleção, para qualquer dúvida no julgamento realizado, de acordo com os critérios estabelecidos, o estudo seguirá para a próxima etapa. Além disso, o processo de seleção foi realizado de forma colaborativa pelo autor e por um colaborador a fim de minimizar conflitos de decisões e chegar a um nível de credibilidade aceitável. Dessa forma, mitigar influências negativas que possam ser ocasionadas pelo viés pessoal na compreensão do estudo.

• Ameaças à validade externa: Está relacionada com o grau que os estudos primários serão representativos para o conteúdo da revisão. As ameaças mais comuns são quando os estudos alcançados não conseguem abranger o conteúdo desejado. Isso pode acontecer quando a literatura identificada não tem validade externamente. Esse problema será resolvido com base no critério de exclusão, da pesquisa, que elimina os estudos resultante de literatura cinza. Além disso, para mitigar influências externas ao trabalho, o protocolo de pesquisa foi debatido e testado com o consenso do autor, colaborador e da orientadora, para então ser validado.

• Ameaças à validade de conclusão: Se trata de obter uma relação coerente entre o tratamento e o resultado do estudo. A metodologia de (KITCHENHAM; CHARTERS, 2007) estabelece a necessidade de todos os estudos primários relevantes possam ser retornados e avaliados no estudo. As soluções para amenizar essa ameaça foram: realização de testes iterativos constantemente utilizando diversas expressões e palavras sinônimas para a construção desta revisão sistemática.

Para minimizar ameaças à validade no processo de seleção dos estudos calculou-se o índice de concordância na etapa de classificação entre os trabalhos rejeitados pelo autor e o colaborador. A partir dos resultados, a menor porcentagem de concordânica, realizada a partir dos trabalhos rejeitados de cada base, foi 84,4% e estão ilustrados na Tabela 12. A fórmula utilizada no cálculo da porcentagem de concordância é demonstrada a seguir:

$$\frac{concordância}{concordância + discordância} \cdot 100 \qquad (1)$$

Tabela 7 – Índice de concordância.

	Autor Rejeitou	Colaborador Rejeitou	Concordâcia
ACM	4	4	100%
SCOPUS	375	364	84,4%
SPRINGER	327	316	96,0%

Fonte: Elaborado pela autor.

6 RESULTADOS

O processo de seleção dos estudos, apresentado na Figura 3, foi dividido em algumas etapas essenciais. Dos 859 resultados de pesquisa, 789 eram únicos. Posteriormente, foi realizada a leitura do título e abstract dos artigos, foram excluídos 725 estudos, baseados nos critérios de exclusão. Em seguida, 64 trabalhos permaneceram no processo de seleção para a leitura de texto completo, obteve-se 53 artigos relevantes para extração de dados.

Figura 3 – Distribuição dos trabalhos na revisão.

ACM	Scopus	Springer	TOTAL
Números de Trabalhos			
7	454	398	859
Duplicados			
0	20	50	70
Leitura de Título e Abstract			
7	434	348	789
Leitura Completa			
3	37	24	64
Classificados			
3	31	19	53

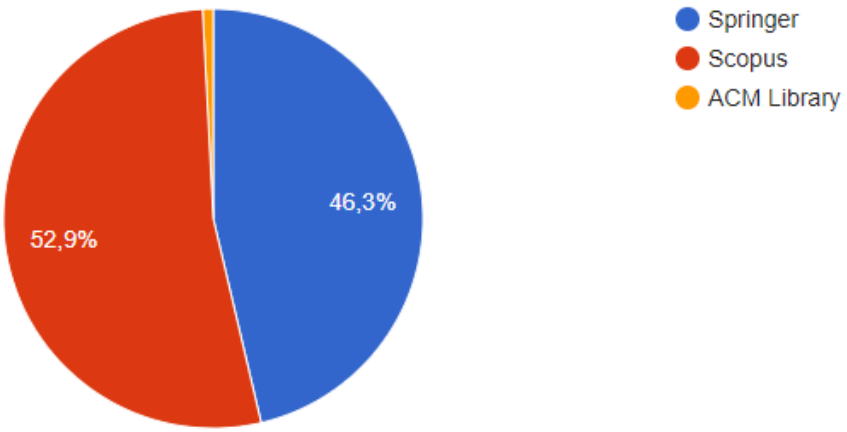
Fonte: Elaborado pela autor.

Os dados foram extraídos dos 53 estudos que satisfizeram os critérios de inclusão descritos na Tablela 5 e conforme o fórmulario de extração descrito na Tabela 6. Uma visão geral das características gerais dos estudos é demonstrada. Para que, por fim, os fatores de impacto na estimativa de esforço, que é o principal objetivo desse estudo, e as análises das perguntas de pesquisa sejam apresentadas.

6.1 Visão geral dos estudos

A *string* definida na seção 5.2 inserida nas bases de dados utilizas no estudo retornaram 859 trabalhos distibuídos em ACM: 7, Scopus: 454, Springer: 398. A disposição dos mesmos foi ilustrada na Figura 4.

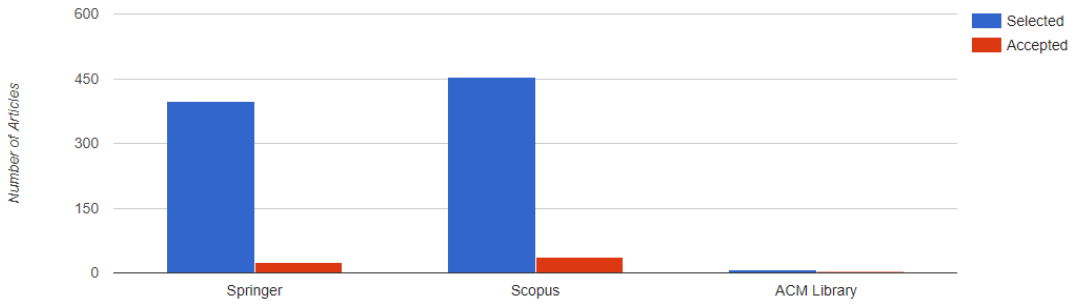
Figura 4 – Artigos por base



Fonte: Ferramenta Parsif.al.

Os trabalhos classificados por cada base totalizaram 53 estudos sendo distribuídos da seguinte forma: ACM: 3, Scopus: 31, Springer: 19, ver Figura 5

Figura 5 – Artigos aceitos por base

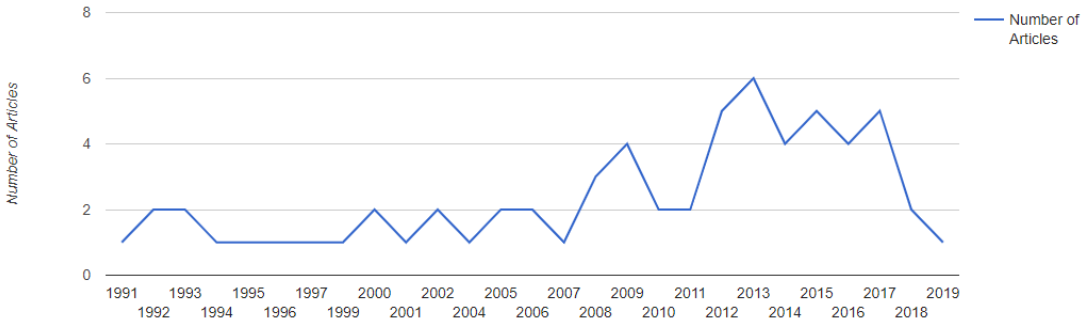


Fonte: Ferramenta Parsif.al.

A partir do fato de que a complexidade do desenvolvimento de *software* está em uma crescente cada vez maior, torna-se crítico identificar os problemas que influenciam a estimativa de esforço. Constatou-se notável aumento em pesquisas tanto no âmbito da indústria, como na academia, preocupadas em especificar fatores que impactam a estimativa de esforço ao longo dos anos. A extração de dados mostrou que a pesquisa por fatores que impactam a estimativa

de esforço é relavente diante da grande necessidade de estimativas mais precisas para projetos de desenvolvimento de *software*. A Figura 6 representa os anos de publicação dos trabalhos avaliados no estudo.

Figura 6 – Ano de publicação dos trabalhos



Fonte: Ferramenta Parsif.al.

6.2 (QPP). Quais são os fatores de impacto na estimativa de esforço?

A partir da análise dos trabalhos classificados, foram identificados 29 fatores que afetam direta ou indiretamente a estimativa de esforço em projetos de *software*. Os trabalhos referenciados são os que possuem a definição do fator associado. Foi possível identificar fatores de naturezas diversas que afetam tanto estimativas de cronograma, como estimativas de custo. Os fatores que influenciam a estimativa de esforço estão descritos e definidos a seguir:

– Disponibilidade de dados históricos: Os dados são informações disponíveis pela empresa relacionadas ao esforço desempenhado em projetos passados (RAMMAGE *et al.*, 2010) (ZAPATA; CHAUDRON, 2012) (MATOS *et al.*, 2013) (ABRAN, 2015) (XU, 2006).

– Uso de sistema de gerenciamento: O uso de sistema de gerenciamento ajuda na recuperação de um documento de projeto que foi realizado anteriormente (MOHAPATRA; GUPTA, 2011).

– Complexidade do *software*: O melhor preditor da estimativa de esforço pode ser um problema de complexidade (RAMMAGE *et al.*, 2010). A complexidade do projeto estão diretamente associadas as regras de negócio, integração com outros sistemas, banco de dados, níveis de segurança, dentre outros (MORGENSHTERN *et al.*, 2007) (KOSLOSKI; OLIVEIRA, 2005) (TRENDOWICZ; JEFFERY, 2014).

– Tamanho do *software*: Um projeto de *software* de tamanho pequeno ou médio

tem menos chances de erros de estimativa do que um projeto de tamanho maior (MCCONNELL, 2006) (TSUNODA *et al.*, 2013) (KOSLOSKI; OLIVEIRA, 2005). Gerentes de projetos precisam ter estimativas de tamanho antecipadas para estimar o esforço de *software* desenvolvido (SALMANOĞLU *et al.*, 2014).

– Natureza dos projetos: A natureza do projeto pode estar associada ao ambiente ou contexto que o *software* está inserido. Comparar projetos cujos atributos subjacentes são diferentes pode ocasionar estimativas tendenciosas (KHATIBI; BARDSIRI, 2015) (BAJWA; GENCEL, 2009).

– Envolvimento do cliente: A interação com o cliente é necessária desde a fase de análise de requisitos até o final do projeto (MOHAPATRA; GUPTA, 2011). Para alcaçar o potencial efeito positivo da pressão de cronograma é preciso cooperação entre clientes e equipes de desenvolvimento (NAN; HARTER, 2009) (TRENDOWICZ; JEFFERY, 2014) (MATOS *et al.*, 2013).

– Experiência na tecnologia: A experiência passada da equipe do projeto na mesma tecnologia do projeto atual diminuirá o tempo necessário para execução das atividades e, portanto, aumenta a produtividade. A acurácia da estimativa é influenciada diretamente pela familirialidade da equipe com a tecnlogia utilizada (MOHAPATRA; GUPTA, 2011) (KOSLOSKI; OLIVEIRA, 2005) (TRENDOWICZ; JEFFERY, 2014) (LAKHANPAL, 1993).

– Disponibilidade de recursos: Erro comum de projetos é estimar o desenvolvimento considerando um certo número de pessoas disponíveis, onde muitas vezes isso acaba não acontecendo. Outro problema é relacionado a velocidade do hardware e o desempenho do sistema que podem afetar o tempo gasto pelo projeto (MOHAPATRA; GUPTA, 2011).

– Otimismo do estimador: Os estimadores geralmente subestimam as tarefas de desenvolvimento ou tem o pensamento equivocado que tudo irá correr bem (ZAPATA; CHAUDRON, 2013) (ERASMUS; DANEVA, 2013) (MOHAGHEGHI *et al.*, 2005).

– Experiência em projetos de *software*: A experiência, das equipes de desenvolvimento, em projetos passados está ligada ao erro de estimativa (ZAPATA; CHAUDRON, 2013) (KOSLOSKI; OLIVEIRA, 2005) (TRENDOWICZ; JEFFERY, 2014) (LAKHANPAL, 1993) (BHARDWAJ; RANA, 2015).

– Envolvimento do estimador: Esse fator refere-se ao grau que o estimador está envolvido com a implementação. É importante que os responsáveis pelas estimativas sejam atribuídos ao projeto (ZAPATA; CHAUDRON, 2013).

– Tamanho da equipe: O tamanho da equipe afeta as decisões do cronograma, que também são conhecidas como um fator importante no sucesso do projeto (VERNER *et al.*, 2007) (KAKIMOTO *et al.*, 2017) (BAJWA; GENCEL, 2009) (PENDHARKAR; RODGER, 2007). Além disso, o tamanho da equipe ideal é importante sobre uma eventual partição de projetos em subprojetos menores (RODRÍGUEZ *et al.*, 2012).

– Pressão externa para reduções de estimativa ou comportamento político de estimativa: A estimativa orçamentária pode sofrer impacto negativo, por exemplo, de clientes que usam de sua autoridade para não aceitarem estimativas válidas (NAN; HARTER, 2009). Geralmente, ocorre alinhamento de objetivos ligados a pressões de aumentar ou diminuir as estimativas (LEDERER; PRASAD, 1991).

– Clareza dos objetivos: Isso ocorre quando os objetivos e necessidades são entendidos por todas as partes envolvidas em estimar o projeto (HILL *et al.*, 2000) (DIEV, 2006) (TSUNODA *et al.*, 2013).

– Requisitos voláteis: Existem duas fontes principais de volatilidade dos requisitos: a probabilidade de uma mudança nos requisitos no futuro e a imprecisão dos requistos (SAVOLAINEN; KUUSELA, 2001) (LAVAZZA; VALETTO, 2000) (LEDERER; PRASAD, 1993) (BARRY *et al.*, 2002) (MATOS *et al.*, 2013) (FELLIR *et al.*, 2015).

– Considerar o *design* do sistema: O *design* especifica como o sistema deve ser feito, facilitando o entendimento do cliente e dos estimadores (MCCONNELL, 2006) (DIEV, 2006) (ANDA *et al.*, 2001).

– Detalhamento antecipado dos requisitos: É preciso identificar os requisitos claramente antes de estimiar o *software*. Muitas vezes a veracidade desses dados são omissos, inconsistentes, duplicados ou supérfolos (TRUDEL; ABRAN, 2009) (UNGAN *et al.*, 2009).

– Confiança do cliente: Se trata da confiança que o cliente possui nas estimativas do estimador. Essa fator tem influência direta no processo de estimativa (MOSES, 2002).

– Fatores ambientais: É importante considerar os fatores ambientais dentro da estimativa. Fatores ambientais exigem uma avaliação da competência da equipe de projeto. Problemas como: motivação, habilidades, conhecimentos, recursos humanos existentes, distribuição geográfica de instalações, canais de comunicação, base de dados etc, precisam ser considerados na hora da estimativa (ANDA *et al.*, 2001) (BARRY *et al.*, 2002).

– Gerenciamento de riscos: A falha no gerenciamento de riscos pode ocasionar impactos na acurácia da estimativa. Esse gerenciamento pode ser definido com uma tentativa

de formalizar correlatos de sucesso de desenvolvimento orientados a riscos em um conjunto de princípios e práticas prontamente aplicáveis (ROPPONEN; LYYTINEN, 1997).

– Uso de método de estimativa: utilizar um método de estimativa que esteja alinhado com os ojetivos definidos e os recursos disponíveis (TRENDOWICZ *et al.*, 2008).

– Inexperiência do estimador: A inexperiência do estimador pode contribuir diretamente na imprecisão de estimativas (WALKERDEN; JEFFERY, 1999) (MOHAGHEGHI *et al.*, 2005).

– Experiência na área de negócios do sistema: A falta de familiaridade contribui para o erro de estimativa (KOSLOSKI; OLIVEIRA, 2005).

– Uso de metodologia de desenvolvimento: Metodologias de *software* são usadas em projetos de *software* para gerenciar atividades, artefatos, funçoões e disciplinas. Pesquisadores acreditam que o uso de metodologias podem permitir aos gerentes do projeto lidar com a natuureza incerta e complexa dos projetos (KHATIBI *et al.*, 2012).

– Subestimar custos: Previsões de custo precisas para *software* são importantes. Os excessos de custos resultantes de estimativas imprecisas são frequentes (LEDERER; PRASAD, 1992).

– Colaboração da equipe: A colaboração é apontada como fator que impactam a acurácia das estimativas. Geralmente, são examinados os efeitos da colaboração da equipe na coesão do grupo, tarefa do redesenho, estabelecimento de metas, restrições situacionais, taxa de rotatividade voluntária, comportamento social e produtividade. A comunicação interna da equipe resultam no cumprimento das estimativas iniciais (LAKHANPAL, 1993).

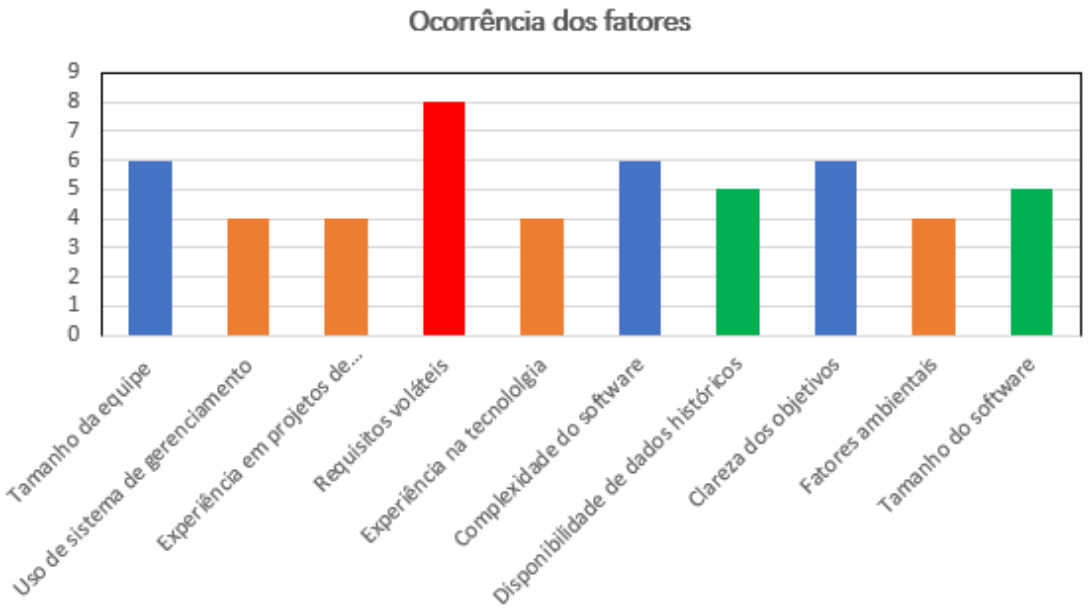
– Detalhamento antecipado do planejamento: Os principais fatores que contribuem para melhorar as estimativas é o planejamento em nível detalhado (LEDERER; PRASAD, 1992).

– Imprevisibilidade das tarefas: É preciso levar em consideração a imprevisibilidade das tarefas na hora de estimar (BVANSS; RAMAIAH, 2017).

– Reserva de planejamento (buffer): As reservas são uma precaução no planejamento do projeto para mitigar os riscos de cronograma e custo (XAVIER *et al.*, 2005).

É possível perceber que muitos trabalhos na literatura citam a volatilidade de requisitos, complexidade do *software*, clareza dos objetivos e o tamanho da equipe como fatores de impacto nas estimativas. A Figura 7 ilustra a ocorrência dos fatores que apareceram 4 (quatro) vezes ou mais nos estudos classificados.

Figura 7 – Ocorrência dos fatores

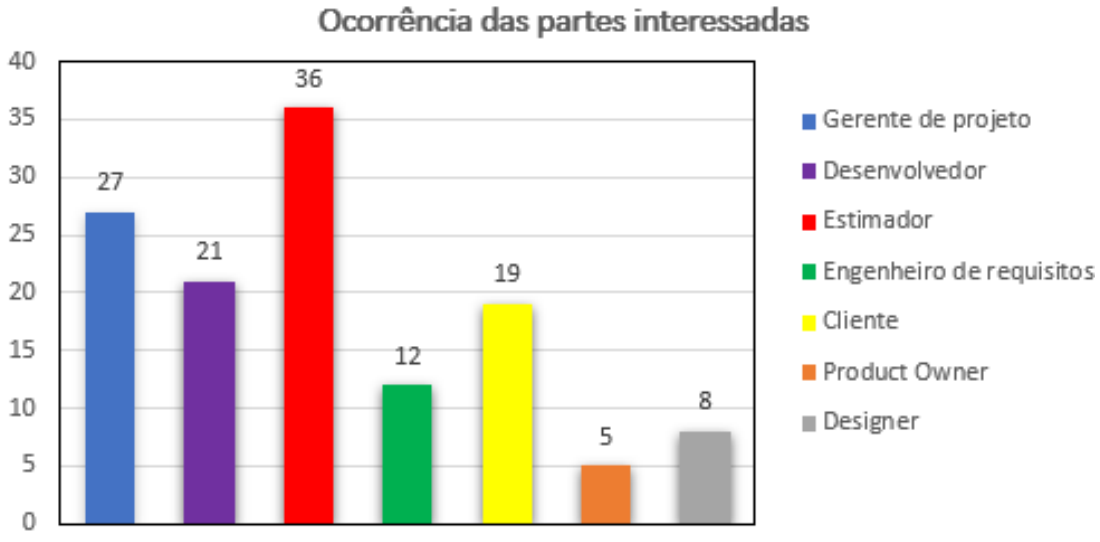


Fonte: Elaborado pelo autor.

6.3 (QPS1). Quais são os *stakeholders* envolvidos?

A Figura 8 ilustra os *stakeholders* identificados presentes nos estudos analisados durante a revisão. Observou-se uma frequência maior de "Estimadores", "Gerente de projetos", "Desenvolvedores"e "Cliente". O objetivo foi verificar as partes afetadas direta ou indireta, positiva ou negativamente dentro de cada estudo revisado.

Figura 8 – Stakeholders



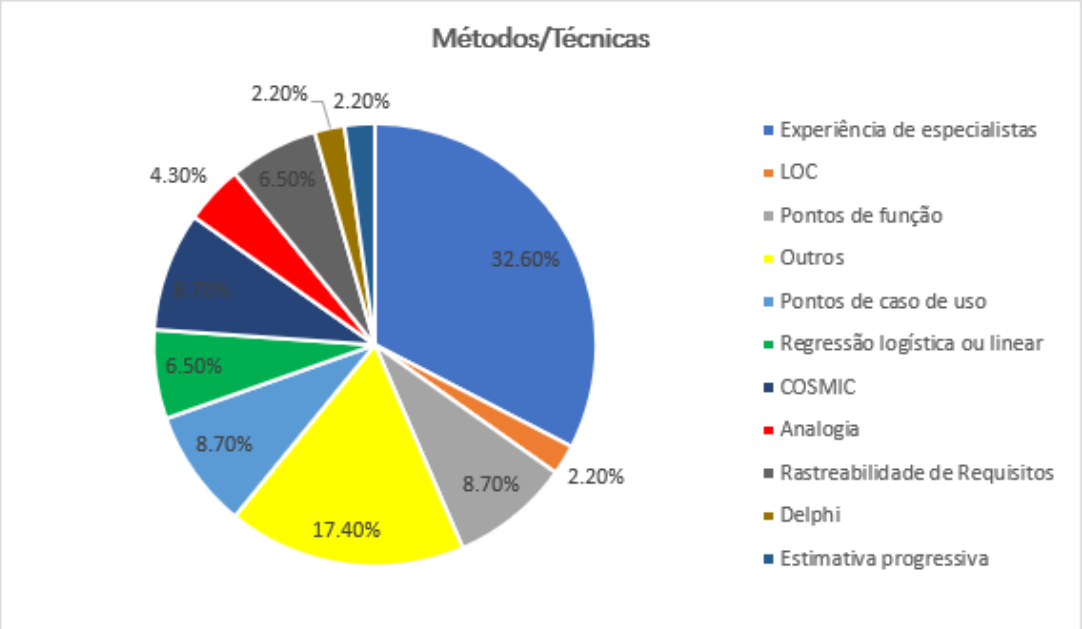
Fonte: Elaborado pelo autor.

6.4 (QPS2). Quais os métodos/técnicas utilizadas nos estudos?

A Figura 9 ilustra os métodos e técnicas utilizadas nos estudos relevantes desta revisão. O propósito da questão de pesquisa não é demonstrar os métodos/técnicas mais utilizadas para estimativas de esforço em projetos de *software*, pois essa informação está fora do escopo deste trabalho. O gráfico demonstra os métodos/técnicas que são abordados nos trabalhos que possuem fatores de estimativa provenientes dos estudos relevantes desta revisão.

Na maior parte dos estudos revisados, a pesquisa não aplicava um método ou técnica. Os estudos abordados identificam e demonstram os fatores através de estudos explanatórios ou comparativos com uma grande base de projetos finalizados dentro ou fora das estimativas previstas. Por fim, é possível verificar na Figura 9 que o método mais utilizado nos estudos que possuem fatores de impacto na estimativa de esforço foi o de "Experiência de especialistas".

Figura 9 – Métodos e técnicas abordados nos estudos



Fonte: Elaborado pelo autor.

6.5 (QPS3). Quais são os problemas de estimativas apontados nos estudos?

A partir da análise dos trabalhos e após identificação dos fatores, foram demonstrados os problemas de estimativa que os estudos relatavam em suas pesquisas. O objetivo foi apresentar a maior quantidade de problemas reais que acontecem durante as estimativas. Foi possível

identificar grande diversidade de problemas relacionados aos fatores encontrados, principalmente relacionados a volatilidade de requisitos e inexperiência do estimador. Para facilitar a visualização dos problemas de estimativa recomenda-se verificiar o Apêndice A. Os problemas de estimativa que envolvem os fatores que afetam a estimativa de esforço estão descritos a seguir:

– Disponibilidade de dados históricos: Técnicas de estimativa orientada a dados requerem dados históricos de anos para serem confiáveis. Isso significa que o esforço de anos pode ser necessário antes que a organização perceba benefícios práticos de estimativa (RAMMAGE *et al.*, 2010).

– Uso de sistema de gerenciamento: Na pesquisa de (ZAPATA; CHAUDRON, 2013) para saber quão útil é o *software* atual da empresa, que guarda dados de estimativa passadas, para realizar estimativas futuras, conclui-se que os dados da amostra não são úteis para prever o trabalho do projeto. Porém, para prever o cronograma e o custo, o resultado é mais útil e pode ser usado como uma estimativa aproximada. Sistemas de gerenciamento devem seguir padrões como ISO, CMMI etc.

Estes sistemas são muito importantes para a gestão do conhecimento que resultam em melhores estimativas de projeto (MOHAPATRA; GUPTA, 2011). Outro autor analisa a precisão de diferentes dimensões: cronograma, orçamento, e esforço e suas relações. Foi determinado também, quão útil as estimativas atuais para prever estimativas futuras (ZAPATA; CHAUDRON, 2012).

– Complexidade e tamanho do *software*: A complexidade do projeto é difícil de quantificar ou medir diretamente. Na prática, torna-se mais simples medir várias outras variáveis do projeto que podem correlacionar-se com a complexidade do problema. Essas variáveis são frequentemente chamadas de "*metrics*"ou "*called proxy*". A maioria desses preditores, que podem incluir tipos de requisitos, pontos de função, número de arquivos, complexidade cíclica ou números de casos de teste, fornecem quantificação do tamanho do problema. Linhas de cógido fonte (SLOC) são frequentemente utilizadas, mas por si só não podem ser um preditor porque o tamanho do *software* não é conhecido até que o mesmo seja finalizado (RAMMAGE *et al.*, 2010).

Outro estudo aborda que é preciso estimar o tamanho do *software* antecipadamente, ou seja, antes do ciclo de desenvolvimento. Entradas para modelos de estimativa de tamanho são geralemente artefatos de *software* que são produzidos posteriormente do ciclo de vida de desenvolvimento do *software* (SALMANOĞLU *et al.*, 2014).

– Natureza dos projetos: Comparar estimativas de projetos que estão inseridos em contextos diferentes pode ser um problema. O peso dos atributos podem ser diferentes de um grupo para o outro de acordo com a natureza dos projetos. Melhorar os desvios de comparações não confiáveis podem melhorar consideravelmente a precisão e a confiabilidade das estimativas (KHATIBI; BARDSIRI, 2015) (MATOS *et al.*, 2013).

– Experiência na tecnologia: A produtividade do tarefa desempenhada depende da experiência de quem está realizando a atividade. O tempo que um desenvolvedor leva para executar uma atividade varia com o tempo de outro desenvolvedor para realizar a mesma atividade de acordo com sua experiência na tecnologia do projeto (MOHAPATRA; GUPTA, 2011).

– Envolvimento do cliente: Quanto mais o cliente apoiar nas atividades, mais fácil será pra a equipe do projeto avançar na sua execução. É comum erros de estimativa ocasionados pela falta de interação entre o cliente, estimador e equipe de desenvolvimento de *software* (MOHAPATRA; GUPTA, 2011). Uma exploração adicional dos projetos incluídos na pesquisa de (ZAPATA; CHAUDRON, 2013) revelou que o baixo envolvimento de clientes no desenvolvimento de *software* podem acabar com os benefícios da pressão do cronograma.

– Disponibilidade de recursos: Problemas surgem quando as estimativas de *software* consideram que o número de pessoas, computadores dentre outros recursos relacionados na estimativa serão de qualidade ou estarão sempre disponíveis, sendo que muitas vezes isso não acontece. A indisponibilidade de pessoas e desses recursos ou mau funcioanamento deles podem causar atrasos significativos no desenvolvimento (MOHAPATRA; GUPTA, 2011).

– Otimismo do estimador: O otimismo do estimador precisa ser considerado, pois causa falhas nas estimativas. Entretando, esse otimismo pode ser mitigado por estimadores experientes, conhecimento do domínio comercial ou conhecimento técnico (ZAPATA; CHAUDRON, 2013) (ERASMUS; DANEVA, 2013).

– Experiência em projetos de *software*: A experiência da equipe associa-se diretamente com acurácia das estimativas. Esse fator é resultante de imprecisões e ocasiona atrasos no cronograma do projeto (ZAPATA; CHAUDRON, 2013).

– Envolvimento do estimador: O não envolvimento do estimador com o projeto pode causar imprecisão de estimativas. Na pesquisa realizada 14% dos entrevistados responderam que participaram entre 50% e 75% dos projetos que estimiram, e 59% responderam entre 75% e 100% (ZAPATA; CHAUDRON, 2013).

– Tamanho da equipe: Se o tamanho ideal da equipe puder ser encontrado, a decom-

posição de projetos em pedaços menores poderia se tornar uma prática chave no gerenciamento de distribuição de equipes por projeto. Esse tamanho ideal pode ocasionar melhorias na produtividade e, consequentemente, nas estimativas (RODRÍGUEZ *et al.*, 2012). Outro estudo assumem que o tamanho máximo da equipe incluem erros quando o esforço é estimado. O autor conclui que o uso do tamanho máximo da equipe como variável idependente foi satisfatório quando a taxa de erro não é muito grande (igaul ou inferior a 50%). Porém quando a taxa é igual ou superior a 100%, a precisão da estimativa é pior na maioria dos casos (KAKIMOTO *et al.*, 2017).

– Pressão externa para reduções de estimativas ou comportamento político de estimativa: Problemas de estimativa surgem por causa de orçamentos apertados, com isso o responsável pelo gerenciamento do projeto pode optar por economizar tempo e mão de obra. O estudo indica que o tempo de ciclo reduzido sob pressão orçamentária não é compensado pelo aumento de trabalho. Por fim, os gerentes podem usar dados históricos para estimar a faixa benéfica de pressão orçamentária e com isso realizar o gerenciamento do *software* com mais eficácia (NAN; HARTER, 2009). Problemas surgem nas estimativas de custo quando os mesmos são influenciados por comportamentos políticos de estimativa (LEDERER; PRASAD, 1991).

– Clareza dos objetivos: O estudo indica que estimadores cometeram, no estudo de caso avaliado, maiores erros médios subestimados em tarefas que não foram claramente definidas. Como resultado gerentes de projeto tendiam a superestimar cerca de 60% das tarefas, mas se subestimavam, os erros de estimativa eram maiores (HILL *et al.*, 2000). Outro autor indica a clareza das metas como fator que têm forte ligação com a superação do custo de desenvolvimento de *software* (TSUNODA *et al.*, 2013).

– Requisitos voláteis: Usuários raramente tem suas necessidades bem definidas, principalmente no início de desenvolvimento do projeto (MRENAK, 1990). O estudo (CAO, 2008) conclui que o *feedback* rápido das iterações anteriores e a transparência do processo de desenvolvimento em metodologias ágeis permite o ajuste contínuo da estimativa durante o desenvolvimento do *software*. Outra abordagem é que o tamanho e a complexidade do código são estimados com base no tamanho e complexidade dos requisitos. Tudo isso com o intuito de melhorar a precisão de estimativas de custo (LAVAZZA; VALETTO, 2000). Segundo (FELLIR *et al.*, 2015), o mal entendimento sobre os requitos não funcionais e as relações entre os requistos funcionais podem afetar as estimativas ao longo do processo de desenvolvimento.

No estudo de (LEDERER; PRASAD, 1993), o grande problema especificado é

dado pelas solicitações frequentes de alteração pelo usuário. Pode ser muito difícil para os desenvolvedores e estimadores antecipar e ter controle destas alterações. O estudo conclui que o gerenciamento eficaz, em vez de uma estimativa precisa pode ser a chave para a conclusão do projeto dentro das estimativas de custos especificadas no início do projeto.

– Considerar o *design* do sistema: O objetivo do estudo é demonstrar que para fornecer estimativas com base em casos de uso é necessário levar em consideração o *design* do sistema (DIEV, 2006).

– Detalhamento antecipado dos requisitos: Durante as fases iniciais de um ciclo de vida de desenvolvimento de *software*, os requistos documentados são usados como insumo para o processo de estimativa, inclusive para medir o tamanho funcional do *software* a ser desenvolvido. A qualidade de um requisito documentado terá impacto na consistência dos resultados de medição, bem como na confiança dos resultados obtidos (TRUDEL; ABRAN, 2009). Problemas de medição defeituosa que acontecem quando o medidor não pôde aplicar as regras do método COSMIC. E de suposições incorretas, onde se inclui mais requisitos do que o especificado, baseado na experiência pessoal. Como, também, de interpretações diferentes onde o medidor alcança diferentes resultados de medição analisando o mesmo problema sob diferentes perspectivas são problemas que ocasionam discrepâncias e variações ma estimativa (UNGAN *et al.*, 2009).

– Confiança do cliente: O trabalho de (MOSES, 2002) propõe que, medindo as incertezas na estimativa de esforço, pode-se melhorar a confiança de duas maneiras. Em primeiro lugar, os estimadores podem ser ajudados a melhorar a consistência de suas estimativas e declarações podem ser feitas aos cientes sobre o provável esforço que um projeto realmente levará, além de demonstrar erros de estimativa para acima ou abaixo do estipulado.

– Fatores ambientais: O estudo apresenta alguns pontos fracos do método pontos de caso de uso. Um desses pontos fracos trata-se de que fatores ambientais estejam desatualizados (HUANCA; ORÉ, 2016). Outro estudo, que utiliza o método de pontos de caso de uso, cita a dificuldade de designar pesos a fatores ambientais. Acaba sendo necessário recorrer a projetos, dentro do mesmo contexto de aplicação, que já realizam essa atividade e chegaram a estimativas satisfatórias (ANDA *et al.*, 2001).

– Gerenciamento de riscos: O estudo investiga características e práticas de gerenciamento de riscos e procura fatores ambientais e de processo que se relacionam e melhoram o gerenciamento de riscos. Mitigando o impacto de estimativas de tamanho incorretas (ROPPO-

NEN; LYYTINEN, 1997).

– Uso de método de estimativa: Problemas ocorrem quando o método aplicado não está alinhado com os objetivos definidos ou os recursos disponíveis. O método de estimativa requer mais recursos do que existe atualmente na organização, por exemplo. (TRENDOWICZ *et al.*, 2008).

– Inexperiência do estimador: O estudo compara vários métodos de estimativa de esforço de *software* baseado em analogia entre si e, também com um modelo de regresssão linear simples. O autor propõe que estimadores sem experiência definam os ajustes de cada método (WALKERDEN; JEFFERY, 1999). Julgamentos inconsistentes e excesso de confiança são um problema. O estudo de (MOHAGHEGHI *et al.*, 2005) explica que o parecer de um especialista apenas, não deve ser usado com método de estimativa de custos. Aponta-se a necessidade de complemantar o julgamento com processos que respondem pelo viés observado (MOHAGHEGHI *et al.*, 2005).

– Experiência na área de negócios do sistema: O estudo de (KOSLOSKI; OLIVEIRA, 2005) propõe definir uma estrutura de características que afetam a produtividade do projeto de *software* e suas estimativas.

– Uso de metodologia de desenvolvimento: O estudo procura entender o efeito do uso de metodologias nas estimativas de desenvolvimento de *software*. O estudo conclui que apesar de apresentar vantagens, o uso de metodologias podem levar a estimativas negativas do esforço de desenvolvimento, porque não existem evidências de quanto o uso de metodologias de desenvolvimento podem influenciar na estimativa final do projeto (KHATIBI *et al.*, 2012).

– Subestimar custos: Subestimar custos nas suas estimativas destrem a credibilidade dos estimadores e desenvolvedores. Segundo o autor, quase dois terços de todos os grandes projetos superam suas estimativas (LEDERER; PRASAD, 1992).

– Colaboração da equipe: Problemas de desempenho do desenvolvimento ocasionados por falta de colaboração da equipe são apontados no estudo. Abordam, também, implicações importantes para as equipes de desenvolvimento que trabalham em projetos de *software* (LAKHANPAL, 1993).

– Detalhamento antecipado do planejamento: O estudo procura analisar a mitigação de erro de estimativa do *software* com base nas especificações do planejamento inicial do sistema (LEDERER; PRASAD, 1992).

– Imprevisibilidade das tarefas: A confiabilidade da estimativa leva ao sucesso

do projeto ou falha. A ideia do autor é trabalhar com a análise de pontos de função e incluir o conceito de agendamento de força de trabalho de uma maneira melhora na hora de tomar decisões de estimativa (BVANSS; RAMAIAH, 2017).

– Reserva de planejamento (buffer): As reservas são uma precaução no planejamento do projeto para mitigar os riscos de cronograma e custo (XAVIER *et al.*, 2005).

7 CONCLUSÃO E TRABALHOS FUTUROS

Atividades de gerenciamento de projetos, como cronograma, tempo e custos são importantes para evitar falhas no projeto. Com base no gerenciamento de projetos, a estimativa de esforço desempenham um papel fundamental. A estimativa do esforço é uma das fases mais críticas de todo o processo de desenvolvimento de *software* uma vez que uma grande parte dos problemas que ocasionam atrasos de cronograma e custos excedidos são derivados de estimativas fora da realidade ou situações imprevistas no esforço de desenvolvimento, ocasionados por fatores específicos.

Existem fatores que impactam, positivamente ou de maneira negativa, a estimativa de esforço em diferentes fases do planejamento e desenvolvimento do *software*. Dessa forma, a identificação dos fatores que influenciam essa estimativa é vital para obter melhores resultados no cronograma, tempo e custos dos projetos de *software*. Nesse contexto, proporcionar melhorias na tarefa de estimar um *software* é uma constante prioridade das empresas que prezam por repassar estimativas precisas e de boa qualidade ao seus clientes. Além disso, todas as fases seguintes de desenvolvimento são afetadass pelas estimativas inicias de um projeto.

Este trabalho teve como objetivo identificar fatores que impactam a estimativa de esforço em projetos de *software* em todas as fases de seu desenvolvimento. Para alcançar nossos objetivos foi realizada uma revisão sistemática de litaratura. A revisão retornou 859 trabalhos dos quais 53 foram relevantes para com os objetivos desta pesquisa.

Dessa forma, este trabalho possibilitou responder a seguinte pergunta de pesquisa principal:

(QPP). Quais são os fatores associados a estimativa de esforço?

- Este trabalho identificou 29 fatores de impacto na estimativa de esforço. Todos esses fatores oriundos de naturezas diversas e que afetam a estimativa de esforço direta ou indiretamente.

Com o objetivo de melhorar o entendimento a respeito do contexto desta pesquisa 3 (três) perguntas secundárias foram respondidas:

(QPS1). Quais são os *stakeholders* envolvidos?

- Observou-se uma frequência maior de "Estimadores", "Gerentes de projetos", "Desenvolvedores"e "Cliente". Foi possível concluir com este estudo que os fatores que envolvem estimativas de esforço influenciam, pricipalmente, os "Estimadores"e o *stalkeholders* menos afetado por esses fatores é o "Product Owner".

(QPS2). Qual o método/técnica utilizada no estudo?

- O objetivo dessa questão de pesquisa foi demonstrar, somente, os métodos/téncicas utilizados nos trabalhos relevantes desta revisão. Foi possível inferir que vários estudos analisados tinham como técnica principal a experiência de especialistas para relatar os fatores de impacto nas estimativas. Porém, a maior parte dos trabalhos analisados eram de origem explanatória ou comparativa com outros projetos de *software* finalizados.

(QPS3). Quais são os problemas de estimativas apontados no estudo?

- O objetivo foi apresentar o máximo possível de problemas associados ao fator de impacto. Foi possível identificar maior ocorrência de problemas de estimativas relacionados aos fatores "volatilidade dos requisitos"e "inexperiência do estimador".

Como trabalhos futuros, sugere-se um estudo detalhado a respeito dos fatores identificados. Esta revisão pode ter prosseguimento com as seguintes ideias de pesquisa sugeridas:

- Validar esta revisão a fim de comprovar sua autenticidade. A ideia é fazer pesquisas semiestruturadas com especialistas em estimação de esfoço e gerenciamento de projetos de *software* com o intuito de comprovar o impacto real dos fatores alcançados.
- Catalogar os fatores alcançados nesta revisão. O cátalago é utilizado para demonstrar os fatores existentes e para inserir novos fatores. A construção da plataforma do catálago precisa ser definida e o mesmo será populado com base na revisão sistemática de literatura realizada neste estudo.
- Medir o grau de impacto que os fatores alcançados possuem sobre a estimativa de esforço. Com isso, verificar o grau em que os fatores realmente pioram ou contribuem no processo de estimativa de esforço.
- Validar a qualidade dos estudos utilizados nesta revisão.

REFERÊNCIAS

ABRAN, A. **Software project estimation**: the fundamentals for providing high quality information to decision makers. [S.l.: s.n.], 2015. 1-261 p.

ANDA, B.; DREIEM, H.; SJØBERG, D. I.; JØRGENSEN, M. **Estimating software development effort based on use cases—experiences from industry**. In: Springer. **International Conference on the Unified Modeling Language**. [S.l.], 2001. p. 487–502.

BAJWA, S. S.; GENCEL, C. **What are the significant cost drivers for COSMIC functional size based effort estimation?** In: Springer. **International Workshop on Software Measurement**. [S.l.], 2009. p. 62–75.

BARRY, E. J.; MUKHOPADHYAY, T.; SLAUGHTER, S. A. Software project duration and effort an empirical study: an empirical study. **Information Technology and Management**, Springer, v. 3, n. 1-2, p. 113–136, 2002.

BERKUN, S. **The art of project management**. [S.l.]: O'Reilly, 2005. v. 4.

BHARDWAJ, M.; RANA, A. Impact of size and productivity on testing and rework efforts for web-based development projects. **ACM SIGSOFT Software Engineering Notes**, ACM, v. 40, n. 2, p. 1–4, 2015.

BOEHM, B.; ABTS, C.; CHULANI, S. Software development cost estimation approaches—a survey. **Annals of Software Engineering**, Springer, v. 10, n. 1-4, p. 177–205, 2000.

BVANSS, P.; RAMAIAH, P. S. A case study on software project development cost, schedule, and effort estimation. **Asian Journal of Pharmaceutical and Clinical Research**, v. 10, p. 10–14, 2017.

CAO, L. **Estimating agile software project effort**: an empirical study. **14th Americas Conference on Information Systems, AMCIS 2008**, [S. l.: s. n.], p. 1907–1916, 2008.

DIEV, S. Use cases modeling and software estimation applying: use case points. **ACM SIGSOFT Software Engineering Notes**, ACM, v. 31, n. 6, p. 1–4, 2006.

ERASMUS, I.; DANEVA, M. Erp effort estimation based on expert judgments. **Proceedings - Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement, IWSM-MENSURA 2013**, [S.l.]:IEEE, p. 104–109, 2013.

FELLIR, F.; NAFIL, K.; TOUAHNI, R. Analyzing the non-functional requirements to improve accuracy of software effort estimation through case based reasoning. **International Conference on Intelligent Systems: Theories and Applications, SITA**, [S. l.: s. n.], 2015.

GARCIA, L. **Engenharia de software**. [S. l.: s. n.], 2013.

HARMAN, M.; MCMINN, P.; SOUZA, J. T. D.; YOO, S. Search based software engineering: Techniques, taxonomy, tutorial. In: **Empirical Software Engineering and Verification**. [S.l.]: Springer, 2010. p. 1–59.

HILL, J.; THOMAS, L.; ALLEN, D. Experts' estimates of task durations in software development projects. **International Journal of Project Management**, v. 18, n. 1, p. 13–21, 2000.

HUANCA, L. M.; ORÉ, S. B. Factors affecting the accuracy of use case points. In: Springer. **International Conference on Software Process Improvement**. [S.l.], 2016. p. 133–142.

INC, P. Um guia do conhecimento de gerenciamento de projetos (guia pmbok®). 6. ed. **Newton Square**, 2017.

JØRGENSEN, M. A review of studies on expert estimation of software development effort. **Journal of Systems and Software**, Elsevier, v. 70, n. 1-2, p. 37–60, 2004.

JORGENSEN, M.; MOLOKKEN-OSTVOLD, K. Reasons for software effort estimation error: impact of respondent role, information collection approach, and data analysis method. **IEEE Transactions on Software Engineering**, IEEE, v. 30, n. 12, p. 993–1007, 2004.

KAKIMOTO, T.; TSUNODA, M.; MONDEN, A. Should duration and team size be used for effort estimation? In: Springer. **International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing**. [S.l.], 2017. p. 91–105.

KHATIBI, B. V.; JAWAWI, D.; KHATIBI, E. Investigating the effect of using methodology on development effort in software projects. **International Journal of Software Engineering and its Applications**, v. 6, n. 2, p. 35–46, 2012.

KHATIBI, E.; BARDSIRI, V. Model to estimate the software development effort based on in-depth analysis of project attributes. **IET Software**, v. 9, n. 4, p. 109–118, 2015.

KITCHENHAM, B.; CHARTERS, S. **Guidelines for performing systematic literature reviews in software engineering**. Citeseer, 2007.

KOSLOSKI, R.; OLIVEIRA, K. M. de. An experience factory to improve software development effort estimates. In: Springer. **International Conference on Product Focused Software Process Improvement**. [S.l.], 2005. p. 560–573.

LAKHANPAL, B. Understanding the factors influencing the performance of software development groups: an exploratory group-level analysis. **Information and Software Technology**, v. 35, n. 8, p. 468–473, 1993.

LAQRICHI, S.; GOURC, D.; MARMIER, F. Toward an effort estimation model for software projects integrating risk. **arXiv:1509.602**, 2015.

LAVAZZA, L.; VALETTO, G. Requirements-based estimation of change costs. **Empirical Software Engineering**, Springer, v. 5, n. 3, p. 229–243, 2000.

LEDERER, A.; PRASAD, J. The validation of a political model of information systems development cost estimating. **Proceedings of the ACM SIGCPR Conference**, [S. l.: s. n.], p. 164–173, 1991.

LEDERER, A.; PRASAD, J. Nine management guidelines for better cost estimating. **Communications of the ACM**, v. 35, n. 2, p. 51–59, 1992.

LEDERER, A. L.; PRASAD, J. Information systems software cost estimating: a current assessment. **Journal of Information Technology**, Springer, v. 8, n. 1, p. 22–33, 1993.

LOPES, J. d. S. **Guia prático em análise de ponto de função**. [S.l.]: Departamento de Informática, Viçosa, 2011.

MATOS, O.; FORTALEZA, L.; CONTE, T.; MENDES, E. Realising web effort estimation: a qualitative investigation. **ACM International Conference Proceeding Series**, [S. l.: s. n.], p. 12–23, 2013.

MCCONNELL, S. **Software estimation**: demystifying the black art. [S.l.]: Microsoft press, 2006.

MENDONÇA, A.; ALENCAR, F. **Métricas de tamanho de software com métodos ágeis no setor público**: uma revisao sistemática. 2019.

MOHAGHEGHI, P.; ANDA, B.; CONRADI, R. Effort estimation of use cases for incremental large-scale software development. **Proceedings - 27th International Conference on Software Engineering, ICSE05**, [S. l.: s. n.], p. 303–311, 2005.

MOHAPATRA, S.; GUPTA, D. Finding factors impacting productivity in software development project using structured equation modelling. **International Journal of Information Processing and Management**, v. 2, n. 1, p. 90–100, 2011.

MOLOKKEN, K.; JORGENSEN, M. A review of software surveys on software effort estimation. In: IEEE. **International Symposium on Empirical Software Engineering**. [S.l.], 2003. p. 223–230.

MORGENSHTERN, O.; RAZ, T.; DVIR, D. Factors affecting duration and effort estimation errors in software development projects. **Information and Software Technology**, Scopus, v. 49, n. 8, p. 827–837, 2007.

MOSES, J. Measuring effort estimation uncertainty to improve client confidence. **Software Quality Journal**, Springer, v. 10, n. 2, p. 135–148, 2002.

MRENAK, G. Evolving concepts, or why users often don't recognize the software they asked for. In: ACM. **Proceedings of the seventh Washington Ada symposium on Ada**. [S.l.], 1990. p. 17–22.

NAN, N.; HARTER, D. Impact of budget and schedule pressure on software development cycle time and effort. **IEEE Transactions on Software Engineering**, v. 35, n. 5, p. 624–637, 2009.

NGUYEN, T.-H.; MARMIER, F.; GOURC, D. A decision-making tool to maximize chances of meeting project commitments. **International Journal of Production Economics**, Elsevier, v. 142, n. 2, p. 214–224, 2013.

O'KEEFFE, J. F. F. **Análise de fatores de impacto no erro de estimativa de esforço e de duração em projetos de software**. Pontifícia Universidade Católica do Rio Grande do Sul, 2012.

PENDHARKAR, P. C.; RODGER, J. A. An empirical study of the impact of team size on software development effort. **Information Technology and Management**, Springer, v. 8, n. 4, p. 253–262, 2007.

PRESSMAN, R.; MAXIM, B. **Engenharia de software**. 8. ed [*S.l*]: McGraw Hill Brasil, 2016.

PUTNAM, L. H.; MYERS, W. **Five core metrics**: the intelligence behind successful software management. [S.l.]: Pearson Education, 2013.

RAMMAGE, R.; LEI, H.; CLAUS, M. Expert software development estimation with uncertainty correction. **2nd International Conference on Software Engineering and Data Mining**, [S. l.: s. n.], p. 624–630, 2010.

RODRÍGUEZ, D.; SICILIA, M.; GARCÍA, E.; HARRISON, R. Empirical findings on team size and productivity in software development. **Journal of Systems and Software**, Scopus, v. 85, n. 3, p. 562–570, 2012.

ROPPONEN, J.; LYYTINEN, K. Can software risk management improve system development: an exploratory study. **European Journal of Information Systems**, Taylor & Francis, v. 6, n. 1, p. 41–50, 1997.

SALMANOĞLU, M.; DEMIRÖRS, O.; TÜRETKEN, O. **Exploration of a method for COSMIC size estimation from S-BPM**. In: Scopus. [S.l.], 2014. p. 55–66.

SAVOLAINEN, J.; KUUSELA, J. Violatility analysis framework for product lines. In: ACM. **ACM SIGSOFT Software Engineering Notes**. [S.l.], 2001. v. 26, n. 3, p. 133–141.

SOMMERVILLE, I. **Software engineering**. [S.l.]: Addison-Wesley, 2011.

TRAVASSOS, G. H.; GUROV, D.; AMARAL, E. **Introdução à engenharia de software experimental**. UFRJ, 2002.

TRENDOWICZ, A.; JEFFERY, R. Common factors influencing software project effort. In: **Software Project Effort Estimation**. [S.l.]: Springer, 2014. p. 47–80.

TRENDOWICZ, A.; MÜNCH, J.; JEFFERY, R. State of the practice in software effort estimation: a survey and literature review. In: Springer. **IFIP Central and East European Conference on Software Engineering Techniques**. [S.l.], 2008. p. 232–245.

TRUDEL, S.; ABRAN, A. Functional size measurement quality challenges for inexperienced measurers. In: Springer. **International Workshop on Software Measurement**. [S.l.], 2009. p. 157–169.

TSUNODA, M.; MONDEN, A.; MATSUMOTO, K.; HATANO, R.; NAKANO, T.; FUKUCHI, Y. Analyzing risk factors affecting project cost overrun. In: **Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing 2012**. [S.l.]: Springer, 2013. p. 171–184.

UNGAN, E.; DEMIRÖRS, O.; TOP, Ö. Ö.; ÖZKAN, B. An experimental study on the reliability of cosmic measurement results. In: Springer. **International Workshop on Software Measurement**. [S.l.], 2009. p. 321–336.

VALENTE, F. F. R.; FALBO, R. Uso de gerência de conhecimento para apoiar a realização de estimativas. In: **Proceedings of the XXVIII Latin-American Conference on Informatics-CLEI**. [S.l.: s.n.], 2002.

VERNER, J. M.; EVANCO, W. M.; CERPA, N. State of the practice: An exploratory analysis of schedule estimation and software project success prediction. **Information and Software Technology**, Elsevier, v. 49, n. 2, p. 181–193, 2007.

VLIET, H. V.; VLIET, H. V.; VLIET, J. V. **Software engineering**: principles and practice. [S.l.]: Citeseer, 2008. v. 13.

WALKERDEN, F.; JEFFERY, R. An empirical study of analogy-based software effort estimation. **Empirical Software Engineering**, Springer, v. 4, n. 2, p. 135–158, 1999.

WOHLIN, C.; RUNESON, P.; HÖST, M.; OHLSSON, M. C.; REGNELL, B.; WESSLÉN, A. **Experimentation in software engineering**. [S.l.]: Springer Science & Business Media, 2012.

XAVIER, C. M.; VIVACQUA, F.; MACEDO, O.; XAVIER, L. S. **Metodologia de gerenciamento de projetos**. Scopus, 2005.

XU, B. Knowledge based micro-estimation in task arrangement for cooperative global software design. **Proceedings of the 2006 International Conference on Machine Learning and Cybernetics**, [S. l.: s. n.], p. 682–686, 2006.

ZAPATA, A.; CHAUDRON, M. An empirical study into the accuracy of it estimations and its influencing factors. **International Journal of Software Engineering and Knowledge Engineering**, v. 23, n. 4, p. 409–432, 2013.

ZAPATA, A. H.; CHAUDRON, M. R. An analysis of accuracy and learning in software project estimating. In: IEEE. **2012 38th Euromicro Conference on Software Engineering and Advanced Applications**. [S.l.], 2012. p. 414–421.

APÊNDICE A – TABELAS DOS PROBLEMAS DE ESTIMATIVA

Tabela 8 – Problemas de estimativa (Parte 1).

Fatores e Referências	Problemas
Disponibilidade de dados históricos: (RAMMAGE *et al.*, 2010)	Técnicas de estimativa orientada a dados requerem dados históricos de anos para serem confiáveis. Isso significa que o esforço de anos pode ser necessário antes que a organização perceba benefícios práticos de estimativa.
Uso de sistema de gerenciamento: (ZAPATA; CHAUDRON, 2013)	Na pesquisa do autor para saber quão útil é o *software* atual da empresa, que guarda dados de estimativa passadas, para realizar estimativas futuras, conclui-se que os dados da amostra não são úteis para prever o trabalho do projeto. Porém, para prever o cronograma e o custo, o resultado é mais útil e pode ser usado como uma estimativa aproximada. Sistemas de gerenciamento devem seguir padrões como ISO, CMMI etc.
Complexidade e tamanho do *software*:(RAMMAGE *et al.*, 2010) (SALMANOĞLU *et al.*, 2014)	A complexidade do projeto é difícil de quantificar ou medir diretamente. Na prática, torna-se mais simples medir várias outras variáveis do projeto que podem correlacionar-se com a complexi–dade do problema. Essas variáveis são frequentemente chamadas de "*metrics*"ou "*called proxy*". A maioria desses preditores, que podem incluir tipos de requisitos, pontos de função, número de arquivos, complexidade cíclica ou números de casos de teste, fornecem quantificação do tamanho do problema. Linhas de cógido fonte (SLOC) são frequentemente utilizadas, mas por si só não podem ser um preditor porque o tamanho do *software* não é conhecido até que o mesmo seja finalizado. Outro estudo aborda que é preciso estimar o tamanho do *software* antecipadamente, ou seja, antes do ciclo de desenvolvimento. Entradas para modelos de estimativa de tamanho são geralemente artefatos de *software* que são produzidos posteriormente do ciclo de vida de desenvolvimento do *software*.
Colaboração da equipe: (LAKHANPAL, 1993)	Problemas de desempenho do desenvolvimento ocasionados por falta de colaboração da equipe são apontados no estudo. Abordam, também, implicações importantes para as equipes de desenvolvimento que trabalham em projetos de *software*.

Fonte: Elaborado pela autor.

Tabela 9 – Problemas de estimativa (Parte 2).

Fatores e Referências	Problemas
Natureza dos projetos: (KHATIBI; BARDSIRI, 2015) (MATOS *et al.*, 2013)	Comparar estimativas de projetos que estão inseridos em contextos diferentes pode ser um problema.O peso dos atributos podem ser diferentes de um grupo para o outro de acordo com a natureza dos projetos. Melhorar os desvios de comparações não confiáveis podem melhorar consideravelmente a precisão e a confiabilidade das estimativas.
Experiência na tecnologia: (MOHAPATRA; GUPTA, 2011)	A produtividade do tarefa desempenhada depende da experiência de quem está realizando a atividade. O tempo que um desenvolvedor leva para executar uma atividade varia com o tempo de outro desenvolvedor para realizar a mesma atividade de acordo com sua experiência na tecnologia do projeto.
Envolvimento do cliente: (MOHAPATRA; GUPTA, 2011) (ZAPATA; CHAUDRON, 2013)	Quanto mais o cliente apoiar nas atividades, mais fácil será pra a equipe do projeto avançar na sua execução. É comum erros de estimativa ocasionados pela falta de interação entre o cliente, estimador e equipe de desenvolvimento de *software*. Uma exploração adicional dos projetos incluídos na pesquisa de (ZAPATA; CHAUDRON, 2013) revelou que o baixo envolvimento de clientes no desenvolvimento de *software* podem acabar com os benefícios da pressão do cronograma.
Disponibilidade de recursos: (MOHAPATRA; GUPTA, 2011)	Problemas surgem quando as estimativas de *software* consideram que o número de pessoas, computadores dentre outros recursos relacionados na estimativa serão de qualidade ou estarão sempre disponíveis, sendo que muitas vezes isso não acontece. A indisponibilidade de pessoas e desses recursos ou mau funcioanamento deles podem causar atrasos significativos no desenvolvimento.
Otimismo do estimador: (ZAPATA; CHAUDRON, 2013) (ERASMUS; DANEVA, 2013)	O otimismo do estimador precisa ser considerado, pois causa falhas nas estimativas. Entretando, esse otimismo pode ser mitigado por estimadores experientes, conhecimento do domínio comercial ou conhecimento técnico.
Experiência em projetos de *software*: (ZAPATA; CHAUDRON, 2013)	A experiência da equipe associa-se diretamente com acurácia das estimativas. Esse fator é resultante de imprecisões e ocasiona atrasos no cronograma do projeto.
Envolvimento do estimador: (ZAPATA; CHAUDRON, 2013)	O não envolvimento do estimador com o projeto pode causar imprecisão de estimativas. Na pesquisa realizada 14% dos entrevistados responderam que participaram entre 50% e 75% dos projetos que estimiram, e 59% responderam entre 75% e 100%.

Fonte: Elaborado pela autor.

Tabela 10 – Problemas de estimativa (Parte 3).

Fatores e Referências	Problemas
Tamanho da equipe: (RODRÍGUEZ *et al.*, 2012) (KAKIMOTO *et al.*, 2017)	Se o tamanho ideal da equipe puder ser encontrado a decomposição de projetos em pedaços menores poderia se tornar uma prática chave no gerenciamento de distribuição de equipes por projeto. Esse tamanho ideal pode ocasionar melhorias na produtividade e, consequentemente, nas estimativas. Outro estudo assumi que o tamanho máximo da equipe incluem erros quando o esforço é estimado. O autor conclui que o uso do tamanho máximo da equipe como variável idependente foi satisfatório quando a taxa de erro não é muito grande (igaul ou inferior a 50%). Porém quando a taxa é igual ou superior a 100%, a precisão da estimativa é pior na maioria dos casos.
Pressão externa para reduções de estimativas ou comportamento político de estimativa: (NAN; HARTER, 2009) (LEDERER; PRASAD, 1991)	Problemas de estimativa surgem por causa de orçamentos apertados, com isso o responsável pelo gerenciamento do projeto pode optar por economizar tempo e mão de obra. O estudo indica que o tempo de ciclo reduzido sob pressão orçamentária não é compensado pelo aumento de trabalho. Por fim, os gerentes podem usar dados históricos para estimar a faixa benéfica de pressão orçamentária e com isso realizar o gerenciamento do *software* com mais eficácia. Problemas surgem nas estimativas de custo quando os mesmos são influenciados por comportamentos políticos de estimativa.
Clareza dos objetivos: (HILL *et al.*, 2000) (TSUNODA *et al.*, 2013)	O estudo indica que estimadores cometeram, no estudo de caso avaliado, maiores erros médios subestimados em tarefas que não foram claramente definidas. Como resultado gerentes de projeto tendiam a superestimar cerca de 60% das tarefas, mas se subestimavam, os erros de estimativa eram maiores. Outro autor indica a clareza das metas como fator que têm forte ligação com a superação do custo de desenvolvimento de *software*.
Considerar o *design* do sistema: (DIEV, 2006)	O objetivo do estudo é demonstrar que para fornecer estimativas com base em casos de uso é necessário levar em consideração o *design* do sistema.
Confiança do cliente: (MOSES, 2002)	O autor propõe que, medindo as incertezas na estimativa de esforço, pode-se melhorar a confiança de duas maneiras. Em primeiro lugar, os estimadores podem ser ajudados a melhorar a consistência de suas estimativas e declarações podem ser feitas aos cientes sobre o provável esforço que um projeto realmente levará, além de demonstrar erros de estimativa para acima ou abaixo do estipulado.

Fonte: Elaborado pela autor.

Tabela 11 – Problemas de estimativa (Parte 4).

Fatores e Referências	Problemas
Requisitos voláteis: (MRENAK, 1990) (CAO, 2008) (LAVAZZA; VALETTO, 2000) (FELLIR *et al.*, 2015) (LEDERER; PRASAD, 1993)	Usuários raramente tem suas necessidades bem definidas, principalmente no início de desenvolvimento do projeto. Os estudos concluem que o *feedback* rápido das iterações anteriores e a transparência do processo de desenvolvimento em metodologias ágeis permite o ajuste contínuo da estimativa durante o desenvolvimento do *software*. Outra abordagem é que o tamanho e a complexidade do código são estimados com base no tamanho e complexidade dos requisitos. Tudo isso com o intuito de melhorar a precisão de estimativas de custo. No estudo de Lederer e Prasad (1993), o grande problema especificado é dado pelas solicitações frequentes de alteração pelo usuário. Pode ser muito difícil para os desenvolvedores e estimadores antecipar e ter controle destas alterações. O estudo conclui que o gerenciamento eficaz, em vez de uma estimativa precisa pode ser a chave para a conclusão do projeto dentro das estimativas de custos especificadas no início do projeto.
Detalhamento antecipado dos requisitos: (TRUDEL; ABRAN, 2009) (UNGAN *et al.*, 2009)	Durante as fases iniciais de um ciclo de vida de desenvolvimento de *software*, os requistos documentados são usados como insumo para o processo de estimativa, inclusive para medir o tamanho funcional do *software* a ser desenvolvido. A qualidade de um requisito documentado terá impacto na consistência dos resultados de medição, bem como na confiança dos resultados obtidos. Problemas de medição defeituosa que acontecem quando o medidor não pôde aplicar as regras do método COSMIC. E de suposições incorretas, onde se inclui mais requisitos do que o especificado, baseado na experiência pessoal.
Detalhamento antecipado do planejamento: (LEDERER; PRASAD, 1992)	O estudo procura analisar a mitigação de erro de estimativa do *software* com base nas especificações do planejamento inicial do sistema.
Reserva de planejamento (buffer): (XAVIER *et al.*, 2005)	As reservas são uma precaução no planejamento do projeto para mitigar os riscos de cronograma e custo.

Fonte: Elaborado pela autor.

Tabela 12 – Problemas de estimativa (Parte 5).

Fatores e Referências	Problemas
Fatores ambientais: (HUANCA; ORÉ, 2016) (ANDA *et al.*, 2001)	O estudo apresenta alguns pontos fracos do método pontos de caso de uso. Um desses pontos fracos trata-se de que fatores ambientais estejam desatualizados.
Gerenciamento de riscos: (ROPPONEN; LYYTINEN, 1997)	O estudo investiga características e práticas de gerenciamento de riscos e procura fatores ambientais e de processo que se relacionam e melhoram o gerenciamento de riscos. Mitigando o impacto de estimativas de tamanho incorretas.
Uso de método de estimativa: (TRENDOWICZ *et al.*, 2008)	Problemas ocorrem quando o método aplicado não está alinhado com os objetivos definidos ou os recursos disponíveis. O método de estimativa requer mais recursos do que existe atualmente na organização, por exemplo.
Inexperiência do estimador: (WALKERDEN; JEFFERY, 1999) (MOHAGHEGHI *et al.*, 2005) (MOHAGHEGHI *et al.*, 2005)	O estudo de Mohagheghi *et al.* (2005) explica que o parecer de um especialista apenas, não deve ser usado com método de estimativa de custos. Aponta -se a necessidade de complemantar o julgamento com processos que respondem pelo viés observado.
Experiência na área de negócios do sistema: (KOSLOSKI; OLIVEIRA, 2005)	O estudo de Kosloski e Oliveira (2005) propõe definir uma estrutura de características que afetam a produtividade do projeto de *software* e suas estimativas.
Uso de metodologia de desenvolvimento: (KHATIBI *et al.*, 2012)	O estudo procura entender o efeito do uso de metodologias nas estimativas de desenvolvimento de *software*. O estudo conclui que apesar de apresentar vantagens, o uso de metodologias podem levar a estimativas negativas do esforço de desenvolvimento, porque não existem evidências de quanto o uso de metodologias de desenvolvimento podem influenciar na estimativa final do projeto.
Subestimar custos: (LEDERER; PRASAD, 1992)	Subestimar custos nas suas estimativas destrem a credibilidade dos estimadores e desenvolvedores. Segundo o autor, quase dois terços de todos os grandes projetos superam suas estimativas.
Imprevisibilidade das tarefas: (LEDERER; PRASAD, 1992)	A confiabilidade da estimativa leva ao sucesso do projto ou falha. A ideia do autor é trabalhar com a análise de pontos de função e incluir o conceito de agendamento de força de trabalho de uma maneira melhora na hora de tomar decisões de estimativa.

Fonte: Elaborado pela autor.