UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

GIOVANA BRITO OLIVEIRA

ANTECEDENTES E CONSEQUENTES EM MODELOS DE SD PARA GESTÃO DE PROJETOS DE SOFTWARE

CAMPINA GRANDE - PB

2023

GIOVANA BRITO OLIVEIRA

ANTECEDENTES E CONSEQUENTES EM MODELOS DE SD PARA GESTÃO DE PROJETOS DE SOFTWARE

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharela em Ciência da Computação.

Orientador: Professor Dr. José Antão Beltrão Moura

CAMPINA GRANDE - PB

2023

GIOVANA BRITO OLIVEIRA

ANTECEDENTES E CONSEQUENTES EM MODELOS DE SD PARA GESTÃO DE PROJETOS DE SOFTWARE

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharela em Ciência da Computação.

BANCA EXAMINADORA:

Professor Dr. José Antão Beltrão Moura

Orientador – UASC/CEEI/UFCG

Professor Dr. Leandro Balby Marinho

Examinador – UASC/CEEI/UFCG

Professor Tiago Lima Massoni

Professor da Disciplina TCC – UASC/CEEI/UFCG

Trabalho aprovado em: 14 de Fevereiro de 2021.

CAMPINA GRANDE - PB

ABSTRACT

When making decisions, software project managers need to consider several factors, making this activity a very complex one. In order to help managers with these decisions, tools have been used to simulate the impact of these factors on project results. Models in System Dynamics (SD) have been shown to be a good option for such simulations because they have dynamic characteristics and use "Feedback" systems, typical aspects of software project development. The aim of this work is to identify which factors (antecedents) influence software projects and what they influence (consequents) in projects that have already been modeled in SD. For this, a mapping was performed using the Web of Science (WoS) indexing base. The articles were evaluated by order of publication, starting with the most recent and considering the proposed inclusion and exclusion criteria. Some of the factors mapped were: customer influence, team promotion, schedule pressure, overtime (antecedents); team productivity, project cost and project duration (consequents). The expected contribution is to help researchers who intend to build SD models to find factors not yet modeled or reuse factors already modeled by other researchers, avoiding re-work.

Keywords: Software Project, System Dynamics, Management, Simulation Modeling.

ANTECEDENTES E CONSEQUENTES EM MODELOS DE SD PARA GESTÃO DE PROJETOS DE SOFTWARE

Giovana Brito Oliveira
giovana.oliveira@ccc.ufcg.edu.br
Universidade Federal de Campina Grande

Giseldo da Silva Néo
giseldo@copin.ufcg.edu.br
Universidade Federal de Campina Grande

José Antão Beltrão Moura
antao@computacao.ufcg.edu.br
Universidade Federal de Campina Grande

RESUMO

Ao tomar decisões, gerentes de projetos de software precisam considerar diversos fatores, tornando essa uma atividade muito complexa. Para auxiliar os gerentes com essas decisões, ferramentas têm sido utilizadas para simular o impacto desses fatores nos resultados do projeto. Modelos em Dinâmica de Sistemas (SD) têm se mostrado uma boa opção para tais simulações por possuírem características dinâmicas e utilizarem sistemas de "Feedback", aspectos próprios do desenvolvimento de projetos de software. O objetivo deste trabalho é identificar quais os fatores (antecedentes) que influenciam os projetos de software e o que eles influenciam (consequentes) nos projetos que já foram modelados em SD. Para isso foi realizado um mapeamento por meio da base de indexação Web of Science (WoS). Os artigos foram avaliados por ordem de publicação partindo do mais recente e considerando os critérios de inclusão e exclusão propostos. Alguns dos fatores mapeados foram: influência do cliente, promoção da equipe, pressão de cronograma, horas extras (antecedentes); produtividade do time, custo do projeto e duração do projeto (consequentes). A contribuição esperada é auxiliar pesquisadores que pretendem construir modelos SD a encontrarem fatores ainda não modelados ou reaproveitar os fatores já modelados por outros pesquisadores evitando re-trabalho.

PALAVRAS-CHAVE

Projeto de software, Dinâmica de sistemas, Gerenciamento, Modelagem de simulação.

1. INTRODUÇÃO

O gerenciamento de projetos de software é a aplicação de conhecimentos, habilidades, ferramentas e técnicas a todos os procedimentos para atender a todos os tipos de requisitos do projeto. Uma vez que os gerentes precisam levar em consideração dependências, restrições e incertezas complexas, estimar esses parâmetros corretamente é um empreendimento difícil [11]. As decisões tomadas por um gerente de projetos podem gerar grande impacto no andamento e consequentemente no resultado final do mesmo.

Dada a natureza complexa do trabalho de um gerente de projetos, muito se tem pesquisado a respeito de fatores que possam influenciar na produtividade do time de desenvolvimento [2]. Quando se sabe quais fatores possuem grande poder de influência no andamento de projetos de software é possível criar ferramentas que simulam a atuação destes fatores para auxiliar nas tomadas de decisões gerenciais.

Dinâmica de Sistemas (SD) é a aplicação de princípios e técnicas de sistemas de controle de feedback para problemas gerenciais, organizacionais e socioeconômicos [1]. O método SD de modelagem do sistema permite analisar um processo gerenciado de conclusão do projeto de modo a: modelar as formas pelas quais seus componentes de informação, ação e consequências interagem para gerar comportamento dinâmico; diagnosticar as causas do comportamento defeituoso; ajustar seus loops de feedback para obter um melhor comportamento [8].

Sendo assim, o modelo de dinâmica de sistemas pode ser usado como laboratório para diferentes simulações de situações, que ajudam os gerentes a formar uma compreensão mais clara dos eventos presentes e futuros" [11]. Dadas suas características dinâmicas, os modelos de SD oferecem suporte para insights mais estratégicos quanto a eficácia de diferentes decisões gerenciais se comparados a outras ferramentas que consideram apenas características estáticas.

Dado o devido contexto, este trabalho possui o objetivo de mapear os modelos SD para gerenciamento de projetos de software que já foram desenvolvidos pela academia ao responder a seguinte questão de pesquisa:

- QP1 - Quais são os fatores (ou parâmetros, ou influenciadores, ou antecedentes) - e o que eles influenciam (ou consequentes) - que foram mapeados em SD para projetos de software?

Este trabalho busca lançar luz sobre possível(is) resposta(s) à QP1 e sua principal contribuição é auxiliar pesquisadores que pretendem construir modelos SD a encontrarem fatores influenciadores em gerência de projetos de software que ainda não foram modelados ou reaproveitar os fatores que já foram modelados por outros pesquisadores evitando re-trabalho.

Este trabalho foi realizado em acompanhamento à pesquisa do aluno de doutorado da UFCG Giseldo da Silva Neo, cujo tema de pesquisa é "Engenharia de Software: Estimativa de esforço em desenvolvimento de software". Seu trabalho tem como objetivo o desenvolvimento de uma ferramenta híbrida que se utiliza de técnicas de Inteligência Artificial e SD para realizar estimativas de esforço de desenvolvimento.

O restante deste trabalho é organizado da seguinte maneira: a Seção 2 descreve os principais conceitos e terminologias presentes neste trabalho; a Seção 3 descreve a metodologia empregada para a seleção de estudos; a Seção 4 apresenta os resultados colhidos; a

Seção 5 apresenta a discussão sobre os resultados; e, por fim, a Seção 6 apresenta as conclusões, limitações e trabalhos futuros.

2. EMBASAMENTO TEÓRICO

Esta seção introduz terminologias utilizadas neste trabalho e discute brevemente acerca do funcionamento da Dinâmica de Sistemas.

2.1 Dinâmica de Sistemas

A Dinâmica de Sistemas, nascida inicialmente com o nome de "Dinâmica Industrial" por Jay Forrester em 1961, originou-se da teoria da dinâmica não linear e controle realimentado da matemática, física e engenharia [3]. Dinâmica de Sistemas (SD) é a aplicação de princípios e técnicas de sistemas de controle de feedback para problemas gerenciais, organizacionais e socioeconômicos [1].

A filosofia da Dinâmica de Sistemas é baseada e algumas premissas ((Forrester, 1961), e (Roberts, 1981)):

1. O comportamento (ou história temporal) de uma entidade organizacional é causado principalmente por sua estrutura. A estrutura inclui não apenas os aspectos físicos, mas principalmente as políticas e procedimentos, tangíveis e intangíveis, que dominam a tomada de decisão na entidade organizacional.
2. A tomada de decisão gerencial ocorre em uma estrutura que pertence à classe geral conhecida como sistemas de feedback de informação.
3. Nosso julgamento intuitivo não é confiável sobre como esses sistemas mudarão com o tempo, mesmo quando temos um bom conhecimento das partes individuais do sistema.
4. A experimentação de modelos agora é possível para preencher a lacuna onde nosso julgamento e conhecimento são mais fracos --- mostrando a maneira pela qual as partes separadas conhecidas do sistema podem interagir para produzir resultados gerais do sistema inesperados e problemáticos [1].

Com base nessas crenças filosóficas, foram estabelecidos dois fundamentos principais para a operacionalização da técnica. Esses são:

1. O uso de sistemas de *feedback* de informação para modelar e entender a estrutura do sistema (Premissas 1 e 2)
2. O uso de simulação de computador para entender o comportamento do sistema (Premissas 3 e 4) [1].

(a) O uso de sistemas de *feedback* de informações: "*Feedback*" é o processo no qual uma ação tomada por uma pessoa ou coisa acabará afetando essa pessoa ou coisa. Um *loop* de *feedback* é uma sequência fechada de causas e efeitos, um caminho fechado de ação e informação. Os *loops* de *feedback* dividem-se naturalmente em duas categorias que são rotuladas *feedback* de amplificação de desvio (DAF) ou *loops* positivos, e *feedback* de neutralização de desvio (DCF) ou *loops* negativos. Um conjunto interconectado de *loops* de *feedback* é um sistema de *feedback* (Richardson e Pugh, 1981) [1].

O primeiro ano de exploração (em *System Dynamics*) apontou para os conceitos de sistemas de feedback como sendo muito mais gerais, mais significativos e mais aplicáveis a sistemas sociais do que se pensava comumente... Os processos de *feedback* emergiram como universais em sistemas sociais e pareciam manter a chave para estruturar e esclarecer relacionamentos que permaneceram desconcertantes e contraditórios" (Forrester, 1968) [1].

A importância e aplicabilidade do conceito de sistemas de *feedback* para sistemas gerenciais tem, desde então, sido substanciada por um grande número de estudos no campo da Dinâmica de Sistemas. (Ver, por exemplo, Roberts, 1981). Mas o que talvez seja mais interessante é ver "endossos" do conceito de fora da comunidade *System Dynamics* [1]

2.2 SD no Gerenciamento de Projetos de Software

A presença do impacto do *feedback* no comportamento e melhoria do processo de desenvolvimento de software já foi demonstrado. SD se utiliza de técnicas de *feedback,* se mostrando adequado para a criação de modelos em processos de projetos de software.

Depois de revisar brevemente as dificuldades de alcançar grandes melhorias no processo global de desenvolvimento e manutenção de software, evolução do software, o artigo apresenta a hipótese FEAST. Isso afirma que tais problemas podem, pelo menos em parte, ser devidos à natureza de feedback desse processo e que isso é, geralmente, negligenciado na busca por tal melhoria. Os resultados até o momento do projeto FEAST/1 apóiam a hipótese e também as leis da evolução do software conforme reconhecidas e aprimoradas ao longo de cerca de 25 anos. O projeto está explorando o fenômeno em profundidade, identificando, modelando e simulando o comportamento evolutivo de vários projetos de software industrial usando técnicas de caixa preta e dinâmica de sistemas. Agora está demonstrando a presença e o impacto do *feedback* no comportamento e melhoria do processo e derivando diretrizes para planejamento e gerenciamento de processos de software [6].

Para uma compreensão mais profunda do tema é sugerida a leitura do livro *Software Process Dynamics* do autor Raymond J. Madachy[1].

3. METODOLOGIA

Para realização deste trabalho foi utilizada a metodologia de Mapeamento Sistemático dos fatores influenciadores já usados em modelos SD para gerenciamento de projetos de software.

Uma revisão sistemática da literatura é um estudo secundário com o objetivo de identificar, analisar e interpretar todas as evidências disponíveis de estudos primários relacionados a uma questão de pesquisa específica [5].

Um mapeamento sistemático de engenharia de software é um método definido para construir um esquema de classificação e estruturar um campo de interesse da engenharia de software. A análise dos resultados concentra-se nas frequências de publicações das categorias dentro do esquema. Assim, a cobertura do campo de pesquisa pode ser determinada. Diferentes facetas do esquema

[1]https://www.amazon.com.br/Software-Process-Dynamics-Raymond-Madachy/dp/0471274550

também podem ser combinadas para responder a questões de pesquisa mais específicas [7].

Na engenharia de software, mapeamentos sistemáticos têm se concentrado em estudos quantitativos e empíricos, mas existe um grande conjunto de métodos para sintetizar resultados de pesquisas qualitativas (Dixon-Woods et al. 2005) [7]. Mapeamento Sistemático foi escolhido como metodologia pois não apresenta a necessidade de uma ordenação de qualidade, sendo mais simples que uma revisão sistemática.

O Web of Science (anteriormente conhecido como Web of Knowledge) é um site que fornece acesso baseado em assinatura a vários bancos de dados que fornecem dados abrangentes de citações para muitas disciplinas acadêmicas diferentes. Foi originalmente produzido pelo *Institute for Scientific Information* (ISI) e atualmente é mantido pela *Clarivate Analytics* (anteriormente o negócio de Propriedade Intelectual e Ciência da Thomson Reuters) [10].

Por se tratar de um Trabalho de Conclusão de Curso, em que o tempo de desenvolvimento era limitado há 4 meses, o Web of Science foi escolhido como base de indexação dada a sua grande cobertura temporal e disciplinar, além de abranger mais de 50.000 livros acadêmicos, 12.000 periódicos e 160.000 anais de conferências [1], uma vez que não haveria tempo hábil para lidar com diversas bases.

3.1 Base de Busca e Termo

Foi feita uma pesquisa nos títulos, resumos e nas palavras-chave, na base Web of Science com os termos de busca na Tabela 1.

Tabela 1: Termos de busca

Termos de busca
(”system dynamics”) and (software and project and management)

Esses termos foram selecionados com o objetivo de capturar tudo que já foi publicado em SD na área de projetos de software.

Os critérios de aceitação e rejeição são apresentados na Tabela 2 e na Tabela 3.

Apresentar um diagrama SD é essencial como critério de inclusão com vistas a garantir a reprodutibilidade dos trabalhos aqui mapeados, dado o objetivo de auxiliar pesquisadores que podem ter interesse em reutilizar esses diagramas.

Tabela 2: Critérios de rejeição

Critérios de Rejeição
Não estar disponível para leitura completa
Não ser sobre engenharia de software (após leitura completa)
Não ser sobre engenharia de software (após leitura do título e do resumo)
Ser um revisão sistemática
Não conter um diagrama SD como uma imagem no texto

Tabela 3: Critérios de aceitação

Critérios de Aceitação
Apresentar um diagrama SD
Ser aplicado em projetos de software
Delinear um experimento

3.2 Procedimentos

Primeiro foi realizada uma busca no ***Web of Science*** a partir dos termos de buscas selecionados da Tabela 1.

Em seguida os documentos foram exportados para o ***End Note Web***, que é uma ferramenta de gestão de bibliografia na web com integração nativa com o *Web of Science*, com um clique os artigos resultantes da busca foram indexados no *End Note Web*. Já essa lista com os artigos no *End Note Web* é privada para cada usuário, diferente da consulta a partir dos termos no *Web of Science*. Caso haja interesse em conhecer os recursos do *End Note Web* basta realizar um cadastro gratuito e utilizá-lo online.

O principal uso do *End Note Web* para os autores dessa revisão foi gerar as referências no formato ***bibtex*** para serem importadas em um outro software de gestão de bibliografia para organizar os PDFs baixados, o ***Mendeley Desktop***, é possível saber mais no site oficial da ferramenta.

Apesar do *End Note* e do *Mendeley Desktop* serem softwares de gestão de bibliografia semelhantes e de apresentarem funcionalidades similares, a disponibilidade e velocidade do *Mendeley Desktop*, por sua natureza nativa para o *Windows,* foi preferido para gerir e manter os PDFs dos artigos selecionados em detrimento do *End Note Web*, existem versões do *End Note* para *Windows* mas elas não são gratuitas, diferentemente do *Mendeley Desktop* que é gratuito.

O processo de leitura do título e do resumo dos artigos iniciou e terminou no *Mendeley Desktop*.

Em seguida, para os documentos que não foram rejeitados na primeira leitura do título e do resumo, foi feita a tentativa de baixar os PDFs para leitura completa.

O próximo software utilizado para suporte da revisão sistemática foi o ***Parsif.al***, a partir do arquivo *bibtex* gerado pelo *End Note Web* os dados foram importados para o *Parsif.al*, pois a partir dele podemos gerenciar a extração dos dados dos artigos que serão lidos e que tem seus PDFs indexados no *Mendeley Desktop*, mantendo controle no processo de extração.

Uma das funcionalidades do *Mendeley Desktop* que foi mais utilizada pelos autores é o processo de anotações no texto pdf do documento, permitindo em momento posterior a impressão somente das anotações, este foi o fator que justificou o uso da ferramenta.

Uma das funcionalidades do *Parsif.al* é manter o registro de todo o processo de revisão sistemática e posterior impressão de relatórios e gráficos sobre o processo, sendo este o principal fator motivador do uso dessa ferramenta.

Dado limitações de tempo e equipe dos autores dessa pesquisa foi feita somente a leitura e extração até o limite de 6 artigos aceitos mais recentes. Não desclassificando os demais artigos coletados na pesquisa que não foram lidos e incluídos neste trabalho.

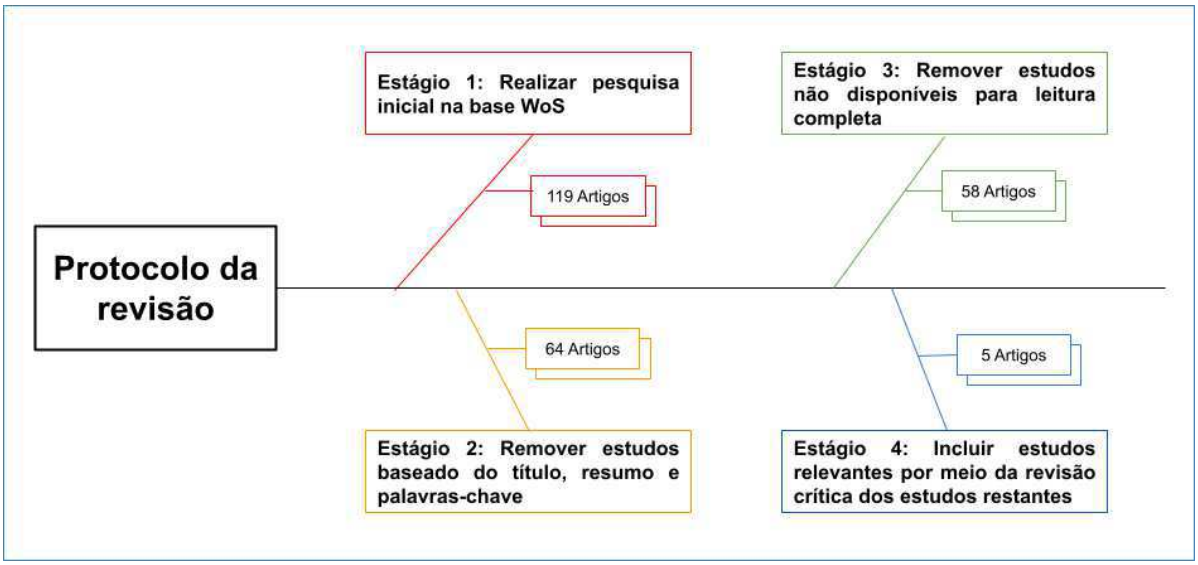

Figura 1: Protocolo de revisão

4. RESULTADOS

A consulta no WoS retornou 119 documentos que podem ser vistos neste link[2]. Para visualizar a lista é preciso realizar um login no Web of Science, o cadastro é gratuito. O resumo do protocolo de revisão seguido pode ser visto na Figura 1.

Em um primeiro momento foram rejeitados 55 artigos a partir da leitura do título e do resumo, segundo o critério ”C1 - Não ser sobre engenharia de software após leitura do título e do resumo”.

Para os 64 documentos restantes, que passaram em um primeiro momento pela leitura do título e do resumo, foi feita a tentativa de baixar os PDFs para leitura completa. Os 58 pdfs que estavam disponíveis foram baixados, com o suporte do Google Acadêmico e anexados ao *Mendeley Desktop*. Já 6 documentos não estavam disponíveis para leitura completa pelos autores (era exigida assinatura não fornecida pela UFCG) e portanto foram removidos pelo critério "C2 - Não estar disponível para leitura completa".

Dentre os 58 artigos disponíveis para leitura completa, 10 foram rejeitados pelo critério "C3 - Não ser sobre Engenharia de Software após leitura completa". 2 foram rejeitados pelo critério "C5 - Ser uma revisão sistemática". 2 foram rejeitados pelo critério "C1 - Não contém um diagrama SD como imagem". Dos artigos restantes, por uma questão de tempo, não foi possível realizar a leitura de 38 deles.

Dos artigos restantes, foram lidos os 20 mais recentes[3] e destes, 6 foram selecionados para leitura completa, minuciosa e constam na Tabela 4.

[2]https://www.webofscience.com/wos/woscc/summary/8d848a18-50e3-4fbd-8dee-21884b981184-64f09921/relevance/1

[3] Não foram (ainda) lidos todos por limitação de tempo no período 2022.1.

5. DISCUSSÃO

5.1 Artigos incluídos

O primeiro e mais recente artigo aceito nesse mapeamento sistemático foi (Pietron, 2020), onde foram mapeados como antecedentes a ***influência do cliente ou o risco de intervenção***, e ***a promoção da equipe***, para projetos que utilizam abordagem ágil, estes antecedentes influenciam na ***performance do projeto*** e nos ***resultados econômicos (custo)***.

Os diagramas de (Pietron, 2020) relacionados aos 2 antecedentes foram criados utilizando o software *Vensim PLE*. Estes antecedentes estão relacionados ao impacto das solicitações de mudança dos requisitos feitas pelo cliente no projeto, podemos entender esse antecedentes como mudança de escopo (*scope change*), dando uma taxonomia padronizada para esse antecedente, possivelmente essa classificação padronizada (para utilizar termos comuns) vai facilitar o entendimento do que foi modelado no SD nos próximos artigos a serem analisados.

O trabalho conta com uma descrição criteriosa do processo de criação do modelo, e contribui para o estado da arte ao modelar a influência de dois fatores. Como trabalho futuro, o modelo poderia ser utilizado com dados reais de diferentes projetos para identificar como os mesmos poderiam ser melhor sucedidos.

Em seguida o artigo (Hiekata et al., 2019) foi aceito, onde foram mapeados como antecedentes a *pressão no cronograma*, e ***horas extras***, a abordagem utilizada no experimento não foi informada, os antecedentes em questão influenciam no ***desempenho de saída*** do projeto.

Na figura 2 pode-se ver o Diagrama de Loop Causal para produtividade. Diagramas de laço causal (CLDs) ilustram relações de causa e efeito entre diferentes variáveis. Nos CLDs, as setas são usadas para indicar os relacionamentos e a polaridade é usada

Tabela 4: Resultados da extração dos dados dos artigo

#	Citação	Antecedentes	Consequentes	Metodologia
[9]	(Pietron, 2020)	Influência do cliente ou o risco de intervenção; Promoção da equipe	Performance do projeto; Resultados econômicos	Agile; Scrum
[4]	(Hiekata et al., 2019)	Pressão de cronograma; Horas extras	Desempenho de saída	Indefinido
[10]	(van Oorschot et al., 2018)	Comprimento da iteração	Qualidade do software; Custo do projeto; Duração do projeto	Agile
[13]	(Zhang et al., 2018)	Mudanças no projeto	Mão de obra, Produtividade; Custo do projeto; Duração do projeto; Qualidade do software	Agile
[2]	(Fatema and Sakib, 2017)	Gerenciamento da equipe; Eficácia da equipe;\\ Motivação, Dependência externa; Habilidade; Cultura; Fatores externos	Produtividade do time	Agile
[7]	(Mishra and Mahanty, 2016)	Divisão do trabalho local e offshore; Esforço de treinamento de equipe local e offshore	Custo do projeto; Duração do projeto; Qualidade do software	Indefinido

para definir o status do relacionamento. Nesta figura, o loop de burn out define que, devido às horas extras, o funcionário fica exausto e, portanto, diminui a produtividade e também aumenta o erro. A produtividade reduzida mantém a quantidade de trabalho a fazer maior e, portanto, é necessário mais tempo de conclusão. Isso também aumenta a contratação de mão de obra e causa diluição de competências. Por outro lado, aumentar os recursos humanos também aumenta a sobrecarga de comunicação, o que diminui a produtividade e aumenta a fração de erro. No caso de contratação de novos integrantes, é um processo demorado e a defasagem entre a contratação e a força de trabalho define essa situação [4] .

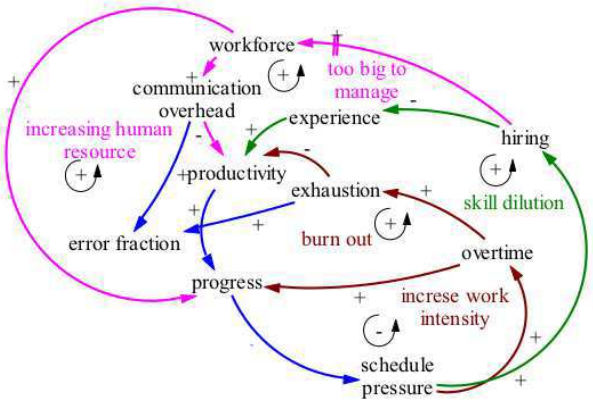

Figure 3. Causal Loop Diagram for Productivity.

Figura 2. Adaptado de (Hiekata et al., 2019)

A figura 3 mostra o diagrama de fluxo de estoque para descoberta de requisitos e desempenho de saída. Este foi projetado com base no ciclo de retrabalho. No diagrama, três partes rotuladas em cores diferentes mostram o fluxo de estoque para descoberta de requisitos, ciclo de retrabalho e conclusão de requisitos. Na parte superior do diagrama, foram utilizados estoques para descoberta de requisitos. Inicialmente, foi assumido que a maioria dos requisitos não foram descobertos e, depois de descobri-los, eles foram categorizados como requisitos flexíveis e requisitos rígidos. As taxas de endurecimento de requisitos e suavização de requisitos categorizam os requisitos como requisitos rígidos e requisitos flexíveis. As taxas de solicitação de recursos e remoção de requisitos definem a mudança de escopo. Depois de descobrir os requisitos, todos eles vão para o trabalho de estoque para fazer no ciclo de retrabalho que é mostrado na parte inferior do diagrama. Na parte central do diagrama, a conclusão dos requisitos é mostrada usando os requisitos rígidos de estoque atendidos e os requisitos flexíveis atendidos. A taxa de estouro entre esses dois estoques define a conclusão das necessidades temporárias. E o desempenho geral de um projeto depende da conclusão de requisitos rígidos e flexíveis [4].

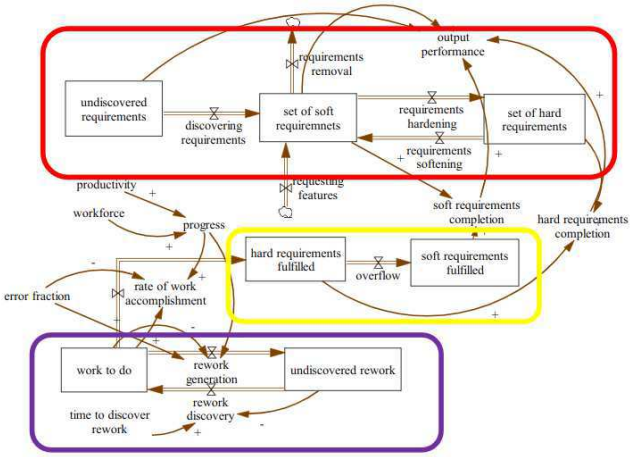

Figure 4. Stock-Flow Diagram for requirements discovery and output performance.

Figura 3. Adaptado de (Hiekata et al., 2019)

Os antecedentes considerados neste trabalho possuem impacto direto sobre a necessidade de mudança de escopo. Foi observado

por meio deste trabalho que a mudança de escopo pode influenciar negativamente no prazo final da conclusão do projeto, mas que o prazo pode chegar a ser cumprido mesmo com a mudança de escopo se a quantidade de pressão do cronograma e horas extras for ideal. Como trabalho futuro, os autores poderiam investigar, por meio da simulação com dados de diferentes projetos, o que é necessário para se definir uma quantidade ideal de pressão no cronograma e horas extras em um projeto de software.

O artigo (van Oorschot et al., 2018) possui como antecedente o ***comprimento das iterações*** em desenvolvimento ágil, o objetivo do trabalho é entender como o tamanho das iterações afeta a pressão do cronograma e consequentemente como essa pressão influencia na ***qualidade, custo e duração do projeto***.

O diagrama criado para a realização de simulação neste trabalho se trata da evolução de um modelo amplamente aceito e utilizado em diversos trabalhos da academia. "O modelo original, desenvolvido por Abdel-Hamid e Madnick (1991), baseado em entrevistas de campo de gerentes de projeto de software em cinco organizações, foi complementado por um extenso banco de dados de descobertas empíricas da literatura." [9]. O modelo deste trabalho foi desenvolvido através do software *iThink*.

A principal contribuição do estudo foi desenvolver um modelo de *feedback* que combina o aspecto técnico do desenvolvimento ágil com os fatores humanos subjacentes a esse desenvolvimento [9]. Os resultados do estudo mostram que as melhores durações de iteração estão mais próximas de 43 a 65 dias úteis do que do ciclo mensal de 20 dias úteis normalmente usado no campo. Ao contrário da prática generalizada, não existe um comprimento de iterações de "tamanho único" [9]. O estudo possui algumas limitações, como, por exemplo, considerar implicitamente que os clientes mudam a funcionalidade previamente definida, mas não adicionam nova funcionalidade ao longo do tempo, o que não é o que acontece no mundo real [9]. A evolução do modelo para lidar com a adição de novos requisitos é uma sugestão de trabalho futuro.

O artigo (Zhang et al., 2018) possui como antecedente as ***mudanças no projeto*** e tem como principal objetivo avaliar o impacto dessas mudanças na ***mão de obra, produtividade, custo, duração e qualidade do projeto***.

Como o modelo precisaria lidar com diversos fatores para apresentar uma visão robusta do projeto, com o intuito de reduzir a complexidade o modelo foi segmentado em 5 subsistemas: *Community subsystem, Issue traking subsystem, Issue resolving subsystem, Quality assurance subsystem* e *Schedule control subsystem*. Os diagramas foram criados por meio da ferramenta de software *Vensim PLE*.

Uma limitação encontrada no estudo é que o mesmo foi realizado com dados do projeto de código aberto do Spring Framework e não foi possível haver uma revisão por parte do da equipe do projeto de código aberto. Outra limitação encontrada é que dados aleatórios são utilizados na simulação de melhorias do processo de software. Fica como sugestão de trabalho futuro a simulação dos processos utilizando dados reais de mais de um projeto para observação do comportamento do modelo nas diferentes situações.

O artigo (Fatema and Sakib, 2017) possui como antecedente o ***gerenciamento da equipe, eficácia do time, motivação, dependência externa, habilidade, cultura e fatores externos*** e tem como principal objetivo avaliar o impacto destes fatores na ***produtividade da equipe ágil***. Diversos fatores foram considerados neste trabalho, eles foram extraídos por meio de entrevistas em empresas e revisão da literatura. Estes fatores passaram para uma avaliação qualitativa até que se chegasse nos mais relevantes aqui apresentados. O modelo SD criado pelos autores se aplica no contexto de metodologia ágil. Como limitação do trabalho, tem-se que apenas o Diagrama de Loop Causal foi desenvolvido, sendo a criação do modelo de estoque e fluxo uma sugestão para um trabalho futuro.

O próximo artigo incluído foi o (Mishra and Mahanty, 2016). O trabalho possui como antecedentes: ***divisão do trabalho local e offshore***; ***esforço de treinamento de equipe local e offshore***. E como consequentes: ***custo do projeto; duração do projeto; qualidade do software***. Esse estudo considera o Desenvolvimento de Software Global (GSD) e é aplicável para diferentes metodologias de desenvolvimento, não é específico para metodologia ágil como muitos dos que foram vistos até então. Foi observado neste estudo que o desenvolvimento *offshore* pode ser útil para diminuir o custo do projeto mas que se perde em produtividade. Quanto mais desenvolvedores *offshore* existirem, maior será *a* sobrecarga de comunicação e adaptação, diminuindo a produtividade e consequentemente alargando o cronograma.

Trata-se de um estudo inovador por não existirem muitas iniciativas buscando modelar o desenvolvimento global. Como limitação, tem-se que o modelo não considera vários tipos de conhecimento de negócios. Como trabalho futuro, propõe-se a extensão do trabalho para encontrar diferentes estruturas de equipe (com base na experiência) necessárias no local e *offshore* para executar projetos de desenvolvimento em GSD.

6. CONCLUSÕES

Concluímos que recentemente foram produzidos diversos modelos em SD para projetos de desenvolvimento de software. Estes podem ser utilizados pelos gerentes para simular diferentes situações e auxiliá-los nas tomadas de decisões. Também foi possível notar que, em modelos de SD para gerenciamento de projetos de software, é comum que haja reaproveitamento de modelos existentes para evolução dos mesmos em trabalhos posteriores.

Não existe uma padronização nos nomes dos fatores dos artigos selecionados até então. Como a inclusão de poucos artigos foi feita, não chegou a haver repetição de fatores, sendo assim, ainda não é possível indicar quais os fatores mais comuns já existentes. Todos menos um dos trabalhos aceitos utilizavam a abordagem ágil, isso pode ter acontecido pelo fato da análise dos artigos ter sido feita iniciando do mais recente.

Limitações deste estudo acontecem, primeiro, por não ter sido possível avaliar todos os artigos selecionados da base WoS para leitura completa; e, segundo, pelo uso apenas da base WoS para as buscas.

As principais contribuições deste trabalho foram o mapeamento dos 6 mais recentes modelos de SD para gerenciamento em projeto de software da base WoS, assim como apresentar os fatores influenciadores que foram utilizados para criação dos mesmos, antecedentes e consequentes.

Como trabalho futuro sugere-se utilizar outros indexadores de artigos científicos de qualidade, tais como ACM Digital Library, IEEE, Scopus, Spring Science. Além da finalização da avaliação dos trabalhos coletados na base WoS. Propõe-se, também, uma classificação dos fatores já mapeados, parecidos com arquétipos, ou padrões, a efeito do que é realizado em orientação a objetos, um catálogo dos fatores e suas melhores práticas, específicos para

projetos de software.

AGRADECIMENTOS

Este trabalho só foi possível por causa do esforço de muitas mãos, e eu não poderia deixar de dedicar um espaço para demonstrar minha gratidão. Minha gratidão a: mãe e pai, por me incentivarem e oferecerem o suporte que precisei; Igor, por me apoiar insistentemente e me ajudar a não desistir nas muitas vezes em que quis fazer isso; meus amigos e família, pelo suporte e orações; Giseldo, pela preciosa mentoria sem a qual eu não conseguiria desenvolver este trabalho; meu orientador, Antão, por toda a paciência e generosidade; e, por fim, pequena Alice, por adicionar aventura e muita alegria durante a jornada.

REFERÊNCIAS

[1] Abdel-Hamid, T. K. (1984). The dynamics of software development project management an integrative system dynamics perspective. PhD thesis, Massachusetts Institute of Technology.

[2] Fatema, I. and Sakib, K. (2017). Factors influencing productivity of agile software development teamwork: A qualitative system dynamics approach. In 24th Asia-Pacific Software Engineering Conference (APSEC), Asia-Pacific Software Engineering Conference, pages 737–742, NEW YORK. Ieee.

[3] Forrester, J. (1961). Industrial dynamics. Cambridge, Mass. [u.a.]: Productivity Press , Wiley.

[4] Hiekata, K., Khatun, M. T., and Chavy-Macdonald, M. A. (2019). System Dynamics Modeling to Manage Performance Based on Scope Change for Software Development Projects. In 26th ISTE International Conference on Transdisciplinary Engineering, volume 10 of Advances in Transdisciplinary Engineering, pages 675–684, AMSTERDAM. Ios Press.

[5] Kitchenham, B. A. and Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. Technical Report EBSE 2007-001, Keele University and Durham University Joint Report.

[6] Lehman, M. M. and Ramil, J. F. (1999). The impact of feedback in the global software process. Journal of Systems and Software, 46(2-3):123–134.

[7] Mishra, D. and Mahanty, B. (2016). A study of software development project cost, schedule and quality by outsourcing to low cost destination. Journal of Enterprise Information Management, 29(3):454–478.

[8] Petersen, K., Feldt, R., Mujtaba, S., and Mattsson, M. (2008). Systematic mapping studies in software engineering. Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering, 17.

[9] Pietron, R. (2020). Scrum project management dynamics simulation. In 40th Anniversary International Conference on Information Systems Architecture and Technology (ISAT), volume 1052 of Advances in Intelligent Systems and Computing, pages 232–243, CHAM. Springer International Publishing Ag.

[10] van Oorschot, K. E., Sengupta, K., and Van Wassenhove, L. N. (2018). Under pressure: The effects of iteration lengths on agile software development performance. Project Management Journal, 49(6):78–102.

[11] Wikipedia (2022). Web of science. url-https://pt.wikipedia.org/wiki/Web of Science

[12] Xie, M. L., Li, C. D., Chen, J., and Ieee (2008). System dynamics simulation to support decision making in software development project. In 4th International Conference on Wireless Communications, Networking and Mobile Computing, International Conference on Wireless Communications Networking and Mobile Computing-WiCOM, pages 7722–+, NEW YORK. Ieee

[13] Zhang, X., Wang, X., and Kang, Y. N. (2018). Change-oriented open source software process simulation. Ieee Access, 6:70145–70163.