

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SÃO PAULO**
Departamento de Tecnologia da Informação

DENIS BRUNO VIRÍSSIMO

**Desenvolvimento de software científico: análise dos fatores e
aspectos em um ambiente de pesquisa do Estado de São Paulo**

**São Paulo
2018**

Denis Bruno Viríssimo

Desenvolvimento de software científico: análise dos fatores e aspectos
em um ambiente de pesquisa do Estado de São Paulo

Trabalho de conclusão apresentado ao Instituto Federal de São Paulo - IFSP, como requisito parcial para conclusão do curso de Especialização em Gestão da Tecnologia da Informação.

Orientador: Prof. Dr. Domingos Bernardo Gomes Santos

São Paulo
2018

Catalogação na fonte
Biblioteca Francisco Montojos - IFSP Campus São Paulo
Dados fornecidos pelo(a) autor(a)

V813d	<p>Viríssimo, Denis Bruno</p> <p>Desenvolvimento de software científico: análise dos fatores e aspectos em um ambiente de pesquisa do estado de são paulo / Denis Bruno Viríssimo. São Paulo: [s.n.], 2018.</p> <p>72 f. il.</p> <p>Orientador: Prof. Dr. Domingos Bernardo Gomes Santos</p> <p>() - Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, IFSP, 2018.</p> <p>1. Software Científico. 2. Processo de Desenvolvimento de Software. 3. Desenvolvimento de Software. I. Instituto Federal de Educação, Ciência e Tecnologia de São Paulo II. Título.</p>
-------	--

CDD

Denis Bruno Viríssimo

Desenvolvimento de software científico: análise dos fatores e aspectos em um ambiente de pesquisa do Estado de São Paulo

Trabalho de conclusão apresentado ao Instituto Federal de São Paulo - IFSP, como requisito parcial para conclusão do curso de Especialização em Gestão da Tecnologia da Informação.

Data da aprovação: 11 / 08 / 2018

Prof. Dr. Domingos Bernardo Gomes Santos
(Orientador)
IFSP – Instituto Federal de São Paulo

Membros da Banca Examinadora:

Prof. Dr. Domingos Bernardo Gomes Santos (Orientador)
IFSP – Instituto Federal de São Paulo

Prof. Ivan Francolin Martinez (Membro)
IFSP – Instituto Federal de São Paulo

Prof. Me. Daniel Marques Gomes de Moraes (Membro)
IFSP – Instituto Federal de São Paulo

AGRADECIMENTOS

À Deus pela vida e pela saúde.

Ao meu orientador, Prof. Dr. Domingos Bernardo Gomes Santos, pelos ensinamentos, disponibilidade, dedicação e ajuda durante a elaboração do trabalho.

Aos meus pais, Jorge e Lilia, por tudo aquilo que me possibilitou chegar até aqui.

À minha esposa, Silvia, pelo imenso amor, apoio, compreensão e paciência.

À todos os professores do Curso, pelo conhecimento obtido durante as aulas e pelas contribuições para a presente pesquisa.

Aos colegas do Curso, em especial Andreia, Leandro e Orlando, pelo apoio e pelas discussões que colaboraram na construção do trabalho.

À todos que, de alguma forma, contribuíram para a realização deste trabalho.

RESUMO

O desenvolvimento de software científico tem chamado a atenção dos pesquisadores da comunidade de desenvolvimento de software. As características intrínsecas ao ambiente de pesquisa e aos cientistas que utilizam estes softwares colocam diversos problemas no processo de desenvolvimento como, por exemplo, a dificuldade na especificação dos requisitos e na garantia da confiabilidade nos resultados fornecidos por tais programas. Alguns estudos vêm buscando identificar e evidenciar os principais aspectos e fatores presentes no ambiente de pesquisa que, de alguma forma, influenciam no processo de desenvolvimento de software científico. Dessa forma, este trabalho tem como objetivo compreender e analisar como estes aspectos e fatores podem afetar o processo de desenvolvimento. Para alcançar este objetivo, foi realizada a aplicação de um estudo de caso múltiplo, de caráter exploratório, por meio de entrevistas com pesquisadores diretamente envolvidos com o desenvolvimento destes softwares, aplicados em quatro áreas de conhecimento distintas. Os resultados obtidos revelam que aspectos do ambiente científico como o objetivo da pesquisa e a pressão para a publicação são aqueles que apresentam a maior influência sobre o processo de desenvolvimento, afetando, principalmente, as atividades de requisitos, projeto e implementação. Além disso, foi percebido que o prévio conhecimento em áreas correlatas à Tecnologia da Informação não é fator preponderante para o desenvolvimento de um software científico que esteja sob a influência destes fatores.

Palavras-chave: Software científico, processo de desenvolvimento de software, desenvolvimento de software.

ABSTRACT

The development of scientific software has caught the attention of researchers in the software development community. The intrinsic characteristics of the research environment and the scientists who use this software pose several problems in the software development process, such as the difficulty in specifying the requirements and in ensuring the reliability of the results provided by such programs. Some studies have sought to identify and highlight the main aspects and factors present in the research environment that, in some way, may influence the process of scientific software development. Thus, this work aims to understand and analyze how these aspects and factors can affect the development process. To achieve this goal, it was undertaken a multiple exploratory case study, through interviews with researchers directly involved in the development of these software, applied in four different areas of knowledge. The results show that aspects of the scientific environment such as the research goal and the publishing pressure are those that have the greatest influence on the development process, mainly affecting the activities of requirements, design and implementation. Furthermore, it was found that previous knowledge in areas related to Information Technology is not a preponderant factor for the development of scientific software that is under the influence of these factors.

Keywords: Scientific software, software development process, software development.

Lista de ilustrações

Figura 1	Pilares de sustentação da ciência	11
Figura 2	Processo geral de desenvolvimento de software científico	18
Figura 3	Modelo simplificado de reputação científica	22
Figura 4	Processo de desenvolvimento - Grupo de Pesquisa A	34
Figura 5	Processo de desenvolvimento - Grupo de Pesquisa B	40
Figura 6	Processo de desenvolvimento - Grupo de Pesquisa C	45
Figura 7	Processo de desenvolvimento - Grupo de Pesquisa D	49

Lista de quadros

Quadro 1	Aspectos e fatores do ambiente de software científico	24
Quadro 2	Métodos de Pesquisa e situações de aplicação	26
Quadro 3	Etapas da execução da pesquisa	28
Quadro 4	Impacto dos Objetivos da pesquisa	54
Quadro 5	Impacto do Histórico de pesquisa	56
Quadro 6	Impacto da Organização e obtenção dos dados	57
Quadro 7	Impacto do Progresso da pesquisa	58
Quadro 8	Impacto da Pressão para publicação	59
Quadro 9	Impacto do Reconhecimento do trabalho de desenvolvimento	60

SUMÁRIO

1. INTRODUÇÃO	6
1.1. Objetivo	7
1.2. Justificativa	7
1.3. Método de Trabalho	8
1.4. Organização do Trabalho	9
2. DESENVOLVIMENTO DE SOFTWARE CIENTÍFICO	11
2.1. Software científico	11
2.2. Características do software científico	14
2.3. Processo geral de desenvolvimento	16
2.4. Perfil e características dos desenvolvedores	20
2.5. Importância do software científico	21
2.6. Fatores e aspectos do ambiente	23
3. METODOLOGIA DA PESQUISA	26
3.1. Estratégia de Pesquisa	26
3.2. Seleção dos casos	29
3.3. Coleta de dados	30
3.4. Tratamento dos dados	31
4. ESTUDO DE CASO: DESENVOLVIMENTO DE SOFTWARE CIENTÍFICO	33
4.1. Grupo de Pesquisa A	33
4.1.1. Características e tipo de pesquisa	33
4.1.2. Perfil do pesquisador	34
4.1.3. Processo de desenvolvimento	34
4.1.4. Impacto dos aspectos	35
4.1.4.1. Objetivo da Pesquisa	35
4.1.4.2. Disponibilidade do desenvolvedor	36
4.1.4.3. Histórico de Pesquisa	36
4.1.4.4. Organização e obtenção dos dados	37
4.1.4.5. Progresso da pesquisa	37
4.1.4.6. Pressão para publicação	38
4.1.4.7. Reconhecimento do trabalho de desenvolvimento	38
4.2. Grupo de Pesquisa B	39
4.2.1. Características e tipo de pesquisa	39
4.2.2. Perfil do pesquisador	40

4.2.3.	Processo de desenvolvimento	40
4.2.4.	Impacto dos aspectos	41
4.2.4.1.	Objetivo da Pesquisa	41
4.2.4.2.	Disponibilidade do desenvolvedor	41
4.2.4.3.	Histórico de Pesquisa	42
4.2.4.4.	Organização e obtenção dos dados	42
4.2.4.5.	Progresso da pesquisa	43
4.2.4.6.	Pressão para publicação	43
4.2.4.7.	Reconhecimento do trabalho de desenvolvimento	43
4.3.	Grupo de Pesquisa C	44
4.3.1.	Características e tipo de pesquisa	44
4.3.2.	Perfil do pesquisador	44
4.3.3.	Processo de desenvolvimento	45
4.3.4.	Impacto dos aspectos	45
4.3.4.1.	Objetivo da Pesquisa	46
4.3.4.2.	Disponibilidade do desenvolvedor	46
4.3.4.3.	Histórico de Pesquisa	46
4.3.4.4.	Organização e obtenção dos dados	47
4.3.4.5.	Progresso da pesquisa	47
4.3.4.6.	Pressão para publicação	48
4.3.4.7.	Reconhecimento do trabalho de desenvolvimento	48
4.4.	Grupo de Pesquisa D	48
4.4.1.	Características e tipo de pesquisa	48
4.4.2.	Perfil do pesquisador	49
4.4.3.	Processo de desenvolvimento	49
4.4.4.	Impacto dos aspectos	50
4.4.4.1.	Objetivo da Pesquisa	50
4.4.4.2.	Disponibilidade do desenvolvedor	50
4.4.4.3.	Histórico de Pesquisa	51
4.4.4.4.	Organização e obtenção dos dados	51
4.4.4.5.	Progresso da pesquisa	52
4.4.4.6.	Pressão para publicação	52
4.4.4.7.	Reconhecimento do trabalho de desenvolvimento	52
4.5.	Análise comparativa dos resultados	53
4.5.1.	Objetivo da Pesquisa	53
4.5.2.	Disponibilidade do desenvolvedor	55
4.5.3.	Histórico de Pesquisa	55

4.5.4.	Organização e obtenção dos dados	57
4.5.5.	Progresso da pesquisa	58
4.5.6.	Pressão para publicação	58
4.5.7.	Reconhecimento do trabalho de desenvolvimento	60
4.5.8.	Aspectos complementares	61
5.	Considerações finais	62
5.1.	Trabalhos futuros	63
	REFERÊNCIAS	65
	APÊNDICE A – E-mail de Apresentação	71
	APÊNDICE B – Roteiro de Perguntas	72

1.INTRODUÇÃO

O desenvolvimento de softwares científicos tem se tornado um importante tema de pesquisa nas publicações científicas (FARHOODI et al., 2013). Isto se deve ao fato de que os softwares científicos são, cada dia mais, instrumentos fundamentais para o avanço da ciência (MORRIS; SEGAL, 2009). Cientistas tais como meteorologistas, biólogos, químicos e engenheiros, para citar apenas alguns deles, dependem dessas ferramentas para realizarem suas pesquisas envolvendo reação a eventos quase que em tempo real, fenômenos que ocorrem a um ritmo muito vagaroso e fenômenos que não podem ser observados em condições naturais (CRABTREE et al., 2009; HEATON; CARVER, 2015; MAXVILLE, 2009).

Esses tipos de pesquisa dependem altamente dos resultados obtidos pelos softwares científicos. Devido a esta dependência é crucial que esses programas forneçam resultados precisos, caso contrário, danos irreparáveis à pesquisa científica podem ser causados¹ (MILLER, 2007; SANDERS; KELLY, 2008).

Segundo Kelly (2015), um software científico é aquele composto por um grande componente de conhecimento do domínio de aplicações científicas, usado para o aumento do conhecimento científico com o propósito de resolver problemas do mundo real. Devido a este propósito, o software científico distingue-se do software comercial por ser visto como um meio e não um fim em si mesmo (HEROUX; WILLENBRING, 2009; WAGNER, 2007). Assim, o desenvolvimento de software científico apresenta-se em um contexto diferenciado.

De acordo com Basili et al. (2008), três características principais compõem esse contexto: o conhecimento sobre desenvolvimento de software, o aumento não planejado do tamanho dos projetos e o conjunto de usuários. Essas características, intrínsecas ao ambiente de pesquisa e aos cientistas que utilizam estes softwares, colocam diversos problemas no processo de desenvolvimento de software científico.

Um dos maiores problemas apontados no desenvolvimento de software científico está na falta de especificações definidas. Por serem softwares desenvolvidos, na grande maioria dos casos, pelos próprios cientistas e com uma

¹ Em um caso recente, um erro descoberto em um software de ressonância magnética pode fazer com que 15 anos de pesquisas científicas na área sejam invalidadas (COX; REYNOLDS; TAYLOR, 2016; EKLUND; NICHOLS; KNUTSSON, 2016).

finalidade única, não existe a preocupação em se detalhar, especificar e documentar os requisitos (ACKROYD et al., 2008; SANDERS; KELLY, 2008; SEGAL, 2009).

Além disso, o domínio da pesquisa científica caracteriza-se pela alta taxa de incerteza e mudanças constantes, geradas principalmente pelos resultados das pesquisas realizadas. Isso implica em rápidas mudanças inesperadas nos softwares (ACKROYD et al., 2008). Dessa forma, o ideal é que os softwares sejam desenvolvidos acompanhando o avanço da pesquisa científica (ACKROYD et al., 2008).

1.1. Objetivo

O objetivo deste trabalho é compreender e analisar como os fatores e aspectos do ambiente de pesquisa em que ocorre o desenvolvimento de software científico podem afetar o seu processo de desenvolvimento, traduzido na seguinte questão de pesquisa:

“De que forma os fatores e aspectos do ambiente de pesquisa no qual se desenvolve o software científico podem afetar o respectivo processo de desenvolvimento?”

A identificação dos fatores e aspectos do ambiente de pesquisa será feita por meio da revisão bibliográfica de trabalhos já publicados referentes ao assunto. Por sua vez, a análise do impacto destes fatores no processo de desenvolvimento será realizada por meio de quatro estudos de caso.

Os estudos de caso em questão estão restritos ao desenvolvimento de software científico nas áreas das ciências exatas e engenharias, e limitados à quatro grupos distintos dentro de um ambiente de pesquisa.

1.2. Justificativa

Apesar da grande importância do software científico, o processo de desenvolvimento de software adotado pelos desenvolvedores não costuma seguir processos de desenvolvimento já estabelecidos na literatura (GUO; LYSTER, 2000). O software científico normalmente é desenvolvido pelos próprios pesquisadores, a partir do conhecimento da área de pesquisa, sendo normalmente baseados nas experiências locais dos pesquisadores (ACKROYD et al., 2008; SEGAL; MORRIS, 2008; SLETHOLT et al., 2012).

A maturidade da pesquisa científica e as grandes mudanças que ocorrem no ambiente de pesquisa também influenciam diretamente nos processos de desenvolvimento que são concebidos, gerando diversas diferenças entre os ambientes de pesquisa (SLETHOLT et al., 2012; WILSON et al., 2014). Além disso, a complexidade e a diversidade dos ambientes de pesquisa e das técnicas utilizadas pelos cientistas colocam restrições práticas em projetos de desenvolvimento de software científico (LOYNTON et al., 2009).

Por outro lado, a introdução direta de metodologias, técnicas e práticas de engenharia de software no contexto do desenvolvimento de software científico não tem se mostrado eficaz (CARVER et al., 2007; KELLY, 2007). Para se obter algum sucesso nessa abordagem é necessário compreender o software científico como um sistema de produção independente, examinando o contexto no qual está inserido e os incentivos que fundamentam a sua produção (HOWISON; HERBSLEB, 2010).

Dessa forma, torna-se importante a realização de estudos que demonstrem as características dos processos de desenvolvimento adotados, bem como os fatores e decisões tomadas pelos desenvolvedores, de modo que melhorias do processo possam ser propostas (MESH, 2015). O conhecimento sobre o que direciona e restringe as práticas atuais dos desenvolvedores de software científico forma a base para que intervenções no processo de desenvolvimento possam melhorar a situação atual (HOWISON; HERBSLEB, 2010).

Nos últimos anos, alguns estudos vêm buscando identificar e evidenciar os principais aspectos e fatores presentes no ambiente de pesquisa que, de alguma forma, influenciam no processo de desenvolvimento, dentre os quais se podem destacar os objetivos da pesquisa científica, a pressão para publicação dos resultados e a taxa de avanço da pesquisa científica (MESH, 2015; MESH; HAWKER, 2013). Entretanto, estes estudos ainda não identificaram como, de fato, estes aspectos impactam no processo de desenvolvimento.

1.3. Método de Trabalho

A pesquisa foi planejada para realizar as seguintes atividades:

1. Revisão bibliográfica

Esta atividade tem dois objetivos: em primeiro lugar, compreender os conceitos e definições que fundamentam o software científico; em

segundo lugar, descrever os princípios e características fundamentais do processo de desenvolvimento de software científico.

2. Identificação dos aspectos e fatores

A partir da revisão bibliográfica, são identificados nesta atividade os principais aspectos e fatores que podem impactar e afetar o processo de desenvolvimento de software científico.

3. Definição da metodologia e estratégia de pesquisa

Nesta atividade é definida a metodologia e a estratégia de pesquisa a ser adotada, bem como são definidos os casos selecionados para estudo.

4. Elaboração do instrumento de coleta

Nesta atividade são definidos e elaborados os questionários de entrevista para os participantes do estudo de caso bem como o roteiro das entrevistas realizadas, embasados no referencial bibliográfico estabelecido na atividade 1 e nos aspectos que são estudados.

5. Coleta dos dados

Execução do estudo de caso em si, com a aplicação dos questionários e coleta de dados.

6. Análise dos resultados

Avaliação dos principais resultados encontrados referentes ao impacto dos fatores e aspectos no processo de desenvolvimento de software científico.

1.4. Organização do Trabalho

O Capítulo 2, Desenvolvimento de Software Científico, apresenta uma revisão bibliográfica sobre o software científico, seu processo de desenvolvimento e principais fatores e características do ambiente de pesquisa e problemas associados.

O Capítulo 3, Metodologia da Pesquisa, expõe a metodologia de pesquisa, abordando cada uma das etapas do estudo, e apresentando o desenho de pesquisa em detalhe bem como a construção dos instrumentos de coleta.

No Capítulo 4, Estudo de Caso: desenvolvimento de software científico, apresenta-se o detalhamento dos estudos de casos e o resultado das entrevistas realizadas, bem como a avaliação sobre o impacto dos fatores e aspectos do desenvolvimento de software científico nos casos estudados.

O Capítulo 5, Considerações finais, apresenta a conclusão sobre os principais resultados, contribuições desta pesquisa e sugestões para trabalhos futuros.

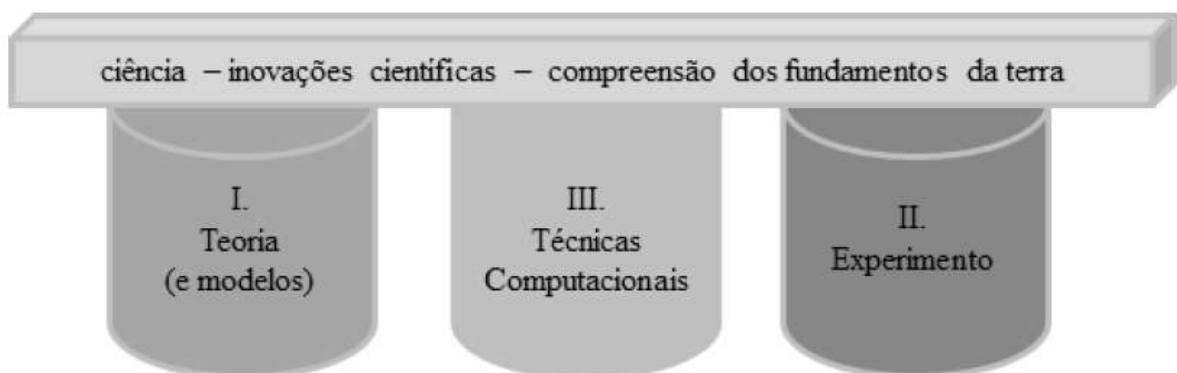
2.DESENVOLVIMENTO DE SOFTWARE CIENTÍFICO

Este capítulo apresenta a revisão bibliográfica sobre software científico e seu processo de desenvolvimento. Primeiramente, são apresentadas as definições de software científico e principais conceitos associados. Em seguida, o processo geral do desenvolvimento destes softwares é descrito, bem como as características que o compõem. Após isso, são destacados os perfis dos desenvolvedores e a importância do software científico. Por fim, são apresentados os principais fatores e aspectos do ambiente de pesquisa no qual ocorre o desenvolvimento destes softwares, os quais norteiam a pesquisa proposta por este trabalho.

2.1. Software científico

A pesquisa científica tem, nas últimas décadas, obtido grandes avanços por meio do uso de recursos tecnológicos e computacionais (RIEDEL et al., 2008). Conforme destacam Riedel et al. (2008) e Prabhu et al. (2011), além dos já conhecidos pilares da teoria (ou modelos específicos em uma determinada linha de pesquisa) e da experimentação, a computação e a tecnologia da informação passaram a ser reconhecidas como o terceiro pilar de sustentação e apoio da ciência (Figura 1), permitindo simulações baseadas na eficiência de métodos numéricos e leis físicas conhecidas (CARVALHO et al., 2006).

Figura 1 - Pilares de sustentação da ciência



Fonte: adaptado de Riedel et al. (2008)

Grande parte destes recursos tecnológicos se manifesta na forma de softwares e programas computacionais, e sua importância é inquestionável. Desde grupos que trabalham exclusivamente com problemas computacionais, até cientistas tradicionais que atuam no laboratório ou em campo, têm em sua atividade diária a operação da ciência em busca do desenvolvimento de novos algoritmos, da análise

e da organização de grandes volumes de dados gerados por projetos de pesquisa individuais, e da combinação de diferentes conjuntos de dados na análise de problemas sintéticos (WILSON et al., 2014).

A aplicação científica destes softwares pode se dar também em campos da ciência aplicada (SANDERS; KELLY, 2008), como, por exemplo, na pesquisa meteorológica e de fenômenos dos oceanos, no estudo e projeto sobre motores de aeronaves, na simulação de carga e capacidade de pontes, na orientação de cirurgias ortopédicas e na simulação de subsistemas para geração de energia nuclear (KELLY, 2009).

Goble (2014) destaca que a pesquisa científica moderna não seria possível sem o uso de softwares, sejam eles planilhas complexas ou grandes softwares robustos como aqueles por trás da operação do Grande Colisor de Hádrons (do inglês, *Large Hadron Collider - LHC*) e do grande telescópio *Square Kilometer Array* (SKA).

Heaton e Carver (2015), por sua vez, destacam que modelos computacionais tem grande importância para a solução de problemas outrora quase impossíveis.

Em primeiro lugar, destacam que estes modelos permitem reagir a eventos em tempo real, como no caso da meteorologia, em que informações da previsão do tempo podem ser obtidas com base nas condições atuais e nos efeitos de potenciais mudanças. Sem modelos computacionais os cientistas teriam de extrapolar os dados históricos, o que consumiria muito tempo e seria inviável para previsões em tempo real (HEATON; CARVER, 2015).

Em segundo lugar, softwares computacionais possibilitam que sejam estudados fenômenos que ocorrem muito lentamente, como aqueles encontrados no campo da geologia, por meio da simulação acelerada destes eventos. Softwares computacionais também criam condições para estudo de fenômenos muito precisos para a observação manual, como aqueles presentes no campo da astronomia e da astrofísica (HEATON; CARVER, 2015).

Por fim, Heaton e Carver (2015) afirmam que o estudo científico em campos de alto risco também são possíveis graças ao uso destes softwares. Na astrofísica, é muito mais segura a reprodução e simulação de experimentos sobre diversos tipos de reações nucleares, do que realizá-los fisicamente.

Apesar da concordância de diversos autores sobre a aplicação e importância destes softwares na pesquisa científica, ainda há certo conflito e ambiguidade na

definição, terminologia e nomenclatura usada para descrever esses tipos de softwares (HEROUX; WILLENBRING; PHENOW, 2007). Durante a pesquisa bibliográfica, este trabalho encontrou diversas formas para se referir a estes softwares e diversas maneiras de lhes descrever, sendo o termo “software científico” o mais utilizado.

Nos trabalhos de Heroux, Willenbring e Phenow (2007), Hannay et al. (2009), Farhoodi et al. (2013), Mesh e Hawker (2013), Heaton e Carver (2015) e Kelly (2015), por exemplo, é utilizado o termo “software científico”. Já nos trabalhos de Smith (2006) e de Li et al. (2011), utiliza-se o termo “computação científica”. Heroux e Willenbring (2009) também empregam o termo “software de pesquisa”.

Heroux, Willenbring e Phenow (2007) definem software científico como a aplicação de software (normalmente de modelagem) que fornece dados para apoiar diretamente as decisões científicas e que permitem ao cientista examinar uma situação computacionalmente. Sletholt et al. (2011), por sua vez, o definem como software usado para simulações e cálculos complexos, por vezes usado para se testar uma teoria científica.

Em outra definição, Farhoodi et al. (2013) referem-se ao software científico como aquele que é desenvolvido por cientistas e para cientistas e que implementa algoritmos complexos, como aqueles utilizados para resolver sistemas de equações matemáticas ou para fornecer simulações para físicos. Mesh e Hawker (2013) estabelecem uma visão mais geral, dizendo que o software científico é todo aquele desenvolvido para apoiar a pesquisa científica, independentemente das necessidades de recursos para processamento ou de dados. Heaton e Carver (2015) adotam uma definição mais simplista, dizendo que softwares científicos são aqueles criados por cientistas e engenheiros.

No espectro daqueles que adotam o termo “computação científica”, a definição parece seguir uma linha de raciocínio consonante à adotada por Heaton e Carver (2015), mas com um enfoque maior à matemática. Li et al. (2011) descrevem a computação científica como o projeto e análise de abordagens para a resolução de problemas matemáticos nos campos da ciência e da engenharia. Já Smith (2006) adota a definição de uso de ferramentas de computador para analisar ou simular modelos matemáticos de sistemas de engenharia ou de importância científica do mundo real, de modo que seja possível entender e prever o comportamento desses sistemas.

Dentre todas as definições adotadas e utilizadas pelos autores, aquela que parece possuir uma visão mais ampla e que busca contextualizar de modo mais holístico o software científico se encontra no trabalho de Kelly (2015).

Kelly (2015) define o software científico como um software de aplicação que possui um grande componente de conhecimento do domínio da aplicação científica e é usado para enriquecer o conhecimento da ciência, com o propósito de resolver problemas do mundo real, incluindo também os softwares utilizados para as engenharias. Kelly (2015) exemplifica o software científico como aquele que inclui softwares para modelar a carga em pontes, estudar a operação segura de usinas nucleares, rastrear caminhos de furacões, localizar planetas e satélites em imagens de telescópios, procurar falhas em rochas de minas, modelar procedimentos médicos para tratamento de câncer, modelar padrões de dispersão para partículas tóxicas e estudar o impacto ecológico das correntes oceânicas.

Além disso, Kelly (2015) exclui da definição as ferramentas genéricas utilizadas pelos cientistas, como bibliotecas matemáticas e softwares que abstraem a complexidade de ambientes de computação de alto desempenho. Porém, considera qualquer aplicação desenvolvida com base nessas ferramentas, desde que tenha o propósito de resolver um determinado problema científico.

Neste trabalho, o termo a ser adotado será “software científico”, e a definição utilizada para se referir a estes softwares é aquela proposta por Kelly (2015), incluindo os softwares desenvolvidos para o campo das engenharias, conforme reforçado por Heaton e Carver (2015).

2.2. Características do software científico

Conforme analisado, existem diversas definições estabelecidas para o software científico. Isso também se repete ao se analisar as características destes softwares.

Mesh e Hawker (2013) agrupam em um conjunto de três características principais aquelas que definem, em sua visão, o software científico:

1. O propósito fundamental do produto de software é apoiar a exploração da questão científica da pesquisa, e não responde-la diretamente;
2. O produto de software é desenvolvido pelos próprios especialistas do domínio (pesquisadores) ou por interações junto a eles; e

3. O software científico carece de falta de resultados conhecidos e saídas esperadas, fazendo com que a verificação e a validação destes softwares sejam difíceis.

As duas primeiras características apresentadas por Mesh e Hawker (2013) também são reforçadas por Ahmed e Zeeshan (2014) e Neumann, Espina e Daza (2017), que destacam também que o software científico deve ser projetado e finalizado antes da pesquisa científica em si, já que ele é considerado um meio para se obter os resultados necessários. Geralmente, tendo sido coletados os resultados de um experimento ou simulação, o software científico é abandonado e um novo software é desenvolvido para o próximo experimento, apesar da existência de similaridades com o software anterior (MONTEITH; MCGREGOR; INGRAM, 2014).

Já Men, Kelly e Dean (2011) abordam as características do software científico por um viés mais técnico. Definem que geralmente:

- Os softwares desenvolvidos por cientistas utilizam linguagens de programação como C, C++ e Fortran, ou linguagens específicas do domínio científico, como MATLAB e R;
- Os softwares são orientados à dados, extraídos a partir de medições do mundo real ou fornecidos pelos próprios pesquisadores;
- As entradas dos softwares são usadas para analisar, simular, prever, visualizar e realizar atividades científicas;
- Os dados de entrada são transformados com base em modelos matemáticos, que podem ser modelos com aproximações e restrições de implementação.

Outra característica fundamental dos softwares científicos é que são desenvolvidos como prova de conceito e costumam gerar resultados experimentais, que tendem a ser refinados em interações. O desenvolvimento é realizado por cientistas e pesquisadores altamente qualificados, e não engenheiros de software (HEROUX; WILLENBRING, 2009; MONTEITH; MCGREGOR; INGRAM, 2014).

Kelly (2015), por sua vez, caracteriza o software científico da seguinte forma:

- Um especialista do domínio científico está necessariamente envolvido no processo de desenvolvimento do software;

- O usuário deste tipo de software possui um conhecimento mínimo do domínio científico, de modo que ele possa interpretar corretamente os resultados fornecidos pelo software;
- O usuário recebe toda a informação processada pelo software, de modo que estes tipos de aplicações não são utilizados para controlar equipamentos;
- O propósito principal do software é fornecer dados para a compreensão e entendimento de problemas específicos do mundo real; e
- A definição fundamental para qualidade do software é a exatidão (ou mais precisamente, a fidedignidade) dos resultados; se o software não é fidedigno, todas as demais características são irrelevantes.

2.3. Processo geral de desenvolvimento

O software científico é comumente desenvolvido pelos próprios pesquisadores, a partir do conhecimento da área de pesquisa e visando a obtenção dos resultados necessários para sua pesquisa científica (ACKROYD et al., 2008; SEGAL; MORRIS, 2008).

Entretanto, o processo de desenvolvimento de software adotado por esses pesquisadores não costuma seguir processos de desenvolvimento já estabelecidos na literatura. Segundo Guo e Lyster (2000), os processos de desenvolvimento para software científico são caracterizados frequentemente como *ad hoc*, senão caóticos.

Os métodos utilizados para desenvolver o software que emergem no ambiente de pesquisa normalmente são baseados nas experiências locais dos pesquisadores (SLETHOLT et al., 2012). Segundo estudo de Wilson et al. (2012), mais de 90% dos pesquisadores que desenvolvem software são autodidatas.

Além disso, as grandes mudanças que ocorrem no domínio, a maturidade da pesquisa científica e a motivação no desenvolvimento de software científico influenciam diretamente nos processos de desenvolvimento que são concebidos, o que gera diversas diferenças nos ambientes de pesquisa (SLETHOLT et al., 2012; WILSON et al., 2014).

Não obstante, alguns estudos tem sido capazes de identificar um processo geral de desenvolvimento de software científico, com passos e etapas que a grande maioria dos pesquisadores costuma adotar.

Um destes trabalhos indica que desenvolvedores de software científico costumam adotar um procedimento de cinco passos (SMITH; LAI, 2005):

1. Definir o problema;
2. Criar um modelo matemático com base em hipóteses adequadas;
3. Identificar um método computacional;
4. Refinar e implementar uma solução; e
5. Validar a solução.

O trabalho de Nanthaamornphong et al. (2013) não apresenta um processo específico de desenvolvimento, mas relata uma série de práticas adotadas pelos desenvolvedores durante a construção do software e que comumente são percebidas como métodos ágeis de desenvolvimento, dentre as quais se podem destacar:

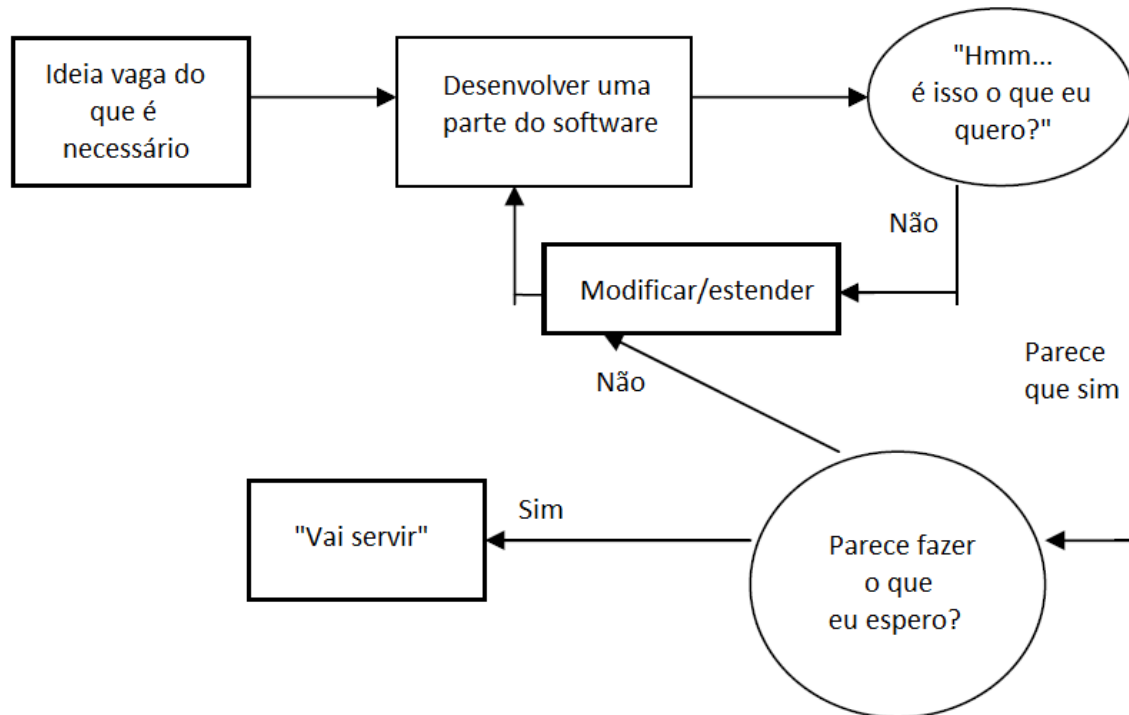
- Realização de uma série de atividades, como execução de experimentos, análise e comparação dos resultados;
- Iteração no processo de desenvolvimento, com curto prazo e tempo limitado no cronograma;
- Inclusão de novas funcionalidades de acordo com novas descobertas; e
- Flexibilização para realizar experimentos sempre que possível.

Em uma linha de pesquisa similar, o trabalho de Heaton e Carver (2015) categoriza várias práticas utilizadas pelos desenvolvedores. Os autores traçam um paralelo entre as práticas encontradas na literatura e as atividades comumente encontradas no ciclo de desenvolvimento de software, a saber: requisitos, projeto, implementação, testes, refatoração e documentação. Nenhum processo específico de desenvolvimento de software é citado, mas o trabalho sugere que as tarefas realizadas pelos desenvolvedores de software científico podem ser enquadradas neste conjunto de atividades, de modo similar ao que pode ser encontrado nos trabalhos de Farhoodi et al. (2013) e de Ahmed e Zeeshan (2014).

A abordagem comumente encontrada na literatura para processo de desenvolvimento de software científico, e que também é indicada e referenciada por vários autores, é aquela apresentada na Figura 2. Esta abordagem possui muita similaridade com a abordagem iterativa e incremental de desenvolvimento de software.

Segundo Segal (2009), o processo se inicia com uma ideia básica sobre o que precisa ser desenvolvido (requisitos), mas com um alto grau de entendimento sobre o domínio e sobre que resultados se deseja obter. Isto é necessário tanto para o início do processo como para a sua conclusão.

Figura 2 - Processo geral de desenvolvimento de software científico



Fonte: adaptado de Segal (2009)

Em seguida, o desenvolvedor avalia informalmente o software desenvolvido, a partir de questões do tipo “Este software faz o que eu quero?” e “Ainda pode ser estendido?” (HEATON; CARVER, 2015).

Após isso, o desenvolvedor modifica ou estende o código desenvolvido até que as respostas para as questões anteriormente citadas sejam, respectivamente, “sim” e “não” (HEATON; CARVER, 2015).

Por fim, o desenvolvedor “testa” o software questionando se “a saída do programa era aquilo que se esperava?”. Quando a resposta para esta questão é positiva, o desenvolvedor considera concluído o projeto do software (HEATON; CARVER, 2015).

Apesar de ser prática corriqueira e que tem dado resultado até o presente momento, este tipo de processo de desenvolvimento de software, associado à natureza da pesquisa científica, coloca diversos problemas (ACKROYD et al., 2008; SANDERS; KELLY, 2008; SEGAL, 2009; SLETHOLT et al., 2011).

Em primeiro lugar, tem-se o problema da elicitación e especificación de requisitos. Dada a natureza exploratória de vários projetos científicos, a elicitación e especificación de requisitos são problemáticas, já que os requisitos podem ser pouco claros ou desconhecidos. Na verdade, muitas vezes os requisitos só poderão ser completamente compreendidos quando o software já estiver quase concluído (SLETHOLT et al., 2011).

Como consequência, os requisitos do software científico não são coletados e analisados de forma sistemática, tornando a documentação inexistente ou incompleta (SMITH; LAI, 2005). Além disso, como o ambiente de pesquisa tende a ser muito dinâmico e altamente volátil e, por sua vez, também os requisitos, torna-se altamente difícil a verificação e validação do software frente a estes requisitos (SLETHOLT et al., 2011).

Em segundo lugar, outros problemas surgem das atividades de codificação e testes. Os desenvolvedores não tendem a se atentar para questões como compreensibilidade do código desenvolvido, o que implica em dificuldades para continuar evoluindo e mantendo o software (MORRIS; SEGAL, 2009).

O controle sobre o código é inexistente, fazendo com que seja difícil identificar qual versão do software produziu os resultados utilizados na pesquisa, causando também problemas para replicação dos resultados por outras pesquisas (HEROUX, 2011).

Os testes do código, conforme citado anteriormente, são apenas checagens e verificações a respeito do funcionamento em uma situação particular de uso, o que não garante a geração de resultados adequados quando aplicado em outros contextos (MORRIS; SEGAL, 2009).

A documentação, por sua vez, não costuma receber grande importância (NGUYEN-HOAN; FLINT; SANKARANARAYANA, 2010). Como na grande maioria dos casos o desenvolvedor é o próprio usuário do software, a necessidade de documentação do software acaba por ser afastada, podendo prejudicar a sua manutenção (KELLY, 2011).

Em algumas situações, o software científico tende a ser compartilhado com outros pesquisadores e comunidades de cientistas. Caso o software não tem sido testado, verificado e validado de maneira adequada, um erro pode ser propagado para outras pesquisas sem que seja percebido (WILSON et al., 2014).

Por fim, como o objetivo principal em ambientes de pesquisa não é produzir software, e sim conhecimento (BASILI et al., 2008), os desenvolvedores não se sentem motivados e tendem a dar pouca importância ao método e processo de desenvolvimento, o que pode afetar a qualidade do software desenvolvido (QUEIROZ; SPITZ, 2016; STORER, 2017).

2.4. Perfil e características dos desenvolvedores

Um dos principais componentes do desenvolvimento do software científico é o próprio pesquisador, uma vez que, na grande maioria dos casos, é o responsável por todo o processo de construção do software científico, desde a concepção até a finalização (SEGAL, 2009).

Entretanto, em algumas ocasiões o trabalho de desenvolvimento não está limitado somente ao pesquisador (KELLY, 2007). Diversos trabalhos têm estabelecido diversos tipos de “pesquisadores desenvolvedores” e descrito o perfil de pessoas envolvidas na construção de software científico.

Kelly (2007) aborda a existência de três grupos principais de desenvolvedores:

- Desenvolvedores da indústria e engenharias, que trabalham em seu respectivo domínio de aplicação;
- O pesquisador científico, por vezes atuando na academia; e
- O estudante de engenharia ou cientista que, eventualmente, se juntará a um dos dois outros grupos.

Já no trabalho de Goble (2014), identifica-se dois grupos distintos: o dos desenvolvedores altamente especializados e treinados, que atuam em grupos de pesquisa e normalmente são pessoas contratadas (como pós-doutores); e o dos pesquisadores autodidatas.

Dentro do contexto de pesquisa do presente trabalho, o maior interesse se dá pelos pesquisadores que não possuem conhecimento explícito sobre desenvolvimento de software. Segundo a literatura, esses pesquisadores-desenvolvedores são comumente chamados de “programadores usuários-finais” (KELLY, 2015).

Ko et al. (2011) descrevem o “programador usuário-final” como aquele que desenvolve um software com o intuito principal de obter um programa para uso pessoal. Eles enfatizam que esta definição se refere somente à intenção de

desenvolver o programa, e não está relacionada com a identidade do desenvolvedor, com ferramentas utilizadas e muito menos com a definição de a atividade estar relacionada com propósitos pessoais ou profissionais do indivíduo.

A razão mais óbvia para se caracterizar os cientistas e pesquisadores como “programadores usuários-finais” é que eles mesmos não se consideram no negócio de desenvolvimento de software (KELLY, 2015).

Em seu estudo, Segal (2005) estabelece que os pesquisadores que desenvolvem software científico possuem grande capacidade de trabalhar em ambientes onde o conhecimento é altamente desenvolvido, têm proficiência em linguagens e abstrações, e não apresentam grande dificuldade em codificar e aprender linguagens de programação.

Por fim, os desenvolvedores de software científico não apresentam distinção de papéis e função dentro das equipes de desenvolvimento; não existe separação dos membros por tarefas, como analista, testador, engenheiro de requisitos, etc. Ao invés disso, membros de projeto de software científico são identificados pela sua área de conhecimento, como, por exemplo, especialista em líquidos bifásicos (KELLY, 2015). Conforme já destacado, espera-se que estes desenvolvedores vivenciem todas as atividades do processo de desenvolvimento do software científico (KELLY, 2015).

Apesar da grande capacidade demonstrada por estes pesquisadores, problemas relacionados ao seu reconhecimento são frequentemente relatados.

2.5. Importância do software científico

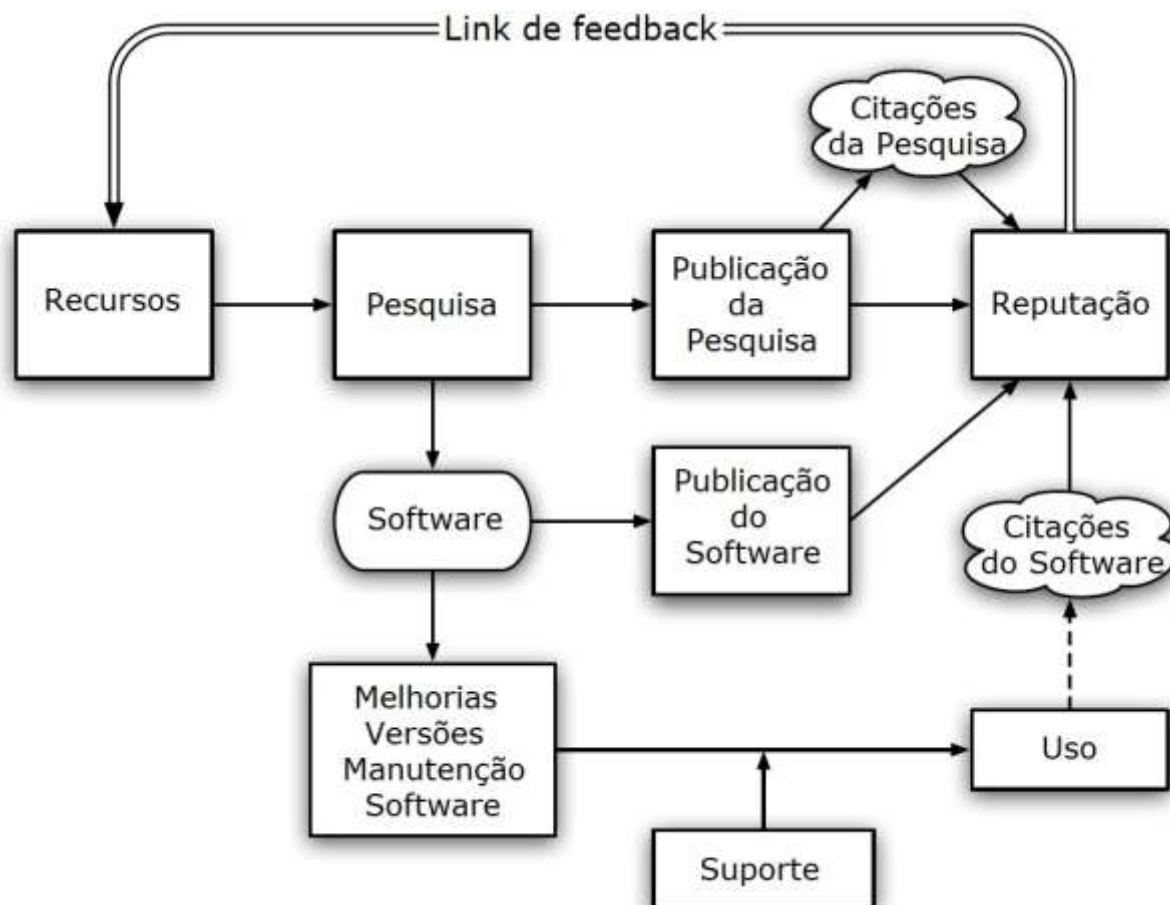
Além da grande importância para o desenvolvimento da ciência, o software científico apresenta também importância reconhecida pelos próprios pesquisadores. Em estudos conduzidos por Hannay et al. (2009), cerca de 91% dos respondentes afirmam que o uso de software científico é importante para sua pesquisa, e 84% disseram que não só o uso, mas também o desenvolvimento é importante.

Entretanto, a comunidade científica e acadêmica não tem dado a devida importância ao software científico, no que diz respeito às publicações científicas e aos próprios desenvolvedores (GOBLE, 2014).

Na ciência, o reconhecimento, a reputação e a recompensa dos pesquisadores advêm da publicação de artigos científicos (HOWISON; HERBSLEB, 2010; QUEIROZ; SPITZ, 2016). A Figura 3 apresenta um modelo simplificado de

reputação científica, identificado por Howison e Herbsleb (2011a) em seus estudos. Para que uma pesquisa científica possa ser realizada, recursos (financeiros ou não) são necessários e, normalmente, esses recursos são fornecidos para pesquisadores e instituições que tem maior reputação. Esta, por sua vez, é obtida por meio de citações às publicações dos pesquisadores.

Figura 3 – Modelo simplificado de reputação científica



Fonte: adaptado de Howison e Herbsleb (2011a)

Entretanto, obter publicações para os softwares desenvolvidos dentro do ambiente científico é extremamente árduo (HOWISON; HERBSLEB, 2011). Ao contrário das publicações científicas, a publicação de software raramente é útil por si só; geralmente o software é que é útil para outros cientistas (HOWISON; HERBSLEB, 2011). Uma das poucas exceções é para pesquisadores que estejam inseridos em áreas como a bioinformática, que possuem periódicos exclusivos para publicação de artigos sobre softwares (GOBLE, 2014).

A parte de baixo do fluxo diz respeito à publicação do software em si. Essas publicações podem trazer alguma reputação para o pesquisador; entretanto é o seu uso que trará a maior parte do reconhecimento sobre a importância daquele

software. No fluxo apresentado, a linha tracejada entre o uso e a citação indica que não há uma correlação direta entre os dois elementos, uma vez que nem todas as publicações da pesquisa citam os softwares utilizados, dificultado assim o reconhecimento do desenvolvedor de software científico (HOWISON; HERBSLEB, 2010).

Outra opção possível para se obter reputação é o compartilhamento do software desenvolvido com outros grupos de pesquisa. Muito provavelmente o pesquisador conseguirá certo reconhecimento destes grupos de pesquisa, pois isto facilitará o trabalho dos demais e ajudará na publicação de novos resultados. Entretanto, dificilmente o desenvolvedor receberá reconhecimento da instituição em que atua (GOBLE, 2014).

Atrelado a tudo isso se associa o fato de que a transição do desenvolvimento de software científico, de meras habilidades ocasionais para uma carreira formal, demanda políticas, práticas e estruturas adicionais que motivem a busca pela excelência deste campo (NEUMANN; ESPINA; DAZA, 2017; TERREL; TOBIS; THIRUVATHUKAL, 2015), o que pode ser dificultado por algumas correntes de pensamento que creem que o desenvolvimento de software não pode ser um obstáculo no caminho da ciência (STORER, 2017).

Algumas iniciativas pontuais tem tentado modificar essa realidade, como é o caso da parceria entre a *Mozilla Science's Code* e o *Github* (reconhecido repositório de códigos de programas e softwares). Os dois grupos se juntaram em ação para que seja possível atribuir um código de DOI (*Digital Object Identifier*) aos softwares hospedados, tornando assim possível uma referência direta ao software em artigos científicos (GOBLE, 2014).

Outra iniciativa que busca alterar este paradigma é o Movimento da Pesquisa Reproduzível (do inglês, *Reproducible Research Movement*), a qual tem trabalhado para adaptar o padrão de comunicação da pesquisa científica de modo que incluam os dados e o código do software associados aos resultados obtidos (DONOHO et al., 2009; STODDEN; MIGUEZ, 2014).

2.6. Fatores e aspectos do ambiente

Diversos estudos tem buscado identificar os problemas associados ao processo de desenvolvimento de software científico e às práticas aplicadas. Entretanto, poucos deles têm tentado compreender como, aspectos e fatores ligados

em maior parte ao ambiente e à atividade de pesquisa em si, podem influenciar e impactar este processo de desenvolvimento de software científico.

Alguns trabalhos realizados nos últimos anos tem buscado identificar, classificar e compreender o impacto destes fatores, de modo que possam ser propostas atividades de melhoria e aperfeiçoamento dos processos de desenvolvimento de software científico (CARVER et al., 2007; MESH, 2015; MESH; BURNS; HAWKER, 2014; MESH; HAWKER, 2013; MESH; TOLAR; HAWKER, 2016).

Particularmente, o trabalho de Mesh (2015) apresenta um estudo interessante sobre estes fatores, indicando que propostas de melhorias para o processo de desenvolvimento devem considerar estes fatores, e devem ser construídas com base numa abordagem orientativa, fornecendo subsídios para que os pesquisadores e cientistas decidam, por si próprios, quais práticas adotar. Essas práticas devem estar disponíveis em uma base de conhecimento, a qual o pesquisador teria acesso e, associado ao conhecimento do domínio, optaria por empregar durante o desenvolvimento do software científico.

Para chegar aos fatores e aspectos do ambiente, Mesh (2015) conduziu algumas entrevistas com pesquisadores atuantes em projetos de pesquisa de diversas áreas de conhecimento. O resultado deste levantamento é apresentado no Quadro 1.

Quadro 1 – Aspectos e fatores do ambiente de software científico

Observado em > 50% das entrevistas	Observado em < 50% das entrevistas
Objetivos da pesquisa	Estabilidade dos dados
Progresso da pesquisa	Grau de financiamento da pesquisa
Disponibilidade do desenvolvedor do software	Automação da pesquisa
Funcionalidades do software	Complexidade do domínio da pesquisa
Organização e obtenção dos dados	Conhecimento do desenvolvimento de software
ROI do esforço de desenvolvimento do software	Acurácia da documentação do software
Disponibilidade do código-fonte do software	Mudanças nos requisitos do software
Impactos da pesquisa	Robustez do software
Pressão para publicação	Uso do software
Histórico de pesquisa	Mudanças tecnológicas

Tempo de desenvolvimento do software	Restrições tecnológicas
Manutenibilidade do software	Comercialização da pesquisa
Reprodutibilidade do software	Reconhecimento da comunidade
Usabilidade do software	Confiança nos resultados
	Gerenciamento de projeto
	Padrões de publicação
	Tempo da pesquisa
	Rastreabilidade do projeto do software
	Viabilidade do software
	Saídas esperadas do software
	Confiabilidade do software
	Escalabilidade do software
	Estabilidade do software
	Metas dos estudantes
	Ensino

Fonte: Adaptado de Mesh (2015)

Este capítulo apresentou uma revisão bibliográfica sobre o software científico e o processo de desenvolvimento comumente adotado nos ambientes de pesquisa. Foram introduzidas as definições, conceitos e características destes tipos de software, bem como foram descritos os perfis dos desenvolvedores e a importância que normalmente não lhes é conferida.

Por fim, os fatores e aspectos do ambiente de pesquisa foram apresentados, de modo a fornecer os subsídios necessários para o desenvolvimento deste trabalho. Os fatores selecionados para investigação serão utilizados na composição da metodologia de pesquisa do trabalho, descrita no próximo capítulo.

3.METODOLOGIA DA PESQUISA

Este capítulo apresenta a metodologia de pesquisa utilizada neste trabalho. São apresentadas as justificativas para a escolha do método qualitativo de estudo de caso bem como a estratégia de pesquisa. Além disso, são descritas as etapas de seleção, coleta, análise e tratamento de dados.

3.1. Estratégia de Pesquisa

Este trabalho, conforme definido no item 1.1, tem por objetivo a compreensão e a análise de como os fatores e aspectos do ambiente de pesquisa podem impactar no processo de desenvolvimento de software científico.

De modo a possibilitar a escolha de um método de pesquisa adequado para o alcance deste objetivo, faz-se importante compreender os diversos métodos e situações nas quais estes podem ser aplicados. Yin (2008) apresenta em seu estudo uma relação entre estes métodos e situações de aplicação, que pode ser observada no Quadro 2.

Quadro 2 - Métodos de Pesquisa e situações de aplicação

Método de pesquisa	Questões da pesquisa	Necessidade de controle sobre comportamento	Eventos contemporâneos
Experimento	“Como?” “Por quê?”	Sim	Sim
Pesquisa social (<i>survey</i>)	“Quem?” “O quê?” “Onde?” “Quanto?”	Não	Sim
Análise de arquivos	“Quem?” “O quê?” “Onde?” “Quanto?”	Não	Sim/Não
Pesquisa histórica	“Como?” “Por quê?”	Não	Não
Estudo de caso	“Como?” “Por quê?”	Não	Sim

Fonte: Adaptado de Yin (2008)

Nota-se que cada um dos métodos de pesquisa existentes na literatura, e amplamente adotados nos trabalhos científicos, busca responder diferentes questões e são, costumeiramente, aplicados em situações distintas.

Assim, considerando-se o objetivo do presente trabalho, tem-se que os métodos de pesquisa mais adequados seriam o experimento, a pesquisa histórica e o estudo de caso, visto que são aqueles que buscam responder questões de pesquisa relacionadas ao “como”.

Esta pesquisa optou pela adoção do estudo de caso como método para alcançar o objetivo proposto, uma vez que se buscou o entendimento sobre eventos contemporâneos do desenvolvimento de software científico, sem a necessidade da influência sobre o comportamento dos pesquisadores envolvidos.

Segundo Yin (2008), o estudo de caso possibilita entender um caso do mundo real, considerando-se as condições de contexto que influenciam este entendimento. Dessa forma, um estudo de caso contribui no entendimento e conhecimento de fenômenos individuais, organizacionais, políticos, sociais e de grupos, sendo assim um método de pesquisa comumente utilizado em ciências sociais (YIN, 2008).

O desenvolvimento de software é realizado por indivíduos, grupos e organizações. Portanto, questões políticas e sociais são importantes para o desenvolvimento de software, o que faz com que estudos de casos nessa área sejam uma abordagem relevante.

Além disso, neste trabalho foi feita a escolha de se realizar um estudo de caso múltiplo. Isto pode ser justificado por duas razões, amplamente relacionadas com o tema da pesquisa.

Em primeiro lugar, uma pesquisa com um único estudo de caso, estritamente ligado a um único grupo de pesquisa, poderia não fornecer as informações e evidências necessárias para o aprofundamento da análise proposta. Em segundo lugar, o desenvolvimento de software científico normalmente se dá por pouquíssimas pessoas (em geral, um único pesquisador), o que, novamente, poderia limitar a análise em questão (NGUYEN-HOAN; FLINT; SANKARANARAYANA, 2010). Corroboram-se estas justificativas com o fato de que é preferível a adoção de estudos de caso múltiplos a estudos de casos únicos, já que isto pode trazer um benefício substancial à pesquisa (GUSTAFSSON, 2017; YIN, 2008).

Os estudos de caso, e outros métodos de pesquisa, podem ser classificados em três categorias: exploratória, descritiva e explanatória (ou pesquisa causal) (YIN, 2008).

Pesquisas exploratórias buscam realizar algum tipo de investigação preliminar, normalmente para casos os quais não existam informações completas sobre o assunto, com o intuito de identificar novas hipóteses sobre os fenômenos estudados. Já as pesquisas descritivas, procuram descrever, de maneira mais precisa, os fenômenos estudados. Pesquisas explanatórias, por sua vez, objetivam fornecer explicações causais de fenômenos conhecidos (YIN, 2008).

Novamente, tomando-se por base o objetivo do trabalho, a categoria adotada para classificar o estudo de caso aqui proposto poderia ser tanto exploratória como explanatória. Exploratória do ponto de vista da elaboração de ideias e hipóteses sobre o impacto dos fatores do ambiente no processo de desenvolvimento; explanatória no sentido de que poderia prover explicações sobre relações de causa e efeito entre os fatores do ambiente e o processo de desenvolvimento.

Avaliando o estado da arte sobre o tema, e levando em consideração que esta é uma área de recente estudo, optou-se por adotar uma abordagem exploratória do problema.

Por fim, no que se refere às etapas da execução da pesquisa, adotou-se como estratégia uma adaptação daquela proposta por Eisenhardt (1989), que estabelece os passos apresentados no Quadro 3.

Quadro 3 - Etapas da execução da pesquisa

Passo	Atividades
Início	Definição da questão de pesquisa
Seleção de casos	Seleção dos casos específicos
Elaboração de instrumentos e protocolo	Métodos múltiplos para coleta de dados, com coleta de dados qualitativos e quantitativos
Entrada em campo	Realização da pesquisa de campo, associação de protocolos de pesquisa e anotações de campo
Análise de dados	Análise e cruzamento de dados
Formulação de hipóteses	Busca de razões no inter-relacionamento dos dados
Comparação com literatura	Comparação dos dados obtidos com a pesquisa bibliográfica realizada
Fechamento	Finalização das análises

Fonte: Adaptado de Eisenhardt (1989)

3.2. Seleção dos casos

Aspecto fundamental da pesquisa baseada em estudos de caso, a seleção de casos estabelece critérios objetivos para a escolha dos casos que serão analisados. Estes critérios, entretanto, são escolhidos baseando-se em motivos teóricos, e não estatísticos, visto que em estudos de caso a amostragem é teórica (EISENHARDT, 1989).

Dentro desta etapa, compreende-se a atividade de definição da população alvo do estudo. Eisenhardt (1989) estabelece que uma população determina o conjunto de elementos que será considerado como amostras da pesquisa. Além disso, a população auxilia na definição dos limites de possível generalização dos resultados.

Para a realização deste estudo, a escolha da população seguiu os seguintes passos:

1. A definição de uma rede de possíveis pesquisadores que desenvolvem software científico dentro do ambiente de pesquisa em que o autor se encontra;
2. Análise detalhada do perfil dos pesquisadores envolvidos e dos softwares objetos do desenvolvimento; e
3. Classificação e identificação dos casos em elegíveis e não elegíveis.

Para a execução dos passos 2 e 3 foram estabelecidos os seguintes critérios de seleção, tendo como base princípios e aspectos discutidos no Capítulo 2:

- Casos com pesquisadores com larga experiência em desenvolvimento destes softwares (mais de dez anos);
- Casos com pesquisadores com pouca experiência em desenvolvimento destes softwares (menos de 5 anos);
- Casos com pesquisadores em áreas correlatas à de Tecnologia da Informação;
- Ampla distinção da área de pesquisa, dentro do grupo de ciências exatas e engenharias, a qual se destina o software.

Após aplicação dos critérios foram escolhidos quatro casos.

3.3. Coleta de dados

Fontes para coleta de informações e evidências dos estudos de casos podem advir de diversas origens, sendo as seis principais: documentação, registros em arquivo, entrevistas, observação direta, observação com participante e artefatos físicos (YIN, 2008). Para este trabalho foi adotado a entrevista como fonte principal para a coleta de dados, tendo em vista a necessidade de compreender como o processo de desenvolvimento de software científico e de como os fatores do ambiente de pesquisa impactam neste processo.

Para a coleta de dados foram utilizadas entrevistas com um roteiro de perguntas semiestruturadas (CHARMAZ, 2006), tornando este um estudo de caso flexível (ANASTAS, 1999; ROBSON, 2002). Entrevistas semiestruturadas são compostas de questões abertas e fechadas, o que pode facilitar a coleta de informações e dados não previstos anteriormente (BONI; QUARESMA, 2005; SEAMAN, 1999).

A fim de garantir a proteção e privacidade dos indivíduos envolvidos, este trabalho considerou dois aspectos propostos por Yin (2008):

- Informar aos participantes, de maneira clara e objetiva, a natureza do estudo e o grau de exposição, solicitando formalmente o aceite para participação no estudo; e
- Proteger a privacidade e confidencialidade dos participantes de modo que não possam ser colocados em nenhuma posição indesejada.

Assim sendo, optou-se por não apresentar o nome das pessoas e empresas nesta pesquisa.

Foi elaborado um protocolo de pesquisa de forma a auxiliar os procedimentos de coleta de dados (YIN, 2008). Este protocolo foi dividido em duas partes:

1. Apêndice A – E-mail de Apresentação: mensagem eletrônica enviada aos participantes, contendo a visão geral do estudo e as regras para proteção, solicitando a permissão para participação voluntária; e
2. Apêndice B – Roteiro de perguntas: documento com as perguntas e tópicos a serem abordados na entrevista semiestruturada, de modo a orientar a coleta dos dados.

Foram entrevistadas quatro pessoas, cada uma delas relacionada a um ambiente e tipo de pesquisa diferente, entre os meses de maio e agosto de 2018. As

entrevistas foram realizadas presencialmente, no próprio ambiente de pesquisa do entrevistado, e foram gravadas com a devida autorização dos entrevistados.

3.4. Tratamento dos dados

Para a análise e tratamento de dados coletados, adotou-se nesta pesquisa uma abordagem pautada nos procedimentos de análise de conteúdo sugeridos por Bardin (2011). Segundo Bardin (2011), a análise de conteúdo é

um conjunto de técnicas de análise das comunicações visando a obter, por procedimentos sistemáticos e objetivos de descrição do conteúdo das mensagens, indicadores (quantitativos ou não) que permitam a inferência de conhecimentos relativos às condições de produção/recepção (variáveis inferidas) destas mensagens (BARDIN, 2011, p. 47).

Esta técnica é subdividida em três fases: pré-análise, exploração do material e tratamento dos resultados (inferência e interpretação).

A fase de pré-análise busca constituir a organização do trabalho de análise, estabelecendo um esquema com procedimentos bem definidos e flexíveis. Nesta fase, as entrevistas são transcritas, obedecendo às regras de exaustividade, representatividade, homogeneidade, pertinência e exclusividade.

A fase de exploração do material é caracterizada pela organização e caracterização dos dados resultantes das entrevistas, tornando possível o processo de tratamento dos dados.

Por fim, na fase de tratamento dos resultados, ocorre, de fato, a interpretação dos dados e informações pelo pesquisador, sempre à luz do conhecimento obtido no levantamento bibliográfico.

Considerando o objetivo da presente pesquisa, foram elencados os seguintes fatores para análise, escolhidos a partir daqueles mais citados no trabalho de Mesh (2015):

- Objetivo da pesquisa;
- Disponibilidade do desenvolvedor;
- Histórico de pesquisa;
- Organização e obtenção dos dados;
- Progresso da pesquisa durante o desenvolvimento;
- Pressão para publicação;

Além disso, optou-se por adicionar mais um aspecto à lista daqueles já apresentados por Mesh (2015). Conforme discutido na seção 2.5, a necessidade de reconhecimento do desenvolvedor é um fator relevante para o aperfeiçoamento desta área de pesquisa. Assim, considerou-se importante analisar também o impacto da falta de reconhecimento do desenvolvedor de software científico.

A escolha destes aspectos e fatores foi feita em razão da indicação na literatura de que estes fatores são aqueles de maior ligação às características do ambiente de pesquisa e do trabalho de pesquisa, bem como do próprio pesquisador (DOORN; AERTS; LUSHER, 2016; GOBLE, 2014; KANEWALA; BIEMAN, 2013; LIST; EBERT; ALBRECHT, 2017; PRABHU et al., 2011). De acordo com o objetivo do trabalho, estes fatores serão estudados de modo a compreender o seu impacto no processo de desenvolvimento de software científico.

Essa análise foi realizada com base na classificação das atividades de desenvolvimento proposta por Heaton e Carver (2015). São elas: requisitos, projeto, implementação, testes, refatoração e documentação.

De modo análogo ao trabalho de Heaton e Carver (2015), foi realizada a classificação das declarações dos pesquisadores entrevistados com base no uso do Dicionário Padrão de Computação da IEEE (GERACI et al., 1991), de modo a compreender a qual atividade cada uma das afirmações estava relacionada. Adicionalmente, de modo a complementar a análise, fez-se uso da norma ISO 12207 (IEEE, 2017), a qual define um arcabouço comum para processos do ciclo de vida de software, compreendendo uma terminologia padronizada para todas as atividades envolvidas.

A análise foi realizada individualmente para cada estudo de caso e, posteriormente, de forma comparativa entre os casos.

Este capítulo descreveu a metodologia de pesquisa proposta para este trabalho. O método escolhido será o de estudo de caso, com adoção de múltiplos casos. Será adotada a técnica de entrevistas semiestruturadas, com análise de resultados baseada na análise de conteúdo.

Os casos selecionados para realização do estudo e seu detalhamento são apresentados no capítulo seguinte.

4. ESTUDO DE CASO: DESENVOLVIMENTO DE SOFTWARE CIENTÍFICO

Este capítulo apresenta os estudos de caso realizados para quatro grupos de pesquisa distintos, dentro de um instituto de pesquisas do Estado de São Paulo, denominados neste trabalho de Grupo de Pesquisa A, Grupo de Pesquisa B, Grupo de Pesquisa C e Grupo de Pesquisa D, sendo cada um deles descrito em uma seção separada.

Para cada um dos casos é apresentado: as características do ambiente e do tipo de pesquisa realizada; o perfil do pesquisador envolvido no processo de desenvolvimento do software científico; características do processo de desenvolvimento utilizado; e a análise do impacto dos aspectos do ambiente de pesquisa no processo de desenvolvimento. Por fim, é apresentada a análise comparativa dos resultados de cada ambiente.

4.1. Grupo de Pesquisa A

A seguir são expostos os resultados e análises sobre o desenvolvimento de software científico e aspectos do Grupo de Pesquisa A.

4.1.1. Características e tipo de pesquisa

O desenvolvimento de software no Grupo de Pesquisa A ocorre no âmbito de pesquisas dentro da área de química, no Laboratório de Processos Químicos e Tecnologia de Partículas. A área de engenharia química possui razoável demanda de software científico, pois trabalha com a simulação, análise e verificação de formulações químicas, oriundas de experimentos. Em *surveys* já realizados, costuma figurar entre as dez áreas mais citadas no desenvolvimento de software científico (FARHOODI et al., 2013; NGUYEN-HOAN; FLINT; SANKARANARAYANA, 2010).

Duas pesquisas em específico estão envolvidas com o desenvolvimento de software científico: a) o tratamento de dados de espectros de massa para caracterização de amostras de petróleo; e b) a avaliação de dados para melhoria de controle do processo de solubilidade na cristalização de mentol. A principal aplicação dos softwares desenvolvidos é no tratamento e modelagem de dados. Estão envolvidos nestas pesquisas quatro pesquisadores, sendo que apenas um deles participa do desenvolvimento de software.

4.1.2. Perfil do pesquisador

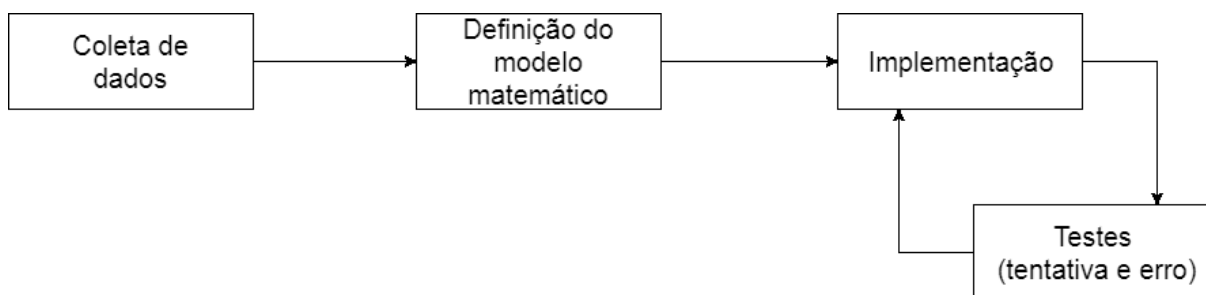
O pesquisador responsável pelo desenvolvimento no grupo de pesquisa possui formação de bacharelado na área de engenharia química e está realizando o mestrado na mesma área. Tem atuado no desenvolvimento de software científico há menos de 5 anos (tempo equivalente àquele em que adentrou a área) e possui menor experiência em pesquisa científica do que os demais pesquisadores do Grupo de Pesquisa A. Iniciou o trabalho com desenvolvimento de software em razão das necessidades da pesquisa científica desenvolvida pelo grupo de pesquisadores.

O conhecimento adquirido no desenvolvimento de software científico ocorreu na época da graduação, quando participou da disciplina de algoritmos. Entretanto, o pesquisador utiliza de diversas fontes de informação para obter auxílio e novos conhecimentos de modo a aplicar no processo de desenvolvimento de software científico, como recursos da Internet, conteúdo de auxílio das próprias ferramentas utilizadas e consulta a outros pesquisadores mais experientes que trabalham em outros ambientes de pesquisa. Nota-se que a cultura do ambiente de pesquisa pode apresentar certa influência sobre como essa busca por conhecimento se manifesta.

4.1.3. Processo de desenvolvimento

O processo de desenvolvimento utilizado pelo desenvolvedor (Figura 4) ocorre de maneira mais empírica, mas de certa forma, assemelha-se ao processo geral de desenvolvimento de software científico descrito no Capítulo 2.

Figura 4 - Processo de desenvolvimento - Grupo de Pesquisa A



Fonte: elaborado pelo autor

Primeiramente, o pesquisador busca obter e coletar todos os dados necessários à pesquisa e que necessitarão de simulação ou modelagem matemática. Essa coleta é feita de forma estruturada, utilizando planilhas e arquivos eletrônicos para o registro dos dados. Há uma etapa de conferência e validação para garantir que todas as informações necessárias foram obtidas.

Após isso, o pesquisador define o modelo matemático e método estatístico mais apropriado para o tratamento do problema de pesquisa. Essa definição influencia diretamente a escolha da ferramenta de desenvolvimento. No caso, é utilizado o software MATLAB para a programação dos modelos. Devido ao perfil do pesquisador envolvido, o processo de implementação e desenvolvimento do software envolve a busca, na própria ferramenta, de como desenvolver certos trechos de código.

Após a finalização da implementação, o pesquisador passa à etapa de testes que, segundo o ele, é a mais importante do processo de desenvolvimento. Os testes validam não só o software desenvolvido, mas também o modelo matemático escolhido.

De acordo com a entrevista realizada, o processo de desenvolvimento não envolve uma etapa específica para o projeto do software a ser construído, com exceção do que diz respeito à escolha do modelo a ser implementado. Além disso, cabe citar frase do entrevistado a respeito do processo em que diz que “o desenvolvimento é meio que na tentativa e erro”.

4.1.4. Impacto dos aspectos

No caso do Grupo de Pesquisa A, foram percebidos diversos impactos dos aspectos no processo de desenvolvimento.

4.1.4.1. Objetivo da Pesquisa

Com relação ao impacto do objetivo da pesquisa, a principal influência neste caso se dá sobre a etapa de especificação dos requisitos do software. O pesquisador entrevistado citou que o objetivo de sua pesquisa define os dados a serem coletados e que deverão ser modelados e inseridos no software.

Além disso, foi citado que o objetivo da pesquisa influencia na escolha da ferramenta de desenvolvimento a ser utilizada, o que pode ser considerado etapa costumeiramente ligada ao projeto do software. Apesar de haver uma única ferramenta utilizada no ambiente, o entrevistado indicou que existe essa correlação.

Por outro lado, do ponto de vista do pesquisador entrevistado não há a percepção de que, de alguma forma, o objetivo da pesquisa venha a influenciar a implementação do software.

Por fim, ainda relacionado ao objetivo de pesquisa, o pesquisador afirmou que as incertezas da pesquisa existem e que influenciam no processo de desenvolvimento, entretanto, em suas experiências essa influência não existiu.

4.1.4.2. Disponibilidade do desenvolvedor

A respeito da disponibilidade do desenvolvedor, foi solicitado ao entrevistado que fizesse uma estimativa da porcentagem do tempo a qual se dedica ao desenvolvimento de software científico, frente ao tempo total despendido na pesquisa científica. Nas palavras do pesquisador, estima-se “30% do tempo só no código”, sem considerar outras atividades. Entretanto, o entrevistado indicou que a maior parte do esforço está relacionada ao código e a programação do software em si, bem como a realização dos testes.

Este dado vai ao encontro do que costuma ser observado em pesquisas já realizadas, nas quais a porcentagem de tempo reservada para atividades de desenvolvimento de software costuma variar entre 20% e 35% (PRABHU et al., 2011).

Uma hipótese a ser levantada é a de que, devido à concentração de esforço na etapa de implementação do software, atividades como definição de requisitos e projeto do software não apresentam relevante importância para o processo de desenvolvimento do Grupo de Pesquisa A. Isto também pode estar diretamente relacionado ao perfil do desenvolvedor e à sua experiência no desenvolvimento de software.

4.1.4.3. Histórico de Pesquisa

Passando ao aspecto de histórico de pesquisa verifica-se que, neste caso, possui grande impacto no processo de desenvolvimento. Devido ao perfil e experiência do pesquisador envolvido, a atividade de desenvolvimento é realizada quase que de forma empírica, com base no aprendizado contínuo pautado pela tentativa e erro.

A pouca experiência do pesquisador tem maior influência na atividade de definição dos requisitos do software. Foi relatada, por diversas vezes durante a entrevista, a necessidade de constante modificação do software para inclusão de funcionalidades cuja necessidade não havia sido percebida quando do início da construção do software.

Além disso, do ponto de vista da implementação, o histórico de pesquisa do desenvolvedor impede que sejam adotadas novas práticas e padrões de desenvolvimento pela falta de confiança em aplica-las.

4.1.4.4. Organização e obtenção dos dados

Por sua vez, a organização e obtenção dos dados é aspecto crucial do Grupo de Pesquisa A. A organização dos dados que serão modelados e calculados estabelece uma etapa de concepção de funcionalidades necessárias ao software. Segundo análise do caso, entende-se que este aspecto apresenta maior influência na atividade de definição dos requisitos do software científico.

O software começa a ser projetado a partir do momento em que todos os experimentos já foram realizados e todos os dados necessários já foram obtidos, o que, na visão do pesquisador, torna a atividade de implementação muito mais estável e menos suscetível ao impacto deste aspecto.

Por outro lado, apesar da obtenção de dados antes do início do desenvolvimento, o pesquisador relatou que, após a execução dos testes do software desenvolvido, ocorreram casos de haver a necessidade de se redefinir funcionalidades e realizar ajustes no software. Isto normalmente ocorre devido à natureza da instabilidade dos requisitos nestes ambientes, conforme já indicado nos trabalhos de Ackroyd et al. (2008) e Storer (2017).

Foi indicado ainda pelo pesquisador entrevistado que este aspecto também tem certa influência na escolha da ferramenta a ser utilizada para o desenvolvimento, o que pode ser entendido como parte que compõe a atividade de projeto do software.

4.1.4.5. Progresso da pesquisa

O aspecto de progresso da pesquisa durante o desenvolvimento apresenta certa contribuição para o surgimento de alguns problemas no processo de desenvolvimento, principalmente aqueles relacionados aos requisitos e projeto do software.

Em primeiro lugar, o entrevistado indicou que é realizada uma concepção geral do software no começo da pesquisa científica, mas que o projeto do software só é concretizado na etapa de desenvolvimento. Isto novamente reforça o entendimento da literatura, a qual indica que o processo de desenvolvimento inicia-

se com uma ideia vaga do quê deve ser desenvolvido, acarretando nos problemas citados no Capítulo 2 (SEGAL, 2009).

Em segundo lugar, o entrevistado disse que, se possível, começaria o desenvolvimento do software o mais cedo possível, o que auxiliaria na melhor estruturação do software científico e seus componentes internos.

O pesquisador também mencionou que o progresso da pesquisa implica em necessidade de refatoração do código já desenvolvido, para que seja possível se adaptar às novas descobertas da pesquisa.

4.1.4.6. Pressão para publicação

A pressão para publicação, ao contrário dos demais aspectos já discutidos, é um aspecto que apresenta maior influência sobre a etapa da implementação, segundo o entrevistado. Sendo os resultados da pesquisa de suma importância para sua conclusão, a pressão para a publicação faz com que sejam realizados cortes no que está sendo desenvolvido, principalmente quando já existem resultados preliminares que apresentam confiabilidade adequada. Como são os testes do software que garantem essa confiabilidade dos resultados, não há impacto sobre esta etapa do processo de desenvolvimento.

O entrevistado também relatou que, devido a estes cortes, em alguns casos o software acabou por ficar incompleto, do ponto de vista do que era esperado que fosse construído. Citou o caso em que, em um determinado desenvolvimento, uma parte do tratamento dos dados teve de ser realizada fora do software desenvolvido (planilha), mas que se houvesse tempo, essa funcionalidade de tratamento dos dados estaria incorporada ao software final.

Como comprovação desta análise, cita-se passagem da entrevista, na qual o pesquisador indica que “enquanto tem tempo (para melhorar), vou mudando o que é possível...” e que “sempre há curiosidade para testar outras ferramentas e algoritmos”.

4.1.4.7. Reconhecimento do trabalho de desenvolvimento

Por fim, o aspecto referente ao reconhecimento do trabalho de desenvolvimento se faz muito presente no Grupo de Pesquisa A.

O entrevistado indicou que há valorização dos demais pares do seu laboratório pelo trabalho de desenvolvimento, mas que não vê este reconhecimento

na publicação de seu trabalho de pesquisa, como já constatado anteriormente (GOBLE, 2014). Entretanto, o pesquisador informou que possui a preocupação de tornar o código do software disponível, para que isso auxilie em pesquisas futuras.

Além disso, o entrevistado citou que o reconhecimento é fator motivacional para a melhoria do seu próprio processo de desenvolvimento, fazendo com que esteja preocupado em trabalhar na melhor estruturação e projeto dos softwares, bem como buscar novas ferramentas para o desenvolvimento.

Uma hipótese para o comportamento deste pesquisador em específico pode estar diretamente relacionada ao perfil e histórico de pesquisa, ainda que esta hipótese não possa ser confirmada.

Além dos fatores elencados para estudo neste trabalho, cabe-se ressaltar algumas considerações a respeito do processo de desenvolvimento e atividades correlatas, realizadas pelo próprio entrevistado:

- Possíveis melhorias na implementação do software só são realizadas após a obtenção de resultados satisfatórios para conclusão da pesquisa científica;
- Há alguma preocupação com reusabilidade do software desenvolvido, mas isso tende a variar conforme o tipo da pesquisa científica; e
- A etapa de documentação do software desenvolvido não é realizada de maneira sistemática, apenas pontualmente.

4.2. Grupo de Pesquisa B

A seguir são expostos os resultados e análises sobre o desenvolvimento de software científico e aspectos do Grupo de Pesquisa B.

4.2.1. Características e tipo de pesquisa

O desenvolvimento de software no Grupo de Pesquisa B ocorre no âmbito de pesquisas dentro da área de bionanomanufatura, no Laboratório de Biotecnologia Industrial. Dentro do ambiente de pesquisa, esta é considerada uma área nova, e isto tem gerado a demanda por desenvolvimento de software científico, principalmente para simulação.

São realizados diversos tipos de pesquisa, mas uma linha de pesquisa em específico demanda o desenvolvimento de software científico: a simulação de processos fenomenológicos em biorreatores, de modo a otimizar a sua utilização.

Mais de dez pesquisadores participam, em diferentes momentos, dessa linha de pesquisa, mas apenas uma pesquisadora é responsável pelo desenvolvimento do software.

4.2.2. Perfil do pesquisador

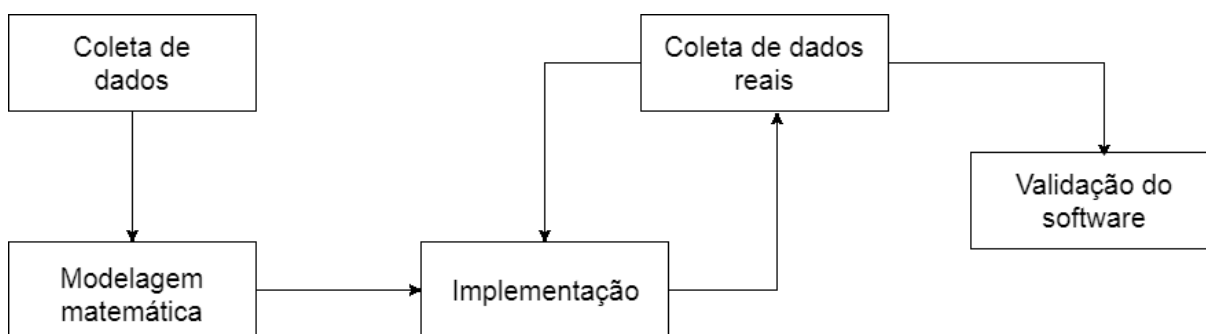
A pesquisadora responsável pelo desenvolvimento no grupo de pesquisa possui mestrado e doutorado em biotecnologia. Tem atuado na área de pesquisa científica há mais de 25 anos, e possui experiência no desenvolvimento de softwares há pelo menos 18 anos. É uma das pesquisadoras seniores do Grupo de Pesquisa B.

O conhecimento de desenvolvimento de software científico advém da própria experiência nestas atividades, obtido nesse longo período de desenvolvimento de software. No entanto, a pesquisadora também tem realizado cursos complementares e mantém contato direto com pesquisadores de outras instituições de modo a aprimorar seu conhecimento.

4.2.3. Processo de desenvolvimento

O processo de desenvolvimento adotado no Grupo de Pesquisa B (Figura 5) é estruturado com algumas atividades bem definidas, tomando a mesma sequência do processo geral apresentado no Capítulo 2.

Figura 5 - Processo de desenvolvimento - Grupo de Pesquisa B



Fonte: elaborado pelo autor

Em primeiro lugar, são realizadas coletas de dados, por meio de ensaios laboratoriais, que fornecerão embasamento para definição da modelagem matemática que servirá como alicerce para a simulação a ser implementada no software.

Com base nos dados coletados, a pesquisadora define a modelagem do processo químico e efetua a implementação do modelo, em diversas ferramentas e linguagens de programação como MATLAB, Fortran e C.

Em seguida, são realizadas coletas de dados reais para que sejam feitos ajustes do software desenvolvido. Esses ajustes podem levar vários meses para serem realizados, de acordo com os dados reais que são obtidos.

Por fim, ocorre a validação final do software junto ao pesquisador responsável pela pesquisa, acompanhado de testes estatísticos.

4.2.4. Impacto dos aspectos

A seguir são apresentados os pontos de influência dos aspectos no Grupo de Pesquisa B.

4.2.4.1. Objetivo da Pesquisa

O aspecto relacionado ao objetivo da pesquisa coloca dois impactos no processo de desenvolvimento do Grupo de Pesquisa B: um primeiro na especificação dos requisitos e um segundo, no projeto do software, ambos correlacionados.

Os softwares desenvolvidos dentro deste ambiente devem possuir duas características básicas: desempenho e precisão. Estes são requisitos definidos conjuntamente ao objetivo da pesquisa, e que devem ser, obrigatoriamente, satisfeitos. Requisitos deste tipo são normalmente mencionados em outros trabalhos sobre desenvolvimento de software científico (NEUMANN; ESPINA; DAZA, 2017; TANG, 2008).

Associado a isso, a entrevistada informou que a linguagem de programação adotada no desenvolvimento pode variar. Do ponto de vista da pesquisadora, não há uma correlação direta entre seu conhecimento e a linguagem escolhida; essa definição depende muito mais dos requisitos colocados pelo objetivo da pesquisa.

4.2.4.2. Disponibilidade do desenvolvedor

A disponibilidade do desenvolvedor está altamente relacionada à etapa de implementação do software, porém de forma mais específica se dá em maior parte na etapa de ajustes no software, citada no processo de desenvolvimento.

Não obstante, a pesquisadora informou que deve ter disponibilidade também durante a etapa de validação do software, que envolve a aplicação de testes estatísticos.

Quando solicitada para realizar uma estimativa do tempo a qual se dedica às atividades de desenvolvimento de software científico, a pesquisadora disse que normalmente passa de 2 a 3 meses (de um ano de pesquisa) desempenhando estas atividades (percentualmente, de 16% a 25%).

4.2.4.3. Histórico de Pesquisa

O histórico de pesquisa da entrevistada é um fator que exerce grande influência no processo de desenvolvimento de software.

Em primeiro lugar, por ter grande experiência, a pesquisadora tem facilidade em compreender os requisitos do software e em projetar o que deve ser desenvolvido. Foi relatado que são projetadas diversas condições de contorno dentro do software para evitar problemas relacionados aos dados como, por exemplo, tratamentos estatísticos.

Além disso, o histórico de pesquisa da desenvolvedora ajuda a lembrar softwares já desenvolvidos para outras pesquisas, auxiliando na recuperação de trechos de código para reuso.

A etapa de implementação também se beneficia desta experiência da pesquisadora, com maior facilidade de resolução de problemas que poderiam acarretar em atrasos no desenvolvimento ou na geração de resultados insatisfatórios para o pesquisador.

A atividade de documentação do software é citada pela desenvolvedora, mais precisamente no que se refere à documentação do código-fonte do software. Entretanto, a própria desenvolvedora admite que não percebe grande melhoria no processo de desenvolvimento, uma vez que atividades de manutenção de softwares já desenvolvidos continuam sendo complexas.

4.2.4.4. Organização e obtenção dos dados

A organização e obtenção dos dados apresenta impacto em duas etapas do processo de desenvolvimento: requisitos e testes. Neste caso, os dados são utilizados tanto para definir os requisitos como para verificar a implementação destes requisitos durante os testes.

A entrevistada ressaltou a grande dificuldade de se obter os dados previamente, confirmando o que já havia sido observado em outros trabalhos (KANEWALA; BIEMAN, 2013), fazendo com que haja grande dificuldade na especificação dos requisitos. Entretanto, quando isto se torna possível, essa obtenção de dados auxilia na definição do protocolo de validação do software.

4.2.4.5. Progresso da pesquisa

Por sua vez, o progresso da pesquisa durante o desenvolvimento do software relaciona-se diretamente com o último aspecto mencionado. Os dados necessários à implementação dos ajustes do software são obtidos conforme a pesquisa científica progride. Desta forma, mudanças inesperadas podem ocorrer.

Outro fator importante destacado pela pesquisadora entrevistada é que, situações corriqueiras na pesquisa científica afetam de forma inesperada a etapa de implementação. Um exemplo é o caso da falha de equipamentos utilizados nos ensaios, e que geram os dados necessários à modificação do software. A pesquisadora comentou que, costumeiramente, necessita adicionar modificações ao software para tratar a ausência de dados gerados por equipamentos que não serão concertados em tempo hábil ao término da pesquisa.

4.2.4.6. Pressão para publicação

No Grupo de Pesquisa B a pressão para publicação não configura um aspecto de grande impacto no processo de desenvolvimento de software científico, contrariando o que é relatado na literatura existente (DOORN; AERTS; LUSHER, 2016; LIST; EBERT; ALBRECHT, 2017; PATRICK et al., 2016). A entrevistada afirma existir a pressão para publicação dos resultados da pesquisa, no entanto disse que não vê influência direta em atividades de desenvolvimento do software. Foi citado que isto poderia acarretar em necessidade de redução do número de testes realizados no software, mas a pesquisadora enfatizou que “(...) a experiência auxilia na definição de condições de contorno dentro do próprio software”.

4.2.4.7. Reconhecimento do trabalho de desenvolvimento

Por fim, a desenvolvedora destacou grande importância ao fator de reconhecimento do software científico. Segundo sua percepção, a ausência de reconhecimento acarreta na falta de interesse de outros pesquisadores para

aspectos de melhoria do processo de desenvolvimento de software científico, principalmente àqueles associados às atividades de implementação e programação. Ainda segundo sua visão, novos desenvolvedores ingressantes na área somente tem a preocupação de ter o software funcional, da maneira mais rápida possível, sem considerar a possibilidade de melhorias do código desenvolvido ou da estrutura do software.

Como comentário adicional, a entrevistada citou que há alguma preocupação em prever a reutilização de softwares já desenvolvidos, porém, a natureza das pesquisas desenvolvidas costuma impossibilitar que isto seja alcançado, já que os softwares são muito distintos.

4.3. Grupo de Pesquisa C

A seguir são expostos os resultados e análises sobre o desenvolvimento de software científico e aspectos do Grupo de Pesquisa C.

4.3.1. Características e tipo de pesquisa

No Grupo de Pesquisa C, o desenvolvimento de software científico ocorre dentro da área de engenharia elétrica, no Laboratório de Automação, Governança e Mobilidade Digital. Esse laboratório, em específico, pertence a uma área diretamente relacionada à Tecnologia da Informação, que desenvolve pesquisas ligadas à computação e automação.

A pesquisa objeto do estudo trata-se da localização de objetos por meio de identificação por radiofrequência, e o software científico é utilizado para simulação. Trabalham nesta pesquisa cinco pesquisadores, inclusive aquele único responsável pelo desenvolvimento de software utilizado no laboratório.

4.3.2. Perfil do pesquisador

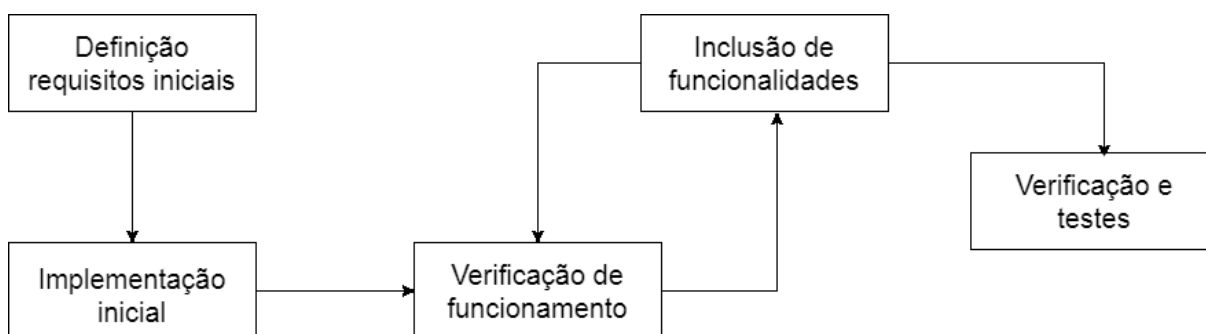
O pesquisador que desenvolve software possui experiência na área de engenharia elétrica há mais de 10 anos, entretanto só desenvolve softwares para esta área há 5 anos. É formado em engenharia elétrica e possui mestrado em engenharia de computação (realizado após ingressar na área de desenvolvimento). O nível de experiência em pesquisa científica é o mesmo dos demais membros da equipe.

Devido ao mestrado em engenharia de computação, o pesquisador entrevistado possui relativo conhecimento sobre práticas para desenvolvimento de software em geral, bem como maior facilidade para o desenvolvimento de software.

4.3.3. Processo de desenvolvimento

O processo de desenvolvimento adotado pelo pesquisador (Figura 6) subdivide-se em algumas etapas, que seguem de forma análoga as atividades do processo descrito no Capítulo 2.

Figura 6 - Processo de desenvolvimento - Grupo de Pesquisa C



Fonte: elaborado pelo autor

De modo a iniciar o processo, o pesquisador estabelece alguns poucos requisitos necessários ao software como, por exemplo, variáveis indispensáveis à leitura da radiofrequência.

Após essa definição, segue para a implementação inicial de modo a verificar a viabilidade da obtenção dos dados necessários e a validar o funcionamento do software.

Com a garantia de que é possível obter os dados necessários, o pesquisador prossegue de um modo incremental, adicionando novas funcionalidades ao software e validando o seu funcionamento.

Por fim, ocorre a verificação de todos os dados gerados e testes gerais para garantia de que o software apresenta um funcionamento conforme o esperado.

4.3.4. Impacto dos aspectos

A seguir são descritos os pontos de influência dos aspectos no Grupo de Pesquisa C.

4.3.4.1. Objetivo da Pesquisa

Do ponto de vista dos impactos do objetivo da pesquisa sobre o processo de desenvolvimento do software, constatou-se que há alguma correlação entre o objetivo e a linguagem de programação escolhida para a implementação do software. No caso, o pesquisador havia escolhido utilizar a linguagem Java para desenvolver o software, devido ao maior conhecimento e afinidade que tinha com a linguagem. Entretanto, optou por alterar esta escolha para a linguagem C# pouco tempo depois, devido a dois fatores: utilização da linguagem de programação por outros pesquisadores em pesquisas similares e dificuldades na implementação inicial, impossibilitando a extração de resultados preliminares necessários ao início da pesquisa.

As demais atividades de desenvolvimento, como requisitos, implementação e testes, não sofreram influências relacionadas ao objetivo da pesquisa.

4.3.4.2. Disponibilidade do desenvolvedor

Quanto ao aspecto de disponibilidade do desenvolvedor, o entrevistado disse que, durante o período em que estava trabalhando no desenvolvimento, optou por dedicar-se de maneira quase que exclusiva às atividades de desenvolvimento do software, retornando à pesquisa somente para verificar se os resultados obtidos eram satisfatórios.

Do ponto de vista de tempo de dedicação ao software, foi indicada uma estimativa de cerca de 20% do tempo de pesquisa somente para o desenvolvimento do software.

4.3.4.3. Histórico de Pesquisa

No caso do Grupo de Pesquisa C, o histórico de pesquisa do desenvolvedor tende a gerar influências no projeto e na implementação do software.

Conforme observado, as experiências prévias do pesquisador fizeram com que o projeto de desenvolvimento fosse estabelecido em função da linguagem de programação de maior afinidade, o que, por sua vez, causou certa dificuldade na implementação inicial. Entretanto, esse histórico de pesquisa foi fundamental para o pesquisador superar problemas e adversidades durante a etapa de programação do software. Por possuir conhecimento prévio em computação, o pesquisador soube

buscar soluções para os problemas em fontes mais específicas (como fóruns de discussão).

4.3.4.4. Organização e obtenção dos dados

O aspecto de organização e obtenção dos dados mostrou-se fundamental neste caso, com impactos diretos na especificação de requisitos e teste do software.

Em primeiro lugar, o objetivo de pesquisa colocado tornava obrigatória a obtenção de dados para comparação entre os diversos métodos de localização por radiofrequência. Não houve preocupação quanto à obtenção de dados quando da definição dos requisitos do software, na qual o pesquisador não se atentou para o fato de que algumas das variáveis de leitura de radiofrequência poderiam não ser obtidas.

Em segundo lugar, houve sério problema nos testes de verificação do software. Em dado momento, os resultados gerados pelo software causaram certa desconfiança quanto à confiabilidade do software desenvolvido. O pesquisador relatou ficar em dúvida quanto à origem do problema: má coleta dos dados obtidos a priori do desenvolvimento ou erro na implementação do software. Dúvida essa que só foi sanada após replicação da coleta com equipamento laboratorial, que constatou efetivo problema na coleta de dados, descartando possibilidade de falha na solução desenvolvida.

4.3.4.5. Progresso da pesquisa

Sob o olhar do aspecto de progresso da pesquisa, houve indicativo de impactos positivos e negativos no processo de desenvolvimento. Segundo análise do caso, o progresso da pesquisa acompanha, em mesmo nível, o desenvolvimento do software. Isto quer dizer que, à medida que ocorrem análises preliminares dos resultados, novos requisitos são necessários.

Por outro lado, a escrita da pesquisa e da análise de seus resultados levou o pesquisador a refletir quanto à necessidade de melhorias no software, deixando de lado a implementação de algumas funcionalidades e mudanças estruturais para dar maior atenção a outras.

4.3.4.6. Pressão para publicação

A pressão para publicação da pesquisa demonstrou uma clara influência no processo de desenvolvimento do software científico.

O pesquisador afirmou que havia previsto diversas funcionalidades para o software durante a etapa de projeto, mas que não conseguiu dar prosseguimento a implementação, devido à necessidade de se dedicar à escrita da pesquisa.

A etapa de documentação teve de ser ignorada durante o processo de desenvolvimento, bem como atividades de refatoração e melhoria da estrutura do software. A reusabilidade do software e melhorias de interface, que também eram intenção do pesquisador, não puderam ser projetadas.

Ademais, algumas ferramentas externas tiveram de ser utilizadas para complementação da análise dos resultados, o que não era desejado pelo pesquisador, que esperava poder ter desenvolvido dentro do software todas as funcionalidades necessárias para análise dos dados.

4.3.4.7. Reconhecimento do trabalho de desenvolvimento

Finalmente, o entrevistado revelou não perceber o reconhecimento do trabalho fora do grupo de pesquisa no qual está inserido. Os resultados da pesquisa e o software chegaram a ser apresentados, mas a discussão se limitou somente aos resultados.

A análise deste aspecto revela que há implicações diretas à confiabilidade do software. Quando da apresentação, não houve questionamento quanto à garantia de que os mesmos não foram produzidos incorretamente devido a falhas na implementação do software.

4.4. Grupo de Pesquisa D

A seguir são expostos os resultados e análises sobre o desenvolvimento de software científico e aspectos do Grupo de Pesquisa D.

4.4.1. Características e tipo de pesquisa

No Grupo de Pesquisa D, o desenvolvimento de software científico ocorre dentro da área de engenharia mecânica, no Laboratório de Vazão. São desenvolvidos diversos tipos de pesquisas relacionadas a estudos de vazão e pressão, bem como pesquisas sobre comportamento do vento em estruturas.

O software científico objeto de desenvolvimento neste ambiente é utilizado para tratamento de dados e simulação dos processos físicos. Trabalham nesta pesquisa cinco pesquisadores, inclusive aquele único responsável pelo desenvolvimento de software utilizado no laboratório.

4.4.2. Perfil do pesquisador

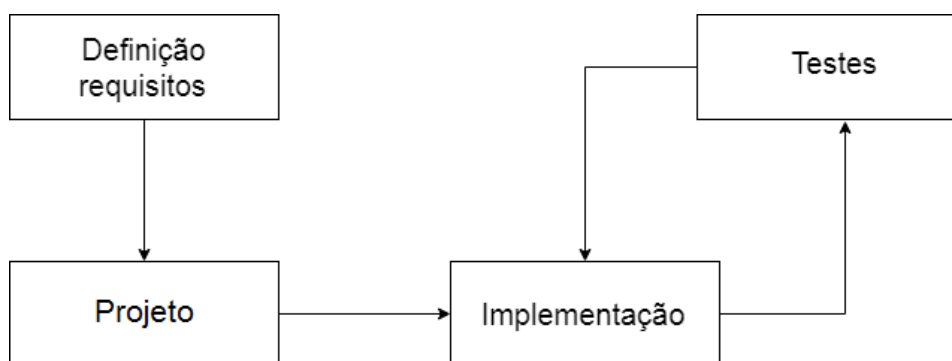
O pesquisador responsável pelo desenvolvimento no grupo de pesquisa possui mestrado e doutorado em engenharia mecânica. Tem atuado na área de pesquisa científica há mais de 20 anos, e possui experiência no desenvolvimento de softwares há mais de 25 anos. É um dos pesquisadores seniores do Grupo de Pesquisa D.

Em específico, o pesquisador tem grande experiência em programação, pois já desenvolvia este tipo de atividade antes mesmo de sua graduação, tendo conhecimento de diversas linguagens de programação. Relatou sempre ter tido interesse na área e que, em suas palavras, “não sei se sou um programador, mas talvez trabalhe com programação tanto quanto pessoas da área de computação”.

4.4.3. Processo de desenvolvimento

O processo de desenvolvimento adotado pelo pesquisador (Figura 7) subdivide-se em algumas etapas, com certo grau de estruturação.

Figura 7 - Processo de desenvolvimento - Grupo de Pesquisa D



Fonte: elaborado pelo autor

Primeiramente, o pesquisador procura identificar os requisitos necessários ao desenvolvimento do software pretendido. São elencados requisitos funcionais e não funcionais como, por exemplo, desempenho e portabilidade do software. Também são definidos modelos de cálculo e métodos de tratamento de dados.

Após este levantamento, é realizado um projeto básico do software, tendo em conta a preocupação com organização da arquitetura do mesmo, e levando em consideração conceitos como componentização e modularização do software.

Tendo os requisitos e o projeto definidos, o desenvolvedor passa a implementação propriamente dita, no que é considerado por ele como uma “programação exploratória”, na qual vão se testando várias implementações até se chegar ao resultado desejado. São utilizadas diversas linguagens de programação, tais como C, R e Python.

Por fim, são realizados alguns testes para validação do software desenvolvido. Importante destacar que, no último ano o pesquisador tem buscado introduzir o conceito de desenvolvimento orientado à testes, com a construção e definição de testes conjuntamente à etapa de implementação.

4.4.4. Impacto dos aspectos

A seguir são descritos os pontos de influência dos aspectos no Grupo de Pesquisa D.

4.4.4.1. Objetivo da Pesquisa

Com relação ao impacto do objetivo da pesquisa, foram evidenciados impactos sobre diversas etapas do desenvolvimento. O pesquisador entrevistado citou que o objetivo de sua pesquisa define os requisitos do software, estabelecendo as principais funcionalidades que deverão constar na versão final.

Além disso, foi citado que o objetivo da pesquisa influencia na escolha da linguagem de programação a ser utilizada, o que pode estar ligado à etapa de projeto do software.

Por fim, o pesquisador afirmou que o objetivo da pesquisa gera uma demanda em documentação do software que está sendo desenvolvido, tanto para fins de análise dos modelos e métodos de tratamento de dados, como para fins de manutenção e modificação futura do software.

4.4.4.2. Disponibilidade do desenvolvedor

Quanto ao aspecto de disponibilidade do desenvolvedor, o entrevistado disse que, por sua grande experiência no trabalho de desenvolvimento, não costuma necessitar dedicar grande parte do tempo para estas atividades.

Do ponto de vista de tempo de dedicação ao software, foi indicada uma estimativa de cerca de 10% do tempo de pesquisa somente para o desenvolvimento do software, principalmente na etapa de implementação.

4.4.4.3. Histórico de Pesquisa

O histórico de pesquisa neste grupo de pesquisa é um fator que exerce grande influência no processo de desenvolvimento de software.

Em primeiro lugar, por ter grande experiência, o pesquisador tem facilidade em compreender os requisitos e em elaborar um projeto de software que contemple conceitos como modularização e separação de funcionalidades. Além disso, o histórico de pesquisa do desenvolvedor ajuda a busca por componentes e outros softwares desenvolvidos de modo a promover a reutilização e reuso.

Foi percebido também um impacto direto na construção de casos de teste que permitam a verificação e validação da implementação do software. Como já citado, o pesquisador agora tem começado a trabalhar em desenvolvimento orientado à testes.

A atividade de documentação do software também se beneficia deste histórico do pesquisador, uma vez que os códigos desenvolvidos são amplamente documentados, inclusive utilizando ferramentas especiais para esta atividade.

4.4.4.4. Organização e obtenção dos dados

No Grupo de Pesquisa D a organização e obtenção de dados tem grande importância na etapa de implementação do software científico. O pesquisador relatou que normalmente precisa integrar diversas fontes de dados gerados por equipamentos e por outras pesquisas, o que acaba por gerar grande esforço de programação.

Outro ponto citado foi que, em grande parte dos softwares desenvolvidos, a organização dos dados necessários à pesquisa ocorre em paralelo com a própria programação do sistema (ou até mesmo dentro do programa), aumentando a complexidade do software desenvolvido, bem como o tempo necessário para o desenvolvimento.

4.4.4.5. Progresso da pesquisa

Do ponto de vista do aspecto de progresso da pesquisa, o processo de desenvolvimento sofre algumas restrições nas atividades de implementação e testes, tendo certo relacionamento com o aspecto de organização e obtenção dos dados.

O pesquisador informou que, devido à grande quantidade de fontes de dados, ocorrem certos problemas na implementação do software que, muitas vezes, não são solucionados da forma mais adequada. Normalmente são resolvidos paliativamente para possibilitar a continuidade da pesquisa. Entretanto, o pesquisador relatou que costuma retomar estes pontos para aplicar correções adequadas.

Outro ponto importante é que, conforme ocorre avanço da pesquisa, alguns testes deixam de ser realizados, ainda que necessários.

4.4.4.6. Pressão para publicação

A pressão para publicação é fator de alta relevância na atividade de implementação do software científico, segundo visão do pesquisador entrevistado. Apesar da vasta experiência em desenvolvimento de software científico, o pesquisador informou que este é um fator que ainda não conseguiu mitigar.

Assim como no aspecto de progresso da pesquisa, o pesquisador informou que a pressão para publicação implica em resolução de problemas de forma não estruturada e inadequada, diminuindo a qualidade do código produzido e, potencialmente, a qualidade do software.

Também informou que algumas funcionalidades dos softwares costumam ser retiradas para que o prazo para publicação de resultados possa ser atendido.

4.4.4.7. Reconhecimento do trabalho de desenvolvimento

Por fim, o pesquisador indicou que não consegue perceber o reconhecimento pelo trabalho de desenvolvimento de software científico.

Como consequências diretas, o pesquisador entrevistado disse que a qualidade do código e do software produzido seria muito maior caso houvesse maior reconhecimento da comunidade científica.

Além disso, o entrevistado citou a importância do reconhecimento como fator primordial para que movimentos que buscam viabilizar a reprodutibilidade da

pesquisa possam ter sucesso, em consonância com o que foi apontado nos trabalhos de Donoho et al. (2009) e Stodden e Miguez (2014).

4.5. Análise comparativa dos resultados

Após a apresentação e avaliação de cada estudo de caso, algumas análises comparativas podem ser realizadas.

Em primeiro lugar, é interessante notar que, embora os grupos de pesquisa analisados sejam completamente distintos, com áreas de aplicação, perfis de pesquisadores e objetivos de pesquisa totalmente diferentes, os processos de desenvolvimento adotados em cada um deles é significativamente similar.

Em todos os casos, há uma concepção preliminar do problema ou dos dados que se deseja obter, seguido de uma implementação inicial da solução e uma série de iterações de desenvolvimento, até se chegar a um software “final” que, de certa forma, atende à necessidade do pesquisador.

Essa constatação se mostra importante, pois confirma que o modelo geral de desenvolvimento proposto por Segal (2009) apresenta-se verdadeiro, e continua válido até o presente momento.

4.5.1. Objetivo da Pesquisa

O impacto do objetivo da pesquisa sobre o processo de desenvolvimento se traduz, basicamente, em influências sobre a etapa de requisitos e projeto do software (Quadro 4).

O objetivo da pesquisa, cuja questão o pesquisador busca responder, define o escopo da pesquisa, bem como quais resultados buscam-se atingir. Sendo o software o meio pelo qual esses resultados serão atingidos, seus requisitos estarão diretamente relacionados ao objetivo.

No caso dos Grupos de Pesquisa A e B e D, ficou constatada essa relação direta. O Grupo de Pesquisa A apresentou necessidade de funcionalidades no software de acordo com os dados que serão obtidos na pesquisa, entretanto isso não foi percebido no Grupo de Pesquisa B, que apenas destacou a necessidade de cumprimento de requisitos não funcionais; o Grupo de Pesquisa D tem uma percepção completa de requisitos. Uma possível explicação para essas diferenças pode estar na experiência dos desenvolvedores que, frente a diversas pesquisas já realizadas, possuem grau de maturidade elevado na definição de requisitos.

Quadro 4 - Impacto dos Objetivos da pesquisa

Atividades do Desenvolvimento	Impacto dos Objetivos da pesquisa			
	Grupo de Pesquisa A	Grupo de Pesquisa B	Grupo de Pesquisa C	Grupo de Pesquisa D
Requisitos	Define os modos de operação do software	Define os requisitos não funcionais	-	Define os requisitos do software
Projeto	Define a ferramenta a ser utilizada no desenvolvimento	Define linguagens e plataformas de desenvolvimento	Define a linguagem de programação a ser utilizada	Define linguagens e plataformas de desenvolvimento
		Define o projeto de interface do usuário		
Implementação	-	-	-	-
Testes	-	-	-	-
Refatoração	-	-	-	-
Documentação	-	-	-	Estabelece condições básicas de documentação

Fonte: elaborado pelo autor

Já com relação do impacto dos objetivos da pesquisa no projeto do software, há unanimidade entre os quatro casos estudados. Em todos eles foi constatado que o objetivo da pesquisa influencia diretamente a escolha de ferramentas e linguagens de programação pelos pesquisadores.

As linguagens de programação citadas encontram-se entre aquelas costumeiramente encontradas na literatura (FARHOODI et al., 2013; NGUYEN-HOAN; FLINT; SANKARANARAYANA, 2010; PRABHU et al., 2011; STORER, 2017; TANG, 2008). Entretanto, o fato de que a escolha se dá muito mais pelo objetivo da pesquisa do que pela experiência do pesquisador contraria o estudo de Nguyen-Hoan, Flint e Sankaranarayana (2010), o qual indica que as três principais razões para a escolha das linguagens são a compatibilidade entre plataformas, as funcionalidades e a experiência do desenvolvedor.

4.5.2. Disponibilidade do desenvolvedor

Em todos os casos citados, o desenvolvedor dedica a maior parte do seu tempo à codificação do software em si, realizando outras tarefas relacionadas ao desenvolvimento no restante do tempo disponível, reforçando o que já foi constatado em estudos anteriores (PRABHU et al., 2011; TANG, 2008). É importante notar que os desenvolvedores estudados dedicam de 10% a 30% do tempo total da pesquisa para o processo de desenvolvimento de software, o quê vai ao encontro de outros estudos na área (HANNAY et al., 2009; PRABHU et al., 2011).

Há indicativos de que essa maior dedicação dos pesquisadores à atividade de implementação pode fazer com que não seja dada a devida atenção às demais atividades, visto que os demais aspectos e fatores apontaram problemas nos requisitos e projeto. As evidências coletadas não permitiram estabelecer um paralelo entre os impactos deste aspecto e as demais atividades do processo de desenvolvimento de software científico.

4.5.3. Histórico de Pesquisa

Passando ao aspecto de histórico de pesquisa do desenvolvedor, o impacto sobre o processo de desenvolvimento se traduz na forma de manifestação da experiência e do perfil do pesquisador (Quadro 5).

O Grupo de Pesquisa A demonstrou dificuldades sob o ponto de vista de definição de requisitos, o que não ocorre nos Grupos de Pesquisa B, C e D. Em particular, no caso do C, isto se deve mais ao fato de que o objetivo da pesquisa realizada não demonstrou impacto sobre os requisitos, o que não exigiu do desenvolvedor uma atenção especial a esta atividade. Ainda no caso do Grupo de Pesquisa C, houve impacto direto na etapa de implementação.

Quadro 5 - Impacto do Histórico de pesquisa

Atividades do Desenvolvimento	Impacto do Histórico de pesquisa			
	Grupo de Pesquisa A	Grupo de Pesquisa B	Grupo de Pesquisa C	Grupo de Pesquisa D
Requisitos	Redefine requisitos	Auxilia a antecipação de riscos e restrições nos requisitos	-	Facilita a compreensão de requisitos
Projeto	Auxilia o pensamento em projetar para o reuso	Projeta com base em reuso	Interfere na escolha da linguagem	Projeta com base em reuso
	Impede a aplicação de novas práticas e padrões, pois falta confiança			
Implementação	-	Inserir tratamentos de controle	Busca de conhecimento para resolução de problemas em fontes específicas	-
Testes	-	-	-	Auxilia na elaboração de plano de testes
Refatoração	-	-	-	-
Documentação	-	Auxilia documentação do código-fonte	-	Auxilia documentação do código-fonte

Fonte: elaborado pelo autor

Em ambas as atividades (requisitos e implementação) isto não foi percebido no Grupo de Pesquisa B e D. A causa mais provável, e conforme citado na análise dos casos, tende a ser a grande experiência dos desenvolvedores na área (trabalham há mais de 20 anos como pesquisadores), ao contrário dos outros dois desenvolvedores, que atuam há menos de 10 anos. Esta experiência em pesquisa e conseqüentemente, em desenvolvimento dos softwares científicos, auxilia na diminuição dos riscos e problemas enfrentados, bem como na atividade de documentação.

4.5.4. Organização e obtenção dos dados

Em relação ao aspecto de organização e obtenção de dados, houve impacto nas principais áreas do processo de desenvolvimento de software científico, a saber: requisitos, implementação, projeto e testes (Quadro 6).

Quadro 6 - Impacto da Organização e obtenção dos dados

Atividades do Desenvolvimento	Impacto da Organização e obtenção dos dados			
	Grupo de Pesquisa A	Grupo de Pesquisa B	Grupo de Pesquisa C	Grupo de Pesquisa D
Requisitos	Auxilia a elicitação e especificação dos requisitos	Dificulta a elicitação e especificação dos requisitos	Implica a melhor especificação dos requisitos	-
Projeto	Direciona a escolha da ferramenta	-	-	-
Implementação	-	-	Indica possível falha	Aumento da complexidade do software
				Aumento do esforço de desenvolvimento
Testes	Redefinem requisitos	Possibilita a definição do protocolo de validação do software de antemão	Demonstram problema de confiabilidade	-
Refatoração	-	-	-	-
Documentação	-	-	-	-

Fonte: elaborado pelo autor

Todos os casos estudados (exceto o D) demonstraram influência sobre a definição de requisitos, sendo que a obtenção prévia dos dados é fator preponderante para a execução desta atividade. Em apenas um caso (Grupo de Pesquisa A) foi citada explicitamente a influência sobre a etapa de projeto do software, já que os dados modificam a escolha da ferramenta de desenvolvimento. Nos casos dos Grupos de Pesquisa C e D foi evidenciado maior impacto na atividade de implementação do software.

4.5.5. Progresso da pesquisa

O progresso da pesquisa tende a gerar influências positivas e negativas no processo de desenvolvimento de software científico. Nos quatro casos avaliados foi percebida a necessidade de diversos ajustes em todas as etapas de desenvolvimento, excetuando-se a documentação (Quadro 7).

Quadro 7 - Impacto do Progresso da pesquisa

Atividades do Desenvolvimento	Impacto do Progresso da pesquisa			
	Grupo de Pesquisa A	Grupo de Pesquisa B	Grupo de Pesquisa C	Grupo de Pesquisa D
Requisitos	-	Introduz novos requisitos	Prioriza requisitos	-
Projeto	Torna o software incompleto	-	Limita a definição da modularização do software	-
Implementação	-	Tratamento de situações adversas	Define modelo de implementação das funcionalidades (incremental)	Limita a correção de problemas de forma adequada
Testes	-	-	Define modelo de teste das funcionalidades (incremental)	Limita a realização de testes
Refatoração	Implica em melhoria de código para se adaptar a novas descobertas da pesquisa	-	-	-
Documentação	-	-	-	-

Fonte: elaborado pelo autor

Entretanto, o progresso da pesquisa pode auxiliar no refinamento de funcionalidades do software, eliminando aquelas que inicialmente seriam necessárias e possibilitando que o desenvolvedor volte sua atenção para outras mais importantes.

4.5.6. Pressão para publicação

A pressão para publicação (Quadro 8) exerce forte influência sobre a etapa de implementação e codificação do software necessário. Isso esteve presente nos

Grupos de Pesquisa A, C e D, nos quais foi indicada a necessidade de cortes em funcionalidades do software.

Quadro 8 - Impacto da Pressão para publicação

Atividades do Desenvolvimento	Impacto da Pressão para publicação			
	Grupo de Pesquisa A	Grupo de Pesquisa B	Grupo de Pesquisa C	Grupo de Pesquisa D
Requisitos	-	-	-	-
Projeto	Diminui a possibilidade de projeto da reusabilidade do software	Inclui condições de tratamento adicionais	Impede o projeto da interface	Torna o software incompleto
	Dificulta o projeto para reprodutibilidade	-	Impede o projeto da reusabilidade	-
	Torna o software incompleto	-	Torna o software incompleto	-
Implementação	Diminui a qualidade de código	Inclui condições de tratamento adicionais	Diminui a qualidade de código	Diminui a qualidade de código
Testes	Estabelece um nível de confiabilidade do software, para que não sejam necessários novos testes	-	-	-
Refatoração	-	-	-	-
Documentação	-	-	Elimina a documentação prevista	-

Fonte: elaborado pelo autor

Além disso, nestes casos foi relatado que o software serviu ao propósito da pesquisa, mas acabou por ficar incompleto de certa forma; a necessidade de algumas funcionalidades foi suprida pelo uso de ferramentas externas, apesar de estarem previstas no projeto do software.

Do ponto de vista da atividade de implementação, a diminuição da qualidade de código diz respeito às questões de legibilidade e boas práticas de programação que ficam prejudicadas devido à necessidade de se publicar os resultados.

No Grupo de Pesquisa B, por sua vez, não foi detectada influência sobre o processo de desenvolvimento. Uma das hipóteses a se considerar seria a de que o vasto histórico e experiência da pesquisadora poderia mitigar esse fator, entretanto o problema se manifesta no Grupo de Pesquisa D (o qual tem o mesmo nível de experiência).

4.5.7. Reconhecimento do trabalho de desenvolvimento

Com relação ao último aspecto analisado, o reconhecimento do desenvolvedor (Quadro 9), todos os quatro entrevistados possuem a visão compartilhada de que não existe reconhecimento do software produzido junto à comunidade de pesquisa, confirmando a percepção geral indicada por Goble (2014).

Quadro 9 - Impacto do Reconhecimento do trabalho de desenvolvimento

Atividades do Desenvolvimento	Impacto do Reconhecimento do trabalho de desenvolvimento			
	Grupo de Pesquisa A	Grupo de Pesquisa B	Grupo de Pesquisa C	Grupo de Pesquisa D
Requisitos	-	-	-	-
Projeto	Aumenta a preocupação com projeto estruturado	-	-	-
Implementação	Aumenta a qualidade percebida do código	Aumenta a preocupação com programação	Reforça a preocupação com confiabilidade do software	Aumenta a qualidade percebida do código
				Aumenta a preocupação com programação
Testes	-	-	-	-
Refatoração	-	-	-	-
Documentação	-	-	-	-

Fonte: elaborado pelo autor

Nos Grupos de Pesquisa A e B e D, os desenvolvedores concordam que o reconhecimento poderia auxiliar na melhoria do processo de desenvolvimento como um todo, a partir do momento que criaria condições para que mais pessoas da área de pesquisa se interessassem por esta atividade.

4.5.8. Aspectos complementares

Esperava-se que, pelo fato dos desenvolvedores dos Grupos de Pesquisa C e D terem maior conhecimento sobre aspectos de computação e de tecnologia da informação, haveria maior controle sobre a influência dos fatores do ambiente de pesquisa no processo de desenvolvimento. Entretanto, isso não foi constatado na análise dos resultados; pelo menos não em sua totalidade.

Mesmo tendo um processo de desenvolvimento mais estruturado que os demais casos, os pesquisadores dos Grupos de Pesquisa C e D têm dificuldades para lidar com a pressão para publicação dos resultados, incorrendo em corte de funcionalidades previstas e diminuição da qualidade do código.

Como outro ponto complementar, destaca-se o fato do pesquisador desenvolvedor no Grupo de Pesquisa C levantar dúvidas sobre até que ponto o software desenvolvido por alguns pesquisadores é confiável, já que na publicação do trabalho não houve a preocupação de questionamento sobre o software apresentado. Isto reforça a importância de iniciativas que permitam a reprodução da pesquisa utilizando-se do mesmo software, bem como maior atenção da comunidade acadêmica para a confiabilidade dos softwares utilizados (STODDEN; MIGUEZ, 2014).

Por fim, algumas etapas rotineiras de desenvolvimento de software não apresentaram evidências suficientes durante a análise dos casos. Etapas como documentação e refatoração foram citadas ao acaso e não de forma sistemática dentro de cada um dos processos de desenvolvimento. Esta constatação acompanha outros trabalhos já existentes, que indicam a baixa importância dada a estas atividades pelos desenvolvedores de software científico (CARVER et al., 2013; KOTESKA; MISHEV, 2013; SEGAL, 2007; TANG, 2008).

5.Considerações finais

A importância do uso de software e sistemas computacionais no apoio ao desenvolvimento e produção científica é cada vez mais notável. Sem o auxílio de softwares científicos diversas pesquisas se tornariam inviáveis.

Entretanto, a maneira como estes softwares são desenvolvidos ainda não se mostra adequada. Por vezes, os softwares são desenvolvidos pelos próprios pesquisadores, sem o devido conhecimento e sem que seja seguido um processo de desenvolvimento adequado e estruturado. Isto pode acarretar em problemas como a falta de confiabilidade nos resultados obtidos, a dificuldade na reprodução dos experimentos e o aumento da complexidade na evolução destes softwares.

Além disso, aspectos do ambiente de pesquisa colocam novos desafios para o processo de desenvolvimento de software científico, já que fatores como o objetivo da pesquisa envolvida e a pressão para a publicação dos resultados têm influência direta na construção destes produtos.

Dessa forma, o presente trabalho se propôs a investigar o impacto de sete destes fatores no processo de desenvolvimento de software científico. Para tanto, foram realizadas entrevistas com pesquisadores em um estudo de caso múltiplo.

Os resultados obtidos indicam que todos os aspectos estudados apresentam algum tipo de influência nas atividades do processo de desenvolvimento de software científico, tanto positivamente como negativamente.

O primeiro fator abordado foi o objetivo da pesquisa, que define o que será realizado na pesquisa científica. Do ponto de vista do software, o principal impacto se dá na definição dos requisitos e no projeto do software, sendo este aspecto de crucial importância nos quatro casos estudados.

O segundo fator analisado, disponibilidade do desenvolvedor, apresenta correlação direta com a atividade de implementação do software, sendo este fator o de maior importância para esta atividade.

Na análise do aspecto de histórico de pesquisa, nota-se reflexos nas etapas de requisitos e implementação. Importante notar que, à medida que há maior experiência em pesquisa científica por parte do desenvolvedor, os impactos tendem a ser minimizados.

Já na análise do impacto da organização e obtenção dos dados, constatou-se que há um impacto em grande parte do processo de desenvolvimento. Como estes

softwares são desenvolvidos para processar dados de experimentos e obter novos dados, era esperado que fosse um dos fatores mais importantes para o processo de desenvolvimento, com destaque especial para a atividade de testes.

A análise da influência do progresso da pesquisa, por sua vez, demonstra que é importante a adoção de um processo de desenvolvimento incremental, uma vez que este aspecto auxilia no refinamento do software desenvolvido à medida que a pesquisa ganha maturidade.

Do ponto de vista da pressão para publicação, as atividades que sofrem maior influência são as de projeto e implementação, que tendem a sofrer alterações e cortes devido ao tempo disponível para o desenvolvimento.

Por fim, o fator de reconhecimento do desenvolvedor tende a impactar o modo como os desenvolvedores projetam seu crescimento e evolução dentro da área de pesquisa, mostrando-se um aspecto de crucial importância para a melhoria do processo de desenvolvimento de software científico.

Não obstante, fatores não elencados para a análise proposta neste trabalho foram encontrados quando da avaliação dos resultados, como no caso da reusabilidade e da reprodutibilidade do software, o que indica uma limitação da presente pesquisa.

Os resultados deste trabalho tendem a contribuir para que melhorias do processo de desenvolvimento de software científico possam ser estudadas e propostas, a partir do entendimento do impacto dos fatores e das razões pelas quais os desenvolvedores de software científico tomam determinadas decisões.

Sob a ótica da gestão do processo de desenvolvimento, os resultados aqui obtidos contribuem para que projetos de desenvolvimento de software científico tenham como premissa a análise dos aspectos do ambiente de pesquisa, o que tende a minimizar riscos durante o desenvolvimento.

5.1. Trabalhos futuros

Considerando-se as restrições e premissas adotadas na condução deste trabalho, alguns trabalhos futuros puderam ser identificados.

Em um primeiro momento, sugere-se a aplicação da presente pesquisa em novos estudos de casos, de modo a possibilitar a comparação com os resultados aqui obtidos. Por outro lado, a análise dos fatores e aspectos do ambiente científico limitou-se a casos de desenvolvimento de software científico dentro da área das

ciências exatas e engenharias. Assim sendo, uma extensão do presente trabalho seria a investigação do impacto destes fatores em outras áreas de conhecimento como, por exemplo, as ciências biológicas e áreas correlatas, como a bioinformática.

Conforme destacado, alguns dos fatores e aspectos do ambiente de pesquisa que não foram selecionados para análise neste trabalho foram citados em um ou mais dos casos estudados. Dessa forma, novas pesquisas poderiam ser conduzidas considerando estes fatores adicionais.

Os resultados aqui alcançados também podem ser utilizados para a extensão de pesquisas em gestão de projetos de desenvolvimento de software científico, de modo a buscar o aprimoramento dos métodos e técnicas utilizados atualmente.

REFERÊNCIAS

- ACKROYD, K. S.; KINDER, S. H.; MANT, G. R.; MILLER, M. C.; RAMSDALE, C. A.; STEPHENSON, P. C. Scientific software development at a research facility. **IEEE Software**, v. 25, n. 4, p. 44–51, 2008.
- AHMED, Z.; ZEESHAN, S. Cultivating software solutions development in the scientific academia. **Recent Patents on Computer Science**, v. 7, n. 1, p. 54–66, 2014.
- ANASTAS, J. W. **Research design for social work and the human services**. 2. ed. [s.l.] Columbia University Press, 1999. 1999.
- BARDIN, L. Análise de conteúdo. 1. ed. São Paulo: Edições 70, 2011. 2011.
- BASILI, V. R.; CARVER, J.; CRUZES, D. S.; HOCHSTEIN, L.; HOLLINGSWORTH, J. K.; SHULL, F.; ZELKOWITZ, M. V. Understanding the High Performance Computing Community: A Software Engineer's Perspective. **IEEE Software**, v. 25, n. 4, p. 29–36, 2008.
- BONI, V.; QUARESMA, S. Aprendendo a entrevistar: como fazer entrevistas em Ciências Sociais. **Em Tese**, v. 2, n. 3, p. 68–80, 2005.
- CARVALHO, A. C. P. L. F.; BRAYNER, A.; LOUREIRO, A.; FURTADO, A. L.; STAA, A. V.; LUCENA, C. J. P.; DE SOUZA, C. S.; MEDEIROS, C. M. B.; LUCCHESI, C. L.; SILVA, E. S.; WAGNER, F. R.; SIMON, I.; WAINER, J.; MALDONADO, J. C.; DE OLIVEIRA, J. P. M.; RIBEIRO, L.; VELHO, L.; GONÇALVES, M. A.; BARANAUSKAS, M. C. C.; MATTOSO, M.; ZIVIANI, N.; NAVAU, P. O. A.; TORRES, R. S.; ALMEIDA, V. A. F.; W, M. J.; KOHAYAKAWA, Y. **Grandes Desafios da Pesquisa em Computação no Brasil – 2006 – 2016Sbc**. [s.l.: s.n.].
- CARVER, J. C.; KENDALL, R. P.; SQUIRES, S. E.; POST, D. E. **Software Development Environments for Scientific and Engineering Software: A Series of Case Studies**. Proceedings of the 29th international conference on Software Engineering. **Anais...2007**.
- CARVER, J.; HEATON, D.; HOCHSTEIN, L.; BARTLETT, R. Self-perceptions about software engineering: A survey of scientists and engineers. **Computing in Science and Engineering**, v. 15, n. 1, p. 7–11, 2013.
- CHARMAZ, K. **Constructing grounded theory: a practical guide through qualitative analysis**. 1. ed. London: SAGE Publications, 2006. 2006.
- COX, R. W.; REYNOLDS, R. C.; TAYLOR, P. A. AFNI and Clustering: False Positive Rates Redux. **bioRxiv**, p. 1–15, 2016.
- CRABTREE, C. A.; KORU, A. G.; SEAMAN, C.; ERDOGMUS, H. **An empirical characterization of scientific software development projects according to the Boehm and Turner model: A progress report**. Proceedings of the 2009 ICSE Workshop on Software Engineering for Computational Science and Engineering, SECSE 2009. **Anais...2009**.

DONOHO, D. L.; MALEKI, A.; SHAHRAM, M.; RAHMAN, I. U.; STODDEN, V. Reproducible research in computational harmonic analysis. **Computing in Science and Engineering**, v. 11, n. 1, p. 8–18, 2009.

DOORN, P.; AERTS, P.; LUSHER, S. **Research Software at the Heart of Discovery**. [s.l.], 2016.

EISENHARDT, K. M. Building Theories from Case Study Research. **Academy of Management Review**, v. 14, n. 4, p. 532–550, 1989.

EKLUND, A.; NICHOLS, T. E.; KNUTSSON, H. Cluster failure: Why fMRI inferences for spatial extent have inflated false-positive rates. **Proceedings of the National Academy of Sciences**, v. 113, n. 28, p. 7900–7905, 2016.

FARHOODI, R.; GAROUSI, V.; PFAHL, D.; SILLITO, J. Development of Scientific Software: a Systematic Mapping, a bibliometrics Study, and a Paper Repository. **International Journal of Software Engineering and Knowledge Engineering**, v. 23, n. 4, p. 463–506, 2013.

GERACI, A.; KATKI, F.; MCMONEGAL, L.; MEYER, B.; PORTEOUS, H. **IEEE Standard Computer Dictionary. A Compilation of IEEE Standard Computer Glossaries**. New York: IEEE Computer Society, 1991. 1991.

GOBLE, C. Better software, better research. **IEEE Internet Computing**, v. 18, n. 5, p. 4–8, 2014.

GUO, J.; LYSTER, P. M. **Bottom-up, Open, and Concurrent: A perspective on Development Processes for Scientific Software Systems**. [s.l.], 2000.

GUSTAFSSON, J. Single case studies vs. multiple case studies: A comparative study. **Academy of Business, Engineering and Science Halmstad University, Sweden**, p. 15, 2017.

HANNAY, J. E.; MACLEOD, C.; SINGER, J.; LANGTANGEN, H. P.; PFAHL, D.; WILSON, G. **How do scientists develop and use scientific software?** Proceedings of the 2009 ICSE Workshop on Software Engineering for Computational Science and Engineering, SECSE 2009. **Anais...2009**

HEATON, D.; CARVER, J. C. **Claims about the use of software engineering practices in science: A systematic literature review**. Information and Software Technology. **Anais...2015**

HEROUX, M. A.; WILLENBRING, J. M. **Barely sufficient software engineering: 10 Practices to improve your CSE software**. Proceedings of the 2009 ICSE Workshop on Software Engineering for Computational Science and Engineering, SECSE 2009. **Anais...2009**

HEROUX, M. A.; WILLENBRING, J. M.; PHENOW, M. N. **Improving the development process for CSE software**. Proceedings - 15th EUROMICRO International Conference on Parallel, Distributed and Network-Based Processing,

PDP 2007. **Anais...2007**

HEROUX, M. A. **Improving CSE Software Through Reproducibility Requirements**. Proceedings of the 4th International Workshop on Software Engineering for Computational Science and Engineering. **Anais...2011**

HOWISON, J.; HERBSLEB, J. D. **Scientific software production and collaboration**. CSCW. **Anais...2010**

HOWISON, J.; HERBSLEB, J. D. **Scientific software production: incentives and collaboration**. Proceedings of the ACM 2011 conference on Computer supported cooperative work. **Anais...2011**

IEEE. **ISO/IEC/IEEE 12207: Systems and software engineering — Software life cycle processes**. Geneva: ISO/IEC/IEEE, 2017. 2017.

KANEWALA, U.; BIEMAN, J. M. **Techniques for testing scientific programs without an oracle**. 2013 5th International Workshop on Software Engineering for Computational Science and Engineering, SE-CSE 2013 - Proceedings. **Anais...2013**

KELLY, D. A software chasm: Software engineering and scientific computing. **IEEE Software**, v. 24, n. 6, p. 120–119, 2007.

KELLY, D. Determining factors that affect long-term evolution in scientific application software. **Journal of Systems and Software**, v. 82, n. 5, p. 851–861, 2009.

KELLY, D. An Analysis of Process Characteristics for Developing Scientific Software. **Journal of Organizational and End User Computing**, v. 23, n. 4, p. 64–79, 2011.

KELLY, D. Scientific software development viewed as knowledge acquisition: Towards understanding the development of risk-averse scientific software. **Journal of Systems and Software**, v. 109, n. C, p. 50–61, 2015.

KO, A. J.; MYERS, B.; ROSSON, M. B.; ROTHERMEL, G.; SHAW, M.; WIEDENBECK, S.; ABRAHAM, R.; BECKWITH, L.; BLACKWELL, A.; BURNETT, M.; ERWIG, M.; SCAFFIDI, C.; LAWRENCE, J.; LIEBERMAN, H. The state of the art in end-user software engineering. **ACM Computing Surveys**, v. 43, n. 3, p. 1–44, 2011.

KOTESKA, B.; MISHEV, A. Software engineering practices and principles to increase quality of scientific applications. In: **ICT Innovations 2012**. New York: Springer Heidelberg, 2013. p. 245–254.

LI, Y.; NARAYAN, N.; HELMING, J.; KOEGEL, M. **A domain specific requirements model for scientific computing: NIER track**. 2011 33rd International Conference on Software Engineering (ICSE). **Anais...2011**

LIST, M.; EBERT, P.; ALBRECHT, F. Ten Simple Rules for Developing Usable Software in Computational Biology. **PLoS Computational Biology**, v. 13, n. 1, p. 1–5, 2017.

LOYNTON, S.; SLOAN, D.; BUREL, J. M.; MACAULAY, C. **Towards a project community approach to academic scientific software development.** e-science 2009 - Proceedings of the 2009 5th IEEE International Conference on e-Science Workshops. **Anais...2009**

MAXVILLE, V. **Preparing scientists for scalable software development.** Proceedings of the 2009 ICSE Workshop on Software Engineering for Computational Science and Engineering, SECSE 2009. **Anais...2009**

MENG, N. J.; KELLY, D.; DEAN, T. R. **Towards the Profiling of Scientific Software for Accuracy.** Proceedings of the 2011 Conference of the Center for Advanced Studies on Collaborative Research. **Anais...: CASCON '11.** Riverton, New Jersey: IBM Corp., 2011

MESH, E. S. **Supporting Scientific SE Process Improvement.** Proceedings - International Conference on Software Engineering. **Anais...2015**

MESH, E. S.; BURNS, G.; HAWKER, J. S. Leveraging expertise to support scientific software process improvement decisions. **Computing in Science and Engineering**, v. 16, n. 3, p. 28–34, 2014.

MESH, E. S.; HAWKER, J. S. **Scientific software process improvement decisions: A proposed research strategy.** 2013 5th International Workshop on Software Engineering for Computational Science and Engineering (SE-CSE). **Anais...2013**

MESH, E. S.; TOLAR, D. M.; HAWKER, J. S. **Exploring process improvement decisions to support a rapidly evolving developer base.** Proceedings of the 38th International Conference on Software Engineering Companion - ICSE '16. **Anais...2016**

MILLER, G. A Scientist's Nightmare: Software Problem Leads to Five Retractions. **Science (New York, N.Y.)**, v. 314, n. 5807, p. 1856–1857, 2007.

MONTEITH, J. Y.; MCGREGOR, J. D.; INGRAM, J. E. **Scientific Research Software Ecosystems.** Proceedings of the 2014 European Conference on Software Architecture Workshops - ECSAW '14. **Anais...2014**

MORRIS, C.; SEGAL, J. **Some challenges facing scientific software developers: The case of molecular biology.** 2009 Fifth IEEE International Conference on e-Science. **Anais...2009**

NANTHAAMORNPHONG, A.; MORRIS, K.; ROUSON, D. W. I.; MICHELSEN, H. A. **A Case Study: Agile Development in the Community Laser-Induced Incandescence Modeling Environment (CLiIME).** 5th International Workshop on Software Engineering for Computational Science and Engineering (SE-CSE). **Anais...2013**

NEUMANN, G. C.; ESPINA, H. A.; DAZA, V. P. Development of Scientific Software

and Practices for Software Development: A Systematic Literature Review. **JSW**, v. 12, n. 2, p. 114–124, 2017.

NGUYEN-HOAN, L.; FLINT, S.; SANKARANARAYANA, R. **A survey of scientific software development**. Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement - ESEM '10. **Anais...2010**

PATRICK, M.; ELDERFIELD, J.; STUTT, R. O. J. H.; RICE, A.; GILLIGAN, C. A. **Software testing in a scientific research group**. Proceedings of the 31st Annual ACM Symposium on Applied Computing. **Anais...2016**

PRABHU, P.; KIM, H.; OH, T.; JABLIN, T. B.; JOHNSON, N. P.; ZOUFALY, M.; RAMAN, A.; LIU, F.; WALKER, D.; ZHANG, Y.; GHOSH, S.; AUGUST, D. I.; HUANG, J.; BEARD, S. **A survey of the practice of computational science**. 2011 International Conference for High Performance Computing, Networking, Storage and Analysis (SC). **Anais...2011**

QUEIROZ, F.; SPITZ, R. The Lens of the Lab: Design Challenges in Scientific Software. **The International Journal of Design Management and Professional Practice**, v. 10, n. 3, p. 17–45, 2016.

RIEDEL, M.; STREIT, A.; WOLF, F.; LIPPERT, T.; KRANZLMÜLLER, D. **Classification of different approaches for e-science applications in next generation computing infrastructures**. 2008 IEEE Fourth International Conference on eScience. **Anais...2008**

ROBSON, C. **Real World Research: A Resource for Social Scientists and Practitioner-Researchers**. 2. ed. Malden, Massachusetts: Blackwell Publishers Inc, 2002. 2002.

SANDERS, R.; KELLY, D. Dealing with risk in scientific software development. **IEEE Software**, v. 25, n. 4, p. 21–28, 2008.

SEAMAN, C. B. Qualitative methods in empirical studies of software engineering. **IEEE Transactions on Software Engineering**, v. 25, n. 4, p. 557–572, 1999.

SEGAL, J. When software engineers met research scientists: A case study. **Empirical Software Engineering**, v. 10, n. 4, p. 517–536, 2005.

SEGAL, J. **Some problems of professional end user developers**. Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing. **Anais...2007**

SEGAL, J. **Some challenges facing software engineers developing software for scientists**. Proceedings of the 2009 ICSE Workshop on Software Engineering for Computational Science and Engineering, SECSE 2009. **Anais...2009**

SEGAL, J.; MORRIS, C. Developing scientific software. **IEEE Software**, v. 25, n. 4, p. 18–20, 2008.

SLETHOLT, M. T.; HANNAY, J.; BENESTAD, H. C.; LANGTANGEN, H. P.; PFAHL, D. **A Literature Review of Agile Practices and Their Effects in Scientific Software Development**. Proceedings of the 4th International Workshop on Software Engineering for Computational Science and Engineering. **Anais...**2011

SLETHOLT, M. T.; HANNAY, J. E.; PFAHL, D.; LANGTANGEN, H. P. What Do We Know about Scientific Software Development's Agile Practices? **Computing in Science & Engineering**, v. 14, n. 2, p. 24–37, mar. 2012.

SMITH, S. **Systematic development of requirements documentation for general purpose scientific computing software**. Proceedings of the IEEE International Conference on Requirements Engineering. **Anais...**2006

SMITH, S.; LAI, L. **A New Requirements Template for Scientific Computing**. Proceedings of SREP'05. **Anais...**Paris: 2005

STODDEN, V.; MIGUEZ, S. Best Practices for Computational Science: Software Infrastructure and Environments for Reproducible and Extensible Research. **Journal of Open Research Software**, v. 2, n. 1, p. 1–6, 2014.

STORER, T. Bridging the Chasm: A Survey of Software Engineering Practice in Scientific Programming. **ACM Computing Surveys**, v. 50, n. 4, p. 32, 2017.

TANG, J. **Developing Scientific Computing Software: Current Processes and Future Directions**. [s.l.] McMaster University, 2008.

TERREL, A.; TOBIS, M.; THIRUVATHUKAL, G. K. Scientific software communities. **Computing in Science and Engineering**, v. 17, n. 1, p. 8–10, 2015.

WAGNER, M. **Evolution from a scientific application to an applicable product**. Proceedings of the European Conference on Software Maintenance and Reengineering, CSMR. **Anais...**2007

WILSON, G.; ARULIAH, D. A.; BROWN, C. T.; HONG, N. P. C.; DAVIS, M.; GUY, R. T.; HADDOCK, S. H. D.; HUFF, K.; MITCHELL, I. M.; PLUMBLEY, M.; WAUGH, B.; WHITE, E. P.; WILSON, P. Best Practices for Scientific Computing. **PLOS Biology**, v. 12, n. 1, p. 1–7, 2014.

YIN, R. K. **Case Study Research: Design and Methods (Applied Social Research Methods)**. 4. ed. Thousand Oaks, California: Sage Publications Ltd, 2008. 2008.

APÊNDICE A – E-mail de Apresentação

Prezado Pesquisador(a),

Meu nome é Denis Bruno Virissimo, e sou aluno do curso de Especialização em Gestão de Tecnologia da Informação no Instituto Federal de Educação, Ciência e Tecnologia de São Paulo.

Minha monografia de conclusão de curso busca compreender e analisar como os fatores e aspectos do ambiente de desenvolvimento de software científico podem impactar no processo de desenvolvimento de software científico, com o intuito de auxiliar a identificação de pontos de melhoria destes processos.

Assim, gostaria de requisitar sua participação em um estudo de caso a respeito de sua experiência, a ser realizado por meio de uma entrevista presencial. De acordo com as regras de proteção e privacidade definidas nesta pesquisa, sua pessoa não será identificada.

Desde já, agradeço sua atenção e colaboração.

Atenciosamente,

Denis Bruno Virissimo

APÊNDICE B – Roteiro de Perguntas

1. Qual o tempo de experiência no desenvolvimento de softwares científicos?
2. Como se dá o processo de desenvolvimento do software envolvido em sua pesquisa?
3. Como você obtém recursos e conhecimentos necessários ao desempenho destas atividades?
4. De que forma você divide seu tempo entre as atividades de pesquisa e de desenvolvimento de software?
5. De que forma os dados necessários (e sua obtenção) à realização da pesquisa influenciam no desenvolvimento do software?
6. De que forma os objetivos da pesquisa influenciam na forma como você desenvolve o software?
7. De que forma a pressão pela publicação da pesquisa influencia na forma como você desenvolve o software?
8. De que forma o andamento da pesquisa influencia na forma como você desenvolve o software?
9. Como se dá o reconhecimento pelo software por você desenvolvido?