



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE RUSSAS
CURSO DE GRADUAÇÃO EM ENGENHARIA DE SOFTWARE

CARLOS HENRIQUE MELO DA SILVA

**CATALOGAÇÃO DE PROBLEMAS E DOCUMENTAÇÃO DE PADRÕES DE
SOFTWARE PARA O CONTEXTO DE COMUNICAÇÃO INTERPESSOAL**

RUSSAS
2021

CARLOS HENRIQUE MELO DA SILVA

CATALOGAÇÃO DE PROBLEMAS E DOCUMENTAÇÃO DE PADRÕES DE
SOFTWARE PARA O CONTEXTO DE COMUNICAÇÃO INTERPESSOAL

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Software do Campus de Russas da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Software.

Orientadora: Profa. Dra. Anna Beatriz dos Santos Marques.

Coorientadora: Profa. Dra. Valéria Lelli Leitão Dantas

RUSSAS

2021

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

S579c Silva, Carlos Henrique Melo da.
Catalogação de problemas e documentação de padrões de software para o contexto de comunicação interpessoal / Carlos Henrique Melo da Silva. – 2021.
70 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Russas, Curso de Engenharia de Software, Russas, 2021.

Orientação: Prof. Dr. Anna Beatriz dos Santos Marques.

Coorientação: Prof. Dr. Valéria Lelli Leitão Dantas.

1. Padrões de software. 2. Problemas na Comunicação. 3. Comunicação Interpessoal. 4. Stakeholders. I. Título.

CDD 005.1

CARLOS HENRIQUE MELO DA SILVA

CATALOGAÇÃO DE PROBLEMAS E DOCUMENTAÇÃO DE PADRÕES DE
SOFTWARE PARA O CONTEXTO DE COMUNICAÇÃO INTERPESSOAL

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Software do Campus de Russas da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Software.

Aprovada em:

BANCA EXAMINADORA

Profa. Dra. Anna Beatriz dos Santos Marques (Orientadora)
Universidade Federal do Ceará (UFC)

Profa. Dra. Valéria Lelli Leitão Dantas (Coorientadora)
Universidade Federal do Ceará (UFC)

Prof. Ms. José Osvaldo Mesquita Chaves
Universidade Federal do Ceará (UFC)

Dedico este trabalho a minha família. Nilsa Melo, Antônio Carlos e Nayana Melo. Muito obrigado por sempre estarem presentes na minha vida.

AGRADECIMENTOS

Primeiramente a Deus, por me proporcionar saúde e forças para enfrentar as dificuldades.

À minha mãe Nilsa Melo e ao meu pai Antônio Carlos por toda a educação que me deram e por sempre proporcionar o apoio e incentivo necessário nas horas difíceis.

À minha irmã Nayana Melo que sempre foi fonte de inspiração na minha vida estudantil. Agradeço aos meus pais e a minha irmã por toda força e por nunca deixar desistir. Amo vocês.

Ao meu amigo de infância José Augusto, por todos os momentos e histórias divertidas.

À minha professora Dra. Valéria Lelli Leitão Dantas por compreender as minhas dificuldades pessoais. Muito obrigado por todo esforço e empenho dedicado a este trabalho.

À minha professora Dra. Anna Beatriz dos Santos Marques por todo apoio necessário.

“O impossível não é um fato, impossível é uma opinião.”

(Muhammad Ali)

RESUMO

No ambiente de desenvolvimento de software, vários aspectos utilizados são modificáveis, entre eles: a utilização ou não de ferramentas para o gerenciamento de projetos, utilização de processos de desenvolvimento (e.g., SCRUM e XP). Um dos aspectos que está contido em todos os projetos, independente da metodologia ou tecnologias utilizadas, é o fator humano e suas inter-relações. Estas inter-relações ocorrem por meio da comunicação interpessoal, a mesma pode ser alvo de vários problemas que podem impactar no andamento do projeto. Objetiva-se neste trabalho, fazer um estudo para identificação de problemas recorrentes na comunicação interpessoal em projetos de software, bem como as suas soluções. Para tal fim, foi realizado um estudo bibliográfico para a descoberta de problemas e soluções comunicacionais, realizou-se também uma identificação dos aspectos em comuns entre as problemáticas detectadas, e com base neles, uma categorização foi feita. Posteriormente, baseado na identificação dos problemas e suas soluções, foi analisado se os mesmos podem ser documentados na forma de padrões de software. Como resultados, foram identificados vários problemas que impactam a comunicação durante o desenvolvimento de um projeto de software, entre eles: (i) equipes distribuídas e que utilizam metodologias ágeis possuem a comunicação afetada devido à pouca documentação e constante troca de informações presentes na metodologia; (ii) diferenças linguísticas ocasionam errôneo entendimento das informações; (iii) a falta de conhecimento do cliente a respeito do que o produto deve possuir ocasiona a omissão de informações; e (iv) cronograma com curto tempo planejado impacta negativamente a comunicação. Além dos problemas identificados, foram apresentadas neste trabalho algumas soluções encontradas na literatura bem como uma categorização dos mesmos de acordo com os seguintes critérios: problemas ambientais, verbais, pessoais e de decisão de projeto. Neste trabalho, também foram documentadas três soluções no formato de padrão de software, são eles: (i) *Communication Owner*, que facilita a comunicação presente entre o cliente e os membros da equipe em projetos que utilizam metodologias ágeis; (ii) Dependência, priorização antes da execução, que inclui os membros da equipe na atividade de priorização de tarefas com base na dependência de artefatos; e (iii) *Unproductive Stand-Up Meetings*, que indica melhores práticas para reuniões de *stand-up meeting* com o propósito de evitar perdas comunicacionais.

Palavras-chave: Padrões de software; Problemas na Comunicação; Comunicação Interpessoal; Stakeholders.

ABSTRACT

In the software development environment, different aspects used are changeable, between them: the use or not of tools for project management, use of development processes (e.g., SCRUM and XP). One of the aspects that are contained in all projects, regardless of the methodology or the technologies used, is the human factor and its interrelationships. These interrelationships happen through interpersonal communication, it can be the target of different problems that can impact the progress of the project. The objective of this work is to do a study to identify recurrent problems in interpersonal communication in software projects, as well as their resolutions presented for their correction. There was also an identification of common aspects between the problems presented, and based on them, a categorization was made. Subsequently, based on the identification of problems and their solutions, it was analyzed whether they can be documented in the form of software patterns. As a result, different problems that impact communication during the development of a software project were identified, including: (i) distributed teams that use agile methodologies have affected communication due to the lack of documentation and dynamic exchange of information present in the methodology; (ii) linguistic differences cause misunderstanding of information; (iii) the customer's lack of knowledge about what the product must have caused the omission of information; and (iv) schedule with short planned time negatively impacts communication. In addition to the identified problems, some solutions found in the literature were presented in this work, as well as a categorization of them according to the following criteria: environmental problems, verbal, personal, and project decision problems. In this work, three solutions were also documented in the software pattern format, are: (i) Communication Owner, which facilitates communication between the client and team members in projects that use agile methodologies; (ii) Dependency, prioritization before execution, which includes team members in the task prioritization activity based on artifact dependency; and (iii) Unproductive Stand-Up Meetings, which indicate best practices for stand-up meetings to avoid communication losses.

Keywords: Software Patterns; Communication Problems; Interpersonal Communication; Stakeholders.

LISTA DE FIGURAS

Figura 1 – Processo comunicacional proposto por Osgood e Schramm	19
Figura 2 – Fatores de influência em equipes terceirizadas e separadas geograficamente .	26
Figura 3 – Atividades do procedimento metodológico	38
Figura 4 – Modelo Comunicacional	51
Figura 5 – Exemplificação da estruturação das atividades de um projeto	56

LISTA DE TABELAS

Tabela 1 – Comparativo entre os trabalhos relacionados	35
Tabela 2 – Resultados da aplicação da string na primeira etapa de seleção	39
Tabela 3 – Trabalhos selecionados	40
Tabela 4 – Problemas, indicações de soluções e categorização	45
Tabela 5 – Cálculo da pontuação da Tarefa 1	57
Tabela 6 – Cálculo da pontuação da Tarefa 2	57
Tabela 7 – Cálculo da pontuação da Tarefa 3	58
Tabela 8 – Priorização das tarefas	58

LISTA DE QUADROS

Quadro 1 – Fórmula para priorização de tarefas	55
--	----

LISTA DE ABREVIATURAS E SIGLAS

IEEE	Institute of Electrical and Electronic Engineers
ISO	International Organization for Standardization
PMBOK	Project Management Body of Knowledge
PLOP	Pattern Languages of Programs

SUMÁRIO

1	INTRODUÇÃO	14
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	Gerência de Projetos	17
2.2	Gerência de Comunicação	18
2.2.1	<i>Comunicação Interpessoal</i>	18
2.2.2	<i>Processo de Comunicação</i>	19
2.2.3	<i>Boa Comunicação</i>	19
2.2.4	<i>Padrões de Software</i>	20
3	TRABALHOS RELACIONADOS	22
3.1	Guias e Processos de Comunicação	22
3.2	Padrões de Software	30
3.3	Comparativo	35
4	OBJETIVOS	37
4.1	Objetivo Geral	37
4.2	Objetivos Específicos	37
5	PROCEDIMENTOS METODOLÓGICOS	38
5.1	Estudo Bibliográfico	38
5.1.1	<i>Estratégia de Pesquisa</i>	39
5.1.2	<i>Análise e Critérios de Seleção</i>	40
5.2	Listagem de Problemas e Soluções	43
5.3	Categorização dos Problemas Identificados	43
5.4	Documentação das Soluções no formato de padrões de software	44
6	RESULTADOS	45
6.1	Problemas e Soluções Recorrentes a Comunicação	45
6.2	Padrões de Software	48
6.2.1	<i>Primeiro Padrão: Communication Owner</i>	49

6.2.2	<i>Segundo Padrão: Team and Task Prioritization</i>	54
6.2.3	<i>Terceiro Padrão: Unproductive Stand-Up Meetings</i>	60
7	CONCLUSÃO E TRABALHOS FUTUROS	65
	REFERÊNCIAS	66

1 INTRODUÇÃO

O fator humano dentro do ambiente de desenvolvimento de software é algo primordial. O processo de criação de um produto de software pode ser observado como um conjunto de interações sociais entre as pessoas envolvidas no projeto, e dentre diversos fatores que colaboram para o seu sucesso, um dos seus principais é a relação dos fatores humanos (TENENBERG, 2008).

De acordo com Braham (1997) os envolvidos em um projeto não podem agir de forma independente, eles terão que trabalhar com diferentes pessoas e de diferentes setores para que o projeto tenha sucesso. Refletindo da mesma forma, Kumar e Wallace (2014) expressam que a qualidade da comunicação interpessoal entre os envolvidos de um projeto de software é de extrema importância para o sucesso ou fracasso do produto, e que devido a sua intangibilidade e grande mutabilidade exigem enorme atenção na comunicação oral e escrita.

A comunicação interpessoal dentro deste ambiente possui duas vertentes principais: a comunicação com o cliente, que é o contratante do serviço de desenvolvimento de um software, ou o próprio indivíduo que será o usuário da aplicação, ambos são os responsáveis por expor suas necessidades; e a comunicação entre os membros da equipe de desenvolvimento (DAMIAN, 2001). Da mesma forma, esta comunicação pode ocorrer de forma verbal ou não verbal, a comunicação verbal é aquela em que por meio de palavras, sejam escritas ou faladas, transmitem alguma informação, já a comunicação não verbal, as transmissões de informações ocorrem por meio de ações corporais (ALY; TAPUS, 2013).

A comunicação com os envolvidos no projeto, os chamados *stakeholders*, pode ser alvo de problemas que impactam no andamento do projeto (WU, 2012). Por exemplo, a presença de ruídos na troca de informações, ruídos estes que podem ser ocasionados por ações (e.g., uma porta abrir em um ambiente onde está ocorrendo uma reunião) que impactam na atenção dos envolvidos na comunicação.

Com a possibilidade de ocorrência destes problemas na comunicação, a organização deve ter um plano de comunicação para os seus projetos de desenvolvimento de software. Este plano de comunicação deve identificar todas as atividades necessárias para as trocas de informações dentro do projeto, é neste plano também que são pensadas soluções como medida de contingência destes problemas, estas medidas devem ser documentadas em um plano de contingência do projeto. Plano de contingência é o documento responsável por indicar uma solução a ser tomada caso algum problema atrapalhe o andamento do projeto (PMBOK, 2017). É na gerência de comunicação em que o plano de comunicação é definido.

O ambiente de estudo da gerência de comunicação possui um número considerável de pesquisas, e que possuem a abordagem de problemas e soluções comunicacionais que ocorrem entre os fatores humanos. Alguns dos trabalhos que estudam esta temática são os propostos por (SANTOS *et al.*, 2014), (MAK; KRUCHTEN, 2006) e (STRAY; MOE; SJOBERG, 2020).

O trabalho proposto por Santos *et al.* (2014) propõe uma solução comunicacional para equipes de software distribuídas que utilizam metodologias ágeis. A problemática apresentada é que a comunicação é prejudicada pela falta de documentação e dinamismo das informações presentes em metodologias ágeis.

Já o trabalho de Mak e Kruchten (2006) apresenta uma solução para a falta de comunicação na atividade de priorização de tarefas, como solução ele apresenta uma fórmula simples de matemática que leva em consideração a participação dos membros da equipe para a decisão de priorização de tarefas.

Pensando na mesma linha de identificar problemas comunicacionais e propor solução, o trabalho de Stray, Moe e Sjoberg (2020) identifica problemas presentes na atividade de *daily stand-up meeting* que impactam na comunicação entre os membros, e apresenta soluções para estes problemas, algumas delas são: a prática da reunião logo antes do horário de almoço e distribuição igualitária do tempo de fala entre os participantes da reunião.

Apesar da grande quantidade de trabalhos que abordam problemas e soluções comunicacionais, segundo Monasor *et al.* (2013) os trabalhos presentes na literatura, apresentam soluções de forma muito simplista, enquanto organizações necessitam de soluções mais bem estruturadas, soluções a nível de padrões. Refletindo sobre isso, as soluções apresentadas por (SANTOS *et al.*, 2014), (MAK; KRUCHTEN, 2006) e (STRAY; MOE; SJOBERG, 2020) foram documentadas no presente trabalho de conclusão de curso no formato de padrão de software.

O objetivo deste presente trabalho é identificar um conjunto de problemas recorrentes que impactam na comunicação entre os *stakeholders*, e as suas soluções, através de uma pesquisa na literatura. Com base na identificação dos problemas e suas soluções aplicadas no ambiente de desenvolvimento de software, foi analisado se os mesmos podem ser documentados na forma de padrões de software. Posteriormente também ocorreu uma identificação dos aspectos comuns entre as problemáticas apresentadas, e com base neles, uma categorização foi feita. Tendo em vista o objetivo, as questões de pesquisa investigadas neste trabalho são as seguintes:

- Q1: Quais são os problemas recorrentes à comunicação interpessoal no ambiente de desenvolvimento de software?
- Q2: Quais são as soluções adotadas para os problemas de comunicação identificados?
- Q3: Esses problemas e suas soluções podem ser documentadas na forma de padrões de software?

A metodologia utilizada neste trabalho, está subdividida em quatro partes, são elas: (i) um estudo bibliográfico para identificar problemas e soluções; (ii) listagem dos problemas e soluções identificadas; (iii) categorização dos problemas; e (iv) documentação no formato de padrão das soluções mais bem descritas nos trabalhos identificados.

Este trabalho está organizado em sete seções descritas a seguir:

- Seção 1, já apresentada, mostrou a introdução do trabalho.
- Seção 2 aborda a fundamentação teórica, que descreve conceitos importantes para a compreensão do leitor sobre a área de pesquisa deste trabalho.
- Seção 3 aborda os trabalhos relacionados e apresenta uma descrição do que é proposto por eles, bem como a diferenciação desses trabalhos comparados a esta proposta de pesquisa de conclusão de curso.
- Seção 4 apresenta os objetivos gerais e específicos da pesquisa.
- Seção 5 descreve a metodologia utilizada neste trabalho.
- Seção 6 apresenta os resultados obtidos.
- Seção 7 apresenta a conclusão e os trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção será exposto o conceito de algumas palavras que são importantes para a compreensão do leitor sobre a área de pesquisa deste trabalho.

2.1 Gerência de Projetos

As etapas típicas do processo de desenvolvimento de software são:

- Etapa de requisitos, é nesta etapa em que são compreendidas as necessidades do cliente com relação ao seu desejo do desenvolvimento do produto de software. Com base nessas necessidades, são definidas as funcionalidades do sistema.
- Etapa de projeto, é nesta etapa em que são pensadas possíveis soluções para a construção do produto de software. Também é nesta etapa que são definidos aspectos como: a linguagem de programação que será utilizada; a definição de ferramentas que serão usadas no desenvolvimento do software.
- Etapa de teste, é nesta etapa em que são aplicados testes para a identificação de problemas no produto de software. Os testes possuem como intuito, evitar que problemas cheguem juntos ao software na entrega para o cliente.
- Etapa de manutenção. Após a entrega do software para o cliente, algumas modificações podem ser necessárias, tais como: a correção de erros identificados pelo usuário da aplicação, ou mesmo o surgimento da necessidade de uma nova funcionalidade. É nesta etapa que ocorre a melhoria destes aspectos.

Na execução destas etapas, são necessários recursos, sejam eles humanos ou materiais (e.g., gravador de voz para a elicitação de requisitos; atribuição da responsabilidade de aplicação de teste, em que deverá ser executada por um membro da equipe). Segundo Camargo (2018), a gerência de projetos é uma forma de administrar todos os recursos envolvidos em um projeto.

Dessa forma, a gerência de projetos no âmbito do desenvolvimento de software, cujo encarregado por esta atividade é o gerente de projetos, é responsável por controlar todas as etapas que ocorrem em todo o processo de desenvolvimento de um software, garantindo assim que ocorram da melhor forma possível. É na gerência de projetos que é praticado a gerência de comunicação.

2.2 Gerência de Comunicação

Dentre as várias vertentes de estudo do PMBOK (2017) no ambiente de desenvolvimento de software, uma delas é o gerenciamento de comunicação. Esse trabalho define a gerência de comunicação como sendo um conjunto de processos que buscam garantir que as trocas de mensagens ocorram de modo que se consiga alcançar a satisfação dos *stakeholders*.

A gerência de comunicação, ainda segundo o PMBOK (2017), é o conjunto dos seguintes processos:

- Planejamento do Gerenciamento das Comunicações: é o processo responsável por identificar todas as atividades necessárias para a troca de informações em um projeto de software.
- Gerenciamento das Comunicações: é o processo responsável por garantir a não perda das informações trocadas entre os *stakeholders*. Também é neste processo que é definido a forma de arrecadação das informações do projeto.
- Monitoramento das Comunicações: é o processo responsável por verificar se as informações estão sendo trocadas da maneira planejada. Neste processo são identificados os possíveis problemas recorrentes à comunicação, e com base nesta identificação, deve-se aplicar a sua respectiva solução.

Dessa forma, o gerenciamento de comunicação dentro do ambiente de desenvolvimento de software objetiva gerenciar as atividades de comunicação interpessoal entre os *stakeholders*.

2.2.1 Comunicação Interpessoal

A comunicação interpessoal, é a troca de mensagens verbais ou não verbais por duas ou mais pessoas, em que esta comunicação possui os seguintes atores: o emissor, que é o responsável por transmitir uma mensagem; e o receptor, que é o responsável por receber esta mensagem e decodificá-la (PMBOK, 2017). A decodificação de uma mensagem é o ato de interpretar o que o emissor deseja transmitir.

Esta comunicação pode ocorrer de forma verbal ou não verbal. A comunicação verbal é aquela em que por meio de palavras, sejam escritas ou faladas, transmitem alguma informação. Já a comunicação não verbal, é aquela que as informações são transmitidas por meio de aspectos e características apresentadas pelos fatores humanos envolvidos nesta

comunicação (e.g, uma expressão facial pode demonstrar satisfação ou insatisfação sobre o algo que está sendo comunicado).

No ambiente de desenvolvimento de um software, esta comunicação possui alguns envolvidos, são eles: o cliente, que é o contratante pelo serviço de desenvolvimento de software; o possível usuário da aplicação; os membros da equipe de desenvolvimento. Estes envolvidos utilizam o processo de comunicação para realizar esta comunicação interpessoal.

2.2.2 Processo de Comunicação

O processo de comunicação é o conjunto de atividades necessárias para a troca de mensagens, ou seja, para a emissão e recebimento de informações (BERLO, 2003). A comunicação interpessoal utiliza o processo de comunicação. Uma representação deste processo é demonstrada na (Figura 1):

Figura 1 – Processo de comunicação proposto por Osgood e Schramm



Fonte – Adaptado de Oliveira (2010) traduzido de McQuail; Windahl, 1993, p.19.

Este processo demonstrado na (Figura 1) proposto por Osgood e Schramm, demonstra que os aspectos humanos envolvidos em uma comunicação, praticam a execução de duas atividades principais, são elas: codificação e decodificação. A atividade de codificação ocorre de maneira que o emissor da mensagem, pensa e faz uma estruturação da informação que deseja enviar antes de submetê-la ao receptor. Já a atividade de decodificação, o receptor recebe esta informação e intuitivamente tenta compreendê-la. É de extrema importância que os envolvidos neste processo mantenham uma boa comunicação.

2.2.3 Boa Comunicação

Para que ocorra uma boa comunicação entre os envolvidos no processo de comunicação, o emissor deve expressar claramente o que deseja emitir como mensagem, e o

receptor deve decodificar esta mensagem, compreendendo exatamente o que o emissor deseja transmitir. A interferência de um problema recorrente a comunicação pode acarretar ao receptor o entendimento errôneo desta mensagem, ou seja, gerando um entendimento não desejado pelo emissor da informação. De acordo com esta problemática apresentada, viabiliza-se o uso de padrões de software voltados para a resolução de problemas comunicacionais.

2.2.4 Padrões de Software

No ambiente de Engenharia de Software, um padrão de software é uma descrição detalhada de uma problemática recorrente no ambiente de desenvolvimento de software, e com base nesta problemática propõe uma solução para a mesma (ALEXANDER, 1977). A existência de um padrão, possui como contribuição a possibilidade de outras pessoas reutilizarem esta solução já existente.

Na temática de pesquisa de problemas recorrente a comunicação interpessoal, alguns trabalhos na literatura ao expressarem a existência de um destes problemas, acabam contribuindo também com as suas soluções utilizadas pelos autores.

Segundo Vlissides (1997) citado por Santos (2004), padrões proporcionam quatro benefícios importantes, são elas:

- A possibilidade de adquirir conhecimento compartilhado por profissionais experientes.
- A facilitação da comunicação entre os desenvolvedores.
- A facilitação de entender a estrutura de um software proporcionada aos engenheiros de software.
- A facilitação da prática de uma possível reestruturação do software.

Ainda segundo Santos (2004), a documentação de um padrão é feita através da descrição de suas seções, chamadas de campos. Dessa forma, os campos essenciais de um padrão são:

- Nome: nome padrão, este nome deve ser intuitivo e deve expressar a problemática que ele propõe a solução.
- Contexto: descrição de onde a problemática costuma ocorrer.
- Problema: descrição da problemática em que será proposto a solução.
- Solução: descrição de como o problema pode ser resolvido.

- Forças: descrição de pontos positivos e negativos proporcionado pela aplicação do padrão.
- Usos conhecidos: descrição de exemplificações da aplicação do padrão.
- Contexto resultante: descrição dos aspectos resultantes proporcionados ao sistema após a aplicação do padrão.
- Padrões relacionados: descrição da relação deste padrão proposto com outros padrões que resolvem a mesma problemática.

O campo “Usos conhecidos” é vital para viabilizar a aplicação de um padrão. Na literatura, sugere-se que cada padrão deve apresentar pelo menos três usos conhecidos.

3 TRABALHOS RELACIONADOS

Nesta seção são apresentados os trabalhos que estimularam e inspiraram este projeto de pesquisa. Cada trabalho apresentado possui a descrição dos seus objetivos de pesquisa bem como os seus resultados.

Esta seção está subdividida em três subseções, são elas: trabalhos relacionados que possuem abordagem de problemas e soluções comunicacionais (subseção 3.1 de Guias e Processos de Comunicação); trabalhos relacionados que documentaram padrões e que auxiliaram na estruturação e escrita dos padrões apresentados neste trabalho (subseção 3.2 de Padrões de Software) e apresentação da comparação desta pesquisa com os trabalhos relacionados (subseção 3.3 Comparativo dos trabalhos relacionados).

3.1 Guias e Processos de Comunicação

O PMBOK (2017) é um guia de boas práticas para o ambiente de desenvolvimento de software, ele possui um conjunto de indicações de processos de boas práticas em todas as etapas de desenvolvimento do software (CRUZ, 2013). Dentre as áreas de pesquisa deste trabalho, encontra-se a abordagem da gerência de comunicação entre os *stakeholders* do projeto. Nele, esta gerência de comunicação possui dois aspectos principais, são eles: primeiramente deve-se planejar uma estratégia para que a comunicação entre os *stakeholders* ocorra da melhor forma possível, e conseqüente à atividade anterior, deve-se executar as atividades planejadas para as comunicações. Este planejamento da comunicação deve ser pensado de acordo com a definição das atividades necessárias para a troca de informações entre os *stakeholders* dentro do ambiente de desenvolvimento do projeto de software.

O artigo de Lu, Liu e Liu (2009) faz uma abordagem da importância da comunicação no ambiente organizacional do desenvolvimento de software. Nesse trabalho foi feita uma revisão literária sobre o gerenciamento da comunicação, e por meio dessa pesquisa, foi possível identificar um conjunto de 48 possíveis atividades que ocorrem na comunicação em projetos de software, dentre elas, destacam-se: a prática da comunicação para a motivação da equipe e o compartilhamento de informações entre os membros da equipe de desenvolvimento. O objetivo do autor é identificar a importância da comunicação em pequenos e médios projetos de software, e quais destas atividades identificadas pelo autor na literatura, possuem maior impacto no sucesso de projetos. Nesse trabalho, foi aplicado um questionário nas organizações selecionadas, para identificar quais destas 48 atividades

impactam de forma mais severa no sucesso dos projetos de desenvolvimento de software. Como resultado, foram identificadas duas atividades que possuem maior impacto, são elas: informar ao cliente sobre o plano de trabalho do desenvolvimento de software, e comunicar ao cliente sobre possíveis problemas ocorridos no projeto e a aplicação das determinadas resoluções para as mesmas.

Delfino (2013) propõe um processo para a gerência de comunicação no ambiente de desenvolvimento *Scrum*. Ele faz um estudo bibliográfico para a identificação de barreiras comunicacionais, barreiras estas que são problemas que impactam na comunicação interpessoal. Além deste estudo na literatura, foi feito também um estudo de campo praticado por meio da aplicação de um questionário em organizações de desenvolvimento de software, questionário este que possui como objetivo, identificar a maneira que as organizações praticavam a gerência de comunicação no ambiente de desenvolvimento *Scrum*. A aplicação deste questionário possuía os dois seguintes critérios: ser aplicado com participantes que estivessem ativos no mercado de trabalho; ser aplicado em organizações de desenvolvimento de software que utilizam a metodologia *Scrum*. Algumas das barreiras comunicacionais identificadas por Delfino, são: as barreiras pessoais, que apresentam problemas recorrentes aos aspectos humanos na comunicação (e.g. raiva e vergonha), aspectos estes que podem causar problemas na emissão ou no recebimento de mensagens; as barreiras físicas, que são problemas ocasionados pelos aspectos físicos e estruturais do ambiente onde ocorrem esta comunicação; e as barreiras semânticas, que estão interligadas a maneira que a mensagem é transmitida, (e.g. utilização de jargões).

O trabalho proposto por Bettenburg (2014) faz um estudo sobre o impacto da comunicação entre os *stakeholders* envolvidos no processo de desenvolvimento do software. O seu objetivo é fornecer indicações a respeito da comunicação para auxiliar as partes interessadas na tomada de decisões com o propósito de aumentar a qualidade do software bem como a eficiência do seu desenvolvimento. Por meio de um estudo empírico, ressalta especialmente os efeitos da comunicação dos desenvolvedores na qualidade e na evolução do software. Entre os resultados apresentados, foi expresso a existência de uma forte interligação entre a comunicação dos desenvolvedores com a qualidade do software, apresentando que a qualidade do produto é extremamente afetada com a ocorrência de problemas comunicacionais, descobrindo assim que a comunicação entre os desenvolvedores possui um papel de grande importância para a manutenção e evolução do software.

O artigo de Damian *et al.* (2007) investiga o impacto sofrido em tarefas executadas em equipes de desenvolvimento de software globais. Para tal fim, ocorreu um

estudo de caso de uma equipe de desenvolvimento de software distribuído no centro avançado de pesquisa e desenvolvimento da IBM Ottawa Software Lab, o estudo ocorreu por meio de observações e entrevistas com a equipe de desenvolvimento e com gerentes, de modo a estudar como ocorrem as práticas colaborativas da equipe distribuída. Como resultado das observações obtidas foi possível identificar três achados, são eles: (i) a cultura organizacional possui grande importância na coordenação, desenvolvimento e na manutenção dos trabalhos realizados; (ii) a comunicação e a interação social ocorrem de forma dinâmica e deve-se ter atenção para que isso não prejudique a execução das atividades; e (iii) o excesso de informações e falhas comunicacionais colaboraram para a interrupção da construção do produto.

O artigo de Puarungroj *et al.* (2018) analisa e busca problemas relacionados ao processo de elicitação de requisitos, os problemas apontados possuem relação com aspectos comunicacionais. Nesta pesquisa, ocorreu a investigação do processo de elicitação de requisitos presente no desenvolvimento de um software universitário que possui o propósito de gerenciar aspectos estudantis, dentre muitas das suas funcionalidades, algumas delas são as gerências de: atividades do aluno; bolsa de estudos e disciplinas. Como resultados da observação do processo de elicitação foi possível identificar alguns problemas que ocasionaram na ocultação de informações, prejudicando assim a comunicação, os problemas identificados foram: (i) os usuários resistiram em informar as suas necessidades; (ii) os usuários não observavam a contribuição que o sistema teria no seu trabalho; e (iii) os usuários solicitavam mudanças logo após o desenvolvimento do sistema. Como possível solução para estes problemas, foi indicado a criação de protótipos como forma de validar os requisitos capitados.

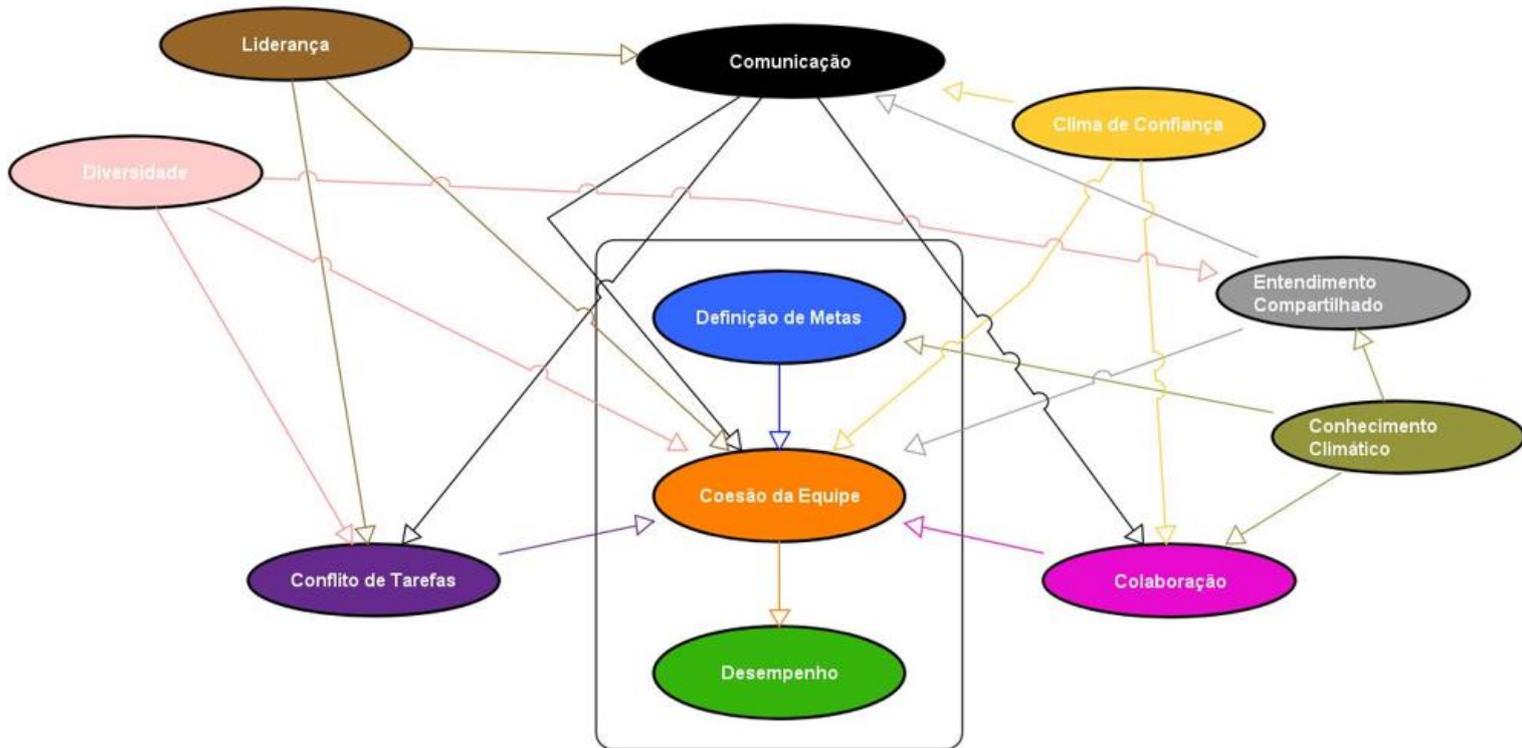
O trabalho proposto por Korte, Systs e Hjelsvold (2015) faz uma comparação entre o trabalho de desenvolvimento de software distribuído e local. O seu objetivo é identificar se as adversidades que ocorrem com equipes de desenvolvimento de software separadas geograficamente são as mesmas sofridas com equipes locais. Para alcançar o objetivo de pesquisa foi feito um curso em conjunto com duas universidades situadas na Finlândia e na Noruega, à vista disso foi possível observar os problemas presentes em equipes locais e equipes que possuem membros localizados nas duas universidades. Como resultado foi possível concluir que equipes locais possuem os mesmos desafios enfrentados por equipes separadas geograficamente, desafios estes que são relacionados à construção da equipe, definição das funções do projeto, problemas de comunicação e de gerenciamento. Também foi concluído que, apesar de os desafios e problemas serem iguais, os problemas ocorridos em

equipes separadas são mais complexos de serem resolvidos e que as adversidades ocorridas possuem maior impacto.

O artigo de Damian (2001) pesquisa sobre engenharia de requisitos igualmente ao trabalho proposto por Puarungroj *et al.* (2018). Porém, Damian (2001) difere em fazer um estudo analisando a presença de requisitos em ambientes de desenvolvimento distribuído. No seu trabalho são apresentadas as lições aprendidas obtidas por meio de um estudo exploratório que ocorre através da análise da execução de reuniões de requisitos de cinco grupos, em que quatro são separados geograficamente e um deles a reunião ocorre presencialmente. Cada grupo é composto pelo usuário do sistema, analista de sistema e um facilitador de reunião. Como lições aprendidas foi possível identificar que: (i) a reunião presencial não foi melhor que as reuniões dos grupos distribuídos; (ii) as reuniões remotas beneficiaram as eliciações de requisitos; e (iii) a reunião presencial foi útil, mas não benéfica, isso se dá pelo o fato de a interação presencial ocasiona uma menor disposição da expressão de opiniões e sugestões.

O trabalho proposto por Rutz e Tanner (2016) se propõe a identificar os fatores que possuem influência no desempenho de equipes terceirizadas e que estão separadas geograficamente. Para identificar estes fatores foram realizados dois estudos, são eles: a coleta feita por meio de estudo literário e um estudo de caso com duas equipes separadas geograficamente e envolvidas em projetos terceirizados pertencentes a uma empresa de desenvolvimento de software. Como resultado foi possível propor uma expansão do modelo proposto por Brahm e Kunze (2012), que também analisa os fatores que influenciam o desempenho de equipes separadas geograficamente. A expansão proposta por Rutz e Tanner (2016) é apresentada na seguinte Figura 2.

Figura 2 – Fatores de influência em equipes terceirizadas e separadas geograficamente.



Fonte: adaptado de Rutz e Tanner (2016)

Por meio da Figura 2 foi possível concluir que:

- Coesão e desempenho da equipe: afeta diretamente o desempenho da equipe, sendo assim, quanto mais forte é a coesão, melhor o desempenho.
- Definição de metas: afeta diretamente a coesão da equipe.
- Conflito de tarefas: afeta a coesão da equipe, criando atrasos e afetando o tempo de desenvolvimento do software.
- Colaboração: afeta a coesão da equipe, concluindo que o trabalho em equipe significa mais pessoas na busca do objetivo em comum.
- Entendimento compartilhado: afeta a coesão da equipe, isso se deve ao fato de os membros da equipe possuindo o mesmo entendimento dos processos, protocolos e normas são capazes de efetuar um melhor trabalho juntos. Além disso, afeta também a comunicação, em que ao possuir estes entendimentos, ocorre uma melhor comunicação.
- Diversidade: afeta a coesão da equipe, isso se deve ao fato de quanto mais diversidade nas contribuições de ideias e sugestões para lidar com um problema, melhor ele será resolvido. Afeta também, positivamente ou

negativamente o conflito de tarefas, em que devido a diversidade de cultura pode ocasionar algumas possíveis divergências de visões. Além disso, afeta o entendimento compartilhado.

- Conhecimento climático: afeta a colaboração, dado que o conhecimento climático proporciona um senso geral de todos os membros da equipe sobre o contexto uns dos outros. Afeta também no estabelecimento de metas, dado que proporciona um maior entendimento do quanto trabalho uma equipe pode realizar e em quanto tempo. Além disso, afeta o entendimento compartilhado, dado que as tarefas podem ser atribuídas aos membros certos da equipe.
- Clima de confiança: afeta a colaboração e a comunicação, em que devido ao clima de confiança, a colaboração ocorre de forma que uns aos outros se ajudam na execução das tarefas. Afeta também a coesão da equipe, já que os membros da equipe são capazes de confiar nos outros membros da equipe.
- Liderança: afeta na redução dos conflitos de tarefas. Afeta também na comunicação, dado que a liderança cria um relacionamento entre os membros da equipe. Além disso, afeta a coesão da equipe.
- Comunicação: afeta o conflito de tarefas, de modo que com uma boa comunicação os conflitos de tarefas são reduzidos. Além disso, a comunicação proporciona a colaboração entre os membros da equipe, melhorando assim a coesão da equipe.

Os trabalhos (WU, 2012), (JUSOH *et al.*, 2018), (HASSAN *et al.*, 2019), e (HOLMSTROM *et al.*, 2006) estudam a abordagem da comunicação presentes em ambientes distribuídos de forma a identificar possíveis problemas e indicações de soluções nesses ambientes, apesar de as indicações serem focadas para ambientes distribuídos, pode-se aplicá-las também caso o problema ocorra em equipes que trabalham no mesmo ambiente, dado que Korte, Systa e Hjelsvold (2015) informam que os problemas presentes em ambientes distribuídos são iguais aos sofridos localmente. Os problemas e indicações apresentadas pelos trabalhos são: (i) dada a distância que afeta a comunicação, deve-se adequar a comunicação para a realidade vivenciada; (ii) a presença de diferentes culturas dificulta a comunicação, no entanto deve-se garantir o conhecimento de todos sobre as políticas e práticas da organização; (iii) o fator de confiança é crucial para uma boa comunicação, de forma que a falta de confiança causa omissão de informações, logo deve-se garantir que seja mantida a confiança

entre os membros da equipe; e (iv) a comunicação deve ser controlada de forma a garantir a sua integridade, evitando a sua perda.

O artigo de Santos *et al.* (2014) propõe uma solução comunicacional para equipes de software distribuídas que utilizam metodologias ágeis. A problemática de pesquisa se trata de que a comunicação existente em equipes distribuídas e que são ágeis, é afetada pela pouca documentação e ao dinamismo das informações presentes em metodologias ágeis. Como forma de propor a solução, foi feita uma pesquisa-ação que aborda: (i) detecção do problema; (ii) planejamento da solução e implantação; (iii) avaliação dos resultados; e (iv) monitoramento. Esta pesquisa-ação foi feita em uma empresa de desenvolvimento de aplicações Web e que a equipe se encontra distribuída. Como solução, dado que a empresa utiliza metodologia *Scrum*, foi proposto a criação de uma nova função chamada “*Communication Owner*” que deve fazer a intermediação da comunicação entre os desenvolvedores da equipe e os clientes e parceiros da empresa. A sua responsabilidade é herdar as características do *Product Owner*, além de possuir algumas outras atividades. Segundo Pham e Pham (2012) o *Product Owner* deve: (i) gerenciar as prioridades e expectativas dos *stakeholders*; (ii) conhecer o produto; (iii) coletar requisitos; (iv) apoiar a equipe; (v) manusear múltiplas atividades; e (vi) comunicar informações do produto para a equipe. Para Santos *et al.* (2014) além destas atividades o *Communication Owner* deve: (i) comunicar com todos os *stakeholders*, incluindo a total atenção que deve ser destinada ao cliente; (ii) documentar as informações e compartilhar com todos os envolvidos; (iii) gerenciar o backlog e auxiliar em possíveis dúvidas de escopo. Com a criação do *Communication Owner*, a empresa foi capaz de melhorar o foco e a produtividade entre os membros da equipe e os escopos dos projetos foram mais bem definidos. Esta solução foi documentada no presente trabalho de conclusão de curso no formato de padrão de software.

O trabalho proposto por Mak e Kruchten (2006) apresenta uma solução para a priorização de atividades. Dado que as atividades executadas durante o projeto possuem dependências de artefatos, a não priorização das atividades de acordo com essas dependências pode acarretar em atrasos na entrega do produto final. Além disso, a falta de comunicação entre os membros da equipe junto ao gerente de projeto pode contribuir para a construção de uma errônea priorização. Sendo assim, Mak e Kruchten (2006) propôs uma maneira de estas atividades serem priorizadas de acordo com os artefatos produzidos e necessitados, e que leva em consideração a comunicação entre os membros da equipe. Apesar de o autor considerar que esta solução deve ser aplicada em equipes ágeis, ela também pode ser utilizada por

equipes que utilizam outras metodologias. Esta solução foi documentada no presente trabalho de conclusão de curso no formato de padrão de software.

O trabalho proposto por Stray, Moe e Sjoberg (2020) pesquisa sobre a prática do *daily stand-up meeting* presente em projetos *Scrum*. *Stand-up meeting*, é a prática de reuniões em pé, que ocorrem diariamente, entre os membros da equipe juntamente com o *scrum master* para comunicar informações pertinentes ao projeto, nesta reunião basicamente os membros se comunicam a partir da reportagem das seguintes informações: (i) o que foi feito ontem; (ii) o que será feito hoje; e (iii) se algum impedimento foi identificado. A partir disso, o objetivo de Stray, Moe e Sjoberg (2020) é identificar aspectos negativos que afetam a realização das reuniões, além disso, no seu trabalho é proposto soluções para estes problemas. Como meio de levantar dados, a pesquisa analisou reuniões de *stand-up meeting* de quatro empresas, todas elas possuindo áreas diferentes de desenvolvimento, as empresas analisadas foram:

- ITConsult: empresa norueguesa que pratica consultorias de TI;
- TelSoft: empresa de desenvolvimento de software de telecomunicações;
- GlobEng: fornece serviço para a indústria de engenharia;
- Norbank: é um banco nórdico.

Nas quatro empresas analisadas, foi possível observar e documentar a realização de 102 reuniões e 60 membros com funções e empresas diferentes foram entrevistados. Como problemas identificados foi possível visualizar que: (i) maior parte do tempo das reuniões é destinada para a reportagem do que foi feito ontem, informação esta que possui impacto positivo maior apenas para o *scrum master*, enquanto os membros envolvidos na reunião não visualizam a importância dessas informações; (ii) algumas pessoas possuem mais tempo de fala que outras, por exemplo, quem fala no início da reunião possui mais tempo de fala em relação a quem fala perto do fim da reunião, ocorrendo assim a perda de informações; (iii) o horário de realização da reunião pode impactar negativamente, já que, por exemplo, a escolha de um horário inadequado pode acarretar um maior tempo de duração da reunião; e (iv) ocorre um impacto negativo motivacional em alguns membros devido a frequência das reuniões, isso se deve ao fato de que algumas equipes não necessitam ter reuniões diárias. Como soluções foi possível concluir que:

- Nas reuniões deve-se parar de reportar o que foi feito ontem, dado que os participantes da reunião dão importância apenas a assuntos que podem contribuir para a resolução dos seus problemas. Apesar da reportagem ser

importante, ela deve ser feita de outra maneira, por exemplo, utilizando alguma ferramenta que permita isso.

- Deve-se dividir igualmente o tempo de fala de cada participante para que todos sejam capazes de se comunicar.
- Normalmente as reuniões ocorrem no início do horário de serviço, porém devido a atrasos, a reunião tende a durar mais que o tempo planejado. Um bom horário de realização da reunião é antes do almoço, já que os participantes são estimulados a terminar a reunião no horário correto devido ao horário de almoço. Além disso, eles são estimulados a almoçar juntos e continuar a comunicação das informações presentes na reunião.
- Deve-se realizar uma análise de quem a reunião vai impactar, para que assim só participe quem irá ser afetado pelas informações presentes na reunião. Sendo assim, deve-se analisar a frequência da participação das equipes.

Essa solução foi documentada no presente trabalho de conclusão de curso no formato de padrão de software.

3.2 Padrões de Software

O trabalho proposto por Dantas *et al.* (2009), é um padrão de software cujo objetivo é auxiliar os gerentes de projetos na resolução de problemas que envolvem conflitos de interesses entre os *stakeholders*. Esse padrão apesar de sua validação ter sido feita no ambiente de desenvolvimento de software, pode ser aplicado em projetos em geral. A solução proposta por esse trabalho no âmbito da comunicação é dada por meio da indicação ao gerente de projetos em incentivar o contato entre os *stakeholders* que possuem conflitos de interesses, o padrão indica como incentivo ao contato, a prática dos seguintes meios de comunicação: reuniões, videoconferências, e-mails, telefonemas, dentre outros.

O trabalho proposto por Rising (1999) possui como abordagem a demonstração da importância do uso de padrões de software, mostrando tanto sua relevância como forma de compartilhamento de soluções previamente utilizadas como para quem utiliza-o de forma a resolver a ocorrência de algum problema. No seu trabalho é mostrado quais campos um padrão deve possuir, são apontados os seguintes campos: nome; problema; contexto; forças; solução e contexto resultante. As descrições destes campos são apresentadas na seção de fundamentação teórica. Rising (1999) como forma de exemplificar estes campos, no seu trabalho é apresentado o padrão de projeto Facade, ou também chamado de Fachada, além

deste padrão, é mostrado também o padrão Broker, padrões estes que são apresentados pelo autor da seguinte forma:

Primeiro padrão: Facade

Nome - Facade

Problema – De que forma deve ocorrer a proteção dos clientes de um subsistema em relação a complexidade das interfaces das classes do subsistema?

Contexto – Um sistema que possui subsistemas e que ocorre dependências entre os subsistemas.

Forças –

- Ao particionar um sistema em subsistemas a sua complexidade diminui.
- Subsistemas se tornam mais complexos de acordo com a evolução do sistema.
- Quanto mais dependências de um subsistema, mais difícil será de entendê-lo e reutilizá-lo.

Solução – Adicione um objeto de fachada para o subsistema, que seja mais simples de entendê-lo do que o conjunto de interfaces para todas as classes do subsistema. Faça com que os clientes que precisam acessar os objetos de um subsistema façam isso por meio do acesso ao objeto fachada.

Contexto Resultante –

- A criação do objeto fachada faz com que a funcionalidade do subsistema não seja acessada diretamente, tornando a interface mais simples.
- Proporciona a redução do acoplamento entre os componentes do subsistema.
- Torna mais fácil as mudanças no subsistema.

Segundo padrão: Broker

Nome - Broker

Problema – Quando um sistema é feito de componentes distribuídos, eles precisam de uma maneira para se comunicar.

Contexto – Você está trabalhando em um sistema distribuído, possivelmente heterogêneo, com elementos independentes e cooperantes.

Forças –

- Particionar o sistema em componentes independentes permite a distribuição.

- Serão necessários serviços para adicionar, remover, trocar, ativar e localizar componentes.
- Os aplicativos que usam um objeto não devem se preocupar com sua localização física.

Solução – Adicione um componente *broker* (corretor) para desacoplar clientes e *sewers* (esgotos). Os servidores se registram no *broker* e os clientes acessam o *sewers* enviando solicitações ao *broker*.

Contexto Resultante –

- Os clientes não precisam saber onde os servidores estão localizados. Os servidores podem ser movidos sem afetar o acesso do cliente.
- O desempenho do sistema pode sofrer como resultado da sobrecarga de usar o *broker* para acessar os servidores.

O trabalho proposto por Salger, Englert e Engels (2010) pesquisa sobre a especificação de requisitos de software com abordagem em padrões de software para resolução de problemas de especificação existentes no contexto de equipes distribuídas. Segundo Salger, Englert e Engels (2010) é muito comum que organizações adaptem, para a sua realidade, soluções já existentes, porém as mesmas não são compartilhadas para que outras instituições possam utilizar. Sendo assim, o seu objetivo é apresentar por meio de documentação de padrões de software as soluções utilizadas por uma empresa chamada Capgemini. Os padrões documentados no seu trabalho foram aplicados em equipes distribuídas que participavam de um projeto com escopo de gerência de vendas automotivas, este projeto contava com a participação de 30 membros, metade dos membros estavam separados geograficamente, enquanto a outra metade trabalhava no mesmo ambiente. Como resultado, ocorreu a documentação de três padrões, foi documentado de cada padrão os campos de nome, contexto, problema, forças, solução e contexto resultante (chamado pelo autor como benefícios). São apresentados, a seguir, os problemas, soluções e benefícios de cada padrão documentado:

Primeiro padrão:

Nome: *Define “Use Case”*

Problema: Existem várias formas de documentar ‘caso de uso’ na literatura. Porém, não existe uma única ‘definição certa’ para a engenharia de software.

Solução:

- a) Os analistas distribuídos devem definir um modelo preciso de como construir o caso de uso. Deve-se dar exemplos de como o modelo deve ser utilizado e como não deve ser utilizado, apresentando casos bons e ruins.
- b) Comunicar aos membros da equipe distribuídos que a solução deve ser adequada ao projeto ao tipo de arquitetura que está sendo utilizado. Isso deve ser comunicado ao gerente de projeto, e deve ser informado que alguns modelos de casos de uso podem ser melhores que outro.

Benefícios:

- a) Todos os membros da equipe possuem o mesmo conhecimento sobre caso de uso.
- b) Ao ocorrer a clara definição, faz com que as modelagens não possuam formatos diferentes.

Segundo padrão:

Nome: *Onboard Business Analyst during Requirements Engineering*

Problema: Em um ambiente de equipes distribuídas, as documentações das especificações de requisitos normalmente não são suficientes para transmitir todas as informações para que ela seja implementada e possua todas as funcionalidades necessárias.

Solução:

- a) A reunião de análise de requisito deve ter a participação do cliente, do analista de negócios que está separado geograficamente e o analista de sistema.
- b) Compartilhe todo o conhecimento disponível.
- c) Mantenha a confiança entre cliente e os analistas, estejam eles separados geograficamente ou não.
- d) Envolve os membros separados geograficamente na atividade de especificação de requisitos.
- e) Gerencie o compartilhamento das informações.
- f) Caso a presença do analista de negócios nem sempre seja possível, solicite a sua presença apenas em atividades mais complexas.

Benefícios: O analista de negócio separado geograficamente pode auxiliar os desenvolvedores em possíveis dúvidas a respeito dos requisitos.

Terceiro padrão:

Nome: *Map Business Terms to Entity Attributes*

Problema:

- a) Para o desenvolvedor separado geograficamente pode ser complicado entender quais entidades devem ter acesso a um caso de uso, já que ele não participa de reuniões de requisitos.
- b) Equipes trabalhando no mesmo ambiente, o analista de requisitos costuma se juntar com a equipe de desenvolvimento, tornando mais fácil o compartilhamento de informações, o que não ocorre em equipes separadas geograficamente.

Solução:

- a) Utilize termos simples para que todos os envolvidos consigam compreender. Além disso, deve-se padronizar o formato documentado para facilitar o reconhecimento de todos.
- b) Adicione identificadores nos documentos, fazendo a relação dos termos específico de uma área para um termo de conhecimento de todos.
- c) Padronize e use o mesmo termo em toda a documentação.

Benefícios: O desenvolvedor será capaz de entender as modelagens de caso de uso, sabendo quais entidades acessam um determinado caso de uso, e entenderá também toda a documentação.

O artigo de Monasor *et al.* (2013) afirma que a abordagem do assunto de problemas que ocorrem em ambientes distribuídos é tratada de forma muito simples pelos trabalhos presentes na literatura, da mesma forma ocorrem com as contribuições de soluções. Entretanto, as organizações necessitam de soluções mais bem estruturadas, soluções a nível de padrões. O objetivo do trabalho é fornecer uma plataforma¹ que permite o compartilhamento de soluções, pertinentes a ambientes distribuídos, bem estruturadas como forma de difundir o conhecimento para a resolução dos problemas. Além da plataforma, o autor estimula que os contribuintes documentem as soluções no formato de padrão de software, sugerindo apenas a divisão do campo contexto em problema e população, o campo problema deve conter a descrição detalhada do problema e quando ele ocorre, já o campo população deve conter a população afetada. Essa divisão ocorre dado que o contexto é de extrema importância para equipes distribuídas. Segundo Monasor *et al.* (2013) um padrão para equipes distribuídas deve

¹ Fornecido em: <<http://global.lero.ie/community/>>

conter os campos de: nome do padrão; problema; população análise do problema; solução e referência de onde a solução foi retirada. Apesar da contribuição da plataforma como meio de compartilhamento de informações, no momento atual da publicação deste trabalho de conclusão de curso a plataforma se encontra indisponível.

3.3 Comparativo

Abaixo é apresentada a Tabela 1 contendo a comparação desta pesquisa de conclusão de curso com os trabalhos relacionados. Nesta tabela, os trabalhos relacionados são comparados de acordo com três aspectos: (i) se o trabalho aborda problemas comunicacionais; (ii) se apresenta soluções de problemas comunicacionais; e (iii) se apresenta documentação de padrão de software.

Tabela 1 – Comparativo entre os trabalhos relacionados

Trabalho	Problemas Comunicacionais	Soluções dos Problemas	Documentação de Padrão
Este trabalho	x	x	x
PMBOK (2017)	x	x	
Lu, Liu e Liu (2009)	x		
Delfino (2013)	x	x	
Bettenburg (2014)	x	x	
Damian <i>et al.</i> (2007)	x	x	
Puarungroj <i>et al.</i> (2018)	x	x	
Korte, Syste e Hjelsvold (2015)	x		
Damian (2001)	x	x	
Rutz e Tanner (2016)	x		
Wu (2012)	x	x	
Jusoh <i>et al.</i> (2018)	x	x	
Hassan <i>et al.</i> (2019)	x	x	
Holmstrom <i>et al.</i> (2006)	x	x	
Santos <i>et al.</i> (2014)	x	x	
Mak e Kruchten (2006)	x	x	

Stray, Moe e Sjoberg (2020)	x	x	
Dantas <i>et al.</i> (2009)	x	x	x
Rising (1999)			x
Salger, Englert e Engels (2010)	x	x	x
Monasor <i>et al.</i> (2013)	x	x	x

Fonte: elaborado pelo autor.

A partir da tabela de comparação entre os trabalhos relacionados, é importante em particular, esclarecer a diferença deste trabalho de conclusão de curso em comparação a 4 trabalhos. Que são os de (DANTAS *et al.*, 2009), (RISING, 1999), (SALGER; ENGLERT; ENGELS, 2010) e (MONASOR *et al.*, 2013).

O trabalho proposto por Dantas *et al.* (2009) é um padrão de software e apresenta como solução a resolução de conflitos de interesse, problema este que afeta a comunicação entre os *stakeholders*, sendo assim, aborda aspectos de problema de comunicação, solução e documentação de padrão. Entretanto, difere-se deste trabalho de conclusão de curso dado que Dantas *et al.* (2013) é um padrão já publicado e oferece solução para um contexto (contexto de resolução de conflitos de interesse), enquanto este trabalho de conclusão de curso documenta padrões de soluções diferentes.

O trabalho proposto por Rising (1999) possui relação com este trabalho de conclusão de curso apenas em relação a documentar padrões, apesar de o padrão presente no seu trabalho não ter correlação ao contexto de comunicação ele auxiliou a este trabalho de conclusão de curso em como documentar padrões de software.

Os trabalhos propostos por (SALGER; ENGLERT; ENGELS, 2010) e (MONASOR *et al.*, 2013) propõem soluções que resolvem problemas de comunicação e é documentada no formato de padrão. Entretanto, difere-se deste trabalho de conclusão de curso dado que apesar das soluções apresentadas serem para o contexto de comunicação as soluções são diferentes.

4 OBJETIVOS

Com base na problemática de pesquisa apresentada nas seções anteriores, as próximas subseções apresentam os objetivos gerais e específicos desta pesquisa.

4.1 Objetivo Geral

Apoiar a área da Gerência de Comunicação, identificando um conjunto de problemas recorrentes que impactam na comunicação entre os *stakeholders* no ambiente de desenvolvimento de software e suas respectivas soluções através de uma pesquisa na literatura.

4.2 Objetivos Específicos

- Identificar problemas e soluções recorrentes à comunicação interpessoal no ambiente de desenvolvimento de software.
- Categorizar estes problemas identificados com base em seus aspectos em comum.
- Documentar soluções no formato de padrões de software.

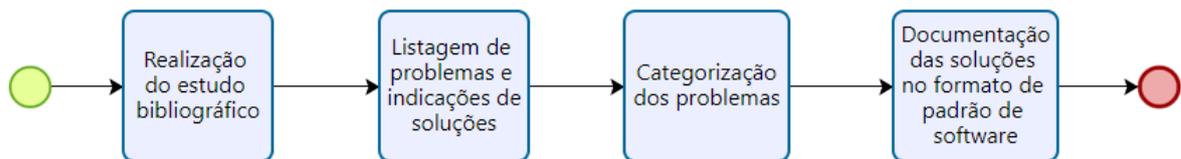
5 PROCEDIMENTOS METODOLÓGICOS

Esta seção descreve as etapas que foram executadas na pesquisa com o propósito de atingir os objetivos planejados.

Preliminarmente, foi definido a área de estudo que a pesquisa deve abranger, em sequência foi identificado e determinado o problema e qual solução este trabalho deverá apresentar, definindo assim as questões e os objetivos de pesquisa.

A metodologia utilizada neste trabalho é demonstrada na Figura 3 e foi dividida nas seguintes atividades: (i) Realização do estudo bibliográfico; (ii) Listagem de problemas e indicações de soluções; (iii) Categorização dos problemas; e (iv) Documentação das soluções no formato de padrão de software.

Figura 3 – Atividades do procedimento metodológico



Fonte: elaborado pelo autor.

Nas próximas subseções são apresentadas detalhadamente como ocorreu a execução de cada atividade expressa na figura anterior (Figura 3).

5.1 Estudo Bibliográfico

Foi nesta etapa que ocorreu o estudo de trabalhos com abordagens no contexto da comunicação interpessoal no ambiente de software. Com base neste estudo, foi possível identificar e documentar neste trabalho, os problemas e soluções recorrentes na comunicação interpessoal.

Para proporcionar um maior rigor na identificação dos trabalhos, como mecanismo de planejamento deste estudo bibliográfico, alguns aspectos foram definidos de acordo com o trabalho proposto por Kitchenham (2004). Sucintamente, o seu trabalho orienta que algumas definições sejam planejadas para o estudo na literatura, que são: (i) definição de como os trabalhos relevantes acerca da temática pesquisada serão identificados; (ii) como ocorrerá a seleção destes trabalhos; e (iii) como os resultados serão identificados e documentados.

Estas definições são apresentadas nas seguintes seções de: estratégia de pesquisa; análise e critério de seleção.

5.1.1 Estratégia de Pesquisa

Como estratégia de pesquisa, foi definida uma *string* de busca. Para a sua definição foi analisado quais palavras-chave refletem a problemática identificada e estas palavras foram unidas por operadores lógicos de AND e OR. A *string* utilizada para a obtenção de trabalhos relacionados foi a seguinte:

Primeira *String*: ("*lesson learned*" OR *problem* OR *conflict*) AND ("*software development*" OR "*software project*") AND *communication*.

Com base nesta *string*, foi realizado uma pesquisa nas bases de dados de sites conceituados de publicações científicas, as bibliotecas digitais utilizadas foram a IEEE (*IEEE Xplore*) e a ACM (*ACM Digital Library*). Porém a busca na ACM retornou mais de 20 mil resultados, o que é inviável para ser analisado. Sendo assim a *string* anterior foi refinada para ser utilizada na ACM, a *string* utilizada foi a seguinte:

Segunda *String*: ("*lesson learned*" OR *problem* OR *conflict*) AND ("*software development*" OR "*software project*") AND ("*communication of members*" OR "*process communication*")

A Tabela 2 apresenta a quantidade de resultados obtidos em cada biblioteca com a execução da *string*.

Tabela 2 – Resultados da aplicação da *string* na primeira etapa de seleção.

BASE DE DADOS	STRING UTILIZADA	RETORNADOS	ANALISADOS	SELECIONADOS
IEEE	Primeira <i>String</i>	953	953	69
ACM	Segunda <i>String</i>	512	270	23
Total:		1465	1223	92

Fonte: elaborada pelo autor.

Na seção 5.1.2 é apresentado o critério de análise e seleção destes trabalhos.

Na base de dados da IEEE não ocorreu nenhuma limitação dos parâmetros permitidos pela plataforma, dois destes parâmetros são:

- Anos: a IEEE permite o retorno de trabalhos no alcance dos anos de 1975 a 2021, e estes foram o intervalo de anos dos trabalhos retornados. A não

delimitação dos anos se deve ao fato da valorização de contribuições científicas antigas a respeito da abordagem pesquisada.

- Campos de pesquisa: todos os campos de pesquisa foram levados em consideração, campos estes que são: conferências, revistas, livros, artigos de acesso antecipado, cursos e padrões.

Na base de dados da ACM também foi utilizada a mesma estratégia empregada na IEEE, em que também não ocorreu nenhuma limitação dos parâmetros permitidos pela plataforma. Porém, devido ao fim da *deadline* planejada para o estudo bibliográfico, diferente da técnica aplicada na IEEE de analisar todos os trabalhos retornados, na ACM foi analisado apenas uma parte dos trabalhos. Apesar da não análise de todos os trabalhos retornados pela ACM, esta ação não impactou negativamente na pesquisa, já que foi possível identificar uma boa base de dados para alcançar os objetivos determinados neste trabalho.

5.1.2 Análise e Critérios de Seleção

Como estratégia de análise destes trabalhos, foram utilizadas duas etapas de seleção. Na primeira etapa, ocorreu a leitura dos títulos, palavras-chave e resumos, que também é nomeado na língua inglesa como *abstract*. Após a leitura do seu resumo, caso a sua proposta de pesquisa não tivesse correlação com os objetivos de pesquisa deste trabalho, ocorria apenas uma análise superficial do mesmo e posteriormente sofria um “descarte”. Caso contrário, se a pesquisa tivesse correlação com os objetivos deste trabalho de conclusão de curso, ocorria o agrupamento de trabalhos selecionados inicialmente. Após a execução da primeira etapa foram selecionados, como demonstrado na Tabela 2, uma quantidade de 69 trabalhos selecionados na IEEE e 23 trabalhos na ACM.

Já como segunda etapa, passando por uma nova seleção, todos os trabalhos selecionados na primeira etapa foram lidos por completo e foi possível selecionar 29 estudos. Os 29 trabalhos selecionados estão presentes na Tabela 3 estruturada com um identificador do trabalho, título, ano de publicação e referência.

Tabela 3 – Trabalhos selecionados

ID	Título	Referência	Ano
T1	<i>Global Software Development Challenges: A Case Study on Temporal, Geographical and Socio-Cultural Distance</i>	(HOLMSTROM <i>et al.</i> , 2006)	2006

T2	<i>Understanding and improving software build teams</i>	(PHILLIPS; ZIMMERMANN; BIRD, 2014)	2014
T3	<i>A Set of Artifacts and Models to Support Requirements Communication Based on Perspectives</i>	(ORAN, 2017)	2017
T4	<i>Global Requirements Engineering: Decision Support for Globally Distributed Projects</i>	(LESCHER; BRUGGE, 2009)	2009
T5	<i>Daily Stand-Up Meetings: Start Breaking the Rules</i>	(STRAY; MOE; SJOBERG, 2020)	2020
T6	<i>Distributed team performance in software development</i>	(HAUSE, 2005)	2005
T7	<i>Towards a Global Software Development Community Web: Identifying Patterns and Scenarios</i>	(MONASOR <i>et al.</i> , 2013)	2013
T8	<i>Improved communication in distributed agile software development</i>	(SANTOS <i>et al.</i> , 2014)	2014
T9	<i>Task Coordination in an Agile Distributed Software Development Environment</i>	(MAK; KRUCHTEN, 2006)	2006
T10	<i>An empirical study of requirements engineering in distributed software projects: is distance negotiation more effective?</i>	(DAMIAN, 2001)	2001
T11	<i>Towards Specification Patterns for Global Software Development Projects - Experiences from the Industry</i>	(SALGER; ENGLERT; ENGELS, 2010)	2010
T12	<i>Patterns: a way to reuse expertise</i>	(RISING, 1999)	1999
T13	<i>Coordination Practices in Distributed Software Development of Small Enterprises</i>	(BODEN; NETT; WULF, 2007)	2007
T14	<i>An architecture for supporting small collocated teams in cooperative software development</i>	(CAMPAGNOLO <i>et al.</i> , 2009)	2009
T15	<i>Global vs. local — Experiences from a distributed software project course using agile methodologies</i>	(KORTE; SYSTA; HJELSVOLD, 2015)	2015

T16	<i>An insight into the interplay between culture, conflict and distance in globally distributed requirements negotiations</i>	(DAMIAN; ZOWGHI, 2003)	2003
T17	<i>Overview of communication in global software development process</i>	(WU, 2012)	2012
T18	<i>Communication Management in Global Software Development Projects</i>	(JUSOH <i>et al.</i> , 2018)	2018
T19	<i>A Policy Recommendations Framework To Resolve Global Software Development Issues</i>	(HASSAN <i>et al.</i> , 2019)	2019
T20	<i>Studying the Impact of Developer Communication on the Quality and Evolution of a Software System: A Doctoral Dissertation Retrospective</i>	(BETTENBURG, 2014)	2014
T21	<i>Using Agile Practices to Solve Global Software Development Problems -- A Case Study</i>	(BEECHAM; NOLL; RICHARDSON, 2014)	2014
T22	<i>User participation in software development projects</i>	(SUBRAMANYAM <i>et al.</i> , 2010)	2010
T23	<i>Dealing with Change in Software Development: A Challenge for Requirements Engineering</i>	(PUARUNGROJ <i>et al.</i> , 2018)	2018
T24	<i>Factors that influence performance in Global Virtual Teams in outsourced software development projects</i>	(RUTZ; TANNER, 2016)	2016
T25	<i>Embracing agile development of usable software systems</i>	(LEE, 2006)	2006
T26	<i>A Systematic Approach for Identifying Requirement Change Management Challenges: Preliminary Results</i>	(ANWER; WEN; WANG, 2019)	2019
T27	<i>Awareness in the Wild: Why Communication Breakdowns Occur</i>	(DAMIAN <i>et al.</i> , 2007)	2007
T28	<i>Deciphering and Analyzing Software Requirements employing the techniques of Natural Language Processing</i>	(MEMON; XIAOLING, 2019)	2019

T29	<i>Relationship Research Between Communication Activities and Success Indexes in Small and Medium Software Projects</i>	(LU; LIU; LIU,)	2009
-----	---	------------------	------

Fonte: elaborada pelo autor.

5.2 Listagem de Problemas e Soluções

De acordo com o estudo bibliográfico executado na pesquisa, foi possível identificar nove problemas que afetam a comunicação bem como as suas soluções. Os problemas e suas soluções foram listados na seção 6.1 de uma forma visivelmente simples para que o leitor consiga aplicar as recomendações na ocorrência da problemática.

5.3 Categorização dos Problemas Identificados

De acordo com os problemas identificados e apresentados na listagem, foi analisado a correlação entre eles para aplicação de uma categorização. Alguns autores propõem formas para a categorização de problemas comunicacionais.

Fox (2001) propõe que os problemas recorrentes de comunicação podem ser categorizados em: problemas ambientais, problemas verbais e problemas interpessoais.

- Os problemas ambientais estão relacionados aos aspectos físicos e estruturais do ambiente onde ocorre esta comunicação, (e.g. ruídos sonoros e visuais que podem atrapalhar a transmissão e a compreensão da mensagem).
- Os problemas verbais estão relacionados a maneira que a mensagem é transmitida (e.g. utilização de jargões, falar de maneira rápida).
- Os problemas interpessoais estão relacionados aos aspectos pessoais da comunicação, são exemplos, as expressões de características emocionais (e.g. raiva e vergonha), aspectos estes que podem alterar a maneira de transmitir ou compreender a mensagem.

Uma outra proposta de categorização é feita por Chiavenato (2010), a sua proposta converge com a proposta de Fox (2001), ele propõe que os problemas de comunicação são decorrentes de três tipos, são eles: problemas pessoais, problemas físicos e problemas semânticos.

Visto a necessidade devido aos problemas identificados, será adicionada a estas maneiras de categorização, os problemas comunicacionais recorrentes a decisões de projeto,

que são problemas impactados por alguma decisão feita na gerência de projeto (e.g., quantidade de reuniões de um projeto menor que o necessário).

Sendo assim, visto que as propostas feitas por Fox (2001) e Chiavenato (2010) convergem e devido a existência de problemas comunicacionais ocasionados por decisões de projeto, os problemas identificados por meio da metodologia utilizada da subseção 5.2, foram categorizados de acordo com as seguintes quatro categorias: (i) problemas ambientais; (ii) problemas verbais; (iii) problemas pessoais; e (iv) problema de decisão de projeto.

5.4 Documentação das Soluções no formato de padrão de software

De acordo com a aplicação do estudo bibliográfico, foi possível identificar trabalhos que possuem soluções bem descritas e que é factível a sua documentação no formato de padrão de software. Apesar da apresentação destas soluções em seus respectivos trabalhos, ao documentar a solução no formato de padrão, documentamos formalmente as soluções, facilitando ainda mais o seu uso caso a adversidade venha a acontecer. Como forma de documentar soluções no formato de padrões de software, o modelo proposto por Coplien (1996) foi utilizado, de forma que os padrões documentados possuem os seguintes campos: (i) nome; (ii) contexto; (iii) problema; (iv) forças; (v) solução; (vi) contexto resultante; (vii) padrões relacionados; e (viii) usos conhecidos. Neste trabalho foi possível documentar 3 padrões de software, são eles:

- *Communication Owner*;
- *Team and Task Prioritization*; e
- *Unproductive Stand-Up Meetings*.

6 RESULTADOS

Nessa seção serão apresentados os resultados obtidos por meio do estudo bibliográfico para a identificação de problemas e soluções recorrentes à comunicação interpessoal.

A seção foi dividida em duas subseções:

- A subseção 6.1 contendo uma lista de problemas e soluções juntamente com a categorização feita.
- A subseção 6.2 contém a documentação de soluções no formato de padrão de software.

6.1 Problemas e Soluções Recorrentes a Comunicação

Com a execução do estudo bibliográfico foram identificados vários problemas comunicacionais bem como as suas soluções. A Tabela 4 apresenta os nove problemas e suas soluções identificadas, bem como a categorização feita. São apresentadas na seção 5.3 os mecanismos de categorizações adotados. A categorização foi feita de acordo com os seguintes grupos: (i) problemas ambientais; (ii) problemas verbais; (iii) problemas pessoais; e (iv) problema de decisão de projeto. Também são apresentados na Tabela 4 um identificador para cada problema e a referência bibliográfica da solução.

Tabela 4 – Problemas, indicações de soluções e categorização

Problemas Ambientais	
Identificador:	P1
Problema:	Reuniões presenciais ocasionam ambientes mais intimidador devido ao contato pessoal, podendo causar a perda de informações.
Solução:	Deve-se dosar as comunicações formais e informais, de forma que ocorra a presença da comunicação informal como forma de proporcionar um ambiente mais confortável para os envolvidos na reunião, entretanto esta comunicação deve ser dosada de forma a não prejudicar a comunicação formal.
Referência:	Damian (2001)

Identificador:	P2
Problema:	A distribuição geográfica dos membros da equipe pode ocasionar uma distância temporal entre eles devido a presença de fuso horários diferentes, ocasionando o atraso de informações.
Solução:	É necessário organizar os padrões de execução das atividades do trabalho de acordo com as diferenças de horário entre os membros separados geograficamente, como forma de facilitar a comunicação. Se possível, deve ocorrer a distribuição das atividades de acordo com a localidade dos membros, de modo que a diferença de horários entre as localidades seja o menor possível, possibilitando a diminuição da distância temporal.
Referência:	Holmstrom <i>et al.</i> (2006)
Identificador:	P3
Problema:	Equipes distribuídas e que usam metodologia Scrum possuem a comunicação afetada devido à baixa exigência de documentação e ao dinamismo das informações geradas nos projetos.
Solução:	Deve-se criar um papel <i>Communication Owner</i> , o responsável por este papel deve fazer a intermediação da comunicação entre o cliente e os desenvolvedores. Ele deve herdar as atribuições destinadas ao <i>Product Owner</i> .
Referência:	Santos <i>et al.</i> (2014), esta solução é apresentada, no formato de padrão de software, na subseção 6.2. O padrão se chama <i>Communication Owner</i> .
Problemas Verbais	
Identificador:	P4
Problema:	Diferenças linguísticas podem impactar a comunicação pelo errôneo entendimento da informação.
Solução:	Deve-se traduzir as políticas e práticas para uma linguagem que possibilite o entendimento de todos os membros da equipe. Na ocasião de incertezas a respeito da interpretação de uma informação, o emissor da informação deve ser consultado para que as incertezas sejam compreendidas.
Referência:	Wu (2012)

Problemas Pessoais	
Identificador:	P5
Problema:	A falta de conhecimento do cliente a respeito do que o software deve possuir ocasiona perda comunicacional devido a omissão de informações.
Solução:	A equipe do projeto além de elicitar requisitos por meio de entrevista deve concentrar-se na atividade de validação dos requisitos. Um modo de validar é construindo protótipos e apresentando-o ao cliente como modo de verificar se o protótipo atende as suas necessidades e expectativas.
Referência:	Puarungroj <i>et al.</i> (2018)
Identificador:	P6
Problema:	Os membros da equipe não dão a devida importância para a utilização de ferramentas que auxiliam a comunicação. Por exemplo, ao utilizar ferramentas para controle da comunicação, os membros apresentam resistência em utilizá-las.
Solução:	Deve-se conscientizar a equipe sobre a importância da utilização das ferramentas como forma de transmitir as informações individuais aos outros interessados no projeto. Sendo assim, dada a conscientização, perdas de informações poderão ser evitadas, contribuindo assim no desenvolvimento do projeto.
Referência:	Damian <i>et al.</i> (2007)
Problemas de Decisão de Projeto	
Identificador:	P7
Problema:	O cronograma apertado de um projeto, por motivo do curto tempo planejado, causa uma sobrecarga em vários aspectos, entre eles a comunicação.
Solução:	Deve-se identificar todas as atividades do projeto e documentá-las em um plano detalhado sobre o produto, contendo o tempo necessário para a sua execução, este tempo deve estar de acordo com a capacidade de produção dos membros da equipe.
Referência:	Hassan <i>et al.</i> (2019)

Identificador:	P8
Problema:	As atividades de um projeto devem ser priorizadas de acordo com as suas dependências de artefatos, no entanto, a falta de comunicação na atividade de priorização pode ocasionar uma errônea ordem de execução.
Solução:	Deve-se levar em consideração, na atividade de priorização de tarefas com base em seus artefatos, a comunicação interpessoal para acordar quais artefatos são necessários, determinando assim os seus graus de dependências.
Referência:	Mak e Kruchten (2006), esta solução é apresentada, no formato de padrão de software, na subseção 6.2. O padrão se chama <i>Team and Task Prioritization</i> .
Identificador:	P9
Problema:	A falta de gerência das reuniões de <i>stand-up meeting</i> , presente na metodologia <i>Scrum</i> , pode ocasionar perdas comunicacionais.
Solução:	Para a execução de uma boa reunião de <i>stand-up meeting</i> alguns aspectos devem ser praticados, são eles: (i) como forma de ganhar mais tempo para assuntos relevantes, não deve ser discutido na reunião o que foi feito ontem; (ii) uma gestão da comunicação deve ser feito, pensando em qual ordem os participantes devem falar e distribuindo o tempo de fala igualmente entre os participantes; (iii) a reunião deve ocorrer antes do horário de almoço; e (iv) deve-se definir a frequência de participação das equipes na reunião.
Referência:	Stray, Moe e Sjoberg (2020), esta solução é apresentada, no formato de padrão de software, na subseção 6.2. O padrão se chama <i>Unproductive Stand-Up Meetings</i>

Fonte: elaborada pelo autor.

6.2 Padrões de Software

De acordo com a aplicação do estudo literário foi possível identificar três trabalhos que possuem soluções, para o contexto da comunicação, bem descritas e que é factível a sua documentação no formato de padrão de software. A seguir são apresentadas as soluções propostas por (SANTOS *et al.*, 2014), (MAK; KRUCHTEN, 2006) e (STRAY; MOE; SJOBERG, 2020).

6.2.1 Primeiro Padrão: Communication Owner

Nome: Communication Owner

Contexto:

Equipes de desenvolvimento de software, que possuem equipes distribuídas e usam metodologia Scrum, possuem a comunicação afetada devido à baixa exigência de documentação e ao dinamismo das informações geradas nos projetos. Os membros dessas equipes devem ser capazes de:

- Compreender como as tarefas são desenvolvidas; e
- Quais objetivos deverão ser alcançados.

Deste modo, a comunicação existente neste ambiente, necessita de recursos que auxiliem a troca de informações, promovendo a compreensão mútua dos membros da equipe.

Problema:

Como devo evitar a perda de informações e manter os *stakeholders* constantemente informados em projetos em que a equipe se encontra separada geograficamente?

Forças:

- A troca de informações entre os *stakeholders* é imprescindível para o desenvolvimento do software. Porém, a pouca documentação e o dinamismo das informações geradas presentes na metodologia Scrum, proporciona uma complexa comunicação entre equipes separadas geograficamente.
- A perda de informações entre os *stakeholders* pode ocasionar a construção de um produto de software que não atende as necessidades e expectativas dos clientes.
- Os *stakeholders* devem ser comunicados constantemente sobre as informações referentes ao projeto.

Solução:

Crie um novo papel dentro da estrutura de desenvolvimento Scrum, o qual deve ser chamado de “*Communication Owner*”. O responsável por esta função herda algumas obrigações destinadas ao “*Product Owner*”. O objetivo do *Communication Owner* é

intermediar as trocas de informações que ocorrem entre os clientes ou parceiros e os times de desenvolvimento da empresa.

As atribuições do *Product Owner* no Scrum, apresentadas por Pham e Pham (2012), são:

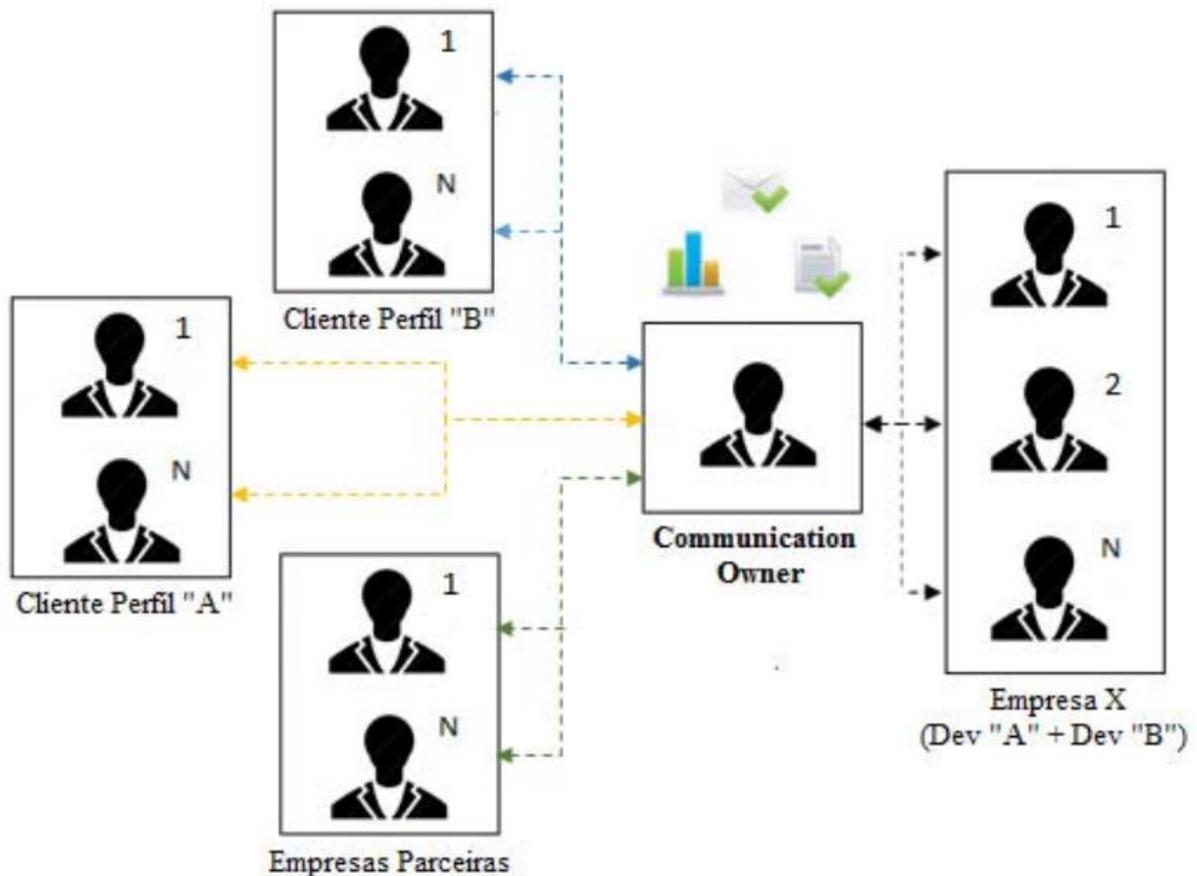
- Saber gerenciar as características e expectativas dos *stakeholders*;
- Conhecer o produto em desenvolvimento;
- Alcançar os requisitos e transforma-los em *Product Backlog*;
- Proporcionar disponibilidade para engajamento constante com a equipe;
- Manusear múltiplas atividades;
- Manter a confiabilidade do time comunicando a todos os *stakeholders* a visão do produto;
- Liderar, negociar, guiar, treinar e auxiliar a equipe quando necessário.

Ao definir um responsável pelo o papel de *Communication Owner*, além das funções herdadas do *Product Owner* citadas anteriormente, ele terá que:

- Manter contato com os clientes, especificando os requisitos coletados por meio das informações reportadas por eles;
- Comunicar aos *stakeholders* a respeito dos requisitos coletados;
- Documentar e compartilhar, por meio de um repositório, as informações a todos os envolvidos.
- Gerenciar junto ao *Scrum Master* o *Product Backlog*, que deve possuir os requisitos organizados por prioridade;
- Esclarecer aos clientes ou parceiros as dúvidas referentes ao escopo e regras de negócio.

A presença do *Communication Owner* é mostrada no modelo de comunicação na seguinte Figura 4.

Figura 4 – Modelo Comunicacional



Fonte: retirado de Santos *et al.* (2014)

De acordo com a Figura 4, é possível visualizar que o *Communication Owner* faz uma intermediação entre os clientes ou empresas parceiras com os desenvolvedores da organização. Este intermédio ocorre de forma a comunicar as informações coletadas da origem aos destinatários, ou seja, de cliente ou empresas parceiras para os desenvolvedores, ou de desenvolvedores para cliente ou empresas parceiras. No entanto, este modelo não impede que desenvolvedores se comuniquem diretamente com os clientes ou parceiros, porém esta comunicação só deve ocorrer em ocasiões de extrema necessidade para a discussão de assuntos técnicos.

Contexto Resultante:

No primeiro caso apresentado no Contexto, se não conduzida a solução para perdas de informações do projeto ocasionada pela existência de uma complexa comunicação possuída por equipes distribuídas, pode-se ocorrer um errôneo entendimento sobre os requisitos, o que impactara na construção de um produto que não atinja as necessidades e

expectativas dos clientes ou empresas parceiras.

A criação do papel *Communication Owner* proporciona ao projeto um melhor entendimento do escopo de atividades e os stakeholders são frequentemente comunicados sobre informações pertinentes ao projeto. Além disso, proporciona também ao time de desenvolvimento um ganho em relação ao foco e a produtividade na entrega de atividades, essa contribuição ocorre devido a existência de um profissional responsável pela a comunicação e a gestão de conhecimento.

Padrões Relacionados:

Conflict Manager e Measurable System Qualities são padrões relacionados.

Usos Conhecidos:

- Santos *et al.* (2014)

A solução apresentada neste padrão de software *Communication Owner* foi proposta pelo trabalho de (SANTOS *et al.*, 2014). O seu trabalho executa esta solução em uma empresa em que os times são formados por membros de diferentes níveis de conhecimento, além disso, os times são separados geograficamente.

A solução proposta foi aplicada e analisada em um projeto com a participação de uma equipe de desenvolvimento e de um cliente X. Como mecanismo de análise, ocorreu a investigação em dois períodos, antes e depois da criação do papel do *Communication Owner*.

O projeto antes da participação do *Communication Owner*, possuía como cenário a reportagem das ocorrências diretamente ao time de desenvolvimento, totalizando 806 ocorrências. Após a criação do *Communication Owner* foi possível identificar 782 ocorrências ao responsável por este papel e apenas 37 ocorrências reportadas diretamente ao time de desenvolvimento.

Como benefícios da criação do *Communication Owner* foi possível identificar que: (i) a produtividade e o foco do time de desenvolvimento melhoraram; (ii) maior qualidade no fluxo comunicacional; (iii) melhoria na qualidade do escopo do projeto; e (iv) distribuição de informações a todos os *stakeholders*.

- ISO 9001:2015 – Sistema de Gestão da Qualidade

É um guia que auxilia organizações na gestão da qualidade. Dois dos princípios abordados pela ISO 9001 são: (i) ocorrência da gestão do relacionamento e (ii) gestão do

engajamento das pessoas.

A ISO 9001 aborda que os papéis organizacionais devem ser bem definidos, de forma que as informações sejam destinadas aos indivíduos que realmente necessitam obter estes conhecimentos. Proporcionando um melhor foco, dado que os indivíduos não receberam informações irrelevantes para a execução das suas atividades.

- Rubin (2012)

O trabalho proposto por Rubin (2012) serve como guia para aplicação da metodologia Scrum. Nele, são apresentados as atividades e ferramentas que devem ser utilizadas na metodologia.

Segundo Rubin (2012), o maior motivo para o fracasso na construção de um software se deve à falta de comunicação ou a sua execução de forma errônea. Sendo assim, deve-se sempre buscar melhores práticas comunicacionais, garantindo que os *stakeholders* impactados por uma determinada informação recebam este conhecimento. Rubin (2012) também informa que deve existir algum papel que intermedia a comunicação entre cliente e desenvolvedor, seja ele de *Product Owner* ou algum outro papel responsável pela intermediação.

6.2.2 Segundo Padrão: Team and Task Prioritization

Nome: Team and Task Prioritization.

Contexto:

Equipes de desenvolvimento de software possuem uma grande quantidade de informações pertinentes ao projeto, estas informações são transformadas em tarefas que devem ser executadas para que o produto final seja alcançado. Porém, existe uma dependência entre os artefatos gerados entre elas, o que torna necessário a prática de priorização de tarefas. Entretanto a dificuldade de troca de informações com os membros da equipe pode causar um negativo impacto na priorização de tarefas, podendo assim:

- Acarretar no atraso da entrega do produto;
- Aumento do custo devido ao atraso; e
- Impactar negativamente as expectativas dos clientes e usuários.

Os Clientes são os responsáveis por financiar o projeto enquanto os usuários usam diretamente o produto.

Problema:

Como devo priorizar as tarefas com base na dependência dos seus artefatos?

Forças:

- Priorizar tarefas é de extrema importância dada a quantidade de informações pertencentes a um projeto, no entanto, essa priorização não é alcançada facilmente sem o uso de uma técnica;
- Priorização de tarefas com base em suas dependências reduz a ociosidade da equipe e contribui para que o produto seja finalizado no tempo estipulado;
- Para a priorização de tarefas deve-se ter conhecimento de informações precisas sobre a sua execução, sendo assim, o gerente deve manter a comunicação com as equipes.

Solução:

Um bom gerente de projeto deve priorizar corretamente as tarefas do projeto com base nas relações dos artefatos gerados. Inicialmente o gerente deve ser capaz de destinar essas tarefas aos respectivos grupos responsáveis pela a sua execução. Tais responsabilidades

devem ser designadas de acordo com a capacidade, carga de trabalho, semelhanças de tarefas e relacionamento entre os membros da equipe. Para ter conhecimento sobre as competências o gerente deve por meio da comunicação conhecer os fatores humanos presentes na equipe.

Após a designação das responsabilidades, o gerente deve se manter comunicado com as equipes de cada tarefa para assim definir os artefatos gerados e os de dependência.

Em seguida, o gerente deve definir uma priorização de tarefas com base na relação dos artefatos, para isso ele deve aplicar a seguinte fórmula apresentada no Quadro 1:

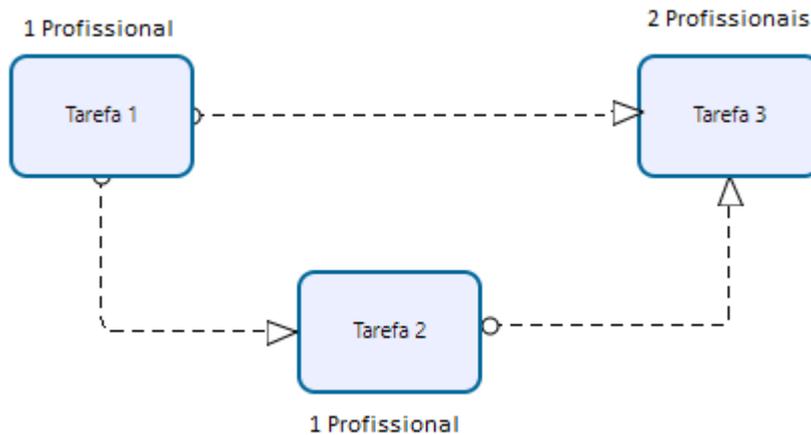
Quadro 1 – Fórmula para priorização de tarefas

Fórmula:	
$PS_{\text{artefato}} = \beta_{\text{tarefas}} \times R_{\text{tarefas}} + \beta_{\text{equipe}} \times R_{\text{equipe}}$	
Elementos	Descrição
PS_{artefato}	Pontuação de uma tarefa T.
β_{tarefas}	Fator de peso definido pelo gerente do projeto de acordo com a importância da atividade T.
R_{tarefas}	Número de tarefas que solicitam os artefatos produzidos por T.
β_{equipe}	Fator de peso definido pelos membros das equipes de acordo com a importância do artefato gerado por T.
R_{equipe}	Número de membros que produzem ou solicitam os artefatos produzidos por T.

O gerente deve determinar junto a equipe os pesos de importância (β_{tarefas} e β_{equipe}).

Como exemplificação da aplicação desta fórmula, podemos imaginar um projeto composto por quatro profissionais, e que as informações do projeto foram divididas em três atividades. Após o gerente definir as responsabilidades e acordar por meio da comunicação quais artefatos cada tarefa necessita, foi possível estruturar o projeto da seguinte maneira apresentado na Figura 5.

Figura 5 – Exemplificação da estruturação das atividades de um projeto.



Fonte: próprio autor.

Podemos analisar na Figura 5 que as dependências de artefato ocorrem de forma que:

- A Tarefa 1 não possui dependência de atividades e possui 1 profissional responsável pela sua execução;
- A Tarefa 2 possui dependência do artefato gerado pela Tarefa 1 e possui 1 profissional responsável pela execução da atividade;
- A Tarefa 3 possui dependência dos artefatos gerados pela Tarefa 1 e Tarefa 2 e possui 2 profissionais responsável pela execução da atividade.

Após uma análise da importância das tarefas, o gerente determinou que todas as tarefas possuíam peso igual, um peso 1, logo $\beta_{tarefas}$ é igual a 1.

Após uma análise da importância dos artefatos, os membros determinaram peso 1, logo β_{equipe} é igual a 1.

De acordo com essas informações, o gerente pode priorizar as tarefas de acordo com a pontuação calculada pela fórmula apresentada no Quadro 1.

As seguintes Tabelas 5, 6 e 7 apresentam o cálculo de pontuação das Tarefas 1, 2 e 3 demonstradas na Figura 5. Os cálculos são feitos por meio da utilização da fórmula apresentada no Quadro 1.

Tabela 5 – Cálculo da Pontuação da Tarefa 1

Nome da Tarefa:	Tarefa 1
Considerações:	<ul style="list-style-type: none"> • $\beta_{tarefas}$ e $\beta_{equipes}$ são iguais a 1; • $R_{tarefas}$ é igual a 2, já que são 2 tarefas que solicitam a Tarefa 1; • $R_{equipes}$ é igual a 4, pois 3 profissionais (2 profissionais da Tarefa 3 e 1 profissional da Tarefa 2) solicitam os artefatos produzidos pela Tarefa 1, e 1 é responsável pela execução da Tarefa 1.
Cálculo da Pontuação:	$PSt1 = 1 \times 2 + 1 \times 4$ $PSt1 = 6$

Fonte: próprio autor.

Tabela 6 – Cálculo da Pontuação da Tarefa 2

Nome da Tarefa:	Tarefa 2
Considerações:	<ul style="list-style-type: none"> • $\beta_{tarefas}$ e $\beta_{equipes}$ são iguais a 1; • $R_{tarefas}$ é igual a 1, já que apenas a Tarefa 3 solicita o artefato da Tarefa 2; • $R_{equipes}$ é igual a 3, pois os 2 profissionais da Tarefa 3 solicitam os artefatos produzidos pela Tarefa 2, e 1 é responsável pela execução da Tarefa 2.
Cálculo da Pontuação:	$PSt2 = 1 \times 1 + 1 \times 3$ $PSt2 = 4$

Fonte: próprio autor.

Tabela 7 – Cálculo da Pontuação da Tarefa 3

Nome da Tarefa:	Tarefa 3
Considerações:	<ul style="list-style-type: none"> • βtarefas e βequipes são iguais a 1; • Rtarefas é igual a 0, já que nenhuma tarefa solicita artefatos da Tarefa 3; • Requeses é igual a 2, pois não recebe nenhuma solicitação e possui 2 responsáveis pela execução da Tarefa 3.
Cálculo da Pontuação:	$PSt3 = 1 \times 0 + 1 \times 2$ $PSt3 = 3$

Fonte: próprio autor.

Após o cálculo das pontuações das Tarefas 1, 2 e 3 com a aplicação da fórmula apresentada no Quadro 1, é possível priorizar as tarefas. A Tabela 8 apresenta a ordenação das tarefas, da maior pontuação para a menor.

Tabela 8 – Priorização das tarefas

Tarefa	Pontuação
Tarefa 1	6
Tarefa 2	4
Tarefa 3	3

Fonte: próprio autor.

Contexto Resultante:

No primeiro caso apresentado no Contexto, a não utilização de uma técnica de priorização de tarefas com base nas suas dependências, pode acarretar em um atraso na entrega do produto por motivo de uma não sequência definida de desenvolvimento das tarefas.

Com a utilização do método proposto para priorização de tarefas com base nas suas dependências, e que para a sua execução utiliza informações do gerente e dos membros da equipe, permite que os envolvidos na execução das tarefas forneçam feedback e participem do processo de priorização, proporcionando o conhecimento mútuo entre todos da equipe.

Padrões Relacionados:

Agree on Quality Targets é um padrão relacionado.

Usos Conhecidos:

- Mak e Kruchten (2006)

A solução apresentada neste padrão de software *Team and Task Prioritization* foi proposta pelo trabalho de (MAK; KRUCHTEN, 2006). Visto que as atividades executadas durante um projeto possuem dependências de artefatos, a não priorização das atividades de acordo com essas dependências pode acarretar em atrasos na entrega do produto final. Além do mais, a comunicação deve ser mantida entre os membros da equipe para que ocorra o um bom entendimento das dependências das atividades. Pensando nisso, Mak e Kruchten (2006) propôs a solução, documentada neste padrão, para priorização de tarefas com base nas dependências de artefatos.

- PMBOK (2017)

O PMBOK é um guia de boas práticas para o ambiente de desenvolvimento de software. No PMBOK são abordados processos, áreas de conhecimento, ferramentas e técnicas, estes aspectos são abordados como forma de facilitar a gerência de um projeto.

Dentre as abordagens apresentadas no PMBOK (2017), uma delas é a priorização de tarefas. Segundo PMBOK (2017), a priorização das tarefas presentes no desenvolvimento de um projeto proporciona aumento na produtividade da equipe.

- Estrada *et al.* (2011)

O trabalho proposto por Estrada *et al.* (2011) possui como temática a gestão do tempo. Dentre os aspectos presentes nesta gestão se encontra a priorização de tarefas. Segundo Estrada *et al.* (2011), ao priorizar as tarefas, podemos gerenciá-las melhor, e com isso, ganhamos maior confiança para a sua execução.

6.2.3 Terceiro Padrão: Unproductive Stand-Up Meetings

Nome: Unproductive Stand-Up Meetings

Contexto:

A prática de *stand-up meeting* é bastante utilizada em equipes que usam a metodologia Scrum, esta prática ocorre diariamente, por meio de reuniões, com o propósito de melhorar a comunicação entre os membros da equipe. Essas reuniões possuem duração média de 15 minutos e ocorrem discussões de informações por meio da resposta de três perguntas, são elas:

- O que eu fiz ontem?
- O que vou fazer hoje?
- Eu vejo algum impedimento?

Apesar das claras diretrizes da prática do *stand-up meeting*, executá-lo diariamente de forma a favorecer todos os membros é uma tarefa desafiadora, isso se deve a alguns problemas enfrentados em sua prática, são eles:

- O *Scrum Master* utiliza as reuniões como forma de identificar o progresso da execução da atividade, prática essa que cria uma reunião de reportagem do progresso e pode afetar a comunicação, já que os membros da equipe não visualizam um claro benefício proporcionado por essas informações, isso se deve ao fato de que boa parte do tempo da reunião os membros pensam em expor seus problemas como forma de buscar soluções.
- Durante a prática é comum que um membro possua mais tempo para falar do que os outros. Normalmente quem fala primeiro possui mais tempo e quem fala por último possui menos tempo dado que a reunião está próxima de acabar. Sendo assim, é um ponto negativo, por exemplo, se um programador possuir mais tempo de comunicação do que o testador.
- A prática do *stand-up meeting* normalmente ocorre durante a manhã, algumas equipes praticam logo no início do horário de expediente como forma de descobrir no que os membros vão trabalhar no dia e como forma de apresentar os impedimentos. Porém, devido ao atraso dos membros, a reunião ocorrendo neste horário tende a durar mais tempo.

- O *stand-up* meeting ocorre diariamente, porém o benefício dessa frequência não ocorre em todas as equipes. Por exemplo, pequenas equipes que possuem elevada comunicação talvez não precisem participar das reuniões diariamente.

Problema:

Como posso ajustar a prática do *stand-up meeting* para adquirir um maior desempenho na execução das reuniões?

Forças:

- A não adequação dos assuntos debatidos de acordo com o interesse dos envolvidos durante o *stand-up meeting* pode ocasionar uma reunião improdutivo, dado que os participantes não possuem motivação.
- A errônea distribuição do tempo de comunicação entre os participantes cria uma incapacidade de fala dos envolvidos, ocasionando a perda de informações que poderiam ser de extrema importância para o projeto.
- A não definição de um adequado horário para a ocorrência da reunião pode ocasionar a necessidade de exceder o tempo determinado.

Solução:

- **Não relate o que foi feito ontem**

Como forma de impedir a prática da reportagem do progresso das atividades, em que os seus benefícios não são claros aos membros, não se deve reportar o que foi feito no dia anterior durante a reunião. Apesar da filosofia do Scrum informar que o *stand-up meeting* não deve ser utilizado para a discussão de soluções, a abordagem de problemas afeta positivamente nas reuniões, já que os membros reconhecem a importância de possuir um tempo para discussão como forma de adquirir uma solução (STRAY; MOE; SJOBERG, 2020).

Dada a importância da reportagem do progresso, esta atividade pode ser feita por meio da utilização de uma ferramenta que possibilite a visualização de todos os membros da equipe sobre a progressão da execução da atividade.

- **Gestão da comunicação**

Deve-se planejar como ocorrerá a comunicação durante o *stand-up meeting*, de forma que seja descoberta como as informações se estruturam, podendo assim definir um fluxo comunicacional. Sabendo a ordem das informações a partir do conhecimento desse fluxo é possível definir uma ordem de quem deve falar primeiro, e na prática da ação, todos os membros devem possuir o mesmo tempo de comunicação.

Além disso, dado que o *stand-up meeting* possui um responsável pela atividade de coordenar a reunião, chamado de facilitador, e que normalmente este papel pertence ao *Scrum Master*, uma atividade que pode ser aplicada é o revezamento do papel de facilitador entre os membros de uma equipe. O rodízio proporciona a existência de vários pontos de vista em relação a execução do *stand-up meeting*.

- **Análise do horário da reunião**

Deve-se ocorrer a análise de qual é o melhor horário para a prática do *stand-up meeting*. Um bom horário é um pouco antes do horário do almoço, este horário soluciona a problemática de atraso entre os participantes da reunião, dado que os membros já estarão presentes na organização. Um outro ponto positivo é que o tempo destinado para a reunião será respeitado, já que os envolvidos são estimulados pela atividade de almoçar que contribui para o fim da reunião no horário atribuído, além disso, devido a hora, os membros são estimulados a almoçar juntos, o que contribui para a continuação do debate ocorrido durante a reunião.

- **Análise da frequência**

Deve-se planejar quais os membros da equipe serão impactados com a prática da reunião, para que assim seja ajustada a frequência de participação destes membros. Ao membro visualizar a importância da sua participação somente quando necessário em uma reunião contribui com para a sua motivação.

Contexto Resultante:

A não adequação da prática do *stand-up meeting*, de acordo com a realidade vivenciada pela organização, pode impactar negativamente nos seguintes aspectos:

- Baixa motivação dos envolvidos dada a não identificação da importância das informações trocadas;
- Divisão desigual do tempo entre os participantes;
- A não prática do tempo estipulado para a reunião;

Por meio das devidas adequações é possível contribuir de forma que os participantes se sintam motivados a participar das reuniões. Possibilitando: (i) assuntos debatidos mais atraentes; (ii) oportunidade para que todos os participantes possuam voz; e (iii) ambiente mais agradável, dado que só terá participação dos membros afetados pelo assunto da reunião e que proporciona um horário confortável para todos.

Padrões Relacionados:

Conflict Manager e Find Essential Qualities são padrões relacionados.

Usos Conhecidos:

- Stray, Moe e Sjoberg (2020)

A solução apresentada neste padrão de software *Unproductive Stand-Up Meetings* foi proposta pelo trabalho de (STRAY; MOE; SJOBERG, 2020). Visando identificar melhores práticas para a execução de reuniões de *stand-up meetings* foram analisadas 102 reuniões, possibilitando a identificação das boas práticas documentadas na solução deste padrão.

Segundo Stray, Moe e Sjoberg (2020) as reuniões devem ser constantemente inspecionadas para proporcionar uma melhor execução da prática de *stand-up meeting*.

A prática das indicações, documentadas na solução deste padrão, proporciona maior produtividade na execução das reuniões.

- Rubin (2012) e Yip (2006)

O trabalho proposto por Rubin (2012) serve como guia para aplicação da metodologia Scrum. Nele, é abordado a prática de reuniões de *stand-up meeting*, bem como apresentado as boas práticas para execução da reunião, são elas: (i) para a execução da reunião, deve-se planejar qual assunto será debatido; (ii) deve-se garantir o término da reunião no horário determinado; (iii) as reuniões devem ser curtas, com duração média de 15 minutos; e (iv) as reuniões devem possuir um facilitador, seja ele o *Scrum Master* ou algum outro

membro da equipe. Da mesma forma, Yip (2006) também identifica boas práticas para a execução do *stand-up meeting*, são elas: (i) ao ser reportado algum obstáculo, deve-se fornecer total apoio para que o mesmo seja superado; (ii) devem participar das reuniões somente os indivíduos afetados pelas informações; e (iii) o *Scrum Master* deve intervir caso o assunto abordado não seja o de maior prioridade.

7 CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho apresentou a documentação de três soluções comunicacionais no formato de padrão de software, são eles: (i) *Communication Owner*; (ii) *Team and Task Priorization*; e (iii) *Unproductive Stand-Up Meetings*. Para isso, como procedimento metodológico, algumas atividades foram executadas, são elas: (i) Realização do estudo bibliográfico; (ii) Listagem de problemas e indicações de soluções; (iii) Categorização dos problemas; e (iv) Documentação das soluções no formato de padrão.

Na realização do estudo bibliográfico foi aplicada como estratégia de pesquisa a utilização de duas *strings* de buscas que foram executadas nas bases de dados da IEEE (*IEEE Xplore*) e a ACM (*ACM Digital Library*), resultando no total de 953 trabalhos na IEEE e 512 na ACM. Na IEEE todos os trabalhos retornados foram analisados, já na ACM ocorreu a análise de 270 trabalhos. Como estratégia de análise destes trabalhos, foram realizadas duas etapas de seleção. Na primeira etapa ocorreu a análise do título e do resumo de todos os trabalhos, e foram selecionados os que abordavam problemas e soluções no âmbito da comunicação interpessoal no ambiente de desenvolvimento de software, sendo selecionado 92 trabalhos. Já na segunda todos os trabalhos selecionados na primeira etapa foram lidos por completo, em que resultou em 29 trabalhos selecionados.

Após a etapa de estudo bibliográfico, foi realizada a listagem de nove problemas e indicações de soluções identificadas, bem como uma categorização foi feita, a Tabela 4 apresenta estes aspectos. O propósito da Tabela 4 é servir de catálogo de problemas e soluções no âmbito da comunicação interpessoal.

De acordo com a aplicação do estudo bibliográfico foi possível identificar três trabalhos que possuem soluções bem descritas e que é factível a sua documentação no formato de padrão de software. Sendo assim, os trabalhos propostos por Santos *et al.* (2014), Mak e Kruchten (2006) e Stray, Moe e Sjoberg (2020) foram documentados no formato de padrão de software.

Como perspectiva futura deste trabalho, com o propósito de avaliar suas eficiências, pretende-se aplicar os padrões documentados em projetos de desenvolvimento de software. Além disso, pretende-se também, analisar os trabalhos restantes retornados pela base de dados da ACM, identificando mais problemas e soluções comunicacionais para a documentação de mais padrões de software.

REFERÊNCIAS

- ALEXANDER, Christopher. **A pattern language: towns, buildings, construction**. Oxford university press, 1977.
- ALY, Amir; TAPUS, Adriana. "**A model for synthesizing a combined verbal and nonverbal behavior based on personality traits in human-robot interaction.**" 2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI). IEEE, 2013
- ANWER, Sajid; WEN, Lian; WANG, Zhe. "**A systematic approach for identifying requirement change management challenges: preliminary results.**" Proceedings of the Evaluation and Assessment on Software Engineering. 2019. 230-235.
- BEECHAM, Sarah; NOLL, John; RICHARDSON, Ita. "**Using agile practices to solve global software development problems--a case study.**" 2014 IEEE International Conference on Global Software Engineering Workshops. IEEE, 2014.
- BERLO, David K. **O processo da comunicação: Introdução à teoria e à prática**. São Paulo: Martins Fontes, 2003.
- BETTENBURG, Nicolas. "**Studying the Impact of Developer Communication on the Quality and Evolution of a Software System: A Doctoral Dissertation Retrospective.**" 2014 IEEE International Conference on Software Maintenance and Evolution. IEEE, 2014.
- BODEN, Alexander; NETT, Bernhard; WULF, Volker. "**Coordination practices in distributed software development of small enterprises.**" International Conference on Global Software Engineering (ICGSE 2007). IEEE, 2007.
- BRAHAN, J. "**Technical skepticism will always be important'.**" Machine design 69.5 (1997): 163-165.
- BRAHM, Taiga; KUNZE, Florian. "**The role of trust climate in virtual teams.**" Journal of Managerial Psychology (2012).
- CAMARGO, Marta Rocha. **Gerenciamento de projetos: fundamentos e prática integrada**. 2. ed. Rio de Janeiro: Elsevier, 2018.
- CAMPAGNOLO, B.; TACLA, C. A.; PARAISO, E. C.; SATO, G. Y.; RAMOS, M. P. "**An architecture for supporting small collocated teams in cooperative software development.**" 2009 13th International Conference on Computer Supported Cooperative Work in Design. IEEE, 2009.

CHIAVENATO, Idalberto. **Comportamento Organizacional: a dinâmica do sucesso das organizações**. 2ed. Rio de Janeiro: Elsevier, 2010.

COPLIEN, J. O. **Software Patterns**, SIGS books and Multimedia, June 1996.

CRUZ, Fábio. **Scrum e PMBOK unidos no Gerenciamento de Projetos**. Brasport, 2013.

DAMIAN, Daniela. "**An empirical study of requirements engineering in distributed software projects: is distance negotiation more effective?**" Proceedings Eighth Asia-Pacific Software Engineering Conference. IEEE, 2001.

DAMIAN, Daniela E.; ZOWGHI, Didar. "**An insight into the interplay between culture, conflict and distance in globally distributed requirements negotiations.**" 36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the. IEEE, 2003.

DAMIAN, Daniela et al. "**Awareness in the wild: Why communication breakdowns occur.**" International Conference on Global Software Engineering (ICGSE 2007). IEEE, 2007.

DANTAS, V. L.L.; CASTRO, R. N. S.; ROSSANA, M. C. Andrade. **Conflict Manager**. In: OOPSLA Mini PloP. 2009.

DELFINO, Samyr Santos. "**Gestão da informação/comunicação em equipes de desenvolvimento Scrum.**" (2013).

FOX, S. **Effective communication: stone age to e-comm**. Proceedings of the Project Management Institute Annual Seminars & Symposium, USA: 2001

HASSAN, Mujtaba et al. "**A Policy Recommendations Framework To Resolve Global Software Development Issues.**" 2019 International Conference on Innovative Computing (ICIC). IEEE, 2019.

HAUSE, Martha L. "**Distributed team performance in software development.**" Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education. 2005.

HOLMSTROM, H.; CONCHÚIR, E. Ó.; AGERFALK, J.; FITZGERALD, B. "**Global software development challenges: A case study on temporal, geographical and socio-cultural distance.**" 2006 IEEE International Conference on Global Software Engineering (ICGSE'06). IEEE, 2006.

JUSOH, Y. Y.; NOR, R. N. H.; MAHMOOD, B. A.; WAFEEQ, M. T.; ALI, M. A.; JUSOH, M. N. B. "**Communication management in global software development projects.**" 2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP). IEEE, 2018.

KITCHENHAM, Barbara., **Procedures for Performing Systematic Reviews** , 2004.

KORTE, Sievi Outi; SYSTA, Kari; HJELSVOLD, Rune. "**Global vs. local—Experiences from a distributed software project course using agile methodologies.**" 2015 IEEE Frontiers in Education Conference (FIE). IEEE, 2015.

KUMAR, Shreya; WALLACE, Charles. "**Instruction in software project communication through guided inquiry and reflection.**" 2014 IEEE Frontiers in Education Conference (FIE) Proceedings. IEEE, 2014.

LEE, Jason Chong. "**Embracing agile development of usable software systems.**" CHI'06 extended abstracts on Human factors in computing systems. 2006.

LESCHER, Christian; BRUGGE, B. "**Global requirements engineering: decision support for globally distributed projects.**" 2009 Fourth IEEE International Conference on Global Software Engineering. IEEE, 2009.

LU, Xiangnan; LIU, Lin; LIU, Li. "**Relationship Research Between Communication Activities and Success Indexes in Small and Medium Software Projects.**" Information Science and Engineering (ICISE), 2009 1st International Conference on. IEEE, 2009.

MAK, David KM; KRUCHTEN, Philippe B. "**Task coordination in an agile distributed software development environment.**" 2006 Canadian conference on electrical and computer engineering. IEEE, 2006.

McQUAIL, Denis; WINDAHL, Sven. **Communication models for the study of mass communications**. 2. ed. London: Longman, 1993.

MEMON, Kamran Ali; XIAOLING, Xia. "**Deciphering and analyzing software requirements employing the techniques of natural language processing.**" Proceedings of the 2019 4th International Conference on Mathematics and Artificial Intelligence. 2019.

MONASOR, Miguel J., et al. "**Towards a global software development community web: Identifying patterns and scenarios.**" 2013 IEEE 8th International Conference on Global Software Engineering Workshops. IEEE, 2013.

OLIVEIRA, Rubi Nei Machado. "**O processo de comunicação entre usuários e analistas de sistemas no contexto das organizações.**", 2010.

- ORAN, Ana Carolina. **"A set of artifacts and models to support requirements communication based on perspectives."** ACM SIGSOFT Software Engineering Notes 41.6 (2017): 1-5.
- PHAM, Andrew; PHAM, Phuong-Van. **Scrum em ação.** Novatec Editora, 2012.
- PHILLIPS, Shaun; ZIMMERMANN, Thomas; BIRD, Christian. **"Understanding and improving software build teams."** Proceedings of the 36th international conference on software engineering. 2014.
- PMI. **Um guia do conhecimento em gerenciamento de projetos.** Guia PMBOK 6a edição – EUA: Project Management Institute, 2017.
- PUARUNGROJ, W.; BOONSIRISUMPUN, N.; PHROMKHOT, S.; PUARUNGROJ, N. **"Dealing with change in software development: a challenge for requirements engineering."** 2018 3rd Technology Innovation Management and Engineering Science International Conference (TIMES-iCON). IEEE, 2018.
- RISING, Linda. **"Patterns: A way to reuse expertise."** IEEE Communications Magazine 37.4 (1999): 34-36.
- RUTZ, Leah; TANNER, Maureen. **"Factors that influence performance in global virtual teams in outsourced software development projects."** 2016 IEEE International Conference on Emerging Technologies and Innovative Business Practices for the Transformation of Societies (EmergiTech). IEEE, 2016.
- SALGER, Frank; ENGLERT, Jochen; ENGELS, Gregor. **"Towards specification patterns for global software development projects-experiences from the industry."** 2010 Seventh International Conference on the Quality of Information and Communications Technology. IEEE, 2010.
- SANTOS, Misael S. **Uma Proposta para a Integração de Modelos de Padrões de Software com Ferramentas de Apoio ao Desenvolvimento de Sistemas.** Diss. Dissertação de Mestrado, Universidade Federal do Ceará, 2004.
- SANTOS, A., et al. **"Improved communication in distributed agile software development."** 2014 9th Iberian Conference on Information Systems and Technologies (CISTI). IEEE, 2014.
- STRAY, Viktoria; MOE, Nils Brede; SJOBERG, Dag IK. **"Daily stand-up meetings: start breaking the rules."** IEEE Software 37.3 (2020): 70-77.
- SUBRAMANYAM, Ramanath, et al. **"User participation in software development projects."** Communications of the ACM 53.3 (2010): 137-141.

TENENBERG, Josh. "**An institutional analysis of software teams.**" International Journal of Human-Computer Studies 66.7 (2008): 484-494.

VLISSIDES, John. "**Patterns: The top 10 misconceptions.**" Object Magazine 7 (1997): 30-33.

WU, Shujian. "**Overview of communication in global software development process.**" Proceedings of 2012 IEEE International Conference on Service Operations and Logistics, and Informatics. IEEE, 2012.