



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

CAIO FELIPE DE SOUSA ROCHA

**SOFTWARE ARCHITECTURE CANVAS: FERRAMENTA DE SUPORTE À
COMUNICAÇÃO EM ANÁLISE E PROJETO DE ARQUITETURA DE SOFTWARE**

QUIXADÁ

2022

CAIO FELIPE DE SOUSA ROCHA

SOFTWARE ARCHITECTURE CANVAS: FERRAMENTA DE SUPORTE À
COMUNICAÇÃO EM ANÁLISE E PROJETO DE ARQUITETURA DE SOFTWARE

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

Orientador: Prof. Me. Carlos Diego Andrade de Almeida

QUIXADÁ

2022

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)

R572 Rocha, Caio Felipe de Sousa.
Software architecture canvas: ferramenta de suporte à comunicação em análise e projeto de arquitetura de software / Caio Felipe de Sousa Rocha. – 2022.
101 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Engenharia de Software, Quixadá, 2022.
Orientação: Prof. Me. Carlos Diego Andrade de Almeida.

1. Arquitetura de Software. 2. Quadro de modelo de negócios. 3. Comunicação. I. Título.

CDD 005.1

CAIO FELIPE DE SOUSA ROCHA

SOFTWARE ARCHITECTURE CANVAS: FERRAMENTA DE SUPORTE À
COMUNICAÇÃO EM ANÁLISE E PROJETO DE ARQUITETURA DE SOFTWARE

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

Aprovada em: ___/___/_____

BANCA EXAMINADORA

Prof. Me. Carlos Diego Andrade de
Almeida (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Enyo José Tavares Gonçalves
Universidade Federal do Ceará (UFC)

Prof. Me. Camilo Camilo Almendra
Universidade Federal do Ceará (UFC)

Aos persistentes e àqueles que precisaram de
mais uma chance...

AGRADECIMENTOS

À UFC e todos os que compõem essa magnífica instituição. À minha mãe pelo apoio incondicional e aos meus familiares pelos exemplos de esforço e superação.

À Carla pelo apoio e pela companhia nesses anos árduos. E aos amigos, que tornaram essa caminhada mais agradável.

A todos os professores que fizeram parte desse caminho. A todos que me ofereceram mais uma oportunidade. Ao professor Diego, pela ajuda e paciência com esse trabalho. Agradeço aos professores participantes da banca examinadora, pelo tempo, críticas e sugestões que ativamente colaboraram para melhorar esse trabalho.

A todos os que participaram dos testes e aplicações, em especial aos membros das empresas Fluid, nKey e TDSOft. Às revisões da professora Daniele e do amigo Pedro. E a todos que direta ou indiretamente contribuíram para a realização desse trabalho e de minha vida acadêmica.

Muito obrigado!

“Eu não falhei. Apenas descobri 10 mil maneiras
que não funcionam.”

(Thomas Edison)

RESUMO

As atividades de análise e projeto de arquitetura de software (AS) impõem ao arquiteto de software diversos desafios e esforços relacionados à comunicação com *stakeholders*. Uma solução a esse respeito pode ser considerada com a ferramenta *Business Model Canvas* (BMC), ferramenta relacionada à Administração, que provê elementos de comunicação e abstração que facilitam o desenvolvimento de modelos de negócios. Buscando obter os mesmos efeitos do BMC, este trabalho teve como objetivo desenvolver uma ferramenta visual e de comunicação, baseada no BMC aplicável às atividades de análise e projeto de AS. Essa ferramenta foi desenvolvida a partir de uma pesquisa bibliográfica e um mapa conceitual, conduzido um estudo de caso com profissionais de desenvolvimento de software, observação direta e aplicação de questionários. A referida ferramenta estabeleceu elementos semelhantes ao BMC como: uma abstração comum (do metamodelo de arquitetura de software) e linguagem comum entre *stakeholders*, para estabelecer uma comunicação satisfatória na percepção dos participantes. Também foi possível obter um certo nível de satisfação dos participantes quanto aos objetivos das atividades de análise e projeto envolvendo os *stakeholders*. No entanto, foram identificadas ressalvas acerca da representação dos elementos na ferramenta, observando a necessidade de refinamentos na pesquisa bibliográfica e no modelo conceitual, para um aprofundamento da representação desses elementos.

Palavras-chave: Arquitetura de Software. Business Model Canvas. Comunicação.

ABSTRACT

The Software Architecture (SA) analysis and design activities imposes to the software architect many challenges and endeavors related stakeholder communication, a solution about this can be considered with the Business Model Canvas (BMC) tool, which provides abstraction and communication elements and that helps the business model development. Seeking the same effects of BMC, this research had as its main goal to develop a communication and visual tool, based on BMC applied to SA analysis and design activities. This tool was developed from bibliographic research and a conceptual map, leading to a study case with direct observation and survey applied in an assay involving software development professionals. The developed tool established similar elements of BMC as: a common abstraction (of software architecture metamodel) a shared language between stakeholders, in a way to establish a up to par communication in participants perception. It was also possible to check a certain level of acceptability from the participants about SA analysis and design activities stakeholders objectives. However, it identified limitations in tool elements representation, leading to considerations about bibliographic research improvements as in the conceptual model, for an improved representation of these elements.

Keywords: Software Architecture. Business Model Canvas. Communication.

LISTA DE FIGURAS

Figura 1 – Business Model Canvas.	20
Figura 2 – Business Model Ontology.	22
Figura 3 – Fluxograma dos procedimentos metodológicos.	27
Figura 4 – Mapa/Modelo conceitual em Arquitetura de Software.	31
Figura 5 – Software Architecture Canvas.	34
Figura 6 – Respostas Fechadas	46
Figura 7 – Mapa Conceitual de Arquitetura de Software - Em detalhes	79
Figura 8 – Canvas de Arquitetura de Software.	80

LISTA DE ABREVIATURAS E SIGLAS

AS	<i>Arquitetura de Software</i>
BMC	<i>Business Model Canvas</i>
BMO	<i>Business Model Ontology</i>
SH	<i>Stakeholders</i>

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Objetivo Geral	14
1.2	Objetivos Específicos	15
1.3	Estrutura	15
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	Arquitetura de Software: Análise e Projeto	16
2.2	Comunicação em Arquitetura de Software	18
2.3	Business Model Canvas e Business Model Ontology	20
3	TRABALHOS RELACIONADOS	24
3.1	The GRADE Decision Canvas for Classification and Reflection on Architecture Decisions	24
3.2	CacOphoNy: Metamodel-Driven Software Architecture Reconstruction	24
3.3	Strategic Model Canvas: a tool proposition to optimize strategic planning	25
4	PROCEDIMENTOS METODOLÓGICOS	26
5	DESENVOLVIMENTO E APLICAÇÃO	30
6	SOFTWARE ARCHITECTURE CANVAS	34
7	ANÁLISE DOS RESULTADOS DA APLICAÇÃO	38
7.0.1	<i>Perfil demográfico-profissional dos participantes</i>	38
7.1	Relatório da Observação Direta	38
7.2	Respostas Questões Abertas	42
7.2.1	<i>Analista de Negócios</i>	42
7.2.2	<i>Líder de QA</i>	43
7.2.3	<i>Gerente de Projeto</i>	43
7.2.4	<i>Analista de integração</i>	44
7.2.5	<i>Desenvolvedor</i>	44
7.2.6	<i>Arquiteto de Software</i>	45
7.3	Respostas Questões Fechadas	45
7.4	Análise dos resultados da aplicação	47
8	AMEAÇAS À VALIDADE E OUTRAS CONSIDERAÇÕES	51
9	CONCLUSÕES E TRABALHOS FUTUROS	52

REFERÊNCIAS	53
APÊNDICE A – Documento de Captura de Conhecimento	56
APÊNDICE B – Mapa Conceitual de Arquitetura de Software	79
APÊNDICE C – Software Architecture Canvas	80
APÊNDICE D – Carta Proposta do Estudo de Caso	81
APÊNDICE E – Questionário do Estudo de Caso	83
APÊNDICE F – Respostas dos participantes	88

1 INTRODUÇÃO

Software é o elemento-chave na evolução de produtos e sistemas baseados em computador, evoluindo de uma ferramenta especializada em análise de informação e resolução de problemas, para uma indústria propriamente dita (PRESSMAN; MAXIM, 2016, p. 11). As ações de desenvolvimento, testes e implantação do software são impactados pela arquitetura do sistema. Fazendo da *Arquitetura de Software* (AS) uma das mais importantes áreas da Engenharia de Software.

A norma ISO 42010:2011 (ISO/IEC/IEEE, 2011), define AS como os conceitos fundamentais do sistema ou propriedades de um sistema em seu ambiente incorporados em seus elementos, relacionamentos, e nos princípios de sua concepção e evolução.

As atividades da AS impõem diversos desafios ao arquiteto de software ao longo das suas atividades de análise e projeto, entre eles, desdobramentos relacionados a desafios e esforços de comunicação com as partes interessadas ou *stakeholders*, onde o arquiteto tem a complicada tarefa de identificar e especificar requisitos arquiteturalmente significativos em cenários de documentação escassa, tendo que reconhecê-los em meio a demandas, preocupações e interesses divergentes e dispersos entre *Stakeholders* (SH) com objetivos, entendimentos e áreas de atuação diferentes.

Evans e Fowler (2004, p. 15) afirmam que o conhecimento sobre o projeto é fragmentado, disperso entre muitas pessoas e documentos, e misturado com outras informações de forma que nem mesmo é possível saber qual conhecimento se faz necessário. Muitas vezes essas informações podem não estar bem documentadas ou bem comunicadas, Bass *et al.* (2012, p. 52-53) afirmam que é raro um documento de requisitos, que quando existe, faz um bom trabalho ao capturar todos os requisitos de qualidade de um sistema. Estes requisitos, bem como as influências de contextos (técnicos, organizacionais e profissionais) têm um papel essencial no processo decisório do arquiteto, que terá a tarefa de expressar suas decisões por meio de descrições da arquitetura do sistema, comunicando-as aos mesmos stakeholders, com suas já citadas particularidades.

Isso tem impacto notável no trabalho do arquiteto de software. Estudos conduzidos por Dasanayake *et al.* (2015) e Falessi *et al.* (2010), identificaram problemas relacionados a "decisões tomadas sob requisitos instáveis e bastante vagos", "problemas de comunicação com cliente" e "falta de gerenciamento adequado do conhecimento".

Apesar desses desafios inerentes às atividades, a AS representa uma abstração co-

num e fornece uma linguagem que pode possibilitar um alinhamento entre SH e o arquiteto. Porém, como obter esses efeitos para solucionar esses desafios de comunicação? Uma solução pode ser pensada olhando para outra área de conhecimento, a Administração. O desenvolvimento de um modelo de negócio pode atuar de forma semelhante a AS, com o envolvimento e a participação de diversos *stakeholders* tais como: fornecedores, investidores, clientes e colaboradores, para a viabilização de um negócio ou seu modelo. O *Business Model Canvas* (BMC) é uma ferramenta visual criada por Osterwalder (2004) para facilitar o desenvolvimento desses modelos, sintetizando-os nos elementos essenciais de um negócio. A característica visual da ferramenta simplifica os elementos que compõem um modelo de negócios para os usuários, abstraindo questões conceituais e focando apenas no modelo em análise ou construção.

Uma pesquisa apresentada por Amarsy (2015) identificou as principais contribuições do BMC para seus usuários, entre elas, "melhores conversas sobre estratégias", "criação de linguagem compartilhada", "melhoria no trabalho em equipe em brainstorms" e "melhorias na colaboração entre disciplinas e fronteiras geográficas".

A ferramenta foi amplamente adotada por *startups* e projetos inovadores de diversas empresas, e com reconhecimento de sua utilidade e benefícios. Outras variações foram desenvolvidas como: *Project Management Canvas* (para gerência de projetos), *Lean Canvas* (focado para *lean startups*), *Value Proposition Canvas* (para proposta de valor em produtos). Cada uma delas identifica elementos relacionados às suas áreas de conhecimento e são utilizados de forma semelhante ao BMC.

Uma vez identificado o paralelo entre as dificuldades na AS explicadas anteriormente, e as características e benefícios proporcionados pelo BMC, este trabalho construiu e observou a atuação de uma ferramenta visualmente semelhante ao BMC, buscando obter os mesmos efeitos na melhoria da comunicação, estabelecimento de uma linguagem comum e auxílio no trabalho dos participantes, mas aplicada a atividades relacionadas à análise e projeto de AS, para minimizar as dificuldades de comunicação entre o arquiteto de software e SH, bem como os impactos em seus objetivos nessas atividades.

1.1 Objetivo Geral

- Desenvolver uma ferramenta visual e de comunicação, baseada no BMC aplicável às atividades de análise e projeto de Arquitetura de Software.

1.2 Objetivos Específicos

- Identificar um modelo conceitual (em um mapa conceitual) dos elementos de trabalho na análise e projeto de arquitetura de software, a partir de uma pesquisa bibliográfica.
- Construir uma ferramenta semelhante ao *Business Model Canvas* a partir do modelo identificado.
- Avaliar a ferramenta, em um estudo de caso com observação direta e aplicação de questionários.

1.3 Estrutura

Apresenta-se aqui a estrutura dos próximos capítulos deste trabalho.

- Capítulo 1: Introdução - O presente capítulo, onde foram apresentados: Proble-mática, Proposta, Objetivos gerais e específicos.
- Capítulo 2: Fundamentação Teórica - Conceitos-chave relacionados ao trabalho.
- Capítulo 3: Trabalhos Relacionados - Exposição de trabalhos relacionados a esses temas, indicando semelhanças e diferenças.
- Capítulo 4: Procedimentos Metodológicos - Atividades para execução deste trabalho.
- Capítulo 5: Execução: Desenvolvimento e Aplicação - Explicações sobre como os procedimentos metodológicos foram aplicados.
- Capítulo 6: *Software Architecture Canvas* - Apresentação da ferramenta e seus itens.
- Capítulo 7: Apresentação e análise dos resultados - Exposição e interpretação dos dados coletados.
- Capítulo 8: Ameaças à validade e outras considerações - Pontos a serem verifica-dos a respeito da validade deste trabalho.
- Capítulo 9: Conclusões e trabalhos futuros - Considerações finais e possibilidades de próximos trabalhos.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção, estão apresentados os principais conceitos relacionados ao trabalho e à contribuição desses para o seu desenvolvimento.

2.1 Arquitetura de Software: Análise e Projeto

O tema da arquitetura de software passou a ser tratado de modo mais intenso nos anos 90, e hoje há um conhecimento consolidado sobre essa área descrito em conceitos, atividades, técnicas e métodos (como ATAM, SAAM), em padrões e estilos arquiteturais, em normas internacionais (como ISOs 42010:2011, 12207:2017), em corpos de conhecimento (como SWEBoK), em processos (como RUP, OpenUP) e diversas publicações de Hofmeister, Kruchten, Bass, Clements, Cervantes, Kazman, dentre outros autores.

Apesar disso, a literatura não busca estabelecer um processo definitivo de AS. Entretanto, analisando as atividades relatadas em Bass *et al.* (2012), ISO 12207:2008 (ISO/IEC/IEEE, 2017) e Kruchten (2003) é possível identificar as seguintes etapas em comum durante o ciclo de vida do projeto de software: análise de arquitetura, projeto de arquitetura, planejamento da implementação da arquitetura, avaliação ou validação e execução (implementação, manutenção e acompanhamento da arquitetura). Neste trabalho, considerou-se a análise e projeto de arquitetura.

Reekie *et al.* (2006, p. 11) entendem "análise arquitetural" como os esforços para levar um conceito inicial sobre o sistema para uma visão mais detalhada e precisa sobre a arquitetura desse sistema. Bass *et al.* (2012) definem esses esforços em duas atividades: criar um caso de negócios para o sistema, identificando objetivos de negócio e entender os requisitos arquiteturalmente significativos, representados por restrições, atributos de qualidade e requisitos funcionais.

A ISO 12207:2017 (ISO/IEC/IEEE, 2017) define as ações a serem tomadas nessa atividade em: estabelecer e documentar os requisitos de software; avaliar os requisitos segundo a rastreabilidade, consistência, testabilidade, viabilidade de projeto, operação e manutenção; conduzir revisões conjuntas, para formar uma *baseline* desses requisitos.

Ao atingir uma maturidade no entendimento dos requisitos arquiteturalmente significativos do sistema, nessas atividades de análise, o arquiteto poderá iniciar as atividades de projeto de arquitetura, Kruchten (2003, p. 237) explica de forma mais ampla que durante o projeto são tomadas decisões estratégicas e táticas para satisfazer os requisitos funcionais e de

qualidade de um sistema.

O objetivo da etapa de projeto é a composição ou seleção de um projeto arquitetural para o sistema, que busque atender os atributos de qualidade identificados, prover decisões adequadas aos riscos, restrições e conciliar demandas divergentes por *tradeoffs*. A respeito disso, Bass *et al.* (2012) explicam que:

As restrições são satisfeitas aceitando a decisão de projeto e reconciliando-a com outras decisões de projeto afetadas. Um projeto de sistema consiste em uma coleção de decisões. Algumas dessas decisões ajudam a controlar as respostas dos atributos de qualidade; outros garantem a realização da funcionalidade do sistema. (BASS *et al.*, 2012, p. 64)

A ISO 12207:2017 (ISO/IEC/IEEE, 2017) propõe atividades de projeto para: transformar os requisitos, em itens estruturais de software de alto nível e componentes; desenvolver um projeto de alto nível para as interfaces externas e bases de dados, assim como a documentação da arquitetura dos itens de software, componentes e projetos;

Na abordagem do *Processo Unificado* explicada por Kruchten (2003), o arquiteto deve definir uma arquitetura candidata em um esboço inicial, a partir de casos de uso arquiteturalmente significativos; executar o refinamento da arquitetura, onde há a transição da análise para o projeto; análise de comportamento; projeto de componentes do sistema, componentes de tempo real e banco de dados. Para Reekie:

Essa arquitetura não surge do nada, mas é o produto de um processo envolvente e repetitivo, durante o qual as alternativas são analisadas e avaliadas, as alternativas de restrições são analisadas e avaliadas, as restrições e limitações são ponderadas em relação às necessidades das partes interessadas e as realidades de custo, cronograma, e a viabilidade técnica contabilizada. Reekie *et al.* (2006, p. 27)

No processo decisório do projeto os requisitos identificados na análise são considerados durante suas atividades, em especial os não-funcionais que possuem os maiores efeitos na arquitetura do sistema, como confirmado em Chen *et al.* (2013). Táticas e Padrões arquiteturais são selecionados softwares e hardwares disponíveis no mercado (*commercial off-the-shelf*) podem ser adquiridos, componentes de software e dados são modelados e podem passar por um ou mais refinamentos, resultando em projeto de componentes, itens de software e base de dados, assim como esses itens gerados pelo arquiteto serão documentados.

2.2 Comunicação em Arquitetura de Software

Nas atividades de análise de arquitetura o arquiteto deve entender em detalhes os requisitos do sistema e objetivos da organização, necessidades e preocupações dos envolvidos, para destilar requisitos arquiteturalmente significativos. Vale entender que o arquiteto não deve restringir-se a questões técnicas, mas também a questões profissionais, comerciais, econômicas, legais entre outras. Sobre isso Kruchten (2003, p. 70) sustenta que:

A arquitetura de software está relacionada não só a estrutura e comportamento, mas também ao contexto: uso, funcionalidade, desempenho, elasticidade, reutilização, compreensão, restrições e intercâmbios tecnológicos e econômicos, e estéticos. (KRUCHTEN, 2003, p. 70)

Essas informações fluem de uma gama de indivíduos, equipes e organizações. Isso é feito com o empenho constante do arquiteto para interagir com esses envolvidos, Bass *et al.* (2012, p. 52-53) afirmam que o arquiteto deve identificar e envolver ativamente os stakeholders para solicitar suas necessidades e expectativas. Nessa hora a comunicação entra em cena, conceitos e termos técnicos, operacionais e contextuais fazem parte da comunicação desses envolvidos no intercâmbio de informações.

Nesse caso, a arquitetura de software servirá como linguagem entre os envolvidos. Nesse sentido, Smolander (2002) caracteriza, entre outras metáforas, a "arquitetura como linguagem", vendo-a de forma a possibilitar um entendimento comum sobre a estrutura do sistema. Esse entendimento do sistema, dado pela arquitetura de software abrange diferentes SH e suas preocupações, considerando a afirmação de Bass *et al.* (2012, p. 29) onde:

A arquitetura de software representa uma abstração comum de um sistema que a maioria, senão todas, as partes interessadas do sistema podem usar como base para criar entendimento mútuo, negociação, formação de consenso e comunicação entre si. A arquitetura - ou pelo menos partes dela - é suficientemente abstrata para que a maioria das pessoas não técnicas possa entendê-la adequadamente, particularmente com algum treinamento do arquiteto, e ainda assim essa abstração pode ser refinada em especificações técnicas suficientemente ricas para orientar a implementação, integração, teste, e implantação. Cada parte interessada em um sistema de software - cliente, usuário, gerente de projeto, codificador, testador e assim por diante - está preocupado com as diferentes características do sistema que são afetadas por sua arquitetura. A arquitetura fornece uma linguagem comum em que diferentes preocupações podem ser expressas, negociadas e resolvidas em um nível que é intelectualmente gerenciável, mesmo para sistemas grandes e complexos. Sem essa linguagem, é difícil entender grandes sistemas o suficiente para tomar as decisões iniciais que influenciam a qualidade e a utilidade. (BASS *et al.*, 2012, p. 29)

Essa abstração comum citada em Bass, refere-se a elementos de uma possível conceitualização da arquitetura de software, um modelo, definido em Kruchten (2003, p. 236) como uma "abstração semanticamente fechada de um sistema", unindo as abstrações e termos comuns do entendimento desses SH. Um entendimento sobre esse modelo pode ser dado por Evans e Fowler (2004, p. 21) como:

[...] um conjunto de conceitos desenvolvidos nas cabeças das pessoas que participam do projeto, com termos e relações que refletem a visão do domínio. Esses termos e inter-relações fornecem a semântica de uma linguagem que é moldada de acordo com o domínio, sendo ao mesmo tempo precisa o suficiente para o desenvolvimento técnico. (EVANS; FOWLER, 2004, p. 21)

Evans e Fowler (2004, p. 21) aplicam essa afirmação como o propósito de entendimento de domínios em desenvolvimento de software. No entanto, vale observar a aplicabilidade desse conceito de modelo no domínio da arquitetura de software, onde esses termos e inter-relações entram na abstração e comunicação dos envolvidos durante as atividades da arquitetura, entendendo a "arquitetura como metamodelo", Holt (2002) alega que "[...] a arquitetura de software é mais utilmente pensado como um modelo mental compartilhado entre as pessoas responsáveis pelo software".

Ao atingir uma maturidade nas decisões e na modelagem do projeto de arquitetura, o arquiteto deverá comunicar suas decisões e projeto dos componentes de software aos envolvidos. No entanto, novos esforços serão necessários, Bass *et al.* (2012, p. 327) afirmam que criar uma arquitetura não é suficiente. Ela deve ser comunicada para permitir que seus stakeholders o usem adequadamente para fazer seu trabalho. A comunicação entra em cena novamente, para que outras atividades de arquitetura de software sejam executadas, como avaliação ou análise da arquitetura, acompanhamento do desenvolvimento, teste e implantação pelo arquiteto, alocação de recursos e tarefas, sobre esta comunicação nas atividades de projeto, Bass *et al.* (2012, p. 47) explicam que:

Para que a arquitetura seja eficaz como a espinha dorsal do design do projeto, ela deve ser comunicada de forma clara e inequívoca a todas as partes interessadas. [...] A documentação arquitetural também deve ser mínima e direcionada aos stakeholders que irão utilizá-la; (BASS *et al.*, 2012, p. 47)

As expressões da arquitetura do sistema são definidas na literatura como descrições de literatura. A ISO 42010:2011 (ISO/IEC/IEEE, 2011) explica a relação entre as representações de arquitetura e a comunicação entre os envolvidos:

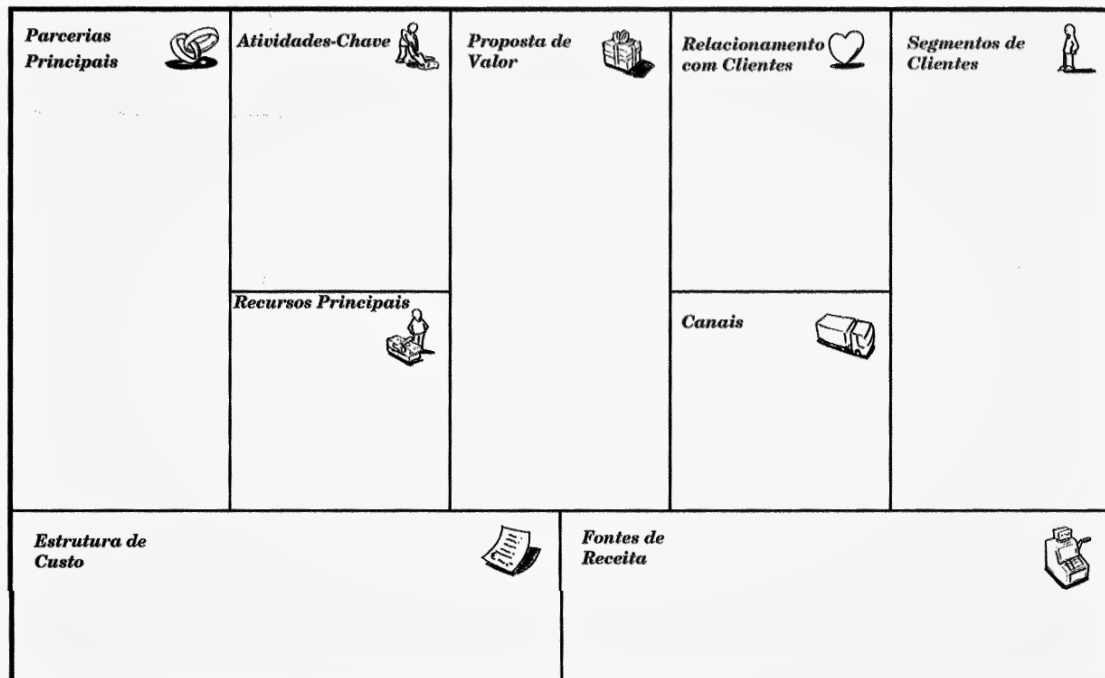
As descrições da arquitetura são usadas pelas partes que criam, utilizam e gerenciam sistemas modernos para melhorar a comunicação e a cooperação, permitindo que trabalhem de maneira integrada e coerente. ISO/IEC/IEEE (2011, p. V)

Para além de desenhar caixas e setas, as expressões da arquitetura devem relacionar estas representações (normalmente visuais) às preocupações e seus stakeholders. Reekie *et al.* (2006, p. 142) afirmam que “há um forte consenso na indústria de software de que é uma boa prática descrever a arquitetura de um sistema em termos de um conjunto bem definido de visões que abordam as preocupações de um determinado grupo de partes interessadas”.

2.3 Business Model Canvas e Business Model Ontology

O *Business Model Canvas* (BMC) (Figura 1) é uma ferramenta visual para modelagem e prototipação de modelos de negócios. O conceito inicial foi desenvolvido e apresentado por Osterwalder (2004) em sua tese, e ao longo dos anos a ferramenta foi desenvolvida e aperfeiçoada resultando na sua publicação (OSTERWALDER; PIGNEUR, 2011).

Figura 1 – Business Model Canvas.



Fonte: Osterwalder e Pigneur (2011, p. 44)

Essa ferramenta consiste em um quadro/tela podendo ser implementado de forma física em papel, ou digital em software, dividido em nove blocos. Cada bloco corresponde elemento presente em um modelo de negócios como: Proposta de valor, Relacionamento com

cliente, Segmentos de mercado, Canais de distribuição, Atividades chave, Parceiros chave, Recursos chave, Custos e Fluxo de rendimentos.

A caracterização visual da ferramenta permite simplificar o entendimento dos elementos de modelos de negócios. No livro *Business Model Generation*, Osterwalder e Pigneur (2011) explicam essa abordagem visual na percepção dos conceitos dos elementos de domínio.

Desenhando todos os elementos do Quadro, você imediatamente dá aos observadores uma visão do todo de um Modelo de Negócios. Um desenho fornece a quantidade certa de informações para permitir ao observador capturar a ideia, sem detalhes demais para distraí-lo. (OSTERWALDER; PIGNEUR, 2011, p. 152)

Como afirmado na introdução deste trabalho, "[...] o desenvolvimento de um modelo de negócio pode atuar de forma semelhante a AS, com o envolvimento e a participação de diversos *stakeholders*". Relacionado a essa afirmação, Osterwalder e Pigneur (2011) sustentam um entendimento compartilhado entre as partes na composição desse modelo e a abordagem visual como motor desse entendimento, onde:

Compreender um Modelo de Negócio exige não apenas o conhecimento dos elementos que o compõem, mas também captar as interdependências entre elementos. Isso é mais fácil de expressar visualmente do que com palavras. [...] Todos em uma organização precisam entender seu modelo, pois todos podem contribuir potencialmente para o seu aprimoramento. No mínimo, os funcionários precisam de um entendimento compartilhado, de modo que possam se mover na mesma direção estratégica. A descrição visual é a melhor forma de criar tal compreensão compartilhada. [...] Visualizar Modelos de Negócios como uma unidade é a maneira mais eficiente de obter uma compreensão compartilhada. Pessoas de diferentes partes de uma organização podem compreender profundamente partes de um Modelo de Negócios, mas carecem de um entendimento definido do todo. Quando os especialistas desenham em conjunto um Modelo de Negócios, todos os envolvidos passam a entender cada componente individual, e desenvolvem uma compreensão compartilhada das relações entre eles. (OSTERWALDER; PIGNEUR, 2011, p. 152, 155, 153)

A estrutura do quadro, sua divisão e organização dos componentes e das informações inseridas pelos utilizadores transmitem os conceitos do domínio de modelo de negócios. O impacto dessa estrutura sobre a comunicação destes utilizadores para um entendimento coletivo sobre o modelo é a base da ferramenta para percepção dos conceitos dos elementos de domínio. Sobre isso Osterwalder e Pigneur (2011) afirmam que:

O Quadro é uma linguagem visual compartilhada. Ele fornece não somente um ponto de referência, mas também um vocabulário e uma gramática que ajudam pessoas a se compreender melhor. Ele se torna um poderoso catalisador de discussões concentradas sobre os elementos do Modelo de Negócio e a forma como se encaixam. [...] O pôster do Quadro de Modelo de Negócios

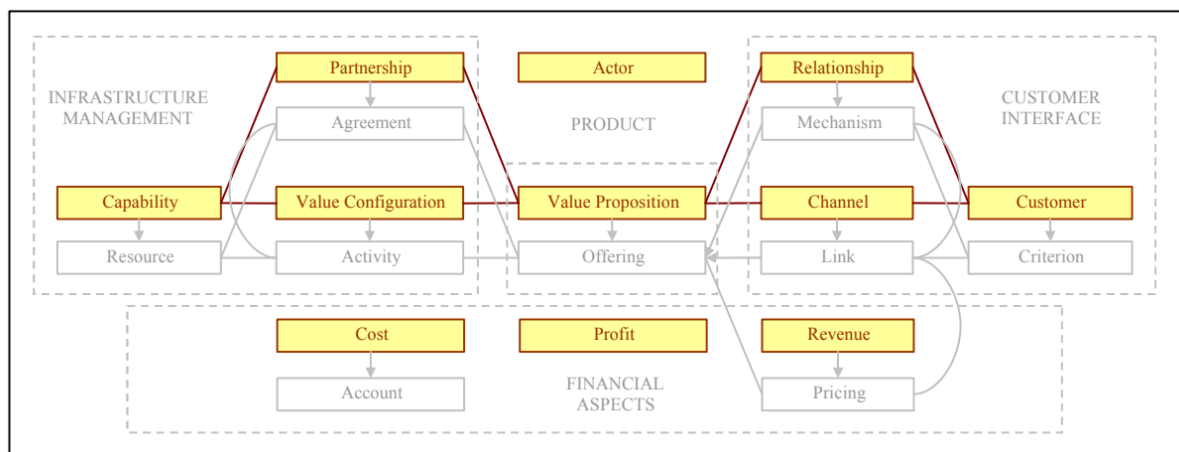
é um mapa conceitual, que funciona como linguagem visual, mas com uma gramática correspondente. Ela diz quais peças de informação inserir no modelo e onde. Fornece um guia visual e textual para toda a informação necessária para desenhar um Modelo de Negócio. (OSTERWALDER; PIGNEUR, 2011, p. 153, 152)

Observa-se ao longo desse tópico a participação de alguns efeitos na percepção da informação, comunicação e integração entre os utilizadores como: entendimento compartilhado, linguagem visual compartilhada, vocabulário e gramática. A respeito desses elementos, Belloni (2016) analisa o BMC e reforça que:

As ferramentas estratégicas, além do seu papel racional e analítico, trazem estímulo para discussão e geração de ideias, tornando a formulação das estratégias um processo interativo e cognitivo, facilitando a comunicação entre os stakeholders do processo. Facilitam também o direcionamento da atenção a conceitos específicos, priorizando a análise, além de trazer uma linguagem comum para conversas estratégicas, melhorando assim os processos organizacionais e possibilitando o compartilhamento de conhecimento e integração frente às diversas áreas das quais um negócio é formado. (BELLONI, 2016, p. 37)

A fundamentação desses elementos do domínio de negócios presentes no BMC foram derivados de uma ontologia, o *Business Model Ontology* (BMO) (OSTERWALDER, 2004) (Figura 2). Esta ontologia de domínio buscava identificar elementos e suas definições em comum nas várias obras da literatura relacionados a modelos de negócios. Na época, havia vários trabalhos recentes relacionados ao assunto, porém até mesmo o termo *business model* não possuía uma definição convencional definitiva. Após uma intensa análise da literatura da época foi concebido um modelo de referência sobre modelos de negócios.

Figura 2 – Business Model Ontology.



Fonte: Osterwalder (2004, p. 44)

Osterwalder (2004) já considerava que o modelo visual representando o modelo identificado na ontologia proposta possibilitaria melhorias nas percepções e comunicação de seus utilizadores antes do estabelecimento desse modelo. Conforme o referido autor:

Uma ferramenta de visualização baseada em ontologia de modelo de negócios pode ajudar os profissionais de negócios a entender mais rapidamente um modelo de negócios e os relacionamentos por trás de seus elementos. [...] Uma ferramenta de visualização baseada em ontologia de modelo de negócios pode ajudar os profissionais de negócios a entender mais rapidamente um modelo de negócios e os relacionamentos por trás de seus elementos. [...] Uma ferramenta baseada em ontologia de modelo de negócios cria uma linguagem comum para abordar questões de modelo de negócios e, nesse sentido, melhora a comunicação entre os profissionais de negócios. (OSTERWALDER, 2004, p. 141)

Neste trabalho, busca-se obter efeitos e impactos semelhantes ao BMC, abordando o domínio da AS. Para isso, o desenvolvimento deste trabalho considera utilizar uma compilação de informações do domínio de AS por meio de uma revisão da literatura relacionada e representá-la em um mapa conceitual, provendo uma fundamentação nos elementos a serem representados visualmente na ferramenta aqui desenvolvida.

3 TRABALHOS RELACIONADOS

3.1 The GRADE Decision Canvas for Classification and Reflection on Architecture Decisions

Papatheocharous *et al.* (2017): Esse trabalho também propõe uma ferramenta visual para arquitetura de software, onde os utilizadores possam documentar, discutir e comunicar decisões arquiteturais. A ferramenta possui uma estrutura semelhante ao BMC. No entanto (dividido em duas partes) conta com um diagrama (circular e segmentado, localizado ao lado esquerdo) da taxonomia GRADE (*Goals, Roles, Assets, Decision, Environment*) (PAPATHEOCHAROUS *et al.*, 2015) representando, categorizando e exemplificando diversos elementos da arquitetura de software. Os itens da taxonomia são selecionados em cada um dos aspectos e desenvolvidos e representados nos campos ao lado direito, com o uso de questões representadas em cada campo.

O trabalho se assemelha a este por estabelecer uma ferramenta visual relacionada à arquitetura de software e sistemas, auxiliando no armazenamento de informações e comunicação com os envolvidos. E se diferencia por buscar uma perspectiva de análise-projeto, ao invés de decisões, onde seja possível simplificar a captação e documentação de informações de modo mais abrangente. Também se diferencia na abordagem de construção, onde se considera um modelo de domínio, enquanto o outro considera a taxonomia GRADE para estruturar a ferramenta.

3.2 CacOphoNy: Metamodel-Driven Software Architecture Reconstruction

Favre (2004): Esse trabalho desenvolve uma metodologia e um processo genérico para reconstrução de arquitetura de software, engenharia reversa de arquitetura de software pelo de código-fonte a partir de um "megamodelo" de arquitetura de software com base a IEEE 1471, que originou a ISO 42010:2011 (ISO/IEC/IEEE, 2011). Também envolve definições de *metaware e appliware* (software em nível de modelo e metamodelo).

O trabalho se assemelha a este por considerar metamodelos relacionados à arquitetura de software e se diferencia por sua finalidade, a engenharia reversa de arquitetura de software, e também por resultar em um processo.

3.3 Strategic Model Canvas: a tool proposition to optimize strategic planning

Azevedo *et al.* (2018): este trabalho propõe uma ferramenta visual semelhante ao BMC, utilizando conceitos de *design thinking*, aplicada ao planejamento estratégico. O trabalho se assemelha a este por definir uma ferramenta, que busca simplificar a visualização de conceitos em um domínio específico, também tendo como base o BMC, pelo uso de abordagens científicas. Se diferencia por aplicar-se ao domínio do planejamento estratégico de organizações e por utilizar os conceitos de *design thinking*.

Um comparativo entre os trabalhos apresentados e o trabalho desenvolvido nesta pesquisa está retratado no Quadro 1.

Quadro 1 – Tabela comparativa de Trabalhos Relacionados

	Papatheocharous <i>et al.</i> (2017)	Azevedo <i>et al.</i> (2018)	Favre (2004)	Este trabalho
Considera o <i>Business Model Canvas</i> ?	Não	Sim	Não	Sim
Cria ferramenta visual?	Sim	Sim	Não	Sim
Considera metamodelos em Arquitetura de Software?	Sim	Não	Não	Sim
Cria processo em Arquitetura de Software?	Não	Não	Sim	Não

Fonte: elaborado pelo autor.

4 PROCEDIMENTOS METODOLÓGICOS

Nesta seção estão descritas as fases necessárias para o desenvolvimento e aplicação do trabalho. A natureza deste trabalho caracteriza-se essencialmente como uma pesquisa aplicada, visando gerar conhecimentos de aplicação prática, para solucionar problemas específicos como sugerido em Ganga (2012).

Em relação aos objetivos, este trabalho conta com objetivos exploratórios. De acordo com Gil (2000, p.27), as pesquisas exploratórias têm como propósito proporcionar maior familiaridade com o problema, com vistas a torná-lo mais explícito ou a construir hipóteses.

Quanto à abordagem, este trabalho aborda os resultados de forma quali-quantitativa, com predominância sobre a abordagem qualitativa, onde Ganga (2012, p. 210) define que: "[...] o papel do pesquisador em uma pesquisa qualitativa é obter informações do fenômeno segundo a visão dos indivíduos, bem como observar e coletar evidências que possibilitem interpretar o ambiente em que a problemática ocorre". Dados quantitativos foram interpretados para corroborar as informações qualitativas, Merriam (2009, p. 14) explica que seu objetivo é alcançar uma compreensão de como as pessoas dão sentido às suas vidas; delinear o processo, da criação de significado e descrever como as pessoas interpretam o que eles vivenciam.

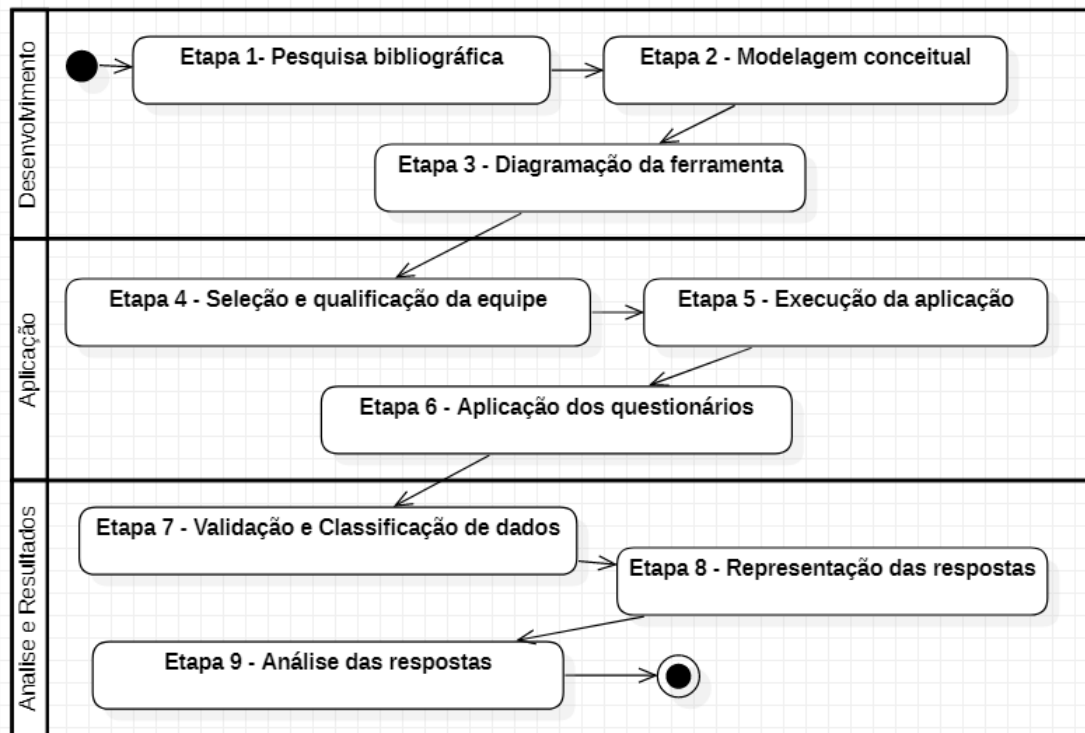
Em relação aos procedimentos, este trabalho executou um estudo de caso único com aplicação de questionários e observação direta, Yin (2015, p. 32) define um estudo de caso como “[...] uma pesquisa empírica que investiga um fenômeno contemporâneo dentro de seu contexto de vida real, especialmente quando os limites entre o fenômeno e o contexto não estão claramente definidos”.

Na Figura 3 apresenta-se o fluxograma indicando os procedimentos metodológicos empregados nesta pesquisa, que foram divididos em três fases: Desenvolvimento, Aplicação e Análise dos Resultados.

Este trabalho inicia-se com a condução de uma pesquisa bibliográfica (etapa 1). Gil (2008, p. 50) explica que essa é desenvolvida a partir de material já elaborado, constituído principalmente de livros e artigos científicos. Nesta pesquisa, essa etapa buscou captar e compilar o conhecimento estabelecido sobre itens de análise e projeto em arquitetura de software envolvidos na comunicação dos *stakeholders* para uma modelagem conceitual do domínio de arquitetura de software, sendo guiada pelos seguintes questionamentos:

1. Quais elementos são considerados nas atividades de análise e o projeto de AS?
2. Quais desses elementos fazem parte da comunicação entre arquiteto de software e sta-

Figura 3 – Fluxograma dos procedimentos metodológicos.



Fonte: elaborado pelo autor.

keholders?

As informações identificadas nessa etapa devem ser documentadas organizando-se em: definições, relações, propriedades, tipos, categorias e exemplos.

Na conclusão da pesquisa bibliográfica, o conhecimento documentado é modelado e representado em um mapa conceitual (etapa 2) (NOVAK, 1977), para caracterizar e hierarquizar os elementos e suas relações. A partir disso, deve-se realizar uma diagramação simples da ferramenta em formato semelhante ao BMC (etapa 3).

Para a execução das etapas de aplicação, análise e resultados, deve-se considerar que esta pesquisa avaliará a ferramenta desenvolvida de acordo com os seguintes critérios e objetivos:

1. A ferramenta representou um metamodelo de AS dos elementos de interesse dos diferentes stakeholders na AS de forma satisfatória? - Esse critério visa entender a abrangência do modelo conceitual presente na ferramenta e manifestações de uma abstração comum por parte dos participantes, correspondente ao metamodelo de arquitetura explicado na fundamentação teórica.
2. A ferramenta foi efetiva em estabelecer uma linguagem comum entre os diferentes stakeholders? - Esse critério busca identificar a existência de uma linguagem comum entre os

participantes nos termos dos elementos do metamodelo.

3. A ferramenta possibilitou uma comunicação satisfatória (para a troca de ideias e informações, discussões, negociações e outros) entre os envolvidos durante as atividades de arquitetura de software? - Esse aspecto busca verificar não apenas as percepções sobre a comunicação individual, mas também da comunicação coletiva, caso algum participante eventualmente não tenha interagido de forma constante, poderá avaliar a comunicação entre os participantes.
4. A ferramenta auxiliou o arquiteto e SH a alcançarem os objetivos das atividades de análise de AS (identificar e representar os requisitos arquiteturalmente significativos) de maneira satisfatória? - Esse aspecto busca compreender a participação da ferramenta na identificação de Requisitos Arquiteturalmente Significativos, identificados aqui como Requisitos Funcionais, Restrições e Atributos de Qualidade.
5. A ferramenta auxiliou o arquiteto e SH a alcançar os objetivos das atividades de AS (representar e comunicar o projeto de arquitetura proposto) de maneira satisfatória? - Esse aspecto busca compreender a participação da ferramenta na representação e comunicação do projeto de arquitetura de software e decisões do arquiteto.

Ao início da fase de aplicação da ferramenta, as atividades de seleção e qualificação da equipe (etapa 4) devem ser executadas:

- Identificação de papéis da equipe: arquiteto de software, gerente de projeto, líderes técnicos, desenvolvedores ou outros *stakeholders*.
- Apresentação da ferramenta para o arquiteto: explicação geral sobre o conceito e funcionamento da ferramenta.
- Apresentação de proposta de exemplo para os participantes para uma análise preliminar, definindo a temática deste estudo de caso.

Definida a equipe, inicia-se o estudo de caso com observação direta (etapa 5), cujo objetivo é entender o uso da ferramenta Software Architecture Canvas em uma equipe de desenvolvimento multidisciplinar, contando com participantes atuantes do mercado de desenvolvimento de software tanto das áreas técnicas, quanto de negócios. Neste estudo, os participantes deverão utilizar a ferramenta para a elaboração da arquitetura de um sistema fictício, em atividades de análise e projeto de arquitetura de software, porém sem a necessidade de desenvolver (codificar) o sistema proposto, servindo apenas com um 'ensaio'. Durante o estudo, os comportamentos relacionados à comunicação e presença do metamodelo nas discussões entre os participantes,

decisões e descrição da arquitetura devem ser observados. Nos momentos finais dessa atividade, um conjunto de questionários foi repassado aos participantes (etapa 6) (disponível no Apêndice E), na seguinte forma, a todos os participantes foram entregues dois questionários:

- Um questionário demográfico-profissional: identificando o perfil dos participantes, como: escolaridade, experiência e cargo.
- Um questionário opcional com questões abertas: que obteve impressões gerais dos participantes como aspectos positivos e negativos e sugestões de melhoria.

Ao arquiteto foi entregue um questionário aberto, que buscou identificar as informações no âmbito técnico da arquitetura. Aos demais participantes, foram entregues outros dois questionários:

- Um questionário com questões fechadas: obtendo impressões relacionadas a ferramenta, classificando as respostas na escala psicométrica de Likert (1932) em: "Discordo Totalmente"(1), "Discordo"(2), "Discordo Parcialmente"(3), "Indiferente"(4), "Concordo Parcialmente"(5), "Concordo"(6), "Concordo Totalmente"(7).
- Um questionário com questões abertas: que obteve informações subjetivas do uso da ferramenta.

As respostas obtidas foram representadas (etapa 8) da seguinte forma: Questões Demográficas foram expostas em um quadro ou gráficos segmentados. Questões fechadas passaram por uma validação (etapa 7) (estimativa de confiabilidade das respostas e consistência interna do questionário) utilizando o coeficiente alfa de Cronbach (CRONBACH, 1951) e representadas por um gráfico de barras empilhadas. Questões abertas foram selecionadas e classificadas pelo papel do participante no estudo de caso.

Por fim, as informações obtidas nas etapas 7 e 8 foram compiladas, analisadas e discutidas (etapa 9) com base nos critérios/objetivos expostos anteriormente, estruturando-se em: Objetivo: Apresentação do objetivo relacionado ao critério selecionado - Análise: Discussões e considerações sobre as informações selecionadas - Conclusão: estabelecendo uma síntese das considerações obtidas na análise.

5 DESENVOLVIMENTO E APLICAÇÃO

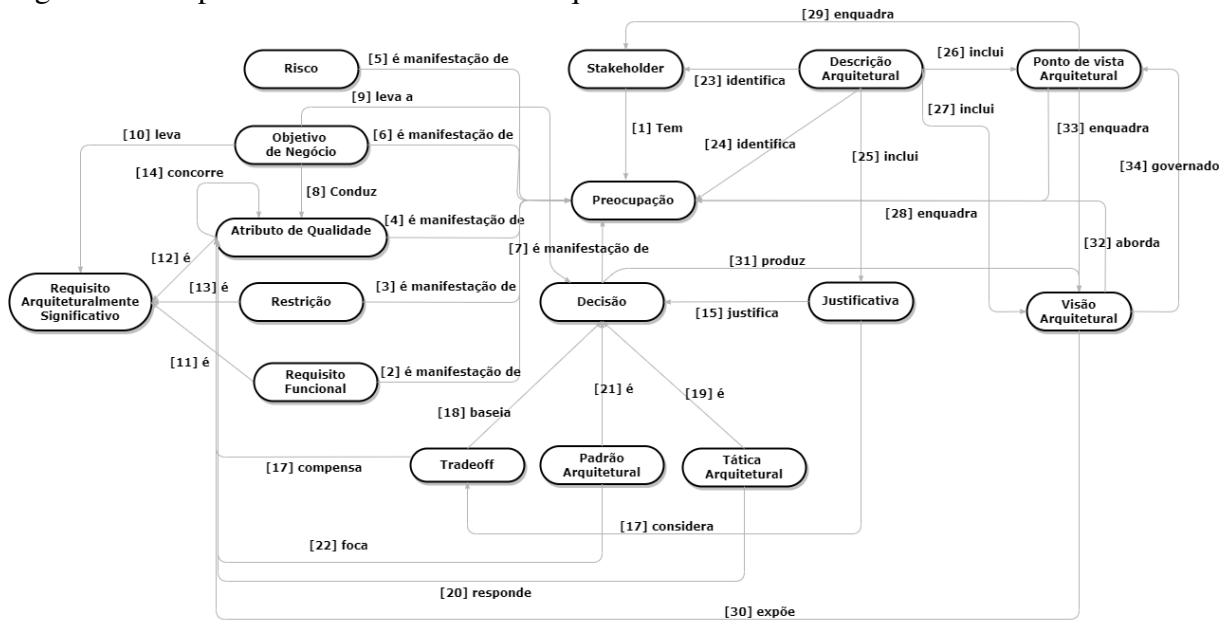
Na etapa 1 foram utilizados livros, normas internacionais, artigos e corpos de conhecimento. As obras nas quais foram encontrados um maior respaldo para capturar informações foram:

- ISO 42010:2011 - ISO/IEC/IEEE (2011): Norma internacional para descrição de arquitetura, contém uma metamodelo de descrição arquitetural, fornecendo as definições iniciais para esta ontologia, como: *Stakeholders*, Preocupações, Decisões, Justificativas, *Viewpoints* e Visões.
- *Software Architecture in Practice* - 3ª edição - Bass *et al.* (2012): Literatura base sobre AS, forneceu diversas informações sobre AS, como: *Stakeholders*, Preocupações, Objetivos de negócios, Requisitos Arquiteturalmente Significativos, Atributos de Qualidade, Táticas e Padrões Arquiteturais.
- *Documenting Software Architectures* - Clements *et al.* (2010): Foi utilizado para obter mais definições e informações sobre soluções arquiteturais e representação de views.
- *Essential Software Architecture* - Gorton (2011): Foi utilizado para adicionar mais definições sobre: Requisitos Arquiteturalmente Significativos, Restrições e *Tradeoffs*.
- SWEBoK - Corpo de conhecimento de Engenharia de Software - Bourque *et al.* (2014): Proveu informações sobre requisitos funcionais, decisões, *tradeoffs* e riscos.
- *Software Architecture as a Set of Architectural Design Decisions* - Jansen e Bosch (2005): Forneceu definições e relações sobre decisões arquiteturais.
- Introdução ao RUP *Rational Unified Process* - Segunda Edição - Kruchten (2003): O Processo Unificado foi selecionado por ser "centrado em arquitetura", e fornece definições sobre *Stakeholders*, Visões e Estilos e Padrões arquiteturais.
- BABoK - *Business Analysis Body of Knowledge* - IIBA (2015): Foi utilizado para obter mais informações sobre termos de negócios e outros termos semelhantes aplicados a AS, como: Objetivos de negócios, Restrições e *Tradeoffs*.
- NBR/ISO 9126-1 - Qualidade de produto, Parte 1: Modelo de qualidade - ABNT/ISO/IEC (2003): norma nacional/internacional para qualidade de software, sendo utilizada para identificar definições, propriedades e exemplos de atributos de qualidade.

As informações obtidas nesta pesquisa estão documentadas no apêndice A. Na Figura 4 apresenta-se o modelo conceitual desenvolvido na etapa 2.

As elipses presentes nesta figura representam os elementos/definições presentes na

Figura 4 – Mapa/Modelo conceitual em Arquitetura de Software.



Fonte: elaborado pelo autor.

Nota: uma imagem com maiores detalhes está disponível no Apêndice B.

pesquisa bibliográfica, as setas representam as relações identificadas entre estes elementos. Vale observar a numeração escrita em cada relação, que corresponde aos trechos onde essas relações foram identificadas das obras desta pesquisa.

Os elementos e relacionamentos levantados na revisão bibliográfica/modelo conceitual foram utilizados na etapa 3 para a criação dos elementos visuais base do *Canvas* de Arquitetura de Software, sendo esses:

- **Colunas:** Os blocos que constituem a ferramenta foram organizados em 4 colunas, “Business Case”, “Requisitos Arquiteturalmente Significativos”, “Decisões”, “Descrição”. Essas colunas possibilitam o arquiteto a focar nos blocos correspondentes de forma linear e conforme as atividades de AS ao longo da utilização da ferramenta. Essas colunas têm origem na identificação dos elementos e atividades descritas na fundamentação teórica deste trabalho.
- **Blocos:** De forma semelhante ao BMC, o Canvas de Arquitetura de Software também foi dividido em blocos. Cada um desses tem o objetivo de representar os elementos presentes na análise e projeto de arquitetura de software, e correspondem aos elementos identificados na revisão bibliográfica desenvolvida anteriormente. Alguns exemplos identificados nessa revisão foram expostos como sugestões para guiar os usuários no uso da ferramenta.
- **Perguntas-gatilho:** Cada bloco no Canvas pode conter um conjunto de "perguntas-gatilho", para orientar o arquiteto a obter informações mais específicas, ou apenas guiar

o arquiteto para levantar dados sobre o campo a ser preenchido. Essas perguntas foram elaboradas a partir de alguns relacionamentos identificados na pesquisa bibliográfica.

Após uma diagramação do protótipo da ferramenta, um teste piloto, não relacionado com o estudo de caso ou a etapa de aplicação do trabalho, foi executado com desenvolvedores e alunos dos cursos de engenharia de software, engenharia de computação e sistemas de informação, em um projeto de software fictício escolhido no momento da execução desse. As informações e observações desse teste além de outras decisões ao longo das atividades deste trabalho guiaram algumas alterações no experimento e refinamentos no formato final na ferramenta, sendo estes:

- Criação de uma carta-proposta fictícia para propor a temática do estudo de caso.
- Junção entre stakeholders e preocupações: Para que a identificação destes seja feita em um mesmo momento.
- Representação dos Requisitos Arquiteturalmente Significativos divididos: Atributos de qualidade, Requisitos Funcionais, Riscos e Restrições: Separando estas preocupações/requisitos para que sejam discutidas individualmente.
- Representação das decisões divididas em: *Tradeoffs*, Táticas e Padrões. Esses herdam as mesmas propriedades de relações do item Decisão; bem como a supressão do item justificativa.
- Supressão dos itens "Viewpoints" e "Descrição Arquitetural", representados apenas pelo campo "Visões e Representações Arquiteturais".

Com a conclusão dos refinamentos, a ferramenta foi diagramada no seu formato final, apresentada no Apêndice B. O modelo conceitual e o *Canvas* de Arquitetura de Software foram diagramadas na ferramenta Diagrams.net, um modelo em branco do *Canvas* está disponível no link do rodapé ¹.

As etapas 4, 5 e 6, ocorreram nos dias 1 e 2 de fevereiro de 2022, entre as 18:00 e 20:00, na plataforma de chamadas de vídeo *Google Meet*, com a participação de seis profissionais da área de desenvolvimento de software. O perfil desses profissionais está informado no Capítulo 7, os quais atuam ou já atuaram em algum momento da carreira nos papéis designados neste estudo de caso. No primeiro dia participaram do estudo de caso, o Arquiteto de Software, o Analista de Negócios, o Gerente de Projetos e o Líder de QA. O treinamento do arquiteto expõe informações gerais sobre o funcionamento da ferramenta e seu uso como descrito no Capítulo 6. Antes da aplicação, uma proposta fictícia do projeto de software (disponível no Apêndice D)

¹ <https://bit.ly/379br5d>

foi apresentada à equipe. Para que as atividades fossem iniciadas, o canvas foi repassado para edição usando a ferramenta Diagrams.net, tornando-se o mediador. No segundo dia participaram, o Arquiteto de Software, o Analista de Negócios, o Líder de QA, a Analista de Integrações e o Desenvolvedor. Ao fim do estudo de caso os questionários foram repassados aos participantes, resultando nas respostas desses e o Canvas de arquitetura de software preenchido, disponível no link do rodapé ² .

² <https://bit.ly/3PBrOJe>

6 SOFTWARE ARCHITECTURE CANVAS

O *Software Architecture Canvas* (Figura 5) é uma ferramenta visual que atua no auxílio da comunicação entre arquiteto e outros SH durante as atividades de arquitetura de software. O arquiteto poderá utilizá-lo em reuniões individuais com um SH específico, ou reuniões coletivas com outros participantes e times.

Figura 5 – Software Architecture Canvas.

Business Case	Requisitos Arquiteturalmente Significativos	Decisões	Descrição
Stakeholders e Preocupações Quaisquer pessoas, equipes ou organizações interessadas ou envolvidas no sistema, bem como seus interesses no sistema. Quais são os stakeholders do projeto? Qual papel estes stakeholders têm no projeto? Quais são as preocupações destes stakeholders? Quais são as influências destas preocupações? Ex. de Stakeholders: Gerentes, Analistas, Clientes, Programadores, Engenheiros, Provedores de Serviços, Usuários, Acadêmicos etc. Ex. Influências: desenvolvimento, tecnologia, empresarial, operacional, organizacional, político, econômico, jurídico, regulatório, ambiental, social. Ex. de preocupações: pressão de sistema, integração de arquitetura para atingir os objetivos de sistema, capacidade de integração e separação de sistema, risco e impacto do sistema para as partes interessadas, funcionalidade, escalabilidade, custo, impacto no sistema, impacto na operação, desempenho, segurança, integridade, disponibilidade, elasticidade, desempenho, utilização de recursos, confiabilidade, segurança, garantia de conformidade, interoperabilidade, abertura, interoperabilidade, autonomia, custo, desempenho, qualidade de serviço, flexibilidade, agilidade, escalabilidade, sustentabilidade, conformidade, conformidade com processos, mudança de estado, integração de arquitetura, sustentabilidade de custos, escalabilidade, sustentabilidade com a regulamentação, garantia, disponibilidade e estratégias de negócios, sustentabilidade de custos, manutenção, sustentabilidade e sustentabilidade.	Restrições Quaisquer impedimentos aplicáveis ao projeto, nas áreas técnicas ou de negócios. Quais são as restrições técnicas? Quais são as restrições de negócios? Esta restrição foi informada por qual(is) stakeholder(s)? Quais objetivos de negócios estão ligados a esta restrição? Tipos de restrições: Operacionais, Tempo, Tecnologia, Infraestrutura, Políticas e Regulamentos, regulamentares, Conformidade com Regulamentos, Orçamentos, restrições a etc.	Tradeoffs Quaisquer compensações entre conflitos (de requisitos) encontrados. Quais são elementos atributos de qualidade, restrições, riscos, em conflito sendo considerado? Quais são as justificativas relacionadas a estas compensações? Quais são as justificativas relacionadas a este tradeoff? Quais regras este tradeoff define ao projeto? Quais restrições são tradeoffs ao projeto? Quais novos requisitos são tradeoffs ao projeto?	Pontos de Vista e Visões Arquiteturais Quaisquer descrições e expressões das perspectivas da arquitetura dos interessados do sistema. A qual stakeholder cada ponto de vista está endereçado? Quais preocupações este ponto de vista está endereçado? Quais atributos de qualidade são visto neste ponto? Quais decisões produzem esta visão? Quais preocupações esta visão aborda? Quais pontos de vista governam esta visão? Qual o nível de formalidade desta visão? Exemplo (NewPMA): (Novos H+1) Loja, Implementação, Process, Implementação, Caixa de Voto (Novos em Recurso) Módulo, Componente & Conexão, Associação.
Objetivos de Negócios Quaisquer fins que a organização pretende atingir, representados em objetivos descritos. Exemplo de Objetivo: Como o recurso, como um novo produto ou serviço, resolve uma desvantagem competitiva ou cria uma nova vantagem competitiva, dentro e fora da organização de acordo com requisitos comerciais. Aumentar a satisfação do cliente. Aumentar a satisfação dos funcionários. Cumprir os requisitos regulamentares. Melhorar a segurança. Melhorar o tempo de entrega de um produto ou serviço.	Requisitos Funcionais Quaisquer funcionalidades ou comportamentos do sistema. Quais objetivos de negócios estão ligados a este requisito? Este requisito foi informado por qual(is) stakeholder(s)? Representações de requisitos funcionais: Requisitos técnicos, Módulos, Sistemas existentes, Classes de uso, "telas" de usuário e etc.	Táticas Arquiteturais Quaisquer decisões arquiteturais que respondam a um atributo de qualidade em específico. Qual atributo de qualidade será atendido com esta tática? Quais são as justificativas relacionadas a esta(s) decisão(s)? Quais são as justificativas relacionadas a esta(s) tática(s)? Quais regras esta tática define ao projeto? Quais restrições esta tática impõe ao projeto? Quais novos requisitos esta tática resulta ao projeto?	
Riscos Quaisquer eventos ou condições incertas que caso ocorram impactará negativamente os objetivos do projeto. Quais stakeholders apontaram estes riscos? Quais são os riscos internos? Quais são os riscos externos? Qual a magnitude deste risco? Qual a probabilidade de ocorrência desse risco? Qual a severidade deste risco? Quais as ações de ação aplicadas a este risco? Tipos de Risco: Cliente (sem alta grau de certeza), fornecedor (sem apoio ou recursos contidos), Regulação de risco, alta, Siquem, tecnologia, sustentabilidade e risco. Alguns exemplos: falhar na implementação de risco. O custo do risco não é menor do que o valor de custo (benefício) e não há um plano de mitigação de risco. Definir um plano de mitigação, monitorar o risco.	Atributos de Qualidade Capacidade do software em satisfazer as necessidades dos stakeholders em determinados aspectos. Quais são os atributos de qualidade internos? Quais são os atributos de qualidade externos? Quais são os atributos de qualidade em uso? Quais objetivos de negócios estão ligados a este atributo de qualidade? Este atributo de qualidade foi informado por qual(is) stakeholder(s)? Exemplos de Atributos: Funcionalidade, Confiabilidade, Usabilidade, Otimismo, Visibilidade, Interoperabilidade, Interatividade, Transparência, Interoperabilidade, Flexibilidade, "telas" e Painel. Representações de Atributos de Qualidade: Diagramas, Matrizes.	Padrões Arquiteturais Quaisquer estruturas arquiteturais (conjuntos de elementos arquiteturais) para soluções de problemas em um contexto específico. Quais decisões fazem parte deste padrão selecionado? Quais justificativas para o uso deste padrão? Quais atributos de qualidade este padrão foca? Em qual contexto este padrão é aplicado? Em quais problemas este padrão se aplica? Qual atributo de qualidade este padrão atende? Quais elementos (responsáveis, produtores ou objetos) fazem parte deste padrão? Quais restrições este padrão impõe? Exemplos (Modelos): Canais, Injeção, Modelo View Controller, Page-And-File, Client-Side, Peer-to-Peer, Service-Oriented Architecture, Publish-Subscribe, Shared-Cache, Map-Reduce, NoSQL etc.	

Fonte: elaborado pelo autor.
Nota: uma imagem com maiores detalhes está disponível no Apêndice F.

A composição visual da ferramenta busca simplificar conceitos para SH diferentes, tanto da área técnica quanto da área de negócios, provendo um modelo que identifica os elementos áreas de trabalho desses ligados à arquitetura de software.

O resultado dessa ferramenta não é uma descrição arquitetural, mas um conjunto de informações sobre a arquitetura do sistema discutidas entre arquiteto e SH, sendo uma ferramenta para uso em processo (de análise e projeto) e não um artefato. Esse conjunto de informações pode ser utilizado posteriormente para atender os objetivos de quaisquer processos, métodos ou técnicas escolhidos pelo arquiteto.

O *Software Architecture Canvas* é dividido em 4 colunas: Business Case, Requisitos

Arquiteturalmente Significativos, Decisões e Descrição, onde agrupam 10 blocos. Cada bloco pode conter um conjunto de questões a serem respondidas pelo arquiteto orientando quais informações podem ser inseridas. Os blocos e perguntas representadas na ferramenta são:

- **Stakeholders e Preocupações:** bloco que representa quaisquer pessoas, equipes ou organizações interessadas ou envolvidas no sistema, bem como seus interesses no sistema.
 - Quais são os stakeholders do projeto? Quais papéis esses stakeholders têm no projeto? Quais são as preocupações desses *stakeholders*? Quais são as influências dessas preocupações?
- **Objetivos de Negócios:** bloco que se refere a quaisquer fins que a organização pretende atingir, representadas em objetivos descritivos.
- **Riscos:** bloco que envolve quaisquer eventos ou condições incertas que caso ocorram impactará negativamente os objetivos do projeto.
 - Quais *stakeholders* apontaram esses riscos? Quais são os riscos diretos? Quais são os riscos indiretos? Qual a magnitude desse risco? Qual a probabilidade de ocorrência desse risco? Qual a severidade desse risco? Quais as serão as ações aplicadas a esse risco?
- **Restrições:** bloco que representa quaisquer impedimentos aplicáveis ao projeto, nas áreas técnicas ou de negócios.
 - Quais são as restrições técnicas? Quais são as restrições de negócios? Essa restrição foi informada por qual(is) *stakeholder(s)*? Quais objetivos de negócios estão ligados a essa restrição?
- **Requisitos Funcionais:** bloco que se refere a quaisquer funcionalidades ou comportamentos do sistema.
 - Quais objetivos de negócios estão ligados a esse requisito? Esse requisito foi informado por qual(is) *stakeholder(s)*?
- **Atributos de Qualidade:** bloco que compreende quaisquer capacidades do software em satisfazer as necessidades dos stakeholders em determinados aspectos.
 - Quais são os atributos de qualidade internos? Quais são os atributos de qualidade externos? Quais são os atributos de qualidade em uso? Quais objetivos de negócios estão ligados a este atributo de qualidade? Esse atributo de qualidade foi informado por qual(is) *stakeholder(s)*?

- **Tradeoffs:** bloco que contempla quaisquer compensações entre conflitos (entre requisitos) encontrados.
 - Quais elementos (atributos de qualidade, restrições, riscos) em conflito serão compensados? Quais são as justificativas relacionadas a essas compensações? Quais são as justificativas relacionadas a esse *tradeoff*? Quais regras esse *tradeoff* define ao projeto? Quais restrições esse *tradeoff* impõe ao projeto? Quais novos requisitos esse tradeoff resultam ao projeto?
- **Táticas Arquiteturais:** bloco que diz respeito a quaisquer decisões arquiteturais que respondam a um atributo de qualidade em específico.
 - Qual atributo de qualidade será atendido com essa tática? Quais são as justificativas relacionadas a essa(s) decisão(ões)? Quais são as justificativas relacionadas a essa(s) tática(s)? Quais regras essa tática define ao projeto? Quais restrições essa tática impõe ao projeto? Quais novos requisitos essa tática resulta ao projeto?
- **Padrões Arquiteturais:** este bloco representa quaisquer estruturas arquiteturais (conjuntos de elementos arquiteturais) para soluções de problemas em um contexto específico.
 - Quais decisões fazem parte deste padrão selecionado? Quais justificativas para o uso desse padrão? Quais atributos de qualidade esse padrão foca? Em quais contextos esse padrão é utilizado? Em quais problemas esse padrão se encaixa? Qual solução esse padrão dá para o problema? Quais elementos (repositórios, processos ou objetos) fazem parte desse padrão? Quais restrições esse padrão impõe?
- **Pontos de vista e Visões Arquiteturais:** bloco onde se registra quaisquer descrições e expressões das perspectivas da arquitetura dos interesses do sistema.
 - A quais *stakeholders* esse ponto de vista está endereçado? Quais preocupações esse ponto de vista enquadra? Quais atributos de qualidade essa visão expõe? Quais decisões produziram essa visão? Quais preocupações essa visão aborda? Quais pontos de vista governam essa visão? Qual o nível de formalidade dessa visão?

As informações podem ser inseridas na ferramenta utilizando fichas, gráficos, ou post-its, sem nenhuma restrição ou recomendação, deixando os utilizadores livres para inserir de

acordo com a representação que for conveniente, seja em: texto, diagramas, notações matemáticas, fotos, cartões, dentre outros. Essas informações podem ser inseridas e validadas em ciclos de acordo com a necessidade do arquiteto ou da equipe.

7 ANÁLISE DOS RESULTADOS DA APLICAÇÃO

Nesta seção serão apresentados o perfil demográfico-profissional dos participantes, o relatório da observação direta e os resultados da aplicação.

7.0.1 Perfil demográfico-profissional dos participantes

A Quadro 2 representa o perfil demográfico-profissional dos participantes, identificando a Escolaridade, Projetos executados, Experiência (em desenvolvimento de software) e Papel executado no estudo de caso.

Quadro 2 – Descrição do Perfil demográfico-profissional dos participantes

Papel	Escolaridade	Projetos Executados	Exp. Profissional
Arquiteto	Sup. Completo	5 ou mais	Acima de 5 anos
Analista de Negócios	Sup. Completo	5 ou mais	Acima de 5 anos
Analista de QA	Sup. Completo	5 ou mais	Acima de 5 anos
Gerente de Projeto	Mestrado	5 ou mais	Acima de 5 anos
An. Integração	Sup. Completo	1 ou 2	3 a 5 anos
Desenvolvedor	Sup. Incompleto	1 ou 2	1 a 3 anos

Fonte: elaborado pelo autor.

Pode-se observar que a participação mais expressiva nesta pesquisa corresponde a pessoas com ensino superior completo, onde a maioria (66.67%) já participou de pelo menos 5 ou mais projetos ao longo da vida profissional, tendo acima de 5 anos de experiência (em desenvolvimento de software). Houve um perfil diversificado de papéis. Assim, a análise da ferramenta contou com diversas visões sobre essa a partir de suas áreas de domínio.

7.1 Relatório da Observação Direta

Serão apresentadas adiante as observações obtidas durante o estudo de caso. As informações contidas nesse relatório não seguem uma ordem cronológica dos fatos, estando somente organizadas em temas.

Intervenções para orientações do uso da ferramenta

No início do estudo de caso ocorreram momentos de dificuldade na utilização da ferramenta por parte de todos os participantes, necessitando intervenções de modo a dar maiores orientações sobre o uso da ferramenta, apenas com um entendimento coletivo do uso da

ferramenta houve uma fluidez no andamento das atividades. Houve orientações, ao longo das atividades, afirmando que o arquiteto pode passar para outros campos, quando considerasse necessário e refine ou reavalie as informações. Os participantes poderiam buscar informações na internet e utilizar quaisquer técnicas para representar informações, comunicando os exemplos informados na ferramenta. O arquiteto teve dúvidas sobre a diferença entre as táticas e padrões arquiteturais, que foram prontamente explicadas. Nos momentos próximos ao fim do estudo de caso, o arquiteto foi orientado a utilizar diagramas retirados da internet para poupar o tempo restante do estudo de caso.

Comunicação entre os participantes

Durante as atividades, houve uma boa comunicação com intercâmbio e discussões de informações técnicas e de mercado. Mesmo em momentos de silêncio e hesitação, houve um esforço coletivo de sustentação de debates, relendo e explorando os itens da carta, de modo a buscar e discutir novas informações. Observou-se que a ferramenta teve como papel principal definir a temática da discussão e restringir o universo semântico das discussões. As trocas de informações entre os participantes iniciaram-se sob domínio bancário e utilização de serviços bancários pelos participantes, passaram por informações relacionadas a custos de desenvolvimento e avaliação de uma reorganização do time de desenvolvimento devido às restrições orçamentárias, contemplaram discussões sobre demissão de funcionários e problemas com rescisões trabalhistas. No entanto, houve um entendimento que isso ficaria a cargo do banco. Também ocorreram considerações sobre rebranding e o papel da equipe para atingir este objetivo. Nisso os participantes ligaram a usabilidade do novo aplicativo para ser utilizada para publicidade das features. Houve dúvidas e discussões sobre como a validação de aberturas de contas será feita. Houve o intercâmbio de especificidades de tecnologias, como o throughput do RabbitMQ, possíveis reduções de custo com Krakend, bem como troca de experiência sobre o uso dessas e outras tecnologias.

Requisitos Arquiteturalmente Significativos

Os participantes empregaram a maior parte do tempo de estudo de caso discutindo sobre funcionalidades e processos, dado que a carta-proposta não contava com detalhes sobre os requisitos, particularidades e contextos organizacionais. Estes esforços buscaram definir uma base preliminar de informações. Os participantes identificaram a necessidade do “cliente” no

teste, pois, segundo o analista de negócios a carta ficou vaga.

Em determinados momentos as perguntas contidas na ferramenta sobre objetivos de negócio relacionando restrições fizeram o arquiteto reavaliar a forma com que os objetivos de negócios foram especificados. Isso também ocorreu com elementos de Atributos de Qualidade e em outro momento com Requisitos Funcionais. O Líder de QA e o Arquiteto de Software tiveram dificuldades para expressar atributos de qualidade de forma detalhada, bem como entender quais atributos se encaixavam em “externos” e “em uso”. Houve uma tentativa de passar para os campos de decisão (Táticas e Padrões Arquiteturais). No entanto os Atributos de Qualidade na ferramenta formaram um “elemento de barreira” entre as atividades de análise e de projeto. Sendo assim, as decisões sobre a arquitetura só conseguiram prosseguir quando os atributos de qualidades foram discutidos. Neste caso, os participantes procuraram as definições dos atributos citados nos exemplos na ferramenta em sites de busca. Neste momento houve um intercâmbio de informações técnicas entre os participantes sobre estratégias para atender atributos de qualidade, mas sem terem definido quaisquer métricas ou especificações para esses atributos. A analista de integração perguntou sobre como o sistema considera a Lei Geral de Proteção de Dados (LGPD), causando epifania coletiva por esquecerem desse aspecto. Em outro momento, já durante a identificação de Visões e Pontos de Vista, o analista de negócios, lembrou que os integrantes não consideraram a existência de um backoffice para gerência de operações internas da organização, causando epifania coletiva sobre por não terem tratado essa questão. Isso levou ao retorno de conversas sobre processos de negócios, e relatos sobre backoffices em sistemas desenvolvidos pelos participantes. Por fim, um novo requisito funcional foi adicionado na ferramenta.

Observações sobre as decisões

A última hora do experimento foi utilizada para estabelecer as decisões do arquiteto. Assim, os primeiros tradeoffs identificados estavam relacionados a restrições de custos e prazos e levaram à revisão dos requisitos funcionais. O arquiteto identificou módulos de auditoria e monitoramento e decidiu passar do campo de tradeoff para o de táticas, ainda sem ter especificado melhor os atributos de qualidade. Na hora de especificar as táticas arquiteturais, surgiu a necessidade e a epifania de não ter especificado melhor os atributos de qualidade, identificando esse como um tema obscuro para esses participantes. Ao entrar nos Padrões Arquiteturais, houve as primeiras menções sobre tecnologias e uma identificação de uma possível arquitetura legada com SOA, onde o arquiteto explicou a estrutura desse padrão para o Líder de QA. Apesar das

dificuldades de estabelecer representações de arquitetura de uma forma detalhada, uma proposta preliminar se formou ao longo do estudo de caso com um conjunto de decisões de alto-nível, considerando restrições orçamentárias, de cronograma e algumas definições imprecisas sobre Atributos de Qualidade, sendo estas:

- Seleção da Plataforma de Nuvem do Google, considerando um orçamento menor.
- Contratos de uso antecipado, instâncias *spot* ou infraestrutura *on-premise*, visando redução de custos e proteção à variação cambial.
- Construção de um *Minimum Viable Product*, considerando pagamentos e depósitos, para diminuição do escopo e dos custos.
- Plataforma móvel desenvolvida em Flutter para Android e iOS.
- Implementação de um *backoffice web* para gerenciamento de operações bancárias e administrativas.
- Estruturação da plataforma em SOA (arquitetura orientada a serviços) e BFF (*Backend for Frontend*).
- Utilização de emails para suporte, evitando custos com plataformas de *chatbots*.
- Sistema de mensageria para envio de mensagens de monitoramento dos serviços - Usando o padrão *Message Driven Architecture* e software RabbitMQ.
- Uso da ferramenta Docker para escalabilidade e disponibilidade.
- Uso do *API Gateway Krakend*.

Observações sobre a descrição da arquitetura

No momento de estabelecer visões, o arquiteto explicou o funcionamento do padrão BFF (*backend for frontend*) utilizando um diagrama genérico. As restrições de horário dos participantes impossibilitaram um refinamento maior das informações, decisões e representações da arquitetura do sistema, onde apenas decisões “padrões-de-mercado” e de alto nível foram consideradas e representadas na ferramenta.

Observações sobre o metamodelo

Alguns elementos do metamodelo desenvolvido no trabalho estiveram presentes ao longo das interações com a ferramenta, como pôde ser observado ao longo do relato, Requisitos Funcionais foram levantados, Atributos de qualidade impediram os participantes de prosseguir, Padrões Arquiteturais foram propostos, Decisões (arquiteturais) foram tomadas pelo arquiteto,

Tradeoffs buscaram compensar as restrições de custo e prazo e *Visões* foram selecionadas e explicadas.

Os blocos da ferramenta representaram as preocupações, demandas dos participantes e decisões do arquiteto sem necessidade de preencher dados fora da ferramenta, todos os campos foram preenchidos. No entanto, o excesso de liberdade na forma de representar as informações permitiu uma especificação bastante informal, por vezes descuidada, das informações por parte dos utilizadores, algumas foram especificadas com mais detalhes como no caso dos Riscos, Stakeholders e Preocupações, outras tiveram pouca profundidade como Atributos de Qualidade, Objetivos de Negócio, Requisitos Funcionais.

7.2 Respostas Questões Abertas

Nesta seção, se apresenta uma descrição seletiva das respostas, classificadas pelos papéis dos participantes.

7.2.1 Analista de Negócios

O analista de negócios, avalia que a ferramenta “ajudou a ter visão” de áreas que não dominava e a “antecipar possíveis problemas”. Também analisa se a ferramenta poderia representar os elementos de outra forma, “Com exemplos de metodologia de todas as áreas para ajudar quem não tem conhecimento de áreas em específico.” Não aprecia nenhum outro elemento a mais que precise ser considerado na ferramenta. Concorda sobre a criação de uma linguagem comum, e relata que “a área de QA e área de negócio estavam bem alinhadas”. Percebeu que isso “acelerou o processo de definição de alguns quadros” como efeitos dessa linguagem no andamento das atividades, e avalia que a ferramenta teve participação nisso “com exemplos que a ferramenta disponibilizou”. Considera que “o entendimento entre as áreas ficou mais claro”. Avalia que “Com o uso da ferramenta, a comunicação se mostrou eficiente e com velocidade. O que fez que houvesse tempo para aprofundar em demais assuntos.” Sobre o papel da ferramenta nessa comunicação entre os participantes, afirma: “Se mostrou fundamental no experimento, sendo o motor que conectou todas as áreas”. Afirma que houve problemas ou limitações de comunicação, “como havia áreas distintas, levou tempo para contextualizar, porém, com a ferramenta esse tempo pode ser controlado tendo uma diminuição, mesmo assim o problema de tempo para gerar contexto existiu.”. Avalia que pode “compreender o todo”, não teve dúvidas

sobre a arquitetura depois de definida. Afirma que conseguiria executar suas atividades com o entendimento da arquitetura que teve e considera que a ferramenta o auxiliou, onde o “processo simples e os diagramas deram o contexto”.

7.2.2 Líder de QA

O Líder de QA, fez considerações positivas sobre como a ferramenta contempla os elementos de domínio de sua área de trabalho em: “muitas áreas como análise de negócio e de requisitos, assim como também aspectos de qualidade de projeto e gerência de projeto”. Quanto a outras formas de representar os elementos, esse acredita que “a questão de Qualidade poderia ser exemplificada para qualquer um da área conseguir pensar de uma forma melhor e dar ideias”. Sobre os elementos presentes na ferramenta, afirma: “discutimos e trabalhamos em todos os temas abordados”. Concorda sobre a criação de uma linguagem comum, em diferentes momentos. Avalia que a ferramenta impactou no andamento das atividades “produtivamente”. Concorda sobre a influência da ferramenta no estabelecimento dessa linguagem e afirma que esta “guiou a discutir e explorar o problema a ser resolvido”. Avalia a comunicação entre os participantes como “Muito boa”, continua, “todos se comprometendo a entender sobre o projeto e buscando dar o seu melhor. Considera o papel da ferramenta na comunicação como “Muito interessante”, onde “ajudou bastante o brainstorm das ideias”. Afirma que houve um caso onde a ferramenta não pôde ajudar. Nesse sentido, considera: “[...] acredito que, na falta de informação e dúvida sobre determinado board do Canvas, um vídeo explicativo ou um exemplo em cada Card ajudaria bastante a responder as perguntas”. Informa que seu entendimento “foi razoável por não ser especialista”. Revela que “não conhecia alguns dos padrões propostos”, mas que o seu entendimento sobre a mensageria “ficou bem claro”, afirma que conseguiria executar as tarefas da sua área com o entendimento que teve e considera que a ferramenta o auxiliou a entender a arquitetura proposta.

7.2.3 Gerente de Projeto

O Gerente de Projeto, avalia a representação de seus elementos “de forma positiva, apesar de ser muita informação, mesmo que agrupada, o que pode tornar a interface um pouco confusa”. Sobre a representação de elementos, esse não considera nenhum outro elemento adicional “no momento”. Acredita que foi possível identificar essa linguagem comum e afirmou que percebeu isso “durante as análises de cada um dos blocos”. Considera que a ferramenta

“Impactou de forma positiva pois acaba ficando mais fácil integrando as diferentes frentes do projeto”. Porém “quanto mais participantes acredito que o tempo de estruturação do documento seria prejudicado. Da mesma forma que com a ausência de 2 membros da equipe (pelo menos no dia em que participei) prejudicou o preenchimento das informações que provavelmente tornaria o documento mais completo caso todos estivessem presentes. Concorda sobre a participação da ferramenta em estabelecer essa linguagem. “já que a ferramenta se propôs a consolidar conhecimentos de várias áreas diferentes do projeto.” Afirma que “não teve problemas com o entendimento” da arquitetura proposta, onde as “informações foram bem claras” e que “até onde acompanhou”, não teve dúvidas. Acredita que conseguiria executar suas atividades com seu entendimento sobre a arquitetura. Considera que a ferramenta o ajudou, “Talvez pela participação no preenchimento, e pela troca de ideias.”

7.2.4 Analista de integração

A analista de integração, considera a representação satisfatória, e não faz nenhuma consideração sobre formas de representar ou outros elementos a serem representados. Compreende que a comunicação entre os participantes satisfatória, bem como um papel satisfatório da ferramenta no estabelecimento da comunicação entre os participantes, não considerou nenhum problema de comunicação entre os participantes muito menos caso onde a ferramenta não pôde auxiliar. Avalia o entendimento sobre a arquitetura proposta como “Satisfatório”, afirmou não ter dúvidas, e acredita que poderia executar as atividades de sua área de trabalho, com seu entendimento. Concorda com a criação da linguagem e a influencia disso nas atividades e afirma que esse “impactou de forma fluida”

7.2.5 Desenvolvedor

O desenvolvedor concorda sobre a criação da linguagem comum e afirma que “Os tópicos foram bem discutidos tanto no momento de definir os requisitos como no apontamento das soluções.”. Concorda positivamente sobre o impacto dessa linguagem no andamento das atividades, e afirma que houve “Um nível de entendimento satisfatório sobre o impacto positivo no fluxo de trabalho”. A respeito da participação da ferramenta nesta linguagem, afirma que sim e considera que “A ferramenta tende a focar em um tópico por vez trazendo a atenção de todos para um determinado assunto. Considera sobre a comunicação entre os participantes como “muito instrutiva e focada na construção eficiente do projeto”. Além disso, que a ferramenta

“teve um papel importante conduzindo a reunião”, e não possui quaisquer ponderações sobre problemas ou limitações na ferramenta ou casos onde a ferramenta não pôde auxiliar. Avalia o seu entendimento da arquitetura como “Satisfatória”, teve dúvidas sobre “como seria implementado padrão ‘BFF’”. Revela que conseguiria executar suas tarefas, mas “com alguma pesquisa sobre as atividades” e avalia que a ferramenta o auxiliou a entender a arquitetura do sistema, “dando uma visão ampla sobre todo o projeto desde sua elaboração até como se espera que seja estando finalizado”.

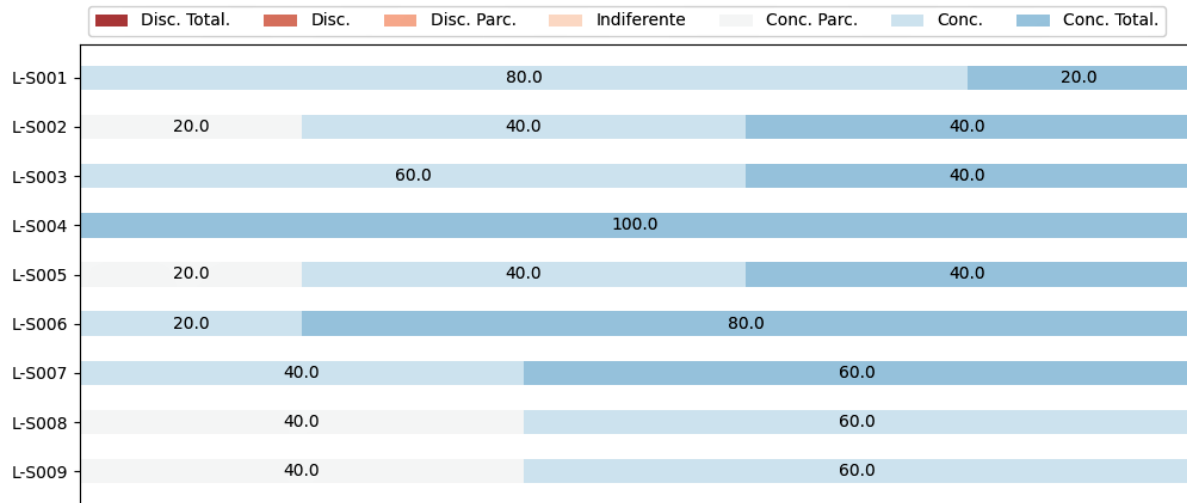
7.2.6 Arquiteto de Software

O Arquiteto avaliou a representação de forma positiva. Afirma que a ferramenta “ajudou o grupo a seguir uma linha de raciocínio e voltar toda vez que eram identificados itens que ficaram de fora em algum card anterior”. Entende que “padrões de projeto poderiam ficar de fora para otimizar o tempo do grupo maior, e que isso é algo que analistas (por exemplo) não conseguem contribuir muito”. Afirma “Não tenho certeza”, sobre os elementos serem considerados de outra forma, mas imagina “que a metodologia/forma proposta para tratar os atributos de qualidade seja algo de muita relevância para o desenvolvimento de um projeto de software.” Considera que “a forma de representá-los não é a mais produtiva”. E propõe “um card com requisitos não funcionais” com uma possível “participação um pouco maior dos analistas”. Afirma que foi possível identificar uma linguagem comum entre os stakeholders, e afirma não ter “sentido uma desconexão em relação aos termos mais técnicos.” Sobre a participação da ferramenta em estabelecer uma linguagem comum, afirma: “Talvez por a ferramenta ter abordado os assuntos de forma mais macro até ir aprofundando em detalhes mais técnicos posteriormente, imagino que isso foi "contextualizando" os stakeholders menos técnicos em arquitetura.”. Avaliou a comunicação do projeto de arquitetura como “bem satisfatória”. Considera que “Poderia ser melhor com uma maior participação dos analistas”. Avalia que “isso vai de grupo para grupo”. Não identificou nenhuma dificuldade de comunicação.

7.3 Respostas Questões Fechadas

Na Figura 6 estão representados os resultados das respostas classificadas em escala Likert. O Alpha de Cronbach obtido dessas respostas é: 0.76 sendo inferior a 0.8 e superior a 0.61, classificando a consistência das questões fechadas como "Substancial".

Figura 6 – Respostas Fechadas



Fonte: elaborado pelo autor.

As interpretações a respeito dos dados representados nesta imagem estão apresentados na próxima subseção.

Entre os participantes, exceto o arquiteto, **todos os participantes concordam em algum nível que a ferramenta representou os elementos de suas áreas de trabalho de forma satisfatória**, dividindo-se em 80% concordando (apenas) e 20% concordando totalmente com esse aspecto.

Sobre a identificação de uma linguagem comum entre os participantes, **mesmo com uma maior segmentação das opiniões, todos concordam em algum nível**, onde 40% concordam totalmente em identificar uma linguagem comum entre os participantes, 40% concordam (apenas) e 20% concordam parcialmente. Já sobre uma participação relevante para buscar estabelecer uma linguagem comum ficou em 60% os participantes concordando (apenas) e 40% concordando totalmente com esta afirmativa.

Quanto aos aspectos de comunicação, **todos os participantes (100%) concordaram (apenas) que foi possível ter uma comunicação pessoal satisfatória para troca de informações entre os participantes**. Quanto ao papel da ferramenta na comunicação pessoal desses participantes (individualmente), 40% concordam totalmente, 40% concordam e 20% concordam parcialmente com essa afirmativa. 80% concordaram totalmente sobre a participação da ferramenta na comunicação entre os participantes (coletiva), 20% concordam com esta afirmativa. Sobre a participação da ferramenta na comunicação entre os participantes, 60% concordam totalmente, 40% concordam (apenas) com esta afirmativa.

Finalizando, sobre os aspectos do entendimento do projeto de arquitetura proposto,

esses não tiveram níveis tão altos quanto à concordância aos aspectos propostos, mas ainda ficaram em algum nível de concordância. Nas respostas das questões relacionadas a esse aspecto, identificou-se que 60% concordaram (apenas) sobre ter um entendimento satisfatório do projeto de arquitetura, 40% concordam parcialmente com essa afirmativa, quanto a participação da ferramenta no entendimento desses no projeto de arquitetura da mesma medida 60% dos participantes concordaram (apenas) e 40% concordam parcialmente com essa afirmativa.

7.4 Análise dos resultados da aplicação

Serão apresentados adiante discussões sobre as percepções dos participantes em conjunto com as percepções da observação direta sobre a ferramenta após o estudo de caso, conforme os critérios estabelecidos.

1. A ferramenta representou um metamodelo de AS dos elementos de interesse dos diferentes stakeholders na AS de forma satisfatória?

Objetivo: esse critério visa entender a abrangência do modelo conceitual presente na ferramenta e manifestações de uma abstração comum por parte dos participantes, correspondente ao metamodelo de arquitetura explicado na fundamentação teórica.

Análise: Não houve nenhuma discordância ou indiferença a respeito da representação das informações na ferramenta, todas as avaliações estiveram nos níveis mais altos de concordância a esse ponto. Alguns participantes (nas respostas abertas) consideraram a identificação de elementos em diferentes áreas de trabalho que promoveram uma "visão, ou linha de raciocínio". Todos os elementos presentes na ferramenta se manifestaram nas discussões e decisões dos participantes. Esses, nas respostas abertas, não identificaram nenhum item fora do escopo da ferramenta e não apresentaram considerações sobre itens adicionais a serem representados na ferramenta, entendendo-se, com essas informações uma abrangência nesses elementos. No entanto, houve considerações sobre a forma da representação de elementos específicos na ferramenta, com maior ênfase nos Atributos de Qualidade, e observações sobre a profundidade dos Objetivos de Negócios representados. Foi possível constatar a manifestação das relações entre os elementos (de acordo com o metamodelo elaborado neste trabalho) em momentos de revisão de informações e na observação do relacionamento entre Atributos de Qualidade e as Decisões (coluna da ferramenta) apresentadas nas perguntas nos campos da coluna decisão,

causando dificuldades na proposição de decisões, sendo um “elemento de barreira”, entendendo com estas informações uma consistência no metamodelo que a ferramenta representa.

Conclusão: Com as informações apresentadas, entende-se que a ferramenta fornece um metamodelo abrangente e consistente de AS, e que, na opinião dos utilizadores foi satisfatório em representar seus elementos de trabalho. Porém, em alguns desses elementos não foi possível uma representação em detalhes e de forma produtiva pela ferramenta, principalmente em Objetivos de Negócios e Atributos de Qualidade.

2. A ferramenta foi efetiva em estabelecer uma linguagem comum entre os diferentes stakeholders?

Objetivo: este critério busca identificar a existência de uma linguagem comum entre os participantes nos termos dos elementos do metamodelo.

Análise: Nas respostas abertas, todos os participantes concordaram sobre a existência de uma linguagem comum durante as atividades, mesmo com uma maior divisão das opiniões nas questões fechadas. No entanto, a maior parte concorda sobre a participação da ferramenta na comunicação pessoal desses participantes. Todas as respostas consideram a identificação de uma linguagem comum. Algumas respostas contemplam efeitos relacionados a essa linguagem entre os participantes.

Conclusão: A ferramenta estabeleceu uma linguagem comum, na avaliação dos utilizadores, tendo um papel relevante nesse aspecto.

3. A ferramenta possibilitou uma comunicação satisfatória (para a troca de ideias e informações, discussões, negociações e outros) entre os envolvidos durante as atividades de arquitetura de software?

Objetivo: esse aspecto busca verificar não apenas as percepções sobre a comunicação individual, mas também da comunicação coletiva. Caso algum participante eventualmente não tenha interagido de forma constante, poderá avaliar a comunicação entre os participantes.

Análise: Na avaliação de todos os participantes houve uma comunicação totalmente satisfatória. Além disso, ocorreu uma divisão maior sobre a concordância da participação da ferramenta na comunicação pessoal. Ainda assim, esses expõem níveis mais altos de concordância

sobre a comunicação entre os participantes, do ponto de vista individual e com participação da ferramenta na comunicação. A observação direta, constatou uma comunicação constante entre os participantes contando com discussões, intercâmbio de informações e alinhamento (coletivo, sobre questões levantadas), nas áreas técnicas e de negócios. Houve considerações na forma que a ferramenta representa as informações. Esse tópico já foi discutido em critérios anteriores.

Conclusão: Houve uma comunicação satisfatória na percepção dos participantes, mas ainda há considerações sobre a representação de elementos da ferramenta.

4. A ferramenta auxiliou o arquiteto e SH a alcançarem os objetivos das atividades de análise de AS (identificar e representar os requisitos arquiteturalmente significativos) de maneira satisfatória?

Objetivo: Esse aspecto busca compreender a participação da ferramenta na identificação de Requisitos Arquiteturalmente Significativos, identificados aqui como Requisitos Funcionais, Restrições e Atributos de Qualidade.

Análise: Como retratado em outros critérios, houve dificuldades na identificação e representação de requisitos de qualidade, com impactos nas decisões do arquiteto ao longo das atividades de projeto.

Conclusão: Há a necessidade de refinamentos na pesquisa bibliográfica que contemplem os Atributos de Qualidade, seus atributos e suas relações.

5. A ferramenta auxiliou o arquiteto e SH a alcançar os objetivos das atividades de AS (representar e comunicar o projeto de arquitetura proposto) de maneira satisfatória?

Objetivo: esse aspecto busca compreender a participação da ferramenta na representação e comunicação do projeto de arquitetura de software e decisões do arquiteto.

Análise: Os participantes apontaram níveis menores de concordância relacionados às afirmações sobre aspectos de entendimento da arquitetura. Apesar de o arquiteto ter definido um conjunto de decisões que consideraram os requisitos identificados, a especificação restringiu-se apenas a decisões e representações de alto-nível. O arquiteto conseguiu estabelecer um conjunto de decisões (como explicado nos resultados da observação direta) que consideraram, mesmo que em parte, os requisitos arquiteturalmente significativos, com a participação direta dos envolvidos

nas decisões do arquiteto nas discussões e troca de informações entre esses. No entanto, a especificação dos Atributos de Qualidade durante as atividades dificultaram a tomada de decisões mais aprofundadas. Reafirma-se o posto exposto pela observação direta: “As restrições de horário dos participantes impossibilitaram um refinamento maior das informações, decisões e representações da arquitetura do sistema”.

Conclusão: As interações com o uso da ferramenta possibilitaram o estabelecimento de um conjunto de decisões mesmo que em alto nível, envolvendo a participação e discussão dos participantes. Entretanto, as restrições de horário dos participantes impossibilitaram um melhor refinamento das informações, decisões e representações da arquitetura do sistema.

8 AMEAÇAS À VALIDADE E OUTRAS CONSIDERAÇÕES

Este trabalho também discute as ameaças à validade consideradas nesta pesquisa. Teixeira *et al.* (2018 apud COOK *et al.*, 1979) expõem uma taxonomia com categorias de ameaças à validade de um experimento, sendo essas: validade de conclusão, validade interna, validade do constructo, e validade externa.

Sobre a validade interna, devem ser considerados o detalhamento da carta-proposta apresentada no ensaio do estudo de caso, o detalhamento da pesquisa bibliográfica e do modelo conceitual desenvolvidos. Também deve-se considerar as habilidades dos participantes em suas áreas de conhecimento, como por exemplo o arquiteto na falta de uma especificação detalhada de atributos de qualidade, habilidades pessoais de comunicação e conhecimento prévio dos participantes sobre BMC. Apenas um grupo de participantes esteve nos dois dias do experimento (arquiteto de software, analista de negócios, Líder de QA). Tempo disponível para o estudo de caso e disponibilidade dos participantes impossibilitaram maiores refinamentos, assim como o cansaço dos participantes, visto que os ensaios foram executados após o horário comercial.

Em relação à validade de constructo, devem ser consideradas as intervenções durante o experimento para orientação sobre dúvidas acerca do uso da ferramenta nos momentos solicitados pelos utilizadores. Os participantes do estudo de caso estavam cientes do objetivo do ensaio e da ferramenta, podendo-se considerar a influência disso no desempenho desses participantes.

Sobre a validade externa, deve ser considerado a incapacidade da generalização dos resultados por fatores de representatividade: participantes não representam todos os papéis e níveis hierárquicos possíveis disponíveis no mercado de desenvolvimento de software, representando apenas um pequeno grupo em relação ao conjunto academia/mercado, a participação de apenas um arquiteto no experimento. Fatores sobre o ambiente operacional devem ser considerados, visto que o experimento foi executado remotamente, em uma plataforma de vídeo chamadas. Vale considerar também o treinamento dado para o arquiteto antes do ensaio.

Outras considerações sobre esta pesquisa envolvem os questionários aplicados, que poderiam capturar outros pontos sobre o perfil dos participantes em maior profundidade, como conhecimento e experiência prévia com BMC. Além disso, perguntas sobre o tempo de experiência dos participantes possuíam faixas muito restritas dado o tempo de experiência dos participantes.

9 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho desenvolveu, apresentou e aplicou uma ferramenta que, conforme os resultados obtidos no estudo de caso, possibilitou elementos semelhantes ao BMC como uma abstração (do metamodelo de arquitetura de software) e linguagem comum entre *stakeholders*, de forma estabelecer uma comunicação satisfatória entre os participantes.

Também foi possível obter níveis satisfatórios, porém não plenos, no auxílio aos *stakeholders* no alcance dos objetivos relacionados às atividades de análise e projeto, contando porém com ressalvas sobre a representação de elementos nessa ferramenta.

Em complemento a esses resultados, observou-se a necessidade de refinamentos na pesquisa bibliográfica, para um aprofundamento da representação dos elementos identificados. Também foram identificadas ameaças à validade dos resultados desta pesquisa, que podem advertir algumas precauções para a execução deste estudo de caso.

Com isso, trabalhos futuros podem ser executados, tais como: aplicação da ferramenta em outras equipes com diferentes cenários, em estudos de casos múltiplos, na criação de outros modelos a partir de uma ontologia, validações do modelo conceitual, adaptação de outras ferramentas e métodos para utilização conjunta com a ferramenta. Também vislumbra-se a possibilidade de desdobramentos deste trabalho em outras áreas da Engenharia de Software, como: requisitos, qualidade, planejamento de testes e gerência ágil de projetos de software.

REFERÊNCIAS

- ABNT/ISO/IEC. **NBR ISO/IEC 9126-1: Engenharia de Software – Qualidade de produto. Parte 1: Modelo de qualidade.** [S. l.], 2003.
- AMARSY, N. **Why and How Organizations Around the World Apply the Business Model Canvas.** 2015. Disponível em: <http://web.archive.org/web/20191227123330/https://www.strategyzer.com/blog/posts/2015/2/9/why-and-how-organizations-around-the-world-apply-the-business-model-canvas>. Acesso em: 01 out. 2020.
- AZEVEDO, P. D. de Abreu Mancini de; REIS, P.; FREITAS, F.; SILVA, S. Strategic model canvas: Uma proposta de ferramenta para otimizar o planejamento estratégico. **Revista de Gestão e Projetos**, Universidade Nove de Julho, v. 9, n. 3, p. 1–32, 12 2018.
- BASS, L.; CLEMENTS, P.; KAZMAN, R. **Software Architecture in Practice: Software architect practice.** [S. l.]: Addison-Wesley Professional, 2012. (SEI Series in Software Engineering). ISBN 0321815734.
- BELLONI, A. **Sistema híbrido Fuzzy e AHP para análise de modelo de negócios baseado em Business Model Canvas e SWOT.** Dissertação (Mestrado) – Instituto de Tecnologia para o Desenvolvimento, Institutos Lactec, Curitiba, 2016.
- BOURQUE, P.; FAIRLEY, R.; SOCIETY, I. C. **Software Engineering Body of Knowledge.** [S. l.]: IEEE Computer Society, 2014. ISBN 9780769551661.
- CHEN, L.; BABAR, M. A.; NUSEIBEH, B. Characterizing architecturally significant requirements. **IEEE Software**, v. 30, n. 2, p. 38–45, 2013.
- CLEMENTS, P.; BACHMANN, F.; BASS, L.; GARLAN, D.; IVERS, J.; LITTLE, R.; MERSON, P.; NORD, R.; STAFFORD, J. **Documenting Software Architectures: Views and beyond.** Second. [S. l.]: Addison-Wesley Professional, 2010. (SEI Series in Software Engineering). ISBN 9780132488594.
- COOK, T. D.; CAMPBELL, D. T.; DAY, A. **Quasi-experimentation: Design & analysis issues for field settings.** [S. l.]: Houghton Mifflin Boston, 1979. v. 351.
- CRONBACH, L. J. Coefficient alpha and the internal structure of tests. **psychometrika**, Springer, v. 16, n. 3, p. 297–334, 1951.
- DASANAYAKE, S.; MARKKULA, J.; AARAMAA, S.; OIVO, M. Software architecture decision-making practices and challenges: an industrial case study. In: IEEE. **2015 24th Australasian Software Engineering Conference.** [S. l.], 2015. p. 88–97.
- EVANS, E.; FOWLER, M. **Domain-driven Design: Tackling complexity in the heart of software.** [S. l.]: Addison-Wesley, 2004. ISBN 9780321125217.
- FALESSI, D.; BABAR, M. A.; CANTONE, G.; KRUCHTEN, P. Applying empirical software engineering to software architecture: challenges and lessons learned. **Empirical Software Engineering**, Springer, v. 15, n. 3, p. 250–276, 2010.
- FAVRE, J.-M. Cacophony: metamodel-driven software architecture reconstruction. In: **11th Working Conference on Reverse Engineering.** [S. l.: s. n.], 2004. p. 204–213.

- GANGA, G. M. D. **Trabalho de conclusão de curso (TCC) na engenharia de produção: um guia prático de conteúdo e forma.** São Paulo: Atlas, 2012.
- GIL, A. **Como Elaborar Projetos de Pesquisa (5a. Ed.).** [S. l.]: Editora Atlas S.A., 2000. ISBN 9788522478408.
- GIL, A. **Métodos e técnicas de pesquisa social.** [S. l.]: Atlas, 2008. ISBN 9788522451425.
- GORTON, I. **Essential Software Architecture.** [S. l.]: Springer Science & Business Media, 2011. ISBN 9783642191756.
- HOLT, R. Software architecture as a shared mental model. **Proceedings of the ASERC Workshop on Software Architecture, University of Alberta,** Citeseer, p. 64, 2002.
- IIBA. **BABoK: A guide to the business analysis body of knowledge.** Toronto, Ontario, Canada: International Institute of Business Analysis, 2015. ISBN 9781927584033.
- ISO/IEC/IEEE. **ISO/IEC/IEEE Systems and software engineering - Architecture description.** [S. l.], 2011. 1-46 p.
- ISO/IEC/IEEE. **ISO/IEC/IEEE International Standard - Systems and software engineering – Software life cycle processes.** [S. l.], 2017. 1-157 p.
- JANSEN, A.; BOSCH, J. Software architecture as a set of architectural design decisions. In: **5th Working IEEE/IFIP Conference on Software Architecture (WICSA'05).** [S. l.: s. n.], 2005. p. 109–120.
- KRUCHTEN, P. **Introdução ao RUP: Rational unified process.** [S. l.]: Ciência Moderna, 2003. (Addison-Wesley object technology series). ISBN 9788573932751.
- LIKERT, R. A technique for the measurement of attitudes. **Archives of psychology,** 1932.
- MERRIAM, S. **Qualitative Research: A guide to design and implementation.** [S. l.]: John Wiley & Sons, 2009. (Higher and adult education series). ISBN 9780470283547.
- NOVAK, J. D. **A theory of education.** [S. l.]: Cornell University Press, 1977.
- OSTERWALDER, A. **The business model ontology a proposition in a design science approach.** Tese (Doutorado) – Université de Lausanne, Faculté des hautes études commerciales, Lausanne, 2004.
- OSTERWALDER, A.; PIGNEUR, Y. **Business Model Generation - Inovação em Modelos de Negócios: um manual para visionários, inovadores e revolucionários.** Rio de Janeiro: Alta Books, 2011. ISBN 9788576085508.
- PAPATHEOCHAROUS, E.; PETERSEN, K.; AXELSSON, J.; WOHLIN, C.; CARLSON, J.; CICOZZI, F.; SENTILLES, S.; CICHETTI, A. The grade decision canvas for classification and reflection on architecture decisions. In: **12th International Conference on Evaluation of Novel Approaches to Software Engineering.** Porto, Portugal: [S. n.], 2017. p. 187–194.
- PAPATHEOCHAROUS, E.; PETERSEN, K.; CICHETTI, A.; SENTILLES, S.; SHAH, S. M. A.; GORSCHKE, T. Decision support for choosing architectural assets in the development of software-intensive systems: The grade taxonomy. In: **Proceedings of the 2015 European Conference on Software Architecture Workshops.** [S. l.: s. n.], 2015. p. 1–7.

PRESSMAN, R.; MAXIM, B. **Engenharia de Software**. 8ª edição. ed. [S. l.]: McGraw Hill Brasil, 2016. ISBN 9788580555349.

REEKIE, J.; MCADAM, R.; MCADAM, R. **A Software Architecture Primer**. [S. l.]: Angophora Press, 2006. ISBN 9780646458410.

SMOLANDER, K. Four metaphors of architecture in software organizations: finding out the meaning of architecture in practice. In: IEEE. **Proceedings International Symposium on Empirical Software Engineering**. [S. l.], 2002. p. 211–221.

TEIXEIRA, E.; FONSECA, L.; SOARES, S. Threats to validity in controlled experiments in software engineering: What the experts say and why this is relevant. In: **Proceedings of the XXXII Brazilian Symposium on Software Engineering**. New York, NY, USA: Association for Computing Machinery, 2018. (SBES '18), p. 52–61. ISBN 9781450365031.

YIN, R. K. **Estudo de Caso: Planejamento e métodos**. [S. l.]: Bookman editora, 2015.

APÊNDICE A – DOCUMENTO DE CAPTURA DE CONHECIMENTO

Este documento contém as informações identificadas na pesquisa bibliográfica, na execução da etapa 1 dos procedimentos metodológicos deste trabalho.

Partes Interessadas (**Stakeholders**)

Definições

ISO 42010, 2011, p. 2: Indivíduo do sistema, equipe, organização ou classes dos mesmos, com interesse em um sistema.

Kruchten, 2003, p. 236: Qualquer pessoa ou representante de uma organização que tenha interesse - adquirido - no resultado de um projeto, ou cuja opinião deva ser acomodada. Um interessado pode ser um usuário final, comprador, contratante, desenvolvedor ou gerente de projeto.

Bass et al., 2012, p. 52: Uma parte interessada é qualquer pessoa que tenha interesse no sucesso do sistema: o cliente, os usuários finais, os desenvolvedores, o gerente de projeto, os mantenedores e até mesmo aqueles que comercializam o sistema, por exemplo.

Exemplos de Stakeholders

Gerente de projetos

- **Kruchten, 2003, p. 224:** O gerente de projeto aloca recursos, formula prioridades, coordena integrações com clientes e usuários, e geralmente tenta manter a equipe do projeto concentrada na meta definida. O gerente de projeto estabelece um conjunto de práticas para assegurar a integridade e a qualidade dos artefatos do projeto.
- **Bass et al., 2012, p. 55:** Responsável por planejar, sequenciar, agendar e alocar recursos para desenvolver componentes de software e entregar componentes para integração e atividades de teste.

Implementador/Desenvolvedor

- **Kruchten, 2003, p. 224:** Um implementador é responsável por desenvolver e testar os componentes conforme os padrões adotados pelo projeto, de forma que eles possam ser integrados em subsistemas maiores. Quando componentes de teste, como condutores e fragmentos, devem ser criados para apoiar o teste, o implementador também é responsável por desenvolver e testar os componentes de teste e subsistemas correspondentes.
- **Bass et al., 2012, p. 55:** Responsável pelo desenvolvimento de elementos específicos de acordo com projetos, requisitos e arquitetura.

Cliente

- **Bass et al., 2012, p. 54:** Paga pelo sistema e garante sua entrega. O cliente geralmente fala ou representa o usuário final, especialmente em um contexto de aquisição do governo.

Avaliador

- **Bass et al., 2012, p. 54:** Responsável por conduzir uma avaliação formal da arquitetura (e sua documentação) em relação a alguns critérios claramente definidos;

Usuário

- **Bass et al., 2012, p. 55:** Os usuários finais reais do sistema. Pode haver tipos distintos de usuários, como administradores, superusuários, etc.

Implantador

- **Bass et al., 2012, p. 54:** Responsável por aceitar o sistema completo do esforço de desenvolvimento e implantá-lo, torná-lo operacional e cumprir sua função de negócios alocada.

Integrador

- **Kruchten, 2003, p. 75:** Responsáveis pela integração integram componentes de softwares principais até mesmo se a implementação deles for muito rudimentar, para verificar as interfaces. Os integradores focalizam principalmente em remover riscos de integração relacionados a componentes principais fora de uso ou reutilizáveis.

Engenheiro de Sistemas

- **Bass et al., 2012, p. 55:** Responsável pelo projeto e desenvolvimento de sistemas ou componentes de sistema nos quais o software desempenha um papel.

Arquiteto de Software

- **Bass et al., 2012, p. 54:** Responsável pelo desenvolvimento da arquitetura e sua documentação. O foco e a responsabilidade estão no sistema.

Analista

- **Bass et al., 2012, p. 54:** Responsável por analisar a arquitetura para garantir que ela atenda a determinados requisitos críticos de atributos de qualidade. Os analistas geralmente são especializados; por exemplo, analistas de desempenho, analistas de segurança e analistas de segurança podem ter posições bem definidas em um projeto.

Testador

- **Kruchten, 2003, p. 75:** Os responsáveis pelos testes testam o protótipo arquitetônico para desempenho e robustez.
- **Bass et al., 2012, p. 55:** Responsável pelo teste (independente) e verificação do sistema ou de seus elementos em relação aos requisitos formais e a arquitetura.

Gerente de Banco de Dados

- **Bass et al., 2012, p. 54:** Envolvido em muitos aspectos dos armazenamentos de dados, incluindo design de banco de dados, análise de dados, modelagem e otimização de dados, instalação de software de banco de dados e monitoramento e administração de segurança de banco de dados.

Gerente de Negócios

- **Bass et al., 2012, p. 54:** Responsável pelo funcionamento da entidade empresarial / organizacional proprietária do sistema. Inclui responsabilidade gerencial / executiva, responsabilidade pela definição de processos de negócios, etc.

Administrador de Rede

- **Bass et al., 2012, p. 55:** Responsável pela manutenção e supervisão de hardware e software de computador em uma rede de computadores. Isso pode incluir a implantação, configuração, manutenção e monitoramento de componentes de rede.

Representante de Sistema Externo

- **Bass et al., 2012, p. 55:** Responsável pela gestão de um sistema com o qual este deve interoperar, e sua interface com o nosso sistema.

Projetista

- **Bass et al., 2012, p. 54:** Responsável pelo projeto de sistemas e/ou software a jusante da arquitetura, aplicando a arquitetura para atender requisitos específicos das partes pelas quais são responsáveis.
- **Kruchten, 2003, p. 225:** O projetista define responsabilidades, operações atributos e relações de uma ou várias classes, determinando como elas deveriam ser ajustadas ao ambiente de implementação. Além disso, o projetista pode ter responsabilidade por um ou mais pacotes do projeto ou subsistemas do projeto, inclusive ou subsistemas do projeto, inclusive classes mantidas pelos pacotes ou subsistemas.

Verificador de conformidade

- **Bass et al., 2012, p. 55:** Responsável por garantir a conformidade com padrões e processos para fornecer confiança na adequação de um produto.

Gerente de Linha de Produto

- **Bass et al., 2012, p. 55:** Responsável pelo desenvolvimento de toda uma família de produtos, todos construídos usando os mesmos ativos principais (incluindo a arquitetura).

Preocupações (Concerns)

Definições

ISO 42010, 2011, p. 2: Interesse em um sistema relevante para uma ou mais de suas partes interessadas.

Bass et al., 2012, p. 52: (...) as partes interessadas, apesar de todos terem uma participação compartilhada no sucesso do sistema, normalmente têm diferentes preocupações específicas que desejam que o sistema garanta ou otimize. Essas preocupações são tão diversas quanto fornecer um determinado comportamento em tempo de execução, ter um bom desempenho em uma determinada peça de hardware, ser fácil de personalizar, alcançar um curto tempo de colocação no mercado ou baixo custo de desenvolvimento, empregar lucrativamente programadores que têm uma especialidade específica ou fornecer uma ampla gama de funções.

ISO 42010, 2011, p. 6: [1] As partes interessadas de um sistema têm preocupações com relação ao sistema de interesse considerado em relação ao seu ambiente. Uma preocupação pode ser mantida por uma ou mais partes interessadas.

ISO 42010, 2011, p. 6: [2, 3, 4, 5, 6, 7] Uma preocupação pode se manifestar de várias formas, como em relação a uma ou mais necessidades, **objetivos**, expectativas, responsabilidades, **requisitos**, **restrições de projeto**, suposições, dependências, **atributos de qualidade**, **decisões de arquitetura**, **riscos** ou outras questões pertencentes ao sistema.

Relacionamentos

#	Classe	Relacionamento	Classe
1	Partes interessadas (Stakeholders)	tem	Preocupações
2	Requisitos Funcionais	é manifestação	Preocupações (Concerns)
3	Restrições (Constraints)	é manifestação	Preocupações (Concerns)
4	Atributos de Qualidade	é manifestação	Preocupações (Concerns)
5	Riscos	é manifestação	Preocupações (Concerns)
6	Objetivos (de negócios)	é manifestação	Preocupações (Concerns)
7	Decisões	é manifestação	Preocupações (Concerns)

Propriedades

Influências	ISO 42010, 2011, p. 2: desenvolvimentista, tecnológico, empresarial, operacional, organizacional, político, econômico, jurídico, regulatório, ecológico, social.
--------------------	---

Exemplos de Preocupações

ISO 42010, 2011, p. 6: funcionalidade, viabilidade, uso, finalidades do sistema, recursos do sistema, propriedades do sistema, limitações conhecidas, estrutura, comportamento, desempenho, utilização de recursos, confiabilidade, segurança, garantia da informação, complexidade, evolucionabilidade, abertura, simultaneidade, autonomia, custo, cronograma, qualidade de serviço, flexibilidade, agilidade, modificabilidade, modularidade, controle, comunicação entre processos, impasse, mudança de estado, integração de subsistema, acessibilidade de dados, privacidade, conformidade com a regulamentação, garantia, objetivos e estratégias de negócios, experiência do cliente, manutenção, acessibilidade e disponibilidade.

ISO 42010, 2011, p. 12: As seguintes preocupações devem ser consideradas e, quando aplicável, identificadas na descrição da arquitetura:

- **os propósitos do sistema;**
- **a adequação da arquitetura para atingir os objetivos do sistema;**
- **a viabilidade de construção e implantação** do sistema;
- **os riscos e impactos potenciais do sistema para suas partes interessadas** ao longo de seu ciclo de vida; manutenibilidade e evolução do sistema.

Objetivos de Negócio (**Business Goals**)

Definições

IBA (BABOK), 2015, p. 442: Um estado ou condição que uma organização está procurando estabelecer e manter, geralmente expresso de forma qualitativa, em vez de quantitativa.

IBA (BABOK), 2015, p. 113: Metas e objetivos de negócios descrevem os fins que a organização busca atingir. Metas e objetivos podem estar relacionados às mudanças que a organização deseja realizar ou às condições atuais que deseja manter.

Bass et al., 2012, p. 50: [8, 9] Alguns **objetivos de negócios podem levar a requisitos de atributos de qualidade** (que levam a arquiteturas), **ou levar diretamente a decisões arquitetônicas**, ou levar a soluções não arquitetônicas.

Bass et al., 2012, p. 297: [13] **Vale a pena capturar os objetivos de negócios explicitamente. Isso ocorre porque eles geralmente implicam ASRs** que, de outra forma, não seriam detectados até que seja tarde demais ou muito caro para resolvê-los. Capturar objetivos de negócios é bem servido por ter um conjunto de objetivos de negócios candidatos à mão para usar como iniciadores de conversas.

Bass et al., 2012, p. 296: Os objetivos de negócios são a razão de ser para construir um sistema. Nenhuma organização constrói um sistema sem uma razão; em vez disso, os líderes da organização querem promover a missão e as ambições de sua organização e de si mesmos. Objetivos comuns de negócios incluem lucro, é claro, mas a maioria das organizações tem muito mais preocupações do que simplesmente lucro, e em outras organizações (por exemplo, organizações sem fins lucrativos, instituições de caridade, governos), o lucro é a coisa mais distante da mente de qualquer pessoa.

Os objetivos de negócios são de interesse dos arquitetos porque geralmente são os precursores ou progenitores de requisitos que podem ou não ser capturados em uma especificação de requisitos, mas cuja realização (ou falta) sinaliza um projeto de arquitetura bem-sucedido (ou menos que bem-sucedido). **As objetivos de negócios geralmente levam diretamente a ASRs.**

Relacionamentos

#	Classe	Relacionamento	Classe
8	Objetivo de Negócio (Business Goal)	conduz	Atributo de Qualidade
9	Objetivos de Negócios (Business Goal)	leva a	Decisão (Decision)
10	Objetivos de Negócios (Business Goal)	leva	ASR

Exemplos de Objetivos de Negócios

IBA (BABOK), 2015, p. 113

- Criar um novo recurso, como um novo produto ou serviço, resolva uma desvantagem competitiva ou criar uma nova vantagem competitiva.
- Melhorar a receita aumentando as vendas ou reduzindo custos.
- Aumentar a satisfação do cliente.
- Aumentar a satisfação dos funcionários.
- Cumprir os novos regulamentos.

- Melhorar a segurança.
- Reduzir o tempo de entrega de um produto ou serviço.

Categorias de Objetivos de Negócios

Bass et al., 2012, p. 297: Nossa pesquisa em objetivos de negócios nos levou a adotar as categorias apresentadas na Tabela 16.2. Essas categorias podem ser usadas como uma ajuda para brainstorming e elicitação. Ao empregar a lista de categorias e perguntar às partes interessadas sobre possíveis objetivos de negócios em cada categoria, obtém-se alguma garantia de cobertura.

Tabela 16.2 Uma Lista de Categorias de Objetivos de Negócios Padrão

1. Contribuir para o crescimento e continuidade da organização
2. Cumprimento dos objetivos financeiros
3. Atingindo objetivos pessoais
4. Responsabilidade com os funcionários
5. Responsabilidade perante a sociedade
6. Reunião da responsabilidade de declarar
7. Responsabilidade da assembleia perante os acionistas
8. Gerenciando a posição de mercado
9. Melhorar os processos de negócios
10. Gerenciando a qualidade e reputação dos produtos
11. Gerenciando mudanças nos fatores ambientais

Essas categorias não são completamente ortogonais. Algumas metas de negócios podem se encaixar em mais de uma categoria, e tudo bem. Em um método de elicitação, as categorias devem suscitar perguntas sobre a existência de objetivos de negócios organizacionais que se enquadram nessa categoria.

Requisitos Arquiteturalmente Significativos (ASRs)

Definições

Bass et al., 2012, p. 291: Um requisito significativo do ponto de vista da arquitetura (ASR) é um requisito que terá um efeito profundo na arquitetura - ou seja, a arquitetura pode muito bem ser drasticamente diferente na ausência de tal requisito.

Kruchten, 2003, p. 235: (Requisito) Descrição de uma condição ou capacidade de um sistema; qualquer uma derivada diretamente das necessidades do usuário ou declarada num contrato, padrão, especificação ou outro documento formalmente imposto.

Gorton, 2011, p. 92: Os requisitos de arquitetura, às vezes também chamados de requisitos significativos do ponto de vista da arquitetura ou casos de uso de arquitetura, são essencialmente os requisitos de qualidade e não funcionais do aplicativo.

Bass et al., 2012, p. 64: Não importa a fonte, todos os requisitos abrangem as seguintes categorias:

[10] Funcionais: Esses requisitos determinam o que o sistema deve fazer, e como ele deve se comportar ou reagir aos estímulos de tempo de execução.

[11] Atributos de Qualidade: Esses requisitos são qualificações de requisitos funcionais ou do produto como um todo. A qualificação de um requisito funcional é um item como a rapidez com que a função deve ser executada, ou quão resiliente deve ser a entrada errônea.

[12] Restrições: Uma restrição é uma decisão de projeto com zero graus de liberdade. Ou seja, é uma decisão de design que já foi feita.

Relações

#	Classe	Relacionamento	Classe
11	Requisitos Funcionais	é	ASR
12	Atributos de Qualidade	é	ASR
13	Restrições (Constraints)	é	ASR

Requisitos Funcionais

Definições

Kruchten, 2003, p. 132: Requisitos funcionais são usados para expressar o comportamento de um sistema especificando a contribuição e as condições esperadas de saída.

Kruchten, 2003, p. 235: (Requisito) Descrição de uma condição ou capacidade de um sistema; qualquer uma derivada diretamente das necessidades do usuário ou declarada num contrato, padrão, especificação ou outro documento formalmente imposto.

Bass et al., 2012, p. 64: Esses requisitos determinam o que o sistema deve fazer, e como ele deve se comportar ou reagir aos estímulos de tempo de execução.

IBA (BABOK), 2015, p. 16: Descrevem os recursos que uma solução deve ter em termos de comportamento e informações que a solução vai administrar.

IEEE (SWEBoK), 2014, p. 34: Os requisitos funcionais descrevem as funções que o software deve ser executado; por exemplo, formatar algum texto ou modular um sinal. Elas às vezes são conhecidas como recursos ou recursos. Um requisito funcional também pode ser descrito como aquele para o qual um conjunto finito de etapas de teste pode ser escrito para validar seu comportamento.

Representações de Requisitos

Bass et al., 2012, p. 64: Os requisitos para um sistema vêm em uma variedade de formas: **requisitos textuais, mockups, sistemas existentes, casos de uso, histórias de usuários** e mais.

Atributos de Qualidade

Definições

Bass et al., 2012, p. 63: Um atributo de qualidade (QA) é uma propriedade mensurável ou testável de um sistema que é usado para indicar quão bem o sistema satisfaz as

necessidades de seus stakeholders. Você pode pensar em um atributo de qualidade como medir a "bondade" de um produto ao longo de alguma dimensão de interesse para uma parte interessada.

NBR/ISO 9126, 2003, p. 17: Qualidade: Totalidade de características de uma entidade que lhe confere a capacidade de satisfazer as necessidades explícitas e implícitas.

NBR/ISO 9126, 2003, p. 17: Atributo: Propriedade mensurável, física ou abstrata, de uma entidade.

Kruchten, 2003, p. 237: (Qualidade): Característica de um artefato que satisfaz ou exerce um conjunto de requisitos definidos e aceitos, é avaliada usando medidas e critérios definidos e aceitos, e é produzida usando um processo definido e aceito.

IBA (BABOK), 2015, p. 16: Requisitos não funcionais ou requisitos de qualidade de serviço: Não se relacionam diretamente com o comportamento de funcionalidade da solução, mas sim descrever as condições sob as quais uma solução deve permanecer eficaz ou qualidades que uma solução deve ter.

Classificações

NBR/ISO 9126, 2003, p. 4: A qualidade do produto de software pode ser avaliada medindo-se os **atributos internos** (tipicamente medidas estáticas de produtos intermediários), os **atributos externos** (tipicamente pela medição do comportamento do código quando executado) ou os **atributos de qualidade em uso**. O objetivo é que o produto tenha o efeito requerido num contexto de uso particular.

NBR/ISO 9126, 2003, p. 17: Qualidade em uso : O quanto um produto, utilizado por usuários especificados, atende às necessidades desses usuários para que eles atinjam as metas especificadas com eficácia, produtividade e satisfação, em contextos de uso especificados.

NBR/ISO 9126, 2003, p. 17: Qualidade interna: Totalidade dos atributos de um produto que determinam sua capacidade para satisfazer necessidades explícitas e implícitas quando utilizado em condições especificadas.

NBR/ISO 9126, 2003, p. 16: Qualidade externa: O quanto um produto satisfaz necessidades explícitas e implícitas quando utilizado em condições especificadas.

Exemplos de Atributos de Qualidade

Funcionalidade - NBR/ISO 9126:2001-1, p. 8

Capacidade do produto de software de prover funções que atendam às necessidades explícitas e implícitas, quando o software estiver sendo utilizado sob condições especificadas.

- **Adequação**
Capacidade do produto de software de prover um conjunto apropriado de funções para tarefas e objetivos do usuário especificados.
- **Acurácia**
Capacidade do produto de software de prover, com o grau de precisão necessário, resultados ou efeitos corretos ou conforme acordados.
- **Interoperabilidade**

Capacidade do produto de software de interagir com um ou mais sistemas especificados.

- **Segurança de Acesso**

Capacidade do produto de software de proteger informações e dados, de forma que pessoas ou sistemas não autorizados não possam lê-los nem modificá-los e que não seja negado o acesso às pessoas ou sistemas autorizados.

Confiabilidade - ISO 9126, p. 9

Capacidade do produto de software de manter um nível de desempenho especificado, quando usado em condições especificadas.

- **Maturidade**

Capacidade do produto de software de evitar falhas decorrentes de defeitos no software.

- **Tolerância a falhas**

Capacidade do produto de software de manter um nível de desempenho especificado em casos de defeitos no software ou de violação de sua interface especificada.

- **Recuperabilidade**

Capacidade do produto de software de restabelecer seu nível de desempenho especificado e recuperar os dados diretamente afetados no caso de uma falha.

Usabilidade

Capacidade do produto de software de ser compreendido, aprendido, operado e atraente ao usuário, quando usado sob condições especificadas.

- **Inteligibilidade**

Capacidade do produto de software de possibilitar ao usuário compreender se o software é apropriado e como ele pode ser usado para tarefas e condições de uso específicas.

- **Apreensibilidade**

Capacidade do produto de software de possibilitar ao usuário aprender sua aplicação.

- **Operacionalidade**

Capacidade do produto de software de possibilitar ao usuário operá-lo e controlá-lo.

- **Atratividade - ISO 2696, p. 10**

Capacidade do produto de software de ser atraente ao usuário.

Eficiência - ISO 2696, p. 10

Capacidade do produto de software de apresentar desempenho apropriado, relativo à quantidade de recursos usados, sob condições especificadas.

- **Comportamento em relação ao tempo**

Capacidade do produto de software de fornecer tempos de resposta e de processamento, além de taxas de transferência, apropriados, quando o software executa suas funções, sob condições estabelecidas.

- **Utilização de recursos**

Capacidade do produto de software de usar tipos e quantidades apropriados de recursos, quando o software executa suas funções sob condições estabelecidas.

- **Conformidade relacionada à eficiência**

Capacidade do produto de software de estar de acordo com normas e convenções relacionadas à eficiência.

Manutenibilidade

Capacidade do produto de software ser modificado. As modificações podem incluir correções, melhorias ou adaptações do software devido a mudanças no ambiente e nos seus requisitos ou especificações funcionais.

- **Analisabilidade**
Capacidade do produto de software de permitir o diagnóstico de deficiências ou causas de falhas no software, ou a identificação de partes a serem modificadas.
- **Modificabilidade**
Capacidade do produto de software de permitir que uma modificação especificada seja implementada.
- **Estabilidade**
Capacidade do produto de software de evitar efeitos inesperados decorrentes de modificações no software.
- **Testabilidade**
Capacidade do produto de software de permitir que o software, quando modificado, seja validado.

Portabilidade

Capacidade do produto de software ser transferido de um ambiente para outro.

- **Adaptabilidade - ISO 2696, p. 11**
Capacidade do produto de software de ser adaptado para diferentes ambientes especificados, sem necessidade de aplicação de outras ações ou meios além daqueles fornecidos para essa finalidade pelo software considerado.
- **Capacidade para ser instalado - ISO 2696, p. 11**
Capacidade do produto de software para ser instalado em um ambiente especificado.
- **Coexistência - ISO 2696, pg 11**
Capacidade do produto de software de coexistir com outros produtos de software independentes, em um ambiente comum, compartilhando recursos comuns.
- **Capacidade para substituir - ISO 2696, p. 11**
Capacidade do produto de software de ser usado em substituição a outro produto de software especificado, com o mesmo propósito e no mesmo ambiente.

Restrições (Constraints)

Definições

IBA (BABOK), 2015, p. 444: Um fator de influência que não pode ser alterado e que coloca um limite ou restrição em uma possível solução ou opção de solução.

IBA (BABOK), 2015, p. 113: As restrições descrevem aspectos do estado atual, aspectos do estado futuro planejado que não podem ser alterados pela solução ou elementos obrigatórios do projeto.

Tipos de Restrições

Gorton, 2011, p. 6:

- **Restrições técnicas:** serão familiares a todos. Eles restringem as opções de design especificando certas tecnologias que o aplicativo deve usar. (...) Geralmente não são negociáveis.
- **Restrições de negócios:** também restringem as opções de design, mas para negócios, não por razões técnicas. (...) Na maioria das vezes, também esses são inegociáveis.

Exemplos de Restrições

IBA (BABOK), 2015, p. 113: As restrições podem refletir qualquer um dos seguintes: restrições orçamentárias, restrições de tempo, tecnologia, a infraestrutura, políticas, limites no número de recursos disponíveis, restrições com base nas habilidades da equipe e das partes interessadas, um requisito de que certas partes interessadas não sejam afetadas pela implementação da solução, conformidade com os regulamentos e qualquer outra restrição.

IBA (BABOK), 2015, p. 89: Restrições de negócios: estatutos regulatórios, obrigações contratuais e políticas de negócios que podem definir prioridades.

Bass et al., 2012, p. 64: Exemplos incluem:

- o requisito de usar uma determinada linguagem de programação ou de reutilizar um certo módulo existente,
- ou um decreto de gestão para tornar o seu sistema orientado a serviço.

Essas escolhas são indiscutivelmente da competência do arquiteto, mas de um fator externo (como não ser capaz de treinar a equipe em um novo idioma, ou ter um acordo comercial com um fornecedor de software, ou empurrar negócios objetivos de interoperabilidade de serviços).

Riscos (Risks)

Kruchten, 2003, p. 237: Preocupação contínua ou intermitente que tem uma probabilidade significativa de afetar adversamente o sucesso de fatos principais.

IBA (BABOK), 2015, p. 16: Risco é o efeito da incerteza sobre o valor de uma mudança, uma solução ou o empreendimento. Os analistas de negócios colaboram com outras partes interessadas para identificar, avaliar e priorizar riscos e lidar com esses riscos, alterando a probabilidade de as condições ou eventos que levam à incerteza: mitigando as consequências, removendo a fonte do risco, evitando o risco completamente por decidir não iniciar ou continuar com uma atividade que leve ao risco, compartilhando o risco com outras partes, ou aceitar ou mesmo aumentar o risco de lidar com uma oportunidade.

IEEE (SWEBoK), 2014, p. 138: Risco e incerteza são conceitos relacionados, mas distintos. A incerteza resulta da falta de informação. O risco é caracterizado pela probabilidade de um evento resultar em um impacto negativo mais uma caracterização do impacto negativo em um projeto. O risco geralmente é o resultado da incerteza. O inverso do risco é a oportunidade, que se caracteriza pela probabilidade de ocorrer um evento com um resultado positivo.

Tipos de Risco

Kruchten, 2003, p. 99: Podemos qualificar os riscos mais adiante como diretos e indiretos:

- Risco Direto: um risco sobre o qual o projeto tem um alto grau de controle.
- Risco Indireto: um risco sobre o qual o projeto tem pouco ou nenhum controle.

Propriedades

Kruchten, 2003, p. 99: [...] Também podemos acrescentar dois atributos importantes:

- A probabilidade de ocorrência
- O impacto no projeto (severidade)

Estes dois atributos podem ser combinados frequentemente num único indicador de **magnitude de risco**, e cindo valores discretos são suficientes: alto, significativo, moderado, secundário e baixo.

Kruchten, 2003, p. 100: Para cada risco percebido, você tem que decidir com antecedência o que vai fazer.

Três rotas principais são possíveis:

- **Fuga do risco:** Reorganizar o projeto de forma que não possa ser afetado pelo risco.
- **Transferência de risco:** Reorganizar o projeto de forma que alguém ou qualquer outra coisa aguarde o risco. (O cliente, o vendedor, o banco ou outro elemento)
- **Aceitação do risco:** Decidir viver com o risco como uma contingência. Monitorar os sintomas de risco e determinar o que fazer se o risco se materializar.

Ao aceitar um risco, você deveria fazer duas coisas:

- **Moderar o risco:** Tornar, imediatamente, medidas proativas para reduzir a probabilidade ou o impacto do risco.
- **Definir um plano de contingência:** Determinar o curso da ação para perceber se o risco chega a ser um problema atual; em outras palavras, criar um “plano B”.

Decisões (Decision)

Definições

Jansen e Bosch, 2005, p2: Uma descrição do conjunto de adições, subtrações e modificações arquiteturais na arquitetura de software, a lógica e as regras de projeto, restrições de projeto e requisitos adicionais que (parcialmente) realizam um ou mais requisitos em uma dada arquitetura.

IEEE (SWEBoK), 2014, p. 54 [14, 15]: O projeto arquitetônico é um processo criativo. Durante o processo de design, os designers de software precisam tomar uma série de decisões fundamentais que afetam profundamente o software e o processo de desenvolvimento. É útil pensar no processo de projeto arquitetônico a partir de uma perspectiva de tomada de decisão, e não de uma perspectiva de atividade. Muitas vezes, **o impacto nos atributos de qualidade e as compensações (tradeoffs) entre os atributos de qualidade concorrentes são a base para as decisões de projeto.**

Relacionamentos

#	Classe	Relacionamento	Classe
14	Atributo de Qualidade	concorre	Atributo de Qualidade

Propriedades

Jansen e Bosch, 2005, p2:

- **Justificativa:** As razões por trás de uma decisão de projeto arquitetônico são a lógica de uma decisão de projeto arquitetônico. Ele descreve por que uma mudança é feita na arquitetura de software.

- As **regras de projeto** e as **restrições de projeto** são prescritas para outras decisões de projeto. As regras são diretrizes obrigatórias, considerando que as restrições limitam o design a permanecer sólido.
- **Restrições de projeto:** As restrições de projeto descrevem o lado oposto das regras de projeto. Eles descrevem o que não é permitido no futuro do design, ou seja, eles proíbem certos comportamentos.
- **Requisitos adicionais:** Uma decisão de projeto pode resultar em requisitos adicionais a serem atendidos pela arquitetura. Esses novos requisitos precisam ser abordados por decisões de projeto adicionais.

Justificativas (**Rationales**)

Definições

Jansen e Bosch, 2005, p 2: As razões por trás de uma decisão de projeto arquitetônico são a lógica de uma decisão de projeto arquitetônico. Ele descreve por que uma mudança é feita na arquitetura de software.

Clements, 2010, p. 25: A justificativa é uma explicação do raciocínio que está por trás de uma decisão arquitetônica.

ISO 42010, 2011, p. 7: [15,16] Registra a explicação, justificativa ou raciocínio sobre as decisões de arquitetura que foram tomadas. **A justificativa para uma decisão pode incluir a base para uma decisão, alternativas e tradeoffs consideradas**, consequências potenciais da decisão e citações de fontes de informações adicionais.

Relacionamentos

#	Classe	Relacionamento	Classe
15	Justificativa	justifica	Decisão
16	Justificativa	considera	Tradeoff

Compensações (**Tradeoffs**)

Definições

IBA (BABOK), 2015, p. 264: Os trade-offs tornam-se relevantes sempre que um problema de decisão envolve objetivos múltiplos e possivelmente conflitantes.

Gorton, 2011, p. 95: Requisitos conflitantes podem até ter a mesma prioridade. É então responsabilidade da solução considerar os tradeoffs apropriados e tentar encontrar aquela “linha tênue” que satisfaça adequadamente ambos os requisitos sem incomodar ninguém ou ter consequências indesejáveis importantes na aplicação.

IEEE (SWEBOK), 2014, p. 54: [17, 18] O impacto sobre os atributos de qualidade e compensações (tradeoffs) entre os atributos de qualidade concorrentes são a base para as decisões de design.

Gorton, 2011, p. 38: [18] Compreender as tradeoffs entre os requisitos de atributos de qualidade e projetar uma solução que faça concessões sensatas é uma das partes mais

difíceis da função do arquiteto. Simplesmente não é possível satisfazer totalmente todos os requisitos concorrentes. É trabalho do arquiteto descobrir essas tensões, torná-las explícitas para as partes interessadas do sistema, priorizar conforme necessário e documentar explicitamente as decisões de design.

Relacionamentos

#	Classe	Relacionamento	Classe
17	Compensação (Tradeoff)	compensa	Atributo de Qualidade (Quality Attribute)
18	Compensação (Tradeoff)	baseia	Decisão (Decision)

Táticas Arquiteturais (Architectural Tactics)

Definições

Bass et al., 2012, p. 70-71: [19, 20] Os requisitos dos atributos de qualidade especificam as respostas do sistema que, com um pouco de sorte e uma dose de bom planejamento, concretizam os objetivos do negócio. Agora nos voltamos para as técnicas que um arquiteto pode usar para alcançar os atributos de qualidade necessários. Chamamos essas técnicas de táticas arquitetônicas. **Uma tática é uma decisão de design** que influencia a obtenção de uma resposta de atributo de qualidade – as táticas afetam diretamente a resposta do sistema a algum estímulo. As táticas conferem portabilidade a um projeto, alto desempenho a outro e integrabilidade a um terceiro. **O foco de uma tática está em uma única resposta de atributo de qualidade.** Dentro de uma tática, não há consideração de tradeoffs. Os tradeoffs devem ser explicitamente considerados e controlados pelo designer. A esse respeito, **as táticas diferem dos padrões de arquitetura**, onde as compensações são incorporadas ao padrão.

Bass et al., 2012, p. 77: [19] Uma tática de arquitetura é uma decisão de projeto que afeta uma resposta de atributo de qualidade. O foco de uma tática está em uma única resposta de atributo de qualidade. Padrões arquiteturais podem ser vistos como “pacotes” de táticas.

Bass et al., 2012, p. 204: [...] Estes são mais simples do que os padrões. As táticas geralmente usam apenas uma única estrutura ou mecanismo computacional e destinam-se a abordar uma única força arquitetônica. Por esse motivo, eles dão um controle mais preciso a um arquiteto ao tomar decisões de projeto do que os padrões, que normalmente combinam várias decisões de projeto em um pacote. As táticas são os “blocos de construção” do design, a partir dos quais os padrões arquiteturais são criados. Táticas são átomos e padrões são moléculas. A maioria dos padrões consiste em (são construídos a partir de) várias táticas diferentes. Por isso dizemos que os padrões empacotam táticas.

Relacionamentos

#	Classe	Relacionamento	Classe
19	Tática Arquitetural (Architectural Tactic)	é	Decisão (Decision)

20	Tática Arquitetural (Architectural Tactic)	responde	Atributo de Qualidade (Quality Attribute)
----	--	----------	---

Padrões Arquiteturais (**Architectural Patterns**)

Definições

Kruchten, 2003, p. 237: Descrição de uma solução arquetípica para um problema de projeto periódico que reflete a experiência de projeto demonstrada.

Clements, 2010, p. 33: Um **padrão de arquitetura** "expressa um esquema organizacional estrutural fundamental para sistemas de software" (**Buschmann et al. 1996, p. 12**). É, acima de tudo, um padrão que, no contexto da arquitetura "descreve um problema de design recorrente particular que surge em contextos de design específicos e apresenta um esquema genérico bem-comprovado para sua solução. O esquema de solução é especificado pela descrição de seus componentes constituintes, suas responsabilidades e relações, e as formas pelas quais eles colaboram " (Buschmann et al. 1996, p. 8) (...) Uma parte essencial de um padrão de arquitetura é seu foco no problema e no contexto, bem como em como resolver o problema nesse contexto. (...) Um estilo de arquitetura foca na abordagem da arquitetura, com orientações mais leves sobre quando um determinado estilo pode ou não ser útil.

Bass et al., 2012, p. 18-19: Em alguns casos, os elementos arquiteturais são compostos de maneiras que resolvem problemas. As composições foram consideradas úteis ao longo do tempo, e ao longo de muitos diferentes domínios, e por isso foram documentadas e divulgadas. Essas composições de elementos arquiteturais, chamados de **padrões arquiteturais**, fornecem estratégias empacotadas para resolver alguns dos problemas enfrentados por um sistema. Um padrão de arquitetura delinea os tipos de elementos e suas formas de interação usadas na solução do problema. Os padrões podem ser caracterizados de acordo com o tipo de elementos arquiteturais que utilizam.

Bass et al., 2012, p. 203: [22] Um padrão de arquitetura:

- é um **pacote de decisões de design** que é encontrado repetidamente na prática,
- tem propriedades conhecidas que permitem a reutilização, e
- descreve uma classe de arquitetura

Bass et al., 2012, p. 36: [23] Padrões arquiteturais orientam o arquiteto e focam o arquiteto nos **atributos de qualidade** de interesse em grande parte, restringindo o vocabulário das alternativas de projeto a um número relativamente pequeno. As propriedades do projeto de software decorrem da escolha de um padrão de arquitetura. Esses padrões que são mais desejáveis para um problema específico devem melhorar a implementação da solução de design resultante, talvez facilitando a arbitragem de restrições de design conflitantes, aumentando a percepção de contextos de design mal compreendidos ou ajudando a revelar inconsistências nos requisitos.

Propriedades

Bass et al., 2012, p. 204: Um padrão de arquitetura estabelece uma relação entre:

- **Um contexto.** Uma situação recorrente e comum no mundo que dá origem a um problema.

- **Um problema.** O problema, adequadamente generalizado, que surge no contexto dado. A descrição do padrão descreve o problema e suas variantes e descreve quaisquer forças complementares ou opostas. A descrição do problema geralmente inclui atributos de qualidade que devem ser atendidos.
- **Uma solução.** Uma resolução arquitetônica bem-sucedida para o problema, adequadamente abstraída. A solução descreve as estruturas arquitetônicas que resolvem o problema, incluindo como equilibrar as muitas forças em ação. A solução descreverá as responsabilidades e os relacionamentos estáticos entre os elementos (usando uma estrutura de módulo) ou descreverá o comportamento de tempo de execução e a interação entre os elementos (definindo um componente e conector ou estrutura de alocação). A solução para um padrão é determinada e descrita por:
 - **Um conjunto de tipos de elementos** (por exemplo, repositórios de dados, processos e objetos).
 - **Um conjunto de mecanismos ou conectores de interação** (por exemplo, chamadas de método, eventos ou barramento de mensagens)
 - **Um layout topológico dos componentes.**
 - **Um conjunto de restrições** semânticas que cobrem topologia, comportamento do elemento e mecanismos de interação.

A descrição da solução também deve deixar claro quais atributos de qualidade são fornecidos pelas configurações estáticas e de tempo de execução dos elementos. Este formulário {contexto, problema, solução} constitui um modelo para documentar um padrão.

Relacionamentos

#	Classe	Relacionamento	Classe
21	Padrão Arquitetural (Architectural Pattern)	é	Decisão (Decision)
22	Padrão Arquitetural (Architectural Pattern)	foca	Atributo de Qualidade (Quality Attribute)

Exemplos de Padrões

Bass et al., 2012, p. 205, 210, 212, 215, 217, 220, 222, 226, 230, 232, 235: Camadas, Broker, Model View Controller, Pipe-And-Filter, Client-Server, Peer-to-Peer, Service-Oriented Architecture, Publish-Subscribe, Shared-Data, Map-Reduce, Multi-tier.

Descrição Arquitetural (**Architectural Description**)

Definições

ISO 42010, 2011, p. 2, 4: Produto de trabalho usado para expressar uma arquitetura. [...] Descrições de arquitetura são produtos de trabalho de arquitetura de sistemas e software.

ISO 42010, 2011, p. 12: [23, 24] Uma **descrição da arquitetura deve identificar as partes interessadas do sistema com preocupações consideradas fundamentais** para a arquitetura do sistema de interesse. Uma **descrição da arquitetura deve identificar as preocupações consideradas fundamentais** para a arquitetura do sistema de interesse.

Uma **descrição da arquitetura deve associar cada preocupação identificada com as partes interessadas identificadas com essa preocupação.**

ISO 42010, 2011, p. 15: [25] Uma descrição de arquitetura deve incluir uma justificativa para cada ponto de vista de arquitetura incluído para uso [...] em termos de seus stakeholders, preocupações, tipos de modelo, notações e métodos. Uma **descrição de arquitetura deve incluir justificativa para cada decisão** considerada uma decisão de arquitetura chave. Uma **descrição de arquitetura deve registrar as decisões de arquitetura** consideradas fundamentais para a arquitetura do sistema de interesse.

ISO 42010, 2011, p. 13: [26] Uma **descrição de arquitetura deve incluir cada ponto de vista de arquitetura** usado nela. Cada ponto de vista de arquitetura incluído deve ser especificado [...]. Cada preocupação identificada [...] deve ser enquadrada por pelo menos um ponto de vista.

Uma **descrição de arquitetura deve incluir exatamente uma visão de arquitetura para cada ponto de vista de arquitetura usado.** Cada visão de arquitetura deve aderir às convenções de seu ponto de vista de arquitetura governante.

ISO 42010, 2011, p. 6: [27] Uma **descrição de arquitetura inclui uma ou mais visões de arquitetura.** Uma visão de arquitetura (ou simplesmente visão) aborda uma ou mais das preocupações dos stakeholders do sistema. Uma visão de arquitetura expressa a arquitetura do sistema de interesse de acordo com um ponto de vista de arquitetura (ou simplesmente, ponto de vista). Há dois aspectos em um ponto de vista: as preocupações que ele estabelece para as partes interessadas e as convenções que estabelece sobre os pontos de vista.

Relações

#	Classe	Relacionamento	Classe
23	Descrição Arquitetural (Architectural Description)	identifica	Partes Interessadas (Stakeholders)
24	Descrição Arquitetural (Architectural Description)	identifica	Preocupações (Concerns)
25	Descrição Arquitetural (Architectural Description)	inclui	Justificativas (Rationales)
26	Descrição Arquitetural (Architectural Description)	inclui	Pontos de Vista Arquitetural (Architectural Viewpoints)
27	Descrição Arquitetural (Architectural Description)	inclui	Visões Arquiteturais (Architectural Views)

Pontos de Vista Arquitetural (Architectural Viewpoints)

Definições

ISO 42010:2011, p. 2: Produto de trabalho que estabelece as convenções para a construção, interpretação e uso de visões arquitetônicas para enquadrar questões específicas do sistema.

ISO 42010, 2011, p. 6: [28] Uma visão de arquitetura expressa a arquitetura do sistema de interesse de acordo com um ponto de vista de arquitetura (ou simplesmente, ponto de vista). Há dois aspectos em um **ponto de vista: as preocupações que ele estabelece para as partes interessadas e as convenções que estabelece sobre os pontos de vista. Um ponto de vista da arquitetura enquadra uma ou mais preocupações. Uma preocupação pode ser enquadrada por mais de um ponto de vista. Uma visão é governada por seu ponto de vista: o ponto de vista estabelece as convenções para construir, interpretar e analisar a visão para abordar as preocupações enquadradas por esse ponto de vista.** As convenções de ponto de vista podem incluir linguagens, notações, tipos de modelo, regras de projeto e/ou métodos de modelagem, técnicas de análise e outras operações em visões.

ISO 42010, p. 13: [28] Uma descrição da arquitetura deve incluir cada ponto de vista da arquitetura usada nela. [...] **Cada preocupação identificada [...] deve ser enquadrada por pelo menos um ponto de vista.**

Kruchten, 2003, p. 71: [29, 30] Uma visão arquitetural é uma descrição simplificada (uma abstração) de um sistema, de uma perspectiva particular ou ponto de desempate, cobrindo preocupações particulares e omitindo entidades não pertencentes a esta perspectiva. Para cada visão, precisamos identificar claramente o seguinte:

- **O ponto de vista - as preocupações e os interessados a serem endereçados.**
- Os elementos que serão capturados e representados na visão, junto com as relações deles.
- Os princípios organizacionais usados para estruturar a visão
- Os elementos desta visão são relacionados aos de outras visões
- O melhor processo para usar ao criar esta visão

Relações

#	Classe	Relacionamento	Classe
28	Ponto de Vista Arquitetural (Architectural ViewPoint)	enquadra/endereça	Preocupação
29	Ponto de Vista Arquitetural (Architectural ViewPoint)	endereça	Stakeholder

Exemplos de viewpoints

4+1 architectural view model (Kruchten, 2003, p. 72-73)

- **Lógica**

Kruchten, 2003, p. 72: Endereça os requisitos funcionais do sistema, em outras palavras, o que o sistema deveria fazer para seus usuários finais. É uma abstração do modelo de projeto e identifica pacotes de projeto principais, subsistemas e classes.

- **Implementação**

Kruchten, 2003, p. 73: Descreve a organização de módulos estáticos de software no ambiente de desenvolvimento, em termos de empacotar e estender em camadas e de gerenciamento de configuração. Endereça os assuntos da facilidade de desenvolvimento de ativos de software, reutilização, subcontrato e componentes fora de uso.

- **Processo**

Kruchten, 2003, p. 73: Endereça os aspectos simultâneos do sistema na execução - tarefas, linhas ou processos, como também suas interações. Focaliza assuntos como concorrência e paralelismo, inicialização de sistema e paralisação completa, tempo de resposta, processamento e isolamento de funções e falhas. Está relacionado a escalabilidade.

- **Implantação**

Kruchten, 2003, p. 73: Esta visão faz um papel especial com respeito à arquitetura. Contém alguns cenários-chave ou casos de uso. Inicialmente, são usados para dirigir a descoberta do projeto da arquitetura no início e nas fases de elaboração, mas depois serão usados para validar as diferentes visões. Estes poucos cenários agem para ilustrar no documento de arquitetura de software como as outras visões funcionam.

- **Casos de Uso**

Kruchten, 2003, p. 73: Contém alguns cenários chave ou casos de uso. Inicialmente são usados para dirigir a descoberta e projeto da arquitetura no início e fases de elaboração, mas depois serão usados para validar as diferentes visões. Estes poucos cenários agem para ilustrar no documento de arquitetura de software como as outras visões funcionam.

Views and Beyond

Módulos

Clements, 2010, p. 55: Hoje, a maneira como o software de um sistema é decomposto em unidades gerenciáveis continua sendo uma das formas importantes de estrutura do sistema. No mínimo, ele determina como o código-fonte de um sistema é particionado em partes separáveis, que tipos de suposições cada parte pode fazer sobre os serviços fornecidos por outras partes e como essas partes são agregadas em conjuntos maiores. A escolha da modularização geralmente determina como as mudanças em uma parte de um sistema podem afetar outras partes e, portanto, a capacidade de um sistema de suportar modificabilidade, portabilidade e reutilização.

Componente & Conector

Clements, 2010, p. 123: Uma visão C&C mostra elementos que têm presença de tempo de execução como: processos, objetos, clientes, servidores e armazenamentos de dados. Esses elementos são chamados de componentes. Além disso, as visões de componente e conector incluem elementos como caminhos de interação, como links e protocolos de comunicação, fluxos de informações e acesso a armazenamento compartilhado. Essas interações são representadas como conectores nas visões C&C.

Alocação

Clements, 2010, p. 123: As visões de alocação apresentam um mapeamento entre elementos de software (...) e elementos não de software no ambiente de software. Começamos considerando a forma mais geral de mapeamento entre a arquitetura de

software e seu ambiente. (...) Os elementos de estilos de alocação são elementos de software mais elementos ambientais. Exemplos de elementos ambientais são um processador, um conjunto de discos, um arquivo ou uma pasta ou um grupo de desenvolvedores. Os elementos de software vêm de módulos ou estilo C&C.

Outros

Bass et al., 2012, pgs. 342-343: Outro tipo de visão, que chamamos de visão de qualidade, pode ser adaptado para partes interessadas específicas ou para tratar de preocupações específicas. Essas visões de qualidade são formadas extraíndo as partes relevantes das vistas estruturais e empacotando-as juntas. Aqui estão cinco exemplos:

- Uma visão de segurança pode mostrar todas as medidas arquitetônicas tomadas para fornecer segurança. Mostraria os componentes que possuem algum direito de acesso ou responsabilidade, como esses componentes se comunicam, quaisquer repositórios de dados para informações de segurança e repositórios que são de interesse de segurança. O as informações de contexto da visão mostrariam outras medidas de segurança (como segurança física) no ambiente do sistema. A parte de comportamento de uma visão de segurança mostraria a operação dos protocolos de segurança e onde e como os humanos interagem com os elementos de segurança. Também capturaria como o sistema responderia a ameaças e vulnerabilidades específicas.
- Uma visão de comunicações pode ser especialmente útil para sistemas que são globalmente disperso e heterogêneo. Essa visão mostraria todos os canais de componente para componente, os vários canais de rede, valores de parâmetros de qualidade de serviço e áreas de simultaneidade. Essa visão pode ser usado para analisar certos tipos de desempenho e confiabilidade (como deadlock ou detecção de condição de corrida). A parte de comportamento desta visão pode mostrar (por exemplo) como a largura de banda da rede é alocada dinamicamente.
- Uma visão de exceção ou tratamento de erros pode ajudar a iluminar e chamar a atenção para os mecanismos de relatório e resolução de erros. Tal visão mostraria como os componentes detectam, relatam e resolvem falhas ou erros. Isto ajudaria identificar as fontes de erros e ações corretivas apropriadas para cada um. A análise da causa raiz nesses casos pode ser facilitada por essa visão.
- Uma visão de confiabilidade seria aquela em que mecanismos de confiabilidade, como replicação e alternância são modelados. Também retrataria problemas de tempo e integridade da transação.
- Uma visão de desempenho incluiria os aspectos úteis da arquitetura para inferir o desempenho do sistema. Essa visão pode mostrar a rede modelos de tráfego, latências máximas para operações e assim por diante.

Visões Arquiteturais (**Architectural Views**)

Definições

ISO 42010, 2011, p. 2: Produto de trabalho que expressa a arquitetura de um sistema a partir da perspectiva de interesses específicos do sistema.

Clements, 2010, p 22: Uma visão é uma representação de um conjunto de elementos do sistema e os relacionamentos associados a eles.

Kruchten, 2003, p. 71: Uma visão arquitetônica é uma descrição simplificada (uma abstração) de um sistema, de uma perspectiva particular ou ponto de desempate, cobrindo preocupações particulares e omitindo entidades não pertencentes a esta perspectiva.

Kruchten, 2003, p. 74: Visões arquiteturais estão como fatias cortadas pelos vários modelos que iluminam só seus elementos importantes e significativos. (...) A utilidade das visões para um sistema é captada em um artefato importante: a descrição da arquitetura de software.

Kruchten, 2003, p. 238: Visão arquitetônica de sistema de uma determinada perspectiva; concentra-se principalmente na estrutura, modularidade, componentes essenciais e fluxos de controle principais.

Clements, 2010, p 22: [30] Diferentes visões também expõem diferentes atributos de qualidade em diferentes graus. Portanto, os atributos de qualidade que mais preocupam você e as outras partes interessadas no desenvolvimento do sistema afetarão a escolha de quais visões documentar.

Bass et al., 2012, pgs. 341-343: [31] A documentação das decisões durante o processo de design (algo que recomendamos fortemente) produz visões, que são o coração de um documento de arquitetura. (...) No momento em que estiver pronto para lançar um documento de arquitetura, você provavelmente terá uma coleção bastante bem elaborada de visões de arquitetura. Em algum momento, você precisará decidir o que concluir, com quantos detalhes e o que incluir em uma determinada versão. Você também precisará decidir quais visões podem ser combinadas de forma útil com outras, de modo a reduzir o número total de visões no documento e revelar relações importantes entre as visões.

Kruchten, 2003, p. 238: Visão da arquitetura de sistema de uma determinada perspectiva; concentra-se principalmente na estrutura, modularidade, componentes essenciais e fluxos de controle principais; (...) Descrição simplificada (uma abstração) de um modelo visto de determinada perspectiva ou ponto de desempate, omitindo entidades que não são pertinentes a esta perspectiva.

ISO 42010, 2011, p. 6: [32, 33, 34] Uma visão de arquitetura (ou simplesmente, visão) aborda uma ou mais das preocupações mantidas pelas partes interessadas do sistema, expressa a arquitetura do sistema de interesse de acordo com um ponto de vista de arquitetura (ou simplesmente, ponto de vista). Um ponto de vista da arquitetura enquadra uma ou mais preocupações. Uma preocupação pode ser enquadrada por mais de um ponto de vista. **Uma visão é governada por seu ponto de vista:** o ponto de vista estabelece as convenções para construir, interpretar e analisar a visão para abordar as preocupações emolduradas por esse ponto de vista. As convenções de ponto de vista podem incluir linguagens, notações, tipos de modelo, regras de design e / ou métodos de modelagem, técnicas de análise e outras operações em vistas.

Relações

#	Classe	Relacionamento	Classe
30	Visão Arquitetural (Architectural Views)	expõe	Atributo de Qualidade (Quality Attribute)
31	Decisão Arquitetural (Architectural Decision)	produz	Visão Arquitetural (Architectural View)
32	Visão Arquitetural (Architectural	aborda	Preocupação (Concern)

	View)		
33	Ponto de Vista Arquitetural (Architectural ViewPoint)	enquadra	Visão Arquitetural (Architectural Views)
34	Visão Arquitetural (Architectural View)	governado	Ponto de Vista Arquitetural (Architectural ViewPoint)

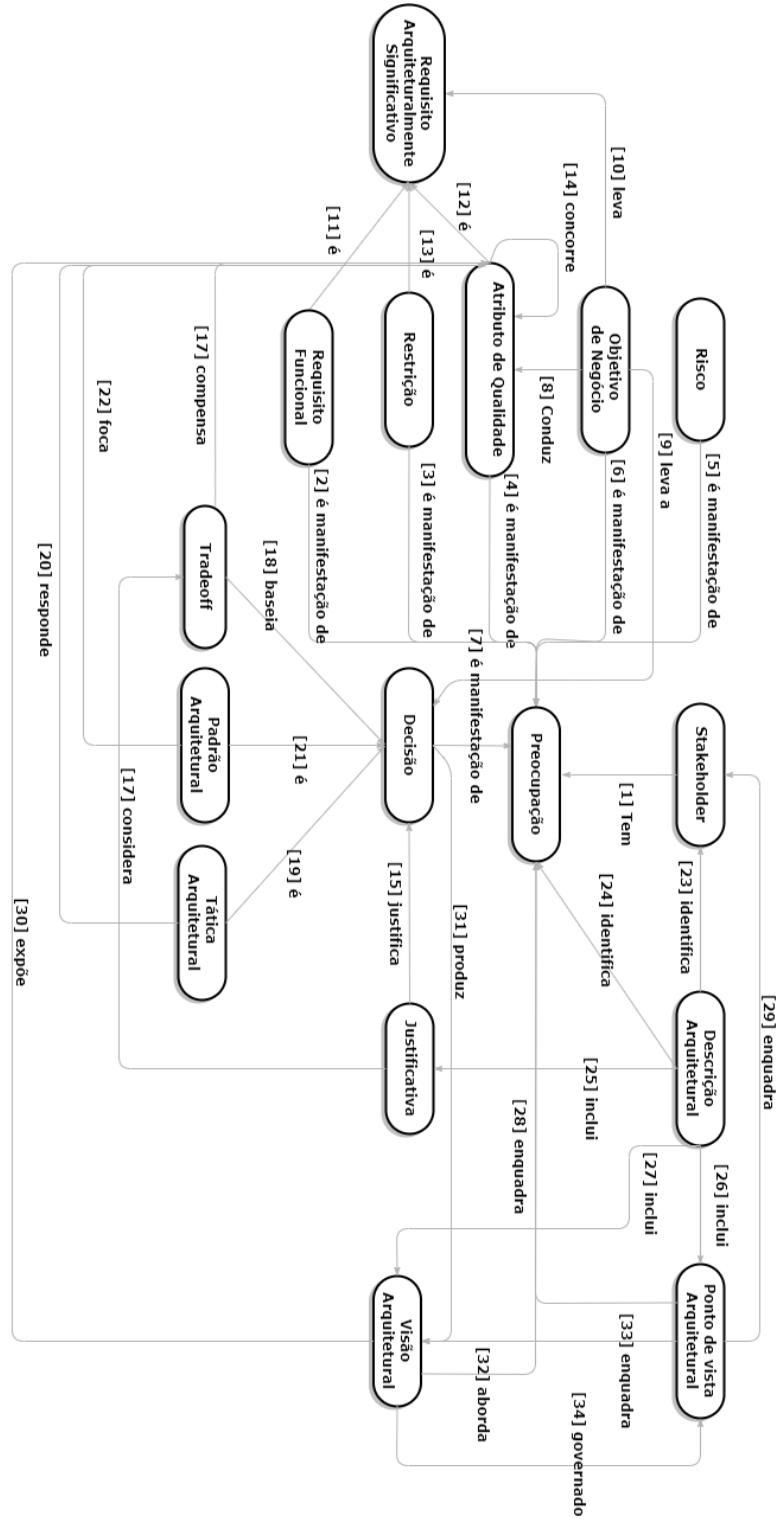
Nível de formalidade

Clements, 2010, p 53: As notações para documentar visões diferem consideravelmente em seu grau de **formalidade**. Grosso modo, existem três categorias principais de notação:

1. Notações **informais**: as visões são representadas (frequentemente graficamente) usando diagramas de propósito geral e ferramentas de edição e convenções visuais escolhidas para o sistema em questão. A semântica da descrição é caracterizada em linguagem natural e não pode ser analisada formalmente.
2. Notações **semiformais**: as visões são expressas em uma notação padronizada que prescreve elementos gráficos e regras de construção, mas não fornece um tratamento semântico completo do significado desses elementos. A análise rudimentar pode ser aplicada para determinar se uma descrição satisfaz as propriedades sintáticas. A Unified Modeling Language (UML) é uma notação semiformal nesse sentido.
3. Notações **formais**: as visões são descritas em uma notação que tem uma semântica precisa (geralmente baseada em matemática). A análise formal da sintaxe e da semântica é possível. Há uma variedade de notações formais para arquitetura de software disponíveis, embora nenhuma delas possa ser considerada amplamente usada, geralmente chamadas de linguagens de descrição de arquitetura (ADLs), elas normalmente fornecem vocabulário gráfico e uma semântica subjacente para representação de arquitetura.

APÊNDICE B – MAPA CONCEITUAL DE ARQUITETURA DE SOFTWARE

Figura 7 – Mapa Conceitual de Arquitetura de Software - Em detalhes



Fonte: elaborado pelo autor.

APÊNDICE C – SOFTWARE ARCHITECTURE CANVAS

Figura 8 – Canvas de Arquitetura de Software.

Software Architecture Canvas		Business Case	Requisitos Arquiteturalmente Significativos	Decisões	Descrição
<p>Stakeholders e Preocupações</p> <p>Quem são os stakeholders do projeto? Quem são os stakeholders do produto? Quais são as preocupações dos stakeholders com o projeto? Quais são as prioridades dos stakeholders com o projeto?</p> <p>Ex. de Stakeholders: Cliente, Usuário, Parceiro, Fornecedor, Regulador, Investidor, etc. Ex. de Preocupações: Segurança, Performance, Escalabilidade, Disponibilidade, etc.</p>	<p>Restrições</p> <p>Quais são as restrições técnicas? Quais são as restrições de negócio? Quais são as restrições de recursos? Quais são as restrições de tempo?</p> <p>Ex. de Restrições: Tecnologia, Recursos Humanos, Orçamento, etc.</p>	<p>Objetivos de Negócios</p> <p>Quais são os objetivos de negócios do projeto? Quais são os objetivos de negócios do produto? Quais são os objetivos de negócios do sistema?</p> <p>Ex. de Objetivos: Aumentar a receita, Reduzir os custos, Melhorar a experiência do usuário, etc.</p>	<p>Requisitos Funcionais</p> <p>Quais são os requisitos funcionais do sistema? Quais são os requisitos funcionais do produto? Quais são os requisitos funcionais do usuário?</p> <p>Ex. de Requisitos: Autenticação de usuário, Gestão de perfil, etc.</p>	<p>Táticas Arquiteturais</p> <p>Quais são as táticas arquiteturais do sistema? Quais são as táticas arquiteturais do produto? Quais são as táticas arquiteturais do usuário?</p> <p>Ex. de Táticas: Microserviços, Cloud Native, etc.</p>	<p>Pontos de Vista e Visões Arquiteturais</p> <p>Quais são os pontos de vista do sistema? Quais são os pontos de vista do produto? Quais são os pontos de vista do usuário?</p> <p>Ex. de Pontos de Vista: Arquitetura de Referência, Arquitetura de Referência, etc.</p>
<p>Riscos</p> <p>Quais são os riscos do projeto? Quais são os riscos do produto? Quais são os riscos do sistema?</p> <p>Ex. de Riscos: Segurança, Performance, Escalabilidade, Disponibilidade, etc.</p>	<p>Atributos de Qualidade</p> <p>Quais são os atributos de qualidade do sistema? Quais são os atributos de qualidade do produto? Quais são os atributos de qualidade do usuário?</p> <p>Ex. de Atributos: Segurança, Performance, Escalabilidade, Disponibilidade, etc.</p>	<p>Padrões Arquiteturais</p> <p>Quais são os padrões arquiteturais do sistema? Quais são os padrões arquiteturais do produto? Quais são os padrões arquiteturais do usuário?</p> <p>Ex. de Padrões: Microserviços, Cloud Native, etc.</p>			

Fonte: elaborado pelo autor.

APÊNDICE D – CARTA PROPOSTA DO ESTUDO DE CASO

Esta é a carta-proposta apresentada aos participantes no início do estudo de caso, na execução da etapa 4 dos procedimentos metodológicos deste trabalho.

PaperBank

Duckburg, 1 de Fevereiro de 2022

Prezados,

Nós do PaperBank estamos iniciando um processo de transformação digital. Somos um banco tradicional de Duckburg tendo iniciado nossas atividades em 1950, oferecendo serviços para pessoas físicas em: contas (corrente, poupança e aposentadoria), cartões de crédito e débito, transferências bancárias, empréstimos, financiamentos e consórcios.

Nosso maior objetivo é nos tornar um banco 100% digital até fevereiro de 2023, buscando a transformação digital de nossas soluções financeiras e rebranding passando a se chamar BitBank. Com isso, esperamos alcançar novos públicos (Classes A, B e D; Faixas etárias: 18-49 anos) mantendo nosso público atual (Classe B e C, Faixas etárias: 50-80+), extinguir nossos processos físicos e reduzir os custos de operação deste processos.

Atualmente, nossos processos internos (RH, Folha de Pagamento, Aquisição, Jurídico e etc) estão sendo executados via ERP SAP, então não há a necessidade de atuar nessas áreas ficando apenas nas operações bancárias/financeiras.

Nesta transformação digital buscamos o que há de mais moderno e seguro no mercado bancário, como: Armazenamento seguro de documentos/contratos digitalizados, auditoria, biometria, verificação de documentos, PIX, antifraude, OpenBanking, operações em tempo real, e outros. Também será necessário propor soluções inteligentes para as integrações com os vagarosos sistemas da receita federal, INSS, COAF e outros sistemas do governo. Pretendemos encerrar as atividades em todas as nossas agências físicas, passando a utilizar o serviço Banco24Horas de caixas eletrônicos.

Atualmente, temos um orçamento de R\$ 720.000,00 (setecentos e vinte mil reais) para o desenvolvimento da plataforma, divididos igualmente pelos meses até a nossa data limite de entrada no mercado, R\$ 280.000,00 (duzentos e oitenta mil reais) para quaisquer aquisições, e um limite de custos de operações de R\$ 50.000,00 (cinquenta mil reais) mensais.

Esperamos receber e avaliar o projeto de arquitetura de software/sistema proposto, mesmo que preliminar, com nossa equipe de TI.

Atenciosamente,

Scrooge McDuck

CEO & Founder PaperBank/BitBank

APÊNDICE E – QUESTIONÁRIO DO ESTUDO DE CASO

Este apêndice contém as perguntas de todos os formulários utilizados no *Google Forms*, para um documento único, as perguntas fechadas estão classificadas na escala de Likert, como mencionado no Capítulo 4.

Este questionário tem como objetivo colher informações sobre o uso da ferramenta Canvas de Arquitetura de Software. Os dados coletados aqui serão utilizados apenas para propósito/em contexto de pesquisa científica. nenhuma informação obtida aqui possibilita sua identificação direta.

AGRADECEMOS SUA COLABORAÇÃO.

Classificação de Perfil

<P-001> - Grau de Escolaridade

- Superior Incompleto
- Superior Completo
- Especialização
- Mestrado
- Doutorado
- Outro: _____

<P-002> - Quantos projetos em desenvolvimento de software você já trabalhou:

- Nenhum
- 1 a 3
- 3 a 5
- 5 ou mais

<P-003> - Tempo de Experiência com Análise/Gerência/Desenvolvimento/Arquitetura de Software:

- Nenhuma
- até 1 ano
- 1 a 3 anos
- 3 a 5 anos
- Acima de 5 anos

<P-004> - Papel no projeto

- Gerente de Projeto
- Arquiteto de Software
- Desenvolvedor
- Testador
- Analista de Negócios
- Outro: _____

Questões Fechadas - Stakeholders

- <L-S001> - A ferramenta considerou e representou os elementos da minha área de trabalho ligados à arquitetura de forma satisfatória.
- <L-S002> - Durante as atividades foi possível identificar uma linguagem comum entre os participantes.
- <L-S003> - Considero que a ferramenta teve um papel relevante em buscar estabelecer uma linguagem comum entre os participantes.
- <L-S004> - Considero que durante as atividades foi possível ter uma comunicação satisfatória (para a troca de ideias e informações, discussões, negociações e outros) com os participantes.
- <L-S005> - Considero que a ferramenta teve um papel relevante na minha comunicação com os participantes.
- <L-S006> - Considero que durante as atividades houve uma comunicação satisfatória (para troca de informações, discussões, negociações e outros) entre os participantes.
- <L-S007> - Considero que a ferramenta teve um papel relevante na comunicação entre os participantes.
- <L-S008> - Considero que tive um entendimento satisfatório do projeto de arquitetura proposto pelo arquiteto.
- <L-S009> - Considero que a ferramenta teve um papel relevante para o meu entendimento do projeto proposto.

Questões Abertas - Stakeholders

- <A-S001> - Como você avalia a forma que a ferramenta representou os elementos/itens da sua área de trabalho?
- <A-S002> - Você acredita que estes elementos/itens poderiam ser representados de outra forma? - Se sim, quais e como?
- <A-S003> - Você considera que outros elementos/itens poderiam ser considerados? Se sim, quais?
- <A-S004> - Durante as atividades, foi possível identificar que os participantes falavam "a mesma língua"? (Uma linguagem comum, em termos e definições) - Se sim, em quais momentos você viu isso?
- <A-S005> - Caso os participantes tenham "falado a mesma língua", isso impactou o andamento das atividades? Se sim, como?
- <A-S006> - Caso os participantes tenham "falado a mesma língua", você acredita que ferramenta teve participação nisso? - Se sim, como?
- <A-S007> - Como você avalia a comunicação -entre- os integrantes ao longo das atividades?
- <A-S008> - Qual sua avaliação sobre o papel da ferramenta na comunicação -entre- os participantes?
- <A-S009> - Houve algum problema ou limitação na comunicação? - Se sim, qual(is)?
- <A-S010> - Houve algum caso onde a ferramenta não pode auxiliar de forma satisfatória na comunicação entre os participantes? Se sim, como isso aconteceu?

- <A-S011> - Em algum momento das atividades você informou ou discutiu sobre Requisitos Funcionais, Atributos de Qualidade ou Restrições? Se não, passe para a pergunta <A-S015>. Se sim, como você avalia a forma com que estas informações foram passadas para o arquiteto?
- <A-S012> - [Caso não tenha informado ou discutido sobre Requisitos Funcionais, Atributos de Qualidade ou Restrições, passe para a pergunta <A-S015>.] Como você avalia a forma com que os itens relacionados a Requisitos Funcionais, Atributos de Qualidade ou Restrições foram representados na ferramenta?
- <A-S014> - [Caso não tenha informado ou discutido sobre Requisitos Funcionais, Atributos de Qualidade ou Restrições, passe para a pergunta <A-S015>.] Você acredita que os itens relacionados a Requisitos Funcionais, Atributos de Qualidade ou Restrições poderiam ser representados de uma outra forma? Se sim, quais e como?
- <A-S015> - Como você avalia o seu entendimento da arquitetura proposta pelo arquiteto?
- <A-S016> - Você teve alguma dúvida sobre a arquitetura definida depois de finalizada? Se sim, qual(is) dúvida(s)?
- <A-S017> - Na sua opinião, você conseguiria executar as atividades da sua área de trabalho com o entendimento que você teve desta arquitetura?
- <A-S018> - A ferramenta te auxiliou a entender a arquitetura proposta? Se sim, como?

Questões Abertas - Arquiteto de Software

- <A-A001> - Como você avalia a forma que a ferramenta representou os elementos da arquitetura de software?
- <A-A002> - Você considera que algum destes elementos da arquitetura poderiam ser representados de outra forma? - Se sim, quais e como?
- <A-A003> - Outros elementos da arquitetura poderiam ser considerados? Se sim, quais?
- <A-A004> - Durante as atividades, foi possível identificar que você falava "a mesma língua" dos stakeholders? - Se sim, foi possível ver que isso aconteceu com stakeholders tanto das áreas técnicas quanto as de negócios?
- <A-A005> - No caso dos stakeholders "falaram a mesma língua", qual foi o papel da ferramenta nisso? - Caso contrário, qual foi o impacto disso (não falarem a "mesma língua") no projeto?
- <A-A006> - Como você avalia a sua comunicação (troca de informações, discussões, negociações e outros) com os stakeholders ao longo das atividades?
- <A-A007> - Houve algum problema ou limitação na comunicação durante as atividades? Se sim, quais?
- <A-A008> - Qual foi o papel da ferramenta na sua comunicação com os stakeholders?
- <A-A009> - Houve algum caso onde a ferramenta não pode auxiliar de forma satisfatória? Se sim, como isso aconteceu?
- <A-A010> - Como você avalia a forma como os Requisitos Arquiteturalmente Significativos foram identificados ao longo das atividades?
- <A-A011> - Houve alguma dificuldade em identificar Requisitos Arquiteturalmente significativos? Se sim, quais?

- <A-A012> - Houve alguma informação sobre estes Requisitos Arquiteturalmente Significativos que estava dispersa entre stakeholders durante as atividades? Se sim, quais e como você obteve essas informações?
- <A-A013> - Houve algum problema ou limitação na identificação destes Requisitos Arquiteturalmente Significativos? Se sim, qual(is)?
- <A-A014> - Na sua opinião, a ferramenta atuou de forma relevante e satisfatória para obter estes Requisitos Arquiteturalmente Significativos? Se sim, como?
- <A-A015> - Como você avalia a forma que a ferramenta representou os Requisitos Arquiteturalmente Significativos?
- <A-A016> - A ferramenta te ajudou a identificar e representar estes Requisitos Arquiteturalmente Significativos de forma produtiva e satisfatória?
- <A-A017> - Os Requisitos Arquiteturalmente Significativos e a forma que foram representados na ferramenta te ajudaram no seu processo decisório? Se sim, como?
- <A-A018> - Houve algum problema ou limitação na representação destes Requisitos Arquiteturalmente Significativos? Se sim, qual(is)?
- <A-A019> - Como você avalia a execução da representação do projeto de arquitetura?
- <A-A020> - Você teve alguma dificuldade de representar o seu projeto de arquitetura? Se sim, qual(is)?
- <A-A021> - Como você avalia a forma com que a ferramenta representou este projeto?
- <A-A022> - Você considera que a ferramenta poderia representar o seu projeto de arquitetura de outra forma? - Se sim, como?
- <A-A023> - Como você avalia a comunicação do projeto de arquitetura proposto para os stakeholders?
- <A-A024> - Você teve alguma dificuldade para comunicar o projeto de arquitetura proposto ou informar mais detalhes sobre o projeto para os stakeholders? Se sim, qual(is)?

Questões Gerais - Todos

Esta seção é opcional e busca obter algumas impressões gerais.

- Como você avalia a utilidade da ferramenta para atividades de Arquitetura de Software?
- Quais aspectos positivos e negativos você considera sobre o uso da ferramenta?
- Quais dificuldades você observou durante o uso da ferramenta?
- Quais sugestões ou melhorias você tem para a ferramenta?

APÊNDICE F – RESPOSTAS DOS PARTICIPANTES

Este apêndice contém as respostas dos questionários, as respostas abertas foram classificadas pelo papel exercido no ensaio, contem as respostas do arquiteto, as questões das impressões gerais, respondidas por todos e também classificadas pelo papel exercido no ensaio. Também contém as respostas das questões fechadas em uma tabela organizando associando o papel pela pergunta, os números correspondem a escala como explicado no Capítulo 4.

Stakeholders

<A-S001> - Como você avalia a forma que a ferramenta representou os elementos/itens da sua área de trabalho?

- Analista de Negócios: Me ajudou a ter visão de áreas que não domino e antecipar possíveis problemas.
- QA Lead: Acredito que a ferramenta ficou bem bacana explorando muitas áreas como análise de negócio e de requisitos, assim como também aspectos de qualidade de projeto e gerência de projeto, por último nitidamente apresenta nos últimos estágios (direita), definições para arquitetura da solução proposta.
- Desenvolvedor: Fácil de visualizar e com bom fluxo de trabalho.
- Analista de Integração: Satisfatório.
- Gerente de Projeto: De forma positiva. Apesar de ser muita informação, mesmo que agrupada, o que pode tornar a interface um pouco confusa.

<A-S002> - Você acredita que estes elementos/itens poderiam ser representados de outra forma? - Se sim, quais e como?

- Analista de Negócios: Sim. Com exemplos de metodologia de todas as áreas para ajudar quem não tem conhecimento de áreas em específico.
- QA Lead: Acredito que a questão de Qualidade poderia ser exemplificada para qualquer um da área conseguir pensar de uma forma melhor e dar ideias. Outro ponto da ferramenta (dinâmica) é não ter um time-box durante a dinâmica.
- Desenvolvedor: Acredito que a representação foi satisfatória para o objetivo.
- Analista de Integração: Não, acredito que foi representado de forma Satisfatória.
- Gerente de Projeto: Talvez em um passo a passo em telas diferentes, com a possibilidade de se retornar para adicionar/alterar o conteúdo. Ou com as informações de orientação em formato de hints.

<A-S003> - Você considera que outros elementos/itens poderiam ser considerados? Se sim, quais?

- Analista de Negócios: Não.
- QA Lead: Acho que outro elemento que condiz com a dinâmica, o cenário fictício aplicados na dinâmica com aspectos de grandes e complexos e complexos projetos acaba deixando muito abrangente a discussão, mas sim, a ferramenta é bacana para exercitar as pessoas do projeto a terem ideias e pensarem em diferentes âmbitos sobre a solução proposta. E com mais tempo poderíamos sair com um bom produto, ou com um cenário menos complexo.
- Desenvolvedor: Os elementos apresentados detalharam bem.
- Analista de Integração: Não, acredito que foi satisfatório.
- Gerente de Projeto: Não consigo pensar em nenhum no momento.

<A-S004> - Durante as atividades, foi possível identificar que os participantes falavam “a mesma língua”? (Uma linguagem comum, em termos e definições) - Se sim, em quais momentos você viu isso?

- Analista de Negócios: Sim. No meu caso verifiquei que a área de QA e área de negócio estavam bem alinhadas

- QA Lead: Sim, em diferentes momentos. O Que me marcou foi a questão levantada sobre a LGPD e Acessibilidade de software, no âmbito social e mostrando a preocupação de todos em tornar a solução para o uso de qualquer pessoa, sendo que todos são da Tecnologia. Outro ponto foi o entendimento do tamanho do problema do cenário aplicado durante a dinâmica Canvas. Todos tomaram a decisão de propor um MVP para o suposto serviço mais utilizados pelas pessoas em um banco físico, julgamos ser depósitos e pagamentos.
- Desenvolvedor: Sim. Os tópicos foram bem discutidos tanto no momento de definir os requisitos como no apontamento das soluções.
- Analista de Integração: Sim, elementos/itens técnicos.
- Gerente de Projeto: Acredito que sim. Durante as análises de cada um dos blocos

<A-S005> - Caso os participantes tenham "falado a mesma língua", isso impactou o andamento das atividades? Se sim, como?

- Analista de Negócios: Percebi que acelerou o processo de definição de alguns quadros.
- QA Lead: Sim impactou produtivamente
- Desenvolvedor: Sim. Um nível de entendimento satisfatório tem impacto positivo no fluxo de trabalho.
- Analista de Integração: Sim, impactou de forma fluida.
- Gerente de Projeto: Impactou de forma positiva pois acaba ficando mais fácil integrando as diferentes frentes do projeto, porém quanto mais participantes acredito que o tempo de estruturação do documento seria prejudicado. Da mesma forma que com a ausência de 2 membros da equipe(pelo menos no dia em que participei), prejudicou o preenchimento das informações que provavelmente tornaria o documento mais completo caso todos estivessem presentes.

<A-S006> - Caso os participantes tenham "falado a mesma língua", você acredita que ferramenta teve participação nisso? - Se sim, como?

- Analista de Negócios: Sim, com exemplos que a ferramenta disponibilizou, o entendimento entre as áreas ficaram mais claras.
- QA Lead: Sim, pois nos guiou a discutir e explorar o problema a ser resolvido.
- Desenvolvedor: Sim. A ferramenta tende a focar em um tópico por vez trazendo a atenção de todos para um determinado assunto.
- Analista de Integração: Técnicas.
- Gerente de Projeto: Sim. Já que a ferramenta se propôs a consolidar conhecimentos de várias áreas diferentes do projeto.

<A-S007> - Como você avalia a comunicação entre os integrantes ao longo das atividades?

- Analista de Negócios: Com o uso da ferramenta, a comunicação ao meu ver se mostrou eficiente e com velocidade. O que fez que houvesse tempo para aprofundar em demais assuntos.
- QA Lead: Muito boa, todos se comprometendo a entender sobre o projeto e buscando dar o seu melhor.
- Desenvolvedor: Foi uma comunicação muito instrutiva e focada na construção eficiente do projeto.
- Analista de Integração: Satisfatória.

- Gerente de Projeto: Por serem membros atuantes da mesma equipe, acabou sendo bem tranquila.

<A-S008> - Qual sua avaliação sobre o papel da ferramenta na comunicação entre os participantes?

- Analista de Negócios: Se mostrou fundamental no experimento, sendo o motor que conectou todas as áreas
- QA Lead: Muito interessante, ajudou bastante o brainstorm das ideias.
- Desenvolvedor: A ferramenta teve um papel importante conduzindo a reunião.
- Analista de Integração: Satisfatória.
- Gerente de Projeto: Acredito que a ferramenta tenha contribuído para uma tomada de decisões mais ágeis e assertivas.

<A-S009> - Houve algum problema ou limitação na comunicação? - Se sim, qual(is)?

- Analista de Negócios: Sim, como haviam áreas distintas, levou tempo para contextualizar, porém, com a ferramenta esse tempo pode ser controlado tendo uma diminuição, mesmo assim o problema de tempo para gerar contexto existiu.
- QA Lead: Talvez falhas da internet devido a dinâmica ter sido feita virtualmente.
- Desenvolvedor: Nenhum.
- Analista de Integração: Não.
- Gerente de Projeto: Acredito que não. Porém se houvesse uma divisão mais clara das responsabilidades dos envolvidos talvez facilitasse o preenchimento das informações.

<A-S010> - Houve algum caso onde a ferramenta não pode auxiliar de forma satisfatória na comunicação entre os participantes? Se sim, como isso aconteceu?

- Analista de Negócios: Sim, contextualizar as áreas.
- QA Lead: Sim, acredito na falta de informação e dúvida sobre determinado board do Canvas, um vídeo explicativo ou um exemplo em cada Card ajudaria bastante a responder as perguntas.
- Desenvolvedor: Não.
- Analista de Integração: Não.
- Gerente de Projeto: Pelo tempo em que participei, acredito que não.

<A-S011> - Em algum momento das atividades você informou ou discutiu sobre Requisitos Funcionais, Atributos de Qualidade ou Restrições? Se não, passe para a pergunta <A-S015>. Se sim, como você avalia a forma com que estas informações foram passadas para o arquiteto?

- Analista de Negócios: Foram passadas em linguagem simples e informal competente de cada área. A ferramenta ajudou neste entendimento.
- QA Lead: Sim discutimos e trabalhamos em todos os temas abordados, acho que os problemas do projeto e a decisão do time foram passadas de forma transparente para o arquiteto, e os objetivos no desenvolvimento de um APP para transferências bancárias foi o software idealizado para atender nessa primeira fase o cliente da dinâmica
- Analista de Integração: Foram passadas acredito eu que de forma clara e objetiva.

- Gerente de Projeto: Sim. As informações foram um pouco superficiais. Não sei se o responsável pelo preenchimento deveria ser o Arquiteto ou alguém como um PO ou o próprio Gerente de Projetos que poderia detalhar a solução

<A-S012> - [Caso não tenha informado ou discutido sobre Requisitos Funcionais, Atributos de Qualidade ou Restrições, passe para a pergunta <A-S015>.] Como você avalia a forma com que os itens relacionados a Requisitos Funcionais, Atributos de Qualidade ou Restrições foram representados na ferramenta?

- Analista de Negócios: Foram descritas de forma linear, seguindo passo-a-passo para resolução do problema, a medida que avançávamos para os próximos quadros, podemos perceber que poderíamos complementar os requisitos, pois o processo ajudou a dar essa visão.
- QA Lead: Requisitos Funcionais e Restrições acredito que foram bem apresentados inclusive foi onde teve a decisão de seguir pela ideia de um MVP para pagamentos e transferências. Atributo de qualidade acho que faltou mais exemplos
- Analista de Integração: Satisfatória.
- Gerente de Projeto: As informações foram um pouco superficiais. Talvez exemplos do que preencher pudessem ajudar e também a possibilidade de adicionar ou alterar essas informações de acordo com cada projeto

<A-S014> - [Caso não tenha informado ou discutido sobre Requisitos Funcionais, Atributos de Qualidade ou Restrições, passe para a pergunta <A-S015>] Você acredita que os itens relacionados a Requisitos Funcionais, Atributos de Qualidade ou Restrições poderiam ser representados de uma outra forma? Se sim, quais e como?

- Analista de Negócios: Não. Da forma que o problema foi apresentado, a forma que preenchemos se mostrou adequada as nossas necessidades
- QA Lead: Sim com exemplos
- Analista de Integração: Não.
- Gerente de Projeto: Talvez com mais exemplos, um visual um pouco minimalista com menos informação em tela e seguindo uma espécie de passo a passo

<A-S015> - Como você avalia o seu entendimento da arquitetura proposta pelo arquiteto?

- Analista de Negócios: Pude compreender o todo, então minha avaliação é que o modelo é simples de compreensão.
- QA Lead: Meu entendimento foi razoável por não ser especialista
- Desenvolvedor: Satisfatória
- Analista de Integração: Satisfatório.
- Gerente de Projeto: Não tive problemas com o entendimento, informações foram bem claras.

<A-S016> - Você teve alguma dúvida sobre a arquitetura definida depois de finalizada? Se sim, qual(is) dúvida(s)?

- Analista de Negócios: Não
- QA Lead: Na verdade não conhecia alguns dos padrões propostos. Mas mensageria ficou bem claro pra mim.

- Desenvolvedor: como seria implementado o padrão "BFF"
- Analista de Integração: Não.
- Gerente de Projeto: Até o momento em que acompanhei não tive dúvidas.

<A-S017> - Na sua opinião, você conseguiria executar as atividades da sua área de trabalho com o entendimento que você teve desta arquitetura?

- Analista de Negócios: Sim
- QA Lead: Sim
- Desenvolvedor: Sim, com alguma pesquisa sobre as atividades
- Analista de Integração: Acredito que sim.
- Gerente de Projeto: Acredito que sim

<A-S018> - A ferramenta te auxiliou a entender a arquitetura proposta? Se sim, como?

- Analista de Negócios: Sim, processo simples e os diagramas deram o contexto
- QA Lead: Sim, pois entendemos no que o arquiteto deve pensar, uma boa parte, assim como outras pessoas do time de desenvolvimento devem se preocupar tbm. Show. a plataforma
- Desenvolvedor: Sim. Dando uma visão ampla sobre todo o projeto desde sua elaboração até como se espera que seja estando finalizado
- Analista de Integração: .
- Gerente de Projeto: Sim. Talvez pela participação no preenchimento, pela troca de idéias

Arquiteto de Software

<A-A001> - Como você avalia a forma que a ferramenta representou os elementos da arquitetura de software?

Avalio de forma positiva. Ajudou o grupo a seguir uma linha de raciocínio e voltar toda vez que era identificado itens que ficaram de fora em algum card anterior.

Entendo que padrões de projeto pudessem ficar de fora para otimizar o tempo do grupo maior, é algo que analistas (por exemplo) não conseguem contribuir muito.

<A-A002> - Você considera que algum destes elementos da arquitetura poderiam ser representados de outra forma? - Se sim, quais e como?

Não tenho certeza, mas imagino que a metodologia/forma proposta para tratar os atributos de qualidade seja algo de muita relevância para o desenvolvimento de um projeto de software. Não digo que estes itens de qualidade não sejam importantes, mas a forma de representá-los não é a mais produtiva.

<A-A003> - Outros elementos da arquitetura poderiam ser considerados? Se sim, quais?

Poderia testar um card com Requisitos não funcionais, talvez teria uma participação um pouco maior dos analistas.

<A-A004> - Durante as atividades, foi possível identificar que você falava "a mesma língua" dos stakeholders? - Se sim, foi possível ver que isso aconteceu com stakeholders tanto das áreas técnicas quanto as de negócios?

Sim, não senti uma desconexão em relação aos termos mais técnicos.

<A-A005> - No caso dos stakeholders "falaram a mesma língua", qual foi o papel da ferramenta nisso? - Caso contrário, qual foi o impacto disso (não falarem a "mesma língua") no projeto?

Talvez por a ferramenta ter abordado os assuntos de forma mais macro até ir aprofundando em detalhes mais técnicos posteriormente. Imagino que isso foi "contextualizando" os stakeholders menos técnicos em arquitetura.

<A-A006> - Como você avalia a sua comunicação (troca de informações, discussões, negociações e outros) com os stakeholders ao longo das atividades?

Senti que os stakeholders mais técnicos dominaram as discussões.

<A-A007> - Houve algum problema ou limitação na comunicação durante as atividades? Se sim, quais?

Acho que a limitação foi o que comentei na pergunta A-A006, senti que os analistas ficaram mais tímidos.

<A-A008> - Qual foi o papel da ferramenta na sua comunicação com os stakeholders?

Servir como um guia sobre o que precisava ser analisado e na ordem ideal de análise.

<A-A009> - Houve algum caso onde a ferramenta não pode auxiliar de forma satisfatória? Se sim, como isso aconteceu?

Senti que alguns cards (especialmente os atributos de qualidade) exigiam o conhecimento de uma metodologia específica para seu preenchimento. Embora fosse uma metodologia conhecida, não era de conhecimento do time. Alguns exemplos ajudariam.

<A-A010> - Como você avalia a forma como os Requisitos Arquiteturalmente Significativos foram identificados ao longo das atividades?

Talvez se as Restrições viessem após os Requisitos Funcionais ficasse mais interessante. É algo pra se pensar e testar.

<A-A011> - Houve alguma dificuldade em identificar Requisitos Arquiteturalmente significativos? Se sim, quais?

Sim, o grupo teve dificuldades pra identificar e principalmente para preencher os cenários dos Atributos de Qualidade, como mencionei na A-A009.

<A-A012> - Houve alguma informação sobre estes Requisitos Arquiteturalmente Significativos que estava dispersa entre stakeholders durante as atividades? Se sim, quais e como você obteve essas informações?

Senti que foi gasto um tempo com a análise de itens que estavam fora de escopo do projeto.

<A-A013> - Houve algum problema ou limitação na identificação destes Requisitos Arquiteturalmente Significativos? Se sim, qual(is)?

Sim, sobre a metodologia de cenários/métricas dos Atributos de Qualidade. Um(ns) exemplo(s) ajudaria e aumentaria a objetividade do preenchimento.

<A-A014> - Na sua opinião, a ferramenta atuou de forma relevante e satisfatória para obter estes Requisitos Arquiteturalmente Significativos? Se sim, como?

Entendo que atuou sim, talvez sem a ferramenta teríamos deixado de lado alguns requisitos importantes.

<A-A015> - Como você avalia a forma que a ferramenta representou os Requisitos Arquiteturalmente Significativos?

Acho que se houvesse um card para explicitamente para Requisitos não funcionais poderia extrair mais da participação dos analistas.

<A-A016> - A ferramenta te ajudou a identificar e representar estes Requisitos Arquiteturalmente Significativos de forma produtiva e satisfatória?

Ajudou sim, embora eu acredite que se houvesse mais um card como comentado na A-A015 poderia extrair mais dos analistas.

<A-A017> - Os Requisitos Arquiteturalmente Significativos e a forma que foram representados na ferramenta te ajudaram no seu processo decisório? Se sim, como?

Sim, eles estarem próximos dos outros assuntos facilitou o "ir e voltar" conforme o grupo ia identificando itens que tinham ficado de fora.

<A-A018> - Houve algum problema ou limitação na representação destes Requisitos Arquiteturalmente Significativos? Se sim, qual(is)?

No geral, um ponto de vista bem particular, não me adaptei com os post-it acima dos cards pequenos e com informações relevantes que ficavam ocultas a partir do momento que o post-it crescia em conteúdo. Para um projeto de software real, onde deve ser esgotada todos os itens/análises, faltará espaço eu imagino.

<A-A019> - Como você avalia a execução da representação do projeto da arquitetura?

Entendo que propiciou bastante a colaboração dos outros stakeholders, embora não tenham sido tão atuantes quanto os membros mais técnicos.

<A-A020> - Você teve alguma dificuldade de representar o seu projeto de arquitetura? Se sim, qual(is)?

Tive um pouco de dificuldade, comentada na A-A018.

<A-A021> - Como você avalia a forma com que a ferramenta representou este projeto?

Também comentada na A-A018.

<A-A022> - Você considera que a ferramenta poderia representar o seu projeto de arquitetura de outra forma? - Se sim, como?

Poderia apresentar e coletar informações de forma mais colaborativa, não limitar a um único membro a escrita dos itens. Além disso, a ferramenta app.diagrams.net não tem uma usabilidade boa.

<A-A023> - Como você avalia a comunicação do projeto de arquitetura proposto para os stakeholders?

Achei bem satisfatória. Poderia ser melhor com uma maior participação dos analistas, mas imagino que isso vai de grupo pra grupo.

<A-A024> - Você teve alguma dificuldade para comunicar o projeto de arquitetura proposto ou informar mais detalhes sobre o projeto para os stakeholders? Se sim, qual(is)?

Não identifiquei nenhuma dificuldade de comunicação.

Impressões Gerais

Como você avalia a utilidade da ferramenta para atividades de arquitetura de software?

- Arquiteto de Software: Entendo como útil, poderia apenas otimizar o tempo (e custo) dos analistas deixando eles de fora do preenchimento de termos mais técnicos como a definição dos padrões de projeto. Uma ferramenta com uma melhor usabilidade do que o diagrams.net também ajudaria.
- Analista de Negócios: Na minha visão, a ferramenta ajudou a prever diversas dificuldades que poderíamos ter se não estivessmos com as áreas que apoiaram o experimento
- Líder de QA: Boa.
- Desenvolvedor: Interessante para traçar estratégias, definir tecnologias a utilizar.
- Analista de Integração: Satisfatória.
- Gerente de Projeto: Acho que é uma ferramenta que promoveu a discussão e a análise entre os membros do projeto para se chegar a uma compreensão do todo.

Quais aspectos positivos e negativos você considera sobre o uso da ferramenta?

- Arquiteto de Software: Os aspectos positivos e negativos eu abordei nas perguntas da página anterior.
- Analista de Negócios: Positivo: Ajudou a equipe a ter contexto geral. Deu velocidade no andamento das questões a serem resolvidas. Antecipação de possíveis problemas. Entendimento de todas as áreas. Visão mais profunda do processo de arquitetura - Negativo: Metodologia científica com poucos exemplos, logos quem não é da área tem que ser contextualizado.
- Líder de QA: Positivos:
- Interação entre as pessoas desde a fase inicial do problema
- Identificação com realidade de projetos que atuo - Negativo: Mais tempo de dinâmica ou mais encontros seriam show pra gente desenvolver melhor a ideia
- Desenvolvedor: Interessante para elaborar e mapear projetos de forma ampla e sendo essa a proposta não identifiquei ainda pontos negativos com o contato que tive.
- Gerente de Projeto: Como positivo, justamente agregar o conhecimento ao mesmo tempo que promove a discussão e análise assim como ajuda a identificar possíveis falhas de comunicação. Como negativo a disposição das informações em tela

Quais dificuldades você observou durante o uso da ferramenta?

- Arquiteto de Software: As dificuldades eu abordei nas perguntas da página anterior.
- Analista de Negócios: Compreender como preencher determinado quadro, quando não domino uma determinada área de negócio. O que me levou a pesquisar para poder completar os quadros.
- Líder de QA: Percebi dificuldades de entendimento sobre o Tópico da Qualidade.
- Desenvolvedor: Nenhum. Foi intuitivo acompanhar a ferramenta e entender.
- Gerente de Projeto: Dificuldade no entendimento do que e como deveria ser preenchido em alguns blocos.

Quais sugestões ou melhorias você tem para a ferramenta?

- Arquiteto de Software: Já abordei em diversas perguntas anteriores, mas acho que vale um resumo: - Usar uma ferramenta mais colaborativa e com usabilidade melhor do que diagrams.net; - Otimizar o tempo dos membros menos técnicos, tirando eles de discussões sobre padrão de projeto a ser adotado; - Os post-it acima dos cards dificultam visualização de suas instruções e falta espaço para preenchimento; Aproveito para parabenizar pela ferramenta, a experiência de usar ela foi positiva!
- Analista de negócios: Oferecer exemplos de metodologia científica para cada quadro. Ter uma versão online em que a equipe pudesse acompanhar ou até mesmo participar da edição em tempo real.
- Líder de QA: TimeBOX - Melhorar aspecto da Qualidade - Ter o Cliente como uma das pessoa na dinâmica para: a) Tirar duvidas, b) Falar para onde o projeto deveria caminhar c) Tomar decisões.
- Desenvolvedor: O contato com a ferramenta não foi suficiente para pensar em possíveis melhorias.
- Gerente de Projeto: Pelo pouco tempo que tive visualizando, acho que o principal ponto seira mesmo melhorias na UI como a disposição dos blocos e como as informações são apresentadas.

Questões Fechadas

	An. Negócios	Líder de QA	Desenv.	An. Integração	Ger. Projeto
L-S001	6	6	6	7	6
L-S002	6	6	7	7	5
L-S003	7	6	6	7	6
L-S004	7	7	7	7	7
L-S005	7	6	6	7	5
L-S006	6	7	7	7	7
L-S007	6	7	7	7	6
L-S008	5	5	6	6	6
L-S009	6	5	6	6	5