

# O Método ASAP para a condução das atividades relacionadas à engenharia de requisitos no desenvolvimento de software

*The ASAP Method for conducting activities related to requirements engineering in software development*

## AUTORIA

Samar Moreira Reis  
UFF, Brasil  
samarmoreira@gmail.com  
Ramon Baptista Narcizo  
UFF, Brasil  
ramon\_narcizo@id.uff.br

## PALAVRAS-CHAVE

Desenvolvimento ágil;  
Engenharia de requisitos;  
Engenharia de sistemas;  
Requisitos;  
Software.

## RESUMO

A engenharia de requisitos desempenha um papel fundamental no desenvolvimento de software, influenciando diretamente o sucesso de um projeto. O objetivo deste trabalho é desenvolver um método para a execução das atividades de engenharia de requisitos no desenvolvimento de software, capaz de integrar as perspectivas e características das abordagens ágeis de desenvolvimento. Como metodologia, foi empregada a *Design Science Research*, bem como uma revisão sistemática de literatura, que culminaram no desenvolvimento de um método composto por atividades, técnicas e artefatos orientados a projetos de software. Após aplicação e validação em contexto ágil, concluiu-se que o método gera requisitos aderentes, completos, consistentes e precisos, servindo como guia para profissionais que buscam aprimorar suas práticas de engenharia de requisitos e desenvolver softwares de alto desempenho e alinhados às necessidades dos usuários.

## KEYWORDS

Agile development;  
Requirements engineering;  
Systems engineering;  
Requirements;  
Software.

## ABSTRACT

*Requirements engineering plays a fundamental role in software development, directly influencing the success of a project. The objective of this work is to develop a method for executing requirements engineering activities in software development, capable of integrating the perspectives and characteristics of agile development approaches. As a methodology, Design Science Research was used, as well as a systematic literature review, which culminated in the development of a method composed of activities, techniques and artifacts oriented to software projects. After application and validation in an agile context, it was concluded that the method generates adherent, complete, consistent, and accurate requirements, serving as a guide for professionals seeking to improve their requirements engineering practices and develop high-performance software aligned with user needs.*

# 1. Introdução

Nas últimas décadas, o desenvolvimento de software evoluiu rapidamente, introduzindo tecnologias mais avançadas e produtos mais complexos. Recentemente, o mercado brasileiro de tecnologia da informação tem demonstrado crescimento considerável, particularmente no que se refere à engenharia de software, principal atividade em 67,7% das aproximadamente 29.000 empresas do país (ABES, 2022). A engenharia de requisitos desempenha um papel crítico nesse contexto, delineando os serviços e as restrições do sistema, sendo um aspecto decisivo para o sucesso do projeto. Essa atividade é parte integrante não apenas do desenvolvimento de software, mas de qualquer projeto técnico que vise criar um produto, serviço ou resultado único (Hull, Jackson e Dick, 2005)

Maté e Silva (2005) destacam a diversidade de configurações possíveis para um sistema, tornando a flexibilidade do software um desafio. Mudanças são aspectos comuns nestes tipos de projetos, exigindo adaptação na gestão do escopo. Assim, é fundamental definir, gerenciar e controlar os requisitos, garantindo o atendimento às necessidades das partes interessadas e a minimização de problemas na implementação do sistema. A engenharia de requisitos, definida por Sommerville (2011, p. 57) como “o processo de descobrir, analisar, documentar e verificar os serviços e restrições do sistema”, é uma fase determinante no desenvolvimento, pois um erro durante sua execução pode afetar todo o projeto (Maté e Silva, 2005).

A qualidade dos requisitos é, portanto, crucial para o sucesso do sistema e pode ser avaliada em termos de aderência, completude, consistência e clareza (Davis, 1994). No entanto, problemas na modelagem de requisitos – tais como especificações excessivamente granulares ou abstratas – podem surgir, dificultando a representação de iterações no seu processo de elicitação (Wagner et al., 2017). Além disso, a transparência na engenharia de requisitos pode ser comprometida em função da complexidade de se representar múltiplos ciclos de revisão da especificação, conduzindo, em alguns casos, a uma convergência artificial (Matheson, 2019).

Neste trabalho propõe-se uma pesquisa baseada na metodologia *Design Science Research* (DSR) para preencher uma aparente lacuna na literatura sobre o desenvolvimento de requisitos. Com uma perspectiva empírica dos autores, incluindo um envolvimento direto em projetos de

desenvolvimento de software técnico-científico, o objetivo é desenvolver um artefato, na forma de método, que integre as perspectivas e características dos métodos ágeis de desenvolvimento na execução da engenharia de requisitos em projetos de software. O artefato desenvolvido visa aprimorar as atividades de engenharia de requisitos, potencializando o sucesso de projetos de software. Sua importância é destacada em sistemas que operam em estreita relação com o ambiente, como sistemas personalizados, oportunizando a inovação nesses projetos. Este artigo inclui uma revisão da literatura sobre o tema, seguida pela descrição da metodologia de desenvolvimento do artefato e dos resultados do estudo de caso. As conclusões e sugestões para futuras pesquisas são apresentadas ao final.

## 2. Revisão da literatura

### 2.1. Engenharia de software

Uma compreensão apropriada sobre a engenharia de software exige uma definição precisa para o termo “software”. O Instituto de Engenheiros Elétricos e Eletrônicos (IEEE), define software como “programas de computador, procedimentos e possivelmente documentação e dados associados relativos à operação de um sistema de computador” (IEEE STD 610.12-1990, 1990, p. 66). A literatura apresenta várias definições para o termo “engenharia de software”. Uma visão que merece destaque é aquela apresentada por Fritz Bauer durante a Conferência da Organização do Tratado do Atlântico Norte sobre Engenharia de Software em 1968, ao estabelecer que a “Engenharia de Software” é definida como a aplicação de princípios sólidos de engenharia ao desenvolvimento de software, com o objetivo de produzir soluções de software eficientes, confiáveis e funcionais em ambientes reais (NATO, 1968).

No processo de construção de um software, não há espaço para a trivialidade. Pelo contrário, exige esforço técnico e especializado. Um processo, de maneira ampla, pode ser definido como “uma série de atividades sistemáticas direcionadas para alcançar um resultado final, de tal forma que se aja em relação a uma ou mais entradas a fim de criar uma ou mais saídas” (PMI, 2017, p. 720). Os processos e métodos ágeis de desenvolvimento de software levaram os conceitos de garantia de qualidade de software para além do simples atendimento aos requisitos, validação e

verificação do cliente. A agilidade abre novos horizontes na área de garantia de qualidade de software. A análise do manifesto ágil, proposto por Beck et al. (2011), revela que o desenvolvimento ágil de software não se trata apenas de atender aos requisitos do cliente, mas sim de atender às mudanças de requisitos até o nível da implantação do produto desenvolvido.

Após o primeiro *eWorkshop* sobre métodos ágeis em 2002, Lindvall et al. (2002) resumiram a definição de métodos ágeis como um grupo de processos de desenvolvimento de software que possuem quatro características. Segundo Stamelos e Sfetsos (2007), eles são, na visão teórica do método ágil de desenvolvimento, iterativos, incrementais, auto-organizáveis e emergentes (Stamelos e Sfetsos, 2007). Ainda segundo esses autores, na visão funcional, suas características são flexibilidade, agilidade, prontidão para o movimento, atividade, destreza em movimento e ajustabilidade.

Dentre os principais métodos ágeis, pautados no manifesto ágil, alguns são destacados por Highsmith (2002), sendo estes o *Scrum*, *Dynamic Systems Development Method*, *Crystal Methods*, *Feature-Driven Development*, *Extreme Programming* e *Adaptive Software Development*. As características das abordagens dos métodos ágeis auxiliam na compreensão do cenário atual enfrentado nos projetos de desenvolvimento de software. Portanto, a incorporação das características dos métodos ágeis de desenvolvimento pode contribuir para a identificação de práticas adicionais que melhorem a qualidade, eficiência e eficácia da engenharia de requisitos.

## 2.2. Engenharia de requisitos

A engenharia de requisitos é a principal fase em um projeto de desenvolvimento de software que direciona o projeto para a solução do problema adereçado (Vliet, 2007; Sommerville, 2011; Hull, Jackson, e Dick, 2005; Young, 2004). Conforme sugere Wiegers e Beatty (2013), não se deve presumir que todas as partes interessadas do projeto compartilham uma noção comum do que são os requisitos. A definição dada no glossário padrão IEEE de Terminologia de Engenharia de Software é a seguinte:

- (1) Uma condição ou capacidade necessária a um usuário para resolver um problema ou atingir um objetivo.
- (2) Uma condição ou capacidade que deve ser satisfeita ou possuída por um sistema ou componente do sistema para satisfazer um contrato, norma, especificação ou outros documentos formalmente

impostos. (3) Uma representação documentada de uma condição ou capacidade como em (1) ou (2) (IEEE STD 610.12-1990, p. 62).

Para Sommerville (2011, p. 58), os requisitos de usuário “são declarações, em uma linguagem natural, com diagramas, de quais serviços o sistema deverá fornecer a seus usuários e as restrições com as quais este deve operar”. Isso demonstra que esses requisitos são escritos do ponto de vista das partes interessadas no sistema. Por sua vez, os requisitos de sistema, segundo Sommerville (2011, p. 58), são definidos como “as descrições mais detalhadas das funções, serviços e restrições operacionais do sistema de software”. Os requisitos funcionais e os requisitos não-funcionais podem ser expressos na perspectiva de requisitos de usuário e em termos de requisitos de sistema. Na ótica de requisitos de usuário, os requisitos são elevados a um nível abstrato, com o intuito de serem compreendidos pelos usuários do sistema. No entanto, numa perspectiva específica dos requisitos, estes descrevem em detalhes as funções do sistema, suas entradas e saídas, exceções etc. (Capuano, 2011; Sommerville, 2011).

A Engenharia de Requisitos é um campo relativamente novo, que se refere ao conjunto de atividades relacionadas à elicitacão, análise, especificacão, validacão e gerenciamento de requisitos para um sistema. Pressman (2011, p. 127) relata que “a engenharia de requisitos constrói uma ponte para o projeto e para a construçãõ”. Para Young (2004, p. 194) “os requisitos são a base para todo o trabalho realizado na engenharia de sistemas e software”. A literatura sobre engenharia de requisitos destaca os benefícios de estabelecer um processo adequado para organizações e projetos de software. Como observado por Sommerville e Sawyer (1997, p. 10), “uma boa descriçãõ do processo de engenharia de requisitos fornecerá orientaçãõ às pessoas envolvidas e reduzirá a probabilidade de que as atividades sejam esquecidas ou realizadas de forma superficial”. Pesquisas na indústria apontam oportunidades significativas para melhorias na prática da engenharia de requisitos (Shoemaker et al., 2011; The Standish Group, 2015).

## 2.3. Trabalhos correlatos

Uma busca por estudos que propõem métodos que englobam a condução das atividades de engenharia de requisitos no desenvolvimento de sistemas foi conduzida nas seguintes bases acadêmicas e científicas: ACM Digital Library, IEEEExplore, Portal de Periódicos CAPES e SCOPUS. As palavras-chave consideradas para a busca no portal de Portal de Periódicos

CAPES foram: “engenharia de requisitos”, “framework”, “método”, “metodologia”, “abordagem”, “ágil” e “técnica”. Nas outras bases acadêmicas, as palavras-chave em português foram traduzidas para o inglês. Após a pesquisa, observou-se que o tema é amplamente estudado globalmente, mas foi encontrado um número restrito de métodos que abordam especificamente todas as atividades relacionadas à engenharia de requisitos em um contexto ágil. Com base nos resultados, dois métodos merecem destaque.

O primeiro é o método SRSONtology, proposto por Elliott (2012), que se concentra nas atividades de engenharia de requisitos, documentação de requisitos de software. O método emprega uma nova abordagem nessas atividades, incorporando princípios de engenharia de ontologias. O trabalho desenvolvido por Elliott (2012), busca responder se a área de conhecimento de requisitos de software do SWEBOK pode ser modelada dentro de uma ontologia. O SWEBOK é um guia para o corpo de conhecimento de engenharia de software. Apesar de a aplicação do método contribuir para a consistência de definição de conceito de domínio, melhoria da qualidade do artefato de software e consistência da documentação de requisitos de software, pouco foi evidenciado sobre as atividades de gerenciamento de requisitos junto à equipe de desenvolvimento da solução.

O segundo método é o SAMEM. Um método pragmático de desenvolvimento de soluções que se concentra em combinar uma abordagem de gerenciamento de projetos iterativo e incremental (ágil), com ênfase na modelagem de artefatos de projeto (*Model driven Development* ou MDD). O método SAMEN, proposto por Matheson (2019), dá início à modelagem nas primeiras etapas do projeto a partir dos objetivos da solução, e dá continuidade ao longo do trabalho de implementação. O SAMEM pretende ser uma metodologia de solução de software prática e pragmática, apresentando uma discussão sobre os benefícios de negócio alcançados nos projetos. Apesar de bem estruturado, a execução de cada etapa não indica especificamente quais são as técnicas e ferramentas que devem ser utilizadas. Ao contrário, é indicado que cada domínio de projeto tenha liberdade para aplicar as técnicas e ferramentas desejadas.

### 3. Procedimentos metodológicos

O objetivo fundamental da pesquisa é resolver problemas mediante o emprego de procedimentos científicos. Segundo Aken (2004), uma ciência que visa prescrever uma solução pode ajudar a reduzir a lacuna existente entre a teoria e a prática. Dessa forma, pesquisas que resultam em prescrição têm aplicação mais fácil, inclusive pelos profissionais das organizações, podendo também promover o reconhecimento de sua relevância para o campo prático (Aken, 2004). É neste contexto que a *Design Science Research* se posiciona: como um paradigma epistemológico que pode orientar a pesquisa para a resolução de problemas e *design* de artefatos. Uma pesquisa tem não apenas o objetivo de estudar a solução de um problema, mas também de aplicar o conhecimento em situações reais, trazendo benefícios para as organizações e gerando conhecimento para outros profissionais e pesquisadores. Além disso, a presente pesquisa está inserida no campo dos sistemas de informação, e conforme corroborado por Peffers et al. (2007), a *Design Science Research* é amplamente utilizada nesse campo.

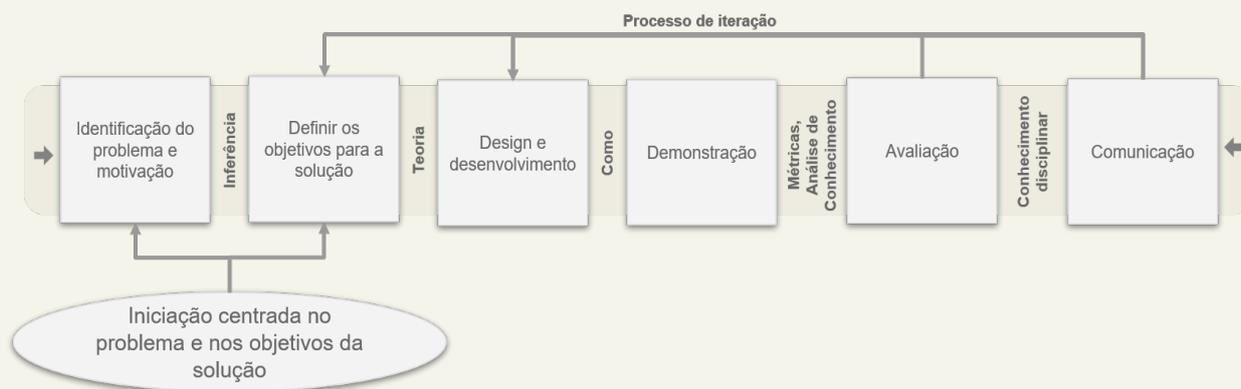
Para Simon (1996), um artefato é aquilo que foi produzido ou sofreu alteração intencional pelo homem, ou seja, algo que não ocorre naturalmente. Um artefato gerado pela *Design Science Research* pode assumir várias formas possíveis, como constructos, modelos, estruturas, arquiteturas, princípios de design, métodos ou instanciações e teorias de *design* (Vaishnavi, Kuechler e Petter, 2004). Para o presente trabalho, o constructo “método” foi o tipo de artefato gerado, primariamente devido à natureza do problema que o cerca. Assim, um método pode ser entendido como “um conjunto de etapas usado para executar uma tarefa” (Vaishnavi, Kuechler e Petter, 2004, p. 17).

Peffers et al. (2007) propõem uma metodologia de *Design Science Research* fundamentada na literatura existente sobre *Design Science* em sistemas de informação e disciplinas relacionadas. A metodologia proposta, fornece um modelo mental para a apresentação de resultados. Peffers et al. (2007) apresentam uma diretriz metodológica geral para pesquisas eficazes em *Design Science*. As atividades propostas para utilização da metodologia do *Design Science Research* estão resumidas na Figura 1.

Na “identificação do problema e motivação”, o objetivo é descrever e justificar o problema, destacando sua importância na resolução e sua utilidade na mitigação do problema. No presente trabalho, essa etapa foi discutida nas seções de introdução e de revisão da literatura, a partir de

uma perspectiva prática, tomando por base a experiência dos autores em projetos de desenvolvimento de software, e de uma perspectiva teórica, considerando uma revisão abrangente da literatura. Essas perspectivas permitiram definir precisamente as características e o escopo do problema, bem como fornecer motivação para a construção de um artefato orientado a um problema relevante no mundo real, que possam também apresentar contribuições úteis à literatura especializada sobre o tema.

**Figura 1.** Processo de iniciação da metodologia de *Design Science Research*.



Fonte: Peffers et al. (2007).

A “definição dos objetivos para a solução” garante que, após a criação da solução, será possível avaliar e, se necessário, propor ajustes. Neste trabalho, os principais atributos e características da operação do artefato, a fim de que ele possa de fato cumprir a sua função, são contextualizados na seção introdutória do documento. Complementarmente, uma avaliação mais detalhada é explorada na seção de resultados e discussão. A etapa de definição dos objetivos para a solução apoiou-se primariamente na experiência prática de vivenciar recorrentemente a natureza do problema, tanto por meio de requisições profissionais, quanto a partir da percepção da existência de uma lacuna em termos das soluções técnicas atualmente disponíveis.

No “*design* e desenvolvimento” cria-se o artefato, ou seja, a solução. Neste trabalho, isso é feito na seção de resultados e discussão, utilizando-se da fundamentação que detalha como o processo de construção do artefato foi conduzido, tanto por meio da perspectiva prática quanto do pensamento científico, sustentados por meio da metodologia e fundamentação teórica. Por sua vez, a atividade de “demonstração” tem o objetivo de demonstrar o uso da solução para a

resolução do problema. Isso também é apresentado na seção de resultados e discussão, por meio do uso do artefato em um caso prático real em um contexto profissional.

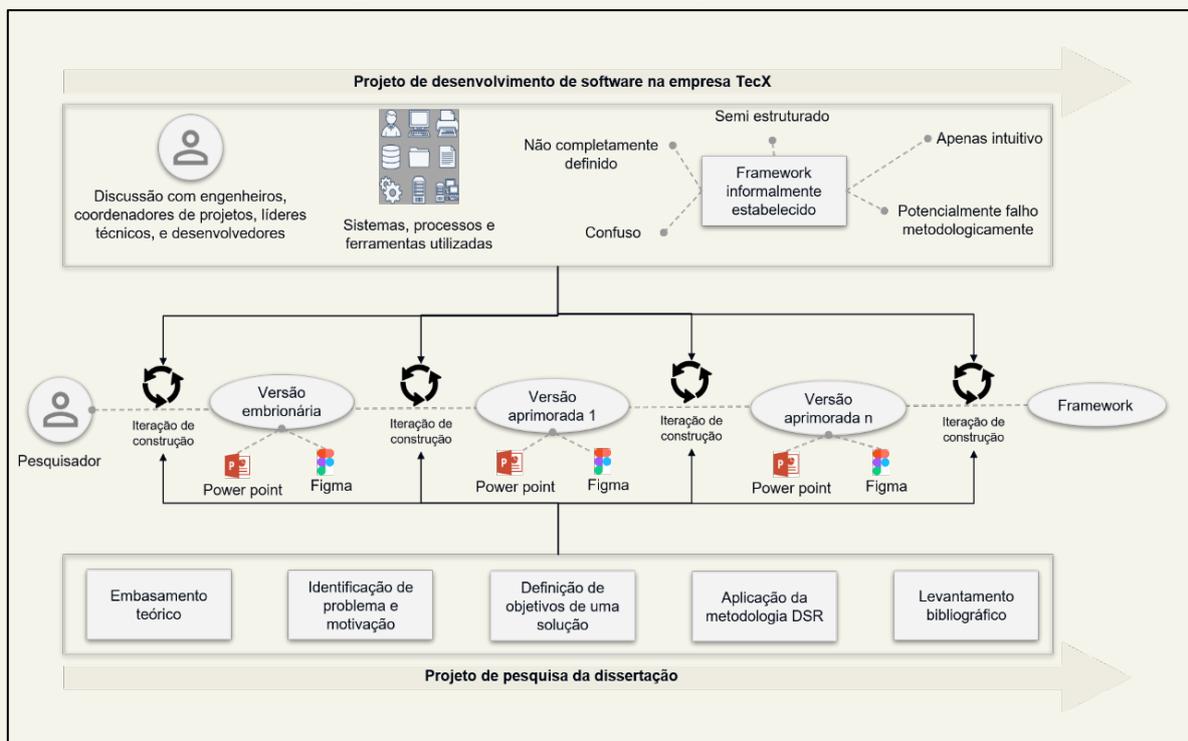
Na atividade de “avaliação” verifica-se o grau de aderência do artefato para a resolução do problema. Essa avaliação é feita por meio de entrevistas com os usuários, e suas opiniões acerca do cumprimento dos objetivos do artefato e da aderência das modificações implantadas com os objetivos de negócio da organização. Por fim, na atividade de “comunicação”, divulga-se o problema, a sua importância, a solução desenvolvida e os resultados obtidos por meio de sua aplicação. Esta atividade é executada, em parte, por meio da publicação e disponibilização do presente trabalho à comunidade acadêmica e profissional interessada no domínio do problema e do artefato desenvolvido para resolvê-lo.

## 4. Resultados e discussão

### 4.1. O processo de construção do método

No processo de desenvolvimento do artefato, duas perspectivas, representadas na Figura 2, foram fundamentais para a sua construção. A prática, na parte superior do esquema, e a teórica, na parte inferior. Várias iterações de construção do artefato foram realizadas levando em consideração os requisitos da própria solução, tais como a orientação a resultados, o foco no cliente, o caráter colaborativo esperado do método, a inclusão de mecanismos de gestão visual, bem como características de empirismo, continuidade e adaptabilidade. A iteração inicial, que gerou a versão preliminar, foi fortemente desenvolvida a partir de um método de levantamento de requisitos usado na empresa na qual o artefato foi testado. Esse método, ainda que estabelecido de modo tácito e carecendo de uma documentação formalizada, era amplamente conhecido e empregado por um dos autores em sua atuação profissional. Outras iterações geraram versões cada vez mais aprimoradas e adaptadas do artefato aos requisitos mencionados, até se chegar em uma solução considerada estruturada e madura. Assim, em síntese, a versão preliminar serviu como um embrião para o planejamento e construção do método.

Figura 2. Processo de construção do artefato.



Fonte: Elaborado pelos autores.

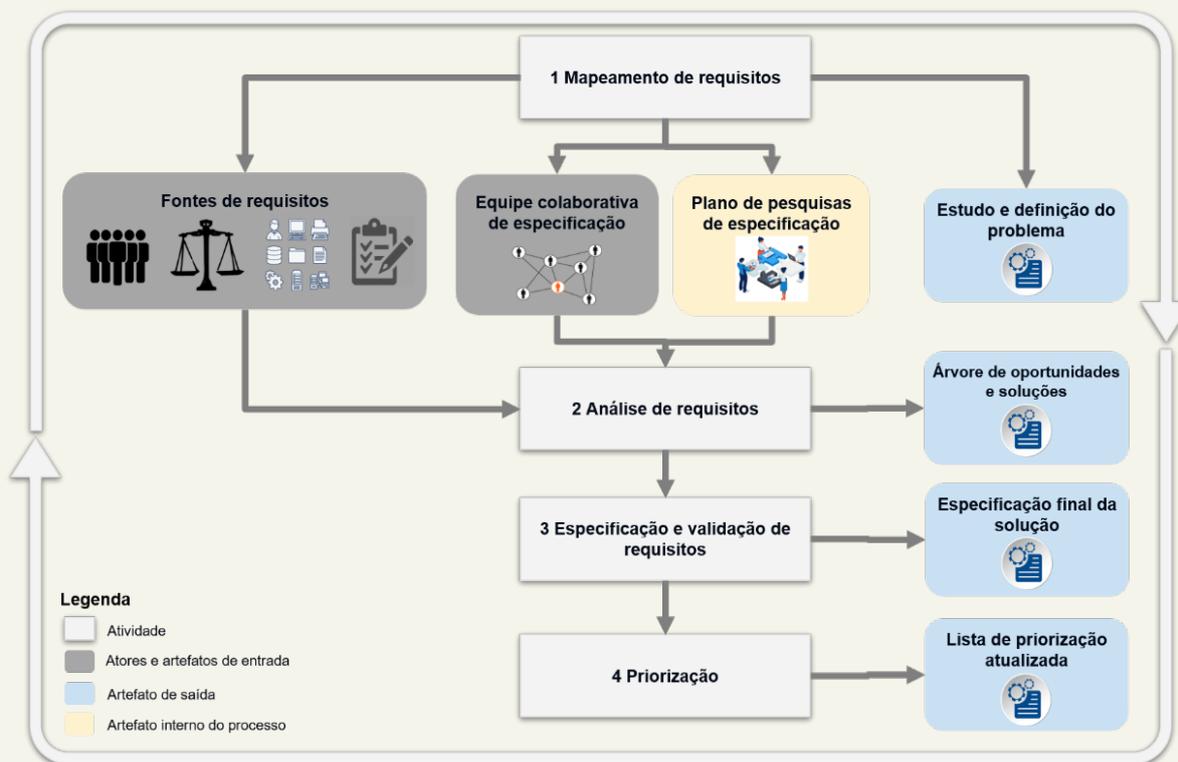
No contexto da pesquisa, em relação ao embasamento que fundamentou a construção do artefato, cada iteração de construção buscou um embasamento teórico alinhado com o problema e a motivação do estudo. A definição clara dos objetivos da solução – por meio da aplicação da metodologia de *Design Science Research* e do levantamento bibliográfico – conduziu os autores a um aprimoramento coerente e consistente da solução.

### 4.3. Método geral

O método para desenvolvimento de requisitos proposto na Figura 3 é o resultado decorrente das várias iterações de construção. A proposta de solução apresenta as etapas de mapeamento de requisitos, análise de requisitos, especificação e validação de requisitos, e priorização de requisitos. Todas as atividades propostas no método devem ser consideradas executáveis em um ciclo de desenvolvimento, e de forma iterativa. Nos modelos ágeis de desenvolvimento, cada equipe determina o tamanho do seu ciclo de desenvolvimento. Portanto, durante um

determinado ciclo, o método proposto deve ser executado a fim de gerar requisitos que venham a ser desenvolvidos na próxima *sprint*, ou seja, no próximo ciclo de desenvolvimento.

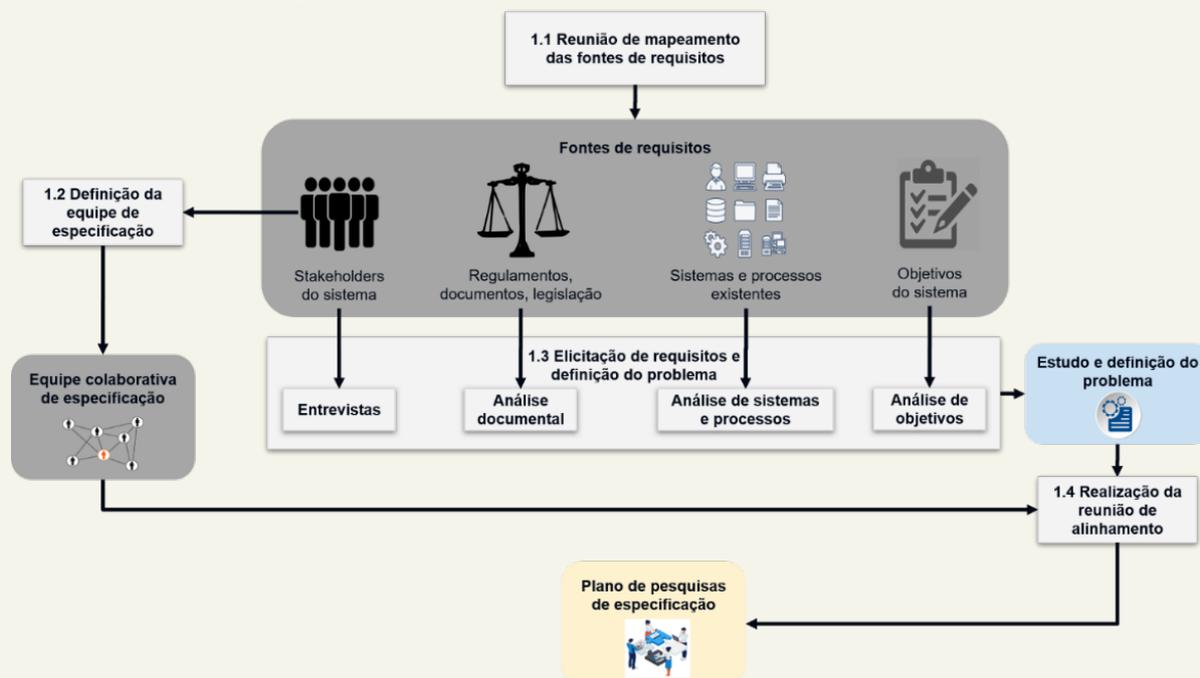
Figura 3. Método para desenvolvimento de requisitos.



Fonte: Elaborado pelos autores.

A etapa de mapeamento de requisitos, detalhada na Figura 4, inicia-se com a reunião de mapeamento das fontes de requisitos. Para esta reunião é importante a participação do engenheiro de requisitos, dos proprietários do negócio, e pelo menos um membro da equipe de desenvolvimento, preferencialmente o líder. Esta reunião busca identificar todos os elementos que podem fornecer informações relevantes para determinar os requisitos do sistema. Ou seja, são considerados as fontes de informações sobre os requisitos, que são, os *stakeholders* do sistema, regulamentos, documentos do domínio e legislação, sistemas e processos existentes e objetivos do sistema.

Figura 4. Etapa de mapeamento de requisitos do método.

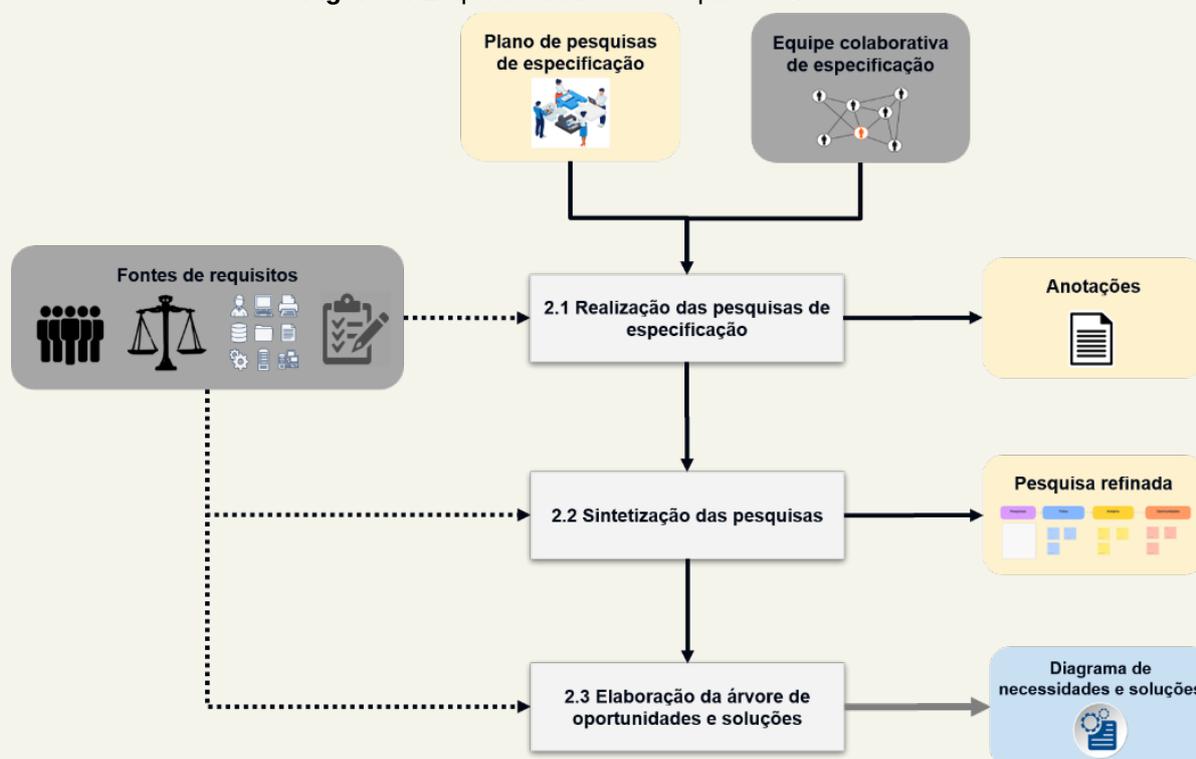


Fonte: Elaborado pelos autores.

Considerando os *stakeholders* do sistema, é criada uma equipe colaborativa de especificação que auxilia o engenheiro de requisitos na atividade de eliciação de requisitos e na definição do problema no qual a pesquisa preliminar se baseará. Portanto, busca uma compreensão do que se espera do produto a ser desenvolvido. Esse estudo tem como objetivo proporcionar uma visão geral da base de conhecimento adquirida a partir das fontes de requisitos selecionadas, bem como identificar os elementos fundamentais que serão utilizados como guia na próxima etapa de análise de requisitos. São utilizadas técnicas de entrevistas, análise documental, análise de sistemas e processos e análises de objetivos. A definição do problema segue a estrutura da seguinte frase: “nós observamos que o estado atual o que faz com que essas personas sofram com suposição sobre o porquê o estado atual não é o ideal”. Um documento de estudo é criado usando parte do material adquirido, e deve estar disponível para acesso por parte dos interessados. Este documento deve conter também a declaração do problema. Por fim, nesta primeira etapa é realizada uma reunião de alinhamento que busca definir as funções da equipe colaborativa de especificação, o particionamento do problema, e o planejamento da realização das pesquisas de especificação.

Na etapa de análise de requisitos, detalhada na Figura 5, primeiramente são realizadas as pesquisas de especificação. Ao longo da pesquisa, compete ao engenheiro de requisitos registrar informações relevantes, incluindo fatos, questões consensuais, conclusões sobre questões menos consensuais e novas ideias ou tópicos de discussão. Para cada pesquisa é criada uma lista de anotações que consolida as informações registradas. Após a realização das pesquisas, as mesmas são sintetizadas com o objetivo de registrar e categorizar adequadamente os dados coletados. Em sequência, é elaborado o diagrama de necessidades e soluções, uma representação visual, em uma estrutura organizada que conecta o resultado esperado com as necessidades identificadas e as possíveis soluções propostas.

**Figura 5.** Etapa de análise de requisitos do método.



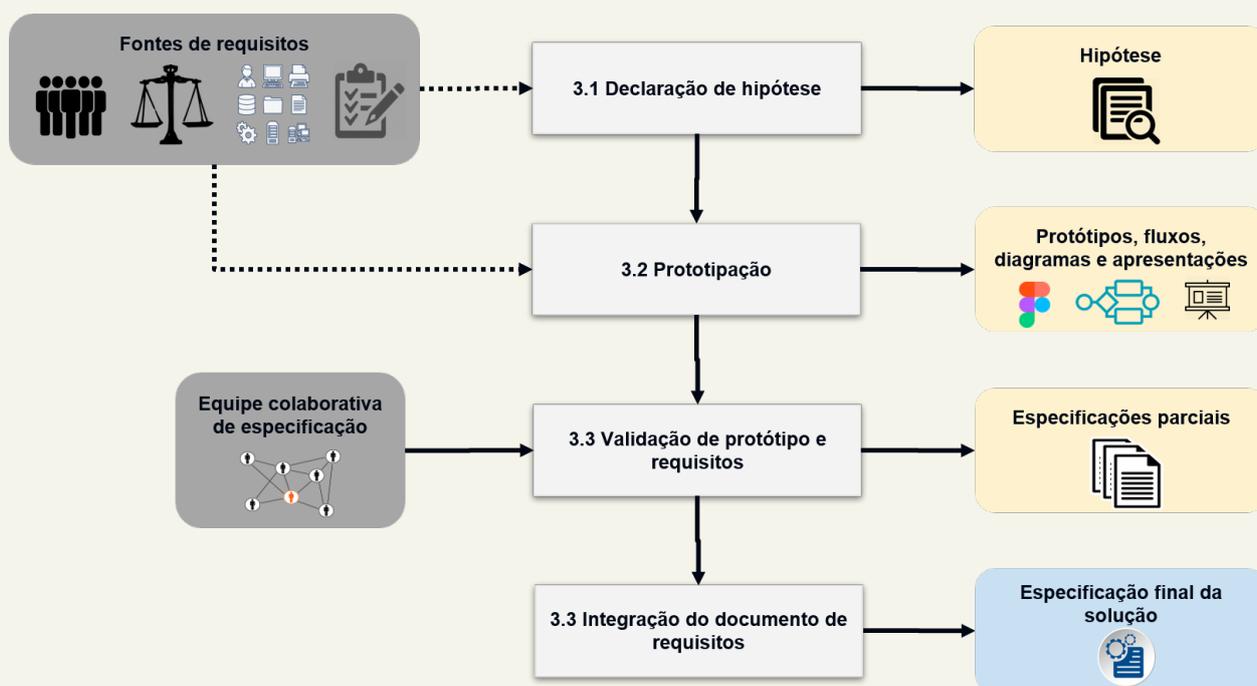
Fonte: Elaborado pelos autores.

A terceira etapa do método, apresentada na Figura 6, trata da especificação e validação dos requisitos. Ela tem o propósito de gerar a especificação final da solução. Primeiramente é realizada a declaração de hipótese, seguindo a estrutura da seguinte frase: “nós acreditamos que fazendo isso para essas personas vamos atingir esse objetivo e saberemos que é verdade

com esse *feedback*, reação ou dado quantitativo”. Neste modelo de hipótese, o objetivo é validar a solução proposta de maneira eficiente, utilizando o feedback do cliente.

A atividade seguinte, prototipação, tem o objetivo de construir protótipos, de baixa ou alta fidelidade, para validação da interface e funcionalidades do sistema com o cliente. Também são descritas as principais regras de negócio que o sistema deve obedecer, bem como os fluxos a serem seguidos ou inseridos no contexto do sistema. Os protótipos são criados a partir das hipóteses de solução geradas no diagrama de necessidades e soluções da etapa 2 (Figura 5), de forma que estejam alinhados com a declaração de hipótese da etapa 3 (Figura 6). Na elaboração dos protótipos é importante consultar as fontes de requisitos levantadas na primeira etapa deste método proposto, a fim de garantir alinhamento e coerência.

**Figura 6.** Etapa de especificação e validação de requisitos do método.



Fonte: Elaborado pelos autores.

A validação do protótipo e da especificação dos requisitos é o processo que determina se o protótipo, juntamente com o levantamento dos requisitos funcionais e não-funcionais é consistente para assegurar que as necessidades dos *stakeholders* serão atendidas. O processo de validação, tanto de protótipos quanto de especificação de requisitos, pode requerer várias

reuniões de apresentação, onde os protótipos são aprimorados a partir dos *feedbacks* recebidos. A última atividade da etapa de especificação e validação de requisitos é a integração do documento de requisitos, no qual é gerado o artefato da especificação final da solução. Nessa atividade, o engenheiro de requisitos deve formalmente documentar uma especificação que abranja todos os aspectos necessários para o desenvolvimento da solução.

A última etapa do método é a priorização, aspecto fundamental para entregar valor rapidamente ao cliente e ao negócio. Após a consolidação e validação dos requisitos, uma lista de tarefas é gerada, representando as novas funcionalidades, melhorias, telas e desenvolvimentos necessários. A estratégia de priorização surge das decisões sobre os resultados desejados, clientes a atender e oportunidades a explorar. Portanto, para cada tarefa levantada, o método propõe a utilização dos três componentes sugeridos no *Planning Poker*, uma técnica criada por Grenning (2002). Esses componentes são: “trabalho manual”, “complexidade” e “incerteza”, e são utilizados para pontuar as tarefas. Adicionalmente, para complementar esta técnica, foi acrescentado o componente “escopo”, sendo este último de responsabilidade do engenheiro de requisitos. Em conjunto, esses componentes auxiliarão a ordenar e priorizar a execução das tarefas.

Considerando que o artefato desenvolvido é um método que valoriza e emprega o modelo mental ágil — onde a busca por respostas de solução para as necessidades dos usuários tem um caráter de urgência — ele foi batizado como “Método ASAP”. A nomenclatura utilizada está baseada na abreviação da expressão em idioma inglês “*as soon as possible*”, que pode ser traduzida ao português como “o mais breve possível” ou “quanto antes”. Além disso, para o método desenvolvido, o termo ASAP também é utilizado como acrônimo para *Accelerated development of Software requirements in Agile Process*, que pode ser traduzido como “Acelerado desenvolvimento de requisitos de Software em Processos Ágeis”.

## 4.4. Aplicação do Método ASAP

O Método ASAP foi testado em um único estudo de caso prático, seguindo as etapas e orientações prescritas por este artefato. Porém, de antemão é importante destacar as potenciais limitações deste tipo avaliação, conduzida em um único cenário organizacional. O estudo envolveu o desenvolvimento de uma plataforma integrada para gerenciar processos de

contratação na cadeia de suprimentos de uma grande empresa do setor energético brasileiro. Dada a complexidade e os extensos prazos verificados tipicamente neste setor da indústria, é fundamental garantir transparência nos processos de contratação, bem como cumprir os prazos e adequar as quantidades contratadas às demandas reais, incluindo a gestão de aditivos para ajustar prazos e verbas contratados de forma eficiente. Desta forma, a situação em questão demonstra-se como um problema adequado à aplicação e ao teste da funcionalidade do artefato desenvolvido. Assim, na execução da primeira etapa do Método ASAP, definiu-se a documentação inicial com a declaração do problema que pode ser observada na Figura 7.

Figura 7. Tela com a página de mapeamento dos requisitos.

The screenshot shows a web-based interface for requirements mapping. The main content area is titled "Mapeamento do processo de acompanhamento de aditivos" and is divided into three sections:

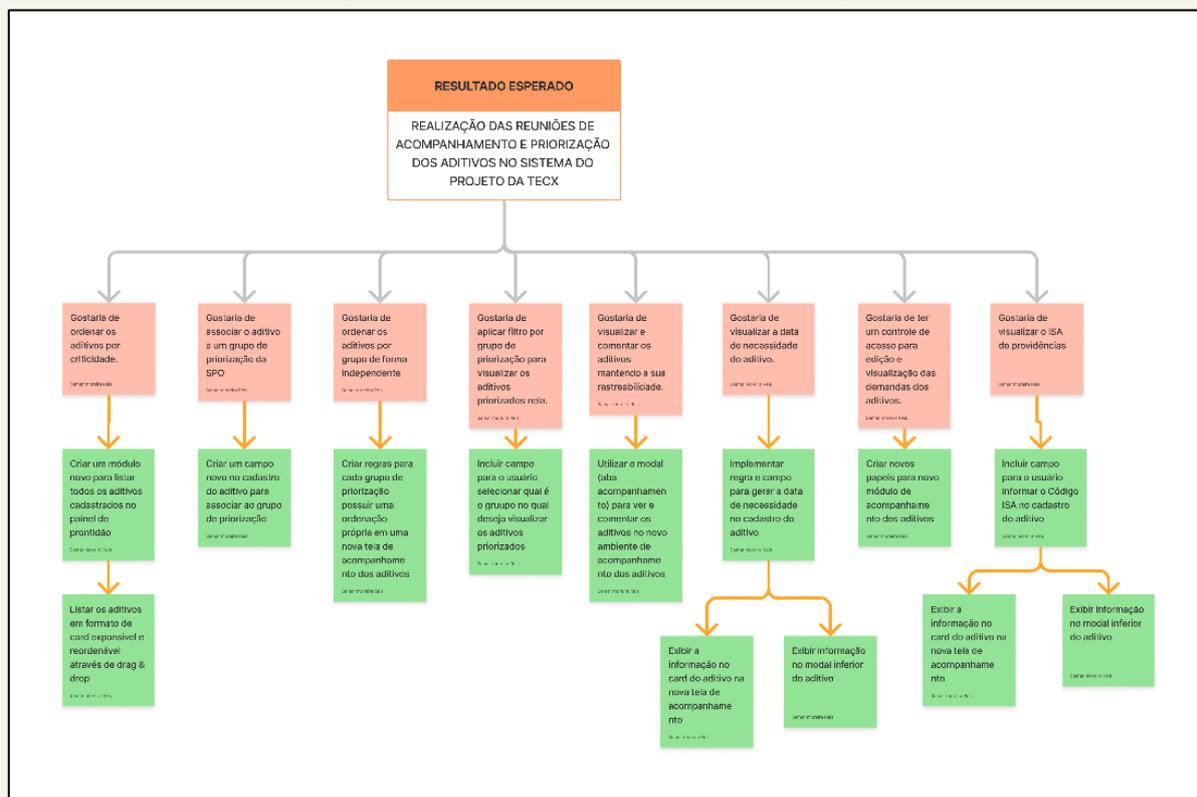
- 1. Introdução:** A brief introduction stating the document's purpose and providing links to "Diagrama de necessidades e soluções" and "Anotações".
- 2. Declaração do problema:** A structured problem statement:
  - Nós observamos que** **houve um aumento do quantitativo de processos de aditivos em andamento**
  - o que faz com que** **os diligenciadores de contratação**
  - sofram com** **a necessidade de priorizar a atuação das equipes naqueles aditivos mais críticos para a área de serviço de poço.**
- 3. Objetivos do sistema:** A comparison between client and system objectives:
  - OBJETIVO DA ÁREA CLIENTE:** Orientar os processos de análise e consulta da prontidão dos bens e serviços da área de poços de petróleo para a execução das intervenções previstas no cronograma ou sinalizadas via proposta de alteração do mesmo. Ademais, orientar a execução do processo de gestão técnica dos contratos para serviços de poços marítimos.
  - OBJETIVO DO SISTEMA:** Permitir um melhor controle dos processos de contratação, dando previsibilidade quanto à sua conclusão e disponibilização dos bens e serviços contratados a fim de evitar atrasos que prejudiquem o cumprimento do cronograma de construção de poços. Ademais, objetiva-se integrar o planejamento e a programação das contratações a fim de dar transparência e flexibilidade a estes processos.

Fonte: Elaborado pelos autores.

Com a declaração do problema, um plano de pesquisas de especificação foi traçado e executado gerando fatos, *insights* e necessidades sobre o acompanhamento dos aditivos, bem

como geradas e organizadas as propostas de solução dessas necessidades em um diagrama de necessidades e soluções, conforme Figura 8.

Figura 8. Diagrama de necessidades e soluções.



Fonte: Elaborado pelos autores.

De maneira geral, as soluções direcionaram para o desenvolvimento de um novo módulo no sistema existente. Consequentemente, a especificação e validação de requisitos culminou no desenvolvimento do protótipo representado na Figura 9. Na atividade de validação do protótipo e dos requisitos, foram apresentadas as telas do protótipo simulando o uso da solução proposta. Esta apresentação foi feita pelo engenheiro de requisitos para todos os membros da equipe colaborativa de especificação. Além do protótipo, diversos requisitos funcionais e não funcionais foram especificados e validados.

Figura 9. Protótipo da tela inicial do acompanhamento e priorização dos aditivos.

Item	Título	Tipo de aditivo	Alcance	Código ISA	Necessidade	Assinatura	Desvio	Diligenciador
1*	AB - SURCO - TAMPÃO MECÂNICO E KIT DE REPARO 5K - PRORROGAÇÃO PRAZO E REMANEJAMENTO	Tampão Mecânico e Kit reparo	31/10/2022	PBR-2021-0051538	31/10/2022	10/11/2022	-10 dias	Samar Moreira Reis
2*	AB - SURCO - TAMPÃO MECÂNICO E KIT DE REPARO 5K - PRORROGAÇÃO PRAZO E REMANEJAMENTO	Tampão Mecânico e Kit reparo	31/10/2022	PBR-2021-0051538	31/10/2022	11/10/2022	+20 dias	Samar Moreira Reis
3*	AB - SURCO - TAMPÃO MECÂNICO E KIT DE REPARO 5K - PRORROGAÇÃO PRAZO E REMANEJAMENTO	Tampão Mecânico e Kit reparo	31/10/2022	PBR-2021-0051538	31/10/2022	31/10/2022	0 dias	Samar Moreira Reis
4*	AB - SURCO - TAMPÃO MECÂNICO E KIT DE REPARO 5K - PRORROGAÇÃO PRAZO E REMANEJAMENTO	Tampão Mecânico e Kit reparo	31/10/2022	PBR-2021-0051538	31/10/2022	10/11/2022	-10 dias	Samar Moreira Reis
5*	ADITIVO	Tampão Mecânico e Kit reparo						

Fonte: Elaborado pelos autores.

Na etapa de priorização, uma lista de tarefas de desenvolvimento foi elaborada. Esta lista foi extraída com base no documento final de integração de requisito, priorizada e implementada através do desenvolvimento do código necessário, realização de testes em ambientes de homologação e geração de nova versão do sistema no ambiente de produção. Essa implementação durou 1,5 mês e correspondeu a 1,5 *sprints*.

## 4.5. Avaliação dos resultados

A avaliação do método restringe-se à sua aplicação em um estudo de caso específico. Ainda que se reconheça essa limitação, os autores entendem que a situação não compromete a relevância do artefato desenvolvido. A implementação do Método ASAP no estudo de caso durou cerca de um mês, e a solução que foi desenvolvida a partir de sua aplicação demandou aproximadamente um 1,5 mês de desenvolvimento. No total, foram necessários 2,5 meses até a liberação ao cliente de uma nova versão do sistema com a solução proposta. Para avaliar a extensão em que o método cumpriu as suas funções pretendidas, foram realizadas entrevistas

com os membros da equipe colaborativa de especificação, que participaram do estudo de caso e empregaram o artefato em suas atividades.

A natureza visual do processo de construção dos requisitos contribuiu para a visibilidade e aplicação do método. Contudo, o *feedback* dos participantes sobre a rapidez com que o projeto avançou, em comparação com as expectativas baseadas em modelos anteriores, também foi um fator importante. A natureza visual do método permitiu a criação de um fluxo iterativo e dinâmico. O método reduziu o trabalho envolvido na produção e revisão de requisitos. Houve melhoria qualitativa na precisão dos requisitos, reduzindo a complexidade acidental e simplificando o trabalho de mudança. Essa redução de trabalho agilizou o processo de projeto, o que implicou em menor custo para o cliente e retorno mais rápido de seu investimento, já que a solução se tornou utilizável mais cedo do que em situações prévias.

Segundo os participantes consultados, os principais pontos fortes incluem a integração entre as características ágeis e uma visão unificada da equipe, promovendo uma reflexão mais aprofundada sobre problemas e soluções. Os principais pontos fracos destacados foram a necessidade de habilidades adicionais para gerar cenários de teste, além da incerteza sobre a aplicabilidade do artefato ciclos de desenvolvimento mais curtos, como os menores que trinta dias, especialmente em casos nos quais haverá demanda por mais tempo e disponibilidade da equipe para tratar de requisitos que não são facilmente modeláveis, como os não-funcionais.

Por fim, ainda considerando uma perspectiva qualitativa, a modelagem de requisitos foi considerada válida, especialmente a partir dos *feedbacks* positivos recebidos dos entrevistados, indicando a eficácia do Método ASAP. Foi possível perceber uma melhoria na comunicação da equipe, que conseguiu estabelecer objetivos mais precisos, bem como progredir de modo ágil na elaboração da especificação final. Obviamente, apesar de os resultados da validação inicial serem positivos, certamente ainda há espaço para melhorias significativas. O método, porém, é flexível e adaptável, permitindo ajustes. Especialmente no teste de avaliação, demonstrou capacidade para sucesso, mesmo considerando as restrições do negócio. Sua adaptação, replicação e uso recorrente são plausíveis. No entanto, recomenda-se testes e avaliações mais amplos, diversificados e aprofundados em outras organizações, contextos de negócios, sistemas e problemas, especialmente em situações que envolvam projetos com ciclos de desenvolvimento inferiores a trinta dias.

## 5. Considerações finais

O objetivo principal desta trabalho foi desenvolver e demonstrar um método para conduzir as atividades de engenharia de requisitos no desenvolvimento de software. Este método visa integrar as características e perspectivas dos métodos ágeis, oferecendo orientações práticas para profissionais da área em seus próprios projetos. Foi realizada uma pesquisa bibliográfica sobre engenharia de software e requisitos, explorando seus conceitos, características e componentes. Com base na metodologia *Design Science Research*, foi desenvolvido um método para conduzir atividades de engenharia de requisitos em projetos de desenvolvimento de software. Baseado em perspectivas e características ágeis, esse método tomou como ponto de partida um modelo informal existente, sendo desenvolvido e aprimorado tomando por base premissas que possibilitaram sua validação em um contexto aplicado e prático.

O resultado foi uma evolução de várias iterações entre o modelo informal original e o desenvolvimento do método, denominado Acelerado Desenvolvimento de Requisitos de Software em Processos Ágeis (ASAP). Após sua elaboração, o método passou por um teste em um estudo de caso prático. É importante observar que a validação da solução resultante foi realizada pelos *stakeholders* e implementada em um sistema de software em contínua evolução. No entanto, é importante ressaltar que a validação verificada não pode ser generalizada apenas com base nesse estudo de caso. Embora promissor, o método requer investigações e aplicações adicionais para uma validação mais abrangente e confiável, o que se sugere para pesquisas futuras, em diferentes contextos. O Método ASAP pode orientar profissionais em busca de aprimorar as atividades de engenharia de requisitos, resultando em softwares de alta qualidade e alinhados às necessidades dos usuários. Além disso, serve como um exemplo da aplicação e construção de artefatos empregando a metodologia *Design Science Research*. Assim, este trabalho contribui fornecendo diretrizes para profissionais conduzirem de modo bem-sucedido suas atividades de engenharia de requisitos em projetos de software.

## Referências

ABES. **Mercado Brasileiro de Software**: panorama e tendências. 2022. Disponível em: <https://abes.com.br/download/51169/>. Acesso em: 27 dez. 2022.

AKEN, J. E. V. Management Research Based on the Paradigm of the Design Sciences: The Quest for Field-Tested and Grounded Technological Rules. **Journal of Management Studies**, v. 41, n. 2, p. 219-246, Março 2004. DOI: <https://doi.org/10.1111/j.1467-6486.2004.00430.x>.

BECK, K.; BEEDLE, M.; VAN BENNEKUM, A.; COCKBURN, A.; CUNNINGHAM, W.; FOWLER, M. et al. **Manifesto for agile software development**. 2011. Disponível em: <http://agilemanifesto.org/>. Acesso em: 28 fev. 2021.

CAPUANO, E. D. **Metodologie e strumenti dell'Ingegneria dei Requisiti della piattaforma MLA**. Tese (Mestrado em Engenharia da Computação), Università degli Studi di Napoli Federico II, 2011. Disponível em: <https://www.yumpu.com/it/document/read/15109202/metodologie-e-strumenti-dellingegneria-dei-requisiti-mobilab>. Acesso em: 28 dez. 2021.

DAVIS, A. M. **Software requirements**: Analysis and specification. Hoboken: Prentice Hall Press, 1994.

ELLIOTT, R. A. **Software requirements elicitation, verification, and documentation**: an ontology-based approach. Tese (Doutorado em Filosofia), Mississippi State University, 2012. Disponível em: <https://scholarjunction.msstate.edu/cgi/viewcontent.cgi?article=4980&context=td>. Acesso em: 10 JUN. 2022.

GRENNING, J. **Planning poker or how to avoid analysis paralysis while release planning**. Hawthorn Woods: Renaissance Software Consulting, 2002. Disponível em: <https://sewiki.iai.uni-bonn.de/media/teaching/labs/xp/2005a/doc.planningpoker-v1.pdf>. Acesso em 28 jun. 2021

HIGHSMITH, J. **Agile Software Development Ecosystems**. Boston: Addison-Wesley, 2002.

HULL, E.; JACKSON, K.; DICK, J. **Requirements Engineering**. 2. ed. Heidelberg: Springer-Verlag London, 2005.

IEEE STD 610.12-1990. **IEEE Standard Glossary of Software Engineering Terminology**. 1990. Disponível em: <https://doi.org/10.1109/IEEESTD.1990.101064>. Acesso em: jan. 2022.

LINDVALL, M. et al. Empirical Findings in Agile Methods. In: **Proceedings of extreme programming and agile methods - xp/agile universe**. Chicago. [s.n.]. p. 197 – 207, 2002. Disponível em: [http://dx.doi.org/10.1007/3-540-45672-4\\_19](http://dx.doi.org/10.1007/3-540-45672-4_19). Acesso em: 28 jan. 2022.

MATÉ, J. L.; SILVA, A. **Requirements Engineering for Social-Technical Systems**. Hershey: Information Science Publishing, 2005.

MATHESON, D. M. **SAMEM: A Methodology for the Elicitation and Specification of Requirements for Agile Model-driven Engineering of Large Software Solutions**. Tese (Doutorado em Ciências Naturais), RWTH Aachen University, 2019. Disponível em: <https://www.se-rwth.de/phdtheses/Diss-Matheson-A-Methodology-for-the-Elicitation-and-Specification-of-Requirements-for-Agile-Model-driven-Engineering-of-Large-Software-Solutions.pdf>. Acesso em 02 jul. 2021.

NATO. **The NATO Software Engineering Conference**. Periodical Co, US, 1968. Disponível em: [http://1stmuse.com/sw/the\\_nato\\_software\\_engineering\\_conference\\_1968/](http://1stmuse.com/sw/the_nato_software_engineering_conference_1968/). Acesso em: 4 fev. 2021.

PEFFERS, K. et al. A design science research methodology for information systems research. **Journal of management information systems**, v. 24, n. 3, p. 45 – 77, 2007. Disponível em: <https://doi.org/10.2753/MIS0742-1222240302>. Acesso em: 15 fev. 2021.

PMI. **Guia do conhecimento em gerenciamento de projetos**. 6. ed. Newtown Square: Project Management Institute, Inc, 2017.

PRESSMAN, R. S. **Engenharia de Software: Uma abordagem profissional**. 7. ed. New York: The McGraw-Hill Companies, Inc., 2011.

SHOEMAKER, D. P. et al. Evaluating Return On Investment For Software Process Improvement Projects. **Journal of Business & Economics Research**, v. 1, n. 8, p. 45 – 55, Fevereiro 2011. DOI: <https://doi.org/10.19030/iber.v1i8.3037>.

SIMON, H. A. **The Science of the Artificial**. 3. ed. Cambridge: The MIT Press, 1996.

SOMMERVILLE, I. **Software Engineering**. 9. ed. São Paulo: Person Prentice Hall, 2011.

SOMMERVILLE, I.; SAWYER, P. **Requirements Engineering: A Good Practice Guide**. Chichester: Wiley, 1997.

STAMELOS, I. G.; SFETSOS, P. **Agile software development quality assurance**. Hershey: Yurchak Printing Inc, 2007.

THE STANDISH GROUP. **CHAOS Report 2015**. 2015. Disponível em: [https://www.standishgroup.com/sample\\_research\\_files/CHAOSReport2015-Final.pdf](https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf). Acesso em: 13 dez. 2021.

VAISHNAVI, V.; KUECHLER, W.; PETTER, S. **Design Research in Information Systems**. 2004. Disponível em: <http://www.desrist.org/design-research-in-information-systems>. Acesso em: 3 ago. 2022.

VLIET, H. V. **Software Engineering: Principles and Practice**. 3. ed. Hoboken: Wiley, 2007.

WAGNER, S. et al. Requirements Engineering Practice and Problems in Agile Projects: Results from an International Survey. In: **20th iberoamerican congress of software engineering**

(cibSE). Buenos Aires, Argentina, 2017. Disponível em:  
<https://doi.org/10.48550/arXiv.1703.08360>. Acesso em: 15 dez. 2021.

WIEGERS, K.; BEATTY, J. **Software Requirements**. 3. ed. Redmond: Microsoft Press, 2013.

YOUNG, R. R. **The Requirements Engineering Handbook**. Norwood: Artech House, Inc., 2004.