

**ANALYSIS OF PBB COMPATIBILITY WITH THE EXPECTED RESULTS OF  
THE MPS.BR REQUIREMENTS DEVELOPMENT PROCESS**

**ANÁLISE DA COMPATIBILIDADE DO PBB AOS RESULTADOS ESPERADOS  
DO PROCESSO DE DESENVOLVIMENTO DE REQUISITOS DO MPS.BR**

**Sandro Ronaldo Bezerra Oliveira** ; <https://orcid.org/0000-0002-8929-5145> UFPA -  
Universidade Federal do Pará

**Jamilli Ynglid Carmo da Cunha** ; <https://orcid.org/0000-0002-0426-0606>  
UFPA - Universidade Federal do Pará



## ANALYSIS OF PBB COMPATIBILITY WITH THE EXPECTED RESULTS OF THE MPS.BR REQUIREMENTS DEVELOPMENT PROCESS.

**ABSTRACT:** The software industry in Brazil has seen a high growth rate. For this reason, software development companies have been adopting agile methodologies, aiming at speeding up their processes. However, the country's software market requires a quality model of the software industry, so it is necessary that the quality model and agile methodologies "talk" to each other in order to maintain quality and speed in the production of software. This paper aims to present a mapping between the MPS.BR software quality model through the expected results of the requirements development process, contained in the MPS.BR Implementation Guide, with the requirements construction steps of the agile methodology Product Backlog Building (PBB). This research deals exclusively with requirements engineering, as it is the main pillar in the construction of a project. To obtain the data for this research, a bibliographical study of the mentioned methodologies was carried out, the analysis of characteristics and similarities between them and, finally, the evaluation of the study by a specialist to consolidate the mapping. The results were satisfactory and will be detailed throughout this paper.

**Keywords:** MPS.BR, PBB, Requirements Engineering.

## ANÁLISE DA COMPATIBILIDADE DO PBB AOS RESULTADOS ESPERADOS DO PROCESSO DE DESENVOLVIMENTO DE REQUISITOS DO MPS.BR

**RESUMO:** A indústria de software no Brasil presenciou uma alta taxa de crescimento. Por esta razão as empresas de desenvolvimento de software têm feito a adoção de metodologias ágeis, visando rapidez em seus processos. Entretanto o mercado de software do país exige um modelo de qualidade da indústria de software, logo mostra-se necessário que o modelo de qualidade e as metodologias ágeis “conversem” entre si para que se mantenha a qualidade e a rapidez na produção de um software. Este artigo tem como objetivo apresentar um mapeamento entre o modelo de qualidade de software MPS.BR por meio dos resultados esperados do processo de desenvolvimento de requisitos, contidos no Guia de Implementação do MPS.BR, com as etapas de construção de requisitos da metodologia ágil *Product Backlog Building* (PBB). Esta pesquisa trata exclusivamente da engenharia de requisitos, por ser o pilar principal na construção de um projeto. Para obtenção dos dados desta pesquisa foi feito um estudo bibliográfico das metodologias citadas, a análise de características semelhanças entre elas e, por fim, a avaliação do estudo por parte de um especialista para consolidar o mapeamento. Os resultados se mostraram satisfatórios e serão detalhados no decorrer deste artigo.

**Palavras-chave:** MPS.BR, PBB, Engenharia de Requisitos.

**Agradecimentos:** Este trabalho pertence ao projeto SPIDER/UFPA (<http://www.spider.ufpa.br>).

## 1. INTRODUÇÃO

A Indústria de Software e Serviços de Tecnologia da Informação e Comunicação (ISSTIC) no Brasil registrou uma produção estimada em US\$ 53,3 bilhões em 2021, valor que responde por 82,8% do total dos serviços produzidos pelo setor de TIC (Tecnologia da Informação e Comunicação) e aponta para um crescimento de 6,5% em relação ao observado no ano anterior. Os dados integram o relatório da Indústria de Software e Serviços de TIC no Brasil, lançado pelo Ministério da Ciência, Tecnologia e Inovações (MCTI) e elaborado pela equipe de pesquisadores do Observatório Softex, unidade de estudos e pesquisas da entidade (AGENCIA BRASIL, 2021 apud SOUSA, 2022).

Dado o crescimento na demanda de produção de software no país, é natural que as empresas de desenvolvimento de software busquem aprimorar sua forma de desenvolver adotando práticas que agilizem o processo e não gere impacto na qualidade do produto.

O MPS-BR ou Melhoria de Processos do Software Brasileiro, é um programa de qualidade de processo criado em 2003 pela Softex (Associação para Promoção da Excelência do Software Brasileiro) para melhorar a capacidade de desenvolvimento de software nas empresas brasileiras (SOFTEX, 2003). Seu Guia de Implementação é um manual que tem como objetivo orientar as empresas sobre como construir um software com qualidade, desde a etapa de levantamento de requisitos até a validação do produto por parte do cliente.

Em virtude da alta demanda do mercado, inúmeras empresas têm adotado a utilização de metodologias ágeis, pelo fato de ser uma abordagem para acelerar entregas de um determinado projeto. Ela consiste basicamente em ciclos menores de entrega para o cliente e com isso problemas inesperados podem ser corrigidos com rapidez.

Então, surge a pergunta: de que modo as empresas podem agilizar a construção de seus produtos sem impactar na qualidade que o mercado exige (MPS.BR) e ainda atender a alta demanda? Alinhado a este questionamento, sabe-se que o Guia de Implementação do MPS.BR também se trata de um documento prescritivo.

As empresas, por meio dos métodos ágeis, além da velocidade buscam o foco no cliente final e na entrega de valor. Entretanto, o mercado exige o modelo de qualidade do MPS.BR. Logo, é válido que uma relação precisa ser estabelecida entre o modelo de qualidade do MPS.BR e as metodologias ágeis existentes. Assim, as empresas podem agilizar seus processos sem gerar impacto na qualidade do produto.

Uma das etapas de construção de um software é o levantamento e análise de requisitos do cliente, pois se trata da base para iniciar o projeto. A engenharia de requisitos é um processo que engloba todas as atividades que contribuem para a produção de um documento de requisitos e sua manutenção ao longo do tempo (DEVMEDIA, 2013).

Desta forma, o objetivo deste artigo é comparar os resultados esperados do processo de desenvolvimento de requisitos (DRE) do MPS.BR com a metodologia ágil PBB (*Product Backlog Building*), ou seja, analisar de que modo esta metodologia atende ao modelo de qualidade do MPS.BR sobre os resultados esperados do DRE.

Este trabalho se encontra organizado da seguinte forma: a Seção 2 descreve a fundamentação teórica utilizada nesta pesquisa; a Seção 3 apresenta a metodologia de pesquisa utilizada; a Seção 4 apresenta os resultados obtidos na análise do mapeamento das práticas do PBB ao MPS.BR; e, por fim, a Seção 5 apresenta as conclusões dessa pesquisa e possíveis trabalhos futuros.

## **2. FUNDAMENTAÇÃO TEÓRICA**

### **2.1. O MPS.BR**

MPS.BR (Melhoria de Processo de Software Brasileiro) é um programa de referência, que tem como objetivo melhorar a qualidade nos processos de desenvolvimento de software e para isso tem como proposta resultados simples de serem implementados. Ele é indicado para organizações de pequeno e médio porte cujo poder de investimento nesta área é limitado (SANTANA, TIMÓTEO & VASCONCELOS, 2006).

O modelo de referência do MPS.BR tem o objetivo de auxiliar no gerenciamento do processo das organizações e ele não define como realizar determinadas tarefas, mas enumera as diretrizes que devem ser seguidas pelas empresas.

O Modelo de Referência MPS para Software baseia-se nos conceitos de maturidade e capacidade de processo para a avaliação e melhoria da qualidade e produtividade de software e serviços correlatos e também para a melhoria da qualidade e produtividade dos serviços prestados (SOFTEX, 2016, p. 5).

De acordo com a SOFTEX (2016), o MPS.BR é formado por um conjunto de guias: Guia Geral de Software, que define o modelo de referência em um âmbito geral; o Guia Geral de Serviços, que contém componentes e as definições comuns necessárias para o entendimento e aplicação de serviços; o Guia de Gestão de Pessoas, que descreve os componentes e as definições comuns necessárias para seu entendimento e aplicação da gestão de pessoas; o Guia de Aquisição, que descreve os processos que tratam de aquisição de software; o Guia de Avaliação que fornece as metas para avaliação; e o Guia de Implementação que define os níveis de maturidade que conseqüentemente mapeiam a evolução dos processos e classificam estes processos segundo seus requisitos de maturidade. Neste estudo somente o Guia de Implementação será utilizado.

### **2.2. O Processo de Desenvolvimento de Requisitos (DRE)**

O propósito do processo Desenvolvimento de Requisitos é definir os requisitos do cliente, do produto e dos componentes do produto (SOFTEX, 2016).

Requisitos são a base de todo projeto de software. Um requisito é uma característica do sistema ou a descrição de algo que o sistema é capaz de realizar para atingir os seus objetivos (PFLEEGER, 2004).

De acordo com o SWEBOK (IEEE, 2004), o Desenvolvimento de Requisitos inclui os seguintes passos:

- Elicitação de requisitos: identificação de forma proativa dos requisitos por meio de entrevistas, questionários, prototipação, etc;
- Análise e negociação de requisitos: exame dos requisitos coletados e negociação com os envolvidos, caso haja requisitos conflitantes;
- Especificação e Modelagem dos requisitos: documentação e criação de modelos dos requisitos com o propósito de obter uma melhor compreensão do problema a ser solucionado;
- Validação de requisitos: exame da especificação para garantir que inconsistências, omissões e ambiguidades tenham sido detectadas e corrigidas.

Após a identificação, os requisitos devem ser modelados para se obter uma melhor compreensão do produto a ser desenvolvido. O modelo dos requisitos deve focar naquilo que o produto deve fazer, não em como ele o faz.

Os resultados esperados do Processo de desenvolvimento de requisitos são 8 no total, cada um deles foca em uma etapa para gerar o documento com os de requisitos do cliente.

**DRE1 - As necessidades, expectativas e restrições do cliente, tanto do produto quanto de suas interfaces, são identificadas.**

Envolve a utilização de métodos adequados para identificar necessidades, expectativas, restrições e interfaces do cliente. Deve-se buscar o envolvimento de representantes do cliente e utilizar técnicas de elicitação de requisitos para identificar de forma proativa requisitos adicionais não discutidos explicitamente pelos clientes.

**DRE2 - Um conjunto definido de requisitos do cliente é especificado e priorizado a partir das necessidades, expectativas e restrições identificadas**

As necessidades, expectativas e restrições do cliente identificadas anteriormente são traduzidas em requisitos do cliente e priorizadas.

**DRE3 - Um conjunto de requisitos funcionais e não-funcionais, do produto e dos componentes do produto que descrevem a solução do problema a ser resolvido, é definido e mantido a partir dos requisitos do cliente.**

O alcance deste resultado esperado compreende a consolidação das necessidades, expectativas e restrições do cliente em um conjunto de requisitos funcionais e não-funcionais do produto e dos componentes do produto.

**DRE4 - Os requisitos funcionais e não-funcionais de cada componente do produto são refinados, elaborados e alocados**

Significa elaborar os requisitos funcionais e não-funcionais de cada componente do produto nos termos técnicos necessários para o desenvolvimento do produto e dos componentes do produto. Para refinar os requisitos funcionais e não-funcionais podem ser utilizadas técnicas de modelagem como especificação de casos de uso de negócio, modelos de contexto (SOMMERVILLE, 2003), entre outras.

**DRE5 - Interfaces internas e externas do produto e de cada componente do produto são definidas**

As interfaces internas e externas do produto e de cada componente do produto devem ser especificadas e documentadas de acordo com a arquitetura definida do produto. As definições das interfaces geralmente são definidas em termos de tipos e formatos de dados de entrada e saída entre os componentes do produto e entre elementos do sistema, especificações de protocolos de comunicação, entre outros.

**DRE6 - Conceitos operacionais e cenários são desenvolvidos**

Exige o desenvolvimento de conceitos operacionais e cenários para o produto e os componentes do produto. Um cenário é uma sequência de eventos possível de ocorrer no uso de um produto e é utilizado para tornar explícitas algumas necessidades dos envolvidos (CMMI Product Team, 2010).

**DRE7 - Os requisitos são analisados, usando critérios definidos, para balancear as necessidades dos interessados com as restrições existentes**

Visa garantir que os requisitos, em seus diferentes níveis, sejam analisados de forma a balancear as necessidades dos interessados com as restrições de projeto existentes. Os requisitos podem ser analisados juntamente com cenários, conceitos operacionais e definições detalhadas dos requisitos, para determinar se eles são necessários, corretos, testáveis e suficientes para atingir os objetivos e requisitos de alto nível (requisitos do cliente) (CMMI Product Team, 2010). Para a análise de balanceamento entre necessidades e restrições podem ser utilizados modelos, simulações, protótipos e avaliações de riscos nos requisitos e na arquitetura funcional.

### **DRE8 - Os requisitos são validados**

O objetivo é garantir que os requisitos sejam validados utilizando-se técnicas adequadas, de forma que o produto terá o desempenho adequado quando instalado no seu ambiente alvo. A validação aumenta a confiança de que os requisitos definidos são capazes de guiar o desenvolvimento satisfatoriamente. Quanto mais cedo problemas forem identificados, menos retrabalho e custo serão necessários para adequar os requisitos às expectativas do cliente.

De modo geral o processo de desenvolvimento de requisitos segue o seguinte fluxo: as necessidades, expectativas, restrições e interfaces do cliente são levantadas e traduzidas em requisitos do cliente; posteriormente, os requisitos do cliente são refinados e descritos em termos técnicos, originando os requisitos funcionais e não-funcionais do produto e dos componentes do produto; na sequência, uma definição desses requisitos, bem como dos cenários e conceitos operacionais requeridos, também devem ser elaborados em um nível de detalhe que permita a realização de projetos (*design*) técnicos e a construção da solução do software para resolver o problema em questão; por fim, os requisitos devem ser analisados, validados e gerenciados ao longo do ciclo de desenvolvimento ou de manutenção de um produto (SOFTEX, 2016).

### **2.3. Metodologias Ágeis**

Em fevereiro de 2001 um grupo de profissionais extraordinários do desenvolvimento de software reuniu-se em um Resort de Ski em Wasatch Range para discutir melhores maneiras de desenvolver softwares. Esse encontro deu origem ao manifesto ágil, vide Figura 1, uma declaração com os princípios que regem o desenvolvimento ágil (GOMES, 2013, p. 2).

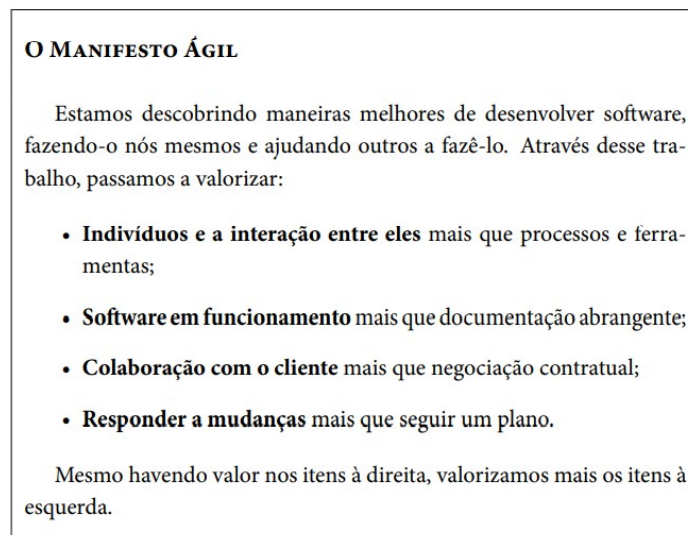


Figura 1 – Manifesto Ágil  
**Fonte:** Gomes (2013).

O Manifesto ágil é o embasamento filosófico de todos os métodos ágeis, e diversos métodos de desenvolvimento de software estão alinhados a ele. A maioria deles se utiliza de ciclos curtos, que são chamados de iterações e normalmente têm duração de poucas semanas, dessa forma garantindo *feedback* frequente e respostas rápidas às mudanças. Cada iteração contém todas as etapas necessárias para que se realize um incremento no produto: planejamento, análise, design, codificação, testes e documentação. Cada um deles

traz uma abordagem diferente que inclui diversos valores, práticas e reuniões. O Scrum, por exemplo, traz abordagem mais voltada para a gestão, com maior foco nas reuniões, no planejamento e na melhoria contínua. Já o XP (*eXtreme Programming*) traz maior enfoque nas práticas técnicas (GOMES, 2013, p. 4).

## 2.4. SCRUM

Scrum é um *framework* ágil para o desenvolvimento, entrega e sustentação de produtos complexos. Foi inicialmente proposto para projetos de desenvolvimento de software, mas tem sido aplicado para qualquer âmbito de projetos complexos e trabalhos inovadores. É especialmente adequado para projetos com requisitos que mudam rapidamente ou são altamente emergentes. O desenvolvimento de software com Scrum progride através de uma série de iterações chamadas de Sprints, que duram, tipicamente, de uma a quatro semanas (CAROLI e AGUIAR, 2020, p. 2).

As regras do Scrum integram os papéis, artefatos e eventos, administrando as relações e interações entre eles (SUTHERLAND e SCHWABEHTPS, 2013, p. 3). Os Times Scrum são auto-organizáveis e multifuncionais, são compostos pelos seguintes papéis:

- *Product Owner* (PO): representa o negócio, os clientes ou usuários, e orienta a equipe para a construção do produto certo. Ele deve liderar o esforço de desenvolvimento por meio de esclarecimentos e prioridades sobre o trabalho. Tipicamente, o PO é a única pessoa responsável por gerenciar o *Backlog* do Produto (CAROLI e AGUIAR, 2020, p. 4);
- Scrum Master: é alguém experiente com o *framework* que pode ajudar o time a usar o processo do Scrum para alcançar objetivos de alto nível (CAROLI e AGUIAR, 2020, p. 4);
- Time de Desenvolvimento: é auto-organizado e multifuncional, com todas as habilidades necessárias para criação do incremento do produto (CAROLI e AGUIAR, 2020, p. 4).

Segundo Espinha (2020), o Scrum prescreve quatro eventos formais, contidos dentro dos limites da Sprint, para inspeção e adaptação:

- Reunião de planejamento da Sprint: tem por objetivo definir duas coisas: 1) o que será entregue ao final da Sprint, e 2) como será o trabalho ao longo dela;
- Reunião diária: ao longo da execução, a equipe se reúne diariamente para avaliar o andamento das tarefas e compartilhar seus progressos e suas dificuldades;
- Reunião de revisão da Sprint: quando a Sprint termina, é realizada uma reunião para inspecionar os resultados obtidos e adaptar o *Product Backlog*, se for o caso;
- Retrospectiva da Sprint: enquanto a revisão busca avaliar o produto e o trabalho da equipe, a retrospectiva é uma oportunidade de a equipe avaliar a si mesma. É o momento em que os colaboradores podem refletir suas práticas e prospectar as melhorias a serem aplicadas.

Os artefatos do Scrum representam o trabalho ou o valor para o fornecimento de transparência e oportunidades para inspeção e adaptação (SUTHERLAND e SCHWABEHTPS, 2013, p. 13). Os artefatos são:

- *Backlog* do Produto: é uma lista ordenada de tudo que deve ser necessário no produto, e é uma origem única dos requisitos para qualquer mudança a ser feita no produto. O *Product Owner* é responsável pelo *Backlog* do Produto, incluindo seu conteúdo, disponibilidade e ordenação (SUTHERLAND e SCHWABEHTPS, 2013, p. 13);

- *Backlog* da Sprint: é um conjunto de itens do *Backlog* do Produto selecionados para a Sprint de forma ordenada (SUTHERLAND e SCHWABEHTTIPS, 2013, p. 15);
- Incremento: é a soma de todos os itens do *Backlog* do Produto completados durante a Sprint e o valor dos incrementos de todas as Sprints anteriores (SUTHERLAND e SCHWABEHTTIPS, 2013, p. 15).

## 2.5. Product Backlog Building (PBB)

Scrum é um *framework* para desenvolver, entregar e manter produtos complexos que adota como ponto de partida o *Product Backlog*, entretanto o *framework* Scrum não define como construir o *backlog*.

Segundo CAROLI e AGUIAR (2020), o *Product Backlog Building* (PBB) é uma metodologia ágil e um canvas que complementa o Scrum na elaboração e criação de um *Product Backlog* efetivo e colaborativo, sendo esse seu principal objetivo.

O PBB esclarece e prioriza as *User Stories*, os itens a serem adicionados ao *backlog* do time Scrum. O método sugere que o time Scrum preencha junto o canvas PBB, como resultado os participantes alcançam um entendimento comum do produto, e obtém o *backlog* para começar a trabalhar, de forma ágil e eficaz (CAROLI e AGUIAR, 2020, p.16).

Segundo CAROLI e AGUIAR (2020), os principais benefícios do PBB podem incluir:

- Ajudar na construção de um *BACKLOG* de uma forma efetiva e colaborativa;
- Construir um entendimento compartilhado do negócio do cliente, facilitando a descoberta e compreensão do produto;
- Buscar uma maneira de descrever a experiência do usuário com o produto;
- Facilitar a descoberta e escrita de *User Stories*.

O PBB é representado por um canvas chamado de PBB Canvas, vide Figura 2, que tem um fluxo bem simples e de fácil compreensão, principalmente para facilitar o entendimento do cliente, pois sua participação é de suma importância nesse processo de construção.

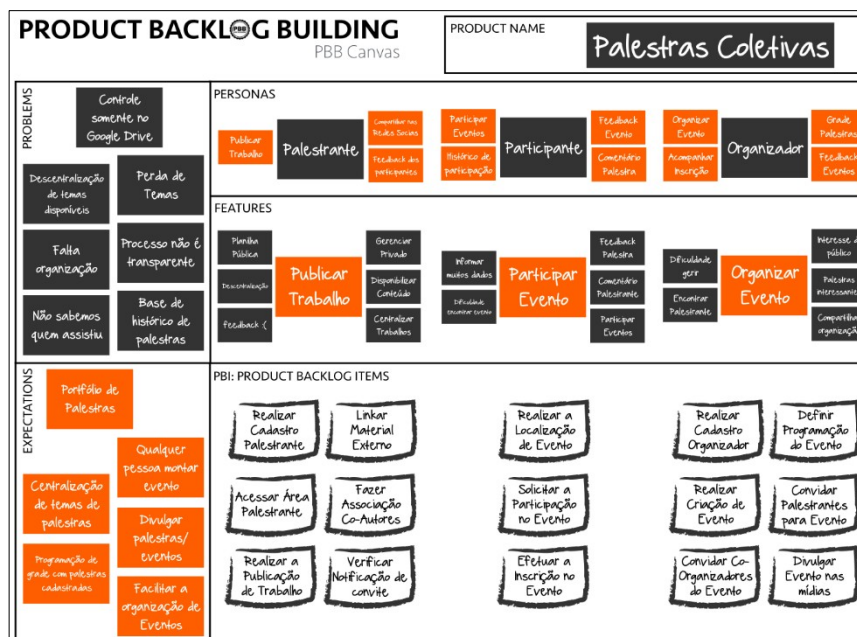


Figura 2 – Exemplo de um PBB Canvas  
Fonte: CAROLI e AGUIAR (2020)



Segundo CAROLI e AGUIAR (2020), o fluxo de construção do *backlog* no PBB segue os seguintes passos:

### Passo 1 - Contextualize o Produto

- **Product Name:** A primeira etapa é identificar o produto que será construído;
- **Problems:** O ponto de partida é identificar e compreender o Estado Atual pontuando um conjunto de problemas. Neste momento, as pessoas de produto e envolvidos do negócio buscam de forma colaborativa a mesma compreensão do estado atual, pontuando os problemas que desejam que sejam resolvidos. É importante conhecer o problema antes de criar a solução;
- **Expectations:** É importante identificar o Estado Desejado, alinhando suas expectativas aos problemas do estado atual, para que, de uma forma compartilhada, todos os envolvidos possam alinhar suas expectativas;
- **Personas:** Saiba quem são os usuários, papéis e responsáveis envolvidos no produto e saiba o que faz e o que espera sobre o produto;
- **Features:** Identifique as funcionalidades, vide Figura 3, que cada persona realiza no produto, mapeando na sequência de uso da esquerda para a direita. Descreva a funcionalidade com uma breve descrição, sempre pontuando os “Problemas” e os “Benefícios” de cada funcionalidade.

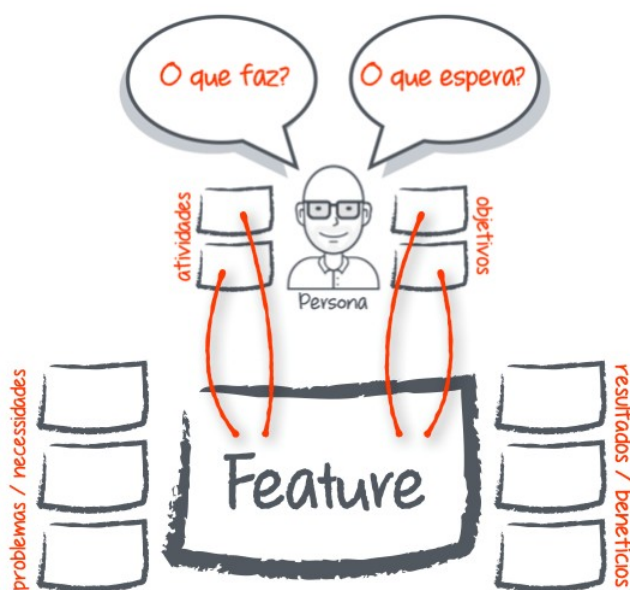


Figura 3- Ilustração do processo de criação de uma feature.

Fonte: CAROLI e AGUIAR (2020)

Finalizando as etapas, para cada passo da *feature*, escreva os PBI (*Product Backlog Item*) que os satisfaçam. No primeiro momento escreva usando o modelo ARO (ação - resultado - objeto) e em seguida pode-se representar como *user stories*.

### Passo 2 - Steps Maps

De acordo com CAROLI e AGUIAR (2020), a quebra de *feature* é feita através do *Steps Maps*, vide Figura 4, é ele que mapeia os passos de uma funcionalidade, de uma *feature*. No primeiro momento, é definido o fluxo de trabalho passo a passo, e no segundo momento deve-se evoluir com perguntas, comentários e idéias. No final cada passo representará um item do *backlog*.

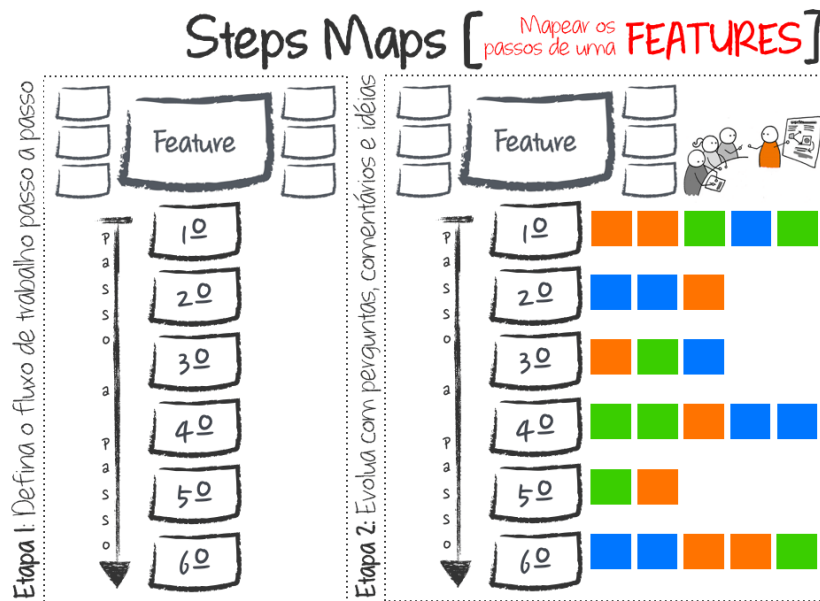


Figura 4 - Steps Maps etapas 1 e 2  
**Fonte:** CAROLI e AGUIAR (2020)

Agora é só priorizar o *backlog* e dar todo direcionamento de trabalho para o time, aplicando uma técnica de priorização.

### **Passo 3 - Priorização dos PBI**

COORG é um acrônimo para Classificar, Ordenar e ORGANizar (CAROLI e AGUIAR, 2020, p.32). O COORG auxilia a equipe a priorizar os PBI dentro do *backlog*. Inicialmente, é necessário que se estabeleça com a equipe os parâmetros para classificação. Deve-se decidir com o grupo quais os critérios com sua respectiva escala, conforme o contexto do produto. Um exemplo de critérios de classificação seria frequência de uso + valor de negócio.

Essa é a dinâmica do *Product Backlog Building*. O preenchimento do Canvas PBB segue uma sequência de passos simples: Product Name > Problems > Expectations > Personas > Features > PBI

O fluxo de uma forma linear ajuda a organizar a visão geral do negócio e alinhar o valor de negócio. A ferramenta *PBB Canvas* deixa toda a concepção do produto organizado de forma visual.

## **3. METODOLOGIA DA PESQUISA**

O objetivo deste artigo é comparar os resultados esperados do processo de desenvolvimento de requisitos (DRE) do MPS.BR com a metodologia ágil PBB (*Product Backlog Building*), ou seja, analisar de que modo esta metodologia atende ao modelo de qualidade do MPS.BR, considerando os resultados esperados do DRE.

A análise apresentada neste artigo é fruto de uma pesquisa que se iniciou com uma revisão do modelo de qualidade do MPS.BR tendo por base o Guia de Implementação – Parte 4: Fundamentação para Implementação do Nível D do MR-MPS-SW:2016 disponibilizado pelo portal da Softex. Partindo do princípio que a área de interesse é a engenharia de requisitos, uma leitura profunda foi realizada sobre o Processo de Desenvolvimento de Requisitos com ênfase nos seus resultados esperados. A primeira etapa de mapeamento foi feita por meio de uma síntese sobre as características de cada

resultado esperado, destacando palavras-chave de cada resultado esperado, enfatizando os procedimentos que devem ser efetuados e que artefato é criado ou incrementado após a aplicação de cada processo.

No segundo passo, um estudo da metodologia PBB foi feito por meio do livro “Product Backlog Building: Um guia prático para criação e refinamento de backlog para produtos de sucesso sobre a metodologia”. O autor trata a engenharia de requisitos a partir do capítulo “O PBB Canvas e o Processo de Construção do Backlog”. Revisando as etapas de criação do PBB que são: nome do produto, problemas, expectativas, personas, features e PBI, assim como no MPS.BR uma síntese foi escrita enfatizando as características de cada processo e qual artefato é gerado ou incrementado ao final de cada etapa.

Durante o estudo do PBB foi necessária a familiarização com dois termos em especial, metodologias ágeis e Scrum. O PBB é uma metodologia ágil, logo é vital a compreensão de tal termo. O livro *Agile*, da Casa do Código, descreve no capítulo 1- introdução às metodologias ágeis, sobre o que foi o manifesto ágil e os diversos métodos de desenvolvimento de software que estão alinhados a ele, como o Scrum. O PBB também é um complemento da metodologia Scrum, e no livro “Product Backlog Building” os autores fazem um *overview* sobre o Scrum. Os estudos sobre o Scrum foram especializados no manual do Scrum disponibilizado no site do Scrum Guides.

Concluída as sínteses, o passo seguinte foi analisar as características semelhantes em ambos os modelos para formar o mapeamento. O processo de mapeamento contou com o auxílio de uma tabela, onde a primeira coluna listava todos os resultados esperados do DRE. A segunda coluna tratava a sobre as etapas correspondentes do PBB em relação ao resultado esperado. Para preencher esta coluna, tendo por base as sínteses realizadas, comparou-se as características de cada resultado esperado com as etapas do PBB. Identificada a relação, a etapa era preenchida na coluna e linha do resultado esperado correspondente. Por fim foi adicionada uma última coluna chamada justificativa, a qual contém uma descrição relatando a razão dessas etapas se relacionarem.

Em sequência, foi feita a revisão por pares do material produzido junto a um implementador e avaliador oficial do MPS.BR. Ao apresentar a tabela, o avaliador detectou pontos que necessitavam de ajustes. Os pontos apontados foram no mapeamento do dos resultados esperados DRE5, DRE7 e DRE8, com foco na justificativa apresentada.

O DRE5 trata de identificar as interfaces do produto. A justificativa apresentada foi que o *Step Map* se adequava ao DRE5 pelo fato de fornecer uma interface interna entre os DRE1 e o DRE2, ou seja, existe uma interface entre as necessidades e os requisitos do cliente. O PBB não trata de interfaces externas, porém, segundo o avaliador, esta afirmação está incorreta, a interface que o PBB trata é entre os PBI de uma *feature*.

O DRE7 não se aplica ao PBB, porque não balanceia, não gera um histórico de versão. O avaliador sugeriu um melhor desenvolvimento da justificativa esclarecendo o termo balancear.

O DRE8 é um resultado esperado que se adequa à Pessoa Facilitadora do PBB. A justificativa apresentada é que a validação é um processo que avalia se os requisitos estão de acordo com as necessidades do cliente. No PBB esse processo pode ser feito pela pessoa que desempenha o papel de facilitador do PBB, como o PO, o gerente de produto ou analista de requisitos durante um evento de validação. O avaliador sugeriu um complemento na justificativa, informando que, embora ocorra a participação do facilitador do PBB neste processo, não há instrumentos dentro do PBB para tal critério, seria necessário a adoção de um instrumento como os critérios INVEST (*Independent, Negotiable, Valuable, Estimable, Small, Testable*).

#### 4. MAPEAMENTO DAS PRATICAS DO PBB AO MPS.BR

Após a finalização do mapeamento das práticas do PBB ao MPS.BR obtivemos os seguintes resultados:

O DRE1 refere-se às **necessidades, expectativas e restrições do cliente, tanto do produto quanto de suas interfaces, que devem ser identificadas**. No PBB essa pratica se espelha na Contextualização do Produto. Segundo o manual de implementação do MPS.BR, no DRE1 acontece o contato inicial com o cliente, nesse momento suas necessidades, expectativas e restrições são levantadas para dar início as etapas iniciais de construção do produto. No PBB o DRE1 se espelha na etapa de Contextualização do Produto, pois é nela que será definido o nome, as expectativas do cliente, suas necessidades em relação ao produto, além da identificação dos possíveis problemas.

O DRE2 refere-se a **um conjunto definido de requisitos do cliente que devem ser especificados e priorizados a partir das necessidades, expectativas e restrições identificadas**. No PBB essa pratica se adequa à definição e descrição dos *personas*, a definição das *features*, ao processo de *Step Map* e ao COORG - Classificação dos PBI. O DRE2 na prática é a tradução das necessidades do cliente em uma linguagem técnica. A partir desse ponto, essas necessidades são chamadas de requisitos do produto. No PBB através das necessidades do cliente, inicialmente, são identificados os **personas** do produto, o qual é a representação de um usuário do sistema que descreve não somente o seu papel, mas também as suas necessidades no sistema. Definidos os *personas*, cada ação ou interação dele com o sistema é descrito por meio de uma **feature**, ou seja, a *feature* nomeia a ação do usuário, como efetuar o login no sistema por exemplo. As ações dos *personas* precisam ser detalhadas assim como no DRE2 e a técnica **Step Map** é um processo que quebra a *feature* em passos menores. O objetivo do *Step Map* é mapear sequencialmente as ações do usuário gerando o PBI. Formado o *Product Backlog*, executa-se o **COORG** (acrônimo para Classificar, Ordenar e ORGANizar), por ser uma técnica que auxilia a equipe a priorizar os PBI do *Product Backlog* para o planejamento de uma *Sprint* (*Time Box* dentro do qual um conjunto de atividades deve ser executado). Um exemplo de classificação das PBI seria o uso do critério “frequência de uso + valor de negócio”.

É válido ressaltar que a partir do DRE3 algumas etapas do PBB se encaixam para o atendimento em mais de um resultado esperado.

O DRE3 refere-se a **um conjunto de requisitos funcionais e não-funcionais, do produto e dos componentes do produto que descrevem a solução do problema a ser resolvido, deve ser definido e mantido a partir dos requisitos do cliente**. No PBB essa pratica se adequa também com a criação de *features* e a descrição das ações do PBI. No DRE 3 ocorre a definição dos requisitos funcionais e não funcionais do produto. Os requisitos funcionais descrevem as funções ou serviços que se espera que o sistema forneça, descreve a interação entre o sistema e seu ambiente é focado no que será feito, enquanto que os não funcionais descrevem como serão feitos e que restrições devem ser obedecidas, por exemplo: a linguagem que o produto será desenvolvido, qual o tempo de resposta máximo do sistema, etc. Os requisitos funcionais referentes ao DRE3 se enquadram na definição de **features**, pois, como já mencionado, a *feature* nomeia uma ação do usuário assim como o PBI que é um item que compõe o *Product Backlog*, ou seja, descreve um requisito funcional. O PBB não faz o tratamento de requisitos não funcionais, pelo fato da metodologia, durante o processo de construção dos requisitos, focar nas ações e interações do produto com o usuário, ao invés de restrições relacionadas à construção do software.

O DRE4 refere-se aos **requisitos funcionais e não-funcionais de cada componente do produto devem ser refinados, elaborados e alocados**. No PBB essa

prática se adequa ao processo Step Map, ao PBI e, por fim, ao Product Backlog. O DRE4 é a etapa em que os requisitos do cliente são refinados. É comum a utilização de técnica de modelagem por meio de diagramas para detalhar o fluxo e funcionamento do conjunto de requisitos. Outra prática comum é a de categorizar os requisitos em grupos por meio de critérios, como por exemplo: propósitos similares, dependência funcional. O **Step Map** é a técnica que se espelha de maneira direta em relação ao DRE4, pois o refinamento das features é feito por meio desse processo, ou seja, a feature é desmembrada para detalhar o requisito, assim como em uma técnica de modelagem é utilizado um diagrama como um recurso visual para detalhar o requisito. Os **PBI** também estão intimamente ligados, pois o PBI descreve a sequência de passos para a formação de uma feature, o conjunto dessas features formará o Product Backlog. O PBI possui uma descrição textual refinada, detalhada de uma ação para: primeiro para dar contexto e segundo identificar unicamente o item. Por exemplo: “Login básico”, “Consultar partidas próximas a um endereço informado”. A primeira é uma breve descrição textual sem seguir um modelo, a segunda segue o modelo ARO (ação|resultado|objeto). Vale ressaltar que, enquanto no DRE2 o PBI é tratado de modo individual, no DR4 o conjunto de PBI organizados sequencialmente vão fornecer o refinamento dos requisitos. Nesse ponto já temos a formação do **Product backlog**, onde os requisitos já estão definidos, refinados e priorizados.

O DRE5 trata das **interfaces internas e externas do produto e de cada componente do produto que devem definidas**. No PBB essa prática se adequa ao Step Map e ao PBI. O DRE5 faz o tratamento das interfaces internas e externas do produto. Interface é o nome dado para o modo como ocorre a comunicação entre duas partes distintas e que não podem se conectar diretamente. No PBB é possível visualizar interfaces internas entre os PBI de uma feature após o processo de **Step Map**, por se tratar de um detalhamento sequencial em termos de prioridade. Ao decompor uma feature em PBI, nota-se que uma PBI está ligada a outra, um exemplo seria a feature “Fazer compra on-line”, onde pergunta-se “quais seriam os passos desta feature?”, tem-se como resposta “Fazer uma consulta sobre o produto”, “Fazer a seleção do produto”, “Fazer o pagamento do produto”, assim por diante. Note que todas essas ações são distintas e não se conectam diretamente, pois a segunda ação depende da primeira e assim por diante. Essa relação evidencia a ligação dos PBI, a interface interna entre elas. O PBB não trata de interfaces externas, então não atende por completo este resultado esperado.

O DRE6 trata dos **conceitos operacionais e cenários que devem ser desenvolvidos**. O DRE6 está ligado à prototipação, modelagem e criação de cenários para verificar como determinado requisito irá funcionar quando o sistema for desenvolvido. O PBB não se aplica ao DRE6 pelo fato de tratar somente do levantamento dos requisitos.

O DRE7 apresenta que os **requisitos devem ser analisados, usando critérios definidos, para balancear as necessidades dos interessados com as restrições existentes**. O DRE7 trata do balanceamento, do histórico de alterações produto. Balancear é o ato de revisar, melhorar e evoluir o produto de modo que, cada vez que esse procedimento é executado, uma nova versão do produto é criada, ou seja, é gerado um histórico. O DRE7 não se aplica ao PBB, pois não existe uma prática explícita na metodologia para histórico de alterações.

No DRE8 **os requisitos devem ser validados**. No PBB essa prática se assemelha à pessoa facilitadora do PBB. A validação é um processo que avalia se os requisitos estão de acordo com as necessidades do cliente. No PBB esse processo pode ser feito pela pessoa que desempenha o papel de facilitador do PBB, como: o PO (que representa o negócio, os clientes e usuário do produto), o gerente de produto ou o analista de requisitos durante um evento de validação. Embora ocorra a participação do facilitador do PBB neste processo,

não há instrumentos dentro do PBB para tal critério, onde seria necessária a adoção de um instrumento como os critérios INVEST por exemplo.

## 5. CONCLUSÃO

A engenharia de requisitos, é um fator chave na produção de software, pois sem ela é impossível iniciar um projeto. Muitas empresas de software têm buscado aprimorar sua forma de desenvolver, adotando práticas ágeis que não gerem impacto negativo na qualidade de seus produtos. Entretanto, o Brasil possui um modelo de qualidade que na prática não é ágil. Logo, a análise contida neste artigo é uma pesquisa relevante, pois é visível a necessidade de um mapeamento para que esse objetivo seja alcançado.

Ao analisar os resultados esperados do processo de desenvolvimento de requisitos (DRE) com as práticas do *Product Backlog Building* é visível a semelhança entre vários pontos de ambos os modelos. Em alguns casos as etapas do PBB se adequavam em mais de um resultado esperado, o que valida a agilidade do PBB na engenharia de requisitos. Isso é muito bom, pois trabalhar de forma ágil sugere ser mais rápido, prático e fácil, uma vez que as metodologias ágeis presam pela interação dos indivíduos, colaboração com o cliente, o software em funcionamento e resposta às mudanças. Essas práticas tornam o processo de desenvolvimento de requisitos mais satisfatório para os envolvidos na construção do projeto que, por fim, impacta na qualidade do software.

O mapeamento também mostrou algumas limitações: o PBB não se aplica em determinados pontos do modelo de qualidade, como nos casos dos resultados esperados DRE6 e DRE7; também não faz tratamento a requisitos não funcionais. Nesse caso esta pesquisa pode contribuir com uma possível adaptação da metodologia de modo que os pontos citados também possam ser incluídos no PBB. Além disso, outro ponto limitante é que essa pesquisa não foi testada em um ambiente corporativo.

A partir desta pesquisa, trabalhos futuros podem ser desenvolvidos como a aplicação dos resultados adquiridos no ambiente corporativo para enriquecer a pesquisa e, conseqüentemente, contribuir com a melhora na construção de requisitos.

## REFERÊNCIAS BIBLIOGRÁFICAS

AGENCIA BRASIL; SOUZA, L. **Indústria de Software e Serviços de TIC cresceu 6,5% em 2021**. Disponível em: <<https://agenciabrasil.etc.com.br/economia/noticia/2022-07/industria-de-software-e-servicos-de-tic-cresceram-65-em-2021>>. Acesso em: 08 out. 2022.

CAROLI, P.; AGUIAR, F. **Product Backlog Building Concepção de um product backlog efetivo**, 2020. Ebook.

DEVMED. **IA Primeiros passos na Engenharia de Requisitos**. Disponível em: <<https://www.devmedia.com.br/introducao-a-engenharia-de-requisitos/29454>>. Acesso em: 08 out. 2022.

ESPINHA, R. G.. **Scrum: o que é e como funciona o método ágil mais utilizado**. Disponível em: <[https://artia.com/scrum/#:~:text=1\)%20Reuni%C3%A3o%20de%20planejamento%20da,que%20as%20tarefas%20ser%C3%A3o%20distribu%C3%ADdas.>](https://artia.com/scrum/#:~:text=1)%20Reuni%C3%A3o%20de%20planejamento%20da,que%20as%20tarefas%20ser%C3%A3o%20distribu%C3%ADdas.>)>. Acesso em: 17 out. 2022.

GOMES, A. F. **Agile Desenvolvimento de software com entregas frequentes e foco no valor de negócio.** 2013, Casa do Código. Disponível em: <https://www.casadocodigo.com.br/pages/sumario-agile>. Acesso em: 19 set. 2022

SANTANA, C. A.; TIMÓTEO, A. L.; VASCONCELOS, A. M. L.. **Mapeamento do modelo de Melhoria do Processo de Software Brasileiro (MPS.BR) para empresas que utilizam Extreme Programming (XP) como metodologia de desenvolvimento.** Disponível em: <<https://sol.sbc.org.br/index.php/sbqs/article/view/15605>>. Acesso em: 10 out. 2022.

SOFTEX, 2003. **MPS.BR.** Disponível em: <<https://softex.br/mpsbr/#>>. Acesso em: 08 out. 2022.

SOFTEX, 2016. **Guia de Implementação – Parte 4: Fundamentação para Implementação do Nível D do MR-MPS-SW:2016.** Disponível em: <[https://www.softex.br/wp-content/uploads/2016/04/MPS.BR\\_Guia\\_de\\_Implementacao\\_Parte\\_4\\_2016.pdf](https://www.softex.br/wp-content/uploads/2016/04/MPS.BR_Guia_de_Implementacao_Parte_4_2016.pdf)>. Acesso em: 19 set. 2022.

SURTHERLAND, J.; SCHWABEr, K. **Guia do Scrum™ Um guia definitivo para o Scrum: As regras do jogo,** 2013. Disponível em: <<https://scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf>>. Acesso em: 19 set. 2022.