

# CRIAÇÃO DE UM MODELO DE PROCESSO HÍBRIDO APLICADO NO DESENVOLVIMENTO DE UM SOFTWARE BASEADO NA INTEGRAÇÃO DAS METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE

Welliton S. da Silva<sup>1</sup>, Gustavo Bisognin<sup>1</sup>

<sup>1</sup>Ciência da Computação – Universidade do Extremo Sul Catarinense (UNESC)  
Santa Catarina – SC – Brasil

wellitonserafim@gmail.com, gustavo@unesc.net

**Abstract.** *Hybrid solutions are composed of the particularity of each project, with the objective of defining and standardizing processes, as well as activities for each reality. This study proposes the creation of a hybrid process model applied in the development of a software, using applied methodology, with descriptive objective and procedures classified as a case study. The results obtained allowed us to observe that the hybrid methodology brought greater fluidity and objectivity in the communication between the development team, product owner and stakeholders, as well as an organizational structure for the execution of the project, in order to ensure the success of the project according to the definitions and expectations of those involved.*

**Resumo.** *As soluções híbridas são compostas pela particularidade de cada projeto, tendo por objetivo a definição e padronização de processos, bem como atividades para cada realidade. Este estudo propõe a criação de um modelo de processo híbrido aplicado no desenvolvimento de um software, sendo utilizado a metodologia aplicada, com objetivo descritivo e procedimentos classificados como estudo de caso. Os resultados obtidos permitiram observar que a metodologia híbrida trouxe maior fluidez e objetividade na comunicação entre a equipe de desenvolvimento, product owner e stakeholders, bem como uma estrutura organizacional para a execução do projeto, a fim de assegurar o sucesso do projeto de acordo com as definições e expectativas dos envolvidos.*

## 1. Introdução

As soluções de software estão cotidianamente presentes nas nossas vidas, sendo essenciais na execução das operações diárias de diversos negócios, como hospitais, escolas, atividades de lazer, segurança, entre outros. Porém, seu desenvolvimento enfrenta vários desafios como controle de custos, prazos e obtenção de qualidade do produto final.

Desta forma, pode-se entender que a engenharia de software é fundamental para o controle e qualidade do desenvolvimento destas soluções. Metodologias e processos de desenvolvimento de software são um conjunto de atividades e critérios que tem por

finalidade produzir um software de qualidade. Onde, pode-se afirmar que o reflexo do produto final está diretamente relacionado com o processo de software empregado.

Por muitos anos, os métodos tradicionais de desenvolvimento de software definiram os padrões para a criação de software no mundo. Contudo, notou-se que os casos de fracasso da indústria aumentaram, fazendo com que alguns líderes experientes contestassem, com novos modos de trabalho, o conceito das metodologias tradicionais. Assim sendo, apesar de não seguirem os padrões do mercado, suas formas de trabalho eram muito eficientes, em comparação com os modelos tradicionais, apresentando resultados mais assertivos em termos de qualidade e velocidade de desenvolvimento, o que chamou bastante a atenção dos especialistas da área. Neste contexto, inicia-se o surgimento de uma nova era na construção de soluções de software com a abordagem de modelos híbridos e mais ágeis. A comprovação destes novos modelos, vem, mais tarde com a aplicação em larga escala na construção de projetos de desenvolvimento, tornando possível o aprimoramento, e, em alguns casos, a transformação e criação de novas metodologias de desenvolvimento de software. Essas metodologias passaram a ser denominadas de ágeis.

De acordo com Pressman (2016, p. 52) “em um sistema, é importante seguir uma série de passos previsíveis, um roteiro que ajude a criar um resultado de alta qualidade e dentro do prazo estabelecido. O roteiro é denominado “processo de software”.

Na metodologia ágil, segundo Vettorazzo (2018, p. 13) “o software não é desenvolvido e disponibilizado integralmente, e sim uma série de incrementos e a cada novo incremento uma nova funcionalidade do sistema.”. Atualmente existem vários métodos ágeis, dentre eles podem-se citar os seguintes: *Scrum*, *Extreme Programming (XP)*, *Open Unified Process (OpenUP)*, *Dynamic Systems Development Method (DSDM)*, *Feature Driven Development (FDD)*, entre outros.

Com grande variedade de práticas, técnicas e ferramentas, a singularidade das organizações, e a necessidade de eficiência nos projetos, a solução mais adequada é aquela aprimorada para cada ambiente. É assim que o conceito de modelo híbrido se destaca, permitindo a customização e organização da gestão, práticas, time do projeto, relacionamento com o cliente e potenciais consumidores, já que, se basear em uma única regra ou método para pequenos ou grandes projetos pode não ser o ideal (CONFORTO, 2017).

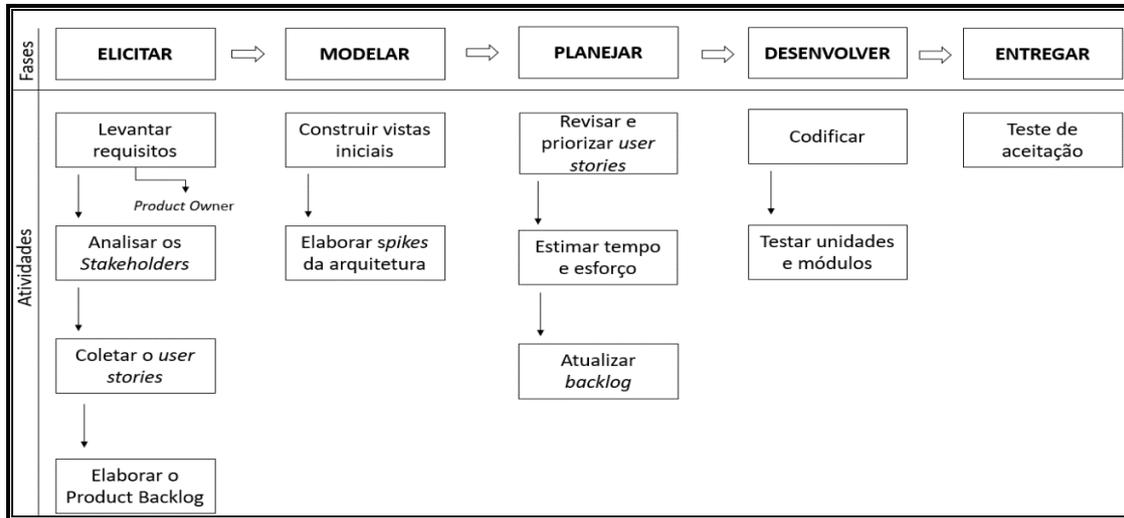
Sendo assim, a pesquisa proposta, enfatiza a criação e união das melhores práticas dos métodos de desenvolvimento de software, para a composição de um modelo híbrido, a fim de ser validado no desenvolvimento de um software.

Objetivos específicos compreenderam: levantar e analisar as metodologias de desenvolvimento de software tradicional, ágil e híbrida de projetos, compreender o conceito de gerenciamento de projeto de software, combinar métodos ágeis e tradicionais para criar um modelo de processo híbrido, desenvolver um software a fim de validar o modelo de processo híbrido.

## **2. Materiais e Métodos**

O presente trabalho se caracterizou como uma pesquisa aplicada e de base tecnológica com objetivo descritivo e procedimentos classificados como estudo de caso, a fim de

gerar conhecimentos voltados a prática. Então, ao estudar e revisar a literatura pode-se criar um modelo de processo híbrido denominado de Modelo de Processo WUP, para a aplicação das fases e atividades no processo de desenvolvimento de um software, como ilustra a Figura 1.



**Figura 1. Modelo de Processo WUP**  
Fonte: Do Autor

## 2.1. Elicitar

A fase Elicitar foi composta pelas seguintes atividades: levantamento de requisitos, que buscou compreender as ideias iniciais de quem solicitou o projeto, tendo como fonte de informação o dono do produto (em inglês, *Product Owner*). Já a análise dos eventuais *Stakeholders*, permitiu a identificação das pessoas que foram envolvidas e afetadas durante o desenvolvimento e finalização do projeto, bem como a medição de seus graus de influência. A coleta dos *User Stories* tornou o processo simplificado e objetivo com a construção de cenários que simularam as principais funções que o software precisava realizar. Todas as atividades foram documentadas a partir da elaboração do *Product Backlog*.

### 2.1.1. Levantamento de Requisitos

O levantamento de requisitos teve como critério a técnica Orientada a ponto de vista. Essa técnica por meio de uma entrevista e aplicação de questionários (Tabela 1) permitiu identificar os diferentes pontos de vista entre os *Stakeholders* e o *Product Owner*. O primeiro questionário aplicado foi o de requisitos funcionais, a fim obter o reconhecimento geral das características e funcionalidades da aplicação na visão dos *Stakeholders* e o *Product Owner*.

**Tabela 1. Questionário de Requisitos Funcionais**

Código	Questionário requisitos funcionais
QF 01	Qual é a insight / ideia a ser desenvolvida?
QF 02	O que esse recurso resolveria/mudaria no dia-dia?
QF 03	Quem é o principal público-alvo do recurso?

QF 04	Onde o usuário vai acessar o recurso?
QF 05	Qual é a principal função do recurso?
QF 06	O que esse recurso precisa fazer?
QF 07	O que se espera que o recurso também faça?
QF 08	Qual o tempo mínimo/máximo para que o recurso seja utilizado?
QF 09	O recurso efetuará cobranças? Se sim, qual o valor?
QF 10	O que impediria o usuário de usar o recurso?
QF 11	Quem seria afetado pelo resultado da interação do usuário com o recurso?
QF 12	Quando o recurso pode não ser útil ou falhar?

Fonte: Do Autor

O segundo questionário aplicado foi o de requisitos suplementares que descrevem o modelo FURPS+ (Funcional, Usabilidade, Confiabilidade, Desempenho, Suportabilidade e restrições) a fim de definir os atributos de qualidade do sistema. A Tabela 2 ilustra as perguntas realizadas.

**Tabela 2. Questionário de Requisitos Suplementares**

<b>Código</b>	<b>Questionário requisitos suplementares</b>
QF 01	Em quais sistemas o usuário terá acesso a aplicação?
QF 02	Qual linguagem de programação que deverá ser utilizada?
QF 03	Qual o banco de dados que deverá ser utilizado?
QF 04	Qual o valor estipulado para o projeto?
QF 05	Qual o tempo estipulado para o desenvolvimento?

Fonte: Do Autor

A partir das respostas dos questionários aplicado aos entrevistados durante a primeira entrevista, foi possível obter os requisitos necessários para a composição do sistema. Dentre eles, os principais, foram destacados e divididos entre as seguintes premissas: código, requisito, descrição, fonte, informações de entrada, informações de saída e restrições lógicas e tecnológicas. Na Tabela 3 é ilustrado o detalhamento dos principais requisitos.

**Tabela 3. Detalhamento do Principais Requisitos**

<b>Código</b>	<b>Detalhamento dos requisitos</b>
<b>RF 01</b>	<p><b>A aplicação dividirá os usuários entre assinantes e não assinantes.</b></p> <p><b>Descrição:</b> O usuário poderá se tornar um assinante a partir do menu “assinar”, fazendo com que obtenha acesso a mais funcionalidades do que um não assinante.</p> <p><b>Fonte:</b> <i>Product Owner</i></p> <p><b>Informações de Entrada:</b> A aplicação solicitará um CPF, data de</p>

	<p>nascimento e número de celular para envio de um código de verificação. Os planos serão divididos entre: mensal, anual, semestral. As formas de pagamento serão: cartão de crédito ou pré-pago, sendo solicitado a bandeira do cartão, número, nome, vencimento e código de segurança.</p> <p><b>Informações de Saída:</b> O usuário será classificado como “PRO” e será efetuado</p> <p><b>Restrições Lógicas e Tecnológicas:</b> O CPF, número de telefone e dados da forma de pagamento serão validados na aplicação.</p>
<p><b>RF 02</b></p>	<p><b>A aplicação permitirá que os usuários possam criar grupos.</b></p> <p><b>Descrição:</b> O usuário poderá criar grupos para realizar o palpite dos jogos, bem como convidar outros usuários.</p> <p><b>Fonte:</b> <i>Product Owner</i></p> <p><b>Informações de Entrada:</b> A aplicação solicitará o nome do grupo, descrição, o envio de um ícone e o convite aos usuários para que se tornem membros do grupo.</p> <p><b>Informações de Saída:</b> Um novo grupo é armazenado no banco de dados, um e-mail é disparado contendo um convite ao usuário e a aplicação apresenta uma listagem com os usuários que estão “pendentes” ou pertencentes ao grupo.</p> <p><b>Restrições Lógicas e Tecnológicas:</b> Usuários da versão “FREE” não poderão criar mais do que um grupo e o número de participantes será limitado a 10, sendo exibido um convite para se tornarem assinantes na tela de “Grupos”.</p>
<p><b>RF 03</b></p>	<p><b>A aplicação permitirá que os usuários possam gerenciar os grupos</b></p> <p><b>Descrição:</b> O usuário poderá alterar os grupos previamente criados.</p> <p><b>Fonte:</b> <i>Product Owner</i></p> <p><b>Informações de Entrada:</b> A aplicação permitirá a exclusão do grupo e participantes, alteração de nome, descrição, ícone e convite a novos usuários.</p> <p><b>Informações de Saída:</b> A alteração é armazenada no banco de dados e os novos dados são apresentados na tela na guia grupos.</p> <p><b>Restrições Lógicas e Tecnológicas:</b> É obrigatório nomear o grupo e criar uma pequena descrição.</p>
<p><b>RF 04</b></p>	<p><b>A aplicação permitirá que os usuários efetuem os palpites dos jogos</b></p> <p><b>Descrição:</b> O usuário poderá realizar o palpite dos jogos de todos os times que estiverem jogando no dia.</p> <p><b>Fonte:</b> <i>Product Owner</i></p> <p><b>Informações de Entrada:</b> A aplicação listará os times do dia e os horários de cada partida para que o usuário possa incluir o palpite do placar do jogo de cada jogo.</p>

	<p><b>Informações de Saída:</b> O palpite é armazenado no respectivo time e o ícone de alteração é habilitado.</p> <p><b>Restrições Lógicas e Tecnológicas:</b> Todos os palpites e alterações só poderão ser efetuados com trinta minutos de antecedência a cada jogo</p>
<b>RF 05</b>	<p><b>A aplicação mostrará o real resultado de cada jogo</b></p> <p><b>Descrição:</b> O usuário poderá alterar os grupos previamente criados.</p> <p><b>Fonte:</b> <i>Product Owner</i></p> <p><b>Informações de Entrada:</b> A aplicação permitirá a exclusão do grupo e participantes, alteração de nome, descrição, ícone e convite a novos usuários, bem como fazer.</p> <p><b>Informações de Saída:</b> A alteração é armazenada no banco de dados e os novos dados são apresentados na tela na guia grupos.</p> <p><b>Restrições Lógicas e Tecnológicas:</b> É obrigatório nomear o grupo e criar uma pequena descrição.</p>
<b>RF 06</b>	<p><b>A aplicação possuirá uma Ranking dos palpites realizados.</b></p> <p><b>Descrição:</b> Uma listagem com a posição de cada usuário poderá ser acessada do menu Ranking.</p> <p><b>Fonte:</b> <i>Product Owner</i></p> <p><b>Informações de Entrada:</b> Acerto dos palpites</p> <p><b>Informações de Saída:</b> Pontuação acumulada e posição</p> <p><b>Restrições Lógicas e Tecnológicas:</b> Usuários não assinantes se posicionam no Ranking junto com os PRO, porém não participam do sorteio mensal. Usuários PRO só se posicionam e visualizam usuários PRO no ranking.</p>
<b>RF 07</b>	<p><b>A aplicação permitirá que usuários não assinantes e PRO acumulem pontos.</b></p> <p><b>Descrição:</b> Ao acertar os palpites dos jogos todos os participantes podem acumular pontos.</p> <p><b>Fonte:</b> <i>Product Owner</i></p> <p><b>Informações de Entrada:</b> Acerto do palpite de cada jogo.</p> <p><b>Informações de Saída:</b> Inclusão de pontos no cadastro de cada usuário e posicionamento no Ranking</p> <p><b>Restrições Lógicas e Tecnológicas:</b> Apenas usuários PRO podem participar dos sorteios mensais.</p>

Fonte: Do Autor.

### 2.1.2. Análise dos *Stakeholders*

A análise dos Stakeholders foi elaborada com base nas pessoas que foram envolvidas e/ou afetadas pelo projeto a partir das seguintes premissas: determinação de quem pode

afetar o projeto, identificação dos pontos de contato de cada interessado com o projeto, identificação de como cada interessado pode ajudar e atrapalhar o andamento do projeto quantificação dos graus de poder/influência e interesse de cada interessado. A Tabela 4 ilustra algumas definições.

**Tabela 4. Descrição dos Stakeholders**

<b>Stakeholders</b>	<b>Descrição</b>	<b>Responsabilidade</b>
Patrocinadores	Pessoa ou grupo que está patrocinando o projeto	Gerenciar e Investir no Projeto
Diretores da Empresa	Pessoas que fazem parte da direção da empresa	Gerenciar o Projeto
Usuários	Pessoas que utilizarão o recurso a ser desenvolvido	Teste de aceitação do projeto
Equipe do Projeto	Pessoas que fazem parte da equipe de desenvolvimento	Gerenciar o Projeto
Funcionários	Pessoas que são colaboradores da empresa	-----

Fonte: Do Autor.

Uma vez que o sucesso do projeto depende também das pessoas que estão envolvidas nele, foi observado e definido as expectativas e interesses dos *Stakeholders*, a fim de entender o impacto que a aplicação desenvolvida gerou e/ou pode continuar gerando. A Tabela 5 ilustra estas definições.

**Tabela 5. Expectativas e Interesses dos Stakeholders**

<b>Stakeholders</b>	<b>Expectativas e Interesses</b>
Patrocinadores	Interesses econômicos do projeto; Cumprimento do prazo, custo e qualidade.
Diretores da Empresa	Interesses econômicos do projeto; Qualidade dos resultados; Satisfação dos usuários; Recursos tecnológicos; Eficiência do tempo; Aumento de clientes.
Usuários	Recursos tecnológicos; Eficiência do tempo.
Equipe do Projeto	Projeto bem sucedido.
Funcionários	Salários e Benefícios.

Fonte: Do Autor.

Com base nas expectativas e interesses dos *Stakeholders* foi desenvolvido uma matriz de grau de influência x interesse (Figura 2) onde o nível alto é composto pelas seguintes preposições: manter satisfeito e gerenciar de perto. Sendo o nível baixo por: monitorar e manter informado.



**Figura 2. Matriz de Grau de Influência x Interesse**  
**Fonte: Adaptada de PMKB 2014**

### 2.1.3. Coleta do *User Stories*

A coleta dos *user stories* foi baseado na Tabela 3 onde é especificado o detalhamento dos principais requisitos. Cada funcionalidade possui as seguintes palavras chaves: como, eu, para. E cada cenário pode possuir: dado, e, então. A finalidade foi representar o ponto de vista do usuário ao interagir com o software, contendo apenas requisitos que agregam valor de negócio para ao cliente, sem muitos detalhes técnicos.

**Tabela 6. *User Stories***

<p><b>Funcionalidade:</b> Classificação dos usuários</p> <p><b>Como</b> usuário não assinante,  <b>Eu</b> quero me tornar membro da plataforma,  <b>Para</b> ter acesso a mais funcionalidades, acumular pontos e participar do sorteio mensal.</p> <p><b>Cenário 1:</b> Assinar  <b>Dado</b> que quero assinar a plataforma,  <b>Então</b> preciso localizar o menu “Assinatura” e informar um CPF, data de nascimento, número de celular, selecionar o plano desejado, tipo do cartão, bandeira, número, nome, vencimento e código de segurança.</p> <p><b>Cenário 2:</b> Não assinar  <b>Dado</b> que eu não quero assinar a plataforma,  <b>Então</b> estou ciente que serei restringido da participação dos sorteios, criação de mais grupos e inclusão de participantes.</p>
--

**Funcionalidade:** Criar um Grupo

**Como** usuário não assinante ou PRO,

**Eu** quero criar um grupo com nome, descrição, ícone e possibilidade de convidar outros usuários,

**Para** realizar os palpites dos jogos.

**Cenário 1:** Gerenciar o Grupo

**Dado** que quero alterar as informações do grupo,

**Então** preciso localizar e clicar no ícone de edição presente na tela de Grupos.

**Cenário 2:** Criar grupos/convidar participantes

**Dado** que sou usuário não assinante,

**E** quero criar mais do que um grupo ou convidar mais do que 10 pessoas,

**Então** preciso realizar a assinatura da plataforma.

**Funcionalidade:** Posicionamento no Ranking

**Como** usuário não assinante ou PRO,

**Eu** quero visualizar minha colocação no Ranking dos jogos,

**Para** verificar a pontuação necessária a classificação.

**Cenário 1:** Acúmulo de pontos

**Dado** que eu quero estar posicionado em primeiro lugar no Ranking,

**Então** preciso acertar os palpites dos jogos.

**Cenário 2:** Participação do sorteio

**Dado** que sou usuário PRO

**E** ganhar os sorteios mensais,

**Então** preciso acertar os palpites dos jogos.

**Funcionalidade:** Listagem dos resultados dos jogos

**Como** usuário não assinante ou PRO,

**Eu** quero visualizar o resultado oficial de cada em diferentes datas,

**Para** verificar se meu palpite está correto.

**Cenário 1:** Resultado

**Dado** que eu quero visualizar os resultados

**Então** preciso acessar o menu “Resultados” e conferir a data correspondente.

**Cenário 2:** Participação do sorteio

**Dado** que sou usuário PRO

**E** ganhar os sorteios mensais,

**Então** preciso acertar os palpites dos jogos.

Fonte: Do Autor.

## 2.2. Modelar

Na segunda fase denominada de Modelagem, foi construída as vistas iniciais com a ferramenta Figma, a fim de simular na prática as funcionalidades e ilustrar melhor o projeto tanto para a equipe de desenvolvimento quanto para o dono do produto. O resultado dos *Spikes* da arquitetura teve como base os critérios de viabilidade e realizabilidade estimando também o tempo e custo do projeto antes do início de seu desenvolvimento.

### 2.2.1. Construção das Vistas Iniciais

As vistas iniciais da aplicação foram construídas utilizando a ferramenta Figma, com base nos detalhes dos principais requisitos (Tabela 3). No Apêndice A encontram-se as demais vistas desenvolvidas.



**Figura 3. Usuário sem grupo**  
Fonte: Do Autor

A Figura 3 representa o RF 02 (Tabela 3) e o *User Stories* Criar Grupo (Tabela 6), nela é possível observar um botão e mensagens que irão auxiliar o usuário a criar um grupo, bem como convidar amigos para realizar os palpites dos jogos. Já a Figura 4, representa as Restrições Lógicas e Tecnológicas da RF 02 (Tabela 3) e o Cenário 2 do mesmo *User Stories* (Tabela 6), onde após ter criado um grupo, o usuário é informado que excedeu o limite e precisa se tornar um assinante da plataforma.



**Figura 4. Limitação de grupo ao usuário não assinante**  
Fonte: Do Autor

A partir de uma segunda reunião foi possível apresentar as vistas iniciais a fim de elaborar os *Spikes* da arquitetura. O Spike proposto foi a inclusão do desenvolvimento de uma aplicação Android para se integrar o sistema Web. Os *Spikes*

foram definidos em: limite de tempo, limite de esforço, limite de orçamento e limite de possibilidade. A medição do tempo foi diretamente relacionada com o esforço necessário, baseando-se nos conhecimentos prévios da equipe de desenvolvimento em relação as linguagens de programação. O entendimento do limite de orçamento e possibilidade foi relacionado com os requisitos já disponíveis no product backlog, evitando assim um retrabalho.

### **2.3. Planejar**

No planejamento, a revisão e priorização dos *User Stories* foram alinhados com a equipe de desenvolvimento e o dono do produto, apresentando a estimativa de tempo e esforço necessários para a implementação do projeto e a atualização da documentação feita no *Product Backlog*.

### **2.4. Desenvolver**

A codificação baseou-se na técnica de programação em par, derivada do método ágil XP (*Extreme Programming*). O objetivo do Modelo WUP (Figura 1) não foi relacionado com a linguagem de programação utilizada, mas sim, em como os desafios que apareceram durante o desenvolvimento foram solucionados. Segunda a Revista Exame (2017), pesquisas indicam que um programador com experiência normalmente insere um erro a cada 10 linhas de código. A partir disso a comunicação foi fundamental entre a equipe de codificação, trocando ideias para evitar e detectar bugs ainda durante o desenvolvimento do projeto.

### **2.3. Teste de unidades e módulos**

A testagem de unidades teve por objetivo a garantia da qualidade do código, isolando-os em unidades para verificar se cada uma executa exatamente o que foi proposto. Após isso, os testes foram feitos em módulos com a integração das unidades.

O modelo WUP, segue a metodologia tradicional de ter pessoas dedicadas apenas para criar e executar os testes de unidades e módulos, mesmo estas podendo também codificar para que a troca de conhecimentos seja elevada. A ideia é que a codificação e os reajustes possam serem executadas em um ciclo contínuo, a fim de não afetar o tempo estipulado para a conclusão do projeto. Os passos dessa técnica foram executados da seguinte maneira: os testadores criaram tarefas para cada funcionalidade com base no detalhamento (Tabela 2) e *user stories* (Tabela 3), os desenvolvedores finalizaram a codificação das unidades, os testadores executaram as tarefas, as falhas foram adicionadas a um documento a ser entregue ao fim dos testes para os desenvolvedores e após as alterações o ciclo se reinicia.

### **2.3. Entregar**

Por fim, a fase de entrega compõe o teste de aceitação. O objetivo deste teste foi averiguar a funcionalidade de todo o sistema conforme as especificações do product owner. O próprio cliente foi responsável pela execução dos testes de aceitação, justamente porque é ele quem continuará utilizando o software. As etapas seguidas compreenderam: valores de entrada, saída e tempo de resposta. Em função da possibilidade de o cliente precisar de suporte, a equipe de desenvolvimento também estava presente na terceira reunião. Já nas eventuais falhas, estas foram adicionadas ao product backlog para serem implementadas na próxima iteração.

### 3. Resultados

Como resultado desse estudo descobriu-se que a abordagem de um modelo híbrido conseguiu proporcionar maior previsibilidade do projeto como um todo. As atividades e técnicas utilizadas na fase de levantamento de requisitos estruturaram o projeto. As reuniões realizadas tanto de maneira presencial quanto online utilizando a ferramenta Google Meet, resultaram na compreensão, clareza e interação entre os envolvidos, bem como a possibilidade de acompanhar o andamento do projeto com as entregas incrementais ao executar cada fase do Modelo WUP (Figura 1).

Com a revisão e priorização do Product Backlog a documentação seguiu os padrões ágeis, não sendo classificada como exacerbada ou burocrática, mas sim organizacional e simplificada, visando um rápido entendimento. A equipe do projeto demonstrou ser flexível as mudanças, já tinha conhecimento e aplicava algumas atividades ágeis nos projetos anteriores, o que facilitou a prática da técnica de programação em par e testagem das unidades e módulos, resultando na melhora da qualidade do código no processo de codificação.

Um dos maiores problemas encontrados foi o número reduzido de pessoas presentes para compor a equipe de desenvolvimento, não sendo posto em prática a divisão entre os testadores e codificadores, bem como, pessoas responsáveis apenas pela revisão do código, em par e de maneira sequencial como sugerido pelo Modelo WUP (Figura 1). Mesmo assim, isso não resultou na diminuição da interação entre a equipe. O segundo problema identificado foi a falta de uma técnica de medição de tempo e esforço necessário, baseando-se apenas na dedução a partir dos conhecimentos prévios da equipe e acordos entre o *Product Owner* e *Stakeholders*, resultando em alterações do prazo de entrega final. Apesar disso, o presente artigo teve êxito na entrega do projeto agregando valores e conhecimentos no gerenciamento de projetos de maneira organizada, interativa, com entregas contínuas em pequenos espaços de tempo objetivando a qualidade final do software.

O trabalho de Jory (2018) apresenta propósitos semelhantes ao da presente pesquisa, visando o gerenciamento de projetos a partir da combinação de diferentes abordagens tradicionais e ágeis. Para isso, foi feito um estudo de caso em organizações públicas e privadas com o intuito de identificar as práticas já utilizadas. Baseando-se na literatura foi aplicada uma solução híbrida em um espaço público afim de otimizar a entrega de serviços relacionados ao desenvolvimento de software. A metodologia consistiu na avaliação de métricas de eficiência, eficácia, qualidade, acompanhamento de nível de maturidade da equipe em relação ao projeto e identificação do modelo híbrido mais adequado. Como resultado teve-se o encerramento antes do prazo estabelecido, melhora na qualidade de desenvolvimento de software, bem como conhecimentos práticos de gerenciamentos de projetos.

O Artigo de Dias (2018), método híbrido de gestão do escopo de projetos de TIC, realizou um estudo de caso em um *startup* Brasileira, a partir de um formulário com perguntas e respostas, teve-se como objetivo melhorar os processos de gerenciamento já realizados, utilizando a combinação de práticas tradicionais e ágeis, com recomendações da autora. Para isso, toda a área de TI atou separada em times *Scrum*, o *Product Owner* produziu os *user-stories*, foi utilizado *sprints*, os desenvolvedores e testadores aturam durante o desenvolvimento, ao final de cada *sprint* foi realizado a revisão e a retrospectiva das *Sprints*. A empresa já aplicava o método de

desenvolvimento em *Scrum*, e contava com o apoio de práticas de outros métodos de gerenciamento. Obtendo os seguintes resultado: conjunto de práticas híbridas de gestão, composto por práticas ágeis que flexibilizaram o escopo do projeto, juntamente com práticas tradicionais que controlam o escopo do projeto, questionários de levantamento de dados que possibilitaram a organização e compreensão compartilhada e um conjunto de recomendações que podem ser utilizadas para implementar as possíveis melhorias em suas práticas.

O artigo de Eloi (2016), apresenta um estudo comparativo entre metodologias ágeis no desenvolvimento de aplicativos móveis. Para isso, foi feita uma análise crítica das principais metodologias ágeis presentes na literatura que podem ser utilizadas para melhorar e facilitar o ciclo de desenvolvimento de software voltados para dispositivos móveis. Também, foi realizado um *survey* com desenvolvedores de software móvel que utilizam metodologias ágeis. O objetivo do *survey* foi comparar a literatura e a prática sobre a utilização de metodologias ágeis no desenvolvimento móvel. Como resultado foi possível observar que os métodos ágeis são considerados ideais, pela sua rápida resposta às mudanças, previsão de interação constante com o cliente além de entregas rápidas, que garantem um alto grau de satisfação do cliente. Sendo a principal dificuldade encontrada a interação constante com o cliente, onde é sugerido uma pessoa da equipe para representar o cliente, tomar as decisões por ele, tanto para aprovar o que está sendo feito quanto para solicitar mudanças que sejam mais adequadas às necessidades dele.

Conforto e Amaral (2016), fizeram um estudo que relata uma análise empírica de uma estrutura de gerenciamento híbrido combinado com o gerenciamento ágil e um modelo *stage-gate* implementado em um projeto baseado em tecnologia. Os resultados indicaram um impacto positivo no desempenho de projetos e desenvolvimento de produtos, trazendo precisão das informações, comprometimento e liderança. Sendo a combinação dessas abordagens uma solução para o gerenciamento de projetos de inovação em empresas de alta tecnologia. Alguns fatores que favorecerão o uso de técnicas e ferramentas foram: características da equipe, competência das pessoas, cultura organizacional, estrutura e recursos disponíveis, incerteza tecnológica e características do mercado. Os autores relataram que um dos desafios no desenvolvimento de estruturas híbridas e na adoção de práticas ágeis em outros setores foi entender as condições, características da organização, projeto e time, a fim de estabelecer um “equilíbrio” entre elas.

#### **4. Conclusão**

Este trabalho criou e aplicou uma metodologia híbrida denominada de Modelo de Processo WUP no desenvolvimento de um software. A combinação de metodologias já existentes estruturou e formalizou o processo realizado, se mostrando compatível com a realidade do projeto.

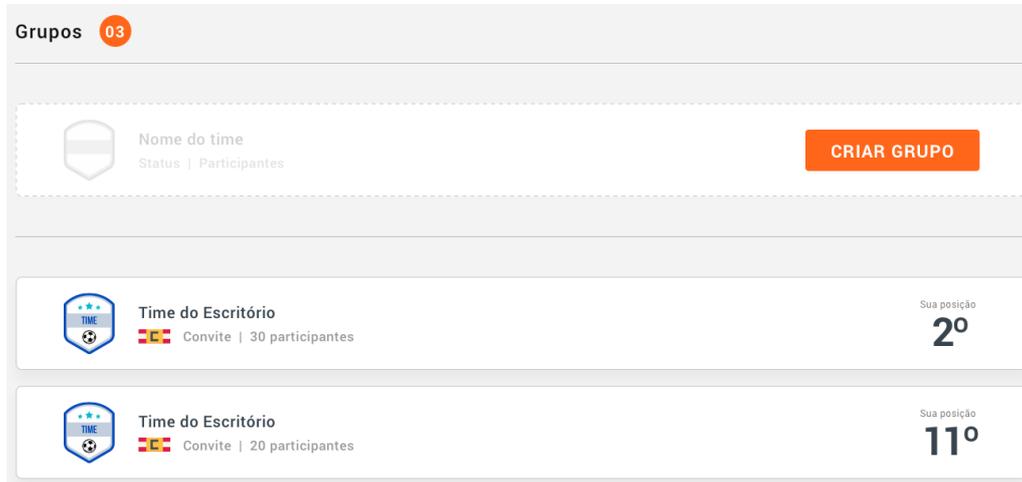
Os requisitos foram coletados utilizando técnicas que não necessitavam de um entendimento certo de todo o projeto, fazendo com que a revisão e priorização se tornassem partes importantes do processo. A flexibilidade a mudanças e interação dos envolvidos proporcionou maior qualidade ao software desenvolvido. Os resultados demonstraram que a metodologia híbrida estruturou o projeto, definindo e organizando sua execução, bem como trazendo fluidez e maior comunicação entre o time de desenvolvimento, product owner e stakeholders, para que o projeto alcançasse êxito.

Como trabalho futuro sugere-se: definição papéis e quantidade na composição de uma equipe de desenvolvimento, bem como execução completa da técnica de programação em par e testagem de unidades e módulos, e por fim, utilização de técnicas para estimular tempo e esforço também durante a criação dos *Spikes* da Arquitetura.

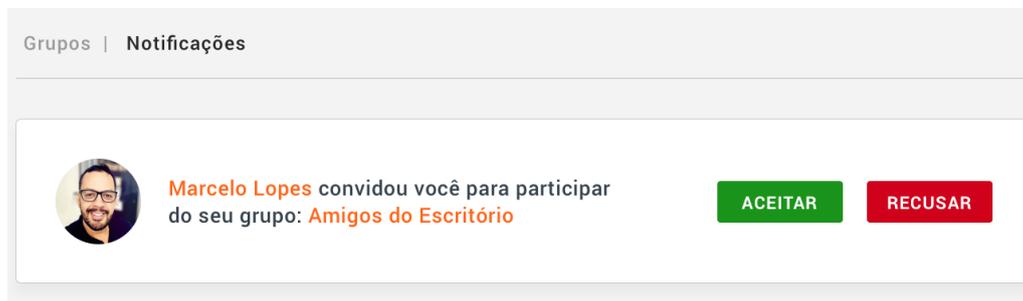
## Referências

- CONFORTO, Edivandro e AMARAL, Daniel (2016). “Agile project management and stage-gate model—A hybrid framework for technology-based companies”. *Journal Of Engineering And Technology Management*, São Carlos, p. 1-74.
- CONFORTO, Edivandro. (2017). “O Hibridismo como Abordagem para a Gestão de Projetos no Mercado Publicitário”. São Paulo: Instituto Mestre. Disponível em: [https://www.mestregp.com.br/wp-content/uploads/2017/05/hibridismo\\_1.pdf](https://www.mestregp.com.br/wp-content/uploads/2017/05/hibridismo_1.pdf). Acesso em: 03 ago. 2020.
- DIAS, Katia Regina Santos (2018). “Método híbrido de gestão do escopo de projetos de TIC: estudo de caso em uma empresa brasileira”. 182 f. TCC (Graduação) - Curso de Engenharia de Computação, Instituto de Pesquisas Tecnológicas do Estado de São Paulo, São Paulo, Disponível em: [http://cassiopea.ipt.br/teses/2018\\_EC\\_Katia\\_Dias.pdf](http://cassiopea.ipt.br/teses/2018_EC_Katia_Dias.pdf). Acesso em: 03 ago. 2020.
- ELOI, Daniel Sulman de Albuquerque (2016). “Um estudo comparativo de metodologias ágeis no desenvolvimento de aplicativos móveis”. 49 f. TCC (Graduação) - Curso de Ciência da Computação, Centro de Informática, Universidade Federal de Pernambuco, Recife.
- JORY, Natan Martins (2018). “Métodos Híbridos para Gerenciamento de Projetos de Desenvolvimento de Software”. Curso de Engenharia de Telecomunicações, Instituto Federal de Santa Catarina, São José. Disponível em: [https://wiki.sj.ifsc.edu.br/wiki/images/9/9a/TCC290\\_Natan\\_Martins\\_Jory.pdf](https://wiki.sj.ifsc.edu.br/wiki/images/9/9a/TCC290_Natan_Martins_Jory.pdf). Acesso em: 03 ago. 2020.
- PRESSMAN, Roger S. (2016). “Engenharia de Software: uma abordagem profissional”. 7. ed. São Paulo: Amgh.
- VETTORAZZO, A. de S. (2018) “Engenharia de Software”. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788595026780/cfi/0!/4/4@0.00:32.6>. Acesso em: 24 jun. 2020.

## Apêndice A



**Figura 5. Grupos do usuário PRO**  
Fonte: Do Autor



**Figura 6. Notificação para participar de um grupo**  
Fonte: Do Autor

Grupos | Grupo do Escritório



**Time do Escritório**  
Vamos ver quem vai ganhar a rodada!  
Grupo aberto | 30 participantes



**Marcelo Lopes**  
Capitão do Grupo

Ranking     Semanal     Mensal     Anual

Participantes	Pontuação	Posição
 <span>Abel Gerra</span> <b>PRO</b>	1.200	1º
 <span>Marcelo Lopes</span>	1.200	2º
 <span>Adriana Simon</span> <b>PRO</b>	900	3º

**Figura 7. Visão geral do grupo Pro/Grátis**  
Fonte: Do Autor

Domingo (21/07) ☺    **Segunda (22/07) ⚡**    Quarta (24/07) ⚡    Quinta (25/07) ☺

🕒 17:00 Palpite válido até 22/07/2019 – 16:00

Série A – Fazer palpite SAO  - 1 - 	<input checked="" type="checkbox"/> Pontos dobrados	<input type="button" value="ENVIAR PALPITE"/>
Série B – Fazer palpite FOR  - - / - + 	<input type="checkbox"/> Pontos dobrados	<input type="button" value="ENVIAR PALPITE"/>
Série A – Palpite realizado BAH  1 × 2 	<input type="checkbox"/> Pontos dobrados	<input type="button" value="✎"/> <input type="button" value="🗑️"/>

**Figura 8. Vista do usuário PRO**  
Fonte: Do Autor