

Universidade de Brasília - UnB  
Faculdade UnB Gama - FGA  
Engenharia de Software  
Melhoria de Processo de Software  
Grupo Os 7

# **Modelo IDEAL**

Fases de Inicialização e Diagnóstico

Diego Barbosa da Mota França  
Gabriel de Souza Clímaco  
Mateus de Oliveira Barbosa  
Thiago Ribeiro  
Vinícius de Castro Cantuária

Brasília, DF  
2019

# Sumário

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Inicialização</b>	<b>4</b>
2.1	Começando . . . . .	4
2.2	Identificação das Necessidades de Negócio e Diretrizes de Melhoria	6
2.3	Construindo uma Proposta de Melhoria de Processo de Software . .	7
2.4	Suporte à Educação e Construção . . . . .	10
2.5	Obter aprovação da MPS e Recursos Iniciais . . . . .	11
2.6	Estabelecer infraestrutura da MPS . . . . .	12
2.7	Avaliar o clima para a MPS . . . . .	15
2.8	Definir metas gerais da MPS . . . . .	16
2.9	Definir os Princípios Orientadores do Programa SPI . . . . .	17
2.10	Lançamento do modelo . . . . .	18
<b>3</b>	<b>Diagnóstico</b>	<b>19</b>
3.1	Referencial Teórico do Diagnóstico . . . . .	19
3.2	<i>Baseline</i> dos Processos . . . . .	23
3.2.1	Estrutura da organização e Responsabilidades de seus Com- ponentes . . . . .	23
3.2.2	Processos Usados pela Equipe de Gamificação . . . . .	24
3.3	Planejamento de Projeto . . . . .	29
3.4	Acompanhamento do Projeto . . . . .	31
3.5	Gerência de Requisitos . . . . .	33
3.6	Gerenciamento de Subcontrato de Software . . . . .	34
3.7	Gerenciamento de Riscos . . . . .	36
3.8	Garantia de Qualidade de Software . . . . .	37
3.9	Gerenciamento de Configuração de Software . . . . .	39
3.10	Planilha com Achados . . . . .	41
3.11	Pontos Fortes . . . . .	58
3.12	Pontos Fracos . . . . .	60

<b>4</b>	<b>Recomendações</b>	<b>63</b>
4.1	Priorização de Melhorias . . . . .	64
4.2	Métricas . . . . .	71
<b>A</b>	<b>Anexos</b>	<b>77</b>

# **1 Introdução**

Para iniciar a melhoria do processo no projeto de Gamificação para os cursos da Escola do Trabalhador, a equipe SPI (Software Process Improvement) foi formada por estudantes do curso de Engenharia de Software da Universidade de Brasília, campus do Gama. O projeto está sendo desenvolvido por meio de uma parceria inicialmente entre o Ministério do Trabalho e a Universidade de Brasília e, depois, com o Ministério da Economia.

O projeto tem como objetivo disponibilizar um portal para qualificação de trabalhadores, oferecendo cursos online à distância como uma forma de capacitar e auxiliar na luta contra o desemprego. Os cursos são gratuitos e sem requisitos de escolaridade, podendo ser iniciados logo após sua inscrição.

## 2 Inicialização

### 2.1 Começando

O objetivo desta atividade é organizar a equipe que irá criar uma proposta de melhoria que dê fundamento para o início da melhoria do processo de software (MPS) . Esse processo tem como etapas:

1. Criar uma equipe de melhoria;
2. Identificar o “clima para a mudança”;
3. Identificar políticas que deem suporte ou impeçam a criação do programa de melhoria de processo de software;
4. Identificar abordagens para a realização da MPS e selecionar a que melhor se enquadra na organização.

A equipe de melhoria é composta por 5 estudantes de Engenharia de Software de semestres distintos e que estão cursando a disciplina de Melhoria de Processo de Software. Dentro da equipe um dos membros é também um membro da equipe de desenvolvimento do projeto no qual será implantado a MPS.

A necessidade de mudança ocorre pois o processo atual vem gerando muito retrabalho, devido principalmente à falta de comunicação entre equipes diferentes. Esta falta de comunicação também resulta em demandas variadas e imprevisíveis.

A equipe de gamificação oferece total suporte para implementação de um programa de melhoria e todos os alunos membros da equipe são facilmente acessíveis, facilitando a comunicação.

Já o maior empecilho que impede a criação do programa são as outras equipes, como a de Pedagogia, pois estas não estão habituadas a um processo formalmente descrito de desenvolvimento. A forma de comunicação é dificultada devido a atuarem em outro campus, e as demandas desta equipe impactam diretamente no processo da equipe de gamificação.

Algumas abordagens de melhoria de processo de software identificadas foram a abordagem iterativa do *Rational Unified Process* (RUP), *Project Site*, Gerência de

Requisitos, *Unified Modeling Language* (UML), Gerência de Mudanças e Verificação Contínua da Qualidade do Software.

Devido aos problemas identificados na organização, as duas abordagens que mais se enquadram nela seriam a Gerência de Mudanças e a Abordagem Iterativa do RUP, pois ambas possuem maior flexibilidade quanto à mudança de requisitos. A abordagem *Project Site* atualmente já vem sendo utilizada pela equipe de desenvolvimento, porém os outros *stakeholders* não acompanham o desenvolvimento destas atividades.

## **2.2 Identificação das Necessidades de Negócio e Diretrizes de Melhoria**

O objetivo desta etapa é compreender as necessidades que tornam necessário a melhoria do processo de software. É importante que essas estejam alinhadas aos objetivos da organização e objetivos de negócio, e não somente aos desejos de um indivíduo. Para chegar nisso, essa etapa é dividida em 3 passos:

1. Identificar as necessidades de negócio, realizando entrevistas com gerentes e analisando documentos a respeito da visão organizacional;
2. Revisar as necessidades, verificando se elas podem ser satisfeitas seguindo um processo de melhoria de software;
3. Definir como o programa de melhoria de processo de software iria satisfazer as necessidades de negócio.

A principal necessidade de negócio da equipe é aumentar a velocidade na entrega de cursos gamificados, evitando o retrabalho constante da equipe. Isso fará com que o tempo de produção dos membros seja gasto de maneira mais adequada, realizando as atividades necessárias para a produção dos cursos gamificados. Além disso, há também uma necessidade de integrar rapidamente novos membros à equipe quando estes são adicionados, explicando o método de trabalho e as rotinas utilizadas.

Analisando estas necessidades, é possível concluir que um ciclo de produção mais detalhado, utilizando um processo de melhoria de software, poderia contribuir para a resolução parcial ou total dessas demandas existentes.

Um processo otimizado poderá ajudar detalhando melhor a cadeia de desenvolvimento. Assim, quando novos membros entrarem na equipe eles saberão quais e como as etapas de desenvolvimento funcionam. Outra forma seria tratando dentro do processo a comunicação entre as equipes, estabelecendo sempre uma forma rápida de validação para que o trabalho seja executado e continuado de modo mais célere.

## 2.3 Construindo uma Proposta de Melhoria de Processo de Software

O objetivo desta etapa é criar uma proposta que será apresentada para o gerente da equipe, explicando o que o programa de melhoria de processo é, porque ele deve ser feito, qual o custo, o tempo para ser executado e a abordagem utilizada. As atividades dessa tarefa são:

1. Identificar os *stakeholders* gerenciais;
2. Chegar a um consenso em relação aos problemas que serão endereçados pela proposta do processo de melhoria;
3. Estabelecer objetivos para o programa de melhoria garantindo consistência com os objetivos de negócio;
4. Iniciar uma visão do estado de maturidade desejado para o processo da organização;
5. Determinar o escopo;
6. Determinar as responsabilidades dos envolvidos;
7. Determinar um plano de alto nível que inclui as atividades e o planejamento na fase de Estabelecimento;
8. Determinar os benefícios para a organização;
9. Iniciar a criação do plano de comunicação da organização.

Os *stakeholders* gerenciais são o professor Izaías Queiroz, mestrando da Universidade de Brasília que ocupa a função de gerente da frente de Gamificação, o professor Wander Pereira que é o coordenador da equipe, a professora Wilsa Maria Ramos como coordenadora da equipe Pedagógica e a professora Therése Hoffman como coordenadora geral do projeto.



O projeto ainda conta com mais de 30 professores da Universidade de Brasília que ocupam outros cargos, mas que não influenciam tão diretamente no processo da equipe de gamificação.

Os problemas que serão abordados no processo de melhoria serão o retrabalho constante da equipe devido à falha de comunicação e de demandas imprevisíveis solicitadas por outras equipes. Também será tratado o problema do alto tempo para total integração de novos membros à equipe.

Os objetivos do processo de melhoria são aumentar a produtividade e a satisfação da equipe, resolvendo os problemas de retrabalho e de adequação dos novos membros. Com o aumento da produtividade e satisfação, a entrega de cursos gamificados pela equipe deve aumentar tanto em rapidez quanto em qualidade.

Para analisar o estado de maturidade do processo existente atualmente, será utilizado o *Capability Maturity Model Integration* (CMMI). Nesse modelo, processos podem evoluir do nível inicial – no qual se encontram desorganizados e não confiáveis – para o estado desejado, que é o de uma estrutura madura capaz de manter processos executados de maneira consistente.

Entre os entregáveis previstos, a equipe de MPS planeja entregar um arquivo Bizagi com a modelagem do processo definida, assim como um arquivo explicando cada uma de suas etapas. Ela também pretende disponibilizar um plano de medição para que seja possível mensurar os resultados obtidos e comparar se estão alcançando os objetivos propostos.

A equipe de desenvolvimento do projeto de gamificação será responsável por seguir a melhoria do processo sugerida, comunicando quaisquer problemas à equipe de MPS. Já a equipe de MPS terá como responsabilidade desenvolver o programa de MPS de acordo com as necessidades da organização, assim como acompanhar a implementação e os resultados obtidos, realizando as mudanças necessárias para que estes objetivos sejam concluídos.

Durante a fase do Estabelecimento, ocorrerá a priorização das atividades de melhoria e serão criadas estratégias para buscar soluções. O esboço do programa de melhoria de processo de software será concluído de acordo com as necessidades da organização. Nessa fase, as métricas serão desenvolvidas a partir dos objetivos

gerais. Algumas das que estão sendo sendo consideradas são o ESI (índice de satisfação) para analisar a satisfação dos membros da equipe de gamificação, além da quantidade de cursos gamificados entregues por mês, para avaliar a produtividade da equipe.

A equipe de gamificação será beneficiada pois o processo de melhoria terá como objetivo atender as necessidades apontadas pela própria equipe. Ela também será favorecida pelo detalhamento dos passos que deverão ser seguidos, uma vez que irá facilitar a construção um padrão de desenvolvimento no qual novos membros que chegarem na equipe poderão rapidamente se adequar e manter a produtividade e satisfação da equipe sempre alta.

A equipe de desenvolvimento forneceu os contatos para os membros da equipe de MPS para que assim a comunicação possa ser rápida e efetiva por meio de qualquer aplicativo de comunicação, como WhatsApp ou Telegram. Um dos membros da equipe de MPS também faz parte do time de desenvolvimento do projeto. Sendo assim, esse ficará responsável por informar os demais membros de ambas as equipes sobre o andamento do projeto, sendo uma ponte entre os dois times.

## **2.4 Suporte à Educação e Construção**

Esta fase tem como objetivo comunicar à organização que o processo de melhoria poderá acontecer nela, informando aos envolvidos as expectativas e outros aspectos relacionados à comunicação da MPS. As fases dessa etapa são:

1. Criar vários briefings que informem quais são os departamentos envolvidos, porque será iniciado, como irá afetar a empresa e quais são os objetivos (essas reuniões tem que ser direcionadas para cada “público alvo da organização”)

Os departamentos envolvidos no processo de melhoria serão a frente de gamificação do projeto na qual será executado o programa de processo de melhoria de software. A equipe de pedagogia fará parte somente das interações com a equipe de gamificação.

A equipe de gamificação foi avisada sobre a implementação deste novo processo e que ele irá afetar o desenvolvimento deles. Mesmo assim, eles deverão seguir o novo processo descrito, relatando quaisquer problemas e sugerindo mudanças que sejam necessárias.

Deverão também compartilhar informações que sejam relevantes para que a equipe de MPS possa realizar as medições necessárias para o acompanhamento da melhoria do processo.

## **2.5 Obter aprovação da MPS e Recursos Iniciais**

A proposta de melhoria de processo de software será apresentada ao gerente da equipe de gamificação para validação e aprovação.

Sendo aprovada a proposta pelo gerente, haverá a alocação de tempo e de recursos para a implementação do plano. Será feita uma destinação inicial de recursos para começar o trabalho da MPS, Será estabelecido a estratégia financeira – identificando quem será o responsável por prover e por gerenciar esses recursos – sendo orçado os recursos necessários e atualizado o plano de comunicação da equipe.

A equipe de desenvolvimento foi avisada sobre a implementação do processo de melhoria e deram o aval para ser iniciada. Estes teriam uma reunião com o gerente no dia 26/04 para informar sobre o novo processo, porém o gerente já havia dado liberdade para que os desenvolvedores evoluíssem o processo atual que vinha sendo utilizado.

## 2.6 Estabelecer infraestrutura da MPS

Esta etapa do processo visa definir de forma clara deveres e responsabilidades, como forma de garantir uma maior chance de sucesso à MPS. Dessa maneira, é possível construir ferramentas que garantam à organização o poder de incorporar o processo e realizar melhorias incrementais ao longo do tempo. Para isso, será considerado o tamanho, estrutura e cultura da organização onde se aplicará o MPS e serão aplicados os seguintes passos:

1. Estabelecer Grupo de Gestão de Direção;
2. Estabelecer Grupo de Processo de Engenharia de Software;
3. Manter a visibilidade em relação ao progresso do processo;
4. Facilitar e estimular o compartilhamento de informação;
5. Guardar lições aprendidas e melhorias desenvolvidas;
6. Criar rede de suporte às pessoas envolvidas no processo de melhoria.

O objetivo de estabelecer um Grupo de Gestão de Direção (GGD) é designar responsabilidades dentro do projeto em relação à MPS. O GGD é essencialmente o *project manager* do processo, e o mesmo deve ser capaz de oferecer suporte e recursos para o esforço a ser despendido. Caso um grupo similar a este já exista, será possível apenas adaptá-lo para que lhe caibam as responsabilidades necessárias.

Estabelecer um Grupo de Processo de Engenharia de Software (GPES) é uma tarefa que busca atribuir responsabilidades para a facilitação e orientação na implementação do processo. O GPES não necessariamente deverá ser responsável pela implementação, mas tem o papel de guiar o responsável. Assim como no caso do GGD, caso já exista algum grupo que possa se encaixar no papel, o mesmo pode ser adaptado para tal.

Esta tarefa é de suma importância pois, por mais que os principais responsáveis pela organização estejam envolvidos na inicialização da MPS, geralmente os mesmos são deixados de lado com o decorrer do tempo. Separar um tempo para dar

*feedback* sobre os avanços do processo garante visibilidade ao mesmo, o que se tornar importante ao longo da implantação, já que os frutos provenientes da MPS são incrementais.

Quanto mais atividades envolvidas na MPS, mais difícil fica para se compartilhar informação entre o GPES e a organização e os envolvidos nessa. Muitas vezes, há mais de uma solução sendo desenvolvida para um mesmo problema. Estabelecer compartilhamentos formais de informação previne que se perca tempo tentando reinventando a roda.

Para que seja validado se o compartilhamento de informações está sendo feito corretamente, pode-se utilizar de formulários com perguntas sobre as informações sendo compartilhadas e o valor delas.

Por mais que o compartilhamento de informação dentro da MPS seja importante, ele serve apenas para o momento atual, de forma que essa informação é perdida caso não haja a preocupação em documentá-la. Assim, como o passo anterior, este passo previne que a equipe do GPES se encontre fazendo retrabalho de maneira desnecessária.

A MPS deve criar ou se adaptar a um existente processo de documentação que permita que a organização de um modo geral cresça com maturidade. Para tal, utilizar-se de um repositório de informações é algo essencial. Dentre o tipo de informação a ser documentada, pode-se listar coisas como:

- O próprio processo de melhoria;
- Exemplos de artefatos gerados pelo processo;
- Soluções desenvolvidas para resolver problemas conhecidos e como aplicá-las.

É comum que em muitas organizações nunca se tenha aplicado a MPS. Durante a aplicação do processo é necessário que se aprenda a fazer certas atividades, bem como deixar certos hábitos morrerem. Isso requer mudanças comportamentais não apenas pessoais, mas da organização como um todo.

Dito isso, criar uma rede *peer-to-peer* entre o GPES e o GGD com pessoas de outras organizações pode ajudar a encontrar mão de obra que ajude a resolver os problemas certos.

O ideal é que isso não seja feito apenas pelo compartilhamento de informações, mas também pelo desenvolvimento conjunto de atividades, bem como treinamentos colaborativos e projetos conjuntos de MPS em ambas as organizações.

## **2.7 Avaliar o clima para a MPS**

A avaliação do clima para a melhoria de processo de software servirá para identificar dificuldades e pontos influentes sobre a equipe que impactarão a proposta de melhoria e para diminuir esses impactos. Se existirem, haverá o desenvolvimento de um plano efetivo para garantir que a melhoria de processo será continuada e posta em prática. Esta etapa possui uma sub-atividade:

1. Avaliar a cultura organizacional, planos de comunicação existentes, resistência ao programa de melhoria e outras barreiras, de maneira a pensar em melhorias e estratégias para contornar possíveis barreiras.

Para a identificação, será analisado a cultura e o histórico de dificuldades passados pela equipe para implementar estratégias aplicadas em dificuldades similares às que serão enfrentadas. Será analisado as resistências à melhoria de processo, alinhadas as propostas que já estiverem ocorrendo com as novas e atualizado o plano de comunicação da equipe para a implementação das novas estratégias.

A cultura organizacional da equipe de gamificação favorece a implementação de novas metodologias e novos processos. A equipe está sujeita à mudanças mas possui o hábito de modificar parte dos processos para se adequarem às demandas atuais e esse hábito pode ser uma barreira para que sigam o novo processo. Mas, caso este obtenha resultado e atenda os objetivos, a equipe se mostra disposta à colaborar.



## 2.8 Definir metas gerais da MPS

O objetivo desta etapa é definir os objetivos para o processo de melhoria, possivelmente passíveis de medição, que possam permitir que os resultados do programa de melhoria sejam avaliados. Esta etapa possui as seguintes tarefas:

- Obter informações sobre quais objetivos organizações bem sucedidas estão conseguindo realizar;
- Definir objetivos de alto nível a partir da visão, plano de negócios e outros artefatos, sejam estes a longo ou curto prazo.

Os temas mais utilizados nos processos de planejamento de objetivos atualmente no mercado são crescimento, rentabilidade, participação de mercado, produtividade e qualidade. Porém, destes somente a produtividade seria aplicável ao presente trabalho. Outros objetivos que vêm crescendo com o tempo dizem respeito ao grau de satisfação dos funcionários do projeto. Grandes empresas como o Google trabalham bastante com isso, pois identificaram que a produtividade dos funcionários aumenta quando estes estão satisfeitos com o trabalho que realizam.

Os objetivos selecionados são aumentar a produtividade e satisfação da equipe. A equipe em 6 meses de trabalho entregou 3 propostas de gamificação e implementação destas dentro da plataforma, assim como um *plugin* para o moodle e um artigo científico. A última proposta de gamificação foi um *template* genérico para que fosse possível replicar em diversos cursos, e o objetivo da equipe seria que com isso conseguisse aumentar a produção de cursos para 1 curso por semana. Quanto à satisfação da equipe, a forma utilizada seria com o ESI, ou índice de satisfação dos funcionários. Essa métrica é extraída através de 3 perguntas que seriam feitas e depois a fórmula seria aplicada para obter o grau de satisfação.

## **2.9 Definir os Princípios Orientadores do Programa SPI**

O propósito desta etapa é definir quais os princípios que servirão de referência para o processo de melhoria de software descrito nesse documento. Os princípios elencados são:

- Respeitar o contexto no qual o processo está inserido;
- Respeitar os participantes do processo sendo melhorado, assim como a experiência deles;
- Pautar mudanças no processo a partir de dados, a menos que os dados não sejam possíveis de serem obtidos.

## 2.10 Lançamento do modelo

A última etapa do processo de inicialização é a definição de como será lançado o modelo. Para isso, será feito um workshop com os participantes deste processo de melhoria de software.

Neste *workshop*, será apresentado aos participantes o que é o SPI. Também será explicado o modelo IDEAL, que está sendo usado como base para realizar a melhoria do processo de produção de software que está sendo alvo do presente trabalho. Por fim, será aberto espaço para que as pessoas possam tirar possíveis dúvidas que tenham surgido durante a apresentação.

Além disso, será feita uma revisão da proposta deste SPI após o *workshop* para verificar se algo precisa ser melhorado ou incluído na proposta antes de ir para a próxima etapa do IDEAL.

## 3 Diagnóstico

### 3.1 Referencial Teórico do Diagnóstico

Para o presente trabalho, o referencial teórico utilizado será o CMMI. Segundo [Team \(2010\)](#), os modelos CMMI® (Capability Maturity Model® Integration) são uma coletânea de boas práticas com o objetivo de auxiliar organizações a melhorarem seus processos.

Neste trabalho será utilizado o CMMI-DEV, que conforme definição de [Team \(2010\)](#), fornece orientação para aplicar as boas práticas do CMMI em uma organização de desenvolvimento e é focado em atividades para desenvolver produtos e serviços de qualidade que atendem às necessidades do cliente e dos usuários finais.

Essas práticas compreendem todo o ciclo de vida do produto, desde o desenvolvimento à manutenção. Elas contêm vinte e duas áreas de processo, sendo dezesseis áreas de processo de núcleo, uma área de processo compartilhada e cinco áreas de processo de desenvolvimento específico, nas quais estão a verificação e validação.

Os conceitos de **Maturidade** e **Capacidade** são de extrema importância para o CMMI. Segundo [Team \(2010\)](#), Maturidade, nesse modelo, é definida por cinco níveis compostos de práticas específicas e genéricas relacionadas a um conjunto predefinido de áreas de processo que melhoram o desempenho global da organização e que auxiliam na previsão dos resultados de futuros projetos. Já a Capacidade também é composta por seis níveis que caracterizam a habilidade do processo para alcançar os objetivos de negócio, atuais ou futuros e são um meio de melhorar, de forma incremental, os processos correspondentes a uma determinada área de processo; estando relacionada com o atendimento aos atributos de processos associados aos processos de cada nível de maturidade ([TEAM, 2010](#)).

Segundo [Team \(2010\)](#), o CMMI é dividido em cinco estágios ou níveis de maturidade, sendo eles:

- **Inicial:** nesse estágio, os processos são imprevisíveis, pouco controlados e

reativos;

- **Gerenciado:** os processos passam a ser organizados por projeto e as ações são frequentemente reativos;
- **Definido:** os processos são caracterizados para a organização e são proativos;
- **Quantitativamente Gerenciado:** os processos atingem o nível de serem medidos e controlados;
- **Otimização:** o foco passa a ser contínuo e visando a melhoria dos processos.

Considerando a divisão por nível de capacidade ou representação contínua, a divisão ocorre da seguinte maneira (TEAM, 2010):

- **Incompleto:** o processo que não é executado ou é executado parcialmente. Uma ou mais metas específicas da área de processo não são satisfeitas e não existem metas genéricas para este nível, já que não há razão para institucionalizar um processo executado parcialmente;
- **Executado:** é um processo que satisfaz às metas específicas da área de processo, apoiando e viabilizando o trabalho necessário para produzir os produtos de trabalho;
- **Gerenciado:** é um processo Executado que dispõe de infraestrutura adequada para apoiar o processo, é planejado e executado de acordo com uma política, emprega pessoas experientes que possuem recursos adequados para produzir saídas controladas, envolve partes interessadas relevantes, é monitorado, controlado e revisado e sua aderência em relação à descrição de processo é avaliada;
- **Definido:** é um processo Gerenciado adaptado a partir do conjunto de processos padrões da organização de acordo com as diretrizes para adaptação da mesma, e contribui com produtos de trabalho, medidas e outras informações de melhoria de processo para os ativos de processo da organização;

- **Gerenciado Quantitativamente:** um processo Definido controlado por meio de técnicas estatísticas e outras técnicas quantitativas. Objetivos quantitativos para qualidade e para desempenho de processo são estabelecidos e utilizados como critérios na gestão de processo. A qualidade e o desempenho de processo são entendidos em termos estatísticos e gerenciados ao longo da vida do processo;
- **Em otimização:** é um processo Gerenciado Quantitativamente e melhorado com base no entendimento das causas comuns de variação inerentes ao processo. O foco de um processo em otimização é a melhoria contínua do desempenho de processo tanto por meio de melhorias incrementais quanto de inovação (CMU/SEI).

Os processos de **Verificação** e **Validação** se encontram no estágio **Definido**.

Segundo [Team \(2010\)](#), os componentes do modelo são classificados em três categorias: requerido, esperado e informativo. Os **requeridos** são essenciais para atingir a melhoria de processo em determinada área de processo, deve ser visivelmente implementado no processo da organização e são as metas específicas e genéricas. Os **esperados** descrevem as atividades que são importantes para conseguir um componente requerido, são as práticas específicas e genéricas e guiam aqueles componentes que implementam melhorias e realizam avaliações. Os **informativos** são aqueles que auxiliam os usuários do modelo CMMI a entenderem os requeridos e esperados.

O CMMI utiliza a seguinte estrutura:

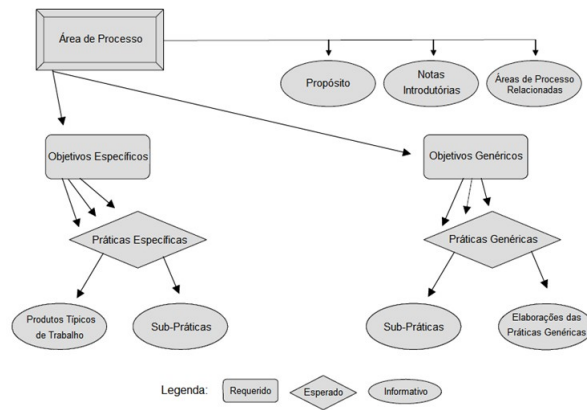


Figura 1: Estrutura do CMMI. Fonte: <http://brunomarota.blogspot.com/2012/04/cmml-para-concursos-parte-2-componentes.html>

## 3.2 *Baseline* dos Processos

O objetivo dessa seção é apresentar a *baseline* dos processos atualmente seguidos pela equipe de gamificação. Com isso, busca-se compreender melhor o processo que será alvo de melhoria.

### 3.2.1 Estrutura da organização e Responsabilidades de seus Componentes

Na Figura 2 está um organograma que mostra a hierarquia à qual está submetida a equipe de gamificação.

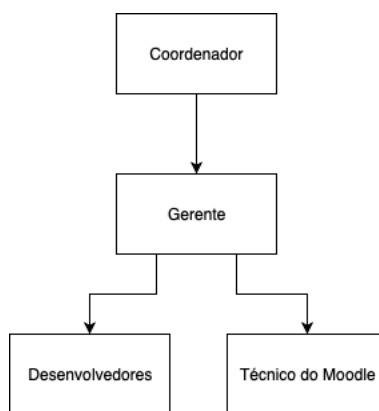


Figura 2: Estrutura hierárquica à qual a equipe de gamificação está submetida.

As responsabilidades dos cargos mostrados no organograma estão explicados a seguir:

- **Coordenador:** Responsável por coordenar a frente, recrutar membros, explicar o projeto e a função da equipe e realizar comunicação efetiva entre outras frentes;
- **Gerente:** Responsável por gerenciar os membros e o projeto, delegando tarefas, determinar as atividades necessárias, levantar requisitos;
- **Desenvolvedores:** Responsáveis por desenvolver e documentar os requisitos solicitados;



- **Técnico do Moodle:** Garantir a funcionalidade correta da infra-estrutura, realizar modificações necessárias para adequação da gamificação e realizar a ponte entre a equipe do Moodle e a equipe de gamificação para subir os cursos na plataforma.

### 3.2.2 Processos Usados pela Equipe de Gamificação

O processo atual é dividido em duas fases, uma de desenvolvimento da proposta de gamificação e outra para implementação desta proposta no curso determinado.

A segunda etapa (ver Figura 4) é baseada no modelo SCRUM, utilizando *sprints* para separar seus entregáveis. Já a primeira etapa (ver Figura 3) utiliza um modelo de Gamificação Adaptativa, ou seja, é definido um tipo de gamificação específica para cada curso em si, com suas peculiaridades dependendo do objetivo do mesmo.

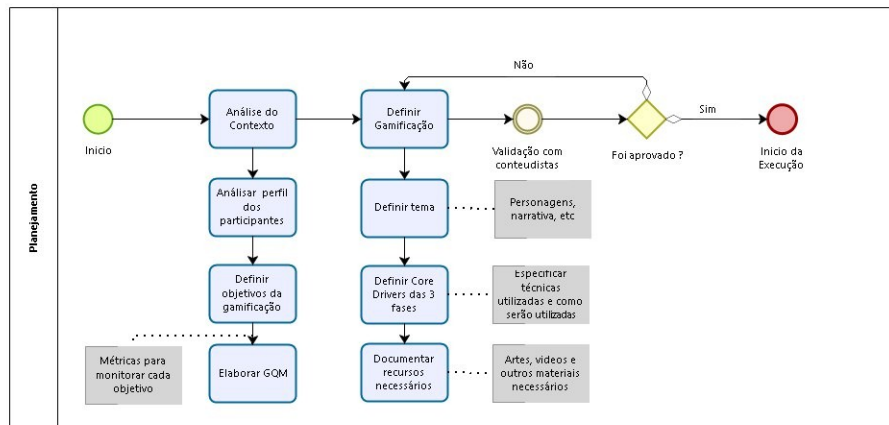


Figura 3: Esquematização do processo de planejamento usado pela equipe de gamificação.

As fases apresentadas na Figura 3 estão apresentadas abaixo, juntamente com o grau de implementação na organização segundo a escala FiLiPiNY:

- **Analisar o contexto:** Ao receber o curso, estuda-se o conteúdo e o assunto abordado para entender no que a gamificação deve auxiliar, de que forma ela

pode melhorar o aprendizado do curso, mantendo o cursista engajado nas atividades. **Grau de Implementação:** FI;

- **Analisar os participantes da gamificação:** Após entender o contexto, a equipe começa a definir os possíveis perfis de cursistas, qual assunto abordado o atrai e qual a melhor forma de abordar esse perfil com as técnicas de gamificação. **Grau de Implementação:** LI (Os conteúdos fornecem esta parte);
- **Definir objetivos da gamificação:** Nesta fase são determinados os resultados esperados da gamificação nos cursos, sendo estes objetivos de como o participante irá interagir e quais comportamentos são esperados que ele desenvolva, mas também objetivos quantitativos a respeito de dados da plataforma. **Grau de Implementação:** PI (É feito o planejamento, mas as métricas não são acompanhadas);
- **Elaborar GQM:** Decidir as métricas e elaborar um plano para coletá-las e assim monitorar cada um dos seus objetivos utilizando medidas quantificáveis. **Grau de Implementação:** NY;
- **Definir o tema da gamificação:** Assim que os dados do contexto e dos possíveis perfis dos cursistas terem sido recolhidos o tema começa a ser trabalhado. O tema da gamificação serve como auxílio para o aprendizado e na escolha das técnicas, o tema ludifica o curso e ameniza a ajuda na quantidade de conteúdo a ser absorvida. Porém o tema não pode se distanciar muito do assunto abordado, um tema muito diferente do assunto que o curso propõe pode causar distração e perda de foco no aprendizado. **Grau de Implementação:** FI;
- **Definir as técnicas:** Com isso, é utilizado o Octalysis para definir de acordo com o contexto, perfil e tema, quais as melhores e mais adequadas técnicas a serem utilizadas. Também é considerado as ferramentas disponíveis na plataforma onde o curso será disponibilizado, visto que isso pode afetar no au-

mento ou diminuição da efetividade das técnicas. **Grau de Implementação:** FI;

- **Definir a jornada do participante:** Ao final de tudo, a equipe organiza as técnicas de acordo com as fases existentes dentro da gamificação: a descoberta, a entrada, o o dia-a-dia e a saída. A descoberta é define como iremos abordar o primeiro contato do cursista com o curso. A entrada determina como iremos ensinar e explicar o funcionamento da gamificação. Durante a fase do dia-a-dia estudamos como a gamificação instiga o aprendizado dentro do curso. A saída assegura que o conteúdo foi bem absorvido. **Grau de Implementação:** PI (A jornada segue o padrão de cursos online, logo a única tratada na gamificação é o dia-a-dia);
- **Documentar recursos necessários:** Após todos estes levantamentos o grupo elabora um documento do que será necessário das outras equipes, mudanças na infra como adição de colunas ou tabelas no banco de dados, artes e vídeos necessários, o que se espera do conteudista e também caso haja necessidade, quais mudanças o desenvolvedor das telas terá de realizar para encaixar a gamificação dentro das telas que já existem. **Grau de Implementação:** LI (Não foi utilizado em todos os cursos).

Após a fase de planejamento da gamificação, a proposta é apresentada para a equipe de Pedagogia. Caso seja recusada, deve-se voltar até o ponto de definir a gamificação e elaborar uma nova proposta com as mudanças solicitadas. Caso seja aprovada, se dará início à fase de execução.

Na fase de execução as etapas realizadas são:

- **Elaboração do *Backlog*:** Listar todas as funcionalidades categorizando-as em épicos, histórias de usuário e tarefas, assim como determinando a prioridade da execução de cada, esta prioridade leva em conta as dependências de uma história para realizar outra. Ao final desta fase é realizado um *Roadmap* para estimar o tempo necessário para finalização da implementação e também para dar um guia para o planejamento das *sprints*. **Grau de**

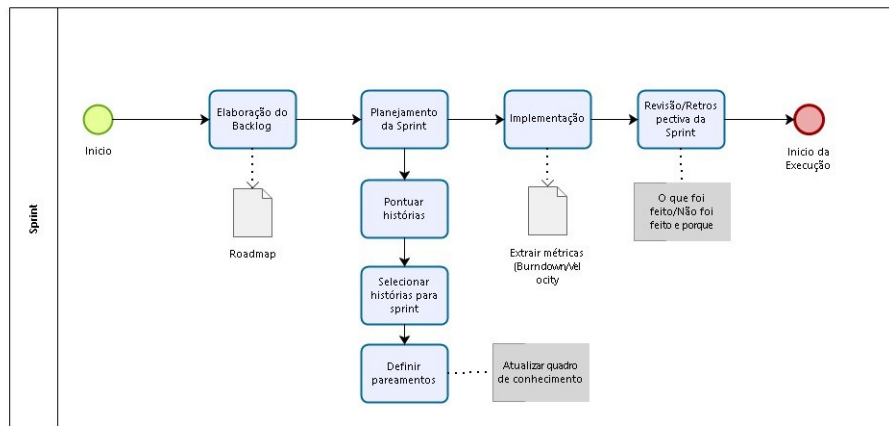


Figura 4: Esquematisação de uma *sprint* usada pela equipe de gamificação.

**Implementação:** PI (Não é sempre utilizado);

- **Pontuar histórias:** As histórias listadas no *Backlog* são pontuadas de acordo com suas dificuldades, é utilizada a sequência de Fibonacci para definir os pontos e são levados em consideração os riscos técnicos sobre a determinada história. **Grau de Implementação:** PI (O acompanhamento para validação dos pontos não é realizado);
- **Selecionar histórias para *sprint*:** No começo o *Roadmap* é utilizado para guiar nesta escolha, porém com o tempo a equipe vai ganhando maturidade através das *sprints* passadas para compreender com maior precisão o que é possível de se realizar no tempo, então as histórias podem ser selecionadas sem necessidade de utilizar do *Roadmap*. **Grau de Implementação:** NI (É utilizado no começo das *sprints*, mas abandonado com o tempo);
- **Definir pareamentos:** As duplas são selecionadas para resolverem as histórias alocadas, é levado em consideração o conhecimento individual de cada um de acordo com o quadro de conhecimento, mas também é realizado uma rotatividade como forma de mesclar e compartilhar os conhecimentos adquiridos. **Grau de Implementação:** PI;

- **Implementação:** Os pareamentos realizam suas histórias e ao longo da *sprint* são coletadas as métricas como *Burndown* e *Velocity*. **Grau de Implementação:** NY (Métricas não são extraídas);
- **Revisão/Retrospectiva da *sprint*:** Nesta fase é descrito o que foi feito e não foi feito na *sprint* assim como o porquê, e também quais medidas deveriam acontecer para evitar os problemas na próxima, assim como as medidas que não funcionaram. **Grau de Implementação:** NY (Não é realizado).

Após a fase de revisão e retrospectiva, o processo volta para o planejamento da próxima *sprint* caso ainda existam *issues* no *Backlog*. Caso não existam mais *issues* a implementação é finalizada e o curso sobe para o ambiente de testes para que as demais equipes testem e validam o curso antes de ir para a plataforma.

### 3.3 Planejamento de Projeto

O objetivo desta fase é estabelecer planos para realizar a Engenharia de Software e a Gerência do Projeto de Software. Essa fase envolve desenvolver estimativas para o trabalho a ser realizado, estabelecendo compromissos necessários e definindo o plano para realizar este projeto.

Ela começa com uma declaração do trabalho a ser realizado e outras constantes e metas definidas pelo projeto de software (que foi definido na gerência de requisitos). O projeto de planejamento deve envolver fases para estimar o tamanho do software e os recursos necessários, produzir um calendário, identificando os riscos e negociando os compromissos de cada envolvido.

Esta fase possui as seguintes atividades:

1. O grupo de desenvolvedores toma parte na definição do backlog
2. O planejamento do projeto de software é feito no início do projeto
3. O grupo de desenvolvedores se reúne com outras partes interessadas para o planejamento do projeto
4. Compromissos de projeto de software feitos a indivíduos e grupos externos à organização são revisados com a alta administração de acordo com um procedimento documentado
5. Um ciclo de vida de um processo de software com etapas pré definidas e tamanho gerenciável é identificado / definido
6. O processo de desenvolvimento de software é feito à partir de um processo documentável
7. O processo é documentado
8. O *backlog* do produto de software é definido
9. É realizada uma estimativa de custo e esforço à partir de um processo documentável

10. É realizada uma estimativa de recurso computacional à partir de um processo documentável
11. Um cronograma é definido à partir de um processo documentável
12. Os riscos de software associados ao custo, recurso, cronograma e aspectos técnicos do projeto são identificados, avaliados e documentados
13. Os planos para as instalações de engenharia de software e as ferramentas de suporte do projeto são preparados
14. Dados de planejamento de software são registrados

### **3.4 Acompanhamento do Projeto**

O propósito desta fase é providenciar uma visibilidade adequada do progresso atual para que assim a gerência possa tomar ações efetivas quando o projeto de software desviar significativamente do projeto planejado. Esta fase envolve rastrear e revisar as conquistas e resultados de acordo com o que foi planejado, e ajustar estes planos baseado nos resultados obtidos. O Planejamento de Projeto de Software é fortemente utilizado nesta fase, e é usado como base para rastrear as atividades realizadas, comunicar o status e revisar os planos feitos. O progresso é determinado comparando o tamanho, custo, esforço e prazos determinados no planejamento com os atuais realizados ao ser completo. Quando o planejamento não está sendo efetivo as ações corretivas são tomadas para se adequar a realidade da organização.

Esta fase possui as seguintes atividades:

1. Um plano de desenvolvimento de software é utilizado para rastrear as atividades e para comunicar o status atual
2. O plano de desenvolvimento de software é revisado de acordo com o procedimento documentado
3. Levantamento de requisitos e mudanças de requisitos com grupos externos são revisados com o gerente de acordo com o procedimento documentado
4. Alterações aprovadas nos requisitos que afetam o projeto de software são comunicados a todos os grupos relacionados no desenvolvimento, tais como grupo de garantia da qualidade, gerência de configuração, entre outros
5. O tamanho dos produtos de software são rastreados e ações corretivas são tomadas de acordo com o planejamento
6. Os esforços e custos do projeto são rastreados e ações corretivas são tomadas caso necessário
7. Os recursos computacionais críticos são rastreados e as ações corretivas são tomadas caso necessário



8. O calendário de atividades é rastreado e as ações corretivas são tomadas caso necessário
9. As atividades técnicas do projeto são rastreadas e as ações corretivas são tomadas caso necessário
10. Todos os riscos associados ao projeto são rastreados
11. Os dados reais de medição e os dados planejados do projeto de software são registrados
12. O grupo de engenharia de software conduz revisões internas para rastrear o progresso, performance, planejamento e problemas a respeito do plano de desenvolvimento
13. Revisões formais sobre as conquistas e resultados do projeto de software são conduzidas em eventos selecionados do projeto

### 3.5 Gerência de Requisitos

O propósito desta fase é de estabelecer um entendimento comum entre o cliente e os requisitos deste que serão utilizados para o projeto de software. Esta fase envolve estabelecer e manter um acordo com o cliente a respeito dos requisitos para o projeto de software, este acordo é conhecido como “Requisitos de sistema alocados para o software”. Este acordo cobre tanto a parte técnica quanto a parte não técnica, como por exemplo datas de entrega, e ele é formado através de planejamento, performance e rastreamento das atividades ao longo do ciclo de vida. E para atingir um controle ideal, a equipe deve revisar sempre os requisitos do sistema como uma forma de identificar e resolver os problemas antes que estes sejam alocados ao projeto, e uma vez que algum requisito alocado ao projeto seja modificado todas suas atividades devem ser ajustadas para manter consistência com os requisitos atualizados.

Para atingir um plano de gerência de requisitos ideal, deve-se atender as seguintes atividades:

1. O grupo de desenvolvedores revisa os requisitos alocados antes de incorporá-los ao projeto de software
2. O grupo de desenvolvedores utiliza os requisitos alocados como uma base para os planejamentos de software, produtos entregues e atividades
3. Mudanças nos requisitos alocados são revisados e incorporados no projeto de software

### **3.6 Gerenciamento de Subcontrato de Software**

O objetivo desta fase é selecionar membros qualificados para o desenvolvimento de software e gerencia-los corretamente. Esta fase envolve selecionar os membros da equipe, rastrear e revisar sua performance e resultados obtidos. Ela também envolve cobrir individualmente cada membro da equipe, assim como o time como um todo e todos os recursos necessários para o desenvolvimento do projeto, tais como hardware e outros componentes necessários. Quando contratando, um documento de acordo cobrindo as partes técnicas e não técnicas tais como prazos de entrega é estabelecido e usado como base para gerenciar o novo membro, suas habilidades devem ser levadas em consideração e um período de adaptação deve ser considerado.

Esta fase possui as seguintes atividades:

1. O trabalho a ser contratado é definido e planejado de acordo com um procedimento documentado;
2. O funcionário é selecionado com base em uma avaliação da capacidade de executar o trabalho de acordo com as necessidades da organização;
3. O contrato entre a empresa e o funcionário é utilizado como base para o gerenciamento do funcionário;
4. O plano de desenvolvimento de software de um funcionário é revisado pelo gerente;
5. As atividades de desenvolvimento de software do funcionário são rastreadas e o gerente tem acesso ao seu status;
6. Alterações no contrato, termos de compromisso e responsabilidades são realizadas de acordo com o procedimento acordado;
7. O gerente conduz revisões periódicas do status do funcionário com a a coordenação do projeto;

8. A troca de conhecimento através de pareamentos e dojos é realizada periodicamente entre os funcionários;
9. São realizadas reuniões para abordar os resultados e conquistas obtidas pelo funcionário de acordo com as etapas selecionadas anteriormente;
10. O grupo de Qualidade de software monitora as atividades de garantia de qualidade do funcionário de acordo com as estabelecidas com a organização;
11. O grupo de GCES monitora o funcionário a respeito dos padrões determinados pela equipe de gerencia de configuração;
12. O gerente realiza testes de aceitação nas entregas dos produtos de software pelo funcionário, para garantir que estas estão de acordo com as estabelecidas;
13. A performance do funcionário é avaliada periodicamente e revisada com o gerente da equipe.

### **3.7 Gerenciamento de Riscos**

A parte de gerenciamento de riscos foi abordada na subseção [3.3](#) - Planejamento de Projeto.

### **3.8 Garantia de Qualidade de Software**

Esta fase envolve a realização de revisões e auditorias no produto de software e em suas atividades para verificar que elas atendem os procedimentos e padrões para fornecer um projeto de software com qualidade e que forneçam métricas para que os gerentes possam trabalhar com os resultados dessas revisões e auditorias. O grupo de qualidade trabalha com o projeto de software desde as primeiras fases para estabelecer os planos, padrões e procedimentos que vão agregar valor ao produto e satisfazer as obrigações do projeto e políticas da organização. Esta fase deve auxiliar o desenvolvimento do produto como uma forma de garantir que as necessidades do cliente serão atendidas e que as auditorias e revisões vão ser utilizadas ao longo do ciclo de vida para que o gerente possa tomar medidas corretivas necessárias nas outras fases-chaves do processo.

Para que esta fase seja concluída, é necessário atender as seguintes atividades:

1. Um plano de qualidade é preparado para o projeto de software de acordo com um procedimento documentado
2. As atividades do grupo de Melhoria são executadas de acordo com o plano
3. O grupo de melhoria participa na preparação e revisão do planejamento de processo de software e em seus procedimentos
4. O grupo de melhoria revisa as atividades de software para verificar sua conformidade
5. O grupo de melhoria audita os produtos de software desenvolvidos para verificar a sua conformidade
6. O grupo de melhoria periodicamente reporta os resultados das atividades para o grupo de engenharia de software
7. Os desvios identificados nas atividades e produtos de software são documentados e tratados de acordo com o procedimento previsto

8. O grupo de melhoria conduz revisões periódicas das suas atividades e descobertas com a equipe de qualidade do cliente, conforme apropriado

### 3.9 Gerenciamento de Configuração de Software

O propósito desta fase é de estabilizar e manter a integridade dos produtos do projeto de software por todo seu ciclo de vida. Esta fase envolve identificar a configuração de software e sistematicamente controlar as mudanças na configuração e mantendo a integridade e rastreabilidade da configuração através do ciclo de vida. Este trabalho deve levar em consideração que produtos devem ser entregues ao cliente, assim como as ferramentas que são necessários para construir estes produtos de software.

Esta fase possui as seguintes atividades:

1. O plano de gerencia de configuração e evolução de software(GCES) é preparado por cada projeto de software de acordo com o procedimento documentado
2. Um plano de GCES documentado é usado como base para realizar as atividades de GCES
3. Um sistema de biblioteca de GCES é estabelecido como um repositório para manter o código
4. O produto de software é a serem colocados no gerenciamento de configuração são identificados
5. As mudanças, relatórios de issues para todos os commits do projeto são iniciados, registrados, revisados e aprovados de acordo com os padrões definidos
6. Mudanças nas baselines são controladas de acordo com o procedimento documentado
7. Produtos da baseline de software são controlados através de releases de acordo com a documentação determinada
8. O status de configuração do produto é registrado de acordo com os procedimentos de documentação



9. Relatórios padrão para reportar as atividades de GCES e os conteúdos do projeto de software são desenvolvidos de forma a se tornar disponível para os grupos e indivíduos afetados.
10. São realizadas auditorias no projeto de software de acordo com os procedimentos documentados

### 3.10 Planilha com Achados

As planilhas de achados foram geradas com base no seguinte modelo

Escala FiLiPiNiNy	
Legenda	Significado
FI - <i>Fully Implemented</i>	Há artefatos o suficiente para comprovar que a atividade foi realizada sem nenhum ponto fraco substancial
LI - <i>Largely Implemented</i>	Há artefatos o suficiente para comprovar que a atividade foi realizada com um ou mais pontos fracos substanciais
PI - <i>Partially Implemented</i>	Não há artefatos ou há artefatos inadequados para comprovar que a atividade foi realizada, alguns artefatos indicam que parte da prática foi realizada, um ou mais pontos fracos substanciais estão presentes. Ou os artefatos são conflitantes entre si, podendo um indicar que prática foi feita e outro indicar que a prática não foi feita
NI - <i>Not Implemented</i>	Não há artefatos o suficiente para comprovar que a atividade foi realizada
NY - <i>Not Yet</i>	O processo ainda não chegou no ponto necessário para realizar a atividade

Planejamento de Projeto				
Atividade	Grau de Definição	Grau de Implementação	Artefato	Obs
O grupo de desenvolvedores toma parte na definição do backlog	LI	LI	Anexo 7	Os desenvolvedores são os principais responsáveis

O planejamento do projeto de software é feito no início do projeto	PI	PI	Documento não fornecido	A equipe de gamificação entrou no meio do projeto, mas havia um planejamento prévio
O grupo de desenvolvedores se reúne com outras partes interessadas para o planejamento do projeto	FI	PI	Documento de presença de reuniões não fornecido	Reuniões semanais ocorrem com todas as equipes, mas nem todos assuntos são tratados
Compromissos de projeto de software feitos a indivíduos e grupos externos a organização são revisados com a alta administração de acordo com um procedimento documentado	NY	NY	Não possuem	Não é realizado
Um ciclo de vida de um processo de software com etapas pré definidas e tamanho gerenciável é identificado / definido	FI	PI	Anexo 9	Nem sempre é respeitado

O processo de desenvolvimento de software é feito a partir de um processo documentável	PI	PI	O modelo Octalysis é utilizado como base	-
O processo é documentado	FI	PI	Anexo 15	Não é completamente implementado
O backlog do produto de software é definido	PI	PI	Anexo 7	Não realizaram em todos os cursos
È realizada uma estimativa de custo e esforço a partir de um processo documentável	NY	NY	Não possuem	Não é realizado
È realizada uma estimativa de recurso computacional a partir de um processo documentável	NY	NY	Não possuem	Não é realizado
Um cronograma é definido a partir de um processo documentável	PI	PI	Anexo 14	Não realizaram em todos os cursos

Os riscos de software associados ao custo, recurso, cronograma e aspectos técnicos do projeto são identificados, avaliados e documentados	NY	NY	Não possuem	Não é realizado
Os planos para as instalações de engenharia de software e as ferramentas de suporte do projeto são preparados	NY	NY	Não possuem	Não é realizado
Dados de planejamento de software são registrados	NY	NY	Não possuem	Não é realizado

Acompanhamento do Projeto				
Atividade	Grau de Definição	Grau de Implementação	Artefato	Obs
Um plano de desenvolvimento de software é utilizado para rastrear as atividades e para comunicar o status atual	NY	NY	Não é realizado	Não possuem

O plano de desenvolvimento de software é revisado de acordo com o procedimento documentado	NY	NY	Não possuem	Não é realizado
Levantamento de requisitos e mudanças de requisitos com grupos externos são revisados com o gerente de acordo com o procedimento documentado	NI	PI	Anexo 8	Não possui rastro que indique as mudanças de requisitos
Alterações aprovadas nos requisitos que afetam o projeto de software são comunicados a todos os grupos relacionados no desenvolvimento, tais como grupo de garantia da qualidade, gerencia de configuração, entre outros	PI	LI	Documento não fornecido	Mudanças drásticas são avisadas em reuniões gerais nas sextas

O tamanho dos produtos de software são rastreados e ações corretivas são tomadas de acordo com o planejamento	NY	NY	Não possuem	Não monitoram o tamanho do software
Os esforços e custos do projeto são rastreados e ações corretivas são tomadas caso necessário	NY	NI	Não possuem	Não monitoram o esforço, o custo do projeto já estava definido anteriormente
Os recursos computacionais críticos são rastreados e as ações corretivas são tomadas caso necessário	NI	NI	Não possuem	Não é rastreado, porém alguns requisitos mínimos de sistema são apresentados na plataforma
O calendário de atividades é rastreado e as ações corretivas são tomadas caso necessárioAs atividades técnicas do projeto são rastreadas e as ações corretivas são tomadas caso necessário	NI	NI	Anexo 14	Não possuem cronograma detalhado de atividades com datas

As atividades técnicas do projeto são rastreadas e as ações corretivas são tomadas caso necessário	LI	LI	Gitlab	Repositório privado, não é possível fornecer o link
Todos os riscos associados ao projeto são rastreados	NY	NY	Não possuem	Não fazem documento de riscos
Os dados reais de medição e os dados planejados do projeto de software são registrados	PI	PI	Link( <a href="http://indicadores.es.gov.br/">http://indicadores.es.gov.br/</a> )	Não possuem dados desejados para comparar com os obtidos
O grupo de engenharia de software conduz revisões internas para rastrear o progresso, performance, planejamento e problemas a respeito do plano de desenvolvimento	NI	NI	Não possuem	Não utilizam métricas para medir esforço da equipe



Revisões formais sobre as conquistas e resultados do projeto de software são conduzidas em eventos selecionados do projeto	NY	NY	Não é realizado	Não possuem
--	----	----	-----------------	-------------

Gerência de Requisitos				
Atividade	Grau de Definição	Grau de Implementação	Artefato	Obs
O grupo de desenvolvedores revisa os requisitos alocados antes de incorporá-los ao projeto de software	PI	MI	Anexos 7 e 10	Os documentos não possuem rastreabilidade quanto a mudança de requisitos
O grupo de desenvolvedores utiliza os requisitos alocados como uma base para os planejamentos de software, produtos entregues e atividades	PI	NI	Anexo 14	Só utilizaram Roadmap como base para os planejamentos, sem rastrear as mudanças

Mudanças nos requisitos alocados são revisados e incorporados no projeto de software	NI	NI	Não possuem	-
--	----	----	-------------	---

Gerenciamento de Subcontrato de Software				
Atividade	Grau de Definição	Grau de Implementação	Artefato	Obs
O trabalho a ser contratado é definido e planejado de acordo com um procedimento documentado	LI	LI	Não fornecido	O projeto possui um documento TAP com as especificações
O funcionário é selecionado com base em uma avaliação da capacidade de executar o trabalho de acordo com as necessidades da organização	NY	PI	Não possuem	Não documentado, é realizado análise de currículo e entrevista presencial
O contrato entre a empresa e o funcionário é utilizado como base para o gerenciamento do funcionário	FI	FI	Anexo 15	É necessário realização de relatórios semestrais que sejam condizentes com o contrato

O plano de desenvolvimento de software de um funcionário é revisado pelo gerente	NY	NY	Não possuem	Não é realizado
As atividades de desenvolvimento de software do funcionário são rastreadas e o gerente tem acesso ao seu status	PI	PI	Anexo <a href="#">12</a>	As atividades podem ser monitoradas através dos commits
Alterações no contrato, termos de compromisso e responsabilidades são realizadas de acordo com o procedimento acordado	NY	NY	Não Possuem	Situação não ocorreu
O gerente conduz revisões periódicas do status do funcionário com a a coordenação do projeto	NY	NY	Não possuem	Não é realizado

A troca de conhecimento através de pareamentos e dojos é realizada periodicamente entre os funcionários	PI	PI	Anexo 14	Pareamentos são definidos no Roadmap para troca de conhecimentos
São realizados reuniões para abordar os resultados e conquistas obtidas pelo funcionário de acordo com as etapas selecionadas anteriormente	PI	PI	11	Reuniões ocorrem sem planejamento
O grupo de Qualidade de software monitora as atividades de garantia de qualidade do funcionário de acordo com as estabelecidas com a organização	NY	NY	Não possuem	Não é realizado

O grupo de GCES monitora o funcionário a respeito dos padrões determinados pela equipe de gerencia de configuração	NY	NY	Não possuem	Não é realizado
O gerente realiza testes de aceitação nas entregas dos produtos de software pelo funcionário, para garantir que estas estão de acordo com as estabelecidas	PI	PI	Não possuem documento	É feita uma validação informal com o gerente antes de apresentar para outras equipes
A performance do funcionário é avaliada periodicamente e revisada com o gerente da equipe	NY	NY	Não possuem	Não é realizado

Garantia de Qualidade de Software				
Atividade	Grau de Definição	Grau de Implementação	Artefato	Obs

Um plano de qualidade é preparado para o projeto de software de acordo com um procedimento documentado	NY	NY	Não possuem	Não é realizado plano de qualidade
As atividades do grupo de Melhoria são executadas de acordo com o plano	NY	NY	Não possuem	Não é realizado plano de qualidade
O grupo de melhoria participa na preparação e revisão do planejamento de processo de software e em seus procedimentos	NY	NY	Não possuem	Não possuem grupo de melhoria
O grupo de melhoria revisa as atividades de software para verificar sua conformidade	NY	NY	Não possuem	Não possuem grupo de melhoria
O grupo de melhoria audita os produtos de software desenvolvidos para verificar a sua conformidade	NY	NY	Não possuem	Não possuem grupo de melhoria

O grupo de melhoria periodicamente reporta os resultados das atividades para o grupo de engenharia de software	NY	NY	Não possuem	Não possuem grupo de melhoria
Os desvios identificados nas atividades e produtos de software são documentados e tratados de acordo com o procedimento previsto	NY	NY	Não possuem	Não é realizado
O grupo de melhoria conduz revisões periódicas das suas atividades e descobertas com a equipe de qualidade do cliente, conforme apropriado	NY	NY	Não possuem	Não possuem grupo de melhoria

Gerenciamento de Configuração de Software				
Atividade	Grau de Definição	Grau de Implementação	Artefato	Obs

O plano de gerência de configuração e evolução de software(GCES) é preparado por cada projeto de software de acordo com o procedimento documentado	NY	PI	Não documentado	Realizam padrões de branches e commits, porém não foi documentado
Um plano de GCES documentado é usado como base para realizar as atividades de GCES	NY	PI	Não documentado	Seguem os padrões que determinaram internamente, mas sem documentação
Um sistema de biblioteca de GCES é estabelecido como um repositório para manter o código	FI	FI	Gitlab	Utilizam um repositório privado no git para manter histórico de versões
O produto de software é a serem colocados no gerenciamento de configuração são identificados	NI	PI	Não possuem documentação	Os projetos são separados dentro da organização



As mudanças, relatórios de issues para todos os commits do projeto são iniciados, registrados, revisados e aprovados de acordo com os padrões definidos	NI	NI	Issues adicionadas	Nem todas as issues realizadas foram adicionadas, sem seguir um padrão de documento
Mudanças nas baselines são controladas de acordo com o procedimento documentado	NY	NY	Não possuem	Não é monitorado
Produtos da base-line de software são controlados através de releases de acordo com a documentação determinada	NY	NY	Não possuem	Não é realizado
O status de configuração do produto é registrado de acordo com os procedimentos de documentação	NY	NY	Não possuem	Não utilizam nenhuma ferramenta de integração contínua

Relatórios padrão para reportar as atividades de GCES e os conteúdos do projeto de software são desenvolvidos de forma a se tornar disponível para os grupos e indivíduos afetados.	NY	NY	Não possuem	Não é realizado
São realizadas auditorias no projeto de software de acordo com os procedimentos documentados	NY	NY	Não possuem	Não é realizado

### 3.11 Pontos Fortes

Após uma análise das atividades definidas e implementadas do processo da organização, é possível evocar pontos fortes que diz respeito ao processo atual, para analisar os pontos fortes, foram utilizadas entrevistas semi-estruturadas com participantes da equipe de gerencia e desenvolvimento para levantamento de alguns pontos, e também foram analisados os artefatos e evidências encontradas na planilha de achados, os resultados encontrados foram.

- Há algumas atividades do processo que embora não estão definidas, estão sendo implementadas, revelando uma proatividade dos integrantes em se adaptar às demandas que surgem na organização.
- Diante de mudanças nos requisitos a equipe é efetiva na comunicação com os demais integrantes, o que mostra um processo de desenvolvimento não burocrático ou engessado em novas demandas mas flexível.
- Há um ambiente favorável para o desenvolvimento profissional mútuo dos integrantes da organização, há uma boa comunicação entre os membros da frente de gamificação e uma diversidade de conhecimento da equipe que é muitas vezes explorada através de pareamentos.

Tabela Pontos Fortes	
Documentos	Pontos Fortes
Planejamento de Projeto	Participação ativa dos desenvolvedores em todas etapas do projeto
Planejamento de Projeto	Processo bem documentado
Planejamento de Projeto	Reuniões semanais entre desenvolvedores e stakeholders
Acompanhamento de Projeto	Mudanças significativas são informadas em reuniões periódicas
Gerencia de Requisitos	Backlog bem definido

Gerencia de Requisitos	Utilizam Roadmap para ter uma estimativa de prazo
Gerenciamento de Subcontrato	Documento de abertura do projeto com todas as fases documentadas
Gerenciamento de Subcontrato	Realizam relatórios semestrais descrevendo as atividades realizadas neste período
Gerenciamento de Configuração de Software	Utilizam plataformas de versionamento para manter o código
Gerenciamento de Configuração de Software	Possuem servidor com ambiente de desenvolvimento configurado para uso e realização de testes

### 3.12 Pontos Fracos

Após uma análise das atividades definidas e implementadas do processo da organização, também é possível evocar pontos fracos que diz respeito ao processo atual, para analisar os pontos fracos, foram utilizadas entrevistas semi-estruturadas com participantes da equipe de gerencia e desenvolvimento para levantamento de alguns pontos, e também foram analisados os artefatos e evidências encontradas na planilha de achados, os resultados encontrados foram:

- Gerente sem experiência em desenvolvimento de software e na utilização de processos, parte da implementação sem comunicação com outras frentes.
- Falta de espaços para treinamento e desenvolvimento dos membros da equipe, os quais não são especialistas e não tem total domínio das ferramentas.
- Muitas atividades evocadas estão sequer parcialmente implementadas, em especial o gerenciamento de riscos e alguma revisão e auditoria para garantir a qualidade das entregas do software.
- Não há uma confiabilidade se os integrantes solucionarão as necessidades da organização, pois nem antes da contratação há uma avaliação de capacidades técnicas específicas para resolver as demandas nem há uma análise de performance do integrante após ser contratado.

Tabela Pontos Fracos	
Documentos	Pontos Fracos
Planejamento de Projeto	Não revisam compromissos com a administração
Planejamento de Projeto	Não realizam estimativa de curso
Planejamento de Projeto	Não realizam estimativa de recurso computacional
Planejamento de Projeto	Não produzem documento de riscos

Planejamento de Projeto	Não preparam planos para instalação
Planejamento de Projeto	Não registram dados de planejamento
Acompanhamento de Projeto	Não produzem plano de desenvolvimento de software
Acompanhamento de Projeto	Não monitoram o tamanho dos produtos de software
Acompanhamento de Projeto	Não rastreiam os esforços e custos do projeto
Acompanhamento de Projeto	Não rastreiam os riscos do projeto
Acompanhamento de Projeto	Não são realizadas revisões formais sobre resultados
Gerencia de Requisitos	Não rastreiam as mudanças de requisitos
Gerenciamento de Subcontrato	O plano de desenvolvimento de software de um funcionário não é rastreado
Gerenciamento de Subcontrato	O gerente não realiza revisões periódicas quanto ao status do funcionário
Gerenciamento de Subcontrato	Não possui grupo de qualidade de software
Gerenciamento de Subcontrato	Não possui grupo de GCES
Gerenciamento de Subcontrato	A performance do funcionário não é revisada pelo gerente da equipe
Garantia da Qualidade	Não realizam quaisquer tipo de medições ou documentações quanto a qualidade

Gerenciamento de Configuração de Software	Não produzem documento de configuração de Software
Processo Atual	Setas para baixo no processo sem a existência de estruturas de decisão

## 4 Recomendações

A partir do levantamento da *baseline*, foram pensadas recomendações no sentido de melhorar tanto o processo interno da equipe quanto a relação com as outras frentes. Também foi analisada a questão do conhecimento necessário para usar as tecnologias e *frameworks* utilizados no projeto.

Pelas informações obtidas, fica claro que um dos maiores empecilhos para a equipe são as outras frentes, pois essas afetam diretamente o trabalho e desempenho daquela, cujo processo está sendo analisado para futura melhoria.

As recomendações para evitar mudanças de última hora no processo e melhorar a comunicação com outras frentes são:

- Deixar claro para as outras frentes que elas atrapalham ao enviar demandas de última hora e com urgência;
- Estabelecer que mudanças solicitadas não serão mais feitas na *sprint* atual a menos que seja algo realmente urgente. Do contrário, serão colocadas no *backlog* do produto e devidamente priorizadas;
  - Se a mudança encaminhada for colocada como urgente pela outra equipe, ela deverá vir com a justificativa para a urgência para que possa ser analisada pela equipe de gamificação;
- Interar o coordenador sobre as atividades e processo de desenvolvimento da equipe.

Quanto à questão das dificuldades da equipe com relação às tecnologias e *framework* utilizados, as informações obtidas mostram que determinadas práticas ágeis podem ajudar a diminuir ou até mesmo a acabar com os problemas.

As recomendações para lidar com falta de experiência dos membros da equipe em algumas tecnologias específicas do *moodle* e dos cursos são:

- Conscientizar os responsáveis pela equipe que é necessário que haja espaço e tempo para que os membros possam trocar conhecimentos e aprenderem sobre as tecnologias e *frameworks* necessários;



- Realizar dojos para compartilhamento de conhecimento entre os membros da equipe;
- Realizar pareamentos entre membros que detêm maior conhecimento com os que têm menos;
- Buscar usar *frameworks* e tecnologias com boa documentação.

A equipe deve analisar os pontos fracos apresentados neste documento, para assim gerarem a documentação necessária para manter uma maior rastreabilidade de todas as etapas. Os documentos recomendados a serem feitos são:

- Documento de custos;
- Documento de Gerencia de Riscos;
- Documento de Garantia de Qualidade;
- Manter rastro dos Requisitos;
- Manter rastro de métricas e indicadores do projeto;
- Documento de Gerencia de Configuração de Software.

Existem também erros de modelagem no processo atual, devido à não utilização da notação adequada como, por exemplo, o uso de setas para baixo que segundo a equipe representam sub-atividades daquela etapa. Sendo assim a recomendação seria:

- Utilização da notação de sub-processos na modelagem;
- Separar corretamente o que são sub-processos, atividades e documentos gerados.

#### **4.1 Priorização de Melhorias**

Sendo assim, a lista de recomendações para a organização ficou dividida em 3 principais categorias, sendo elas:

- Melhorias organizacionais: englobam os aspectos de comunicação e organização interna e externa de acordo com os problemas relatados;
- Elaboração de documentos: elaborar e incrementar documentação básica dos projetos da organização para garantir maior maturidade e rastreabilidade de informações;
- Correções e otimizações: revisar e corrigir erros de notações no processo

Para realizar a priorização das melhorias foi utilizada a técnica MoSCoW, desenvolvida por (CLEGG; BARKER, 1994). Ela consiste em priorizar os requisitos categorizando-os em 4 classificações:

- **Must Have:** Requisitos vitais que devem ser desenvolvidos e incluídos imediatamente;
- **Should Have:** É importante mas não tão vital quanto o *Must Have*, podendo ser retirado se for necessário. Contudo deve ser desenvolvido assim que possível, e tem maior prioridade de implementação do que o *Could have*;
- **Could Have:** É algo que seria bom possuir como um algo a mais, mas não é essencial então será incluído caso o tempo e recursos permitam;
- **Won't Have:** Não é necessário desenvolver ou incluir no projeto. Normalmente acordado pelos *stakeholders* de não entregar em uma determinada entrega, mas a considerar para o futuro.

Sendo assim, para a realização do quadro de priorizações foi gerada a primeira tabela com a priorização de acordo com 3 principais categorias, e depois outra tabela para cada uma destas categorias com as atividades presentes nela.

Priorização MoSCoW por Categoria		
Categoria	Prioridade	Justificativa
Melhorias Organizacionais	Must Have	Grande parte dos problemas apontados pela organização se encontram aqui. Logo tem que ser desenvolvida o quanto antes. Devido a equipe seguir uma metodologia ágil o foco em pessoas é ainda maior que o foco em documentos, e por conta disto os aspectos organizacionais, rituais, etc devem possuir uma alta prioridade.
Elaboração de Documentos	Must Have	Setor no qual mais foram identificados problemas, é a categoria que mais impacta no nível de maturidade atualmente. Então, para que seja atingido o nível 2 desejado pela organização, estes pontos devem ser atacados o quanto antes, e por conta disto a prioridade desta categoria é alta
Correções e Otimizações	Won't Have	As correções e utilizações das notações corretas na modelagem devem ser feitas, contudo, o processo atual será modificado para que atenda as novas necessidades da organização, sendo assim esta recomendação deve ser feita na modelagem do novo processo, após as atividades anteriores serem realizadas

Dentro de cada uma destas categorias foi realizada uma priorização de quais atividades devem ser feitas com maior prioridade seguindo a mesma escala MoSCoW,

resultando assim nas tabelas a seguir:

Priorização MoSCoW das Melhorias Organizacionais		
Atividade	Prioridade	Justificativa
Integrar outras frentes no planejamento da sprint	Must Have	Grande parte dos problemas encontrados ocorrem devido a não participação de outras frentes no planejamento da sprint, tendo estas muitas vezes outras prioridades nas quais apresentam durante uma sprint já iniciada, sendo assim um acordo comum entre as frentes deveria ser realizado no início desta sprint.
Realização de Dojos e Práticas do XP	Should Have	Para balancear e compartilhar o conhecimento entre todos os membros da equipe, estas práticas são fundamentais e consolidadas, e devido a isso seria importante a realização de pareamentos e Dojos.
Buscar usar <i>frameworks</i> e tecnologias com boa documentação	Could Have	Outra forma de facilitar o desenvolvimento e a facilidade de mexer com as tecnologias seria buscar <i>frameworks</i> que facilitassem o uso das tecnologias já presentes no projeto.

Priorização MoSCoW da Elaboração de Documentos		
Atividade	Prioridade	Justificativa
Documento de Custos	Won't Have	Existe um orçamento e levantamento de custos geral do projeto. Esse custo não irá variar, pois os recursos de mídia e outros custos externos necessários já foram levantados. Mas a existência futura do documento é recomendada para analisar se o custo estimado e o custo real estão coincidindo.
Documento de Gerência de Riscos	Must Have	Devido ao não tratamento dos riscos, a organização não consegue ter um controle quanto as ações necessárias para realizar. Diante disso e por conta da ocorrência de vários riscos durante o projeto, esta documentação deve possuir uma alta prioridade.
Documento de Garantia de Qualidade	Should Have	A garantia da qualidade é necessária para acompanhar o andamento do projeto assim como o resultado das tomadas de decisão. Seria através dele que as outras áreas seriam inspecionadas e por conta disso é necessário seu desenvolvimento. Contudo por não ser uma prioridade da organização esta garantia da qualidade ele não seria tão vital quanto alguns outros.

Documento de Gerência de Configuração	Could Have	Um ambiente totalmente configurado com a documentação correta auxiliaria na adaptação de novos membros e por conta disto seria interessante uma atenção maior quanto a este. Contudo não é vital para o funcionamento do projeto atualmente já que o ambiente atual esta devidamente configurado
Rastreabilidade de Requisitos	Could Have	Devido a alguns problemas encontrados, a rastreabilidade de requisitos ajudaria como um maior argumento para conversas com outras equipes a respeito do que foi feito. Contudo dos documentos apresentados o de requisitos é aquele que já possuiu um maior esforço da equipe. Sendo assim as atividades vitais dele já foram abordadas. Logo seria interessante melhorar através da rastreabilidade, mas não é uma atividade vital.
Rastreabilidade de métricas e indicadores do projeto	Could Have	Esta etapa andaria junto com o documento de garantia de qualidade. Após a definição destas métricas e indicadores necessários, o controle destas seria feito de acordo com as especificações de qualidade. Sendo assim é importante que seja realizado, mas não é vital e deve esperar a realização de outros.

Priorização MoSCoW das Correções e Otimizações		
Atividade	Prioridade	Justificativa
Utilização da notação de Sub-Processos na modelagem	Won't Have	Caso o novo processo contenha sub-processos presentes nele, deve-se utilizar a notação correta para eles. Contudo esta atividade pode ser realizada somente após a definição do novo processo.
Separar corretamente o que são sub-processos, atividades e documentos gerados	Won't Have	A notação correta da modelagem do processo deve ser feita. Contudo estas atividades ocorrem somente durante o desenvolvimento após a definição deste novo processo, sendo assim a realização desta atividade não irá ocorrer imediatamente.

Alguma destas atividades dizem respeito à mudança de comportamento da organização e estruturação durante o processo. Já outras que envolvem a realização de documentos e atividades, foram divididas em sprints de acordo com a priorização indicada e considerando o tempo necessário para o desenvolvimento destas. A figura abaixo é a do Roadmap desenvolvido:

Legenda	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5				
Must	Integrar outras frentes no planejamento	Garantia de Qualidade	Gerencia de Configuração						
Should	Documento de Gerencia de Riscos		Buscar Frameworks						
Could									
	Rastreabilidade de Métricas								
	Rastreabilidade de requisitos								
	Práticas do XP								

Figura 5: Roadmap recomendado para realização das recomendações.

Cada sprint teria duração de 2 semanas e como o processo ocorre em ciclos em cada curso gamificado a realização do Roadmap seria no começo do novo

ciclo onde há uma necessidade maior de levantamento de documentação e novos requisitos.

## 4.2 Métricas

Quantidade de horas gasta com retrabalho	
<b>Objetivo de Medição</b>	Medir a quantidade de esforço em horas que está sendo gasto para corrigir ou modificar atividades que já haviam sido feitas.
<b>Descrição</b>	Relação entre o esforço em horas gasto durante uma sprint com retrabalho para e o tempo total gasto nela
<b>Fórmula</b>	$HT = \sum N_p H * HP$ <p>onde:</p> <p>HT = número de horas trabalhadas</p> <p>NpH = número de pontos da história de retrabalho</p> <p>HP = quantidade de horas por ponto de história</p>
<b>Escala</b>	Racional
<b>Coleta</b>	Responsável: Izaias Queiroz(Gerente), Periodicidade: A cada sprint
<b>Procedimento</b>	Verificar no final da sprint a quantidade de horas que foram gastas para realizar as atividades desta com a quantidade de horas gasta para realizar trabalhos passados através dos pontos de cada história



<b>Análise</b>	A meta desta métrica seria que a quantidade de horas gastas com retrabalho seja menor que a quantidade de horas gastas com as histórias planejadas que a sprint. E enquanto mais próximo de 0 melhor. Caso este valor esteja alto significa que o planejamento e a garantia de qualidade não esta sendo feita de forma correta, e devido a isto a equipe vem gastando maior tempo após o desenvolvimento para ajustar e modificar aquilo que já estava pronto
<b>Providência</b>	Verificar quais histórias estão necessitando de retrabalho, quem as realizou e quem validou, para ajustar o padrão de qualidade e o mínimo aceitável para que o produto seja feito.

<b>Quantidade de histórias adicionadas durante a sprint</b>	
<b>Objetivo de Medição</b>	Medir a porcentagem de histórias e atividades adicionadas durante a sprint que são adicionadas e iniciadas durante a mesma
<b>Descrição</b>	Número de histórias não planejadas que foram adicionadas em uma sprint por equipes externas

<b>Fórmula</b>	$P = (H_a / T_h) * 100$ <p>onde:</p> <p>P = porcentagem de histórias da sprint que foram adicionadas no meio</p> <p>Ha = número de histórias adicionadas no meio da sprint</p> <p>Th = Total de histórias da sprint</p>
<b>Escala</b>	Racional
<b>Coleta</b>	<p>Responsável: Izaías Queiroz (Gerente),</p> <p>Periodicidade: A cada sprint</p>
<b>Procedimento</b>	Verificar no final da sprint quantas histórias foram requisitadas que não foram planejadas no começo da sprint
<b>Análise</b>	<p>O objetivo é que o valor encontrado seja 0 ou o mais próximo possível desse valor. Caso o valor esteja muito alto significa que o processo não está sendo respeitado e o planejamento da sprint não está envolvendo ou sendo respeitado pelas outras equipes como deveria.</p>
<b>Providência</b>	<p>Detectar as equipes que mais estão solicitando novas atividades e caso não estejam participando do planejamento da sprint, adicionar eles também nesta fase. Caso já façam parte, reforçar na revisão da sprint que as histórias escolhidas ali serão as realizadas durante o período, sem acréscimo de novas.</p>

Considerando estas novas mudanças e alterações, foi modelado um novo processo que englobasse mais momentos de validação e decisão para que assim fique de acordo com as necessidades da organização. Neste novo processo o começo dele segue um modelo semelhante ao RUP, que no seu começo possui um foco maior em documentação, elaborando os documentos que foram recomendados para a equipe, a segunda parte do processo dá uma atenção maior para o levantamento de requisitos, validando todos estes antes de começar o desenvolvimento, o que não ocorria antes no antigo processo e devido a isso eram adicionados ao longo do desenvolvimento. O planejamento da Sprint deixa de forma clara que este deve ser realizado com outras frentes, e as histórias selecionadas são acordadas por todos antes de iniciar o desenvolvimento, a utilização da metodologia XP é incentivada para a passagem de conhecimento. O novo processo modelado se encontra na figura abaixo:

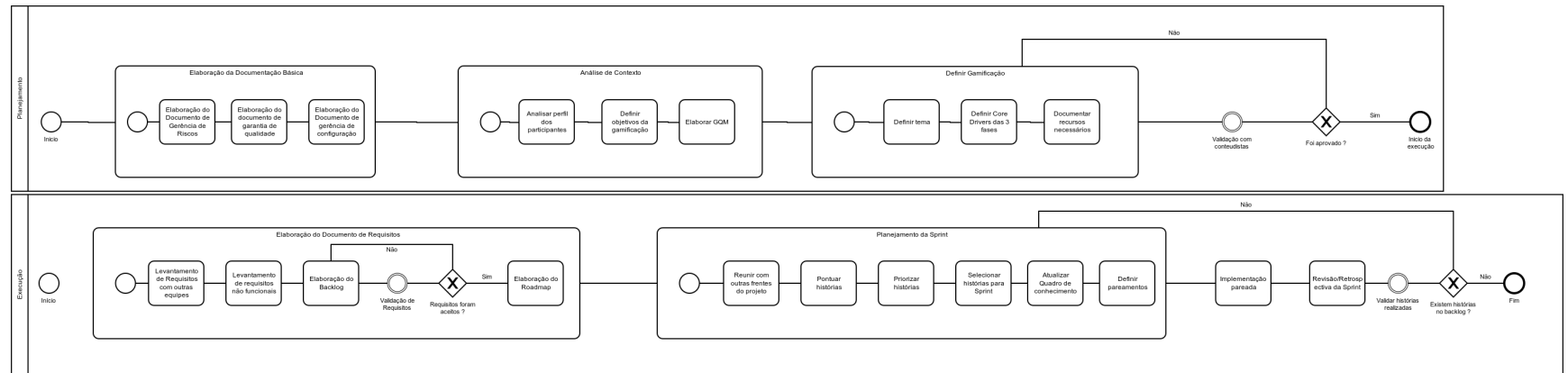


Figura 6: Novo Processo recomendado

## Referências

CLEGG, D.; BARKER, R. *Case Method Fast-Track: A Rad Approach*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1994. ISBN 020162432X.

MCFEELEY, B. *A User's Guide for Software Process Improvement*. [S.l.], 1996. Disponível em: [https://resources.sei.cmu.edu/asset\\_files/Handbook/1996.002\\_001\\_16433.pdf](https://resources.sei.cmu.edu/asset_files/Handbook/1996.002_001_16433.pdf). Acesso em: 01/06/2019.

PAULK, M. C. et al. *Key Practices of the Capability Maturity Model, version 1.1. February 1993*. [S.l.], 1993. Disponível em: [https://resources.sei.cmu.edu/asset\\_files/TechnicalReport/1993.005\\_001\\_16214.pdf](https://resources.sei.cmu.edu/asset_files/TechnicalReport/1993.005_001_16214.pdf). Acesso em: 01/06/2019.

TEAM, C. P. *CMMI for Development, Version 1.3*. Pittsburgh, PA, 2010. Disponível em: <http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=9661>. Acesso em: 01/06/2019.

## A Anexos

Identificador	Épico	Histórias	Tarefas	Prioridade	Pontuação
US01	Perfil de usuário	Eu como usuário, preciso ter uma quantidade de capital que aumenta com os itens adquiridos	-Criar um campo para visualizar seu capital -O valor no campo deve aumentar conforme ele tenha mais itens -O valor do capital deve ficar salvo no banco de dados	A	8
US02		Eu como usuário, preciso ter uma quantidade de pontos para gastar na dashboard	-Criar um campo para visualizar seus pontos - Esses pontos podem ser gastos na tela de dashboard -Os pontos devem ficar salvos no banco de dados	A	8
US03		Eu como usuário, preciso saber a quantidade de tempo de publicidade	-Criar um campo para visualizar sua quantidade de tempo de publicidade - O tempo deve ficar salvo no banco de dados	A	8
US04		Eu como usuário, preciso saber a quantidade de clientes	-Criar um campo para visualizar sua quantidade de clientes - O numero de clientes deve ficar salvo no banco de dados	A	8
US05		Eu como usuário, preciso saber a quantidade de filiais	-Criar um campo para visualizar a quantidade de filiais -O numero de filiais deve ser armazenado no banco de dados	A	8
US06		Eu como usuário, preciso saber as dicas que foram adquiridas	-Criar um campo para que o usuário possa ver as dicas adquiridas - As dicas que foram adquiridas devem ser armazenadas no banco de dados.	A	8
US07	Tela comum do curso	A tela comum do curso deve possuir um icone que encaminhe para a tela overview da empresa	-Adicionar icone para a tela de overview no canto superior esquerdo da tela - icone deve redirecionar para a tela de overview da empresa.		5
US08		A tela comum do curso deve possuir um icone que encaminhe para a tela de dashboard da empresa	-Adicionar icone para a tela de dashboard no canto superior esquerdo da tela - icone deve redirecionar para a tela de dashboard da empresa		5
US09		Eu como usuário, gostaria de um icone para dashboard com sinalização de nova notificação	-Atualizar icone para dashboard com notificação quando uma nova notificação surgir -Ao ser clicado ir para a tela desejada e fazer o icone voltar ao estado normal		5
US10		Eu como usuário, gostaria de um icone para tela de overview com sinalização de nova notificação	-Atualizar icone para dashboard com notificação quando uma nova notificação surgir -Ao ser clicado ir para a tela desejada e fazer o icone voltar ao estado normal		5
US11		Eu, como usuário, desejo visualizar minha empresa por meio de um botão de overview da empresa	-Adicionar o icone para a tela de overview - O botão deve redirecionar o usuário para a tela de overview da empresa quando clicado.		5
US12		Eu, como usuário, desejo visualizar acessar o meu dashboard por meio de um botão de dashboard	-Adicionar icone para a tela de dashboard - O botão deve redirecionar o usuário para a tela de dashboard da empresa quando clicado		5

Figura 7: Backlog do produto

## **Requisitos de Acordo com as Telas**

### **Perfil de Usuário**

- A empresa do usuário possui um nome (duvida aqui). não tem nome é genérica
- O usuário possui um valor em dinheiro.
- O usuário possui itens (comprados na loja).

### **Tela Comum do Curso**

- Deve possuir um ícone para a tela de overview da empresa.
  - Canto superior esquerdo da tela.
- Deve possuir um ícone para a tela de dashboard da empresa.
  - Canto superior esquerdo da tela.

### **Tela Questão**

- São 2 questões de estudo e uma questão de decisão (oportunidade) a cada módulo.
- O usuário sempre vai acertar as questões de estudo e vai receber dinheiro
- Caso acerte a questão de decisão o usuário recebe dinheiro e o valor é adicionado caso erre a questão de decisão o usuário não recebe dinheiro e não é penalizado.
- banco de dados.
- Caso não acerte o usuário pode refazer a questão.
- Após finalizar a questão a própria volta ao seu estado inicial.
- Ao ser respondida deve haver um feedback ao usuário informando se ele acertou ou errou.

### **Tela Oportunidade (modal)**

Figura 8: Requisitos do cliente

PLANO DE PESQUISA

1 – IDENTIFICAÇÃO DO BOLSISTA (Nome do Bolsista)

MATHEUS DE OLIVEIRA BARBOSA

2 – INSTITUIÇÃO A QUAL TEM VÍNCULO

UNIVERSIDADE DE BRASÍLIA

3 – VÍNCULO INSTITUCIONAL

ALUNO DO CURSO DE ENGENHARIA DE SOFTWARE

4 – TÍTULO DO PROJETO

Desenvolvimento de pesquisa no projeto "PESQUISA DE DESENVOLVIMENTO METODOLÓGICO E CAPACITAÇÃO PARA GOVERNANÇA E GESTÃO PÚBLICA EMPREENDEDORA NO BRASIL".

5 – INTRODUÇÃO AO PLANO DE TRABALHO (Problema a ser estudado)

A presente proposta de pesquisa contempla o desenvolvimento de um projeto de pesquisa e extensão com a finalidade desenvolver plataformas de qualificação a distância para atender até 6 milhões de trabalhadores, a partir do estudo e monitoramento de ações no âmbito do Programa de Empregabilidade Qualifica Brasil com vistas ao desenvolvimento metodológico e capacitação para governança e gestão pública da empregabilidade no Brasil.

O projeto prevê a elaboração de um módulo que englobe o estudo do estado da arte do mercado de trabalho no Brasil, visando prospectar as melhores práticas de futuro que demandam qualificação dentro das metodologias IDITAL, Especialização Indagante, Teoria de Sistemas e as estratégias de Gestão Pública e Governança de Pessoas e da Informação.

Para auxiliar o levantamento de dados, algumas bases do próprio Ministério do Trabalho poderão ser usadas, em especial as bases do Observatório Nacional do Mercado de Trabalho (<http://observatorio.mte.gov.br>), a saber: Relatório Anual de Informações Sociais - RAIS, Cadastro Geral de Empregados e Desempregados - CAGED, Base de Dados do Seguro Desemprego e a Base de Dados de Intermediação de Mão de Obra - SINE.

Os cursos ofertados serão direcionados para as demandas respondidas, por setor, com base nos dados gerados pelas pesquisas e o monitoramento das bases, com vistas a possibilitar a geração de oportunidades de trabalho e emprego para os brasileiros. A utilização de ferramentas e metodologias que permitam o desenvolvimento de políticas públicas na área de empregabilidade, apresenta-se como diferencial no programa de pesquisa a ser desenvolvido pela UfB. Além disso, os cursos de capacitação serão propostos e disponibilizados, de acordo com os resultados da pesquisa, em plataformas próprias a ser construídas. Tudo isso, visando à aplicação dos resultados respondidos e a utilização de eficiência do modelo de capacitação proposto.

Dessa forma, os objetivos específicos do presente plano de pesquisa são:

- Apoiar na pesquisa e definição dos processos e etapas de implantação da etapa 2 do projeto e apoiar no desenvolvimento de propostas para qualificação dos conteúdos da plataforma.

6 – METODOLOGIA DO PLANO DE TRABALHO

A metodologia a ser utilizada no projeto consiste na operacionalização de métodos de segurança que visem garantir as propriedades de disponibilidade, integridade e confidencialidade, desempenho das ações realizadas para o desenvolvimento da criação da plataforma integrada para o "Programa de Empregabilidade Qualifica Brasil".

PLANO DE PESQUISA

7 – ETAPAS DE TRABALHO

Etapa 1 – Apoiar a pesquisa para elaboração de propostas de qualificação dos conteúdos da plataforma IDITAL.

Etapa 2 – Apoiar o processo de implantação dos módulos de qualificação.

Etapa 3 – Construção de artigos para publicação. Compilar o relatório de pesquisa para fazer parte do Relatório de Atividades do Projeto.

8 – RESULTADOS ESPERADOS

Publicação de artigos e pesquisas desenvolvidas no âmbito nacional na área de qualificação de processos educacionais para EaD.

9 – CRONOGRAMA DE EXECUÇÃO

ETAPAS	MESES											
	1	2	3	4	5	6	7	8	9	10	11	12
1	X	X	X	X	X	X	X					
2				X	X	X	X	X	X	X	X	X
3										X	X	X

10 – REFERÊNCIAS BIBLIOGRÁFICAS

Aksoy, Z., Garrison, D. R., & Ouden, M. Y. (2009). Development of a community of inquiry in online and blended learning context. *Procedia Social and Behavioral Sciences*, 1, 1834-1838. Recuperado de <http://www.sciencedirect.com/science/article/pii/S1877042809001279>.

BENZIL, D. M.; SOUSA JR, R. T.; BENZIL, V. M. A Model of Trust Applied to the Management of Information Technology. *The International Journal of Forensic Computer Science (Impresso)*, v. 5, p. 30-37, 2010.

BERGAMINI Jr, S. Controles Internos como um Instrumento de Governança Corporativa, revista do BNDI, Rio de Janeiro, v. 12, n. 24, p. 149-188, dez. 2005.

CHAVENATO, Idalberto. *Introdução à Teoria Geral da Administração*. 4. ed. São Paulo: Makron, 1993.

Figura 9: Plano de pesquisa

Sprint 0				
Histórias Planejadas	Status			
SO1	Concluída			Concluída
TS01	Concluída			Não Concluída
TS02	Não Concluída			
TS03	Concluída			
TS04	Concluída			
TS05	Concluída			
Sprint 1				
Histórias Planejadas	Status			
US01	Concluída			
US03	Concluída			
SO2	Concluída			
TS06	Concluída			
TS02	Concluída			
Sprint 2				
Histórias Planejadas	Status			
US05	Concluída			
TS07	Concluída			
TS08	Não Concluída			
TS09	Concluída			
TS10	Concluída			
TS11	Concluída			
Sprint 3				
Histórias Planejadas	Status			
US02	Concluída			
US04	Concluída			
US07	Concluída			
TS08	Concluída			

Figura 10: Histórias planejadas x Histórias concluídas

79



ATA REUNIÃO DIA 18/05/18 9:00h ITRAC

#### Participantes

Izaias Cabral  
Gabriela  
Mateus  
Lucas

#### Assuntos:

- 1) Apresentação das atividades para a Gabriela e Lucas.
- 2) Colocação das novas demandas para a equipe
  - a) Entrega de um documento sobre gamificação (prof. Wander)
  - b) Entrega de pelo menos um capítulo do curso de Arduino para terça-feira (Izaias)
- 3) Auxílio no preenchimento dos documentos.
- 4) Apresentação de uma nova
- 5) Continuação de trabalhos

Figura 11: Ata de reunião

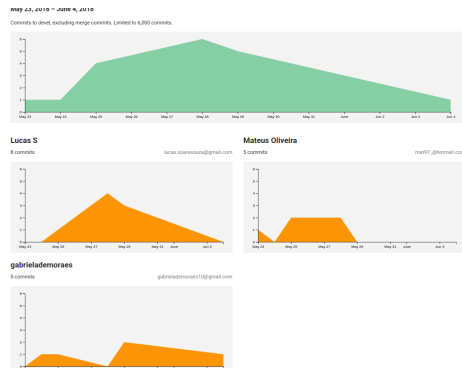


Figura 12: Gráfico de commits

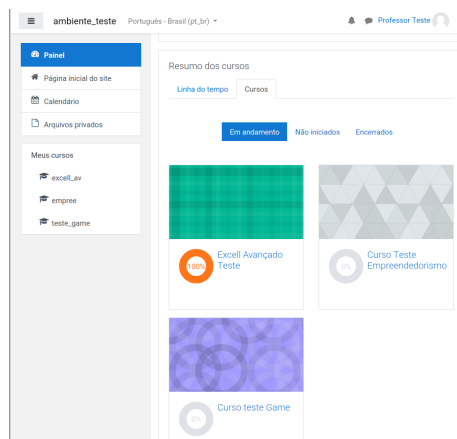


Figura 13: Ambiente de testes

Figura 14: Roadmap

