
Modelo de Processo S.C.U.

*Felipe Issamu de Melo Kamimura, Isabelle Ichikawa Yagi, Lucas
Freitas Costa, Lucas Tarumoto, Nathiely Moraes Macedo, Raniel
Carlos Bispo dos Santos*

Universidade Tecnológica Federal do Paraná - Campus Cornélio Procopio

Este documento descreve um modelo de processo de desenvolvimento de Software S.C.U.

25 de outubro de 2018

Lista de figuras

1	Modelo do Processo	6
2	Diagrama BPMN: Levantamento de Requisitos	7
3	Diagrama BPMN: Projeto e Implementação	8
4	Diagrama BPMN: Verificação e validação	9
5	Diagrama BPMN: Implantação	10
6	Diagrama BPMN: Manutenção e Evolução	10
7	Template Relatório Geral	26
8	Template de Relatório de Instalação	27
9	Template ata de reunião de finalização de fase	28
10	Template ata de reunião de incialização do Projeto	29
11	Template ata de reunião de iniciação de Sprint	30

Lista de tabelas

1	Tabela com descrição dos papeis	11
2	Enumeração e descrição das atividades vinculados com os papeis	12
3	Cronograma com as atribuições aos integrantes	18
4	Tabela de Checklist de Artefatos	20
5	Tabela de Checklist de Atividades	22

Sumário

1	Introdução	4
2	Processo	4
2.1	Papeis	11
2.2	Atividades	12
3	Qualidade do processo	16
3.1	Gerenciamento de Configuração (CM)	16
3.2	Medição e Análise (MA)	16
3.3	Monitoramento e Controle de Projetos (PMC)	17
3.4	Planejamento de Projetos (PP)	17
3.5	Garantia de Qualidade de Processos e Produtos (PPQA)	17
3.6	Gerenciamento de Requisitos (REQM)	17
3.7	Gestão de contratos de fornecedores (SAM)	17
4	Execução do projeto	17
4.1	Backlog e sprints	17
4.2	Estado atual	19
5	Referências bibliográficas	24
6	Anexos	25

1 Introdução

O presente documento descreve um processo de produção de software construído com base no Scrum [1], no Cascata [2] e no Processo Unificado [3], o SCU, que busca unir conceitos de metodologia ágil com metodologia clássica, o objetivo é um processo simples de coordenar e bastante interativo, onde a comunicação entre os envolvidos e o comprometimento com cada fase do processo são evidenciados ao longo do desenvolvimento.

Assim os integrantes que realizarão o projeto serão estes apresentados a seguir, cada um com seu respectivo papel:

- Felipe Kamimura - Integrante do Scrum Team
- Isabelle Ichikawa - Scrum Master
- Lucas Freitas - Integrante do Scrum Team
- Lucas Tarumoto - Integrante do Scrum Team
- Nathiely Moraes - Integrante do Scrum Team
- Raniel Santos - Integrante do Scrum Team

O Product Owner deve ser o cliente do projeto que será desenvolvido utilizando o processo SCU. O endereço do repositório do projeto é:

<https://github.com/FelipeKamimura/GerenciaDeConfig-SCU>.

2 Processo

O processo foi implementado com base em alguns dos elementos do Scrum [1], do Processo Unificado [3] e do modelo Cascata [2]. As fases constantes no cascata [2] foram mantidas no SCU, mas serão trabalhadas como sprints permitindo a retroalimentação e alterações futuras, essas sprints possuem reuniões no início de cada fase, onde são discutidos os objetivos da sprint, as dificuldades encontradas no desenvolvimento da sprint anterior (se houver) e os avanços alcançados (conforme templates de reuniões).

Do processo unificado foram extraídos os artefatos e os papéis [3], entretanto os papéis do scrum [1] também fazem parte do processo, sendo eles o Scrum Master, Product Owner e Scrum Team [1] (os papéis são detalhados no item 2.1 deste documento. Além das sprints tradicionais do Scrum [1], no SCU existem as subsprints, que são conjuntos de atividades, podendo ser feitas em paralelo ou não, com determinado prazo de entrega. A diferenciação em sprints e subsprints se deve às reuniões, pois as sprints dependem das reuniões realizadas antes de sua iniciação, já as subsprints não possuem reuniões formais. A retroatividade do modelo cascata também é mantida, dessa forma, caso haja mudanças nos requisitos, por exemplo, pode-se voltar pra primeira fase e realizar as alterações.

A primeira atividade a ser realizada no processo é a reunião de iniciação, o objetivo desta reunião é envolver todos os stakeholders no processo, é o momento em que o Scrum Master mostra o contexto do projeto para sua equipe e orienta os membros em relação às suas atividades, em seguida as subsprints são definidas e o prazo de entrega estipulado.

Após realizar a reunião de iniciação, dá-se o início das Sprints, que são as fases do modelo cascata. A primeira fase se trata do Levantamento de Requisitos [2], que é formado por cinco subsprints com um total de duração de 2 semanas. O prazo dessa fase se

deve a complexidade das subsprints, os requisitos são fundamentais para o sucesso do projeto, portanto o foco nessa parte do processo é muito importante. As subsprints e suas atividades de backlog podem ser visualizadas no item 4 deste documento, os artefatos gerados estão no item 4.2. Determinadas atividades não necessitam que sejam realizadas de forma ordenada, podendo ocorrer de modo concomitante quando possível.

Ao final das 2 semanas, é realizada a reunião de iniciação sprint, a organização que adota o processo define como serão realizadas as reuniões, não existe nenhuma obrigatoriedade quanto à duração ou permanecer em pé como acontece no scrum, o único requisito é que seja documentada conforme o template disponibilizado.

Com a reunião concluída e documentada, é iniciada a segunda Sprint, que é a fase de Projeto e Implementação do cascata [2], é importante ressaltar que os integrantes da equipe (definido pelo Scrum Team) podem assumir diferentes funções ao longo do processo. Esta fase tem a duração de 5 semanas com 3 subsprints, sendo 4 semanas reservadas para a subsprint de implementação, que consiste na codificação e descrição do código do sistema. Ao final da sprint, novamente é realizada uma reunião de iniciação da sprint.

A terceira sprint do SCU é a fase de Verificação e Validação, com a duração de 2 semanas, essa fase é rápida, porém deve ser realizada com muita atenção, é nela que serão identificados os problemas decorrentes no sistema por meio de testes que devem ser executados com base num conjunto de casos de teste definidos pela organização/Scrum master juntamente com a equipe, que é o plano de teste. Esta fase pode identificar mudanças nos requisitos, por isso é importante realizar uma avaliação dos testes, que ajuda a decidir se essas mudanças são realmente plausíveis, pois nada impede que um teste seja conduzido incorretamente levando a resultados incoerentes com o objetivo da aplicação. Ao esgotar o prazo da sprint e concluir os testes, é o momento de realizar a reunião e dar início a quarta sprint.

A próxima sprint consiste na fase de Implantação do cascata [2], no SCU é estabelecida num período de 2 semanas, com 4 subsprints definidas. Nessa parte do processo o sistema é disponibilizado para os usuários, portanto deve ser entregue um manual e os treinamentos devem ser realizados para utilização do software, além disso, a instalação deve ser documentada conforme o relatório de instalação disponibilizado, isto é importante para controlar a distribuição do sistema. A utilização do sistema pelos usuários pode acarretar em novas mudanças ou aparecimento de bugs, essas informações devem constar no relatório de implantação ou ser discutidas e documentadas na reunião para iniciação da próxima Sprint.

A última sprint é a fase de Manutenção e Evolução do cascata [2], essa fase possui 5 subsprints, a subsprint chamada “gerência de alteração” tem a duração de 2 semanas, é nela que são corrigidos bugs percebidos durante a utilização do sistema e adotadas rotinas para manutenção e testes constantes do produto. Além disso, essa sprint controla as versões do sistema, as possíveis alterações e informações relacionadas ao progresso e resultados do software, as subsprints responsáveis por esses dados não possuem duração, pois acompanham o desenvolvimento e a evolução do produto.

O esquema do processo pode ser visualizado na Figura 1.

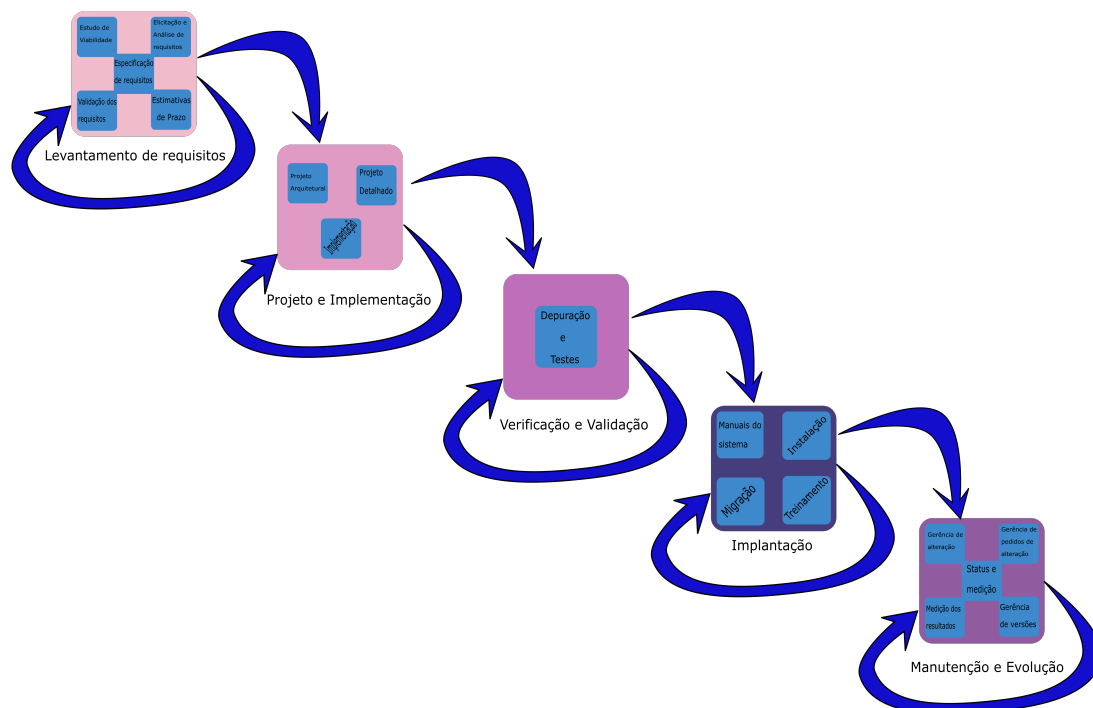


Figura 1: *Modelo do Processo*

O BPMN é o Modelo e Notação de Processos de Negócio que permite visualizar o gerenciamento dos processos de negócio do projeto que será desenvolvido. Os modelos foram divididos por fases, como podemos observar nas imagens abaixo.

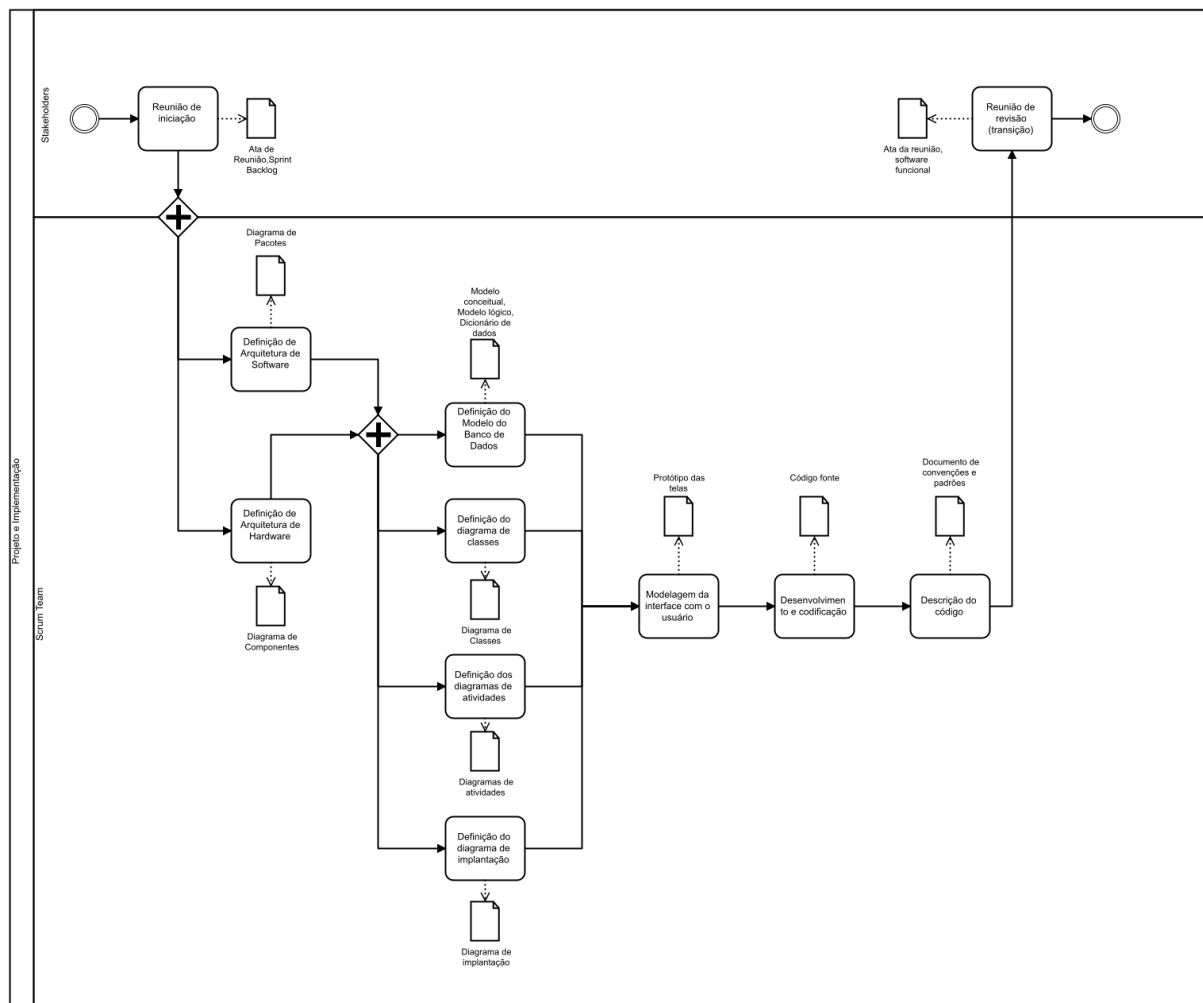


Figura 3: Diagrama BPMN: Projeto e Implementação

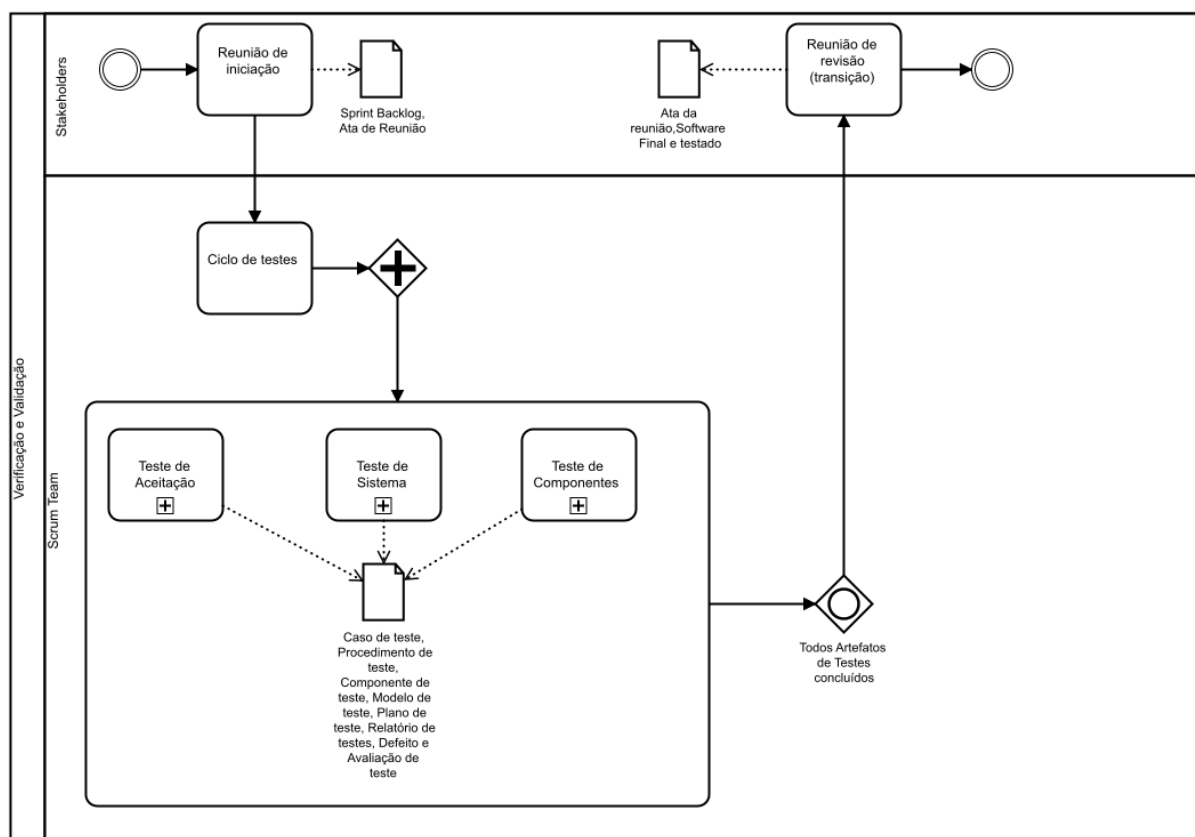


Figura 4: Diagrama BPMN: Verificação e validação

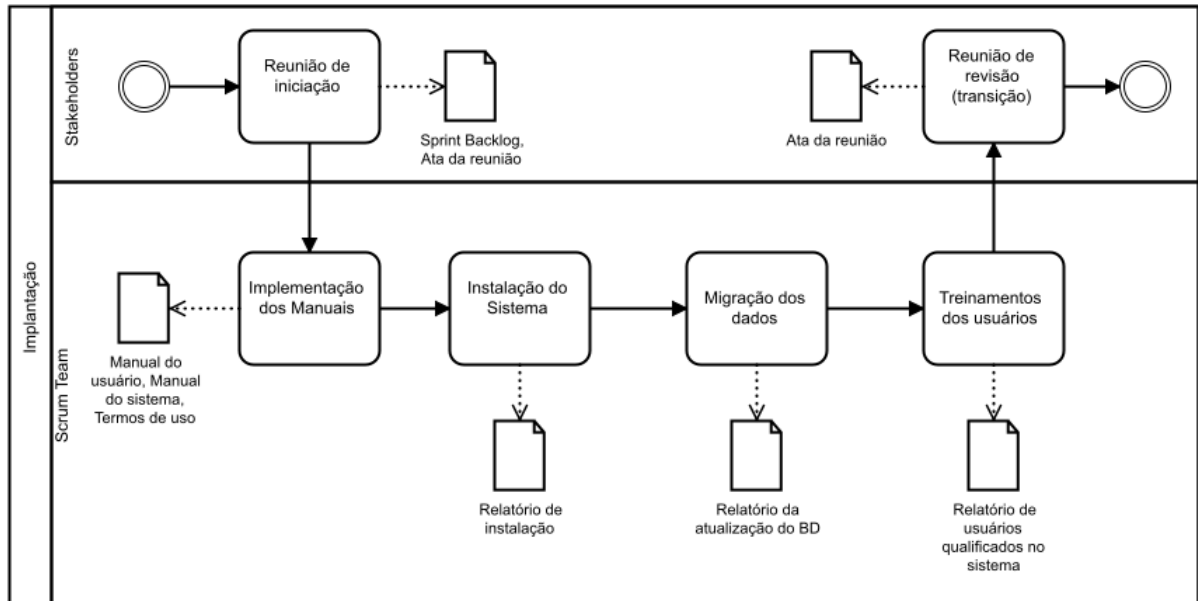


Figura 5: Diagrama BPMN: Implantação

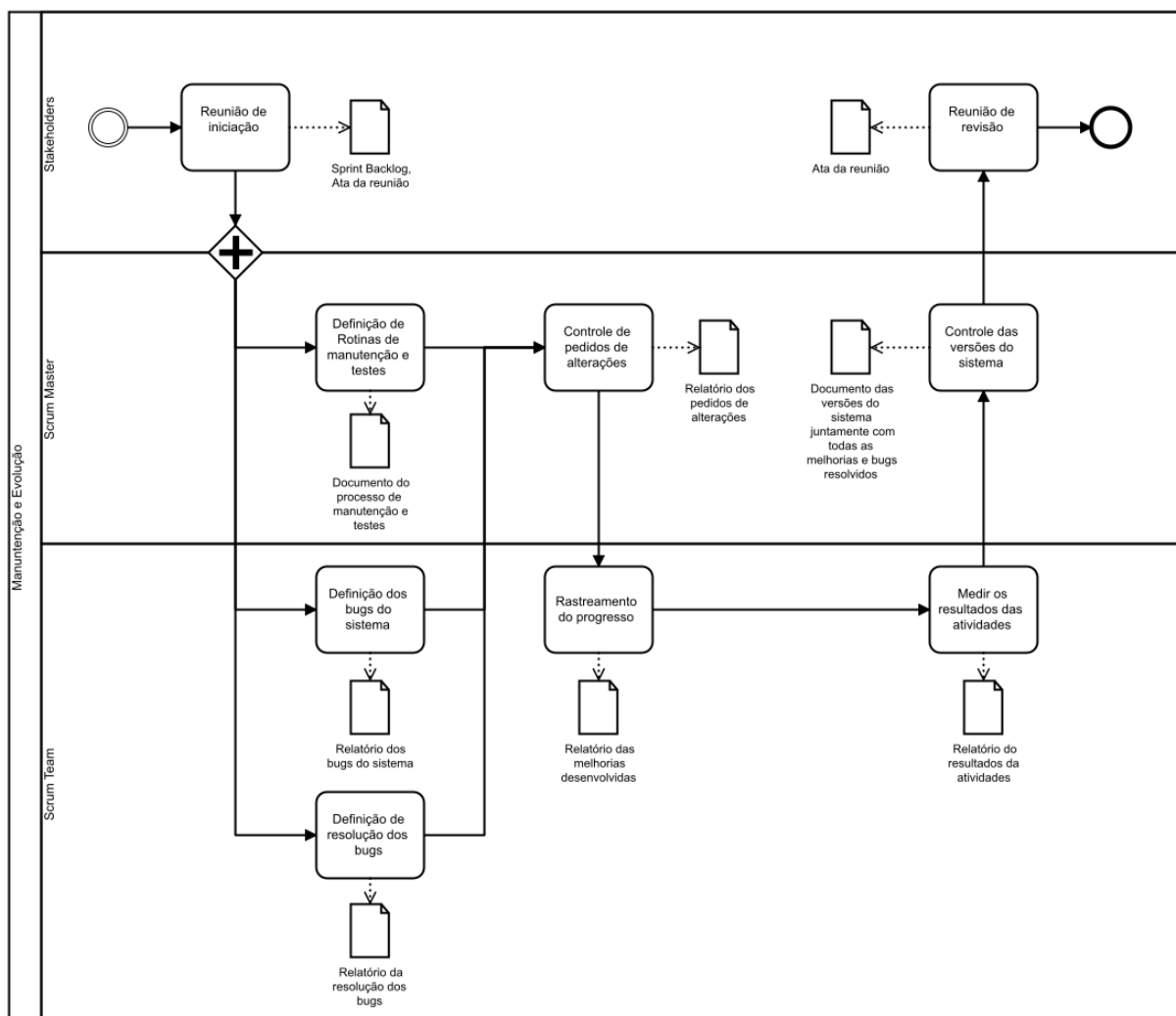


Figura 6: Diagrama BPMN: Manutenção e Evolução

2.1 Papeis

No projeto foram os papeis são agrupados conforme a metodologia Scrum, onde são divididos em Product Owner, Scrum Master e Scrum Team, devido a quantidade de pessoas envolvidas no projeto pode haver rotação e alterações das funções descritas no capítulo 1 deste projeto. O Product Owner seria aquele que possui conhecimento dos requisitos do projeto, no caso o sujeito que deseja o sistema. Já o Scrum Master pode ser exercido por qualquer pessoa da equipe e é responsável por garantir que processo seja seguido e prever possíveis entraves que possam acarretar em atrasos no cronograma. No papel de Scrum Team os integrantes são flexíveis e podem desenvolver em todas áreas do projeto, incluem as funções do gerente de projeto, arquiteto de software analista, engenheiro de teste, tester, arquiteto de banco de dados, designer, projetista de interfaces e analista de requisitos. Para melhor especificação a Tabela 1 descreve cada função existente que será necessária no projeto.//

Tabela 1: *Tabela com descrição dos papeis*

Função	Descrição	Código
Gerente de projeto	Responsável por alocar recursos, garantir qualidade e integridade do sistema e definir prioridades	F01
Arquiteto de software	Responsável por definir interações do sistema, interfaces e por buscar soluções para problemas encontrados durante a construção do sistema	F02
Analista	Mão de obra principal, responsável pela escrita dos códigos e construção do sistema	F03
Engenheiro de Teste	Responsável por planejar os testes do sistema e avaliar os resultados posteriormente	F04
Tester	Responsável por efetuar os testes do sistema	F05
Arquiteto de Banco de Dados	Define o tipo de banco de dados a ser implementado, bem como sua arquitetura	F06
Designer	Desenvolve a interface gráfica de acordo com os modelos definidos pelo projetista de interfaces.	F07
Projetista de Interfaces	Determina a forma visual dos elementos da interface gráfica do sistema. (Desenvolve protótipos de tela)	F08

Tabela 1: *Tabela com descrição dos papéis (continuação)*

Função	Descrição	Código
Analista de Requisitos	Responsável por mapear os requisitos ditos pelos stakeholders, mapear outros requisitos que irão aperfeiçoar o processo de desenvolvimento.	F09

2.2 Atividades

A Tabela 2 descreve cada uma das atividades do processo, relacionando com o respectivo papel responsável por sua execução.

Tabela 2: *Enumeração e descrição das atividades vinculados com os papéis*

	ATIVIDADE	DESCRIÇÃO DE ATIVIDADES	PAPÉIS
1.1	Reunião de iniciação	Reunião para marcar os encontros com os stakeholders para que todos saibam o que é esperado dessa primeira fase, serão definidos os tempos para a execução das sprints e como podem ser realizadas	Product Owner, Gerente de Projeto
1.2	Definição da viabilidade do projeto	Verificar que com o orçamento proposto e tecnologias prontas para serem utilizadas são suficientes para a execução do projeto	Gerente de Projeto, Arquiteto de Software, Arquiteto de Banco de Dados
1.3	Definição de escopo geral	Definir linguagem, limites e restrições, que problemas que devem ser resolvidos e como serão resolvidos	Arquiteto de Software, Arquiteto de Banco de Dados
1.4	Reunião com o Product Owner	Reunião para verificar se o cliente está de acordo com o escopo definido	Product Owner, Gerente de Projeto, Arquiteto de Software
1.5	Análise de requisitos	Onde serão realizados os modelos de casos de uso para auxiliar na análise dos requisitos necessários e verificar as atividades e funcionalidades que precisam estar presentes no software a ser desenvolvido	Analista de Requisitos, Gerente de Projeto, Arquiteto de Software
1.6	Definição dos atores	A partir dos casos de usos são listados e verificados os atores envolvidos no sistema	Arquiteto de Software
1.7	Definição da arquitetura	São especificados os casos de uso gerais sobre as funcionalidades essenciais do sistema	Arquiteto de Software, Arquiteto de Banco de Dados

Tabela 2: Enumeração e descrição das atividades vinculados com os papéis (continuação)

	ATIVIDADE	DESCRIÇÃO DE ATIVIDADES	PAPÉIS
1.8	Especificação dos requisitos	Com as funcionalidades definidas os requisitos funcionais e não funcionais podem ser mais especificados com o fim de validar a necessidade, se não estão ambíguos, e se suprem as demandas	Arquiteto de Software, Analista de Requisitos
1.9	Avaliar e validar o documento de requisitos	A partir dos relatórios dos requisitos, os requisitos são analisados em busca de verificar se possuem rastreabilidade, se estão integros, correlacionados e se o documento de requisitos são	Product Owner, Gerente de Projeto, Arquiteto de Software, Analista de Requisitos
1.10	Prototipação de interface	Com o uso de um design sprint será feito o estudo de um planejamento sobre a interface gráfica do sistema, para certificar que todos os requisitos estarão sendo preenchidos	Design, Projetista de Interface
1.11	Cálculo de pontos por função	Com o uso dos caso de uso, será realizado um cronograma inicial para se obter uma estimativa de prazo de execução do projeto	Gerente de Projetos
1.12	Reunião de revisão (transição)	Reunião de revisão de fase onde é verificado se o planejado foi executado, se há necessidade de mudança na execuções das sprints antes do inicio da proxima fase. É também elaborado o product backlog a partir de tudo que foi elicitado e também definido suas prioridades	Product Owner, Gerente de Projeto, Projetista de Interface
2.1	Reunião de iniciação	Definido o sprint backlog para que os parametros das sprints sejam passados para o scrum team	Scrum Team
2.2	Definição da arquitetura de hardware	São definidos todos os componentes (entregas) referentes ao projeto	Arquiteto de Software
2.3	Definição da arquitetura de software	Através do diagrama de pacote são definidos como os pacotes serão estruturados dentro da arquitetura do sistema	Arquiteto de Software
2.4	Definição do modelo de banco de dados	Nessa atividade é estudado os itens que precisam estar no banco de dados, se será relacional ou não relacional, atributos, entidades, chaves e relacionamentos	Arquiteto de Banco de Dados

Tabela 2: Enumeração e descrição das atividades vinculados com os papéis (continuação)

	ATIVIDADE	DESCRIÇÃO DE ATIVIDADES	PAPÉIS
2.5	Definição do diagrama de classes	Desenvolvimento dos diagramas de classes para que se tenha a visão geral e como as classes serão relacionadas umas com as outras, seus métodos e organização em relação aos objetos, além de separar elementos de design dos elementos de sistema	Analista de Requisitos
2.6	Definição dos diagramas de atividades	Definir o fluxo do funcionamento e controle das atividades realizadas pelo sistema e como se interligam	Analista de Requisitos
2.7	Definição do diagrama de implantação	Desenvolvimento do Diagrama de Implantação que especifica os requisitos mínimos para o funcionamento do sistema tais como as bibliotecas e softwares necessários, assim como as definições de hardware	Analista de Requisitos
2.8	Modelagem da interface com o usuário	Diagramação para planejamento da tela e disposição dos itens na interface e estudo de como o usuário pode ter uma boa experiência de utilização. Usa como base o protótipo inicial porém na aplicação de heurísticas e experiência de usuário	Projetista de Interface
2.9	Desenvolvimento e codificação	Codificação	Analista
2.10	Descrição do código	São documentadas as formas e padrões adotados na codificação, além de descrever determinados blocos de código-fonte	Analista
2.11	Reunião de revisão (transição)	Reunião de revisão de fase onde é verificado se o planejado foi executado, se há necessidade de mudança na execuções das sprints antes do inicio da proxima fase. Se o software atende ao escopo especificado	Product Owner, Gerente de Projeto
3.1	Reunião de iniciação	Reunião de início de fase, onde será definido como serão realizados os testes	Scrum Team
3.2	Ciclo de testes	Realização dos testes e correção no software, se necessário. Os testes devem ser relatados	Engenheiro de Teste
3.3	Teste de Componentes		Tester
3.4	Teste de Sistema		Tester

Tabela 2: Enumeração e descrição das atividades vinculados com os papéis (continuação)

	ATIVIDADE	DESCRIÇÃO DE ATIVIDADES	PAPÉIS
3.5	Teste de Aceitação		Engenheiro de Teste
3.6	Reunião de revisão (transição)	Reunião de finalização do software, realizada com os stakeholders a fim de validar o sistema de acordo com o que era esperado	Product Owner, Scrum Master, Engenheiro de Teste
4.1	Reunião de iniciação	Reunião de início de fase, onde é passado aos stakeholders como serão os procedimentos finais de desenvolvimento e como deve ser documentado as atualizações do sistema	Scrum Team
4.2	Implementação dos Manuais	Desenvolvimento dos Manuais do usuário e sistema para treinamento ou ajuda adicional para usuários, além do desenvolvimento dos Termos de Uso para o software	Arquiteto de Software
4.3	Instalação do sistema	Instalação do sistema no ambiente requisitado pelo Product Owner seguindo as especificações	Suporte
4.4	Migração dos dados	Transferência de dados antigos para o banco de dados arquitetado resumindo no mesmo atualizado	Suporte
4.5	Treinamentos dos usuários	Relatório de resultados do treinamento dos usuários que foram qualificados para o uso do sistema	Suporte
4.6	Reunião de revisão (finalização)	Documentar considerações levantadas na reunião sobre o desenvolvimento das sprints e se o produto final está conforme o TAP, se atende o escopo e se o cliente final necessita de mais alguma alteração	Product Owner, Gerente de Projeto
5.1	Reunião de iniciação	Reunião de início da fase de manutenção e melhorias no sistema, definindo como o cronograma de rotinas será feito e especificação de relatórios de melhorias e bugs encontrados	Scrum Team
5.2	Definição de Rotinas de manutenção e testes	Definição do cronograma e finalidades de manutenção e testes do sistema	Gerente de projeto, Engenheiro de Testes

Tabela 2: Enumeração e descrição das atividades vinculados com os papéis (continuação)

	ATIVIDADE	DESCRIÇÃO DE ATIVIDADES	PAPÉIS
5.3	Definição dos bugs do sistema	Relatório geral com o status de cada bug (se foi ou não resolvido, como foi resolvido, descrição do bug)	Arquiteto de Software, Engenheiro de Testes
5.4	Definição de resolução dos bugs	Relatório geral das resoluções dos bugs encontrados no sistema	Arquiteto de Software
5.5	Controle de pedidos de alterações	Status de cada pedido, se a melhoria será ou não desenvolvida	Gerente de Projeto
5.6	Rastreamento do progresso	Relatório geral com o status de cada melhoria desenvolvida e seus respectivos feedback	Gerente de Projeto
5.7	Medir os resultados das atividades	obs: serve para verificar se as atividades alcançaram realmente o objetivo delas	Gerente de Projeto
5.8	Controle das versões do sistema	Desenvolvimento do documento das versões com os Relatórios de melhorias e resoluções de bugs no sistema, obtendo um controle das versões do sistema	Gerente de Projeto
5.9	Reunião de revisão (transição)	Documentar considerações levantadas na reunião sobre o desenvolvimento das sprints	Product Owner, Gerente de Projeto

3 Qualidade do processo

Para garantir que o processo tenha qualidade, o processo proposto segue as diretrizes do modelo de qualidade CMMI for Development, com nível 2 de maturidade, que torna o processo gerenciado. Para satisfazer este modelo, abaixo são aferidas suas áreas de processo:

3.1 Gerenciamento de Configuração (CM)

Essa área de processo requer a integridade das linhas de base estabelecida e mantida, para isso são utilizados itens de configuração que compreendem a identificação de configuração, controle de configuração, contabilidade de status de configuração e auditorias de configuração. No SCU, o gerenciamento de configuração é realizado a partir do controle de versões do sistema, além do rastreamento do progresso, que geram relatórios acompanhando a documentação.

3.2 Medição e Análise (MA)

O objetivo da área de processo Medição e Análise (MA) é fornecer subsídios para desenvolver e manter uma capacidade de medição para dar suporte às necessidades de informação para

gestão. No SCU, existe a medição dos resultados das atividades, que geram informações importantes para a gestão.

3.3 Monitoramento e Controle de Projetos (PMC)

É responsável por proporcionar a visibilidade do progresso do projeto, de forma que ações corretivas apropriadas possam ser implementadas quando o desempenho do projeto desviar significativamente do plano. Para garantir esse controle, no SCU o monitoramento ocorre através de reuniões de revisões de sprints, revisões de documentos e validações, que são definidos nas atividades.

3.4 Planejamento de Projetos (PP)

Esta área deve estabelecer e manter planos visando definir as atividades de projeto. No SCU, os planos do projeto são definidos desde a primeira sprint, evoluindo durante o desenvolvimento. As tarefas incluem análise de viabilidade, escopo do projeto, requisitos, definições e protótipos.

3.5 Garantia de Qualidade de Processos e Produtos (PPQA)

Essa área de processo possui finalidade organizacional, deve fornecer visibilidade para a equipe e gerência sobre os processos e produtos de trabalho associados.

3.6 Gerenciamento de Requisitos (REQM)

Deve gerenciar os requisitos dos produtos e componentes de produto do projeto e identificar inconsistências entre esses requisitos e os planos e produtos de trabalho do projeto. A gerência de requisitos é realizada por três sprints do SCU (Elicitação e Análise de Requisitos, Especificação dos Requisitos, Validação dos Requisitos).

3.7 Gestão de contratos de fornecedores (SAM)

Área de processo onde ocorre a aquisição de produtos de fornecedores. Não consta no processo SCU.

4 Execução do projeto

Nesta seção descrevemos o cronograma previsto para o projeto e o estado atual do projeto.

4.1 Backlog e sprints

O cronograma da Tabela 3 abrange todas as sprints que serão realizadas durante o desenvolvimento e as respectivas atividades que deverão ser executadas naquele período. Cada atividade tem como responsável um integrante do projeto. A duração total prevista da execução do processo é de 17 semanas.

Tabela 3: *Cronograma com as atribuições aos integrantes*

Sprint	Atividade	Integrante	Duração
Estudo de viabilidade	Definição da viabilidade do projeto	Isabelle	2 semanas
	Definição do escopo geral	Felipe	
Elicitação e Análise de requisitos	Reunião com o Product Owner	Isabelle	
	Definição dos atores	Felipe	
	Análise de requisitos	Isabelle	
	Definição da arquitetura	Nathiely	
Especificação de requisitos	Especificação dos requisitos	Isabelle	
Validação dos requisitos	Prototipação de interface	Raniel	
	Avaliar e validar o documento de requisitos	Nathiely	
Estimativas de Prazo	Cálculo de pontos por função	Lucas Tarumoto	
Projeto Arquitetural	Definição da arquitetura de hardware	Lucas Freitas	1 semanas
	Definição da arquitetura de software	Nathiely	
Projeto Detalhado	Definição do modelo de banco de dados	Lucas Tarumoto	
	Definição do diagrama de classes	Lucas Freitas	
	Definição dos diagramas de atividades	Felipe	
Implementação	Modelagem da interface com o usuário	Lucas Tarumoto	4 semanas
	Descrição do código	Nathiely	
	Desenvolvimento e codificação	Raniel	
Depuração e Testes	Ciclo de testes	Felipe	2 semanas
	Teste de Componentes	Lucas Freitas	
	Teste de Sistema	Nathiely	
	Teste de Aceitação	Raniel	

Tabela 3: Cronograma com as atribuições aos integrantes (continuação)

Sprint	Atividade	Integrante	Duração
Implantação	Manuais do sistema	Lucas Taru-moto	2 semanas
	Instalação	Lucas Freitas	
	Migração	Nathiely	
	Treinamento	Felipe	
Gerência de alteração	Definição de Rotinas de manutenção e testes	Raniel	2 semanas
	Definição dos bugs do sistema	Felipe	
	Definição de resolução dos bugs	Isabelle	
Gerência de pedidos de alteração	Controle de pedidos de alterações	Lucas Freitas	Acompanha o projeto após o sistema ser liberado
Status e medição	Rastreamento do progresso	Lucas Taru-moto	Acompanha o desenvolvimento
Medição dos resultados	Medir os resultados das atividades	Lucas Freitas	Acompanha o desenvolvimento
Gerência de versões	Controle das versões do sistema	Lucas Taru-moto	Acompanha desenvolvimento e a evolução do sistema
			13 semanas

4.2 Estado atual

Para o controle dos artefatos gerados pelo processo e das atividades de cada fase foram utilizados os templates abaixo, que têm por objetivo facilitar a visualização do estado atual do projeto, por meio de checklists.

O template da Tabela 4 é o checklist de todos os artefatos divididos por fases.

O template da Tabela 5 é o checklist de todas as atividades divididas por fases.

Ambas as tabelas foram ordenadas por ordem cronológica, facilitando a visualização de quais artefatos ou atividades já foram realizadas e quais ainda precisam ser entregues.

Tabela 4: *Tabela de Checklist de Artefatos*

Levantamento de Requisitos
() Ata de Reunião
() Relatório da viabilidade do projeto
() Termo de Abertura do Projeto
() Relatório de Requisitos iniciais
() Modelo de casos de uso
() Atores
() Descrição da arquitetura
() Relatórios: requisitos funcionais, não funcionais e de negócio
() Documento de Requisitos
() Protótipo Inicial de interface
() Cronograma inicial
() Product Backlog, Ata da reunião
() Ata de Reunião,Sprint Backlog
Projeto e Implementação
() Diagrama de componentes
() Diagrama de pacotes

Tabela 4: *Tabela de Checklist de Artefatos (continuação)*

() Modelo conceitual, Modelo lógico, Dicionário de dados
() Diagrama de classes
() Diagramas de atividades
() Diagrama de implantação
() Protótipo das telas
() Código fonte
() Documento de convenções e padrões
() Ata da reunião, software funcional
() Sprint Backlog, Ata de Reunião
Verificação e Validação
() Caso de teste, Procedimento de teste, Componente de teste, Modelo de teste, Plano de teste, Relatório de testes, Defeito e Avaliação de teste (Teste de Componentes)
() Caso de teste, Procedimento de teste, Componente de teste, Modelo de teste, Plano de teste, Relatório de testes, Defeito e Avaliação de teste (Teste de Sistema)
() Caso de teste, Procedimento de teste, Componente de teste, Modelo de teste, Plano de teste, Relatório de testes, Defeito e Avaliação de teste (Teste de Aceitação)
() Ata da reunião, Software Final e testado
() Sprint Backlog, Ata da reunião
Implantação
() Manual do usuário, Manual do sistema, Termos de uso

Tabela 4: *Tabela de Checklist de Artefatos (continuação)*

() Sistema instalado
() Banco de dados atualizado
() Relatório de usuários qualificados no sistema
() Ata da reunião
() Sprint Backlog, Ata da reunião
Manutenção e Evolução
() Documento do processo de manutenção e testes
() Relatório dos bugs do sistema
() Relatório da resolução dos bugs
() Relatório dos pedidos de alterações
() Relatório das melhorias desenvolvidas
() Relatório do resultados da atividades
() Documento das versões do sistema juntamente com todas as melhorias e bugs resolvidos
() Ata da reunião

Tabela 5: *Tabela de Checklist de Atividades*

Levantamento	de	Requisitos
() Definição da viabilidade do projeto		
() Definição de escopo geral		
() Reunião com o Product Owner		

Tabela 5: *Tabela de Checklist de Atividades (continuação)*

() Análise de requisitos
() Definição dos atores
() Definição da arquitetura
() Especificação dos requisitos
() Avaliar e validar o documento de requisitos
() Prototipação de interface
() Cálculo de pontos por função
Projeto e Implementação
() Definição da arquitetura de hardware
() Definição da arquitetura de software
() Definição do modelo de banco de dados
() Definição do diagrama de classes
() Definição dos diagramas de atividades
() Definição do diagrama de implantação
() Modelagem da interface com o usuário
() Desenvolvimento e codificação
() Descrição do código
Verificação e Validação
() Ciclo de testes
() Teste de Componentes

Tabela 5: *Tabela de Checklist de Atividades (continuação)*

() Teste de Sistema
() Teste de Aceitação
Implantação
() Implementação dos Manuais
() Instalação do sistema
() Migração dos dados
() Treinamentos dos usuários
() Definição de Rotinas de manutenção e testes
() Definição dos bugs do sistema
() Definição de resolução dos bugs
Manutenção e Evolução
() Controle de pedidos de alterações
() Rastreamento do progresso
() Medir os resultados das atividades
() Controle das versões do sistema

Complementando, nos anexos possui os templates de relatórios Geral na figura 7, relatório de instalação na figura 8, template de finalização na figura 9, template de Inicialização de Sprint na figura 10 e Template de ata de reunião de definição de Sprints na figura 11

5 Referências bibliográficas

- [1] J. Sutherland, *SCRUM: A arte de fazer o dobro de trabalho na metade do tempo*. Leya, 2014.

- [2] I. Sommerville, “Engenharia de software-8ª edição (2007),” *Ed Person Education*.
- [3] K. Scott, *O processo unificado explicado*. Bookman, 2003.

6 Anexos

SCU - Gerenciamento de configurações

Relatório geral

Observações: _____

_____, de _____ de _____

Assinatura

Figura 7: *Template Relatório Geral*

Número de computadores : _____

[illegible]

_____, de _____ de _____

Modelo de Processo S.C.U. • Documentação do modelo

Ata de Reunião - Finalização
Data:
Requisitos Alcançados/Conformidade sistema X requisitos:
Considerações sobre segurança/integridade do sistema:
Considerações sobre usabilidade do sistema:
Outras considerações:
<div>Assinatura do Scrum Master</div>

Figura 9: *Template ata de reunião de finalização de fase*

Ata de Reunião - Inicialização do Projeto	
Data:	
Objetivo Geral do Sistema:	
Organização da Equipe:	
Definição das Sprints e tempo de duração:	
Outras Considerações:	
<div>Assinatura do Scrum Master</div>	

Figura 10: *Template ata de reunião de inicialização do Projeto*

Ata de Reunião - Sprint	
Data:	
Fase/Sprint:	
Atividades Concluídas:	
Atividades Pendentes/Incompletas:	
Problemas/Desafios Encontrados:	
Considerações para a próxima Sprint:	
<hr/>	
Assinatura do Scrum Master	

Figura 11: *Template ata de reunião de iniciação de Sprint*