

Manual de Desenvolvimento Ágil de Sistemas

Residente: Luci Lumiko Tani¹

Orientadora: Gabriela Manzolli Rowlands Lopes.

Coorientador: Guilherme Noguchi

Avaliador: Vítor Augusto Silva Ferragini

Resumo O objetivo deste Trabalho de Conclusão de Residência (TCR) é elaborar um manual sobre desenvolvimento ágil de sistemas. O tema é a racionalização de processos a partir do mapeamento do framework Scrum aplicado na Secretaria Municipal de Educação de São Paulo (SME) com um Ateliê de Software contratado pela Coordenadoria de Tecnologia da Informação e Comunicação (COTIC).

A SME atua sobre mais de 4.000 unidades escolares, mais de 60 mil educadores, e cerca de um milhão de estudantes, organizados em 13 Diretorias Regionais - DREs. Esse público demanda o uso intenso de tecnologia da informação e comunicação. Dentre os grandes desafios enfrentados para atendê-los, existe uma lacuna de Gestão do Conhecimento e o presente estudo veio para preenchê-la.

Um fator de sucesso foi a chegada de um novo Ateliê de Software. Implantaram-se as etapas iniciais do framework, a descoberta e o planejamento dos projetos, como eles se encaixam na estratégia de cada área demandante. Da mesma forma, a equipe de COTIC alinhava os postos de trabalho do Ateliê para que atendesse aos critérios de qualidade, de testes, de arquitetura e de usabilidade de sistemas, dentre outros previstos em contrato, o que foi mapeado no manual.

Espera-se que, ao publicá-lo, ele induza a cultura ágil tanto na SME e COTIC quanto em outros órgãos públicos, aumentando a eficácia e eficiência nos projetos de software, com vistas ao melhor uso de recursos públicos e assim promover maior efetividade nas políticas públicas dependentes de tecnologia da informação.

Palavras-chave: Ágil, Ateliê de Software, mapeamento, Scrum, software,

¹ Bach. em Ciência da Computação - USP, Especialização em Administração de Empresas- FGV e em TI-FGV, Especialização em Gestão Pública-Educamais. Atuou em financeiras como Banco Itaú Unibanco S.A e Banco General Motors S.A, em Fábricas de Software e na Receita Federal.

Sumário

1. Introdução.....	4
2. Justificativa.....	4
3. Metodologia.....	5
Síntese das entrevistas.	7
4. Resultados esperados	11
5. Explicar o Scrum, segundo o Guia DO SCRUM, 2020.....	11
Scrum Team (Time Scrum).....	12
Developers (Desenvolvedores).	13
<i>Product Owner</i> (Dono do Produto).	13
O <i>Scrum Master</i> (Mestre do Scrum).....	13
<i>Sprint</i> (Iteração).....	14
<i>Sprint Planning</i> (Planejamento de Sprint).....	14
<i>Daily Scrum</i> (Scrum Diário).	15
<i>Sprint Review</i> (Revisão de Sprint).	15
<i>Sprint Retrospective</i> (Retrospectiva de Sprint).	15
<i>Scrum Artifacts</i> (Artefatos do Scrum)	15
<i>Product Backlog</i> (Pendências de Produto).	16
Compromisso: Meta do Produto.	16
<i>Sprint Backlog</i> (Pendências da Sprint).	16
Compromisso: Meta da Sprint.	16
<i>Increment</i> (Incremento).	17
Compromisso: Definição de Pronto.....	17
Uma palavra sobre uma história bem escrita: INVEST.	17
6. Conclusão.....	18
Metodologia ágil com framework SCRUM.	18

Resultados positivos.....	19
Caminho promissor.	19
7. Referências.....	20
8. APÊNDICE – O Manual.....	22

1. Introdução

O manual de que trata este Trabalho de Conclusão de Residência (TCR) é produto do mapeamento de processos da metodologia Ágil com o uso do framework Scrum de desenvolvimento de sistemas aplicados na Secretaria Municipal de Educação de São Paulo (SME) com a contratação do segundo Ateliê de Software pela Coordenadoria de Tecnologia da Informação e Comunicação (COTIC).

A pesquisa contou com a orientação de Gabriela M. R. Lopes, da Divisão de Desenvolvimento de Sistemas (DISIS) da COTIC, e coorientação de Guilherme Noguchi, da Secretaria Municipal de Inovação e Tecnologia (SMIT).

O processo, conforme o framework SCRUM descrito por Jeff Sutherland, evolui adaptando-se ao contexto da organização, desde que as diretrizes e pilares sejam respeitados. Logo, para que o manual possa ser replicado por outros órgãos ou entidades, é necessário que se entenda os conceitos do Ágil e do SCRUM, os procedimentos adotados pela SME / COTIC, as responsabilidades dos atores envolvidos no processo e as premissas atendidas na contratação de uma empresa desenvolvedora de sistemas sob medida, o Ateliê de Software. Haverá necessidade de adaptações em alguns casos.

2. Justificativa

A SME provê atendimento a alunos, professores e demais funcionários da Rede Municipal de Ensino de São Paulo, composta por mais de 4.000 unidades escolares, mais de 60 mil educadores, e cerca de um milhão de alunos (dados do sistema EOL, em 30/11/2024), organizados em 13 Diretorias Regionais - DREs. Esses públicos demandam diversos serviços de Tecnologia da Informação e Comunicação (TIC) na forma de sistemas digitais, como lançamento de notas e presenças de alunos, gerenciamento de profissionais da rede e atribuição de aulas e oferta de alimentação. Para suprir essa demanda, a SME tem como responsabilidade a cultura digital, a modernização tecnológica e gestão de indicadores, mais de 23 sistemas em uso, vários serviços e sistemas em desenvolvimento.²

² Estudo Técnico Preliminar página 6, Número do Processo: 6016.2024/0007204-4. SEI, 2024, e dados consultados em 30/12/2024.

Assim, a secretaria possui um planejamento estratégico para aprimorar as ofertas de TIC, como integração dos sistemas internos, unicidade da base de dados, login único, painéis de indicadores para gestores e aprimoramento dos sistemas existentes com novas funcionalidades. Todavia, não há uma equipe de profissionais de tecnologia da informação dedicada ao desenvolvimento de sistemas na SME e há uma pressão por aumento de escala para atualização tecnológica e atendimento no tempo requerido pelas políticas públicas de educação. Portanto, optou-se por contratar um Ateliê de Software, uma empresa que desenvolva soluções de sistemas digitais (software) sob medida, e adotasse uma metodologia ágil para administrar o processo de desenvolvimento de sistema.

Os principais desafios em desenvolvimento de sistemas incluem equilibrar demandas e prioridades com recursos orçamentários, trabalhar com reduzido número de pessoas, gerir o contrato de Ateliê de Software (contrato, entregas, pagamentos, aditamentos, multas), treinar os donos dos projetos da área demandante, exercer dupla fiscalização das entregas (técnica e administrativa), montar e/ou manter a infraestrutura para hospedar os sistemas e responder ao Tribunal de Contas do Município (TCM-SP).

Contudo, o maior problema é a falta de documentação ou mapeamento do processo de gestão do projeto de software. É preciso registrar a sequência das etapas, as decisões, os eventos, os artefatos produzidos, os atores e suas responsabilidades. Não havia gestão desse conhecimento, ele estava na mente das pessoas e parte se perdia com a saída tanto de servidores da SME quanto dos desenvolvedores da contratada, inclusive com a contratação do novo Ateliê. Como consequência, novos colaboradores também seriam impactados negativamente pela falta de uniformidade no processo.

3. Metodologia

As metodologias aplicadas durante a pesquisa foram: curso presencial Programa de Treinamento em Gestão de Processos da SME-SP³, entrevistas semiestruturadas, mapeamento de processos BPMN (Business Process Model and Notation) utilizando o software gratuito Bizagi, Kaizen (melhoria contínua), guias e manuais Copicola, revisão bibliográfica, produção do Manual em Linguagem Simples.

³ AMARAL, T. M. 2024.

Não há dados históricos a coletar, pois o novo Ateliê de Software está formando as equipes de desenvolvimento de sistemas e nenhuma entrega de projeto foi feita até presente data⁴. Igualmente, não há como compará-lo com o anterior, antes do mapeamento do processo, pois o número de desenvolvedores atuais quase triplicou, o contrato atual possui mais cláusulas de qualidade e fiscalização das entregas, aplicação de multas e outros controles, diferentes perfis profissionais de desenvolvimento de software.

Diferenças conceituais⁵

Metodologia de Desenvolvimento Ágil: Técnica de desenvolvimento alternativa ao modelo tradicional, que tem por objetivo aprimorar o valor das entregas durante o desenvolvimento de um projeto, fracionando o todo em entregas incrementais, chamadas de iteração, trabalhando em times auto-organizados, com equipes multidisciplinares, para atingir uma meta estabelecida a cada fase, até a conclusão final da entrega total do projeto. Cada iteração é como um projeto de software em miniatura, dura entre uma e quatro semanas, e inclui todas as tarefas necessárias para implantar um incremento de nova funcionalidade: planejamento, análise de requisitos, projeto, codificação, teste e documentação. A SME /COTIC tomou como parâmetro a metodologia Scrum, mas o termo Desenvolvimento Ágil identifica um grupo de metodologias de desenvolvimento que adotam os seguintes princípios: indivíduos e interações mais que processos e ferramentas; software em funcionamento mais que documentação abrangente; Colaboração com o cliente mais que negociação de contratos; e responder a mudanças mais que seguir um plano.

Metodologia Scrum: Scrum é uma metodologia ágil para gestão e planejamento de projetos de software. Metodologias ágeis de desenvolvimento de software são iterativas, ou seja, o trabalho é dividido em iterações, que são **chamadas** de Sprints. O Sprint representa um período (geralmente de uma a quatro semanas) dentro do qual um conjunto de atividades deve ser executado. As funcionalidades a serem implementadas em um projeto são mantidas em uma lista que é conhecida como Product Backlog. No início de cada Sprint, faz-se um Sprint Planning Meeting, ou seja, uma reunião de planejamento na qual o Product Owner (requisitante, ou dono do produto) prioriza os itens do Product Backlog e a equipe seleciona as atividades que ela será capaz de implementar durante o Sprint que se inicia. As tarefas alocadas em um Sprint são transferidas do Product Backlog para o Sprint Backlog. A cada dia de uma Sprint, a equipe faz uma breve reunião (normalmente de manhã), chamada Daily Scrum. O objetivo é disseminar conhecimento sobre o que foi feito no dia anterior, identificar impedimentos e priorizar o trabalho do dia que se inicia. Ao final de um Sprint, a equipe apresenta as funcionalidades implementadas em uma Sprint Review Meeting (reunião de revisão da Sprint). Finalmente, faz-se uma Sprint Retrospective (retrospectiva da Sprint) e a equipe parte para o planejamento do próximo Sprint. Assim reinicia-se o ciclo.

⁴ Dezembro de 2024.

⁵ Extraído do Termo de Referência páginas 75 e 76, Número do Processo SEI: 6016.2024/0007204-4. São Paulo, 2024.

Percebe-se que, dependendo do autor, Scrum é uma metodologia ágil, mas a partir deste ponto, Scrum será tratado como um framework, isto é, em gestão de projetos, um framework é uma estrutura de trabalho que fornece um conjunto de diretrizes, processos e ferramentas para organizar e conduzir um projeto de forma mais eficiente. É como um mapa que ajuda a equipe a seguir um caminho claro e estruturado, para garantir que as tarefas sejam realizadas de acordo com as melhores práticas e os objetivos do projeto.

Inicialmente, o Scrum foi estudado pela bibliografia de Jeff e seu filho J. J. Shuterland. A palavra “scrum” tem origem inglesa e significa “um grupo compacto de jogadores que trabalham juntos como uma unidade para alcançar um objetivo comum em um jogo de rugby” (SUTHERLAND, J., 2014).

A seguir, os atores de COTIC/DISIS foram entrevistados para se extrair o fluxo de trabalho do processo em estudo: Gabriela M. R. Lopes, Juliana L. Demay, Ronaldo J. Silveira, Vítor A. S. Ferragini. Cada questionário era composto por 2 partes, uma comum com questões abertas e outra específica. A parte específica era estruturada⁶ em 2 tipos: um mais voltado à gestão do processo e outro mais técnico sobre TIC, pois há 2 pessoas de cada perfil na equipe. O Coordenador de COTIC, Sílvio A. de Vasconcelos Jr., foi entrevistado para registrar dados sobre orçamento, história da área, estrutura das equipes, principais desafios da área, o contexto da pesquisa, e, por fim, como poderia adicionar valor para melhorar as entregas para as áreas-fim da Secretaria de Educação.

As entrevistas foram presenciais e individuais. O questionário estruturado foi de múltipla escolha com uma opção “outro” em branco para informação adicional. As não estruturadas, abertas, foram sobre o fluxo de trabalho: desenhe ou descreva como é feito hoje; quais as etapas do processo, qual a mais importante na sua visão, quem são os atores em cada etapa; a metodologia satisfaz ou visualiza melhorias na metodologia; quais os maiores desafios; sugestão como superá-los; e assim por diante.

Síntese das entrevistas.

⁶ REBELLO, 2023.

Os entrevistados apontaram como riscos no desenvolvimento de software do maior para o menor risco: mudanças de prioridade do Gabinete, conflito entre gestores de TI e fornecedores, conflitos entre gestores de TI e áreas demandantes de projetos, fiscalização precária, testes incompletos, postos vazios por longo tempo, ausência de cultura ágil na SME integrando área finalística e área técnica, superdimensionamento de ações simples de entrega de software com baixo valor estratégico e turnover (rotatividade) alta.

Como consequência, desenvolvimento de soluções de baixo valor estratégico, custo elevado de desenvolvimento, limitação ao processo criativo dos times de produção de software, atrasos nas entregas, baixa qualidade de software.

Como evitar esses riscos: adotar premissas como qualificação mínima dos profissionais que irão prestar serviços técnicos especializados, fixação de critérios de aceitação dos serviços prestados, definição dos níveis mínimos de serviço e de qualidade, utilização de metodologia ágil durante o projeto de software.

E as diretrizes apontadas como fundamentais para aumentar a qualidade das entregas de software foram: aprendizado com o Ateliê contratado anterior, clareza do serviço contratado, cláusulas de responsabilidade de cada papel, testes exaustivos e de qualidade, software livre, licença MIT, equipes bem integradas, apoio do Gabinete e das Coordenadorias, COTIC explorar o contrato para prover inovações demandando projetos também, cooperação externa de consultorias, aprender com erros e acertos e evoluir.

Papéis esperados⁷:

Papel	COTIC	Área Finalística	Ateliê de SW	N/A
Gestor de Contrato	X			
Fiscal Técnico da solução	X			
Fiscal Administrativo do Contrato	X			
Product Owner/Dono do Produto		X		
Portfolio Manager/Gerente de Portfólio			X	
Scrum Master			X	
Arquiteto de Sistemas			X	
UX/UI			X	
Designer de Serviço			X	
DEV Team/Time de Desenvolvimento			X	

⁷ Síntese das entrevistas na COTIC.

Desafios e lições aprendidas com o 1º Ateliê de Software (2018-2023).⁸

Desafio	Causa	Ação
A equipe desconhecia roteiro para entendimento do problema e cocriação da solução.	Desenvolvimento de software pelo método em cascata/preditivo.	Adotar um guia ou manual prático da metodologia.
Somente uma pessoa conhecia práticas ágeis do modelo de Ateliê.	Desenvolvimento de software pelo método em cascata/preditivo.	Participar de programas de capacitação.
As informações que chegavam para o time de desenvolvimento eram insuficientes.	As disciplinas de UX e UI eram ainda pouco conhecidas.	Definir os entregáveis com exemplos. Definir um perfil de UX/UI no Ateliê.
Os POs não tinham maturidade para traduzir as necessidades de negócio para um bom mapeamento de história de usuário.	Os POs não tiveram contato prévio com a metodologia ágil e poucos haviam participado em projetos de desenvolvimento de sistemas.	COTIC deve dar apoio aos POs e formá-los no tema.
Os ritos não eram seguidos com o propósito que deveriam.	Por mais que o TR exigisse experiência do Ateliê com metodologias ágeis, havia algumas lacunas na prática.	Alinhar cada rito, sua função e os entregáveis esperados no início do contrato, antes de qualquer projeto.
Dificuldade ou ausência de informações importantes documentadas.	Falta de clareza no contrato.	Adotar documentos mínimos de referência do mercado.
A qualificação dos profissionais do ateliê anterior ficou aquém do esperado.	Todas as contratações ficaram a cargo do ateliê, sem participação de Cotic.	Cotic realizar diligências com cada contratado do ateliê para avaliar a aderência à metodologia ágil, adoção de paradigmas de mercado para qualidade de desenvolvimento, testes e critérios de aceitação de sistemas. Senioridade e experiência.
Retrabalhos a cada projeto, tanto de disposição das funcionalidades quanto de identidade visual e navegação	Falta de arquitetura prévia ao desenvolvimento das funcionalidades.	Definir perfil de arquiteto e de designer de sistema UX/UI no ateliê. Eles devem atuar previamente a qualquer programação de funcionalidades.
O conhecimento por si só não é suficiente para manter as pessoas alinhadas.	Também são necessários os valores, as crenças, a visão e a missão.	Disseminar cultura organizacional que direcione para transformação digital. Obter apoio do Gabinete.

⁸ Entrevista do Diretor de SME/COTIC/DISIS à época, Guilherme Noguchi.

Comparativo evolutivo⁹:

Período	2018-2023	2024-atual
Ateliê de Software	Amcom	Spassu
Equipe no Ateliê	42 postos	108 postos
Envolvimento de Cotic	Desde a concepção	Desde a concepção
Pagamento	Pagamento de valor fixo por sprint executada	alocação de profissionais de TI
Linguagens de Programação	JavaScript, Python, PHP, C#, SQL, JAVA, C/C++	JavaScript, Python, PHP, C#, SQL
Tecnologias Emergentes	Big Data, UX/UI, Lean Inception e Design Thinking, perfis de Arquiteto de Software e Scrum Master	Big Data, UX/UI, Lean Inception e Design Thinking, Cloud Computing, perfis de Gerente de Portfólio, Arquiteto de Software, Scrum Master, UX/UI, Quality Assurance, Designer de sistema
Tipo de trabalho	Presencial	Remoto
Licença de software	MIT	MIT ¹⁰
Ferramentas	Azure DevOps, GitHub	Azure DevOps, GitHub
Grau de maturidade ágil na SME	3	4

A próxima etapa de pesquisa foi a mais desafiadora, com a compilação de todas as informações num único mapa de processo segundo o BPMN, utilizando versão gratuita do software Bizagi e apresentação à DISIS para refinamento e confecção do manual final.

⁹ Síntese das entrevistas na COTIC.

¹⁰ Licença MIT, código aberto. A licença é permissiva, qualquer pessoa que obtém uma cópia do software e seus arquivos de documentação associados pode lidar com eles sem restrição, incluindo sem limitação os direitos a usar, copiar, modificar, mesclar, publicar, distribuir, vender cópias do software. As condições impostas para tanto são apenas manter o aviso de copyright e uma cópia da licença em todas as cópias do software.

Como já pontuado, o SCRUM é um framework adaptável à organização em que é adotado, portanto, seria natural que diferenças emergissem no início dos trabalhos entre DISIS e o Ateliê. Quando este iniciou os trabalhos na Secretaria Municipal de Educação, no final de 2024, alterou-se profundamente o processo de construção de software, nas fases anteriores ao SCRUM. Tornou-se, por isso, desejável adotar um corte no tempo e uma fotografia do processo para que as alterações no TCR não se dilatassem e impossibilitassem sua finalização e entrega.

Uma reunião foi feita em 27/03/2025 para revisar o processo, com vistas a representar o estado esperado do SCRUM, dado que todo o processo de execução de um projeto não poderá ser acompanhado integralmente até a entrega deste trabalho.

4. Resultados esperados

Uniformização do processo; sistemas mais robustos; entregas aderentes ao solicitado pela área-fim; menos retrabalhos; redução do tempo dispendido em cada projeto; atuação imediata sobre os problemas ou impedimentos; gestão do conhecimento.

O SCRUM, pelas entregas parciais, gera a confiança de que o projeto está em andamento e documentado. Se, por uma eventualidade, um membro da equipe sair, a sua parte já estará feita e entregue até a sua saída. A equipe poderá continuar o projeto a partir desse ponto, por estar documentado.

5. Explicar o Scrum¹¹, segundo o Guia DO SCRUM, 2020¹².

SCRUM é um framework que ajuda pessoas, times e organizações a gerar valor por meio de soluções adaptativas para problemas complexos. Em suma, *Scrum* requer um *Scrum Master* para promover um ambiente onde:

1. Um *Product Owner* particiona um problema complexo em histórias num Product Backlog.

2. O *Scrum Team* transforma uma seleção do *Product Backlog*, *Sprint Backlog*, em um *Increment* durante uma *Sprint*.

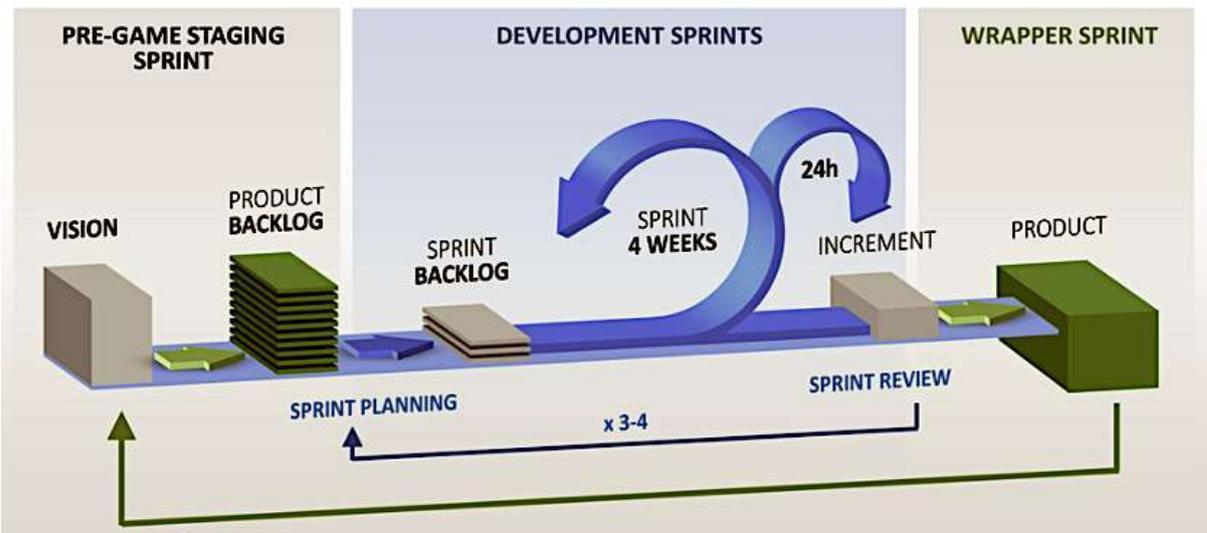
¹¹ Agradecimento a Ronaldo J. Silveira.

¹² SCHWABER, K.; SUTHERLAND, J., 2020

3. O *Scrum Team* e seus *stakeholders* inspecionam os resultados e planejam a próxima *Sprint*.

4. Repita de 1. a 4.

SCRUM – PROCESS



Esta Foto de Autor Desconhecido está licenciado em [CC BY](#)

Scrum é baseado em empirismo e *Lean Thinking*¹³. Empirismo, por ter surgido e evoluído com experiências e tomadas de decisão dos autores e diversos colaboradores pelo mundo, desde a década de 90. *Lean Thinking*, por focar no que é essencial.

Seus pilares são a transparência, a inspeção e a adaptação. A transparência permite a inspeção. A inspeção habilita a adaptação. A adaptação fortalece os 5 valores: compromisso, foco, abertura, respeito e coragem. Compromisso com objetivos e membros do time, foco na *Sprint* para atingir as metas, abertura a desafios, respeito mútuo e independência, coragem de fazer a coisa certa e trabalhar com problemas difíceis.

Scrum Team (Time Scrum).

¹³ *Lean Thinking*, ou Pensamento Enxuto, é uma metodologia de gestão que visa a redução de desperdícios e a maximização do valor para o cliente através da otimização de processos e da eliminação de atividades que não agregam valor. É uma abordagem que busca a eficiência e a melhoria contínua com foco em criar mais valor com menos recursos

A unidade fundamental do Scrum é um Scrum Team, composto por um Scrum Master, um Product Owner e Developers, num total menor que 10. Não há hierarquia, é uma unidade coesa de profissionais focados em um objetivo de cada vez, a Meta do Produto. São multifuncionais (*cross-functional* ou *XFN*) e autogerenciáveis, decidem internamente quem faz o quê, quando e como.

O *Scrum Team* é responsável por todas as atividades relacionadas ao produto, ou seja, por criar um *Increment* valioso e útil a cada *Sprint*.

Developers (Desenvolvedores).

Developers, atuam como programadores, Quality Assurance, gestor de banco de dados, UX/UI, arquiteto de sistemas, designer de serviço, por exemplo. São as pessoas do *Scrum Team* comprometidas em criar qualquer aspecto de um Incremento utilizável a cada *Sprint*.

Responsabilidades: criar um plano para a *Sprint*, o *Sprint Backlog*; introduzir qualidade aderindo a uma Definição de Pronto; adaptar seu plano a cada dia em direção à meta da *Sprint*; responsabilizar-se mutuamente como profissionais.

Product Owner (Dono do Produto).

O *Product Owner* é responsável por maximizar o valor do produto resultante do trabalho do *Scrum Team*.

Ele gerencia o *Product Backlog*, que inclui: definir e comunicar a meta do produto; escrever claramente e priorizar os itens do *Product Backlog*; garantir que o *Product Backlog* seja compreensível.

Para que os *Product Owners* tenham sucesso, toda a organização deve respeitar suas decisões.

O Scrum Master (Mestre do Scrum).

O *Scrum Master* ajuda todos a entender a teoria e a prática do *Scrum* conforme definido no Guia do *Scrum*, tanto no *Scrum Team* quanto na organização. Ou seja, ele é responsável pela eficácia do *Scrum Team*, *incentiva-o* a melhorar suas práticas, são líderes que servem ao *Scrum Team* e à organização como um todo: treina os membros do time em autogerenciamento e cross-funcionalidade (*XFN*); ajuda o *Scrum Team* a se concentrar na criação de incrementos de alto valor que atendem à Definição de Pronto; remove os impedimentos ao progresso do *Scrum Team*; garante que os eventos *Scrum* sejam positivos, produtivos e mantidos dentro do *Timebox*.

O *Scrum Master* auxilia o *Product Owner*: a definir a meta do Produto e gerenciamento do *Product Backlog*; a planejar o produto para um ambiente complexo; a facilitar a colaboração dos stakeholder.

O *Scrum Master* serve a organização: ao organizar a adoção do *Scrum*; ao ajudar os funcionários e os stakeholders a compreender e aplicar uma abordagem empírica para trabalhos complexos; ao remover barreiras entre stakeholders e *Scrum Teams*.

Sprint (Iteração).

Sprints são o coração do *Scrum*, onde ideias são transformadas em valor.

São eventos de duração fixa de um mês ou menos. Uma nova *Sprint* começa imediatamente após a conclusão da anterior.

Fazem parte de uma *Sprint*: *Sprint Planning*, *Daily Scrum*, *Sprint Review* e *Sprint Retrospective*.

Durante a *Sprint*: nenhuma mudança é feita que coloque em risco a meta da *Sprint*; a qualidade não diminui; o *Product Backlog* é refinado conforme necessário; o escopo pode ser esclarecido e renegociado com o *Product Owner* conforme mais se compreende os itens do *Sprint Backlog*.

Sprints permitem previsibilidade, garantindo a inspeção e adaptação do progresso em direção a uma meta do Produto periodicamente. *Sprints* devem ser breves para gerar mais ciclos de aprendizagem e limitar os riscos de custo e esforço.

Sprint Planning (Planejamento de Sprint).

A *Sprint Planning* inicia a *Sprint* pelo trabalho colaborativo de todo o *Scrum Team* ao definir o trabalho a ser realizado. Esse plano é criado com o *Product Owner*, os *Developers* selecionam itens do *Product Backlog* para incluir na *Sprint* atual. O *Scrum Team* pode refinar esses itens durante este processo, o que aumenta a compreensão e a confiança.

Para cada item do *Product Backlog* selecionado, os *Developers* planejam o trabalho necessário para criar um Incremento que atenda à Definição de Pronto. Eles decompõem itens do *Product Backlog* em itens de trabalho menores de um dia ou menos.

A Meta da *Sprint*, os itens do *Product Backlog* selecionados para a *Sprint*, mais o plano para entregá-los são chamados juntos de *Sprint Backlog*.

Limite de 8h/mês.

Daily Scrum (Scrum Diário).

O propósito da *Daily Scrum* é inspecionar o progresso em direção a Meta da *Sprint* e adaptar o *Sprint Backlog* conforme necessário. Dura cerca de 15 minutos e é para os *Developers* do *Scrum Team*, é realizado no mesmo horário e local, todos os dias úteis da *Sprint*. o *Product Owner* ou o *Scrum Master* podem participar.

As *Daily Scrums* melhoram a comunicação, identificam os impedimentos, promovem a rápida tomada de decisões e eliminam a necessidade de outras reuniões.

Contudo, os *Developers* podem ajustar seu plano ao longo do dia em discussões mais detalhadas sobre a adaptação ou replanejamento do resto do trabalho da *Sprint*.

Sprint Review (Revisão de Sprint).

O propósito da *Sprint Review* é inspecionar o resultado da *Sprint* e determinar as adaptações futuras. O *Scrum Team* apresenta os resultados para os principais stakeholders e o progresso em direção a Meta do Produto.

O *Product Backlog* também pode ser ajustado para atender a novas oportunidades. Limite de 4h/mês.

Sprint Retrospective (Retrospectiva de Sprint).

O propósito da *Sprint Retrospective* é planejar maneiras de aumentar a qualidade e a eficácia do processo de Scrum.

O *Scrum Team* inspeciona como foi a última *Sprint* em relação a indivíduos, interações, processos, ferramentas e sua Definição de Pronto. O *Scrum Team* discute o que deu certo durante a *Sprint*, quais problemas encontraram e como esses problemas foram (ou não) resolvidos.

As melhorias mais impactantes podem até ser adicionadas ao *Sprint Backlog* para a próxima *Sprint*.

A *Sprint Retrospective* conclui a *Sprint*. É limitada pelo Timebox de no máximo três horas por mês.

Scrum Artifacts (Artefatos do Scrum)

Os artefatos do *Scrum* representam trabalho ou valor.

Cada artefato contém um compromisso para garantir que ele forneça informações que aumentem a transparência e o foco para medir o progresso: para o *Product Backlog*, é a Meta do produto; para o *Sprint Backlog*, é a Meta da *Sprint*; para o incremento, é a Definição de Pronto.

Product Backlog (Pendências de Produto).

O *Product Backlog* é uma lista ordenada do que é necessário para melhorar o produto. É a única fonte de trabalho realizado pelo *Scrum Team*.

Os itens do *Product Backlog* em uma *Sprint* são selecionados no *Sprint Planning*. O *Product Backlog refinement* é o ato de dividir e incluir definição adicional aos itens do *Product Backlog* para ter itens menores e mais precisos. Esta é uma atividade contínua para adicionar detalhes, como descrição, ordem e tamanho. Os atributos geralmente variam de acordo com o domínio de trabalho.

Os *Developers* que farão o trabalho são responsáveis pelo dimensionamento. O *Product Owner* pode influenciar os *Developers*, ajudando-os a entender e selecionar trade-offs (trocas de itens).

Compromisso: Meta do Produto.

A Meta do Produto descreve um estado futuro do produto serve como um alvo para o *Scrum Team*. A Meta do produto está no *Product Backlog*. O restante do *Product Backlog* define “o que” cumprirá a Meta do Produto.

Um produto é um veículo para entregar valor, tem um limite claro, stakeholders conhecidos, usuários ou clientes bem definidos. Um produto pode ser um serviço, um produto físico ou algo mais abstrato.

A Meta do Produto é o objetivo de longo prazo para o *Scrum Team*. Eles devem cumprir (ou abandonar) um objetivo antes de assumir o próximo.

Sprint Backlog (Pendências da Sprint).

O *Sprint Backlog* é composto pela Meta da Sprint (o porquê), o conjunto de itens do *Product Backlog* selecionados para a *Sprint* (o que), bem como um plano de ação para entregar o Incremento (como).

O *Sprint Backlog* é um plano feito por e para os *Developers*. É uma imagem do trabalho que os *Developers* planejam realizar durante a *Sprint* para atingir a Meta da Sprint. Consequentemente, o *Sprint Backlog* é atualizado ao longo da *Sprint* conforme mais é aprendido. Deve ter detalhes suficientes para que eles possam inspecionar seu progresso na *Daily Scrum*.

Compromisso: Meta da Sprint.

A Meta da Sprint é o único objetivo da *Sprint*. Para os *Developers*. A Meta da Sprint cria coerência e foco, encorajando o *Scrum Team* a trabalhar juntos ao invés de em iniciativas separadas.

A Meta da Sprint é criada durante o evento *Sprint Planning* e então adicionada ao *Sprint Backlog*. Conforme os *Developers* trabalham durante a *Sprint*, eles mantêm a Meta da Sprint em mente. Se o trabalho acabar sendo diferente do que eles esperavam, eles negociam com o *Product Owner* o escopo do *Sprint Backlog* dentro da *Sprint* sem afetar a Meta da Sprint.

Increment (Incremento).

Um *increment* é um trampolim concreto em direção a Meta do produto. Cada incremento é adicionado a todos os incrementos anteriores e testados, garantindo que todos os incrementos funcionem juntos. A soma dos incrementos é apresentada na *Sprint Review*.

No entanto, um incremento pode ser entregue aos stakeholders antes do final da *Sprint*.

Compromisso: Definição de Pronto.

A Definição de Pronto é uma descrição formal do estado do Incremento quando ele atende às medidas de qualidade exigidas para o produto.

Quando um item do *Product Backlog* atende a Definição de Pronto, um incremento nasce.

Se um item do *Product Backlog* não atender à Definição de Pronto, ele não poderá ser liberado ou mesmo apresentado na *Sprint Review*. Em vez disso, ele retorna ao *Product Backlog* para consideração futura.

Se a Definição de Pronto para um incremento faz parte dos padrões da organização, todos os *Scrum Teams* devem segui-la como mínimo. Se não for um padrão organizacional, o *Scrum Team* deve criar uma Definição de Pronto apropriada para o produto.

Os *Developers* devem estar em conformidade com a Definição de Pronto. Se houver vários *Scrum Teams* trabalhando juntos em um produto, eles devem definir e cumprir mutuamente a mesma Definição de Pronto.

Uma palavra sobre uma história bem escrita: INVEST.¹⁴

Histórias dizem o que o usuário (*Persona*) precisa, para quê e por quê. “*Eu, Persona (...), preciso de (...), para(...), por quê*”. Se ela for bem escrita no *Product Backlog*, ficará mais compreensível para o *Scrum Team*, logo, as funcionalidades

¹⁴ Agradecimento a Guilherme Noguchi.

terão mais probabilidade de satisfazer as necessidades do demandante e os Increments terão mais valor.

Há 6 princípios para uma história bem escrita no Scrum (CAROLI, P.,2024):

- Independente: uma história de usuário deve ser desenvolvida independentemente de outras, para evitar atrasos ou travamentos.
- Negociável: as várias histórias devem ter tamanhos equivalentes para poderem ser trocadas entre si quando há espera por algum pré-requisito ou uma necessidade. Assim, é possível manter o desenvolvimento,
- Valiosa: a história deve gerar algum valor para o usuário final. Não se quebra uma história para facilitar o trabalho do desenvolvedor de sistema, mas sim para atender melhor o usuário. Exemplo: quebrar em back-end e front-end; “Eu, nutricionista, preciso de uma tela e um banco de dados para gravar um cardápio.” Está errado! A nutricionista precisa fazer um cardápio, o valor deve ser percebido pelo usuário.
- Estimável: o escopo da história deve permitir uma estimativa de tempo de desenvolvimento. Se for muito grande ou abstrata, dificulta a estimativa.
- Sob medida: o ideal é que a história caiba numa Sprint.
- Testável: depois de tudo pronto, é necessário testar, sejam testes automatizados e/ou manuais que validem que os critérios de aceite foram cumpridos com sucesso.

Os seis princípios em inglês formam o acrônimo INVEST. Então, invista em histórias bem escritas.

6. Conclusão.

Metodologia ágil com framework SCRUM.

A metodologia ágil está em contínua evolução na DISIS com a adoção do framework SCRUM e a interação com o Ateliê de Software. As lições aprendidas com o Ateliê anterior foram fundamentais para descobrir pontos de atenção na gestão de projetos de sistemas digitais. Diretrizes: transparência, inspeção e adaptação.

Assim, o produto deste TCR não se trata de um manual rígido, mas sim um manual de diretrizes, dentro do framework Scrum, em que se salienta os princípios ágeis baseados em 4 pilares: colaboração com o cliente, software funcionando, indivíduos e interações, e resposta a mudanças.

Resultados positivos.

O SCRUM é eficaz na transformação da cultura tradicional para ágil e na melhoria da produtividade.

A equipe DISIS adotou as reuniões Scrum para seu dia a dia, isso melhorou a comunicação e a agilidade em resolver impedimentos, pois todos sabem o que está acontecendo em cada projeto em andamento, apesar de cada um ser responsável por um conjunto de sistemas da SME.

Como piloto, de 26/03/25 a 21/04/25, o final deste TCR foi desenvolvido nos moldes do Scrum: sprints de 7 dias, reuniões Daily informais de 5 minutos, Review e Planning e Retrospective às quartas-feiras em sequência, incrementos entregues em pasta compartilhada com a Orientadora assim que prontos, avisos por e-mail, por exemplo, gravações de entrevistas, diagramas no Bizagi, correções, melhorias, resumo de entrevistas, resumo de leituras.

Caminho promissor.

O novo contrato de Ateliê de Software (2024) é mais robusto que o antecessor, corrigindo algumas falhas e incluindo mais controle de qualidade.

O atual Ateliê de Software irá comportar mais de 100 profissionais dedicados exclusivamente às demandas da SME, divididos em times especializados por área demandante, além de desenvolvedores multifuncionais (XFN) que poderão exercer diversos papéis entre vários times scrum.

Some-se a isso as várias iniciativas em andamento na SME, dentre as quais pode-se destacar a atualização digital do Manual de Procedimentos pela Assessoria de Articulação com as DREs¹⁵ e o mapeamento de processos conduzido pelo Escritório de Processos¹⁶. Ambas contribuirão para uma melhor definição das demandas de projetos de software.

Por fim, espera-se que, ao publicá-lo, este TCR induza a cultura ágil tanto na SME e COTIC quanto em outros órgãos públicos, aumentando a eficácia e eficiência nos projetos de software, com vistas ao melhor uso de recursos públicos e assim promover maior efetividade nas políticas públicas dependentes de tecnologia da informação.

¹⁵ Agradecimento à Ananda Grinkaut.

¹⁶ Agradecimento à Patrícia Ferreira Trindade.

7. Referências

AMARAL, T. M. **Programa de Treinamento em Gestão de Processos da SME-SP**. São Paulo, 2024.

BRASIL. **Guia prático de gestão de processos**, MGI, Brasília, 2024, 1ª ed. Disponível em: <https://www.gov.br/gestao/pt-br/aceso-a-informacao/estrategia-e-governanca/gestaodeprocessos/GuiaPrticodeGestodeProcessosv1maiode20241.pdf>

CAROLI, P. **Direto ao Ponto: criando produtos de forma enxuta**. São Paulo: Casa do Código, 2018.

CAROLI, P. **Histórias do usuário e a construção de produtos de sucesso**. São Paulo, 2024. Disponível em: [https://caroli.org/historias-do-usuario-e-a-construcao-de-produtos-de-sucesso/#:~:text=Em%20seu%20livro%20Extreme%20Programming,small%20\(pequena\)%20e%20test%C3%A1vel](https://caroli.org/historias-do-usuario-e-a-construcao-de-produtos-de-sucesso/#:~:text=Em%20seu%20livro%20Extreme%20Programming,small%20(pequena)%20e%20test%C3%A1vel)

de Almeida, R; Alonso, A.; Lima, M. **Série “Métodos de pesquisa em Ciências Sociais”**, por Sesc/Cebrap – bloco qualitativo. São Paulo, 2016. Disponível em: https://bibliotecavirtual.cebrap.org.br/arquivos/2016_E-BOOK%20Sesc-Cebrap_%20Metodos%20e%20tecnicas%20em%20CS%20-%20Bloco%20Qualitativo.pdf . Acesso em: 15 out. 2024.

REBELLO, B. H. **Análise dos modelos e contratos de desenvolvimento de software na administração direta do governo federal: a crise na produção de software e as alternativas à fábrica de software**. ENAP, 2023. Tese (Mestrado Profissional em Governança e Desenvolvimento) – Escola Nacional de Administração Pública. Brasília, 2023.

SÃO PAULO (Município). **Como formular e implementar iniciativas públicas a partir da experimentação: o caso Programa Linguagem Simples**. São Paulo, 2021. Disponível em: [Como formular e implementar iniciativas públicas a partir da experimentação? – CopiCola](#). Acesso em: 16 out. 2024.

SÃO PAULO (Município). **Como identificar e disseminar inovação a partir da gestão de conhecimento: O caso do CopiCola**. São Paulo, 2021. Disponível em: <https://copicola.prefeitura.sp.gov.br/como-identificar-e-disseminar-inovacao-a-partir-da-gestao-do-conhecimento/>. Acesso em: 16 out. 2024.

SÃO PAULO (Município). **Contratação de serviços de Manutenção, desenvolvimento e sustentação de Software com Práticas Ágeis: Estudo**

Técnico Preliminar 1/2024. Número do Processo SEI: 6016.2024/0007204-4. São Paulo, 2024.

SÃO PAULO (Município). **Contratação de serviços de Manutenção, desenvolvimento e sustentação de Software com Práticas Ágeis**: Termo de Referência 1/2024. Número do Processo SEI: 6016.2024/0007204-4. São Paulo, 2024.

SÃO PAULO (Município). **Unidades educacionais**. São Paulo, 2024. Disponível em: <https://sites.google.com/edu.sme.prefeitura.sp.gov.br/smecoticdie-dadosgerenciais/unidades-educacionais> . Acesso em: 30 dez. 2024.

SCHWABER, K.; SUTHERLAND, J. **Scrum Guide**. EUA, 2020. Disponível em: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>.

SUTHERLAND, J. J. **SCRUM**: guia prático; tradução de Nina Lua. Rio de Janeiro: Sextante, 2020.

SUTHERLAND, J. **SCRUM**: a arte de fazer o dobro do trabalho na metade do tempo; tradução de Nathalie Gerhardt. São Paulo: Leya, 2014.

UFABC. **Guia de normalização de trabalhos acadêmicos**: ABNT, VANCOUVER, IEEE. Santo André, SP; São Bernardo do Campo, SP: Sistema de Bibliotecas da UFABC, 2021. Disponível em: [Guia de Normalizacao.pdf \(ufabc.edu.br\)](#)

_____. SGANDERLA, K. **Blog da IProcess**. Disponível em: [BPMN: Uma atividade para mais de um participante do processo – Blog da iProcess](#)

Manual de Desenvolvimento Ágil de Sistemas com SCRUM

Versão: 2.0 em 21/04/2025

Autora: Luci LumikoTani

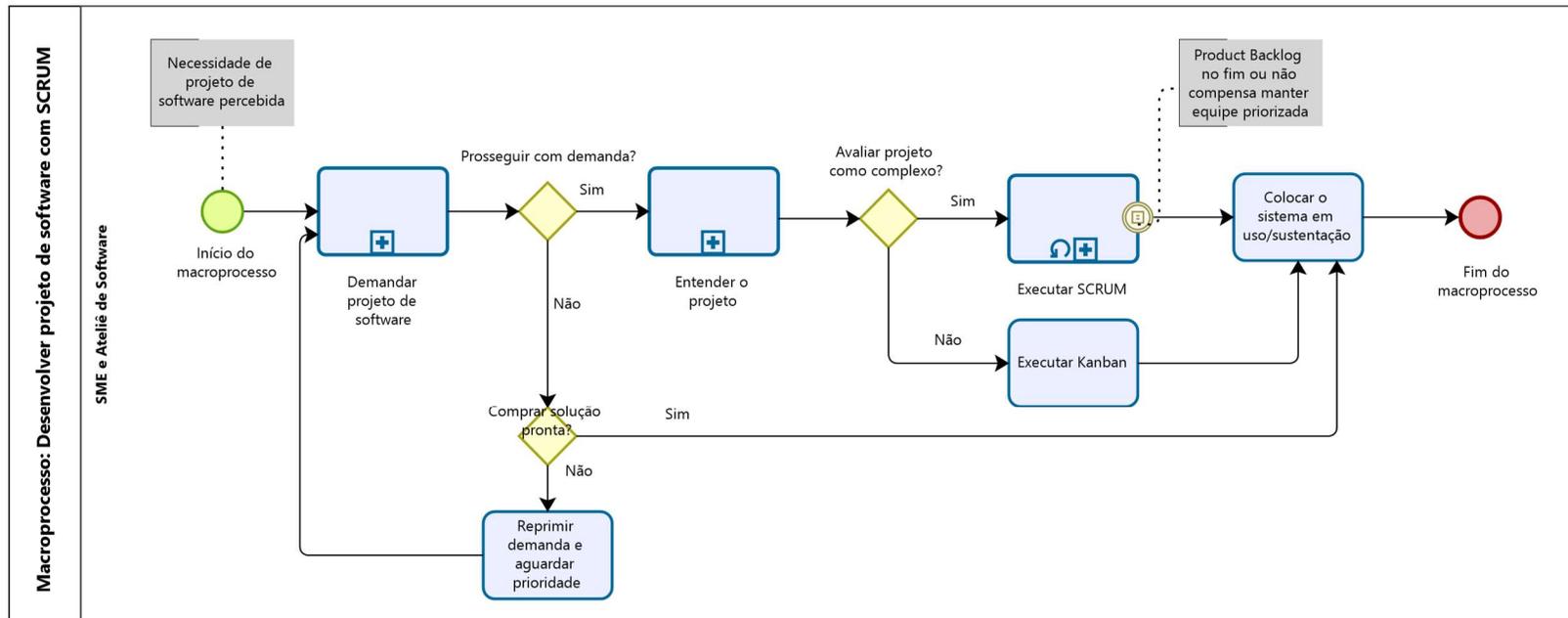
Bizagi Modeler

Índice

Manual de Desenvolvimento Ágil de Sistemas com SCRUM	22
Bizagi Modeler	22
1 1.0.0 MACROPROCESSO DESENVOLVER projeto de software com SCRUM	25
1.1 Macroprocesso: Desenvolver projeto de software com SCRUM.....	26
2 1.1.0 Demandar Projeto de Software.....	32
2.1 Demandar Projeto de Software	33
3 1.1.1 Demandar projeto do Gabinete	36
3.1 Demandar projeto do Gabinete.....	37
4 1.1.2 Demandar projeto de Coordenadorias	38
4.1 Demandar projeto de Coordenadorias.....	39
5 1.2.0 Entender o Projeto.....	41
5.1 Entender o Projeto	41
6 1.2.1 Planejar o Projeto de SW	48
6.1 Planejar o Projeto de SW.....	49

7	1.2.2 Realizar Descoberta do Fluxo	51
	7.1 Realizar Descoberta do Fluxo.....	51
8	1.2.3 Desenhar a solução conceitual	55
	8.1 Desenhar a solução conceitual.....	55
9	1.2.4 Desenvolver protótipo UXUI	57
	9.1 Desenvolver protótipo UX/UI.....	57
10	1.3.0 Executar Scrum	62
	10.1 Executar Scrum	62
11	1.3.1 Desenvolver funcionalidades	68
	11.1 Desenvolver funcionalidades	69
12	1.3.2 Realizar teste piloto	73
	12.1 Realizar Teste piloto	74
13	1.3.2.1 Corrigir erros	77
	13.1 Corrigir erros / melhorar	78
14	Glossário	81

1 1.0.0 MACROPROCESSO DESENVOLVER PROJETO DE SOFTWARE COM SCRUM



Descrição

Macroprocesso é o Processo principal deste mapeamento: Desenvolver projeto de software com Scrum.

1.1 MACROPROCESSO: DESENVOLVER PROJETO DE SOFTWARE COM SCRUM

Descrição

Macroprocesso ou Processo principal: Desenvolver projeto de software com SCRUM. Software é o conjunto de instruções, dados ou programas que permitem que um computador funcione e execute tarefas. É a parte lógica do computador, que se opõe ao hardware, que é a parte física. O SCRUM é descrito no subprocesso "Executar SCRUM".

Executantes

SME/Coordenadorias, Ateliê de Software, SME/COTIC/DISIS, SME/Gabinete

1.1.1 ELEMENTOS DO PROCESSO

1.1.1.1 Início do macroprocesso

Descrição

O macroprocesso é acionado pela necessidade de se desenvolver um sistema (software) para um dado processo, seja ele finalístico, de gestão ou de suporte.

1.1.1.2 Demandar projeto de software

Descrição

Manual de Desenvolvimento Ágil de Sistemas com SCRUM

V2.0–21.04.25

As várias áreas de SME podem solicitar projetos de software para a COTIC/DISIS, inclusive ela própria.

Executantes

SME, SME/COTIC/DISIS, Ateliê de Software

Processo

[1.1.0 Demandar Projeto de Software - Demandar Projeto de Software](#)

1.1.1.3 Prosseguir com demanda?

Descrição

No Subprocesso "Demandar projeto de software" não se optou por solução de mercado e há recursos disponíveis no Ateliê de Software? Então, a demanda será atendida.

1.1.1.4 Entender o projeto

Descrição

Subprocesso em que haverá várias reuniões entre as partes interessadas para reunir elementos de planejamento do projeto.

Executantes

Stakeholders - Partes interessadas

Processo

[1.2.0 Entender o Projeto - Entender o Projeto](#)

1.1.1.5 Avaliar projeto como complexo?

Descrição

A complexidade do projeto depende de vários fatores como quantidade de recursos necessários para o projeto, o tempo de execução estimado, o número de usuários concomitantes, a segurança de dados, número de interfaces, volumetria de dados.

1.1.1.6 Executar SCRUM

Descrição

Desenvolver o sistema. O projeto é particionado em sprints de tempo fixo, geralmente 15 dias corridos, em que um Sprint Backlog é selecionado para ser trabalhado a partir de um Product Backlog. Cada sprint resulta numa entrega. Algumas entregas formam uma parte do sistema que pode ser testada e usada. A solução é colocada em uso. Fim do macroprocesso.

Executantes

Ateliê de Software, PO - Product Owner, SME/COTIC/DISIS

Tipo de loop

Padrão

Condição de ciclo

Product Backlog no fim ou não compensa manter equipe priorizada

Máximo ciclo

0

Tempo de teste

Depois

Processo

1.3.0 Executar Scrum - Executar Scrum

1.1.1.7 Executar Kanban

Descrição

Processo não mapeado porque foge ao escopo deste trabalho. Kanban é outro framework da metodologia ágil, mais simples que o Scrum, baseado em Lean (produção enxuta). As funcionalidades do sistema são desenvolvidas em 3 estados: a fazer, fazendo e feito. Também engloba testes e correções de erros e melhorias. O produto é colocado em uso. Fim do macroprocesso.

Executantes

Ateliê de Software, PO - Product Owner, SME/COTIC/DISIS

1.1.1.8 Comprar solução pronta?

Descrição

Caso haja solução pronta no mercado, optou-se pela sua aquisição. Mas se não, caso não haja recursos para priorizar a demanda no Ateliê de Software, então a área demandante terá que aguardar ou solicitar prioridade ao Gabinete, demandando o projeto de software novamente.

1.1.1.9 Reprimir demanda e aguardar prioridade

Descrição

A demanda de software foi reprimida por falta de recursos.

Executantes

Demandante

1.1.1.10  Colocar o sistema em uso/sustentação

Descrição

O sistema está entregue completo ou há apenas pequenos ajustes de melhoria a se fazer. Parte da equipe de desenvolvedores é liberada para outros projetos. Todos os usuários podem utilizar o software.

Executantes

SME/COTIC/DISIS

1.1.1.11  Fim do macroprocesso

Descrição

Uma demanda de projeto de software ou foi atendida pela COTIC/DISIS por aquisição de solução de mercado ou atendida pelo uso do Ateliê de Software ou foi reprimida por falta de recursos.

1.1.1.12  SME e Ateliê de Software

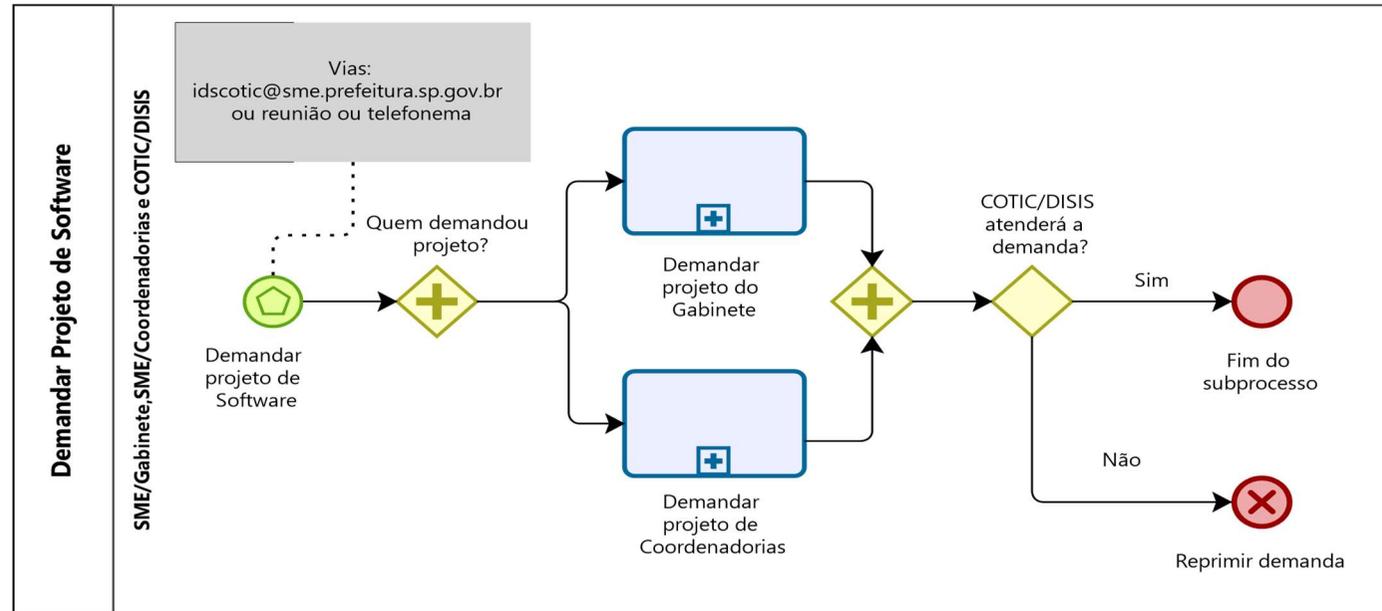
Descrição

Manual de Desenvolvimento Ágil de Sistemas com SCRUM

V2.0–21.04.25

Os atores do processo pertencem à Secretaria Municipal de Educação, com atuação da Divisão de Sistemas de SME/COTIC junto ao Ateliê de Software, contratada para desenvolver o projeto de software.

2 1.1.0 DEMANDAR PROJETO DE SOFTWARE



Descrição

Demandar Projeto de Software por e-mail, telefonema ou reuniões.

2.1 DEMANDAR PROJETO DE SOFTWARE

Descrição

Demandar Projeto de Software por e-mail, telefonema ou reuniões.

Executantes

SME/Coordenadorias, SME/Gabinete, SME/COTIC/DISIS

2.1.1 ELEMENTOS DO PROCESSO

2.1.1.1 Demandar projeto de Software

Descrição

Demandar projeto de software por diversas vias: idscotic@sme.prefeitura.sp.gov.br ou reunião ou telefonema.

2.1.1.2 Quem demandou projeto?

Descrição

Podem chegar diversas demandas de projeto de software ao mesmo tempo.

2.1.1.3 Demandar projeto do Gabinete

Descrição

Demandar projeto do Gabinete.

Executantes

SME/COTIC/DISIS, SME/Gabinete

Processo

[1.1.1 Demandar projeto do Gabinete - Demandar projeto do Gabinete](#)

2.1.1.4 Demandar projeto de Coordenadorias

Descrição

Demandar projeto de Coordenadorias, inclusive, da própria COTIC.

Executantes

SME/Coordenadorias, SME/COTIC/DISIS

Processo

[1.1.2 Demandar projeto de Coordenadorias - Demandar projeto de Coordenadorias](#)

2.1.1.5 Reprimir demanda

Manual de Desenvolvimento Ágil de Sistemas com SCRUM

V2.0–21.04.25

Descrição

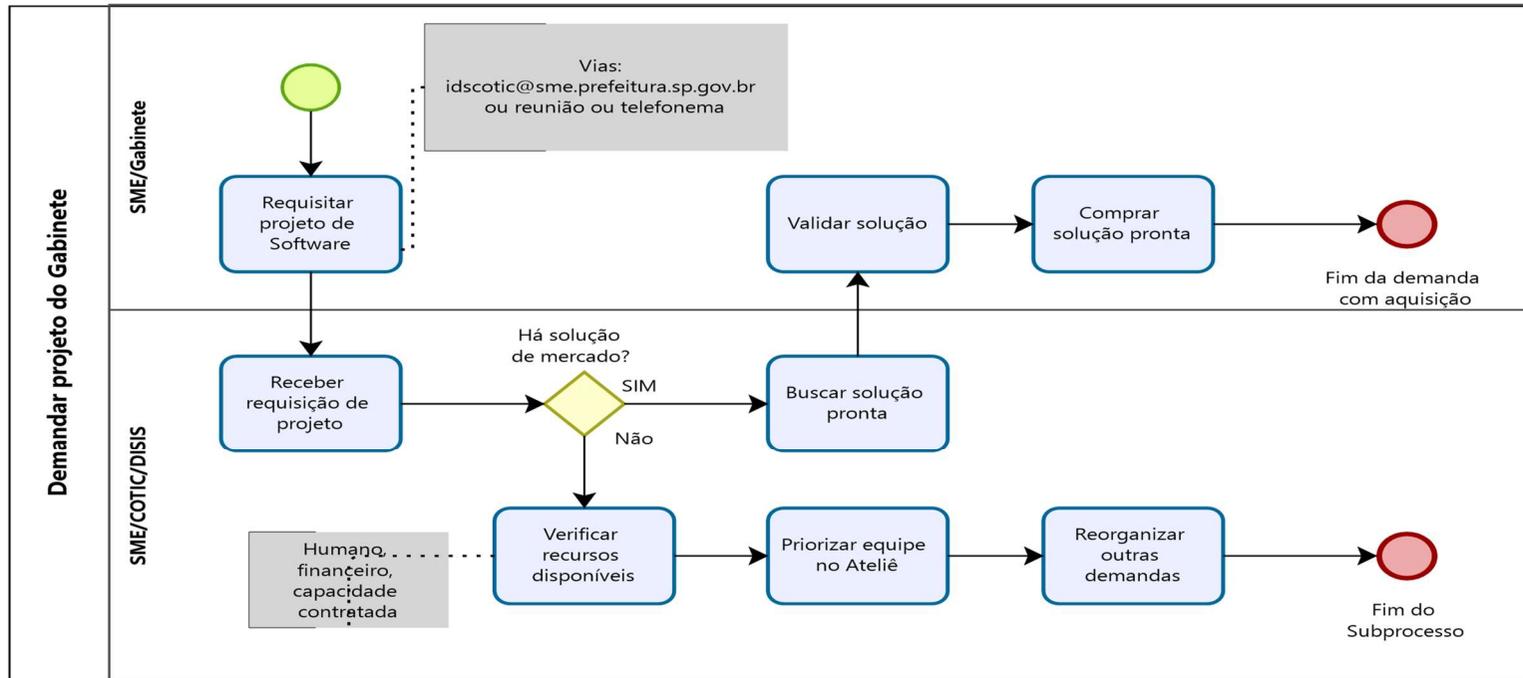
Não há recursos disponíveis.

2.1.1.6 SME/Gabinete, SME/Coordenadorias e COTIC/DISIS

Descrição

A demanda pode ter duas origens: Gabinete da Secretaria Municipal de Educação ou as diversas Coordenadorias dessa Secretaria, inclusive COTIC.

3 1.1.1 DEMANDAR PROJETO DO GABINETE



Descrição

Gabinete tem uma demanda própria ou analisou um projeto de software de uma Coordenadoria como prioritário.

3.1 DEMANDAR PROJETO DO GABINETE

Descrição

Gabinete tem uma demanda própria ou analisou um projeto de software de uma Coordenadoria como prioritário.

Executantes

SME/COTIC/DISIS, SME/Gabinete

3.1.1 ELEMENTOS DO PROCESSO

3.1.1.1 SME/Gabinete

Descrição

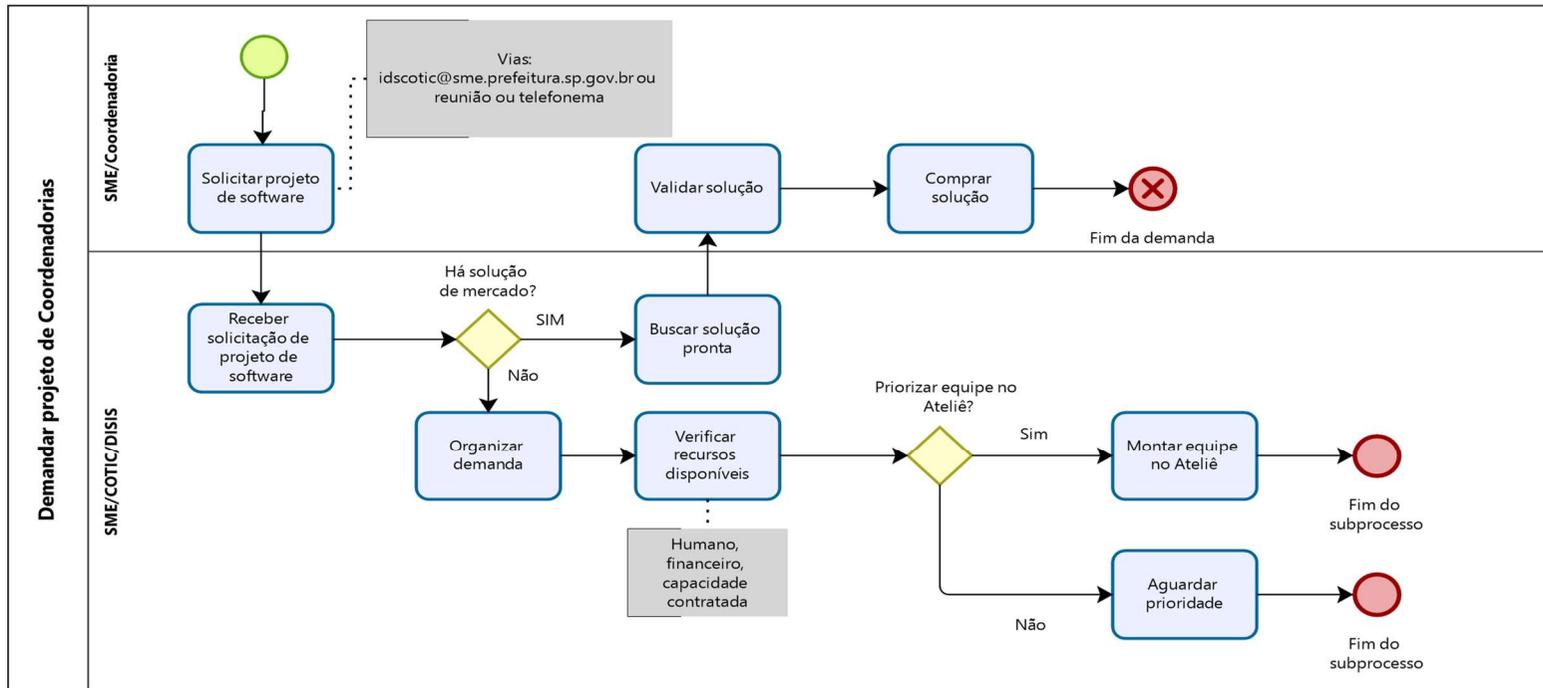
Gabinete da Secretaria Municipal de Educação.

3.1.1.2 SME/COTIC/DISIS

Descrição

Divisão de Sistemas da Coordenadoria de Tecnologia da Informação e Comunicação da Secretaria Municipal de Educação.

4 1.1.2 DEMANDAR PROJETO DE COORDENADORIAS



Descrição

COTIC/DISIS recebe demandas das Coordenadorias, inclusive dela própria.

4.1 DEMANDAR PROJETO DE COORDENADORIAS

Descrição

COTIC/DISIS recebe demandas das Coordenadorias, inclusive dela própria.

Executantes

SME/COTIC/DISIS, SME/Coordenadorias

4.1.1 ELEMENTOS DO PROCESSO

4.1.1.1 Solicitar projeto de software

Descrição

Coordenadoria solicita projeto vias: idscotic@sme.prefeitura.sp.gov.br ou reunião ou telefonema.

Executantes

SME/Coordenadorias

4.1.1.2 SME/COTIC/DISIS

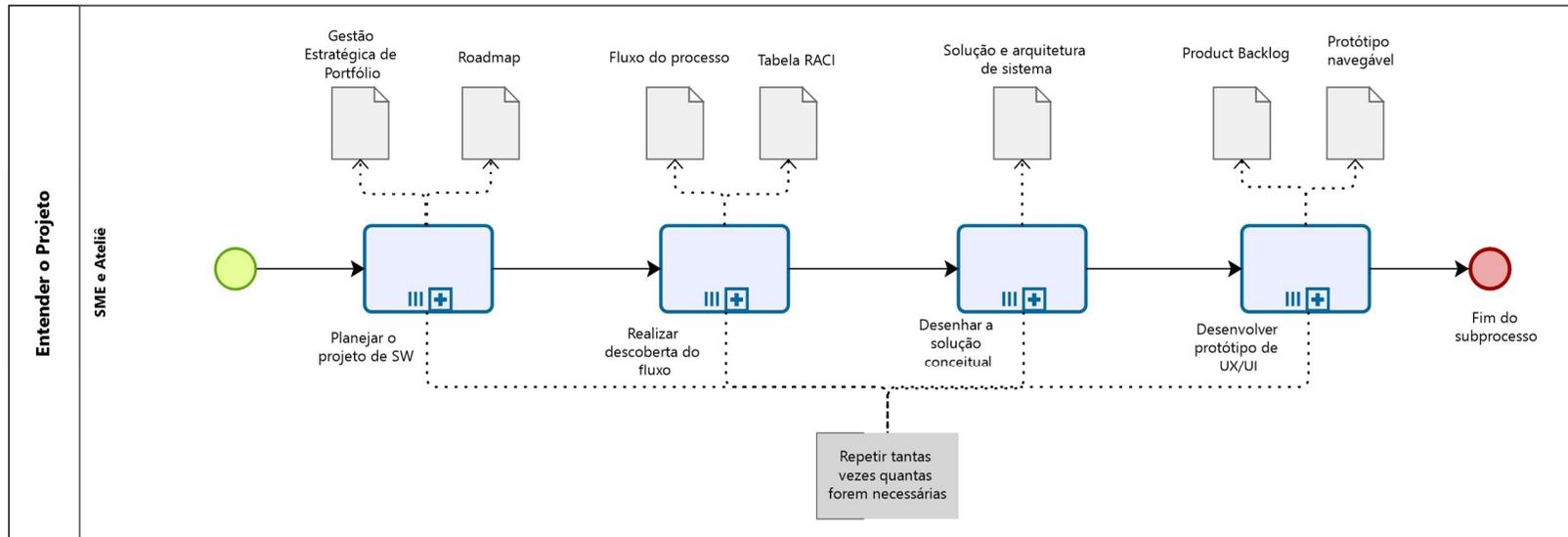
Descrição

Manual de Desenvolvimento Ágil de Sistemas com SCRUM

V2.0–21.04.25

Divisão de Sistemas da Coordenadoria de Tecnologia da Informação e Comunicação da Secretaria Municipal de Educação.

5 1.2.0 ENTENDER O PROJETO



5.1 ENTENDER O PROJETO

Descrição

Entender a ideia do projeto, o problema a resolver, a necessidade do cliente, o fluxo do processo envolvido, as responsabilidades, possíveis soluções e protótipos. São realizadas várias reuniões entre as partes interessadas até que o projeto esteja compreendido por todos.

Executantes

GP - Gerente de Portfólio, PO - Product Owner, SD - Service Designer, SME/COTIC/DISIS, UX/UI - User Experience/User interface

5.1.1 ELEMENTOS DO PROCESSO

5.1.1.1 Planejar o projeto de SW

Descrição

É composto por reuniões, virtuais ou presenciais, entre as partes interessadas no projeto. Objetivo é entender a demanda e traçar as linhas gerais do projeto. Analisar diretrizes, objetivos, metas e alinhamento institucionais.

Executantes

GP - Gerente de Portfólio, PO - Product Owner, SME/COTIC/DISIS, demandante

Tipo de loop

Múltiplas instâncias

Pedido MI

Paralelo

Condição de fluxo

Todos

Processo

[1.2.1 Planejar o Projeto de SW - Planejar o Projeto de SW](#)

5.1.1.2  Realizar descoberta do fluxo

Descrição

Reuniões, virtuais ou presenciais, para mapear o fluxo do processo que será transformado em sistema digital.

Executantes

PO - Product Owner, SD - Service Designer, UX/UI - User Experience/User interface, SM - Sorum Master, SME/COTIC/DISIS

Tipo de loop

Múltiplas instâncias

Pedido MI

Paralelo

Condição de fluxo

Todos

Processo

[1.2.2 Realizar Descoberta do Fluxo - Realizar Descoberta do Fluxo](#)

5.1.1.3  Desenhar a solução conceitual

Descrição

Desenhar a solução conceitual, arquitetura de sistema (aplicação, dados, integrações...).

Executantes

DEV Team, PO - Product Owner, SD - Service Designer, SM - Scrum Master, UX/UI - User Experience/User interface, Arquiteto, SME/COTIC/DISIS

Tipo de loop

Múltiplas instâncias

Pedido MI

Paralelo

Condição de fluxo

Todos

Processo

[1.2.3 Desenhar a solução conceitual - Desenhar a solução conceitual](#)

5.1.1.4  Desenvolver protótipo de UX/UI

Descrição

PO escreve as histórias de usuários com apoio do Ateliê de Software. O Objetivo é elaborar um protótipo navegável e testá-lo, e refinar o Product Backlog.

Manual de Desenvolvimento Ágil de Sistemas com SCRUM

V2.0–21.04.25

Executantes

PO - Product Owner, UX/UI - User Experience/User interface, SD - Service Designer, QA - Quality Assurance, DEV Team, SM - Scrum Master

Tipo de loop

Múltiplas instâncias

Pedido MI

Paralelo

Condição de fluxo

Todos

Processo

[1.2.4 Desenvolver protótipo UXUI - Desenvolver protótipo UX/UI](#)

5.1.1.5  [Gestão Estratégica de Portfólio](#)

Descrição

Detalhes do projeto, como nome, descrição, escopo, dor do cliente, proposta de valor, restrições, cronograma, riscos...

5.1.1.6  [Roadmap](#)

Descrição

Marcos globais de entregas do projeto.

5.1.1.7 Fluxo do processo

Descrição

Fluxo do processo ou Jornada do Cliente: o passo a passo do processo que será transformado em sistema digital.

5.1.1.8 Tabela RACI

Descrição

Matriz de responsabilidades RACI {Responsável, Autoridade, Colaborador e Informado} -> levantar as coisas que a gente faz-> detalhar -> categorizar -> definir perfis e responsabilidades.

5.1.1.9 Solução e arquitetura de sistema

Descrição

Desenho conceitual e arquitetura de sistema (aplicação, dados, integrações...).

5.1.1.10 Product Backlog

Descrição

Conjunto de histórias de usuários, cada história determina quem, o que, por quê. Assim, as histórias serão transformadas em funcionalidades de sistema. Simplificadamente, uma lista de pendências, já refinadas, histórias decompostas em partes menores, detalhadas.

5.1.1.11 Protótipo navegável

Descrição

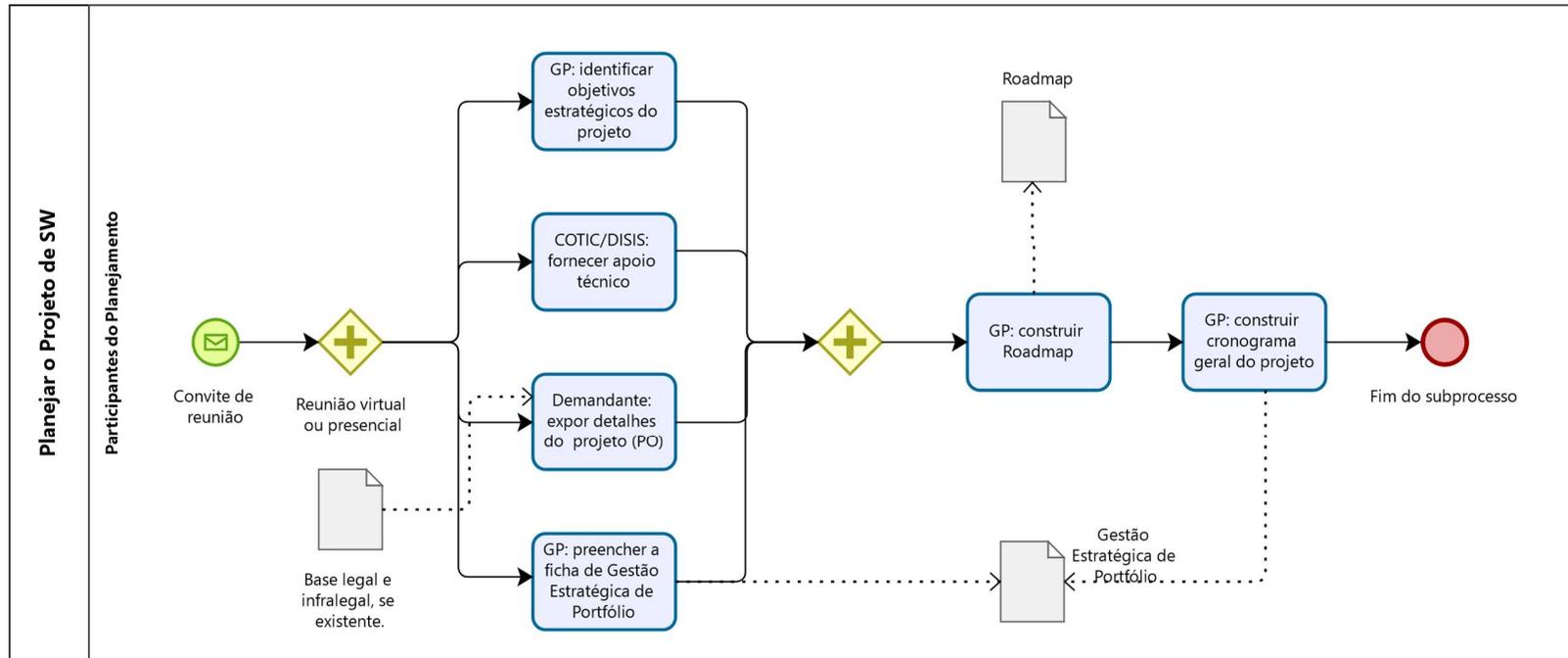
Uma parte visível e funcional em termos de navegação no sistema. O que acontece se clicar num botão, num link.

5.1.1.12 SME e Ateliê

Descrição

Os participantes de cada subprocesso dependem do artefato a ser produzido.

6 1.2.1 PLANEJAR O PROJETO DE SW



6.1 PLANEJAR O PROJETO DE SW

Descrição

É composto por reuniões, virtuais ou presenciais, entre as partes interessadas no projeto. Objetivo é entender a demanda e traçar as linhas gerais do projeto. Analisar diretrizes, objetivos, metas e alinhamento institucionais.

Executantes

Demandante, GP - Gerente de Portfólio, PO - Product Owner, SME/COTIC/DISIS

6.1.1 ELEMENTOS DO PROCESSO

6.1.1.1 GP: identificar objetivos estratégicos do projeto

Executantes

GP - Gerente de Portfólio

6.1.1.2 COTIC/DISIS: fornecer apoio técnico

Executantes

COTIC

6.1.1.3 Demandante: expor detalhes do projeto (PO)

Executantes

Manual de Desenvolvimento Ágil de Sistemas com SCRUM

V2.0–21.04.25

PO - Product Owner

6.1.1.4 GP: preencher a ficha de Gestão Estratégica de Portfólio

Executantes

GP - Gerente de Portfólio

6.1.1.5 GP: construir Roadmap

Executantes

GP - Gerente de Portfólio

6.1.1.6 GP: construir cronograma geral do projeto

Executantes

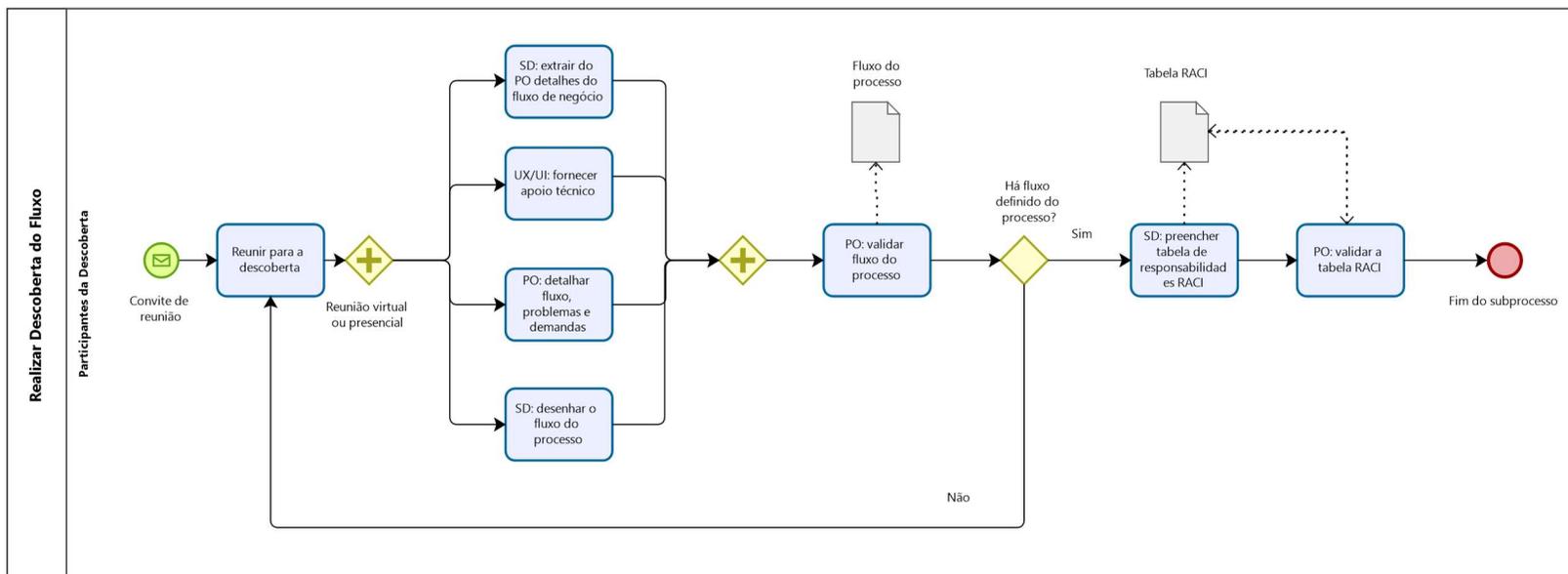
GP - Gerente de Portfólio

6.1.1.7 Participantes do Planejamento

Descrição

Representantes da Área demandante, inclusive o PO-Product Owner, gestão COTIC/DISIS, GP-Gerente de Portfólio do Ateliê de Software.

7 1.2.2 REALIZAR DESCOBERTA DO FLUXO



Powered by
bizagi
Modeler

7.1 REALIZAR DESCOBERTA DO FLUXO

Descrição

Reuniões, virtuais ou presenciais, para mapear o fluxo do processo que será transformado em sistema digital.

Executantes

PO - Product Owner, SD - Service Designer, UX/UI - User Experience/User interface, SM - Scrum Master, SME/COTIC/DISIS

7.1.1 ELEMENTOS DO PROCESSO

7.1.1.1 Reunir para a descoberta

Executantes

PO - Product Owner, SD - Service Designer, UX/UI - User Experience/User interface, SM - Scrum Master, SME/COTIC/DISIS

7.1.1.2 SD: extrair do PO detalhes do fluxo de negócio

Executantes

SD - Service Designer

7.1.1.3 UX/UI: fornecer apoio técnico

Executantes

UX/UI - User Experience/User interface

7.1.1.4 PO: detalhar fluxo, problemas e demandas

Executantes

PO - Product Owner

7.1.1.5 SD: desenhar o fluxo do processo

Executantes

SD - Service Designer

7.1.1.6 PO: validar fluxo do processo

Executantes

PO - Product Owner

7.1.1.7 SD: preencher tabela de responsabilidades RACI

Executantes

SD - Service Designer

7.1.1.8 PO: validar a tabela RACI

Executantes

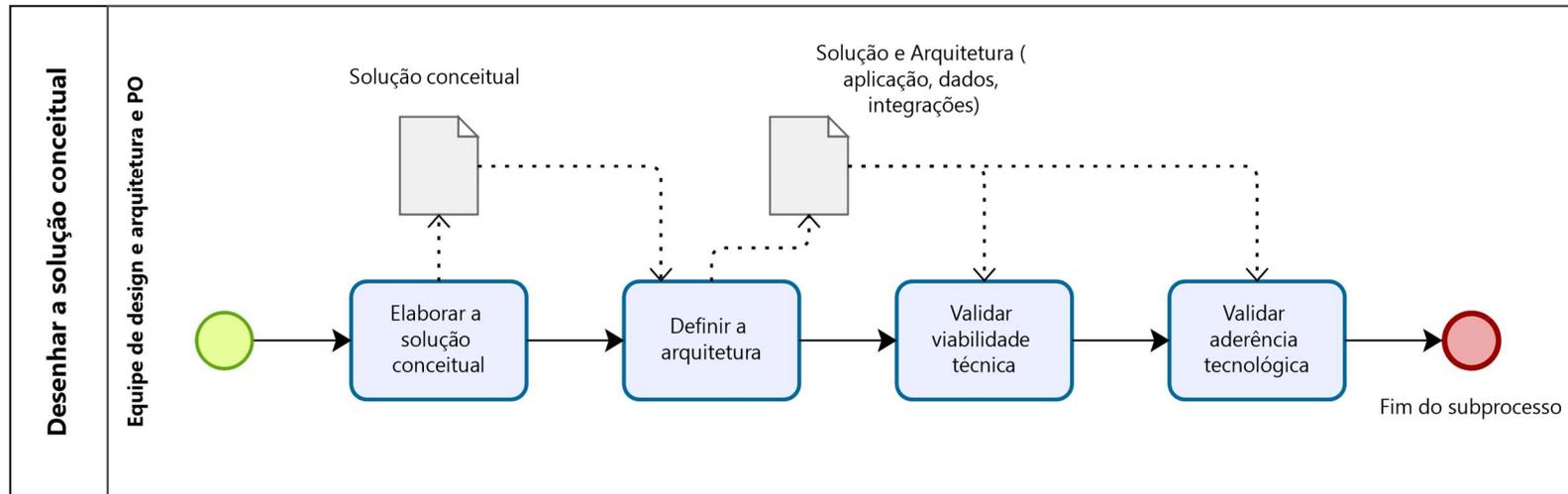
PO - Product Owner

7.1.1.9 Participantes da Descoberta

Descrição

PO Product Owner, SD Service Designer do Ateliê, UX/UI User Experience/User interface do Ateliê, Opcionais: COTIC/DISIS, GP Gerente de Portfólio do Ateliê de SW.

8 1.2.3 DESENHAR A SOLUÇÃO CONCEITUAL



8.1 DESENHAR A SOLUÇÃO CONCEITUAL

Descrição

Desenho da solução conceitual, arquitetura de sistema (aplicação, dados, integrações...).

Executantes

Arquiteto, DEV Team, PO - Product Owner, SD - Service Designer, SM - Scrum Master, SME/COTIC/DISIS, UX/UI - User Experience/User interface

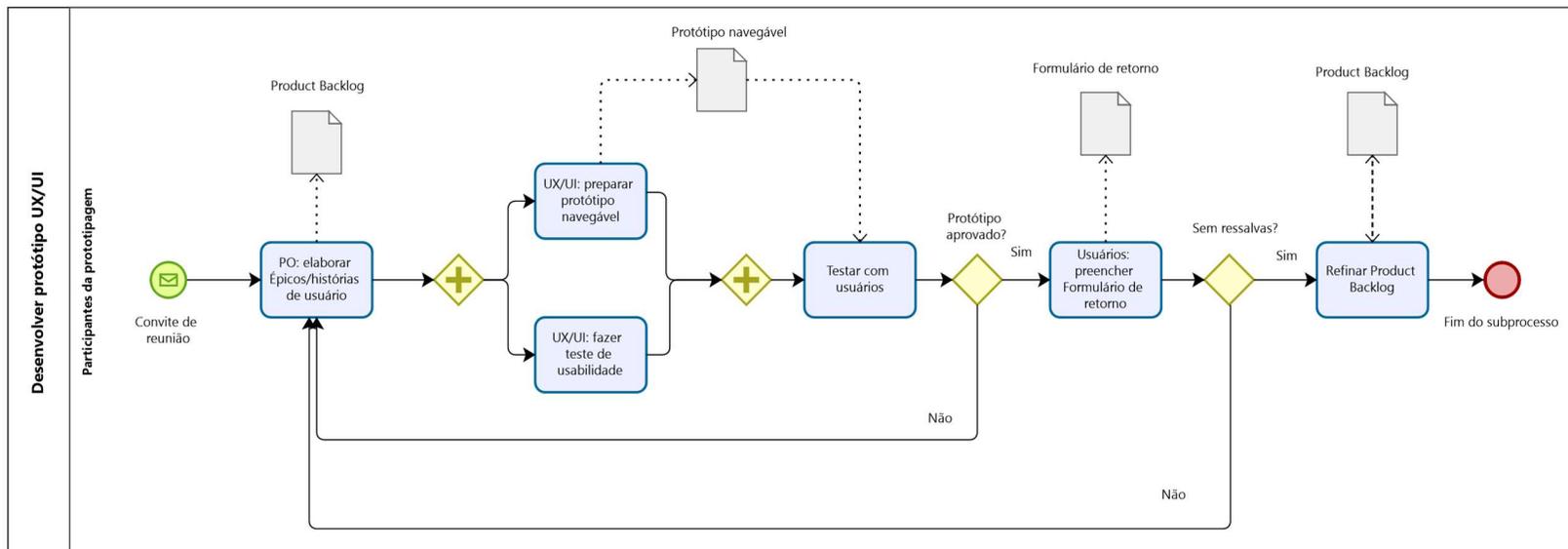
8.1.1 ELEMENTOS DO PROCESSO

8.1.1.1 Equipe de design e arquitetura e PO

Descrição

Arquiteto de soluções, Equipe de desenvolvedores, SM-Scrum Master, PO-Product Owner, COTIC/DISIS, UX/UI e Designer de serviço.

9 1.2.4 DESENVOLVER PROTÓTIPO UXUI



9.1 DESENVOLVER PROTÓTIPO UX/UI

Descrição

PO escreve as histórias de usuários com apoio do Ateliê de Software. O Objetivo é elaborar um protótipo navegável e testá-lo, e refinar o Product Backlog.

Executantes

PO - Product Owner, SD - Service Designer, SM - Scrum Master, UX/UI - User Experience/User interface, DEV Team, QA - Quality Assurance

9.1.1 ELEMENTOS DO PROCESSO

9.1.1.1 PO: elaborar Épicos/histórias de usuário

Executantes

PO - Product Owner, UX/UI - User Experience/User interface, SM - Scrum Master

9.1.1.2 UX/UI: preparar protótipo navegável

Executantes

UX/UI - User Experience/User interface

9.1.1.3 UX/UI: fazer teste de usabilidade

Executantes

UX/UI - User Experience/User interface

9.1.1.4 Testar com usuários

Descrição

Um subconjunto de usuários testará o protótipo.

Executantes

Usuários

9.1.1.5  Protótipo aprovado?

Descrição

Se aprovado, segue o processo. Se não aprovado, o protótipo será melhorado.

9.1.1.6 Usuários: preencher Formulário de retorno

Executantes

Usuários

9.1.1.7 Refinar Product Backlog

Descrição

As histórias do Product Backlog serão mais detalhadas.

Executantes

PO - Product Owner, SM - Scrum Master, UX/UI - User Experience/User interface

9.1.1.8  Product Backlog

Descrição

Lista de geral de histórias que embasarão as funcionalidades a serem desenvolvidas no projeto. Simplificadamente, uma lista de pendências.

9.1.1.9 Protótipo navegável

Descrição

Uma parte visível e funcional em termos de navegação no sistema. O que acontece se clicar num botão, num link.

9.1.1.10 Formulário de retorno

Descrição

Formulário contém a visão dos usuários sobre o protótipo navegável.

9.1.1.11 Product Backlog

Descrição

Lista geral de histórias que embasarão as funcionalidades a serem desenvolvidas no projeto. Simplificadamente, uma lista de pendências, já refinadas, histórias decompostas em partes menores, detalhadas.

9.1.1.12 Participantes da prototipagem

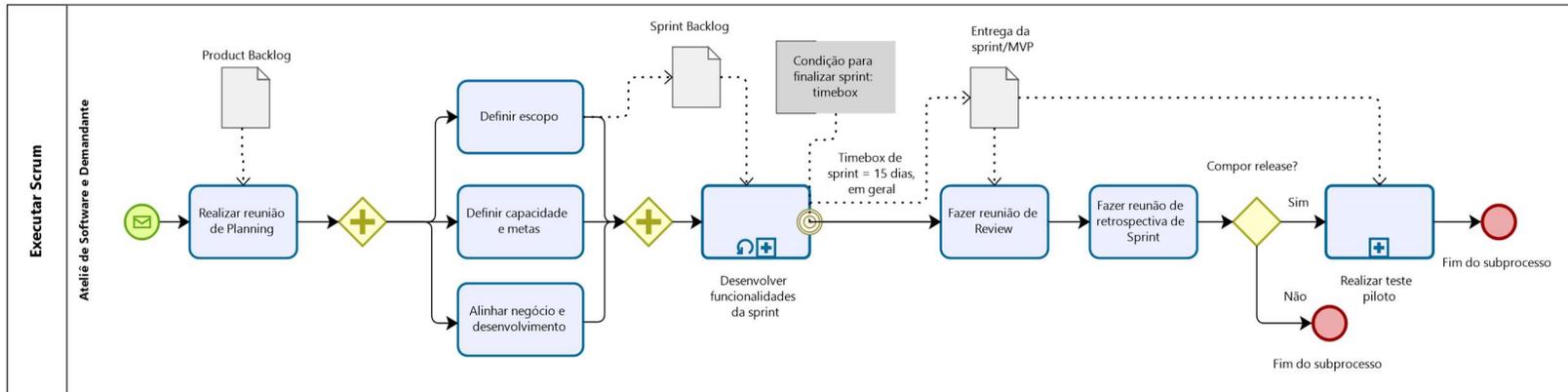
Manual de Desenvolvimento Ágil de Sistemas com SCRUM

V2.0–21.04.25

Descrição

PO, UX/UI, Designer de serviço, Equipe de desenvolvedores, Quality Assurance, Scrum Master. Para teste formal: cerca de 20 usuários.

10.1.3.0 EXECUTAR SCRUM



Powered by
bizagi
Modeler

Descrição

Desenvolver o sistema, é o cerne do Scrum.

10.1 EXECUTAR SCRUM

Descrição

Desenvolver o sistema, é o cerne do Scrum. O projeto é particionado em sprints de tempo fixo, geralmente 15 dias corridos, em que um Sprint Backlog é selecionado para ser trabalhado a partir de um Product Backlog. Cada sprint resulta numa entrega. Algumas entregas formam uma parte do sistema, release, onde um MVP - Produto Mínimo Viável pode ser testado e implantado.

Executantes

Ateliê de Software, PO - Product Owner, Usuários

10.1.1 ELEMENTOS DO PROCESSO

10.1.1.1 Realizar reunião de Planning

Descrição

Reunião de planejamento de sprint.

Executantes

PO - Product Owner, SD - Service Designer, SM - Scrum Master, DEV Team

10.1.1.2 Desenvolver funcionalidades da sprint

Descrição

Desenvolver funcionalidades a partir das histórias do sprint backlog. Ocorre diariamente durante uma sprint. Uma Daily é uma reunião curta entre o time de desenvolvimento e o Scrum Master, em que é revisto o que foi feito no dia anterior, o que deve ser feito no dia e se há algum impedimento para tal. O Scrum Master irá atuar para desfazer os impedimentos. O time de desenvolvimento produz as funcionalidades e o Quality Assurance faz os testes. No fim, acumulam-se as entregas da sprint.

Manual de Desenvolvimento Ágil de Sistemas com SCRUM

V2.0–21.04.25

Executantes

DEV Team, QA - Quality Assurance, SM - Scrum Master, Arquiteto, SD - Service Designer

Tipo de loop

Padrão

Máximo ciclo

0

Tempo de teste

Depois

Processo

[1.3.1 Desenvolver funcionalidades - Desenvolver funcionalidades](#)

10.1.1.3  Evento de temporizador

Ciclo

R1/P15D

10.1.1.4  Fazer reunião de Review

Descrição

Manual de Desenvolvimento Ágil de Sistemas com SCRUM

V2.0–21.04.25

Quando termina o timebox, é feita a entrega da sprint para o PO.

Executantes

SME/COTIC/DISIS, PO - Product Owner, DEV Team, SD - Service Designer, QA - Quality Assurance

10.1.1.5 Fazer reunião de retrospectiva de Sprint

Descrição

Debater lições aprendidas na sprint, busca de melhorias no Scrum, gestão do conhecimento.

Executantes

DEV Team, PO - Product Owner, QA - Quality Assurance, SD - Service Designer, SM - Scrum Master, SME/COTIC/DISIS, UX/UI - User Experience/User interface, GP - Gerente de Portfólio

10.1.1.6 Compor release?

Descrição

A entrega da sprint é um MVP, que já pode ser testado pelos usuários finais? Então, fazer teste piloto com um subconjunto de usuários.

10.1.1.7 Realizar teste piloto

Descrição

Um grupo reduzido de usuários reais irá testar o sistema entregue.

Executantes

PO - Product Owner, Usuários, SM - Scrum Master, QA - Quality Assurance, DEV Team

Processo

[1.3.2 Realizar teste piloto - Realizar Teste piloto](#)

10.1.1.8 Product Backlog

Descrição

Conjunto de histórias de usuários, cada história determina quem, o que, por quê. Assim, as histórias serão transformadas em funcionalidades de sistema. Simplificadamente, uma lista de pendências, já refinadas, histórias decompostas em partes menores, detalhadas.

10.1.1.9 Sprint Backlog

Descrição

Subconjunto do Product Backlog, cujas funcionalidades serão desenvolvidas nesta sprint.

10.1.1.10 Entrega da sprint/MVP

Descrição

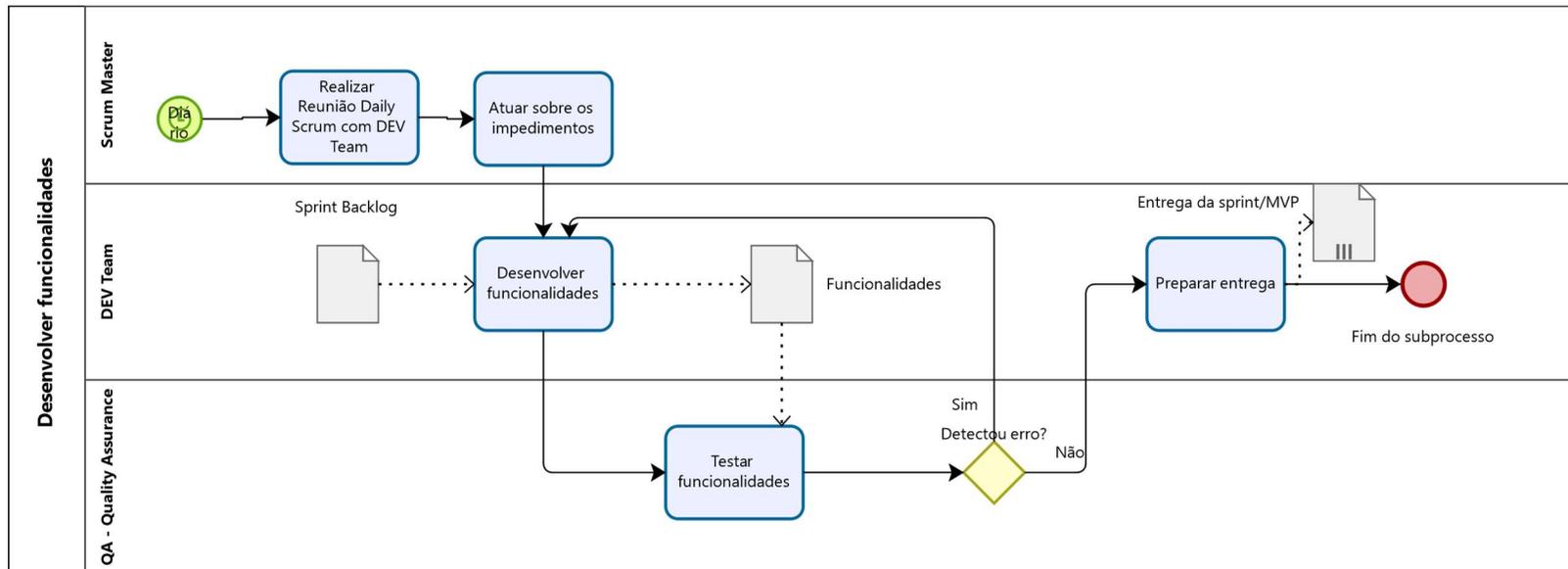
Entrega de um incremento funcional do projeto.

10.1.1.11 Ateliê de Software e Demandante

Descrição

Profissionais do Ateliê de Software; com acompanhamento do PO -Product Ownwer; e Usuários.

11 1.3.1 DESENVOLVER FUNCIONALIDADES



11.1 DESENVOLVER FUNCIONALIDADES

Descrição

Desenvolver funcionalidades a partir das histórias do sprint backlog. Ocorre diariamente durante uma sprint. Uma Daily é uma reunião curta entre o time de desenvolvimento e o Scrum Master, em que é revisto o que foi feito no dia anterior, o que deve ser feito no dia e se há algum impedimento para tal. O Scrum Master irá atuar para desfazer os impedimentos. O time de desenvolvimento produz as funcionalidades e o Quality Assurance faz os testes. No fim, acumulam-se as entregas da sprint.

Executantes

QA - Quality Assurance, SM - Scrum Master, DEV Team

11.1.1 ELEMENTOS DO PROCESSO

11.1.1.1 Diário

Descrição

Reunião Daily Sprint, diária.

11.1.1.2 Realizar Reunião Daily Scrum com DEV Team

Descrição

Reunião diária, em que Scrum Master e DEV Team verificam o que foi feito ontem, o que será feito hoje, e se há impedimentos ou atrasos.

Executantes

SM - Scrum Master, DEV Team

11.1.1.3 Atuar sobre os impedimentos

Descrição

Scrum Master deve agir para mitigar ou eliminar os impedimentos.

Executantes

SM - Scrum Master, COTIC, PO - Product Owner, SME/COTIC/DISIS, Usuários

11.1.1.4 Desenvolver funcionalidades

Descrição

A equipe de desenvolvimento irá elaborar as funcionalidades.

Executantes

DEV Team

11.1.1.5 Testar funcionalidades

Descrição

O perfil de garantia de qualidade faz testes para detectar erros nas funcionalidades.

Executantes

QA - Quality Assurance

11.1.1.6  Preparar entrega

Descrição

Preparar entrega (Deploy).

Executantes

DEV Team

11.1.1.7  Sprint Backlog

Descrição

Subconjunto do Product Backlog, cujas funcionalidades serão desenvolvidas nesta sprint.

11.1.1.8  Funcionalidades

Descrição

Entrega de um incremento funcional do projeto.

11.1.1.9 Entrega da sprint/MVP

Descrição

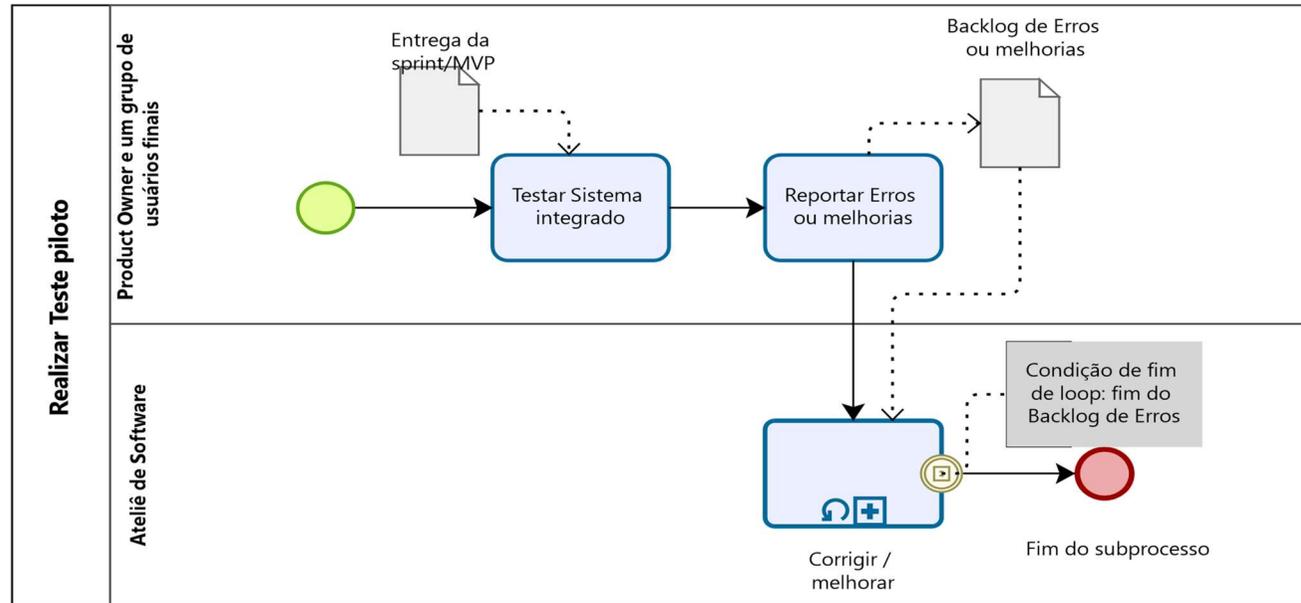
Um conjunto de funcionalidades desenvolvidas nas sprints. Na 1ª sprint, chama-se MVP - Produto Mínimo Viável, capaz de gerar valor para a área demandante. É uma versão enxuta do produto, que visa testar a viabilidade da ideia com mínimo de recursos e esforço. A cada sprint, o MVP irá agregando novas funcionalidades até o produto.

11.1.1.10 DEV Team

Descrição

Time de desenvolvimento

12.1.3.2 REALIZAR TESTE PILOTO



12.1 REALIZAR TESTE PILOTO

Descrição

Um grupo reduzido de usuários reais irá testar o sistema entregue.

Executantes

Usuários, PO - Product Owner, DEV Team, SM - Scrum Master

12.1.1 ELEMENTOS DO PROCESSO

12.1.1.1 Testar Sistema integrado

Executantes

Usuários

12.1.1.2 Reportar Erros ou melhorias

Descrição

Reportar Erros ou melhorias

Executantes

Usuários

12.1.1.3 Corrigir / melhorar

Descrição

Corrigir os erros detectados no teste piloto. Também podem ser melhorias.

Executantes

DEV Team, SM - Scrum Master

Tipo de loop

Padrão

Máximo ciclo

0

Tempo de teste

Depois

Processo

[1.3.2.1 Corrigir erros - Corrigir erros / melhorar](#)

12.1.1.4 Entrega da sprint/MVP

Descrição

Entrega de um incremento funcional do projeto.

12.1.1.5 Backlog de Erros ou melhorias

Descrição

PO e usuários preenchem lista de erros ou melhorias necessárias.

12.1.1.6 Product Owner e um grupo de usuários finais

Descrição

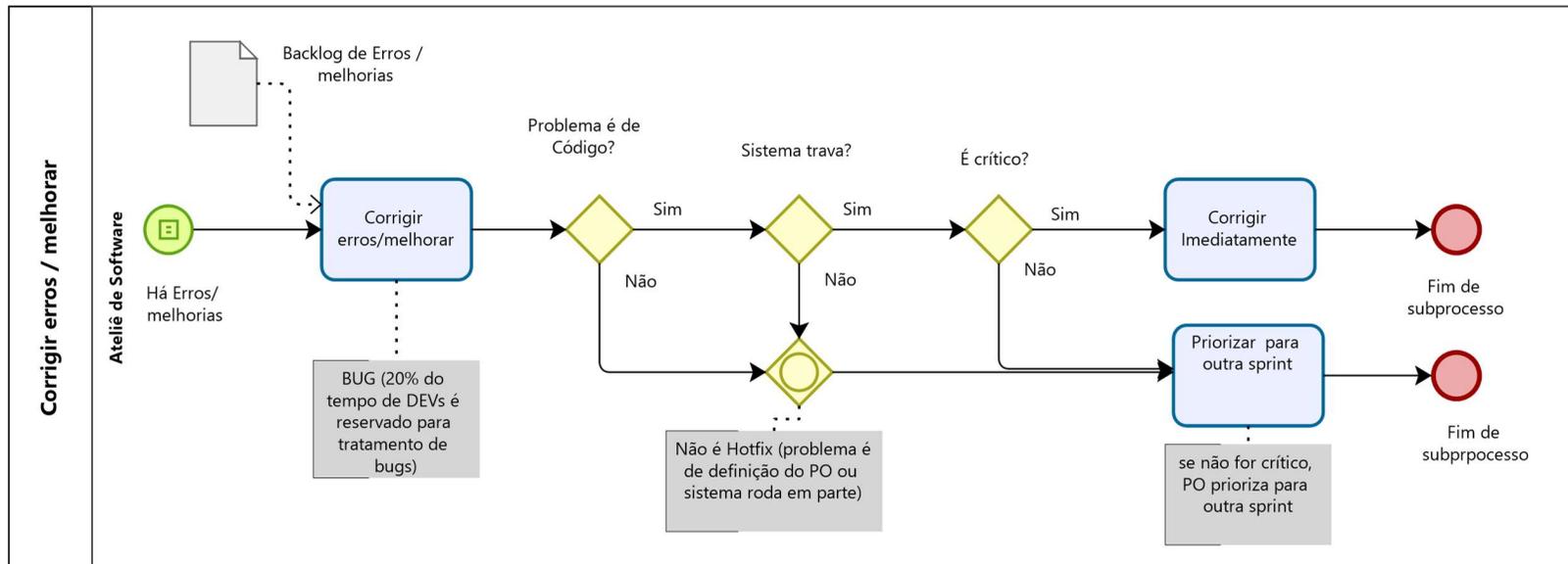
Product Owner e um grupo de usuários finais.

12.1.1.7 Ateliê de Software

Descrição

DEV Team, Scrum Master, QA.

13 1.3.2.1 CORRIGIR ERROS



Descrição

Corrigir os erros detectados no teste piloto. Também podem ser melhorias.

13.1 CORRIGIR ERROS / MELHORAR

Descrição

Corrigir os erros detectados no teste piloto. Também podem ser melhorias.

Executantes

Ateliê de Software

13.1.1 ELEMENTOS DO PROCESSO

13.1.1.1 Há Erros/ melhorias

Descrição

Backlog de erros/melhorias não está vazio.

Condição

Backlog de erros/melhorias não vazio?

13.1.1.2 Corrigir erros/melhorar

Descrição

O time de desenvolvedores irá corrigir e testar a parte entregue.

Manual de Desenvolvimento Ágil de Sistemas com SCRUM

V2.0–21.04.25

Executantes

DEV Team, SM - Scrum Master

13.1.1.3  Corrigir Imediatamente

Executantes

DEV Team, SM - Scrum Master

13.1.1.4  Priorizar para outra sprint

Executantes

PO - Product Owner, SM - Scrum Master

13.1.1.5  Backlog de Erros / melhorias

Descrição

PO e usuários preenchem lista de erros ou melhorias necessárias.

13.1.1.6  Ateliê de Software

Manual de Desenvolvimento Ágil de Sistemas com SCRUM

V2.0–21.04.25

Descrição

Scrum Master, equipe de desenvolvimento

14 GLOSSÁRIO

Recurso	Descrição	Função
Arquiteto	O arquiteto é o responsável por garantir que o sistema seja construído de forma robusta, eficiente e escalável, atendendo às necessidades do negócio e do cliente. Eles são um membro essencial da equipe Scrum, contribuindo para o sucesso do projeto.	Papel
Ateliê de Software	Empresa contratada para desenvolver soluções de software sob medida para a SME.	Entidade
COTIC	Coordenadoria de Tecnologia da Informação e Comunicação.	Entidade
Demandante	Área ou pessoa que solicita um projeto de desenvolvimento de sistemas/software.	Papel
DEV Team	DEV Team - Equipe de Desenvolvimento, composto por programadores, analistas, gestores de banco de dados, testadores, arquitetos, designers. É o grupo de pessoas responsável por construir e entregar o produto potencialmente utilizável a cada Sprint. Eles são multifuncionais, possuem as habilidades necessárias para desenvolver e entregar o software, e são auto-organizados, tomando decisões sobre como melhor realizar o trabalho.	Papel
GP - Gerente de Portfólio	O gerente de portfólio, também conhecido como "Portfolio Manager" em inglês, tem o papel de garantir que todos os projetos do portfólio estejam alinhados com a estratégia da empresa e que a alocação de recursos seja otimizada. Ele acompanha o	Papel

	desempenho de cada projeto, utiliza métricas para análise e toma decisões estratégicas sobre o portfólio como um todo, mantendo uma comunicação clara com as partes interessadas.	
PO - Product Owner	Dono do produto, responsável pela gestão do projeto na área demandante. Seu papel é fundamental para escrever as histórias que irão se converter nas funcionalidades de sistemas. Então, deve ser conhecedor das regras de negócio do processo a ser transformado em sistema. Também é o responsável pela priorização das histórias do Product Backlog para o Sprint Backlog. Contudo, seu papel mais importante talvez seja a de testar as entregas para aceitá-las ou pedir correções ou melhorias a cada sprint ou a cada teste piloto. Assim, Product Owner (PO) é o responsável por maximizar o valor do produto resultado do trabalho da equipe de desenvolvimento. Isso significa que o PO define a visão do produto, prioriza as necessidades dos usuários e stakeholders, e garante que o Product Backlog (lista de pendências) esteja sempre claro e transparente.	Papel
QA - Quality Assurance	No Scrum, o Quality Assurance (QA), ou Garantia de Qualidade, é fundamental para assegurar que o produto atenda às expectativas e padrões de qualidade. Isso envolve garantir que os requisitos sejam cumpridos, os testes sejam realizados adequadamente e a qualidade seja incorporada em todo o processo de desenvolvimento.	Papel

SD - Service Designer	Um designer de serviço desempenha um papel crucial na criação e melhoria de serviços, desde a definição do problema até o teste do protótipo. O designer de serviço contribui com seu conhecimento e habilidades em Design Thinking, pesquisa de usuários e prototipagem, garantindo que a solução final atenda às necessidades do usuário e seja alinhada com os objetivos do negócio.	Papel
SM - Scrum Master	No Scrum, o Scrum Master é o facilitador que garante que a equipe siga a estrutura e as práticas do framework. Ele é responsável por ajudar o Product Owner a definir e comunicar o valor, apoiar a equipe de desenvolvimento a se auto-organizar e a remover bloqueios, e promover a colaboração entre a equipe e as partes interessadas.	Papel
SME	Secretaria Municipal de Educação.	Entidade
SME/Coordenadorias	As Coordenadorias da SME.	Entidade
SME/COTIC/DISIS	Divisão de Desenvolvimento de Sistemas da Coordenadoria de Tecnologia da Informação e Comunicação da Secretaria Municipal de Educação.	Entidade
SME/Gabinete	Secretário(a); Secretário(a) Adjunto(a), Secretário(a) Executivo(a) Pedagógico(a); Chefe de Gabinete; Chefe de Assessoria de Articulação das DREs.	Entidade
Stakeholders - Partes interessadas	Stakeholders - como clientes, usuários finais, patrocinador do projeto, desempenham um papel importante ao participar da reunião de revisão de sprint e fornecer feedback sobre o trabalho concluído.	Papel

Usuários	Em sistemas de TI, os usuários finais são as pessoas que interagem diretamente com o sistema, utilizando-o para realizar suas tarefas e alcançar seus objetivos. Eles são os consumidores finais do produto ou serviço e não estão envolvidos na sua produção, desenvolvimento, manutenção ou suporte.	Papel
UX/UI - User Experience/User interface	O UX Designer preocupa-se com a funcionalidade, a usabilidade e a experiência geral do usuário, enquanto o UI Designer cuida da aparência visual e da forma como a interface é apresentada. Ambos são essenciais para criar produtos digitais que sejam não apenas funcionais, mas também agradáveis e intuitivos para o usuário.	Papel