

# T-AADSP - Uma ferramenta de apoio a implantação adaptativa de processo de software

Felipe Pereira Coelho  
Instituto Federal de Educação, Ciência e  
Tecnologia da Bahia  
Rua Emídio dos Santos, S/N, Barbalho  
Salvador, Bahia  
felipecoelho@ifba.edu.br

Antonio Carlos Santos Souza  
Instituto Federal de Educação, Ciência e  
Tecnologia da Bahia  
Rua Emídio dos Santos, S/N, Barbalho  
Salvador, Bahia  
antoniocarlos@ifba.edu.br

## RESUMO

As abordagens existentes para o gerenciamento de projetos de *software* não são fáceis de serem implementadas nas empresas. Neste projeto, será proposta uma ferramenta baseada na metodologia para gerenciamento de projetos de *software* AADSP, metodologia esta que utiliza conceitos das abordagens ágeis, preservando às práticas consideradas como relevantes ao cenário de desenvolvimento de *software* e as abordagens não ágeis que definem diversos processos e resultados esperados pelas mais diversas fases e atividade de um projeto.

## Palavras-chave

Engenharia de *Software*, Artefatos de *Software*, Gerenciamento de Projetos de *Software*

## 1. INTRODUÇÃO

As empresas em seus diversos portes oferecem a seus clientes produtos e/ou serviços, a qualidade dos produtos e/ou serviços oferecidos está diretamente ligada ao processo utilizado para conceber os mesmos. Um processo é uma sequência de passos realizados para um determinado propósito [9]. Um processo de *software* é definido como um conjunto de atividades, métodos, práticas e transformações que pessoas empregam para desenvolver e manter *software* e produtos associados [13]. O Guia Brasileiro de Melhoria de Processo de *Software* - MPSBR define que qualidade é um fator crítico para o sucesso da indústria de *software*. Para que se tenha um setor de *software* competitivo, nacional e internacionalmente, é essencial que os empreendedores do setor coloquem a eficiência e a eficácia dos seus processos em foco nas empresas, visando à oferta de produtos de *software* e serviços correlatos conforme padrões internacionais de qualidade [11]. Nesse contexto surgem diversas metodologias que visam promover a implantação e a melhoria contínua do processo de *software* nas empresas, neste trabalho...

## 2. FUNDAMENTAÇÃO TEÓRICA

Durante o processo de concepção do presente trabalho, foi necessária a realização de uma revisão da literatura a fim de levantar a fundamentação teórica ao tema proposto. Nesta seção, serão apresentados os principais temas encontrados e utilizados no desenvolvimento desta obra. Tomaremos como base o manifesto ágil de desenvolvimento de *software* [6] e alguns guias de melhoria de processo de *software*. Desse modo, será realizada a separação dos temas analisados em três categorias:

- Abordagens ágeis,
- Abordagens não ágeis, e
- Abordagens híbridas.

Para classificarmos as abordagens adotaremos o seguinte:

1. Neste trabalho, serão consideradas abordagens ágeis aquelas que estiverem de acordo com o manifesto ágil de *software*. Essa classificação será dada independentemente da classificação originalmente atribuída pelos autores de cada abordagem;
2. Para abordagens não ágil adotaremos esta classificação, a medida que, o autor focar seus processos em documentos e descartar ou minimizar os valores e princípios definidos no manifesto ágil de *software*;
3. Será considerada uma abordagem como híbrida, quando o próprio autor fizer esta afirmativa ou quando esta abordagem estiver relacionando seus processos e documentos aos princípios e valores do manifesto ágil de maneira valorativa aos documentos e processos.

### 2.1 Abordagens ágeis

#### 2.1.1 SCRUM

O *Scrum* é definido como uma metodologia ágil para gestão e planejamento de projetos de software [10].

No *Scrum*, os projetos são divididos em ciclos (tipicamente mensais) chamados de *Sprints*. O *Sprint* representa um *Time Box* dentro do qual um conjunto de atividades deve ser executado. Metodologias ágeis de desenvolvimento de

*software* são iterativas, ou seja, o trabalho é dividido em iterações, que são chamadas de *Sprints* no caso do *Scrum* [10].

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* 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* [10].

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 [10]. Ao final de um *Sprint*, a equipe apresenta as funcionalidades implementadas em uma *Sprint Review Meeting*. Finalmente, faz-se uma *Sprint Retrospective* e a equipe parte para o planejamento do próximo *Sprint* [10].

Papéis definidos no *Scrum*

- **Product Owner** O *Product Backlog* é uma lista contendo todas as funcionalidades desejadas para um produto. O conteúdo desta lista é definido pelo *Product Owner* [10].
- **Scrum Master** O *Scrum Master* procura assegurar que a equipe respeite e siga os valores e as práticas do *Scrum*. Ele também protege a equipe assegurando que ela não se comprometa excessivamente com relação àquilo que é capaz de realizar durante um *Sprint* [10].
- **Scrum Team** O *Scrum Team* é a equipe de desenvolvimento. Nela, não existe necessariamente uma divisão funcional através de papéis tradicionais, tais como programador, designer, analista de testes ou arquiteto. Todos no projeto trabalham juntos para completar o conjunto de trabalho com o qual se comprometeram conjuntamente para um *Sprint* [10].

### 2.1.2 XP

*Extreme Programming (XP)* é uma metodologia de desenvolvimento de *software*, nascida nos Estados Unidos ao final da década de 90. Vem fazendo sucesso em diversos países, por ajudar a criar sistemas de melhor qualidade, que são produzidos em menos tempo e de forma mais econômica que o habitual. Tais objetivos são alcançados através de um pequeno conjunto de valores, princípios e práticas, que diferem substancialmente da forma tradicional de se desenvolver *software* [4].

O XP divide os valores em:

- **Comunicação** Embora existam inúmeras formas de se comunicar ideias, algumas são mais eficazes que outras. Por exemplo, quando duas pessoas estabelecem um diálogo presencial, inúmeros elementos da comunicação colaboram para a compreensão do assunto, tais

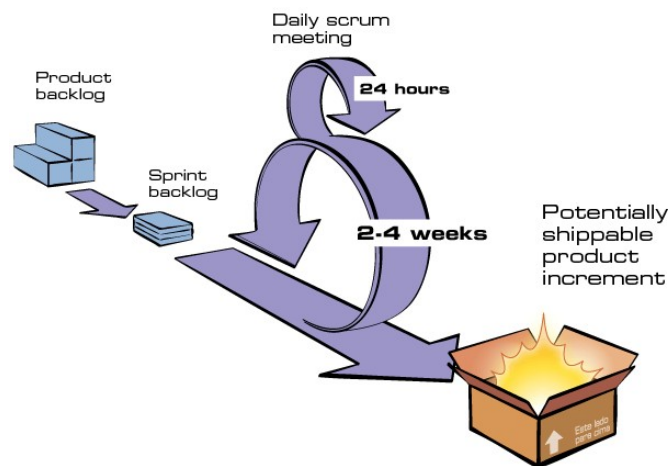


Figura 1: Ciclos do *Scrum* [10]

como gestos, expressões faciais, postura, palavras verbalizadas, tom de voz, emoções, entre outros. Quanto maior a capacidade de compreensão, maiores as chances de evitar problemas como ambiguidades, entendimento equivocados, entre outros [4].

- **Coragem** Equipes XP confiam na eficácia destas práticas e destes mecanismos de proteção e isso é o que as tornam receptivas a mudanças. Assim, ao invés de frear a criatividade do cliente e evitar mudanças, equipes XP as consideram inevitáveis e procuram se adaptar a elas com segurança e com coragem, isto é, com confiança em seus mecanismos de proteção, tais como desenvolvimento orientado a testes, programação em par e integração contínua [4].
- **Feedback** Projetos XP estabelecem formas de encurtar ao máximo a defasagem de tempo entre o momento em que uma ação é executada e o seu resultado é observado. Assim, por exemplo, desenvolvedores procuram entregar novas funcionalidades no menor prazo possível, para que o cliente compreenda rapidamente as consequências daquilo que pediu [4].
- **Respeito** Respeito é um valor que dá sustentação a todos os demais. Membros de uma equipe só irão se preocupar em comunicar-se melhor, por exemplo, se se importarem uns com os outros. Respeito é o mais básico de todos os valores. Se ele não existir em um projeto, não há nada que possa salvá-lo. Saber ouvir, saber compreender e respeitar o ponto de vista do outro é essencial para que um projeto de *software* seja bem sucedido [4].
- **Simplicidade** o conceito de simplicidade em inúmeros aspectos do projeto para assegurar que a equipe se concentre em fazer, primeiro, apenas aquilo que é claramente necessário e evite fazer o que poderia vir a ser necessário, mas ainda não se provou essencial [4].

## 2.2 Abordagens não Ágeis

### 2.2.1 Implantação e gerenciamento de processos com PMBOK

Nesta seção, iremos dar enfoque ao processo de gerenciamento de projetos definidos pelo Guia PMBOK. O Guia do Conhecimento em Gerenciamento de Projeto (Guia PMBOK) foi desenvolvido pelo PMI (*Project Management Institute*), sendo mantido pela mesma instituição desenvolvedora do Guia. O Guia PMBOK fornece diretrizes para o gerenciamento de projetos individuais e define os conceitos relacionados com o gerenciamento de projetos. Ele também descreve o ciclo de vida de gerenciamento e seus respectivos processos, assim como o ciclo de vida do projeto [7]. A partir deste momento, quando nos referirmos ao Guia do Conhecimento em Gerenciamento de Projeto adotaremos a conotação ao mesmo de Guia PMBOK.

Segundo o Guia PMBOK o gerenciamento de projetos é a aplicação do conhecimento, das habilidades, das ferramentas e das técnicas às atividades do projeto para atender aos seus requisitos. O gerenciamento de projetos é realizado através da aplicação e integração apropriadas dos quarenta e sete processos de gerenciamento de projetos, logicamente agrupados em cinco grupos de processos [7], conforme descritos a seguir:

- Iniciação,
- Planejamento,
- Execução,
- Monitoramento e controle, e
- Encerramento.

Para o Guia PMBOK, no processo de gerenciamento de um projeto normalmente se inclui, mas não se limita a identificação dos requisitos, abordagem das diferentes necessidades, preocupações e expectativas das partes interessadas no planejamento e execução do projeto, o estabelecimento e manutenção e execução de comunicações ativas, eficazes e colaborativas entre as partes interessadas, a realização do gerenciamento das partes interessadas visando o atendimento aos requisitos do projetos e a criação de suas entregas [7]. Devendo ser realizado também um equilíbrio das restrições conflitantes de um projeto que incluem:

- Escopo,
- Qualidade,
- Cronograma,
- Orçamento,
- Recursos, e
- Riscos.

O Guia PMBOK descreve que os fatores mencionados estão relacionados de tal forma que se algum deles mudar, pelo menos um outro fator será afetado [7]. Será apresentado a seguir o detalhamento das restrições conflitantes de um projeto, pelo fato destas serem consideradas um fator crucial ao processo de implantação e gerenciamento de um projeto de software.

## Escopo

Para planejar o gerenciamento do escopo deve ser confeccionado um plano onde deve ser documentado como tal escopo será definido, validando e controlando. O principal objetivo deste processo é o fornecimento de orientações e instruções sobre como o escopo será gerenciado ao longo de todo o projeto.

A figura abaixo ilustra o que deverá ser realizado:



**Figura 2: Fluxo do processo: Entradas, ferramentas, técnicas e saídas [2]**

Nesta etapa são definidos alguns artefatos do projeto como plano de gerenciamento do projeto, o termo de abertura do projeto, discriminação dos fatores ambientais da empresa que consiste em:

- Cultura Organizacional,
- Infraestrutura,
- Administração do Pessoal, e
- Condições de Mercado.

Além disso, o Guia PMBOK inclui nesta fase a definição de processos organizacionais onde deve ser incluso as:

- Políticas e procedimentos, e
- Informações históricas e base de conhecimento de lições aprendidas

No âmbito das ferramentas e técnicas, deverá ser consultada uma opinião especializada que consiste em entradas recebidas das partes entendidas e experientes [7]. O Guia PMBOK define que para as reuniões, a equipe do projeto poderá participar das mesmas e promover a definição do plano de gerenciamento do escopo do projeto.

No que se refere as saídas, deverá ser desenvolvido o plano de gerenciamento do escopo do projeto, este que é um componente do plano de gerenciamento do projeto ou do programa que descreve como escopo será definido, desenvolvido, monitorado, controlado e verificado [7].

Para o plano de gerenciamento dos requisitos é definido pelo Guia PMBOK, este primeiro que é um componente do plano de gerenciamento do projeto que descreve como os requisitos serão analisados, documentados e gerenciados.

## Qualidade

O gerenciamento da qualidade do projeto inclui os processos e as atividades da organização executora que determinam as políticas de qualidade, os objetivos e as responsabilidades, de modo que o projeto satisfaça às necessidades para as quais foi empreendido [7]. Os processos que incluem o gerenciamento da qualidade são:

- Planejar o gerenciamento da qualidade,
- Realizar a garantia da qualidade, e
- Controlar a qualidade.

O processo de planejamento do plano de gerenciamento da qualidade é o processo de identificar os requisitos e os padrões de qualidade do projeto, além de suas entregas através da documentação de como o projeto demonstrará a conformidade com os requisitos e/ou padrões de qualidade. Já a garantia da qualidade é o processo que auditoria os requisitos em sua qualidade e dos resultados das medições do controle de qualidade para garantir o uso dos padrões de qualidade e das definições operacionais apropriadas. O controle da qualidade é o processo de monitoramento e registro dos resultados da execução das atividades de qualidade para avaliar o desempenho e recomendar as mudanças necessárias [7].

## Cronograma

O gerenciamento do cronograma do projeto é o processo de estabelecer as políticas, os procedimentos e a documentação para o planejamento, desenvolvimento, gerenciamento, execução e controle do cronograma do projeto.

Nesta etapa deverá ser desenvolvido o plano de gerenciamento do cronograma do projeto. Para este plano deverá ser realizada a sequenciação das atividades do projeto, que é o processo de identificação e documentação dos relacionamentos entre as atividades do projeto. O principal objetivo é definir a sequência lógica do trabalho a fim de obter o mais alto nível de eficiência em face de todas as restrições do projeto [7].

## Orçamento

O Guia PMBOK define que para determinar o orçamento do projeto precisará ser realizado um cálculo estimado do custo de trabalho ou dos pacotes de trabalho a fim de estabelecer uma linha de base do custos autorizada. Para a definição da linha de base do escopo o Guia PMBOK discrimina que deverá ser realizado:

- Especificação do escopo do projeto,
- Estrutura analítica do projeto (EAP), e
- Dicionário da EAP.

Para confecção da EAP devem ser descritas todas as relações entre as entregas do projeto e seus vários componentes.

## Recursos

Nesta etapa o Guia PMBOK considera como recursos todos os itens discriminados na fase de orçamento, além desses são considerados também os recursos humanos do projeto, este último que inclui os processos que organiza gerenciam e guiam a equipe do projeto. A equipe do projeto consiste das pessoas com papéis e responsabilidades designadas para completar o projeto [7].

## Riscos

O gerenciamento dos riscos é o processo de definição de como conduzir as atividades de gerenciamento dos riscos de um projeto. O principal benefício deste processo é que ele garante que o grau, tipo e visibilidade do gerenciamento dos riscos sejam proporcionais tanto aos riscos quanto à importância do projeto para organização [7].

Estes serão os fundamentos bases, mas não únicos, considerados para esta obra na implantação e melhoria do processo de desenvolvimento de *software* definidos pelo Guia PMBOK.

### 2.2.2 Implantação e gerenciamento de processos com MPSBR

A SOFTEX desenvolveu um programa mobilizado e de longo prazo denominado MPS.BR. A sigla MPS.BR está associada ao programa MPS.BR, que é coordenado pela SOFTEX. A sigla MPS é uma marca genérica associada ao Modelo MPS, compreendendo tanto a sigla MPS-SW associada à Melhoria de Processo de *Software* quanto a sigla MPS-SV associada à Melhoria de Processo de Serviços [11], segundo a SOFTEX o programa foi criado em dezembro de 2003. Este programa é atualmente coordenado pela Associação para Promoção da Excelência do *Software* Brasileiro (SOFTEX), que conta com apoio do Ministério da Ciência, Tecnologia e Inovação (MCTI), Financiadora de Estudos e Projetos (FINEP), Serviço Brasileiro de Apoio às Micro e Pequenas Empresas (SEBRAE) e Banco Interamericano de Desenvolvimento (BID/FUMIN) [11].

O objetivo do programa MPS.BR é a melhoria de Processos de Software e Serviços por meio da implantação ou aprimoramento destes através da aplicação dos seus Guias, que contém a descrição geral do Modelo MPS e detalha o Modelo de Referência MPS para Software (MR-MPS-SW) e as definições comuns necessárias para seu entendimento e aplicação.

O Guia geral do MPSBR segmenta os processos em sete níveis que são discriminados a seguir:

- NÍVEL G – Parcialmente gerenciado;
- NÍVEL F – Gerenciado;
- NÍVEL E – Parcialmente definido;
- NÍVEL D – Largamente definido;
- NÍVEL C – Definido;
- NÍVEL B – Gerenciado quantitativamente;
- NÍVEL A – Em Otimização;

Cada nível é detalhadamente descrito em um guia específico para sua implementação, deste modo, a SOTEX oferece sete guias para implementação e melhoria contínua do processo de *software*. Neste trabalho, será dado enfoque ao guia de MPS de *Software* Nível G, este que contém orientações para a implementação do nível G do Modelo de Referência MR-MPSSW:2012 [11], pelo fato deste ser considerado o primeiro nível para implantação de um processo de *software* e um projeto de software voltado a produtos, além disso, este trabalho se propõe a implantação de um processo de *software* de forma adaptativa e alguns dos pontos propostos pela abordagem se fará presente em outras abordagens a serem apresentadas nesta obra.

### *Guia de implementação – Parte 1: Fundamentação para implementação do nível G do MR-MPS-SW:2012*

Segundo o Guia MPSBR, o nível G é o primeiro nível de maturidade do do MR-MPS-SW [11]. O Guia MPSBR descreve sua implementação deve ser executada com cautela por estabelecer o início dos trabalhos em implantação de melhoria dos processos de *software* na organização. Ao final da implantação deste nível a organização deve ser capaz de gerenciar parcialmente seus projetos de desenvolvimento de *software* [11].

O MPSBR define que no nível G os projetos usem os seus próprios padrões e procedimentos, não sendo necessário que se tenha padrões organizacionais. Se, porventura, a organização possuir processos já definidos e os projetos necessitarem adaptar os processos existentes, deve-se registrar essa adaptação durante o planejamento do projeto. Adaptações podem incluir alteração em processos, atividades, ferramentas, técnicas, procedimentos, padrões, medidas, dentre outras [11].

É necessário que as organizações que trabalham com a evolução de produtos adequem a forma que trabalham para se tornarem organizações orientadas a projetos. Ser orientada a projetos significa redefinir algumas operações (atividades de rotina) já em andamento, como projeto, estabelecendo objetivos, prazos e escopo para sua execução [11].

Para a Gerência de Projetos (GPR) são esperados os seguintes resultados:

- GPR1 - O escopo do trabalho para o projeto é definido;
- GPR2 - As tarefas e os produtos de trabalho do projeto são dimensionados utilizando métodos apropriados;
- GPR3 - O modelo e as fases do ciclo de vida do projeto são definidos;
- GPR4 - (Até o nível F) O esforço e o custo para a execução das tarefas e dos produtos de trabalho são estimados com base em dados históricos ou referências técnicas;
- GPR5 - O orçamento e o cronograma do projeto, incluindo a definição de marcos e pontos de controle, são estabelecidos e mantidos;

- GPR6 - Os riscos do projeto são identificados e o seu impacto, probabilidade de ocorrência e prioridade de tratamento são determinados e documentados;
- GPR7 - Os recursos humanos para o projeto são planejados considerando o perfil e o conhecimento necessários para executá-lo;
- GPR8 - (Até o nível F) Os recursos e o ambiente de trabalho necessários para executar o projeto são planejados;
- GPR9 - Os dados relevantes do projeto são identificados e planejados quanto à forma de coleta, armazenamento e distribuição. Um mecanismo é estabelecido para acessá-los, incluindo, se pertinente, questões de privacidade e segurança;
- GPR10 - Um plano geral para a execução do projeto é estabelecido com a integração de planos específicos;
- GPR11 - A viabilidade de atingir as metas do projeto é explicitamente avaliada considerando restrições e recursos disponíveis. Se necessário, ajustes são realizados;
- GPR13 - O escopo, as tarefas, as estimativas, o orçamento e o cronograma do projeto são monitorados em relação ao planejado;
- GPR14 - Os recursos materiais e humanos bem como os dados relevantes do projeto são monitorados em relação ao planejado;
- GPR15 - Os riscos são monitorados em relação ao planejado;
- GPR16 - O envolvimento das partes interessadas no projeto é planejado, monitorado e mantido;
- GPR17 - Revisões são realizadas em marcos do projeto e conforme estabelecido no planejamento;
- GPR18 - Registros de problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas, são estabelecidos e tratados com as partes interessadas;
- GPR19 - Ações para corrigir desvios em relação ao planejado e para prevenir a repetição dos problemas identificados são estabelecidas, implementadas e acompanhadas até a sua conclusão.

Para a Gerência de Requisitos (GRE) são esperados os seguintes resultados:

- GRE1 - O entendimento dos requisitos é obtido junto aos fornecedores de requisitos;
- GRE3 - A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida;
- GRE4 - Revisões em planos e produtos de trabalho do projeto são realizadas visando a identificar e corrigir inconsistências em relação aos requisitos;

- GRE5 - Mudanças nos requisitos são gerenciadas ao longo do projeto.

Por último é esperado aos atributos do de processo no nível G:

- AP 1 - O processo é executado;
- AP 2 - O processo é gerenciado.

Para todos os itens acima mencionados o MPSBR [11] descreve o mesmo detalhadamente em seu Guia de implementação do nível G, bem como, os resultados que são esperados a cada item. Os guias estão disponibilizados no site da SOF-TEX em Português do Brasil para *download*.

## 2.3 Abordagens Híbridas

### 2.3.1 Implantação e gerenciamento de processos com OpenUP

O *OpenUP* é caracterizado como um processo de desenvolvimento de *software* minimamente suficiente – isso significa que somente o conteúdo fundamental é incluído [3]. Dessa forma, o *OpenUP* não provê orientação em alguns tópicos que o projeto possa tratar, como equipes grandes, situações contratuais, aplicações de segurança ou missão crítica, orientação de tecnologia específica, etc. Contudo, *OpenUP* é completo no sentido que ele pode ser apresentado como um processo completo para construir um sistema. Para atender as necessidades que não são cobertas no seu conteúdo, *OpenUP* é extensível para ser usado como base no qual o conteúdo do processo pode ser adicionado ou adaptado como necessário [3]. O *OpenUP* foi criado pela Eclipse.org, e atualmente é mantido pela mesma organização criadora.

O *OpenUP* é definido como um processo ágil e unificado que contém um conjunto mínimo de práticas para ajudar times a serem mais eficientes no desenvolvimento de *software* [3], porém o modelo também preserva características essenciais do *Rational Unified Process* - RUP (ou Processo Unificado da Rational) [?]. O *OpenUP* engloba uma filosofia pragmática e ágil que foca na natureza colaborativa do desenvolvimento do *software*. Ele é um processo independente de ferramenta e de pouca cerimônia que pode ser usado como está ou ser estendido para atender uma ampla variedade de tipos de projetos [3].

A abordagem trabalha com quatro princípios. Estes princípios capturam a interação geral por trás dos processos e cria a fundação para interpretar papéis e produtos de trabalho e realizar tarefas:

- Colaboração para alinhar interesses e compartilhar entendimento,
- Balanceamento de prioridades concorrente para maximizar o valor do cliente,
- Foco antecipado na arquitetura para minimizar os riscos e organizar o desenvolvimento,
- Envolvimento para continuamente obter *feedback* e melhorar.

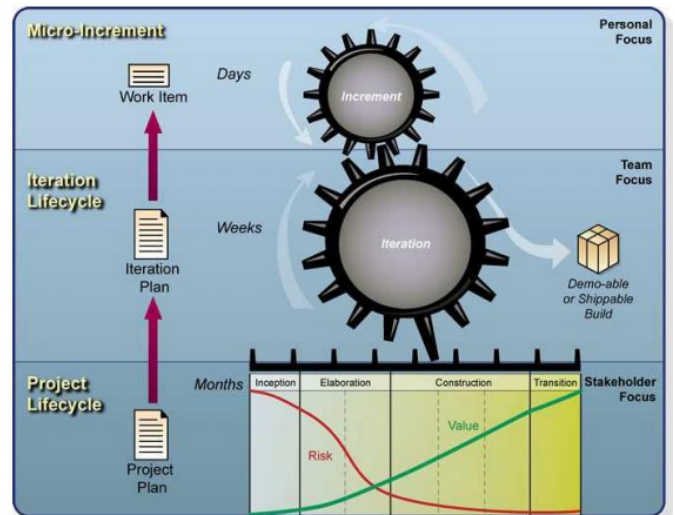
Cada princípio do *OpenUP* suporta uma das afirmações descritas no manifesto ágil de *software*:

Princípio do OpenUP	Afirmação do Manifesto Ágil
Colaborar para alinhar interesses e compartilhar entendimento	Indivíduos e interações sobre processos e ferramentas
Balanceamento de prioridades concorrentes para maximizar valor para o cliente	Colaboração com o cliente mais que negociação de contratos
Foco antecipado na arquitetura para minimizar os riscos e organizar o desenvolvimento	Software em funcionamento mais que documentação abrangente
Envolvimento contínuo para obter feedback e melhoria	Responder a mudanças mais que seguir um plano

**Figura 3: Mapeamento entre os princípios do *OpenUP* e o Manifesto Ágil [3]**

Organização do *OpenUP* é feita em duas dimensões diferentes e correlacionadas: conteúdo de métodos e conteúdo de processos. O conteúdo de método é onde os elementos de método (papéis, tarefas, artefatos e orientação) são definidos, independentemente de como eles são usados no ciclo de vida do projeto [3].

*OpenUP* estrutura o ciclo de vida do projeto em quatro fases: Início, Elaboração, Construção e Transição. O ciclo de vida do projeto proveem os stakeholders fiscalização, transparência e mecanismos de direção para controle dos fundos do projeto, escopo, exposição ao risco, valor fornecido e outros aspectos do processo [3].



**Figura 4: Organização do trabalho e foco do conteúdo no *OpenUP* [3]**

Papéis definidos pelo *OpenUP*:

- **Stakeholder** representa o interesse do grupo cujas necessidades devem ser satisfeitas pelo projeto. Esse é um papel que pode ser executado por qualquer um que é (ou potencialmente será) afetado materialmente pelo resultado do projeto,
- **Analista** representa o cliente e o usuário final no que diz respeito a obter a entrada dos *stakeholders* para entender o problema a ser resolvido e pela captura e configuração das prioridades para os requisitos.

- **Arquiteto** o é responsável por projetar a arquitetura do software, que inclui tomar as decisões técnicas chave que definem todo o design e implementação do projeto.
- **Desenvolvedor** é responsável por desenvolver uma parte do sistema, inclusive projetá-lo de acordo com a arquitetura, e então implementá-lo, executar testes unitários e integrar os componentes que fazem parte da solução.
- **Testador** é responsável pelas principais atividades de testes, como identificar, definir, implementar e conduzir os testes necessários, bem como registrar as saídas dos testes e analisar os resultados.
- **Gerente de projetos** lidera o planejamento do projeto em colaboração com os *stakeholders* e o time, coordenando interações com os *stakeholders*, e mantendo o time do projeto focado em atingir os objetivos do projeto.
- **Qualquer papel** representa alguém no time que pode realizar tarefas gerais.

### 2.3.2 Implantação e gerenciamento de processos com AADSP

O LABRASOFT, Grupo de Pesquisa Laboratório de Desenvolvimento de *Software*, propõe uma abordagem adaptativa para implantação do processo de *software* em empresas soteropolitanas desse porte, a *Adaptive Approach for Deployment of Software Process* (AADSP). Essa abordagem tem como alicerce práticas inovadoras em conformidade com o modelo MPS-BR, desenvolvido no Brasil pela SOFTEX (SOFTEX, 2012), práticas de métodos ágeis e o guia de conhecimento PMBOK da PMI (*Project Management Institute*) para gerenciamento de projetos, sendo adaptativa aos diversos perfis de MPEs [12].

#### Valoração dos artefatos

A AADSP qualifica os artefatos de acordo com seu grau de importância, desse modo esta abordagem busca implementar de forma adaptativa nas MPEs. Os três graus de importância são:

- **ESSENCIAL** Artefatos base para implementação do modelo AADSP, deste modo sua continuidade deverá ser garantida objetivando os resultados propostos pela abordagem [12].
- **IMPORTANTE** Artefatos que são consideráveis, todavia não são obrigatórios, deste modo sua implementação resultará em resultados adicionais ao modelo [12].
- **DESEJÁVEL** Artefatos pouco consideráveis, estes não implicam necessariamente na melhoria do processo ou em resultados satisfatórios [12].

#### Definição da valoração dos artefatos

A abordagem AADSP exige que todos os artefatos que afetam o trabalho estejam integrados. Esta abordagem explicita a dependência entre eles e a define no chamado fluxo de trabalho [12].

#### Confidencialidade e acesso aos artefatos do projeto de software

A confidencialidade de acesso aos artefatos, mesmo quando não declarada pelo cliente, são contempladas pela abordagem. É recomendável, portanto, explicitar a existência ou não de dados confidenciais e acessos a estes [12].

#### Análise da viabilidade do projeto e comprometimento da equipe executora

A abordagem AADSP exige que o gerente de projeto reavalie a viabilidade do projeto periodicamente ou caso mudança ou acréscimo no escopo, de requisito ou de qualquer dos recursos existentes do projeto [12].

#### Monitoramento contínuo dos artefatos e definição de marcos de trabalho do projeto de software

Os marcos de trabalho não dependem das cerimônias de revisão realizada dentro do projeto de *software*. A abordagem AADSP exige o acompanhamento contínuo do projeto comparando o planejado com o realizado. Além disso, deve ser definido no projeto os pontos dentro da escala de importância dos artefatos do projeto.

#### Gerências

- Gerência de Projetos;
- Gerência DE requisitos e modelagem
- Gerência configuração e mudanças;
- Gerência de colaboradores;
- Gerência de Testes.

Este trabalho será focado na gerência de Projetos. Deste modo devemos contemplar os seguintes artefatos definidos para esta gerência:

- Termo de abertura do projeto de *software* – TAP (essencial)
- Estrutura analítica do projeto de *software* – EAP (essencial)
- Documento de estimativa de escopo (essencial)
- Definição das funções da equipe executora do projeto (essencial)
- Cronograma de execução do projeto de *software* (essencial)
- Orçamento do Projeto (essencial)
- Recursos especiais (importante)
- Plano organizacional de dados ou Plano de gerenciamento de dados (importante)
- Plano de riscos do projeto (desejável)



## 2.4 Melhoria do processo de *software* x métodos ágeis

Nesta seção será discutido as abordagens para melhoria do processo de desenvolvimento de *software* em comparativo ao desenvolvimento ágil de *software*. Será considerado para isso a abordagem do MPS.BR desenvolvido pela Associação para Promoção da Excelência do *Software* Brasileiro (SOFTEX) e os modelos de desenvolvimento de *software* baseado no Manifesto Ágil para Desenvolvimento de *Software*. Um aspecto que diferencia os métodos ágeis de um modelo de referência, tais como MR-MPS e outras abordagens convencionais é a granularidade que algumas atividades são realizadas. Portanto, algumas atividades podem executar a mesma função nos dois modelos, mas com um nível diferente de profundidade. Assim, um elemento de fundamental característica técnica para avaliar a percentagem de adesão para os métodos ágeis modelos de maturidade é a realização de uma análise detalhada, considerando as características particulares de cada um deles [8].

Esta análise é um comparativo do modelo de adesão ao MR-MPS-SW:2012 Nível G, Arimoto [8] realizou o comparativo entre os modelos e tabulou os seguintes resultados conforme apresentados na tabela 1.

Arimoto [8] incluiu em seu quadro comparativo o *OpenUP*, que para este trabalho será qualificado enquanto abordagem híbrida para melhoria de processo de *software*, mediante ser fundamentada no *Rational Unified Process* - RUP (ou Processo Unificado da Rational).

## 3. TRABALHOS RELACIONADOS

Nesta seção foram selecionadas três ferramentas e sendo analisando o funcionamento das funcionalidades disponibilizadas aos usuários. São descritas as seguir as principais características das ferramentas analisadas:

- **Bitrix24** É uma plataforma completa de colaboração social, comunicação e ferramentas de gestão. Esta ferramenta trabalha integrada as redes social corporativa disseminando postagens e Comunicados, esta ferramenta trabalha com gerenciamento de Projetos / Grupos de Trabalho, Tarefas e Subtarefas, Modelos de tarefa, Tarefas recorrentes, Gerenciamento de registros (Listas), Diagrama de Gantt, Gerenciamento de tempo [1].
- **Jira** permite a gestão e a criação e delegação de projetos e tarefas, com fluxos de processos personalizados para o acompanhamento dos projetos, controle de acesso e segurança customizado, acesso via web com suporte aos principais browsers do mercado, notificações / alertas por email, centenas de *plug-ins* gratuitos. É uma ferramenta robusta para gestão de projetos e tarefas, bem como para acompanhamento e reporte de defeitos (*bugs*) em projetos de qualquer natureza [5].
- **Wrike** É uma ferramenta interativa para colaboração, planejamento de projetos, otimização de carga de trabalho e acompanhamento do andamento, ajudam a equipe a produzir mais e com maior velocidade, independente de onde se encontrem. Além disso, as

notificações em tempo real apresentam todas as atualizações diárias das atividades do projeto, eliminando a necessidade de reuniões extras e *e-mails*. Os painéis personalizáveis ajudam a monitorar projetos específicos, etapas importantes, trabalhos que você delegou, entre outros.

## 3.1 Comparativo entre ferramentas

A Tabela 2 exibe uma análise comparativa entre as ferramentas estudadas.

Conforme explanado na tabela 2, existem diversas soluções para gerenciamento de projetos. Este trabalho baseado no abordagem AADSP propõe um modelo alternativo para o gerenciamento de projetos de *software*. Destacam-se na ferramenta os recursos como mensuração do tamanho de *software* através da métrica de análise por ponto de função, controle do escopo do projeto através de diagrama da estrutura analítica do projeto e gerenciamento do Termo de Abertura de Projetos (TAP).

## 4. A FERRAMENTA T-AADSP

O T-AADSP é uma ferramenta para gerenciamento de projetos de *software*. Para construção deste sistema foi necessário seguir as etapas inerentes ao projeto de desenvolvimento de sistemas (Levantamento de Requisitos, Análise, Projeto, implementação e Testes) Serão apresentados a seguir as etapas realizadas para o desenvolvimento da ferramenta T-AADSP.

### 4.1 Requisitos do Sistema

Os requisitos funcionais do sistema serão apresentados a seguir, estes foram separados em: Requisitos Funcionais e Não Funcionais.

### 4.2 Requisitos do Funcionais

As principais funcionalidades da ferramenta serão descritas nas subseções seguintes:

#### 4.2.1 Autenticação

Para o usuário ter acesso de login este deve ter sido cadastrado previamente no sistema. Em caso de uma nova instalação do sistema por definição padrão o banco de dados virá configurado com um usuário administrativo com acesso a todas as páginas do sistema. Para acessar com este usuário deverá ser informado os seguintes dados de acesso login: adm e senha: aadsp (figura 5). Caso seja informado dados incorretos a ferramenta informará a mensagem de alerta "Não foi possível realizar a autenticação do usuário!" (figura 6), caso somente a senha esteja incorreta o sistema irá informar "Senha incorreta!" (figura 7)

#### 4.2.2 Solicitar nova senha

Este requisito possibilitará ao usuário solicitar uma nova senha via email, para isso o usuário devera fornecer o email dele e o sistema verificará a existência do email informado pelo usuário e enviará uma nova senha definida automaticamente pelo sistema (figura 8). Caso o sistema envie a nova senha ao email este exibirá a mensagem "senha enviada com sucesso" (figura 9), porem se este não conseguir enviar a nova senha via email o sistema exibirá a seguinte mensagem "Não foi possível realizar o envio de uma nova senha para este email, por favor tente novamente mais tarde!" (figura 10)



**Tabela 1: Análise do produto de trabalho MR-MPS e a abordagem híbrida *OpenUP* [8]**

Gerenciamento	Seção	MR-MPS	Abordagem Híbrida
Requerimentos	1	Critérios de Avaliação e aceitação de Requisitos / exigências de documentos	Discussões face-a-face / Sistema de Pesos para os requisitos / Modelo de Caso de Uso / <i>Check List</i> de tarefas
	2	Aprovação de Requisitos / Avaliação de Impacto	Discussões Face-a-face / <i>Check List</i> de Tarefas
	3	Sistema de Rastreamento de requisitos e de Trabalho Produtos / Matriz de rastreabilidade de requisitos e produtos de trabalho	–
	4	Resultados de revisões / Documentação das Inconsistências / Ações Corretivas	Registros no plano de interação / lista dos itens de trabalho
	5	Status dos requisitos / Repositório dos requisitos / Histórico de solicitações e decisões	Lista de itens de trabalho
Projetos	1	Estrutura Analítica do Projeto (EAP) ou documento de visão	Documento de Visão
	2	Estimativa de Tamanho	Estimativa ágil de Tamanho ( medido tipicamente usando uma unidade neutra tal como pontos )
	3	Fases e Ciclos do Projeto	Fases do Ciclo de Vida do Projeto (especificada no Plano de projeto)
	4	Estimativas esforço do projeto / As estimativas de custos do projeto	Normalmente usando as unidades de dias efetivos ou horas reais / Cálculo do custo aproximado por pessoa pelo tamanho de uma iteração
	5	Cronograma e Orçamento do projeto	Cronograma do Projeto e Orçamento (especificada no Plano de iteração e do Plano do Projeto)
	6	Plano de Gestão de Risco	Os riscos são identificados / Avaliação do impacto e da probabilidade de Ocorrência (especificado no Plano de Projeto ou Plano de Iteração) / Lista de Riscos
	7,8	Plano de Gestão de Recursos	Plano do Projeto / Plano de iterações / lista dos itens de trabalho
	9	Plano de Gestão de Dados	Plano de iteração
	10	Plano geral do projeto	Plano do Projeto
	11,12	Estudo de Viabilidade / Revisão e Compromisso com o Plano de Projeto	Plano do Projeto / Discussões Face-a-face
	13	Avaliação do Progresso do Projeto / Status do Projeto	Plano de iteração / Relatórios ( <i>iteration burn-down e project burndown</i> )
	14	Plano de Gestão de Comunicação	Plano de iterações / Plano do Projeto
	15	Revisão do resultados documentados no projeto	Plano de documentação das iterações
	16,17	Registro de Problemas e Monitoramento da Ações Corretivas / Ações Corretivas Plano	Revisão dos itens de trabalho / plano de iteração

#### 4.2.3 Cadastro de usuário

Para o usuário ter acesso ao sistema deverá ser realizado um cadastro por um usuário administrativo do mesmo. Deve ser fornecido os seguintes dados como: Nome completo, data de nascimento, RG, CPF, E-mail, login, Senha, Função que irá desempenhar no sistema e ao final clicar no botão cadastrar conforme exibido na figura 11.

#### 4.2.4 Consulta de Pessoal

O usuário autenticado e que possua permissões de acesso a este requisito, poderá consultar o pessoal cadastro no sistema. Nesta tela é possível filtrar informações referentes aos usuários cadastrados como; nome, email, login de acesso.

#### 4.2.5 Definir permissão de acesso por função do usuário

O usuário poderá definir acesso as funcionalidades do sistema de acordo com a função do usuário. Desse modo deve ser selecionada a função do usuário e selecionada a tela a qual este terá permissão, as ações nas telas estão segmentadas de acordo com sua característica ex: Consulta e Impressão, Cadastro, Edição e Exclusão.

#### 4.2.6 Cadastro de Páginas

O usuário poderá definir quais páginas será passível de acesso no sistema, estas páginas permitira controlar o acesso por função do usuário no sistema.

#### 4.2.7 Alterar senha

O usuário poderá modificar sua senha por meio deste requisito. O Sistema solicitará que o usuário forneça um nova senha e em seguida confirme esta senha, será informado o grau de dificuldade da senha informada.

#### 4.2.8 Consulta dos termos de abertura de projeto

O usuário poderá consultar todos os termos de abertura de projeto de software. Os termos de abertura de projeto de software é o primeiro documento a ser gerenciado na ferramenta. No termo de abertura irá conter as definições iniciais de um projeto de desenvolvimento de software, além disso, será primeiro artefato documental a ser homologado junto aos stakeholders.

#### 4.2.9 Consulta dos projetos

O usuário poderá consultar todos os projetos de desenvolvimento de software criados no sistema. Um projeto é oriundo de um termo de abertura de projeto de software desse modo todo projeto é fundamentado em um TAP. Cada TAP poderá ter um, e somente um, projeto associado ao mesmo.

#### 4.2.10 Consulta das atas das reuniões do projeto

O usuário poderá consultar todas as atas de reuniões cadastradas no sistema. Uma ata de reunião é a formalização de uma cerimônia onde foram definidas ações consagradas e tomadas de decisões no projeto.

#### 4.2.11 Consulta das estruturas analítica do projeto

**Tabela 2: Análise do produto de trabalho MR-MPS e a abordagem híbrida *OpenUP***

Característica	Bitrix24	Jira	Wrike	T-AADSP
Versão Web	x	x	x	x
Sistema de Autenticação	x	x	x	x
Definição de Cronograma do Projeto	x	x	x	x
Definição de Termo de Abertura de Projeto (TAP)				x
Definição de Equipe executora do Projeto	x	x	x	x
Controle de Reuniões do Projeto	x	x	x	x
Definição da Estrutura Analítica do Projeto (EAP)				x
Controle dos Recursos Especiais do Projeto				x
Controle de Acesso aos conteúdos da plataforma	x	x	x	x
Mensuração de Projeto de software por Análise de Ponto de Função				x

O usuário poderá cadastrar três (Pacote, fases, entregas) tipos distintos de EAP a um projeto de software. É visualizar uma EAP na estrutura de um diagrama ou consulta-la por meio de uma lista.

#### 4.2.12 Consulta dos documentos de estimativa de escopo do projeto

O usuário poderá consultar os projetos a qual será realizada a estimativa de escopo. A estimativa de escopo deverá ser realizada por meio do cálculo do ponto de função do projeto.

#### 4.2.13 Consulta dos cronogramas de execução do projeto

O usuário poderá consultar a relação de projetos no sistema a fim de realizar o cadastro das atividades a compor o mesmo através de um cronograma de execução do projeto.

#### 4.2.14 Consulta dos recursos especiais do projeto

O usuário poderá consultar os projetos para definição dos recursos especiais do mesmo por meio do botão detalhar.

#### 4.2.15 Cadastro do termo de abertura de projeto

O usuário poderá cadastrar um novo termo de abertura de projeto informando o nome do projeto, justificativa, alinhamento estratégico, orçamento previsto, cronograma inicial, premissas do projeto, restrições do projeto e mensurar os riscos do projeto.

#### 4.2.16 Cadastro do projeto

Um projeto é só poderá ser cadastrado mediante existir um termo de abertura a ser associado ao mesmo. Deve ser informado o valor do projeto e seu cronograma.

#### 4.2.17 Cadastro das atas das reuniões do projeto

São realizadas diversas cerimônias em um projeto, poderá ser realizado o cadastro da ata de reunião oriunda desta cerimônia, que desse modo, poderá ser consultada posteriormente e homologada junto aos stakeholders.

#### 4.2.18 Cadastro das estrutura analítica do projeto

Todo projeto poderá possuir três modelos de EAP (Pacote, Entrega e Fase), deve ser cadastrado o pacote a qual uma atividade será inserida, e desse modo, deve ser cadastrada uma ou mais atividades a compor o pacote cadastrado. É possível visualizar a estrutura do projeto em forma de diagrama.

#### 4.2.19 Cadastro dos documentos de estimativa de escopo do projeto

Para estimar o tamanho do projeto deverá ser realizado o cálculo dos pontos de função do projeto. Para este cálculo poderá ser consultado o valor base definido para o cálculo, como também, as quatorze características gerais dos projetos de software objetivando realizar o ajuste do pontos de função. Após a inserção dos dados das funções dos tipos de dados e transação. Deverá ser informado as características gerais do sistema encontradas. Por fim o sistema automaticamente informará o valor total dos pontos de funções calculados e o valor ajustado.

#### 4.2.20 Cadastro do cronograma de execução do projeto

O usuário poderá definir data para realização de atividades do projeto. Estas datas serão catalogadas em um cronograma de execução de atividades.

#### 4.2.21 Cadastro dos recursos especiais do projeto

O usuário poderá cadastrar os recursos especiais do projeto. Para cadastrar um novo recurso deverá ser informado o nome do recursos e o valor do mesmo.

#### 4.2.22 Detalhamento de informações do projeto

Neste requisito será possível informar demais informações ao projeto, como equipe executora do mesmo, visualizar as funções definidas ao projeto por meio do TAP, também será possível visualizar estas informações por meio do visualizações de dados. Através deste requisito o usuário poderá alterar informações do projeto.

#### 4.2.23 Detalhamento de atas das reuniões do projeto

Neste requisito será possível detalhar as informações de um determinada ata de reunião, podendo também ser alterado os dados existentes como título, data realização, organizador, além da pauta e assuntos tratados.

#### 4.2.24 Detalhamento da estrutura analítica do projeto

Através deste requisito será possível detalhar informações ao mesmo tempo que editar/excluir informações já inseridas.

## 5. DESENVOLVIMENTO DA FERRAMENTA T-AADSP

### 5.1 Tecnologias utilizadas

Este projeto foi todo desenvolvido utilizando as seguintes tecnologias:

- **Servidor para o web:** GlassFish 4.1.1.
- **Apache Maven Project:** Apache Maven é uma ferramenta de gerenciamento de projetos de software e compreensão. Baseado no conceito de um modelo de objeto de projeto (POM), Maven pode gerenciar construção de um projeto, elaboração de relatórios e documentação de uma peça central de informações.
- **JSF 2.2:** JavaServer Faces (JSF) é uma especificação Java para a construção de interfaces de usuário baseadas em componentes para aplicações web.
- **Primefaces 5.3** PrimeFaces é um open source User Interface biblioteca (UI) componentes para JavaServer Faces (JSF) aplicações baseadas, criados por PrimeTek.
- **Bootstrap 2.3.2:** Bootstrap é uma estrutura de front-end elegante, intuitiva e poderosa para o desenvolvimento web mais rápida e fácil, criado por Mark Otto e Jacob Thornton, e mantida pela equipe principal com o apoio maciço e envolvimento da comunidade.
- **Hibernate 4.3.6:** O Hibernate é um framework para o mapeamento objeto-relacional escrito na linguagem Java.
- **SGBD (Sistema de Gerenciamento de Banco de Dados):** SQLServer 2008.

## 6. PROJETO EM ESTUDO

## 7. ANÁLISE DE DADOS REAIS

## 8. AVALIAÇÃO DA FERRAMENTA

## 9. CONCLUSÃO

## 10. REFERÊNCIAS

- [1] Bitrix24. <https://www.bitrix24.com.br/#tour>. Acessado em: 11 de julho de 2016.
- [2] Fluxos de processos pmbok. <https://www.linkedin.com/pulse/planejar-o-gerenciamento-dos-custos-diego-mendes-ro>. Acessado em: 24 de julho de 2016.
- [3] Introdução ao openup (open unified process). <http://epf.eclipse.org/uploads/16.pdf>. Acessado em: 10 de julho de 2016.
- [4] Xp. <http://www.desenvolvimentoagil.com.br/xp/>. Acessado em: 10 de julho de 2016.
- [5] Jira. [http://www.oatsolutions.com.br/ferramentas/ferramentas\\_JIRA.htm?gclid=CK0wyovm680CFRIFkQodhCcGZQ](http://www.oatsolutions.com.br/ferramentas/ferramentas_JIRA.htm?gclid=CK0wyovm680CFRIFkQodhCcGZQ), 2016. Acessado em: 15 de julho de 2016.
- [6] BECK, K. Manifesto para o desenvolvimento ágil de software. <http://www.manifestoagil.com.br/>, 2001. Acessado em: 06 de julho de 2016.
- [7] INSTITUTE, P. M. *Um Guia do conhecimento em gerenciamento de Projetos - Guia PMBOK 5º ed.* Project Management Institute, Inc, Newtown Square, Pennsylvania 19073-3299 USA, 2013.
- [8] MAURÍCIO MASSARU ARIMOTO, EDSON MURAKAMI, V. V. D. C. E. M. I. C. Adherence analysis of agile methods according to the mr-mps reference model. *VIII Simpósio Brasileiro de Qualidade de Software*, 249–263.

- [9] OF ELECTRICAL, I., AND ELECTRONICS ENGINEERS. *IEEE-STD-610*. Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries, 1990.
- [10] SABBAGH, R. *SCRUM: Gestão Ágil para Projetos de Sucesso*. Casa do Código, 2013.
- [11] SOFTEX. *MPS.BR - Melhoria de Processo do Software Brasileiro: Guia Geral MPS de Software*. Sociedade SOFTEX, 2012.
- [12] SOUZA, A. C. *Adaptive Approach for Deployment of Software Process - AADSP*. LABRASOFT, 2016.
- [13] WEBER, S., HAUCK, J. C. R., AND VON WANGENHEIM, C. F. Estabelecendo processos de software em micro e pequenas empresas. *Simpósio Brasileiro de Qualidade de Software* (1993), 16.

## 11. ANEXOS

Figura 5: Autenticação do sistema T-AADSP

Figura 6: Mensagem não foi possível realizar a autenticação do usuário!



Figura 7: A senha informada está incorreta!



Figura 10: Não foi possível realizar o envio de uma nova senha ao usuário



Figura 8: Solicitação de nova senha via e-mail

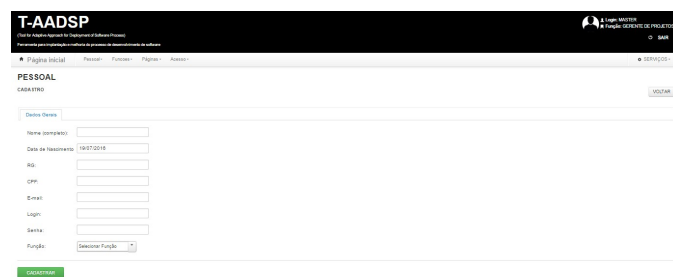


Figura 11: Cadastrando novos usuários no sistema



Figura 9: Foi enviada uma nova senha ao email do usuário

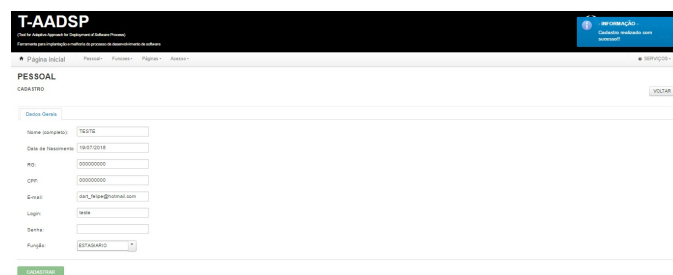
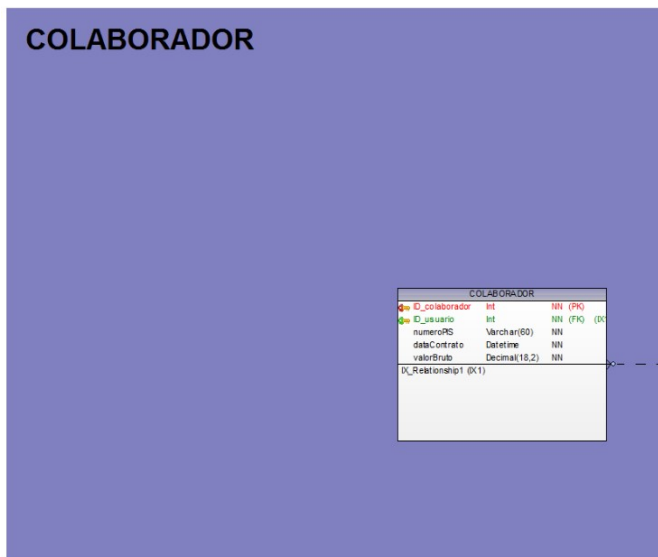
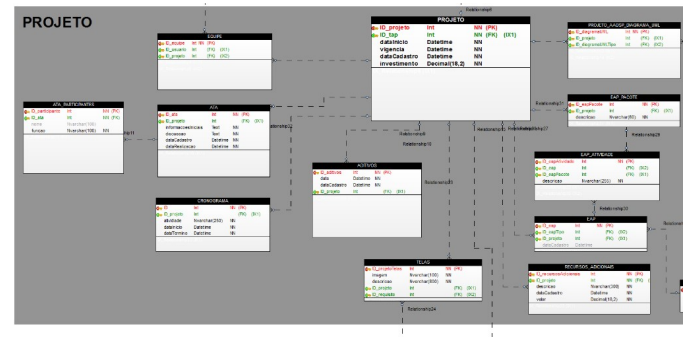


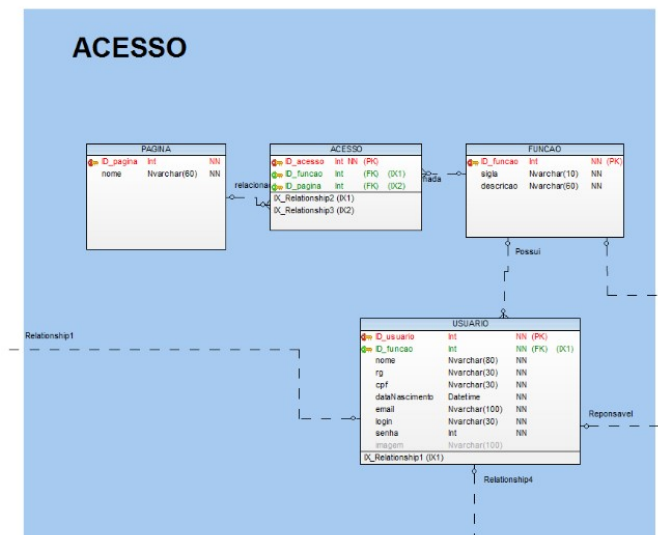
Figura 12: Confirmação de novo usuário cadastrado no sistema



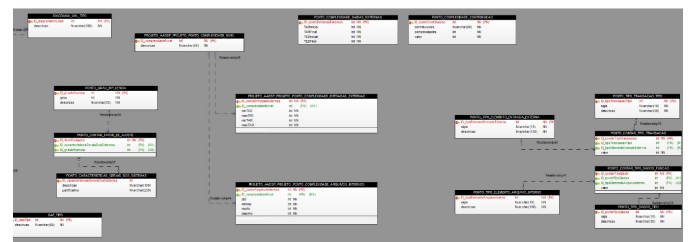
**Figura 13: Diagrama de Entidade e Relacionamento - Schema Colaborador**



**Figura 16: Diagrama de Entidade e Relacionamento - Schema Projeto Parte 1**



**Figura 14: Diagrama de Entidade e Relacionamento - Schema Acesso**



**Figura 17: Diagrama de Entidade e Relacionamento - Schema Projeto Parte 2**

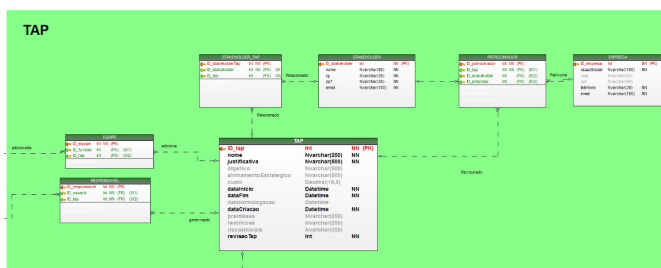


Figura 15: Diagrama de Entidade e Relacionamento  
- Schema Tap