UNIVERSIDADE FEDERAL DO PARÁ

INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS

FACULDADE DE COMPUTAÇÃO

CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

Mariano Jefferson Batista Gomes Junior

UMA DEFINIÇÃO DE *FRAMEWORK* DE PROCESSO PARA EXECUÇÃO DE PROCESSOS DE SOFTWARE

Belém

2011

Mariano Jefferson Batista Gomes Junior

UMA DEFINIÇÃO DE *FRAMEWORK* DE PROCESSO PARA EXECUÇÃO DE PROCESSOS DE SOFTWARE

Trabalho de Conclusão de Curso apresentado como requisito parcial para a obtenção do título de Bacharel em Sistemas de Informação, pelo Curso de Bacharelado em Sistemas de Informação da Universidade Federal do Pará

Orientador: Prof. Dr. Sandro Ronaldo Bezerra Oliveira

Belém

2011

Dedico este trabalho a todos aqueles que contribuíram para a realização deste, de forma direta e, principalmente, aqueles que contribuíram de forma indireta, me apoiando e me incentivando.

AGRADECIMENTOS

Em oração quero agradecer ao meu bom Deus, o qual pude ver sua misericórdia e seu amor em minha vida. Agradeço-te Deus por tudo, muito antes deste trabalho o qual concluí. Na paz que senti durante a prova do vestibular e uma certeza em meu coração que você estava providenciando uma vaga para mim. Agradeço-te pelas oportunidades que colocastes na minha frente, pelas pessoas que pude conhecer, pelas experiências que pude passar, por cada um dos momentos vividos, te agradeço por estar sempre na minha frente, me guiando e cuidando de mim. Sei Pai Celeste que sem ti não chegaria a lugar nenhum e que tu Deus, me deu esta vitória. Sou grato a ti Deus por toda a minha vida.

Deus querido, quero agradecer também pelos meus amados pais, agradecer pela oportunidade de ter sido filho deles, se não fosse por eles talvez eu não te conhecesse como um Deus amoroso e presente e sim teria uma imagem de um Deus distante, ausente. Eles são pais zelosos, que possuem princípios, pais que ao custo de muito suor me deram uma boa educação. Todo o conhecimento que possuo hoje foi gerado em cima da fadiga deles. Pais vocês são meus heróis.

Meu querido pai Sr. Mariano Jefferson B. Gomes o qual tenho orgulho de carregar seu nome, eu agradeço pela paciência, pelos conselhos, por me mostrar o caminho certo a seguir, obrigado por me mostrar, como Jesus também fez, que devemos retribuir a raiva e a traição com amor e um sorriso e principalmente com o perdão. Você meu querido pai é um exemplo de guerreiro, exemplo de pai, exemplo de homem de caráter. Obrigado por ter lutado contra as turbulências da vida, em tempos difíceis você se agarrou nas promessas de dias melhores e lutou pela minha mãe, pelo meu irmão e por mim. Eu entendo que minhas noites de sossego foram custeadas por noites e noites de insônias sua e da mamãe clamando a Deus por dias melhores e ele foi fiel, obrigado pai.

Agradeço a minha querida mãe dona Ilma R. Dezincourt Gomes a qual eu amo tanto, você sempre foi incansável em me mostrar que o estudo é a maior herança que vocês podem deixar para mim, eu sou muito grato pela dedicação que você cuidou não só de mim como também de meu irmão, lembro-me de quando éramos menores e você atravessava a cidade para nos levar para escola, hoje tenho noção do sacrifício que era e isto só aumenta minha gratidão. Você também nos mostrou o valor da honestidade, me mostrou que uma consciência tranquila é mais válida que toda a riqueza desta terra. Obrigado mãe pelos ensinamentos, suas sábias palavras são alicerces para o meu caráter. Obrigado mãe pelo amor demonstrado mesmo durante as repreensões, obrigado por ter sido a pedra de sustentação desta família que tanto amo, nas dificuldades da vida você foi incansável em nos dar animo e nos mostrar que não há dor que seja eterna.

Quero te agradecer Deus pelo meu irmão Daniel Afonso D. Gomes o qual dividi e divido muitos momentos, obrigado Senhor por ter me dado a oportunidade de ser irmão dele, uma pessoa divertida como nosso pai, sábia, reflexiva, persistente e que não se cala diante do errado.

Este pouco espaço que me resta não poderia deixar de agradecer minha namorada, sei que foi você Deus que colocou a Kelly Fernanda B. Vilhena em meu caminho, também sei que Você fez nascer este infinito amor entre eu e ela. Amor obrigado, pela paciência, pela dedicação, pelas palavras de animo, obrigado por apostar em mim, obrigado por me amar e andar comigo ao meu lado, obrigado por me dar seu sorriso que é minha inspiração. São tantas as coisas a qual tenho que te agradecer, mas aqui deixo o meu muito obrigado e o meu “TE AMO”.

Agradeço a todos que o Senhor pôs em meu caminho, em especial quero agradecer ao Prof. Sandro Bezerra e ao amigo Carlos Portela por terem apostado em mim, confiado e ajudado.

E finalmente Senhor, termino esta oração agradecendo a ti mais uma vez pelas conquistas e lutas, Senhor faz de mim vazo de honra, que meus princípios não se esvaiam de mim. Amém.

“Não temas porque estou contigo”

(Isaías, 43:5)

RESUMO

Este trabalho descreve um *framework* para a execução de processos de software, a partir de boas práticas sugeridas nos modelos de qualidade e de um formalismo de execução. Neste *framework* estão identificados os atores responsáveis por realizar cada uma das atividades e as fases as quais estas atividades pertencem. Este *framework* é um dos componentes que integra um ferramental de apoio à execução de processos de software, denominado Spider-PE.

Palavras-chave: Execução de Processo de Software, *Framework* de Processo, Aderência aos Modelos de Qualidade.

ABSTRACT

This paper describes a framework for execution of process's software, from good practices suggested in the models of quality and formalism of a execution. In this framework are identified the actors responsible to do each activity and the phases which belong to these activities. This framework is a component that integrates a tool to support the execution of processes' software, called Spider-PE.

Keywords: Software Process Enactment, Framework, Adherence to Quality Models.

SUMÁRIO

1 INTRODUÇÃO 13

1.1 JUSTIFICATIVA 14

1.2 MOTIVAÇÃO 15

1.3 OBJETIVOS 16

1.3.1 Objetivo Geral 16

1.3.2 Objetivos Específicos 16

1.4 METODOLOGIA 16

1.5 ESTRUTURA DO TRABALHO 17

1.6 CONSIDERAÇÕES FINAIS 17

2 TECNOLOGIA DE PROCESSOS DE SOFTWARE 17

2.1 PROCESSO DE SOFTWARE 18

2.1.1 Definição de Processo de Software 18

2.1.2 Estrutura de Processo de Software 19

2.2 QUALIDADE DO PROCESSO DE SOFTWARE 21

2.3 MODELOS DE QUALIDADE 22

2.3.1 CMMI 23

2.3.2 MPS.BR 26

2.4 EXECUÇÃO DE PROCESSOS 27

2.4.1 Definição 28

2.4.2 Contexto 28

2.5 CONSIDERAÇÕES FINAIS 30

3 O *FRAMEWORK* DE EXECUÇÃO DE PROCESSOS 31

3.1 CONTEXTUALIZAÇÃO 31

3.1.1 Projeto SPIDER 32

3.1.2 Pesquisas Relacionadas 32

3.2 ADERÊNCIA AOS MODELOS DE QUALIDADE 34

3.3 *FRAMEWORK* SPIDER-PE 37

3.3.1 Fases do *Framework* 37

3.3.2 Atores do *Framework* 40

3.3.3 Descrição das Atividades 41

3.3.3.1 Fase de Execução do Processo 42

3.3.3.2 Fase de Execução das Atividades do Processo 61

3.3.3.3 Fase de Aplicação do Formalismo de Execução 66

3.4 APLICAÇÃO DO *FRAMEWORK* 73

3.5 CONSIDERAÇÕES FINAIS 74

4 CONCLUSÕES 75

4.1 RESULTADOS OBTIDOS 75

4.2 TRABALHOS FUTUROS 76

5 REFERÊNCIAS 77

LISTA DE FIGURAS

Figura 2.1 – Elementos de um Processo de Software 21
Figura 2.2 – Estrutura do Modelo CMMI 24
Figura 2.3 – Comparação entre os Modelos 27
Figura 2.4 – Ciclo de Vida do ImPProS (Adaptado de OLIVEIRA, 2007) 29
Figura 3.1 – Ciclo de Vida de Processos no Spider 33
Figura 3.2 – Fase de Execução do Processo 39
Figura 3.3 – Fase de Execução das Atividades do Processo 39
Figura 3.4 – Fase de Aplicação do Formalismo de Execução 40

LISTA DE QUADROS

Quadro 2.1 – Relação entre Capacidade e Maturidade no CMMI 25
Quadro 3.1 – Atributos de Processo do Nível F (SOFTEX,2011)..... 35
Quadro 3.2 – Objetivos Genéricos do Nível 2 (SEI, 2010) 36

LISTA DE ABREVIATURAS

AP:	Atributo de Processo;
CMMI:	*Capability Maturity Model Integration;*
CMMI-DEV:	*CMMI for Development;*
GG:	Generic Goal;
GP:	Generic Practices;
IEC:	*International Engineering Consortium;*
ISO:	*International Standartization Organization;*
MPS.BR:	Melhoria de Processos do Software Brasileiro;
MR-MPS:	Modelo de referência para melhoria do processo de software;
RAP:	Resultado de Atributo de Processo;
Softex:	Associação para Promoção da Excelência do Software Brasileiro;

1 INTRODUÇÃO

Esse capítulo apresenta uma visão geral dos referidos assuntos que são tratados e corresponde a definição deste trabalho. Nele são apresentados a justificativa, a motivação, os objetivos e a metodologia, bem como, a estrutura do mesmo, onde é elucidado no final deste capítulo.

1.1 JUSTIFICATIVA

O desenvolvimento de software envolve um elevado grau de complexidade. Nele é usada uma grande quantidade de conhecimento, tanto tecnológico quanto gerencial, o que possibilita inúmeras alternativas para a execução das atividades do seu processo. Este processo de software é o principal objeto de estudo da Engenharia de Software (REIS, 2003).

Processo de Software (PS) pode ser definido como o conjunto de tarefas necessárias para transformar os requisitos dados pelos usuários em software (HUMPHREY, 1989). Associados a estes processos tem-se o conceito de modelos de processo de desenvolvimento de software. Um modelo de desenvolvimento pode ser definido como uma descrição abstrata do processo de software, onde diversos tipos de informação devem ser integrados para indicar quem, quando, onde, como e por que os passos são realizados (REIS, 2000). Esta descrição formal permite a análise, compreensão e execução de um processo de software.

Paralelamente a isto, nota-se que padrões e modelos de qualidade vêm sendo utilizados amplamente para avaliar a qualidade do software, sendo que o ponto principal dessa análise são os processos adotados para o desenvolvimento. Os mecanismos utilizados para a certificação de qualidade de software são baseados na maturidade atingida pelas organizações no que diz respeito à forma como conduzem seus processos de desenvolvimento de software (OLIVEIRA *et al*., 2005). O uso destes padrões tende a simplificar a

relação entre a organização desenvolvedora de software e o cliente, uma vez que este pode identificar a qualidade dos serviços prestados por determinadas organizações através dos padrões que elas utilizem.

O trabalho proposto está inserido no contexto do projeto SPIDER – *Software Process Improvement: DEvelopment and Research* (OLIVEIRA *et al*.,2010), cujo o objetivo é a realização de pesquisas referentes à Melhoria Contínua de Processos de Desenvolvimento de Software e Modelos e Normas de Qualidade, dentre outras. Especificamente, este trabalho completa parte das pesquisas descritas em PORTELA *et al*. (2011), que possui como finalidade a definição de um ferramental de apoio à execução flexível de processos de software aderente a modelos de qualidade. Este ferramental é tema de uma dissertação de mestrado vinculada ao Centro de Informática da Universidade Federal de Pernambuco (CIn/UFPe) e desenvolvida no âmbito do Projeto SPIDER/UFPa (ver Seção 3.1.1).Portanto, a definição de um *framework* que especifica atividades de apoio à execução de processos de software contribuiu no desdobramento das pesquisas descritas em PORTELA *et al*. (2011) .

A contribuição deste trabalho consiste na definição de um processo genérico, para apoiar às organizações na condução da execução dos seus processos de forma aderente às recomendações dos principais modelos de qualidade adotados no Brasil: CMMI e MPS.BR.

1.2 MOTIVAÇÃO

Este trabalho foi motivado pelo interesse despertado na área de Engenharia de Software, e mais especificadamente na área de Qualidade de Software, incitado tanto pelas disciplinas ministradas em sala de aula, quanto pela participação no Projeto SPIDER. Durante a participação no projeto surgiu a oportunidade de trabalhar com o tema em questão, fazendo com que se conhecesse e optasse por desenvolver o trabalho de conclusão de curso relacionado a área de execução de processos de software.

1.3 OBJETIVOS

1.3.1 Objetivo Geral

Definir um *framework* para execução de processo, a partir de um mapeamento entre modelos de qualidade e da especificação de um formalismo para a execução de processos.

1.3.2 Objetivos Específicos

- Estudo dos modelos de qualidade CMMI-DEV (CMMI *for Development*) e o MR-MPS (Modelo de Referência do MPS) objetivando identificar recomendações referentes à execução de processos;
- Definição de um *framework* de processo para descrição de um fluxo de execução;
- Apoio à definição de um ferramental para execução de processos.

1.4 METODOLOGIA

A metodologia adotada consistiu na realização de pesquisas bibliográficas, realizadas com o objetivo de analisar e comparar os modelos de qualidade CMMI-DEV e o MR-MPS. Após esta análise, foi realizado um mapeamento entre modelos e normas para a execução de processos, sobretudo os modelos CMMI-DEV e MR-MPS, procurando identificar similaridades e diferenças entre as práticas sugeridas pelos mesmos, para que fossem filtradas as práticas necessárias para a execução de um processo de software. O resultado deste mapeamento foi comparado com os estudos publicados a respeito da equivalência destes dois modelos.

Posteriormente, foi especificado um formalismo para acrescentar uma semântica comportamental à linguagem de modelagem de processos SPIDER_ML, para que este suporte a execução de processos.

Os resultados obtidos a partir das pesquisas bibliográficas e do mapeamento serviram de insumo para a definição do *framework* para execução de processos, proposto neste trabalho, aderente às normas e modelos de qualidade CMMI-DEV e MR-MPS.

1.5 ESTRUTURA DO TRABALHO

Além desse capítulo inicial, que apresenta uma visão geral da pesquisa realizada neste trabalho, no Capítulo 2 são apresentados os principais modelos relacionados à qualidade de software, utilizados como base para definição das atividades do *framework*. No Capítulo 3 é apresentada a principal contribuição deste trabalho, o *Framework* Spider-PE, com suas respectivas atividades, fases e atores. Por fim, no Capítulo 4, as conclusões obtidas a partir deste trabalho são descritas.

1.6 CONSIDERAÇÕES FINAIS

Esse capítulo apresentou uma visão geral sobre os assuntos que norteiam e fundamentam a pesquisa para o desenvolvimento do respectivo trabalho de conclusão de curso. Além disso, mostra qual a motivação para o desenvolvimento do mesmo, sua justificativa e seus principais objetivos.

2 TECNOLOGIA DE PROCESSOS DE SOFTWARE

Os processos de software podem apresentar um alto grau de complexidade bem como possibilitar muitas alternativas para execução de suas

atividades. Desta forma, a definição de um processo de software permite que os profissionais da Engenharia de Software possam trabalhar de forma ordenada, possibilitando um melhor entendimento do seu trabalho, bem como de outras atividades executadas por outros membros da mesma equipe (HUMPHREY,1989).

Embora o processo de software seja eficiente para o desenvolvimento do software, não existe um processo que possa ser genericamente aplicado a todos os projetos pelo fato de que nenhum projeto é idêntico a outro. Em cada organização pode-se ter grandes ou pequenas variações nas políticas e procedimentos organizacionais, métodos e estratégias de aquisição, tamanho e complexidade do projeto, requisitos e métodos de desenvolvimento do sistema, equipe (pessoas), entre outros fatores que influenciam na forma como um produto de software é adquirido, desenvolvido, operado e mantido (ISO/IEC 2008).

Neste capítulo, são abordados alguns conceitos que contextualizam o estudo realizado do referido trabalho. Inicialmente, uma definição de Processo de Software é apresentada. Em seguida, há a explanação sobre qualidade do processo de software seguida dos parâmentos da melhoria contínua e dos modelos de qualidade existentes. Por fim, aborda-se a Execução do processo de software, principal objeto de estudo deste trabalho.

2.1 PROCESSO DE SOFTWARE

2.1.1 Definição de Processo de Software

O processo de software é definido como o conjunto de tarefas utilizadas pela Engenharia de Software que se fazem necessárias para transformar os requisitos dos usuários em software (HUMPHREY,1989). Um processo definido provê um conjunto de benefícios para uma organização como (BARROS, 2010):

- possibilita uma comunicação efetiva sobre o processo entre os usuários, desenvolvedores, gerentes, e demais pessoas envolvidas no processo;
- facilita o gerenciamento do processo;
- facilita o reuso do processo;
- facilita o aperfeiçoamento do processo.

2.1.2 Estrutura de Processo de Software

De acordo com OLIVEIRA (2007), dentro do processo de software existem alguns conceitos necessários para o bom andamento deste, como atividades a serem realizadas, recursos utilizados, artefatos consumidos e gerados, procedimentos adotados, paradigmas e tecnologia adotada além do modelo de ciclo de vida utilizado.

- **Atividade**: As atividades são compostas por tarefas ou trabalho a serem realizados. Para a realização de uma atividade, além do uso de recursos, pode-se precisar de artefatos gerados em atividades já realizadas previamente como também gerar artefatos para utilizar ou não em atividades futuras. Para se realizar uma atividade, pode-se adotar um procedimento, além disto, é possível decompor a atividade em sub-atividades. As atividades podem ser classificadas em: atividades de gerência, atividades de suporte e manutenção, atividades de construção e atividades de avaliação da qualidade. Como exemplo de uma atividade realizada no projeto de software pode-se citar “Coletar os Requisitos”;

- **Recursos**: É qualquer fator de produção necessário para que uma atividade seja executada, mas que não seja insumo para esta atividade. Os recursos podem ser classificados em: recursos de hardware, recursos de software e recursos humanos. Como exemplo de recurso humano pode-se citar o “Analista de

Sistema", que poderá fazer uso de um "PC" (recurso de hardware) para desempenhar suas tarefas, que tenha instalado no seu sistema uma ferramenta para modelagem de projetos de software (recurso de software) que automatiza o procedimento UML (*Unified Modeling Language*);

- **Artefatos**: São produtos de software gerados ou consumidos por atividades durante sua execução. Os artefatos podem ser classificados em: artefatos de código, componentes de software, documentos, diagramas, modelos, entre outros. Pode-se citar como exemplo de artefato de entrada (consumido) a "Lista de Requisitos" e como artefato de saída (gerado) da atividade o documento "Plano do Projeto";

- **Procedimentos**: São procedimentos estabelecidos e ordenados que podem ser seguidos para a realização de uma atividade do processo. Quanto à sua natureza, procedimentos podem ser classificados em: métodos, técnicas e diretrizes. Como exemplo, para executar uma determinada atividade definida pode-se ter como procedimento a sugestão de que se faça o uso da UML para a definição de cenários das necessidades do usuário.

Um processo apresenta um conjunto parcialmente ordenado de atividades que interagem entre si e que podem estar associadas a artefatos, recursos humanos ou ferramentas (LONCHAMP *apud* BARROS, 2010). As atividades, ferramentas, artefatos, recursos humanos, entre outros, são denominados elementos de processo.

A Figura 2.1 mostra a relação entre estes elementos que auxiliam no entendimento da definição de processo de software.

Figura 2.1 – Elementos de um Processo de Software

Nas organizações, podem coexistir diversos projetos com características específicas. Contudo, existe um conjunto de elementos fundamentais que podem ser incorporados em quaisquer processos definidos nas organizações. A norma ISO/IEC TR 15504 (ISO/IEC, 2003) define o conjunto destes elementos fundamentais como Processo Padrão, ou seja, o processo básico que guia o estabelecimento de um processo comum na organização. Desta forma, um processo padrão define uma estrutura única a ser seguida por todas as equipes envolvidas em um projeto de software (MAIDANTCHIK *apud* OLIVEIRA, 2007), independente das características do software a ser desenvolvido. A partir de um Processo Padrão, deve-se definir um processo que esteja de acordo com as características de um projeto específico. A esta instância do projeto padrão denomina-se Processo Instanciado (BARROS, 2010).

2.2 QUALIDADE DO PROCESSO DE SOFTWARE

Sendo o processo de software um conjunto de tarefas necessárias para transformar os requisitos fornecidos pelos usuários em software

(HUMPHREY,1989), é preciso que se façam modelos de desenvolvimento do processo de software para serem adotados ou tomados como referências pelas organizações desenvolvedoras de software, permitindo assim aos engenheiros de software especificar, projetar, implementar e manter sistemas de software, avaliando e garantindo suas qualidades.

Um modelo de processo de software pode ser definido como uma descrição abstrata do processo de software, onde diversos tipos de informação devem ser integrados para indicar quem, quando, onde, como e por que os passos são realizados (REIS, 2000). Realizando esta descrição formal é que se permite a análise, compreensão e execução de um processo de software.

Nos últimos anos, diversos padrões e modelos têm sido desenvolvidos com o objetivo de definir, avaliar e melhorar a qualidade dos processos de software. Com o advento de padrões internacionais e a maior preocupação com a qualidade de software, as organizações passaram a adotar referidos padrões e modelos na definição de processos de software, buscando, assim, melhorar a qualidade de seus produtos, aumentar a produtividade de suas equipes, e reduzir os custos e os riscos associados com o desenvolvimento de software (PORTELA *et al*., 2011).

2.3 MODELOS DE QUALIDADE

De acordo com OLIVEIRA (2007), à medida que as organizações aumentam as suas preocupações com a qualidade com que seus softwares são produzidos, mais estas procuram seguir as orientações de modelos e padrões de qualidade de processo, tais como ISO 90003, CMMI, ISO/IEC 12207, ISO/IEC TR 15504 e MPS.BR.

Atualmente a indústria nacional de software se baseia principalmente em dois destes modelos de qualidade de software, o CMMI que tem notoriedade internacional e consiste das melhores práticas relativas às atividades de desenvolvimento e manutenção aplicadas a produtos e serviços, reunindo as

principais práticas no que diz respeito ao processo de software. Ele abrange práticas que cobrem o ciclo de vida do produto desde a concepção até a entrega e manutenção, e se concentra no trabalho necessário para construção e manutenção do produto em sua totalidade. O MPS.BR teve como base o próprio CMMI, além da ISO/IEC 12207 (2008) e a ISO/IEC 15504 (2003). Este modelo se diferencia dos demais por ser voltado para o mercado nacional, tornando mais acessível a implementação de programas de qualidade nas pequenas e médias empresas nacionais. Estes modelos estão melhor apresentados a seguir.

2.3.1 CMMI

O CMMI – *Capability Maturity Model Integration* é um modelo de referência que contém práticas necessárias à maturidade de disciplinas específicas da Engenharia de Software. O CMMI-DEV (*CMMI for Development*) consiste das melhores práticas relativas às atividades de desenvolvimento e manutenção aplicadas a produtos e serviços. Ele abrange práticas que cobrem o ciclo de vida do produto desde a concepção até a entrega e manutenção, e se concentra no trabalho necessário para construção e manutenção do produto em sua totalidade.

No que diz respeito à melhoria da qualidade no processo de software, o CMMI apresenta duas abordagens para realizar estas melhorias: Representação Contínua e Representação por Estágio. Para a Representação Contínua utiliza-se o conceito de “Nível de Capacidade”, permitindo que as organizações melhorem de forma incremental os processos correspondentes a uma ou mais áreas de processo individualmente selecionadas pela organização. Já na Representação por Estágio utiliza-se o conceito de “Nível de Maturidade”, permitindo que as organizações melhorem um conjunto de processos inter-relacionados e, de forma incremental, tratem sucessivos conjuntos de áreas de processo. Ambas as representações possuem os

mesmos componentes e estes têm a mesma hierarquia de configuração, como é possível ver na Figura 2.2.

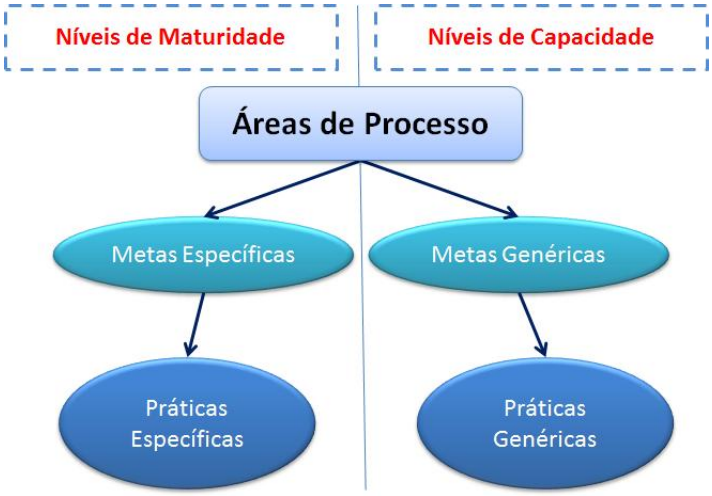


Figura 2.2 – Estrutura do Modelo CMMI

A Representação Contínua foca na capacidade do processo das organizações, tendo suas medidas associadas pelo Nível de Capacidade. Esta associação aplica-se à melhoria de processo da organização em áreas de processo individuais. Estes níveis são uma forma de melhorar, de maneira incremental, os processos correspondentes a uma determinada área de processo. Existem seis níveis de capacidade que vão de 0 (zero) a 5 (cinco). Um nível de capacidade é composto por uma meta genérica (e suas práticas genéricas) que pode melhorar os processos da organização associados àquela área de processo. O progresso nos níveis de capacidade de uma área de processo é alcançado através da aplicação de práticas genéricas ou práticas alternativas adequadas aos processos associados àquela área de processo.

A Representação por Estágio tem seu foco na maturidade de processo, tendo suas medidas associadas ao Nível de Maturidade. Esta associação visa à melhoria dos processos da organização em um conjunto de áreas de processo. Os Níveis de Maturidade auxiliam na previsão dos resultados de futuros projetos da organização. Existe um total de cinco níveis de maturidade que variam de 1 (um) a 5 (cinco). “Um nível de maturidade é composto por práticas específicas e genéricas relacionadas a um conjunto predefinido de áreas de processo que melhoram o desempenho global da organização” (SEI,

2010). Cada Nível de Maturidade possui um conjunto de subconjunto de processos relacionado a eles, cada novo nível que uma organização atinge, significa que os processos relacionados a este nível e aos demais níveis inferiores estão maduros. O progresso nos níveis de maturidade se dá quando uma organização satisfaz metas específicas e metas genéricas de um determinado nível de maturidade, e os demais níveis inferiores, para um conjunto de áreas de processos relacionado a ele(s).

O Quadro 2.1 mostra a relação entre os Níveis de Maturidade e Capacidade dos processos de software no CMMI:

Nível	Representação Contínua Níveis de Capacidade	Representação por Estágio Níveis de Maturidade
Nível 0	Incompleto	Não se Aplica
Nível 1	Executado	Inicial
Nível 2	Gerenciado	Gerenciado
Nível 3	Definido	Definido
Nível 4	Gerenciado quantitativamente	Gerenciado quantitativamente
Nível 5	Em otimização	Em otimização

Quadro 2.1 – Relação entre Capacidade e Maturidade no CMMI

2.3.2 MPS.BR

O Modelo de Referência MR-MPS, criado em 2003, objetiva a melhoria de processos do software brasileiro (SOFTEX, 2011). Ele se baseia nos conceitos de maturidade e capacidade de processos para a melhoria da qualidade dos produtos de software brasileiro, define níveis de maturidade que são uma combinação entre processos e sua capacidade.

A capacidade do processo avalia habilidade do processo para alcançar os objetivos de negócio tanto atuais quanto futuros, ou seja, ela expressa o quão refinado e institucionalizado está o processo que é executado em uma determinada organização. A capacidade está relacionada com o atendimento aos atributos de processo associados aos processos de cada nível de maturidade. À medida que a organização evolui nos níveis de maturidade, esta organização deve atingir um maior nível de capacidade parar desempenhar o processo.

Os níveis de maturidade estabelecem patamares de evolução de processos, caracterizando estágios de melhoria da implementação de processos nas organizações. Dado o nível de maturidade em que se encontra uma organização, é possível prever o desempenho futuro desta ao executar um ou mais processos. Ao todo são 7 (sete) níveis de maturidade, que em uma escala crescente ficariam dispostos da seguinte maneira: 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). A divisão em 7 estágios objetiva possibilitar uma implementação e avaliação adequada às micros, pequenas e médias empresas. O alcance de um determinado nível de maturidade se obtém quando são atendidos os propósitos e todos os resultados esperados dos respectivos processos e os resultados esperados dos atributos de processo estabelecidos para aquele nível, além dos demais níveis inferiores.

Pode-se fazer uma relação entre os dois modelos de qualidade de software, a equivalência entre maturidade e capacidade entre os modelos. Vale ressaltar que no modelo MPS-BR para cada nível de maturidade há um perfil de capacidade referente a este, o que não é exatamente igual no CMMI. Esta relação pode ser vista na Figura 2.3.

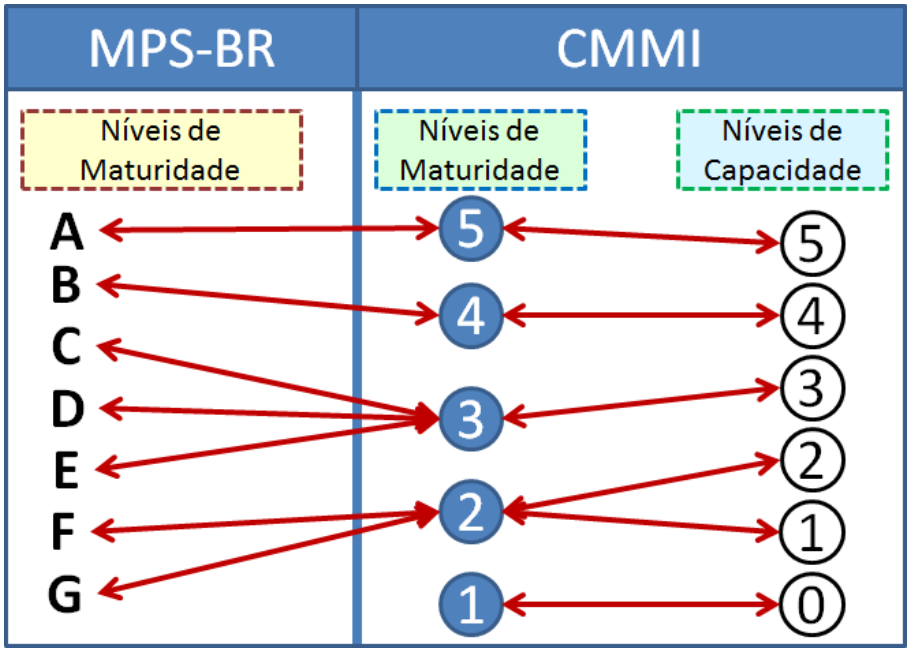


Figura 2.3 – Comparação entre os Modelos

2.4 EXECUÇÃO DE PROCESSOS

A partir das linguagens de modelagem de processo, pode-se realizar a modelagem, simulação, execução e avaliação de processos com o apoio de PSEEs (*Process-Centered Software Engineering Environments*) (REIS, 2003), conhecidos também como ADS (Ambientes de Desenvolvimento de Software) orientados ao processo.

A execução visa colocar em prática tudo aquilo que foi estimado para o processo de software. Com base na estimativa modelada anteriormente, os executores do projeto tem parâmetros para fazer o acompanhamento e manutenção do projeto.

2.4.1 Definição

A execução de processo de software é a fase do ciclo de vida de um processo de software em que as atividades modeladas são executadas. Esta execução se dá tanto pelos desenvolvedores quanto automaticamente por ferramentas de apoio (REIS, 2003). A execução ocorre quando um processo está pronto para ser executado (ou seja, o processo está instanciado) e leva em consideração a coordenação dos desenvolvedores, a interação entre as ferramentas e desenvolvedores, a garantia de que o processo está sendo executado conforme modelado, dentre outras questões (OLIVEIRA, 2006). Com o apoio de um ambiente de desenvolvimento de software, é possível obter resultados relevantes sobre o andamento da execução, além de possibilitar uma melhor gerência do desenvolvimento. De acordo com REIS (2003), apesar de não ser totalmente automatizada, a fase de execução depende de um mecanismo automatizado que compreenda o processo modelado e oriente os desenvolvedores no decorrer do seu trabalho, assim como execute automaticamente algumas tarefas, principalmente as tarefas repetitivas.

2.4.2 Contexto

O apoio automatizado a processos de software através de um PSEE mostrou-se bastante útil nas organizações (GARG e JAZAYERI, 1996), por disciplinar o desenvolvimento, aumentar a produtividade de grupos, permitir a coordenação das atividades de grandes equipes, e por organizar e disponibilizar a documentação do processo e de todos os produtos de software gerados.

Em OLIVEIRA (2007), apresenta-se o ciclo de vida do ImPProS (*Ambiente de Implementação Progressiva de Processo de Software*), com destaque para as fases mais focadas ao tratamento dos processos de software

no que tange aos modelos e normas da qualidade, conforme mostra a Figura 2.4.

O ciclo de vida tem início com a fase Análise das Características do Processo, que identifica as principais características da organização, de projetos e de produtos de software necessárias para a implementação do processo. Posteriormente, a fase Definição do Processo guia a implementação do processo a partir da execução de tarefas de acordo com os níveis em que este processo é usado em função da análise das características.

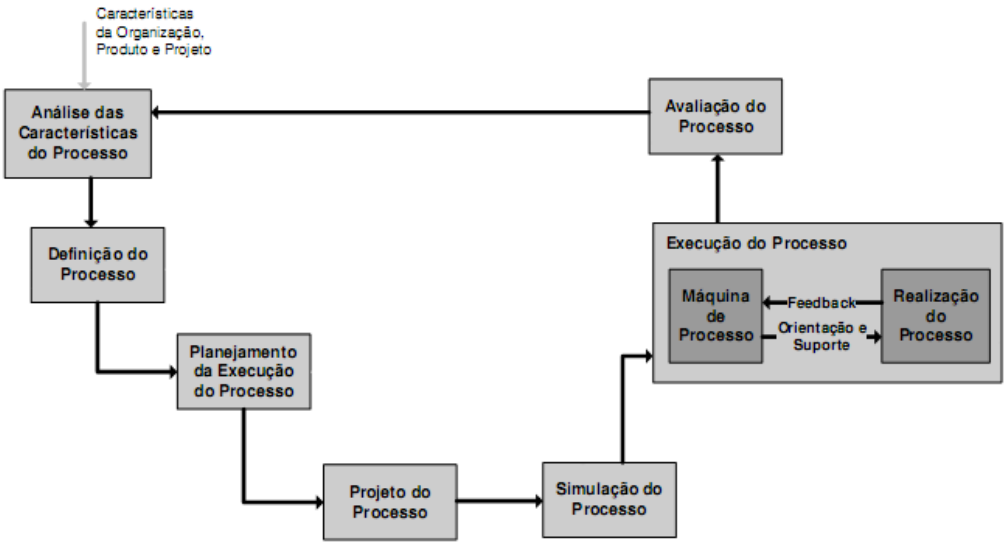


Figura 2.4 – Ciclo de Vida do ImPProS (Adaptado de OLIVEIRA, 2007)

A fase de Planejamento da Execução do Processo recebe um processo abstrato e gera um processo planejado, ou seja, com recursos e agentes alocados, cronograma definido, métricas e estimativas inferidas e previsão inicial do custo. Uma vez o processo planejado, o ciclo segue para a fase de Projeto do Processo, onde, a partir da publicação do processo, produz-se um processo planejado e modelado. Então, é possível que o processo passe pela fase de Simulação do Processo, onde a finalidade é identificar possíveis problemas no planejamento e/ou na definição do processo abstrato.

A próxima fase trata da Execução do Processo, que é apoiada pela máquina de processos, auxiliando na coordenação das atividades realizadas por pessoas (gerentes e equipe de desenvolvimento), e por ferramentas

automatizadas, obtendo, com isso, um *feedback* da realização do processo, a partir da organização das informações constantes no modelo de processo simulado, de acordo com a perspectiva de interesse na fase de execução. Paralelamente à fase de execução, ocorre a Avaliação do Processo que analisa a necessidade de modificação dinâmica do processo.

O foco deste trabalho está na fase de Execução do Processo, a partir do desenvolvimento de um ferramental de apoio à execução de processos, denominado Spider-PE.

2.5 CONSIDERAÇÕES FINAIS

Nos últimos anos, as organizações desenvolvedoras de software, tanto em consequência do aumento da concorrência quanto da necessidade de fornecer produtos e serviços de qualidade, vêm buscando se manter competitivas no mercado. Nesse contexto, estas organizações estão buscando cada vez mais a melhoria dos seus processos de desenvolvimento.

Com o surgimento de padrões e modelos internacionais bem como, recentemente, um modelo nacional de qualidade de software, os quais recomendam práticas que uma determinada organização pode adotar para desenvolver um sistema com qualidade. Muitas organizações estão procurando adaptar seus processos a estes modelos, a fim de melhorar continuamente a qualidade dos seus processos.

Neste contexto organizacional, podem-se ter vários projetos de desenvolvimento de software. Estes projetos adotam, para a execução, processos instanciados que possuem a descrição de como, quem e quando as atividades do processo serão executadas. Durante a execução do processo é possível, com o apoio de um ambiente de desenvolvimento de software, obter resultados relevantes acerca do andamento da execução além de possibilitar uma melhor gerência do desenvolvimento deste processo.

3 O *FRAMEWORK* DE EXECUÇÃO DE PROCESSOS

Este capítulo apresenta a principal contribuição deste trabalho, o *Framework* de Execução de Processos de Software denominado Spider-PE (*Process Enactment*). O *framework* é apresentado em detalhes a partir da apresentação de suas fases, atores e atividades.

Estas atividades que compõem o *Framework* estão relacionadas com os itens referentes à Capacidade dos modelos de melhoria de processo apresentados no Capítulo 2, CMMI-DEV e MR-MPS. As práticas sugeridas por estes modelos foram utilizadas como base para definição deste *framework*.

Neste capítulo, também, é apresentado o contexto no qual este trabalho está inserido, bem como as pesquisas relacionadas.

3.1 CONTEXTUALIZAÇÃO

Este trabalho é parte integrante das pesquisas realizadas neste no contexto do Projeto SPIDER – *Software Process Improvement: DEvelopment and Research* (OLIVEIRA *et al.*, 2011), localizado na Universidade Federal do Pará em parceria com o Centro de Informática da Universidade Federal de Pernambuco (CIn/UFPe), no que diz respeito à Melhoria de Processos de Software a partir da utilização de práticas recomendadas por Modelos de Qualidade.

O apoio à Execução de processos é o principal foco deste trabalho. Alguns dos principais objetivos desse mecanismo são: manter o estado da execução do processo consistente com o estado real da realização das tarefas; monitorar e coletar métricas sobre o processo.

Esses processos de software são definidos e executados por organizações inseridas em ambientes de negócios que, com frequência, sofrem modificações. Faz-se necessário que as organizações sejam capazes de melhorar continuamente os seus processos de software, a fim de produzir software com qualidade e de maneira competitiva.

3.1.1 Projeto SPIDER

O projeto SPIDER, institucionalizado em 2009 no Instituto de Ciências Exatas e Naturais (ICEN) da UFPA, possui como corpo integrante alunos: da Graduação da Faculdade de Computação, a partir dos cursos de Bacharelado em Ciência da Computação e Bacharelado em Sistemas de Informação; e do Mestrado do PPGCC - Programa de Pós-Graduação em Ciência da Computação. Este projeto tem, como um dos focos principais, apresentar um levantamento das ferramentas de software livre com características adequadas para possibilitar a criação de produtos de trabalhos (artefatos) derivados dos resultados esperados descritos nos objetivos das áreas de processo do modelo MPS.BR - Melhoria do Processo de Software Brasileiro, mantido pela SOFTEX - Associação para Promoção da Excelência do Software Brasileiro, o que evidencia a implementação do programa da qualidade organizacional (OLIVEIRA *et al.*, 2011).

3.1.2 Pesquisas Relacionadas

De acordo com os objetivos do Projeto SPIDER, pesquisas relacionadas às Tecnologias de Ambientes de Desenvolvimento de Software (ADS) Centrado em Processo vêm sendo desenvolvidas. Estes ambientes automatizam todas as atividades do ciclo de vida de desenvolvimento do software através da combinação de técnicas, métodos e ferramentas de modo a apoiar as disciplinas da Engenharia de Software durante o processo de

desenvolvimento (REIS, 2003). Estas pesquisas permitiram a instanciação do Ciclo de Vida de Processos do Spider, Figura 3.1, baseado no ciclo de vida do ImPProS, apresentado na subseção 2.4.2.

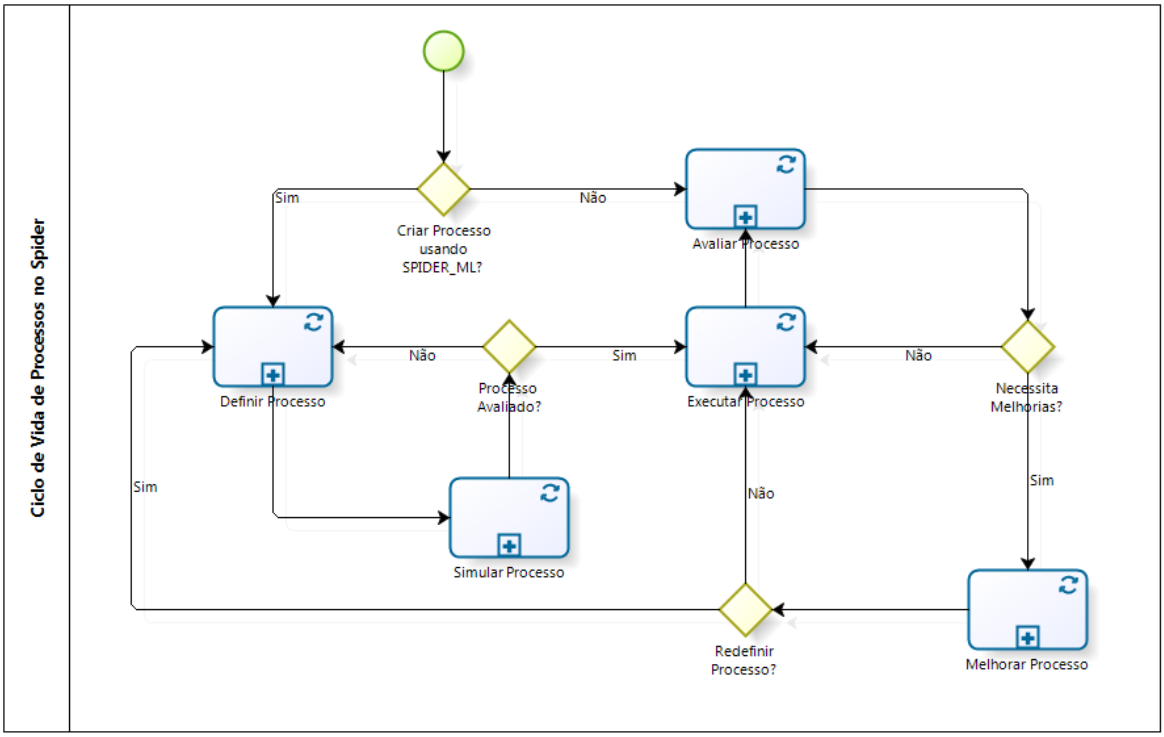


Figura 3.1 – Ciclo de Vida de Processos no Spider

Neste contexto, foi definida a linguagem de modelagem de processos SPIDER_ML (BARROS, 2010), baseada na linguagem padrão da OMG (*Object Management Group*), a SPEM(*Software ProcessEngineering Metal-Model*). Outra contribuição deste trabalho foi a ferramenta Spider-PM, responsável por sistematizar a modelagem de processos utilizando a linguagem SPIDER_ML. Esta pesquisa apoia a atividade Definir Processo (ver Figura 3.1).

Relacionada à atividade Simular Processo, foi proposta a ferramenta de Simulação de Processo de Software SPSM (CHAVES *et al.*, 2011), que utiliza a linguagem SPIDER_ML como base.

Conforme mencionado no inicio do capítulo 3, este trabalho está relacionado à atividade Executar Processo, sendo o *framework* aqui proposto parte de um Ferramental de Apoio Execução Flexível de Processos de

Software, denominado Spider-PE (PORTELA et. al., 2011). Além deste *framework*, este Ferramental é constituído de um Mapeamento entre os perfis de Capacidade dos modelos CMMI-DEV e MR-MPS, uma linguagem de execução de processos, denominada xSPIDER_ML, uma ferramenta de apoio a execução, denominada Spider-PE.

É importante ressaltar que, paralelamente a este trabalho, estão sendo realizadas pesquisas referentes às atividades Avaliar e Melhorar Processo.

3.2 ADERÊNCIA AOS MODELOS DE QUALIDADE

O *framework* foi desenvolvido de forma que seja aderente tanto ao modelo de qualidade CMMI-DEV quanto ao MR-MPS, no que diz respeito à capacidade de executar um processo. De acordo com a SOFTEX (2011), estes dois modelos são equivalentes entre si. O Guia de Implementação - Parte 11 do MPS.BR, que contém orientações para a implementação e avaliação do Modelo de Referência MR-MPS:2009 em conjunto com o CMMI-DEV v1.2, apresenta um mapeamento entre estes dois modelos.

Dentro deste contexto, o *framework* foi desenvolvido a fim de atender as práticas relacionadas à capacidade de processo destes modelos, pois a partir desta, se avalia o refinamento com quem as organizações executam seus processos. Assim sendo, uma organização que seja capaz de atender a capacidade dos modelos provavelmente possui o grau de maturidade necessário para implementar o nível de maturidade que pretende atingir.

Cada nível de maturidade do MR-MPS é uma combinação dos processos e da capacidade dos processos. A capacidade do processo, alvo deste trabalho, é representada por um conjunto de atributos descritos em termos de resultados esperados. Os atributos de processo (AP) são uma característica mensurável da capacidade do processo. O atendimento aos atributos do processo (AP), pelo atendimento aos resultados esperados dos

atributos do processo (RAP), é requerido para todos os processos no nível correspondente ao nível de maturidade.

A relação de cada atributo de processo deste nível está apresentada no Quadro 3.1.

AP	Descrição	Propósito	RAP
AP 1.1	O processo é executado	É uma medida do quanto o processo atinge o seu propósito	RAP 1
AP 2.1	O processo é gerenciado	É uma medida do quanto a execução do processo é gerenciada	RAP 2, RAP 3, RAP 4 RAP 5, RAP 6, RAP 7, RAP 8, RAP 9 e RAP 10
AP 2.2	Os produtos de trabalho do processo são gerenciados	É uma medida do quanto os produtos de trabalho produzidos pelo processo são gerenciados apropriadamente	RAP 11, RAP 12, RAP 13 e RAP 14

Quadro 3.1 – Atributos de Processo do Nível F (SOFTEX,2011)

Equivalente ao Nível F do MR-MPS, tem-se o Nível 2 (Gerenciado) do CMMI-DEV. Um nível de capacidade está relacionado a uma área de processo. Essas áreas de processo são constituídas de Objetivos Genéricos (GG – *Generic Goals*) e Práticas Genéricas (GP – *Generic Practices*) – relacionados a todos os processos e à organização. A função dos objetivos específicos é definir as características únicas que devem estar presentes para que uma determinada área de processo seja satisfeita. Por sua vez, os objetivos genéricos estão associados a mais de uma área de processo e definem as características que devem estar presentes para institucionalizar os processos que implementam a área de processo.

Uma prática genérica é a descrição de uma atividade considerada importante para a satisfação de um objetivo genérico. A relação de cada objetivo genérico de processo do Nível 2 está apresentada no Quadro 3.2.

Tanto os AP e RAP do Nível F do MR-MPS, apresentados no Quadro 3.1, quanto as GG e GP do Nível 2 do CMMI-DEV, apresentados no Quadro 3.2, estão referenciados nas tabelas que detalham as atividades do *framework*, na Subseção 3.3.3, devido as práticas recomendadas por estes servirem de base para a definição destas atividades.

GG	Descrição	Propósito	GP
GG 1	Alcançar Metas Específicas	O processo dá suporte e permite alcançar as metas especificas da área de processo transformando os produtos de trabalho identificáveis de entrada para produzir produtos de trabalho de identificáveis saída.	GP 1.1
GG 2	Institucionalizar um Processo Gerenciado	O processo é institucionalizado como um processo gerenciado.	GP 2.1, GP 2.2, GP 2.3, GP 2.4, GP 2.5, GP 2.6, GP 2.7, GP 2.8, GP 2.9 e GP 2.10

Quadro 3.2 – Objetivos Genéricos do Nível 2 (SEI, 2010)

Este trabalho adota como referência somente os AP correspondentes ao Nível F do MR-MPS e os GG do Nível 2 do CMMI-DEV, onde o Processo é considerado Gerenciado. A escolha destes níveis específicos deve-se ao fato de que eles são níveis iniciais dos modelos em questão e por isso tendem a ser de maior complexidade para implementação da maturidade organizacional (OLIVEIRA *et al.*, 2011). Além disto, nesses níveis o Processo dever ser institucionalizado, ou seja, significa colocá-lo em prática e, portanto, executá-lo.

Com base nestas justificativas e sendo a execução de processo o foco deste trabalho são estes os níveis considerados na definição do *framework*.

3.3 *FRAMEWORK* SPIDER-PE

O conceito de *framework* utilizado neste trabalho retrata a customização de um processo para seguir um ou mais modelos (SOUZA & OLIVEIRA, 2010). No contexto desse trabalho, as Metas e Práticas Genéricas do CMMI-DEV, os Atributos de Processo e Resultados de Atributos de Processo do MR-MPS e requisitos relacionados à execução de Processos deram origem as fases e atividades do *framework*, que podem ser customizadas de acordo com o perfil da organização que irá adotá-lo.

Nas subseções a seguir serão detalhadas as fases, atores e atividades deste *Framework*. A modelagem foi feita utilizando a notação BPMN (*Business Process Modeling Notation*), que é uma notação gráfica para modelagem de processos que possui diversas notações para identificar cada um dos elementos do processo, como as atividades e atores.

3.3.1 Fases do *Framework*

Como forma de organizar a execução das atividades, o *framework* foi divido em 3 fases: Fase de Execução do Processo(Figura 3.2); Fase de Execução das Atividades do Processo (Figura 3.3); e Fase de Aplicação do Formalismo de Execução (Figura 3.4). Esta organização visa à sistematização da execução do processo e suas atividades. É importante ressaltar que estas fases não são consecutivas e sim que uma é executada dentro da outra, caracterizando-se, assim, como subfases.

A Fase de Execução de Processo não atua somente na execução, ela começa desde o Planejamento do processo (definindo cronograma, riscos,

atividades de monitoramento e controle, etc.). Uma vez a etapa de Planejamento devidamente realizada, será iniciada a etapa de Execução do processo, onde a Fase de Execução das Atividades do Processo está presente. Entretanto, a Execução de Processo é constituída de outras atividades de monitoramento e controle que serão melhores detalhadas na subseção 3.3.3.1.

A Fase de Execução das Atividades do Processo é responsável execução do processo propriamente dita. Nela são gerados produtos de trabalho decorrente da execução das atividades do processo, sendo responsável por gerenciar as configurações do processo, além de analisar aderência dos produtos de trabalho e processos executados ao que foi estimado para o processo em execução. Logo que se inicia Fase de Execução das Atividades do Processo, a primeira etapa deste consiste em executar a Fase de Aplicação do Formalismo de Execução. Uma vez terminada esta etapa segue executando todas as sua demais atividades, conforme descrito na subseção 3.3.3.2.

A Fase de Aplicação do Formalismo de Execução por sua vez, aplica a hierarquia de ativos do processo, controla as variáveis do projeto, monitora estados e tempos do processo, realiza *feedback*, entre outros. Todas estas atividades são realizadas durante a execução do processo e são descritas na subseção 3.3.3.3.

As Figuras 3.2, 3.3 e 3.4 mostram o ciclo de vida das fases descritas anteriormente. Vale ressaltar que as atividades em verde indicam que aquela determinada atividade está referenciando uma determinada fase, apresentada na Figura seguinte.

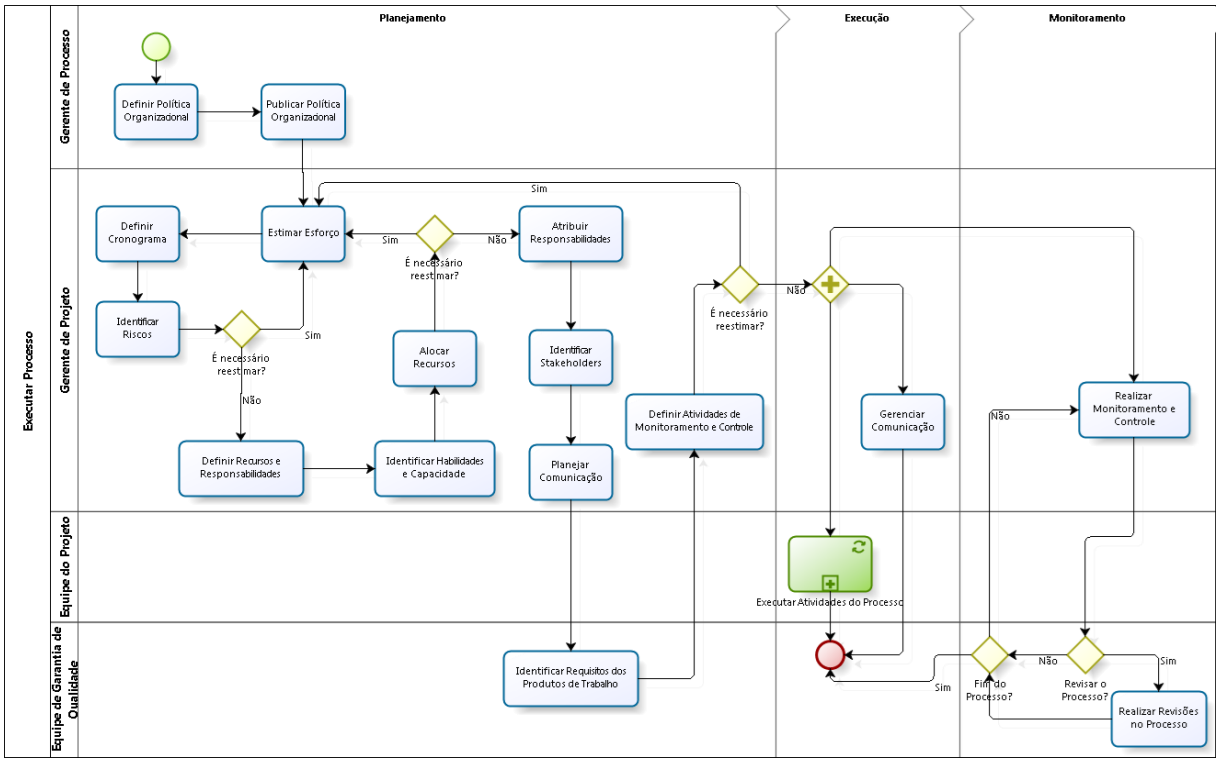


Figura 3.2 – Fase de Execução do Processo

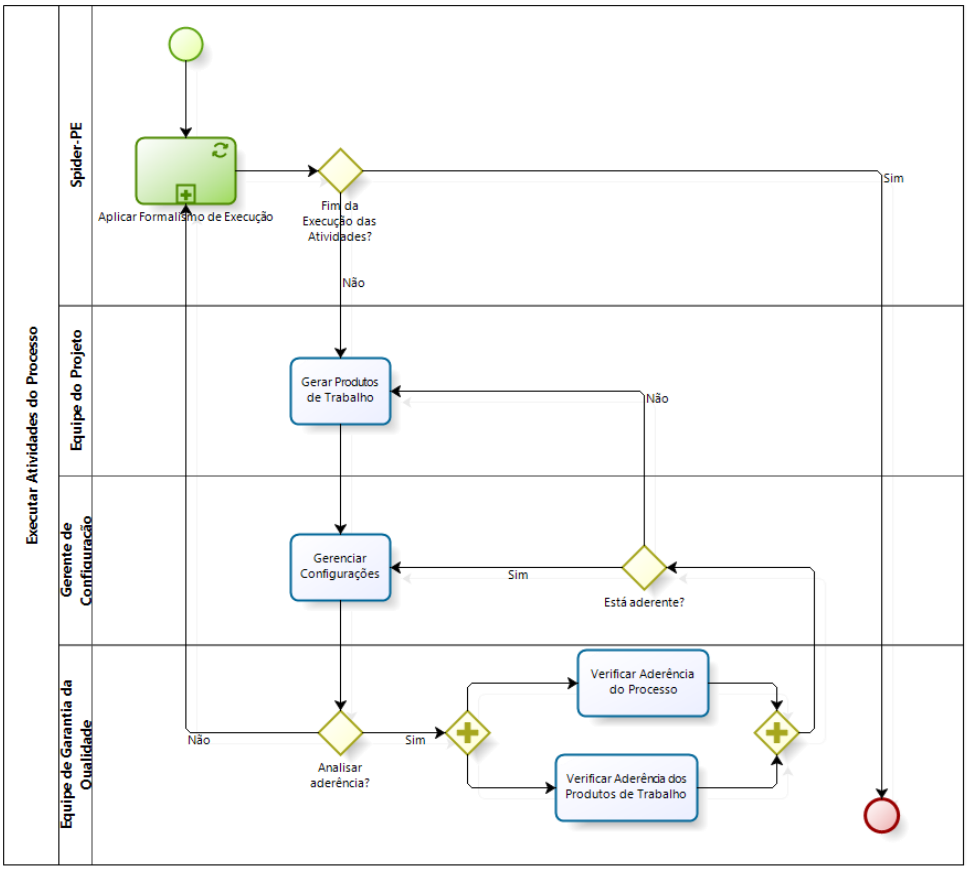


Figura 3.3 – Fase de Execução das Atividades do Processo

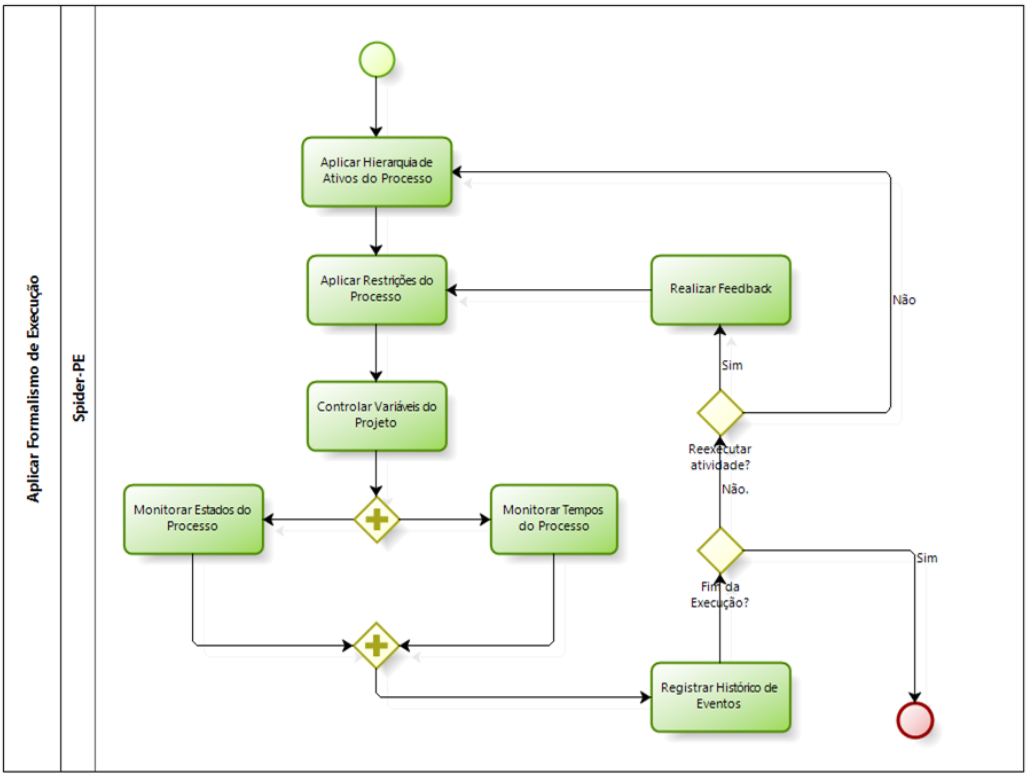


Figura 3.4 – Fase de Aplicação do Formalismo de Execução

3.3.2 Atores do *Framework*

Os atores do *framework* estão divididos de acordo com as funções exercidas dentro do processo e estão divididos nas seguintes funções:

- Gerente de Processo – responsável pela modelagem, execução, visualização, instanciação e acompanhamento dos eventos da execução do processo.

- Gerente de Projeto – responsável por alocar recursos para o projeto, ajuste das prioridades, coordenação das interações com cliente. O gerente de projeto também estabelece um conjunto de práticas que garantem a integridade e a qualidade dos artefatos do projeto.

- Equipe do Projeto – responsável pelas atividades de *design* e de implementação para diferentes aspectos do projeto. Codificam os componentes do software baseado na documentação elaborada para o mesmo, são responsáveis por fazer os testes de cada componente;

- Equipe de Garantia da Qualidade – responsável por identificar requisitos dos produtos de trabalho, verificar aderência do processo e produtos de trabalho. Também é responsável por realizar revisões no processo. Em outras palavras, está encarregada de planejar, gerenciar e controlar a qualidade do software produzido;

- Gerente de Configuração – responsável por preparar e oferecer suporte a um ambiente de gerência de configuração do projeto para a Equipe do Projeto. Ele deve assegurar que este ambiente permita que todos os artefatos possam ser incluídos, versionados e controlados na unidade de implantação, conforme necessário;

- Spider-PE – ferramenta responsável pela execução do processo de software, aplicando hierarquia no processo, gerenciando o acesso dos atores e a realização das tarefas por estes atores.

3.3.3 Descrição das Atividades

As atividades do *framework* serão descritas nas subseções a seguir, de acordo com as suas respectivas Fases. É importante ressaltar que cada uma destas atividades é derivada dos itens relacionados à Capacidade dos Modelos de Qualidade, descritos na subseção 3.2. Sendo assim, as células Itens dos Modelos e Nível de Aderência das tabelas de detalhamento, mostrarão esta

relação, onde para os itens, serão referenciados os RAP do MR-MPS e GP do CMMI-DEV, dos quais a atividade é derivada e o nível de aderência entre estas atividades do *framework* e os itens dos modelos, a partir do estabelecimento de 3 níveis:

a. **Totalmente Aderente:** As atividades do *Framework* são totalmente aderentes às exigências dos modelos de qualidade;

b. **Parcialmente Aderente:** As atividades do *Framework* contemplam parcialmente as exigências dos modelos de qualidade;

c. **Não Aderente:** As atividades do *Framework* não são aderentes às exigências dos modelos de qualidade.

São apresentados protótipos de tela para cada uma das atividades, com objetivo de validar o *framework* proposto neste trabalho, a partir do desenvolvimento de uma ferramenta de software livre denominada Spider-PE. Esta ferramenta encontra-se em desenvolvimento no Projeto SPIDER e caracteriza-se como trabalho futuro desta pesquisa.

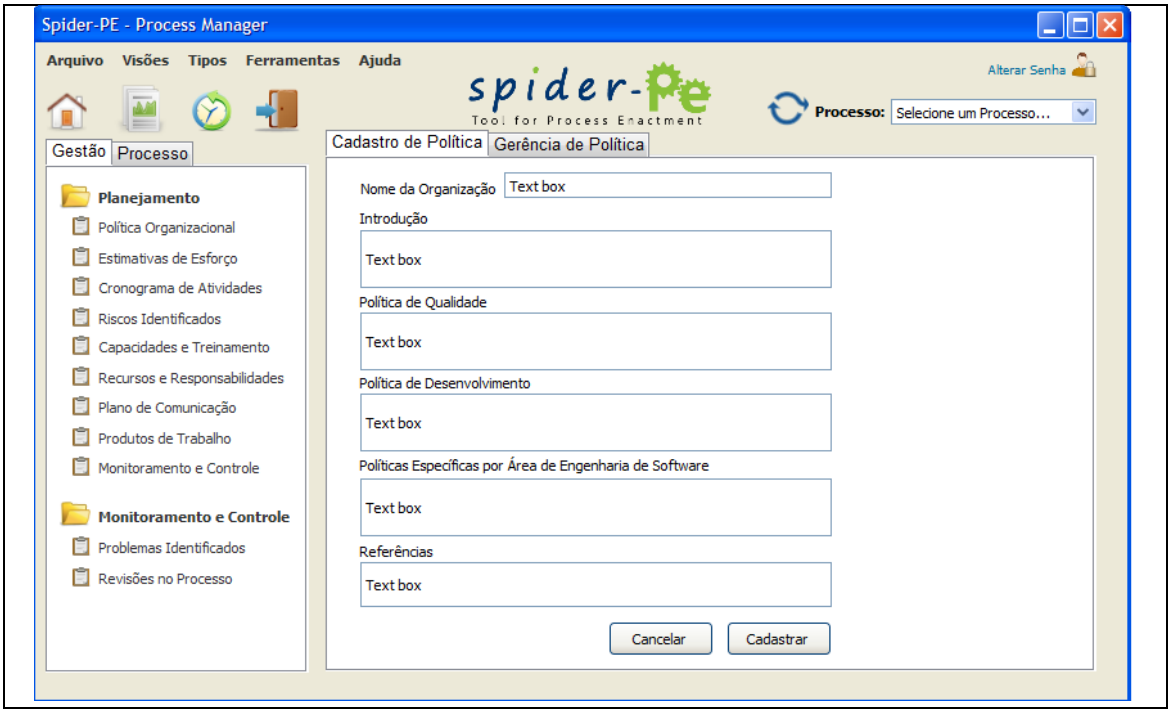
3.3.3.1 Fase de Execução do Processo

Esta fase é responsável por preparar o processo para sua execução, bem como realizar monitoramento e revisão durante sua execução, a fim de mitigar e conter problemas e não-conformidades que podem surgir no decorrer deste. Suas atividades estão divididas da seguinte maneira: Definir Política Organizacional; Publicar Política Organizacional; Estimar Esforço; Definir Cronograma; Identificar Riscos; Definir Recursos e Responsabilidades; Identificar Habilidades e Capacidade; Alocar Recursos; Atribuir Responsabilidades; Identificar *Stakeholders*; Planejar Comunicação; Identificar Requisitos dos Produtos de Trabalho; Definir Atividades de Monitoramento e Controle; Gerenciar Comunicação; Realizar Monitoramento e

Controle e Realizar Revisões no Processo. Os quadros a seguir descrevem de cada uma das atividades acima listadas.

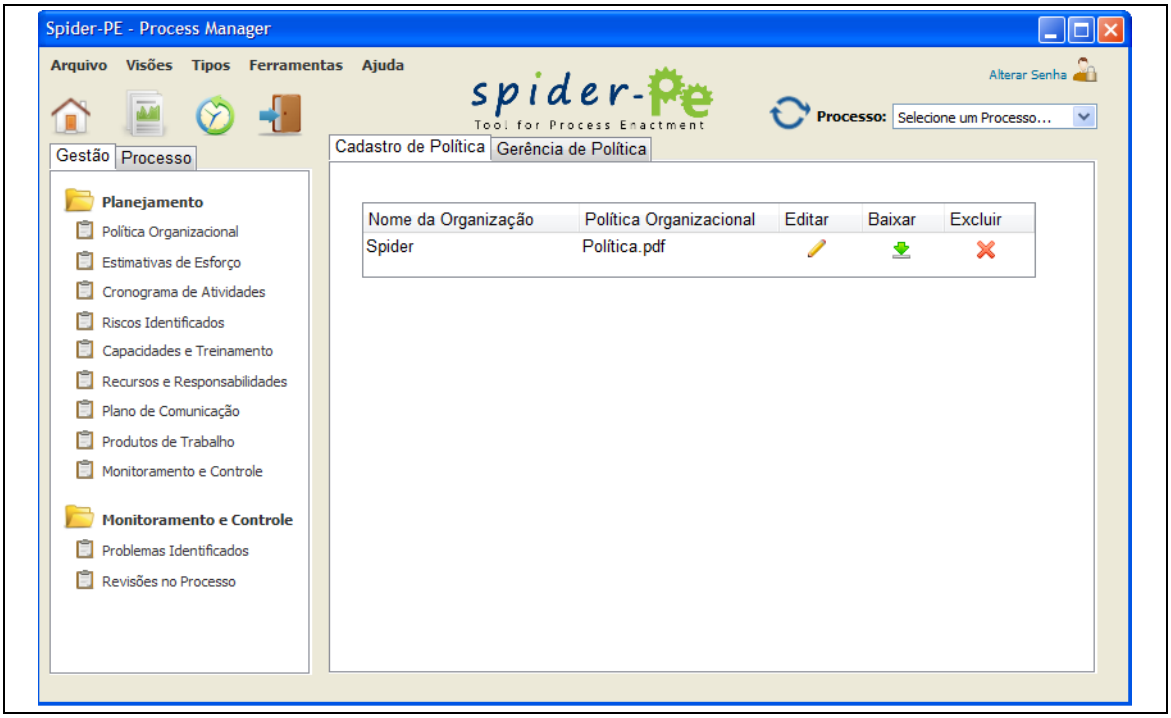
a) Definir Política Organizacional

Itens dos Modelos:	**Nível de Aderência:**
RAP 2 / GP 2.1	Totalmente Aderente.
Objetivo:	
Especificar uma Política de Desenvolvimento de Software para a Organização, a partir da definição de diretrizes, responsabilidades e políticas específicas.	
Critérios de Entrada	**Artefatos de Entrada**
Processo Definido ou em Definição.	Processo e/ou Necessidades/Objetivos Organizacionais.
Passos	
• Definir Diretrizes Gerais e Responsabilidades Gerais; • Definir Políticas de Qualidade, Desenvolvimento e Políticas Específicas por Área de Engenharia de Software;	
Critérios de Saída	**Artefatos de Saída**
Política Organizacional definida.	Política Organizacional.
Responsáveis/Pessoas Envolvidas	
Gerente de Processo.	
Templates	
Template de Política Organizacional.	
Ferramentas de Apoio Utilizadas	
Editor de Texto ou Ferramenta de Execução de Processos.	
Protótipo de Tela Correspondente	



b) Publicar Política Organizacional

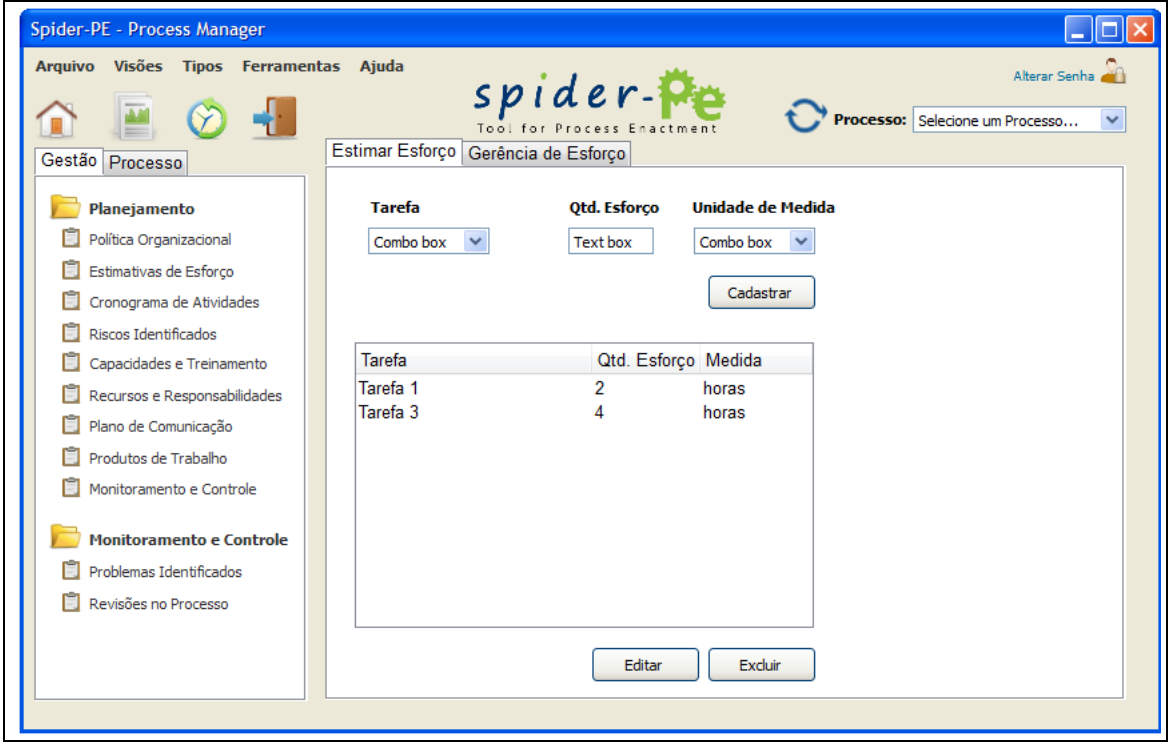
Itens dos Modelos:	**Nível de Aderência:**
RAP 1 / GP 2.1	Totalmente Aderente.
Objetivo	
Publicar e disponibilizar a Política Organizacional a todos os *stakeholders*.	
Critérios de Entrada	**Artefatos de Entrada**
Possuir uma Política Organizacional definida.	Política Organizacional.
Passos	
• Disponibilizar a Política Organizacional em meio adequado; • Comunicar e disseminar a todos os *stakeholders*;	
Critérios de Saída	**Artefatos de Saída**
A Política Organizacional deve está disponível a todos os interessados.	Não se aplica.
Responsáveis/Pessoas Envolvidas	
Gerente de Processo.	
Templates	
Não se aplica.	
Ferramentas de Apoio Utilizadas	
Site na Intranet da Organização e/ou Ferramenta de Execução de Processos.	
Protótipo de Tela Correspondente	



c) Estimar Esforço

Itens dos Modelos:	**Nível de Aderência:**
RAP 3 / GP 2.2	Totalmente Aderente.
Objetivo	
Estimar o esforço necessário para realizar cada tarefa do processo.	
Critérios de Entrada	**Artefatos de Entrada**
O escopo operacional do projeto (tarefas do processo) deve está definido.	Plano de Projeto, Lista de Requisitos e Casos de Uso.
Passos	
• Escolher a técnica de estimativa de tamanho do projeto; • Aplicar a técnica de estimativa de tamanho escolhida; • Escolher a técnica de estimativa de esforço que mais se adeque ao projeto; • Calcular o esforço necessário para realizar as tarefas do projeto segundo a técnica escolhida;	
Critérios de Saída	**Artefatos de Saída**
Estimativas de esforço para o projeto realizadas.	Plano de Projeto atualizado.
Responsáveis/Pessoas Envolvidas	
Gerente de Projeto.	
Templates	

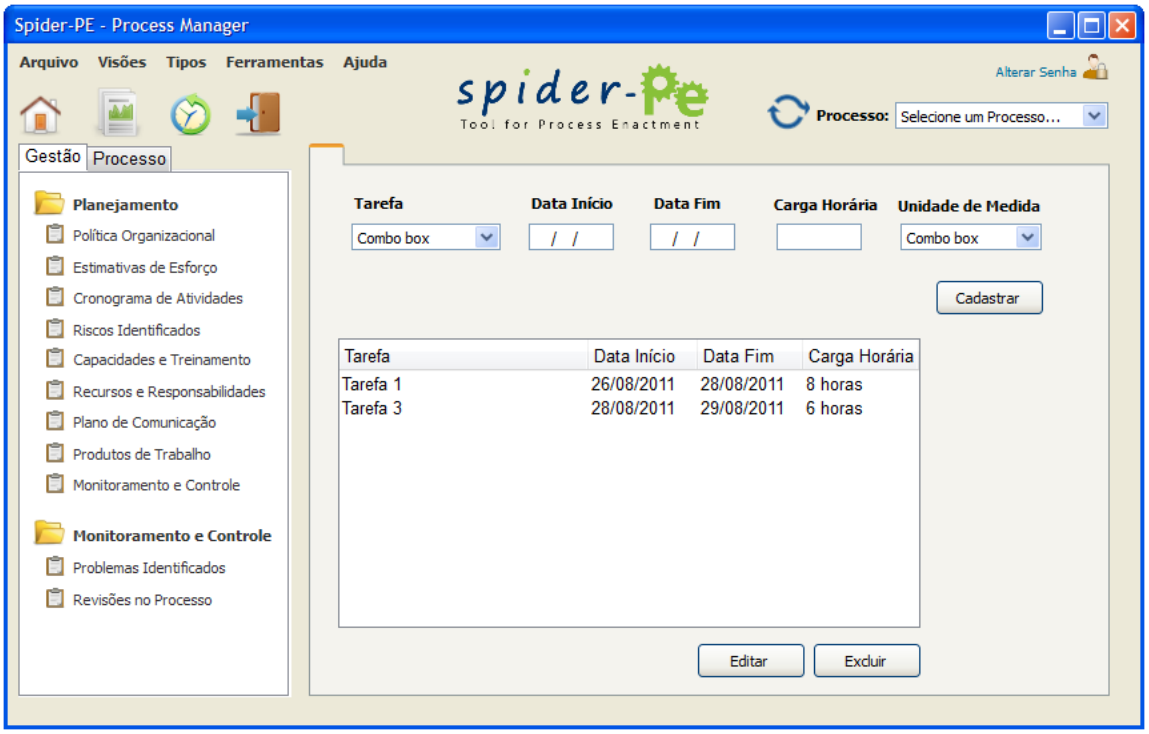
Não se aplica.
Ferramentas de Apoio Utilizadas
Spider-APF ou Spider-UCP e Spider-Cocomo.
Protótipo de Tela Correspondente



d) Definir Cronograma

Itens dos Modelos:	**Nível de Aderência:**
RAP 3 / GP 2.2	Totalmente Aderente.
Objetivo	
Estimar o tempo esperado para início e término de realização das tarefas do processo.	
Critérios de Entrada	**Artefatos de Entrada**
Ter definido o esforço necessário para execução das tarefas do processo.	Plano de Projeto.
Passos	
• Analisar a relação entre o esforço e o tempo para execução de cada tarefa; • Definir os marcos e pontos de controle para a execução do processo; • Baseado na análise do esforço gasto para execução das tarefas, estimar a data de início e término previsto para cada tarefa;	
Critérios de Saída	**Artefatos de Saída**
Cronograma definido.	Plano de Projeto com Cronograma

	definido.
Responsáveis/Pessoas Envolvidas.	
Gerente de Projeto.	
Templates	
Não se aplica.	
Ferramentas de Apoio Utilizadas	
Ferramenta de Execução de Processos.	
Protótipo de Tela Correspondente	



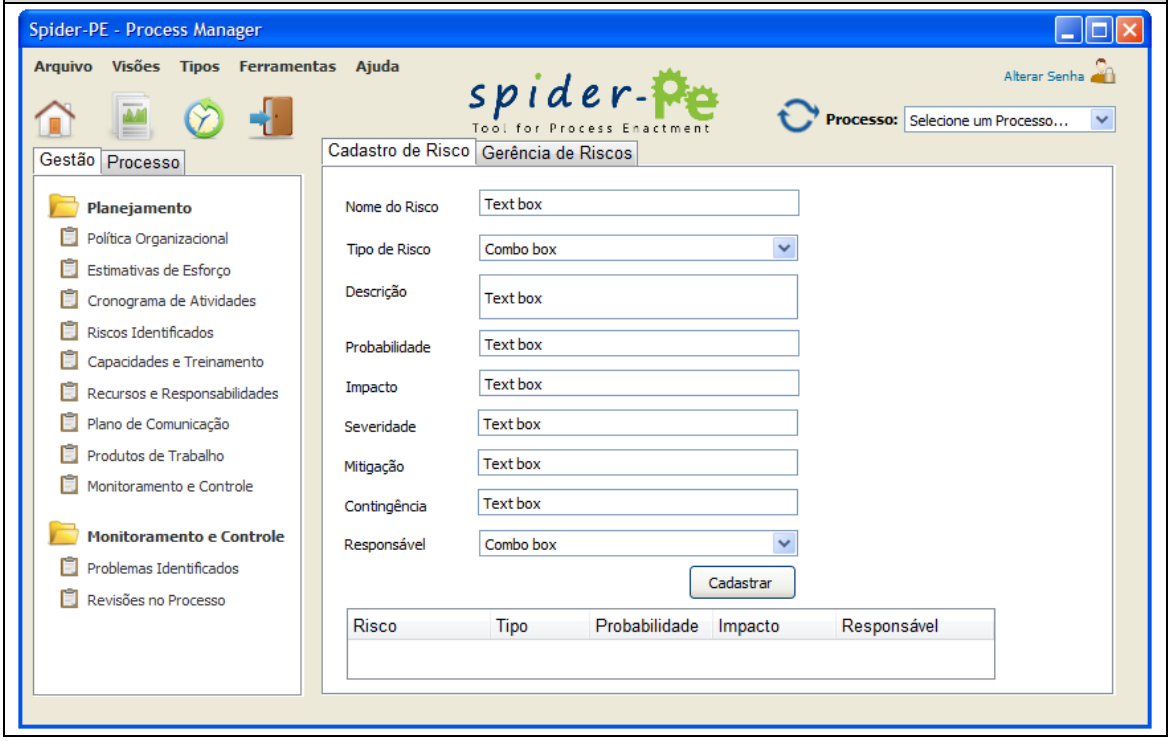
e) Identificar Riscos

Itens dos Modelos:	**Nível de Aderência:**
RAP 3 / GP 2.2	Totalmente Aderente.
Objetivo	
Identificar e cadastrar os riscos inerentes à execução do processo.	
Critérios de Entrada	**Artefatos de Entrada**
Analisar o Plano de Projeto e a Base Histórica.	Plano de Projeto.
Passos	
• Deve-se analisar a Base Histórica e o Plano de Projeto a fim de identificar possíveis riscos; • Cadastrar os riscos identificados; • Caso seja encontrado algum risco durante o desenvolvimento do projeto	

que não foi identificado previamente, este risco deve ser adicionado à lista de riscos, contendo sua severidade, probabilidade de ocorrência e prioridade;

- Um plano de mitigação deve ser definido, para prevenir a ocorrência de riscos identificados;
- Caso um risco previsto ocorra, uma medida de contingência deve ser tomada;

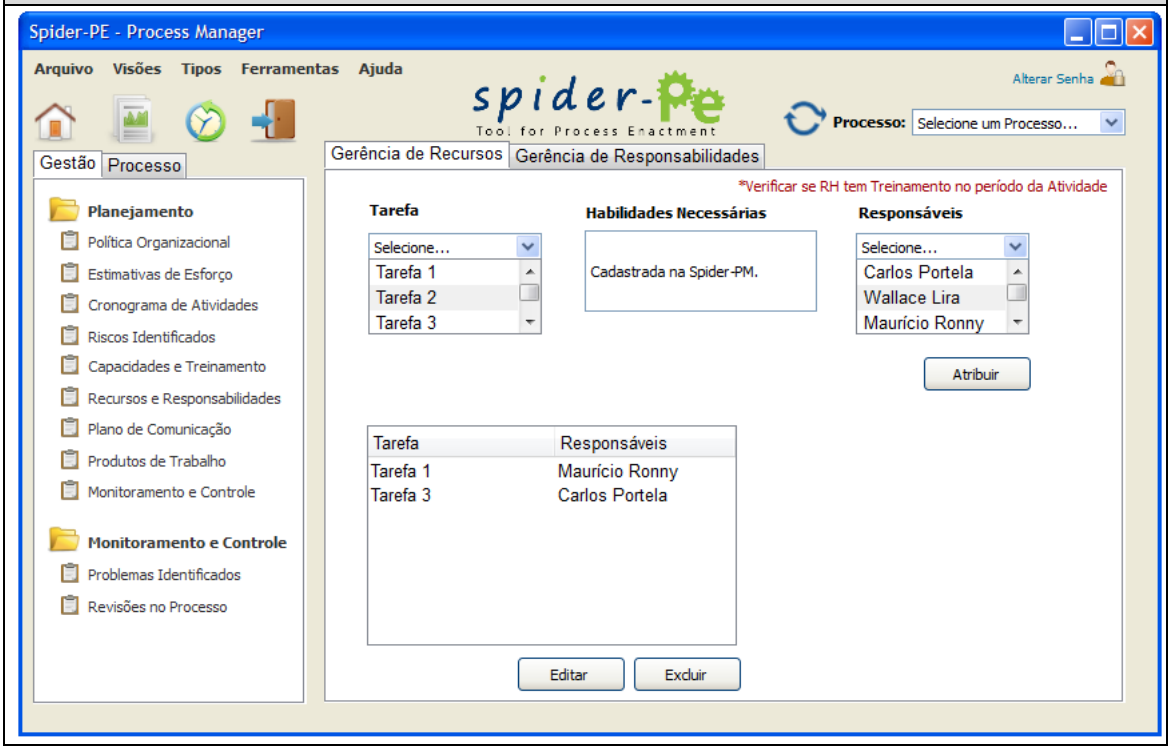
Critérios de Saída	**Artefatos de Saída**
Riscos do Projeto identificados.	Plano de Projeto com Lista de Riscos cadastrada.
Responsáveis/Pessoas Envolvidas	
Gerente de Projeto.	
Templates	
Template de Lista de Riscos.	
Ferramentas de Apoio Utilizadas	
Ferramenta de Execução de Processos.	
Protótipo de Tela Correspondente	



f) Definir Recursos e Responsabilidades

Itens dos Modelos:	**Nível de Aderência:**
RAP 3; RAP 5 / GP 2.2; GP 2.3	Totalmente Aderente.
Objetivo	
Deve-se identificar quais os recursos (hardware e software) e as	

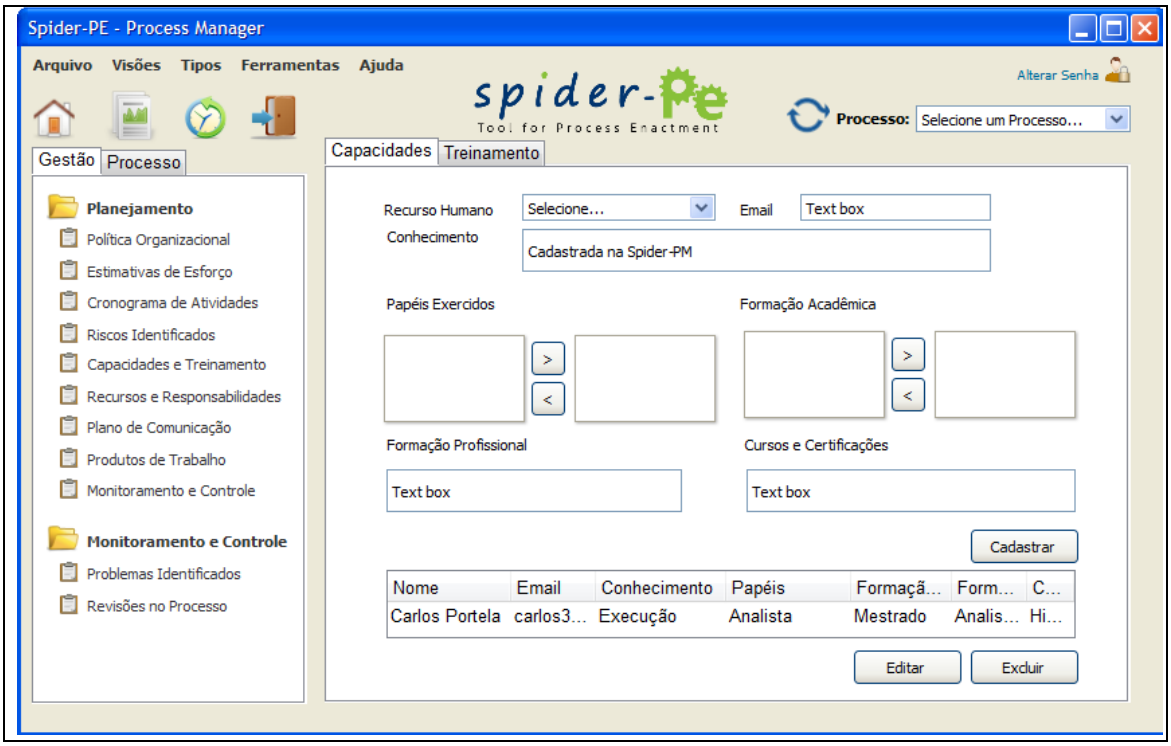
responsabilidades (recursos humanos) necessárias para realizar uma determinada tarefa.	
Critérios de Entrada	**Artefatos de Entrada**
Escopo definido e Esforço mensurado.	Plano do Projeto.
Passos	
• Definir as responsabilidades necessárias para executar o processo; • Definir recursos a serem utilizados em cada tarefa, levando-se em consideração o esforço estimado para realização destas;	
Critérios de Saída	**Artefatos de Saída**
Lista com Recursos Necessários para realizar as Atividades e a Lista de Responsabilidades do Projeto definidas.	Plano de Projeto com Lista de Recursos e Lista de Responsabilidades cadastradas.
Responsáveis/Pessoas Envolvidas	
Gerente de Projeto.	
Templates	
Não se aplica.	
Ferramentas de Apoio Utilizadas	
Spider-PM.	
Protótipo de Tela Correspondente	



g) Identificar Habilidades e Capacidade

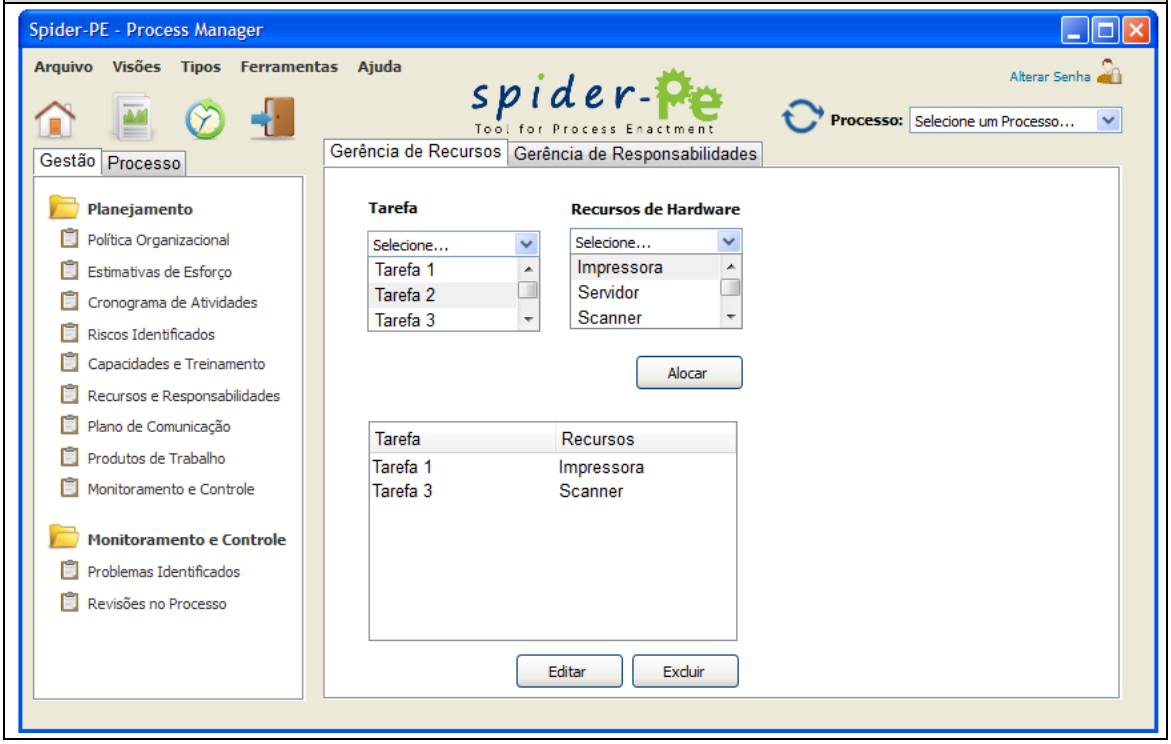
Itens dos Modelos:	Nível de Aderência:

RAP 7 / GP 2.5	Totalmente Aderente.
Objetivo	
Cadastrar pessoas com suas capacidades e habilidades para atender as responsabilidades previamente definidas a fim de verificar quais estão aptos para executar determinadas tarefas. Nesta etapa deve-se também verificar se a pessoa necessita ou não de treinamento.	
Critérios de Entrada	**Artefatos de Entrada**
Verificar os projetos anteriores em que os membros da equipe participaram, bem como seus currículos e treinamentos realizados.	Currículo dos membros da Equipe e Plano de Cargos e Funções.
Passos	
• Analisar os currículos dos membros da equipe e os perfis de desenvolvimento da organização; • Entrevistar os membros da equipe; • Identificar habilidades e capacidades de cada membro da equipe; • Verificar a necessidade de treinamento;	
Critérios de Saída	**Artefatos de Saída**
Capacidade e Habilidades identificadas.	Plano de Projeto atualizado com Recursos Humanos cadastrados.
Responsáveis/Pessoas Envolvidas	
Gerente de Projeto.	
Templates	
Template de Lista de Recursos Humanos do Projeto.	
Ferramentas de Apoio Utilizadas	
Spider-PM.	
Protótipo de Tela Correspondente	



h) Alocar Recursos

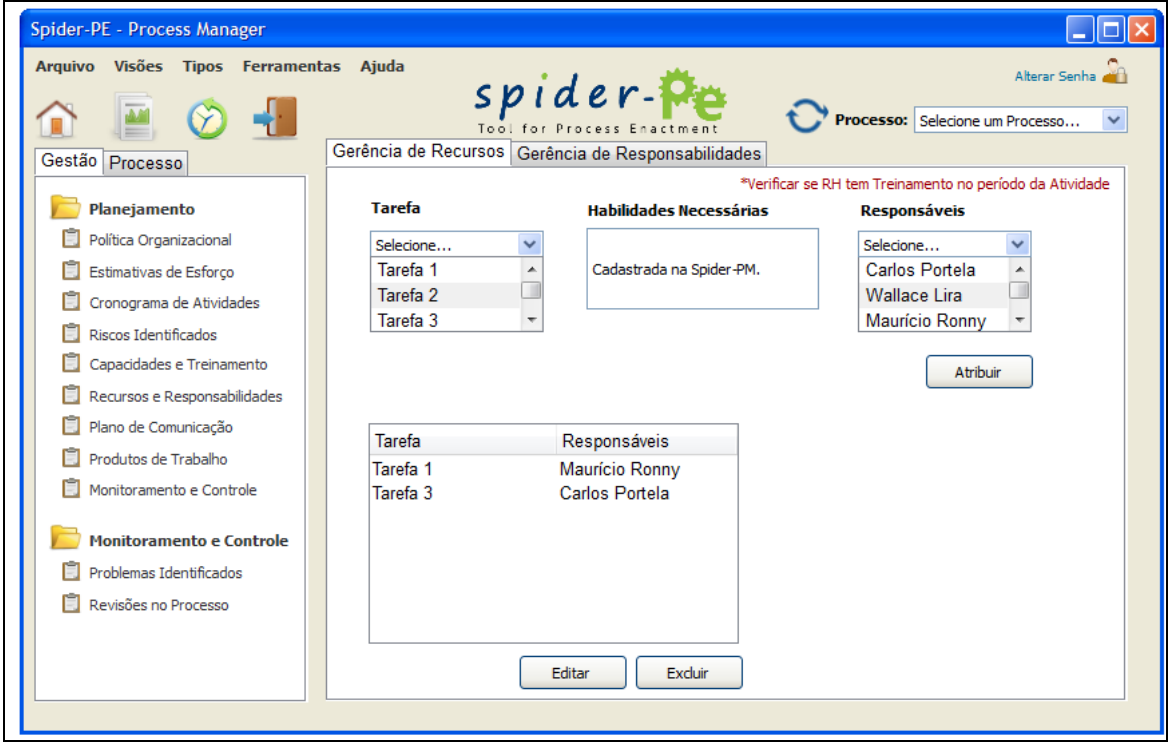
Itens dos Modelos:	**Nível de Aderência:**
RAP 5 / GP 2.3	Totalmente Aderente.
Objetivo	
Visa à distribuição dos recursos (hardware e software) disponíveis para a realização do projeto.	
Critérios de Entrada	**Artefatos de Entrada**
Recursos Necessários para realização das Atividades definidos.	Plano de Projeto.
Passos	
• Verificar o esforço estimado para realizar as tarefas; • Com base no esforço e disponibilidade dos recursos, alocar os recursos de acordo com o esforço estimado para realização das atividades;	
Critérios de Saída	**Artefatos de Saída**
Recursos alocados para cada tarefa.	Plano de Projeto atualizado.
Responsáveis/Pessoas Envolvidas	
Gerente de Projeto.	
Templates	
Não se aplica.	
Ferramentas de Apoio Utilizadas	
Ferramenta de Execução de Processos.	

Protótipo de Tela Correspondente


i) Atribuir Responsabilidades

Itens dos Modelos:	Nível de Aderência:
RAP 6 / GP 2.4	Totalmente Aderente.
Objetivo	
Atribuir aos recursos humanos responsabilidades no projeto levando em consideração o perfil de cada um destes.	
Critérios de Entrada	**Artefatos de Entrada**
Recursos e papéis definidos, capacidade e habilidade dos recursos humanos identificadas.	Plano de Cargos e Funções e Plano de Projeto.
Passos	
• Verificar os Papéis necessários para executar o projeto; • Analisar as Habilidades e Capacidades dos Recursos Humanos para definir os Papéis; • Alocar os membros da equipe às responsabilidades definidas em cada tarefa;	
Critérios de Saída	**Artefatos de Saída**
Responsabilidades atribuídas.	Plano de Projeto atualizado.
Responsáveis/Pessoas Envolvidas	
Gerente de Projeto.	
Templates	

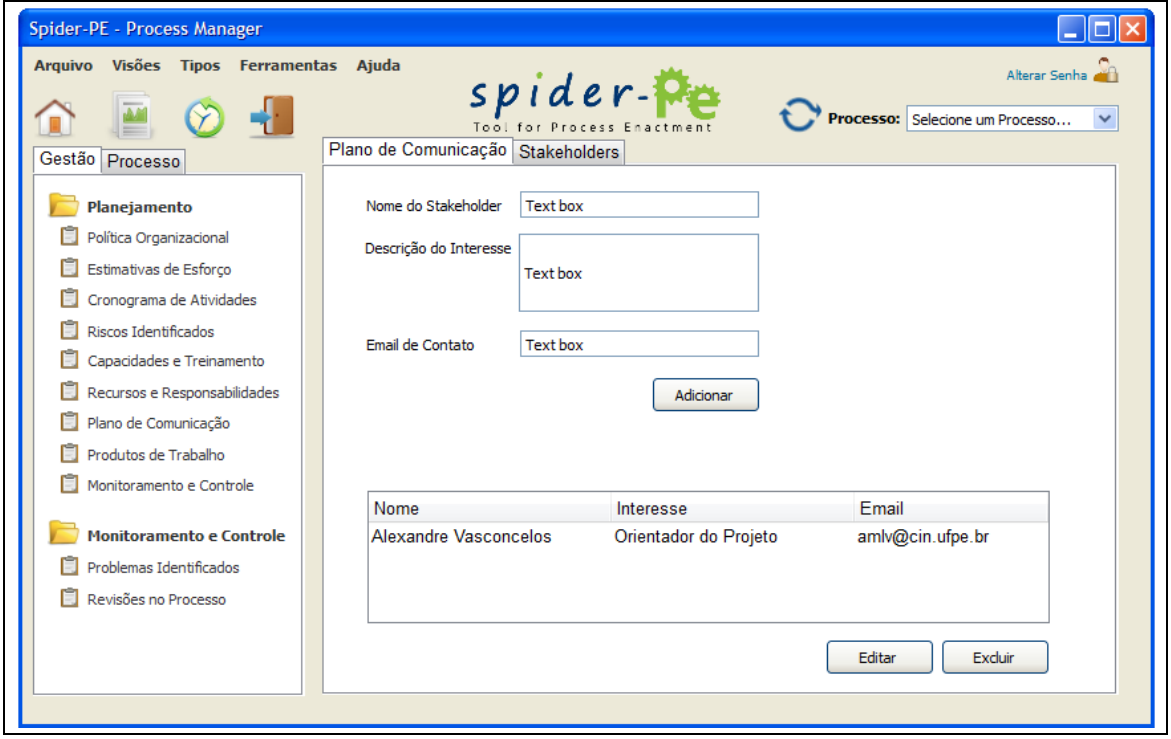
Não se aplica.
Ferramentas de Apoio Utilizadas
Ferramenta de Execução de Processos.
Protótipo de Tela Correspondente



j) Identificar *Stakeholders*

Itens dos Modelos:	**Nível de Aderência:**
RAP 8 / GP 2.7	Totalmente Aderente.
Objetivo	
Cadastrar todos os interessados afetados pelo projeto.	
Critérios de Entrada	**Artefatos de Entrada**
Plano de Projeto definido.	Plano de Projeto.
Passos	
• Identificar todos os interessados no projeto; • Registrar o contato dos interessados;	
Critérios de Saída	**Artefatos de Saída**
Identificar todos os interessados.	Plano de Projeto atualizado.
Responsáveis/Pessoas Envolvidas	
Gerente de Projeto.	
Templates	
Não se aplica.	

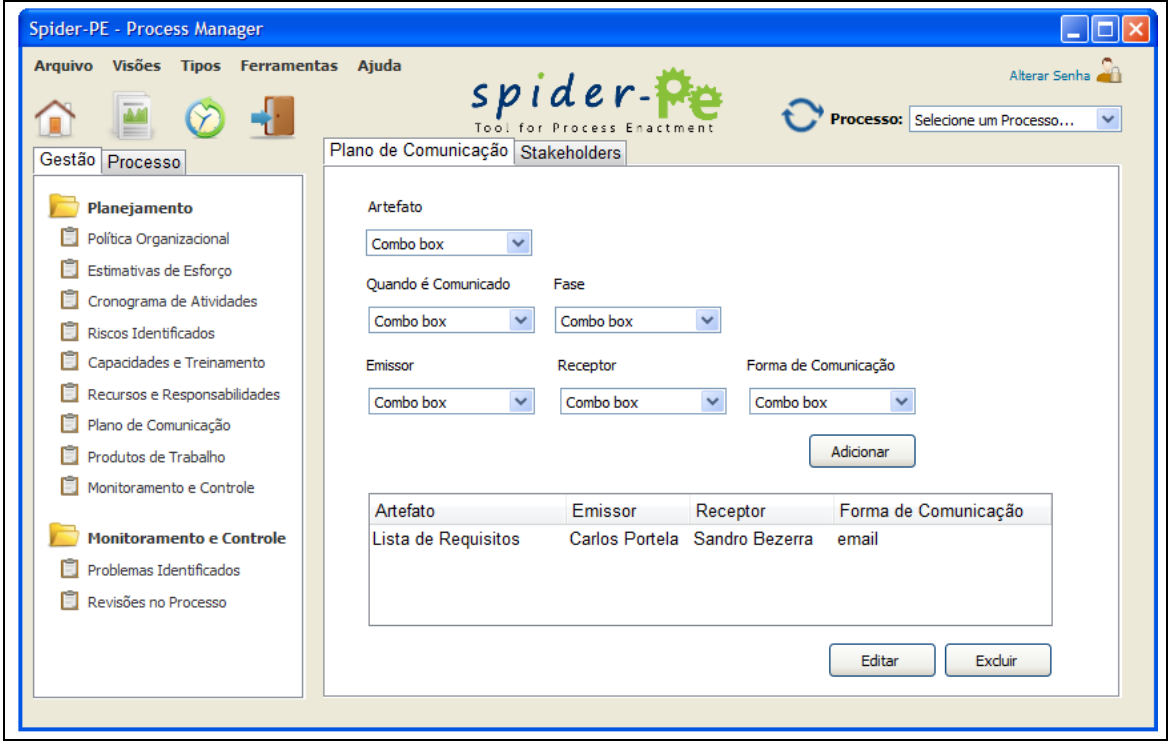
Ferramentas de Apoio Utilizadas
Ferramenta de Execução de Processos.
Protótipo de Tela Correspondente



k) Planejar Comunicação

Itens dos Modelos:	Nível de Aderência:
RAP 8 / GP 2.7	Totalmente Aderente.
Objetivo	
Fazer com que todos os interessados no projeto tenham acesso às informações e documentos relevantes.	
Critérios de Entrada	**Artefatos de Entrada**
Todos os *Stakeholders* devem ser identificados bem como os artefatos relevantes gerados no projeto.	Plano do Projeto com Lista de *Stakeholders* e Lista de Artefatos.
Passos	
• Verificar quais artefatos precisam ser divulgados; • Definir forma e meios de comunicação; • Disponibilizar acesso aos artefatos;	
Critérios de Saída	**Artefatos de Saída**
O Plano de Comunicação deve está criado.	Planode Comunicação.
Responsáveis/Pessoas Envolvidas	

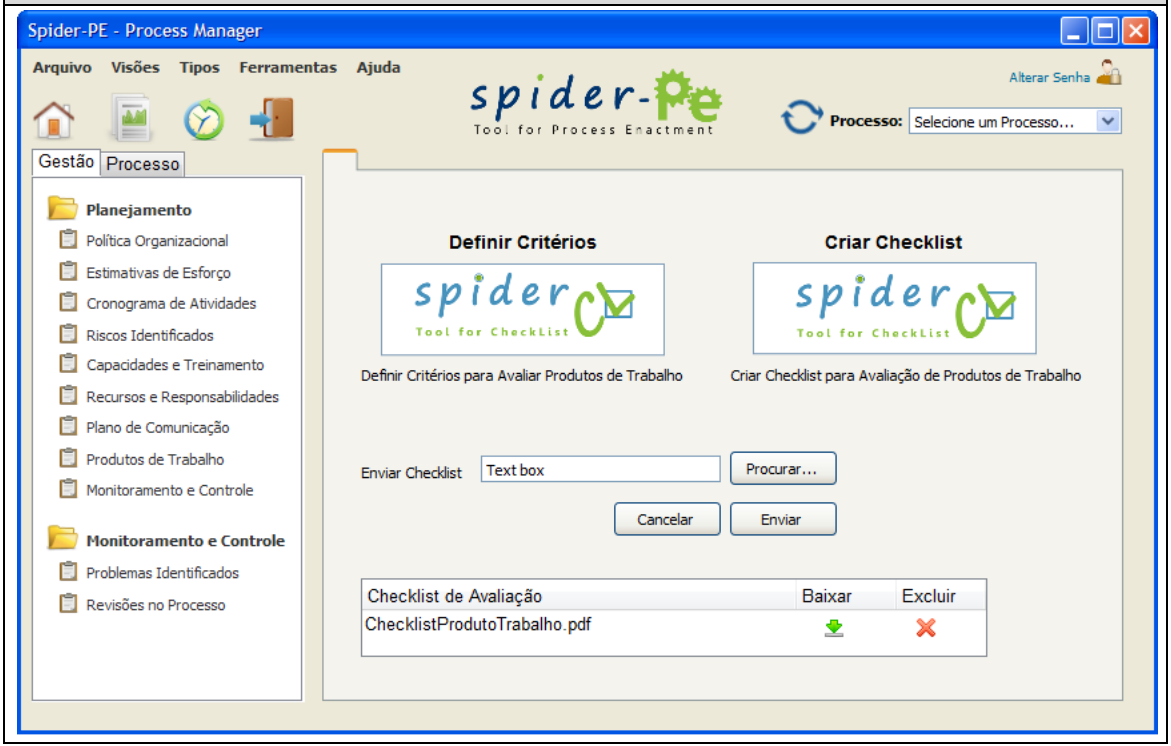
Gerente de Projeto.
Templates
Template de Plano de Comunicação.
Ferramentas de Apoio Utilizadas
Ferramenta de Execução de Processos.
Protótipo de Tela Correspondente



I) Identificar Requisitos dos Produtos de Trabalho

Itens dos Modelos:	**Nível de Aderência:**
RAP 11.	Totalmente Aderente.
Objetivo	
Definir os padrões, templates e demais atributos necessários para a produção de cada produto de trabalho durante o desenvolvimento do projeto.	
Critérios de Entrada	**Artefatos de Entrada**
Plano de Projeto estabelecido.	Não se aplica.
Passos	
• Definir requisitos para a manutenção das informações a serem geradas pelo projeto; • Associar os requisitos aos templates, padrões de documentação e critérios de qualidade relacionados a produtos de trabalho;	
Critérios de Saída	**Artefatos de Saída**
Requisitos dos Produtos de Trabalho	Padrões e Templates de Produtos

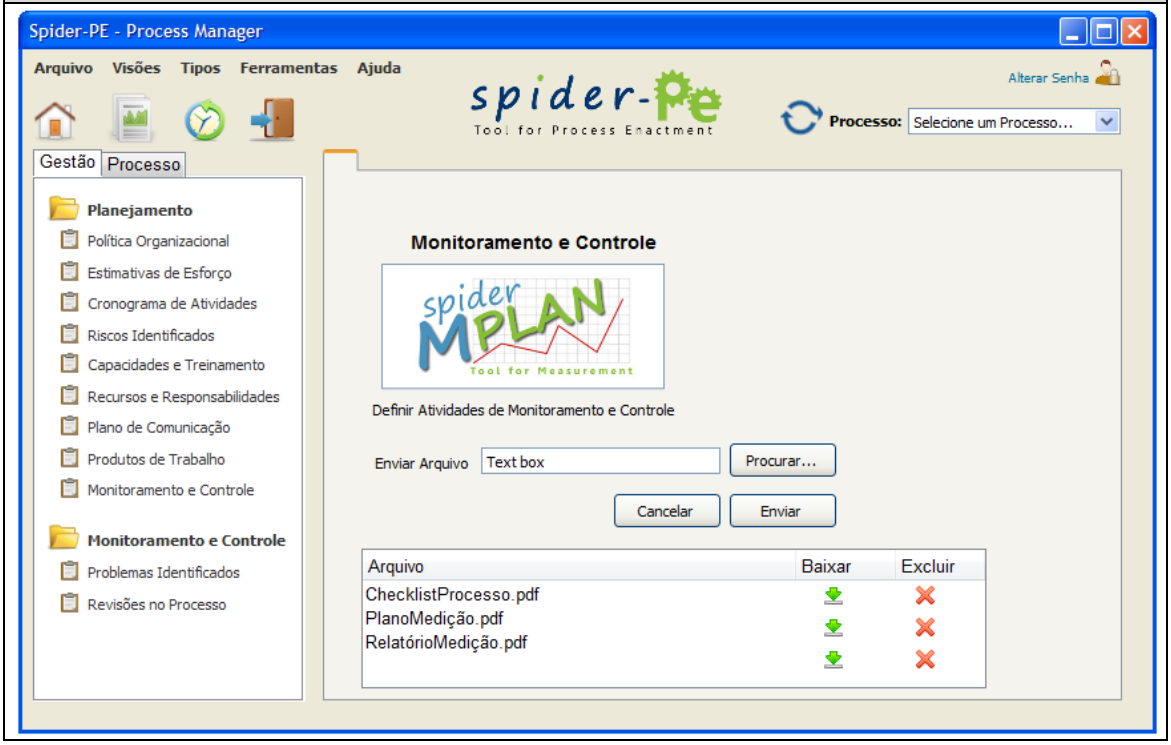
identificados.	de Trabalho.
Responsáveis/Pessoas Envolvidas	
Equipe de Garantia da Qualidade.	
Templates	
Não se aplica.	
Ferramentas de Apoio Utilizadas	
Spider-CL.	
Protótipo de Tela Correspondente	



m) Definir Atividades de Monitoramento e Controle

Itens dos Modelos:	**Nível de Aderência:**
RAP 3 / GP 2.2.	Totalmente Aderente.
Objetivo	
Definir métodos para monitorar e controlar as atividades que estão sendo desenvolvidas durante o projeto, a fim de garantir que a realização destas ocorra conforme o planejado.	
Critérios de Entrada	**Artefatos de Entrada**
Escopo, Cronograma e Lista de Risco para o Projeto definidos.	Plano do Projeto.
Passos	
• Verificar os marcos e os pontos de controle previamente definidos; • Identificar atividades do processo e associar os marcos e os pontos de	

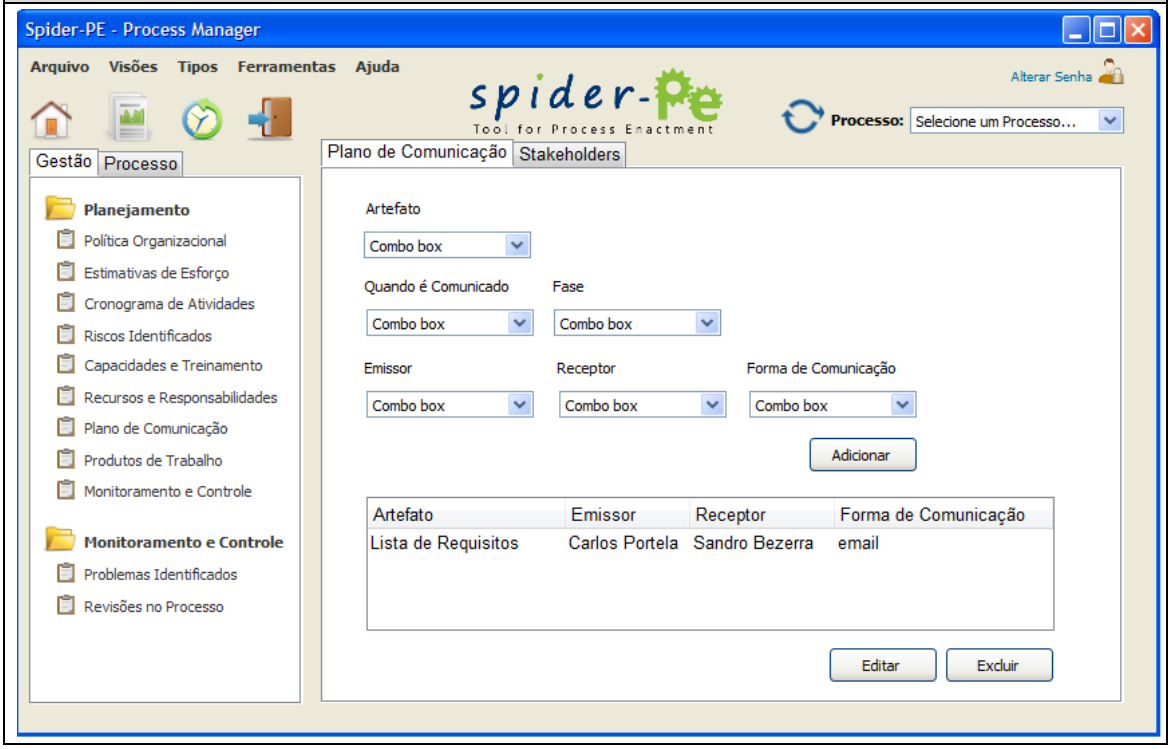
controle a estas; • Definir técnicas para monitorar o andamento da execução do Processo nos marcos e pontos de controle;	
Critérios de Saída	**Artefatos de Saída**
Ter as atividades para Monitoramento e Controle definidas.	Plano de Projeto atualizado.
Responsáveis/Pessoas Envolvidas	
Gerente de Projeto.	
Templates	
Não se aplica.	
Ferramentas de Apoio Utilizadas	
Spider-MPlan e Ferramenta de Execução de Processos.	
Protótipo de Tela Correspondente	



n) Gerenciar Comunicação

Itens dos Modelos:	**Nível de Aderência:**
RAP 8 / GP 2.7	Totalmente Aderente.
Objetivo	
Fazer com que as informações gerais em relação ao progresso de execução das atividades cheguem a todos os *stakeholders*.	
Critérios de Entrada	**Artefatos de Entrada**
Possuir um Plano de Comunicação	Plano de Comunicação.

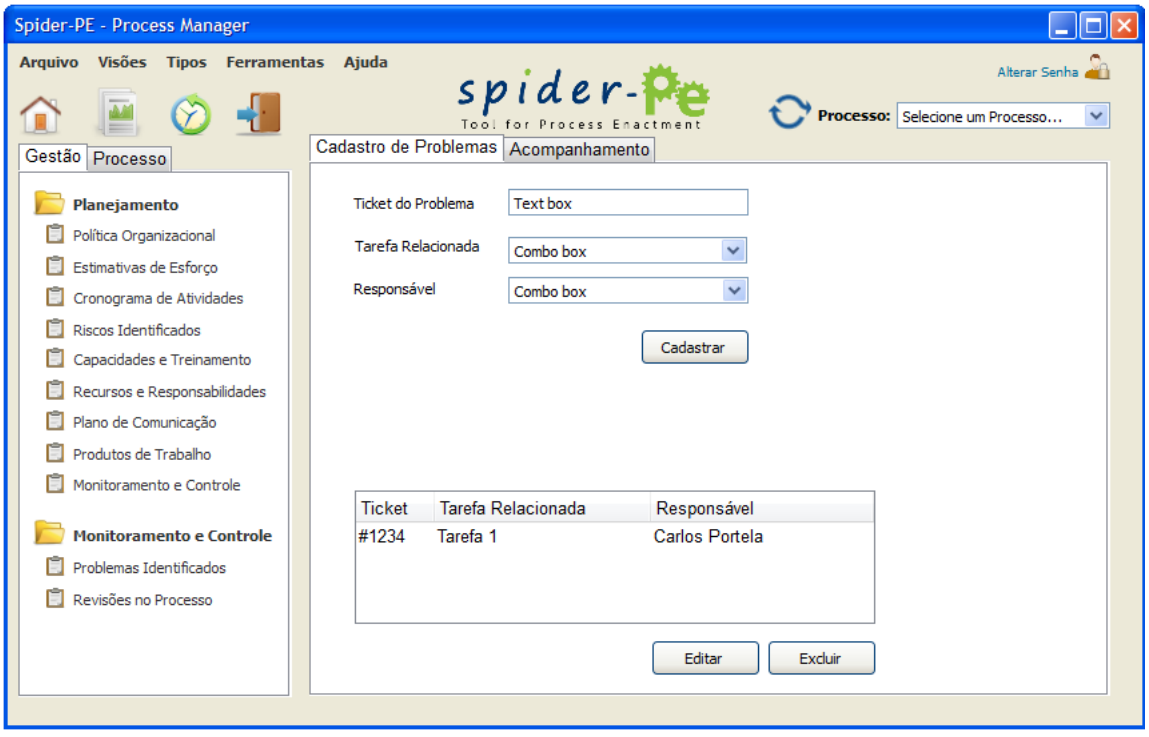
definido.	
Passos	
• Utilizar os meios e formas de comunicação definidos para interação entre os envolvidos no projeto; • Acompanhar a interação entre os envolvidos no projeto e solucionar possíveis problemas que possam surgir;	
Critérios de Saída	**Artefatos de Saída**
Gerenciamento efetivo da Comunicação.	Plano de Comunicação atualizado.
Responsáveis/Pessoas Envolvidas	
Gerente de Projeto.	
Templates	
Template de Plano de Comunicação.	
Ferramentas de Apoio Utilizadas	
Ferramenta de Execução de Processos.	
Protótipo de Tela Correspondente	



o) Realizar Monitoramento e Controle

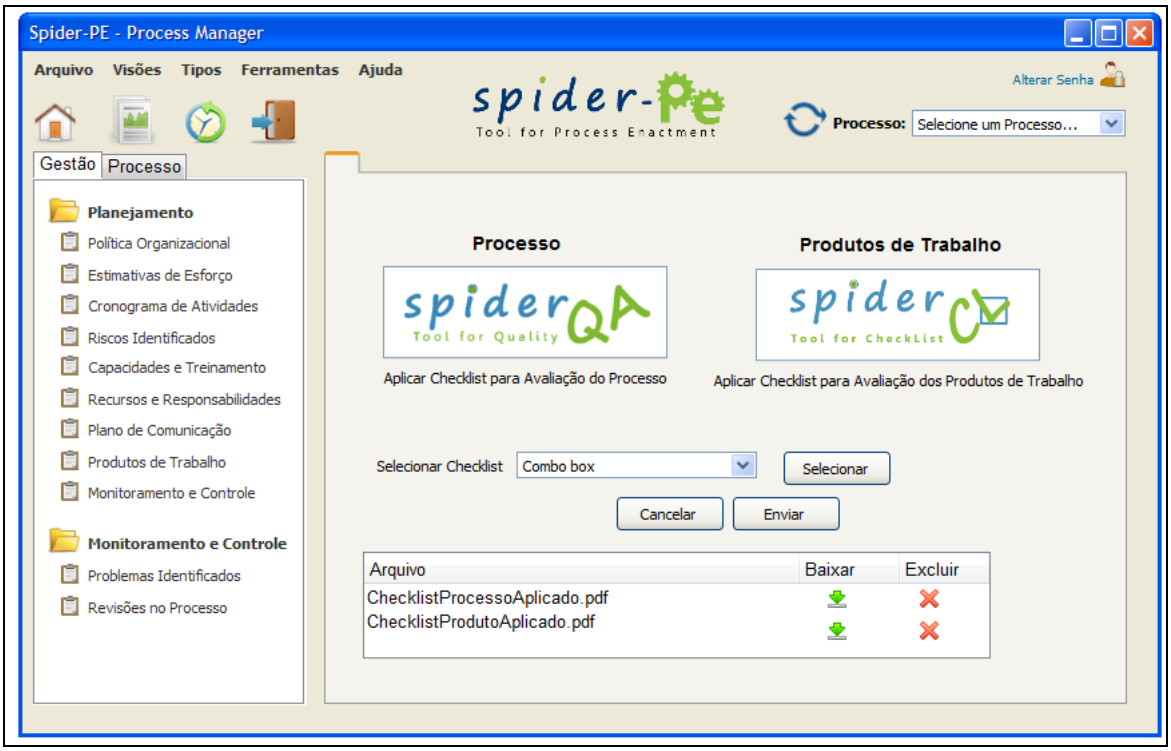
Itens dos Modelos:	**Nível de Aderência:**
RAP 4 / GP 2.8.	Totalmente Aderente.
Objetivo	
Cadastrar e monitorar os possíveis problemas que surgem durante a execução do projeto, além de acompanhar a resolução destes.	

Critérios de Entrada	Artefatos de Entrada
Ter Atividades para Monitoramento e Controle definidas.	Plano de Projeto atualizado.
Passos	
• Monitorar o projeto nos marcos e pontos de controle, verificando se o que foi planejado está de acordo com o que está sendo executado, afim de que ações corretivas apropriadas possam ser realizadas quando necessário; • Proceder o cadastro dos problemas e não-conformidades que ocorrerem durante as etapas do projeto; • Realizar ajustes necessários; • Acompanhar a resolução destes problemas e não-conformidades;	
Critérios de Saída	**Artefatos de Saída**
Problemas identificados cadastrados e acompanhados.	Relatório de Problemas e Não-Conformidades do Projeto.
Responsáveis/Pessoas Envolvidas	
Gerente de Projeto.	
Templates	
Template de Lista de Problemas e Não-Conformidades do Projeto.	
Ferramentas de Apoio Utilizadas	
Spider-MPlan e Redmine.	
Protótipo de Tela Correspondente	



p) Realizar Revisões no Processo

Itens dos Modelos:	**Nível de Aderência:**
RAP 9 / GP 2.10.	Parcialmente Aderente.
Objetivo	
Realizar revisões no processo a fim de garantir que este esteja aderente a sua descrição e tratar as não conformidades encontradas.	
Critérios de Entrada	**Artefatos de Entrada**
A execução de um projeto deve está em um marco.	Padrões e Procedimentos.
Passos	
• Estabelecer critérios de avaliação do processo; • Fazer um *checklist* com os critérios estabelecidos; • Fazer avaliação objetiva; • Listar as não-conformidades e tratá-las; • Elaborar o Parecer técnico da avaliação;	
Critérios de Saída	**Artefatos de Saída**
Revisão no processo efetuada.	*Checklist* e Parecer Técnico da Avaliação.
Responsáveis/Pessoas Envolvidas	
Equipe de Garantia de Qualidade.	
Templates	
Template de *Checklist* e Parecer Técnico.	
Ferramentas de Apoio Utilizadas	
Spider-CL.	
Protótipo de Tela Correspondente	



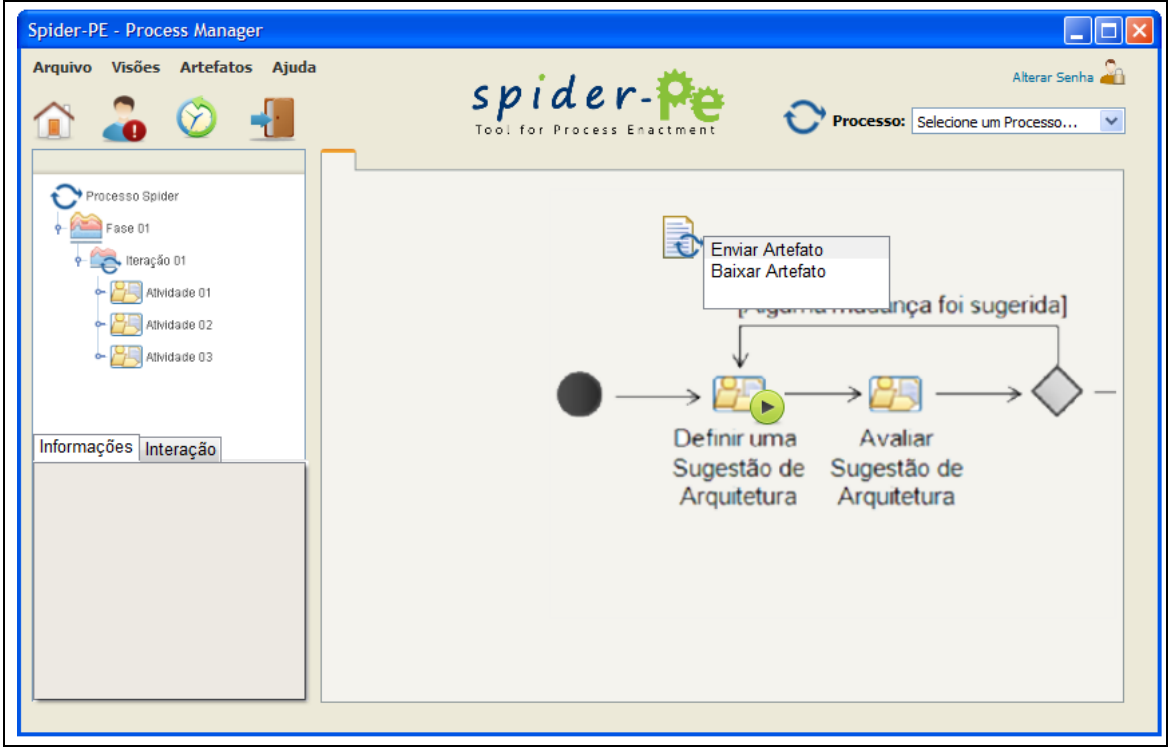
3.3.3.2 Fase de Execução das Atividades do Processo

Esta fase é responsável por verificar se o que é produzido está aderente aos padrões definidos para o processo. Suas atividades estão divididas da seguinte maneira: Gerar Produtos de Trabalho; Gerenciar Configurações; Verificar Aderência do Processo; Verificar Aderência dos Produtos de Trabalho; Os quadros a seguir fazem uma melhor descrição de cada uma das atividades acima listadas.

a) Gerar Produtos de Trabalho

Itens dos Modelos:	**Nível de Aderência:**
RAP 1 / GP 1.1.	Totalmente Aderente.
Objetivo	
Gerar os produtos de trabalho necessários para condução do projeto.	
Critérios de Entrada	**Artefatos de Entrada**
Templates, Padrões e Procedimentos	Modelo do Processo, Padrões,

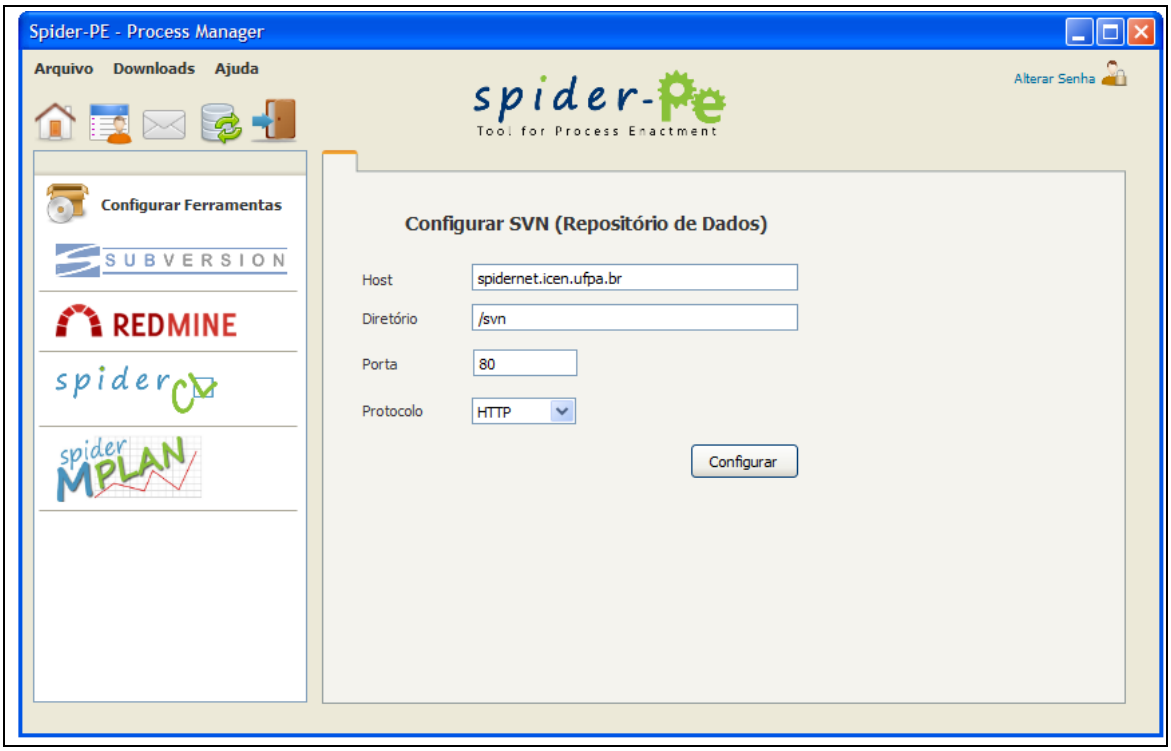
definidos.	Procedimentos e Templates.
Passos	
• Analisar a atividade do processo a ser executada, bem como as suas restrições e limitações; • Identificar artefatos que necessitam ser gerados na execução de uma atividade;	
Critérios de Saída	**Artefatos de Saída**
Produtos de trabalhos gerados.	Artefatos do processo.
Responsáveis/Pessoas Envolvidas	
Equipe do Projeto.	
Templates	
Templates de Artefatos do Processo.	
Ferramentas de Apoio Utilizadas	
Editor de Texto, IDE, Ferramenta de Execução de Processo, etc.	
Protótipo de Tela Correspondente	



b) Gerenciar Configurações

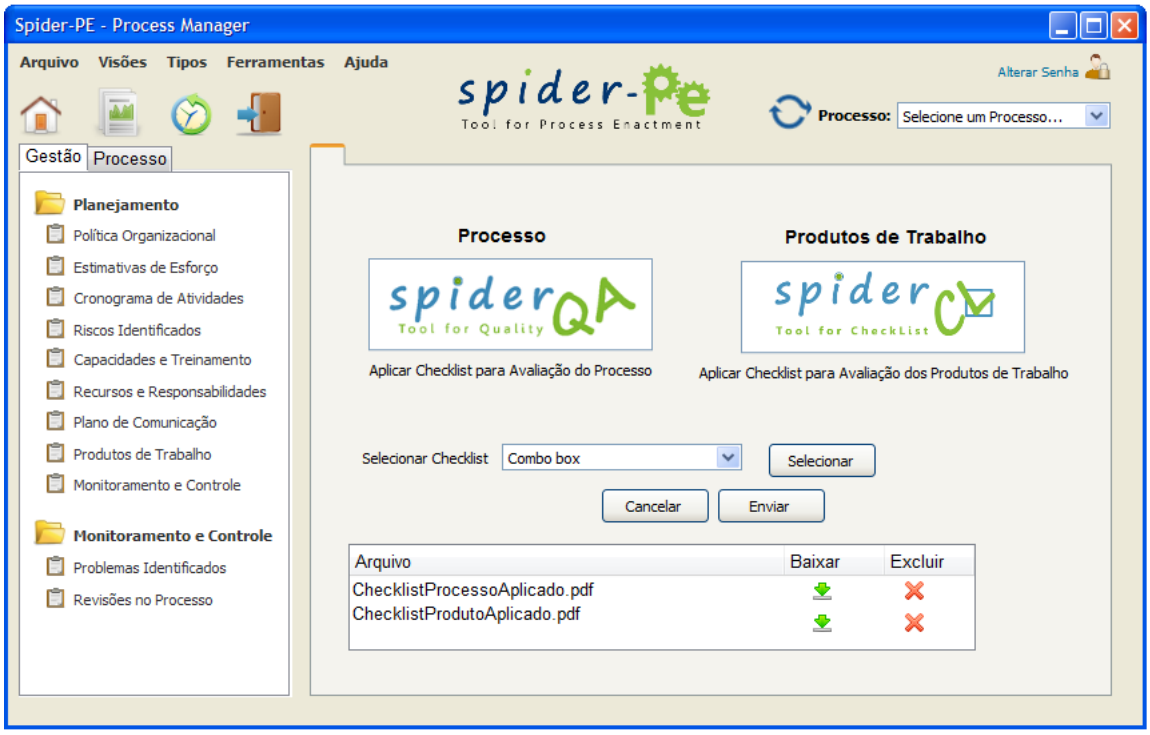
Itens dos Modelos:	**Nível de Aderência:**
RAP 12; RAP 13 / GP 2.6	Totalmente Aderente.
Objetivo	
Gerenciar a evolução do projeto através da gerencia de versões destes bem como a através do controle de acesso aos produtos de trabalho.	

Critérios de Entrada	**Artefatos de Entrada**
Repositório configurado e/ou Política de Configuração definida.	Política de Configuração.
Passos	
• Especificar e manter um sistema de gerência de configuração; • Definir uma política de versionamento; • Definir o nível de controle apropriado para cada produto de trabalho; • Especificar o nível de acesso de cada envolvido no projeto; • Registrar e disponibilizar situação dos produtos de trabalho; • Controlar as mudanças nos produtos de trabalho; • Realizar auditorias para garantir que os itens de configuração estejam íntegros, completos e consistentes;	
Critérios de Saída	**Artefatos de Saída**
Nível de acesso definido, política de versionamento, produtos de trabalhos versionados e disponíveis.	Plano de Configuração atualizado, Lista de Problemas Identificados, Plano de Ação e Relatórios de Auditoria.
Responsáveis/Pessoas Envolvidas	
Gerente de Configuração.	
Templates	
Templates de Plano de Configuração atualizado, Lista de Problemas Identificados, Plano de Ação e Relatórios de Auditoria.	
Ferramentas de Apoio Utilizadas	
SVN e Redmine.	
Protótipo de Tela Correspondente	



c) Verificar Aderência do Processo

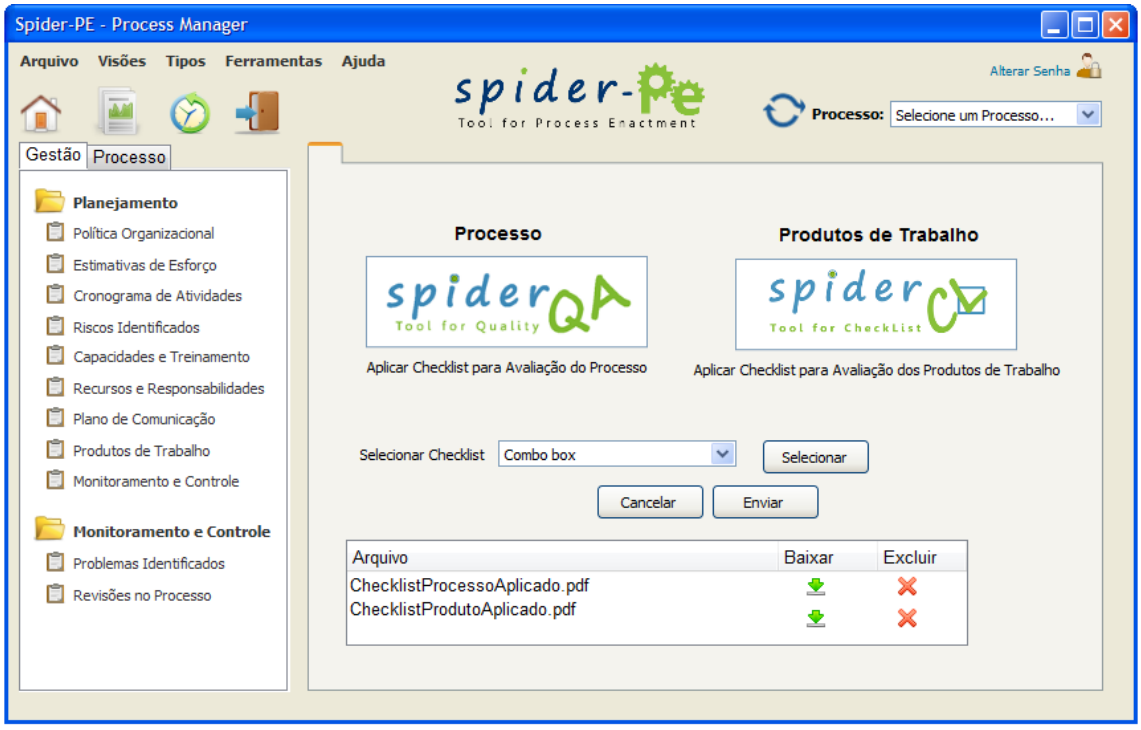
Itens dos Modelos:	**Nível de Aderência:**
RAP 10 / GP 2.9.	Totalmente Aderente.
Objetivo	
Verificar se o processo esta aderente aos padrões definidos.	
Critérios de Entrada	**Artefatos de Entrada**
Checklist de Avaliação, Padrões e Modelo do Processo estabelecidos.	*Checklist* de Avaliação.
Passos	
• Aplicar *checklist* de Avaliação do Processo; • Realizar entrevistas; • Analisar resultados obtidos; • Gerar relatório;	
Critérios de Saída	**Artefatos de Saída**
Aderência do Processo avaliada.	*Checklist* de Avaliação aplicado e Relatório de Avaliação gerado.
Responsáveis/Pessoas Envolvidas	
Equipe de Garantia da Qualidade.	
Templates	
Template de *Checklist* e Relatório de Avaliação.	
Ferramentas de Apoio Utilizadas	

Spider-QA.
Protótipo de Tela Correspondente


d) Verificar Aderência dos Produtos de Trabalho

Itens dos Modelos:	**Nível de Aderência:**
RAP 14	Totalmente Aderente.
Objetivo	
Verificar se os produtos de trabalho estão aderentes aos requisitos estabelecidos.	
Critérios de Entrada	**Artefatos de Entrada**
Checklist de Avaliação definido.	*Checklist* de Avaliação.
Passos	
• Aplicar *checkist*; • Realizar entrevistas; • Analisar resultados obtidos; • Gerar relatório;	
Critérios de Saída	**Artefatos de Saída**
Aderência dos produtos de trabalho checada.	*Checklist* de Avaliação aplicado e Relatório de Avaliação gerado.
Responsáveis/Pessoas Envolvidas	
Equipe de Garantia da Qualidade.	
Templates	

Template de *Checklist* e Relatório de Avaliação.
Ferramentas de Apoio Utilizadas
Spider-QA.
Protótipo de Tela Correspondente

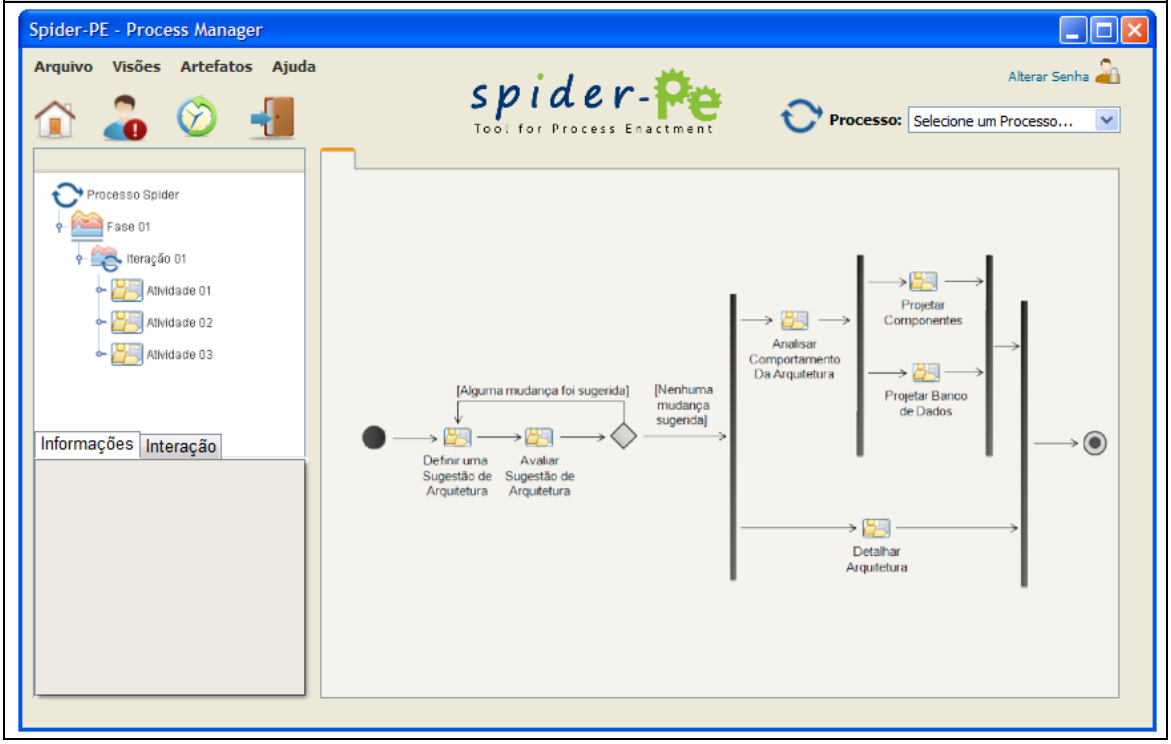


3.3.3.3 Fase de Aplicação do Formalismo de Execução

Esta fase é responsável pela aplicação do formalismo de execução nas atividades do processo que estão sendo executadas. Nela são executadas as seguintes atividades: Aplicar Hierarquia de Ativos do Processo; Aplicar Restrições do Processo; Controlar Variáveis do Projeto; Monitorar Estado do Processo; Monitorar Tempos do Processo; Registrar Histórico de Eventos e Realizar *Feedback*. Os quadros a segui retratam uma melhor descrição de cada uma das atividades acima listadas.

a) Aplicar Hierarquia de Ativos do Processo

Itens dos Modelos:	Nível de Aderência:
RAP 1 / GP 1.1	Parcialmente Aderente.
Objetivo	
Distinguir os elementos do processo de acordo com seu nível hierárquico.	
Critérios de Entrada	
O nível hierárquico da linguagem de modelagem do processo deve estar definido.	
Passos	
• Consultar Especificação Técnica da SPIDER_ML (disponível no site www.spider.ufpa.br); • Avaliar a modelagem do processo de acordo com o nível hierárquico definido na SPIDER_ML;	
Critérios de Saída	
Hierarquia do processo aplicada aos seus elementos.	
Responsáveis/Pessoas Envolvidas	
Ferramenta de Execução de Processos / Equipe de Garantia da Qualidade.	
Protótipo de Tela Correspondente	



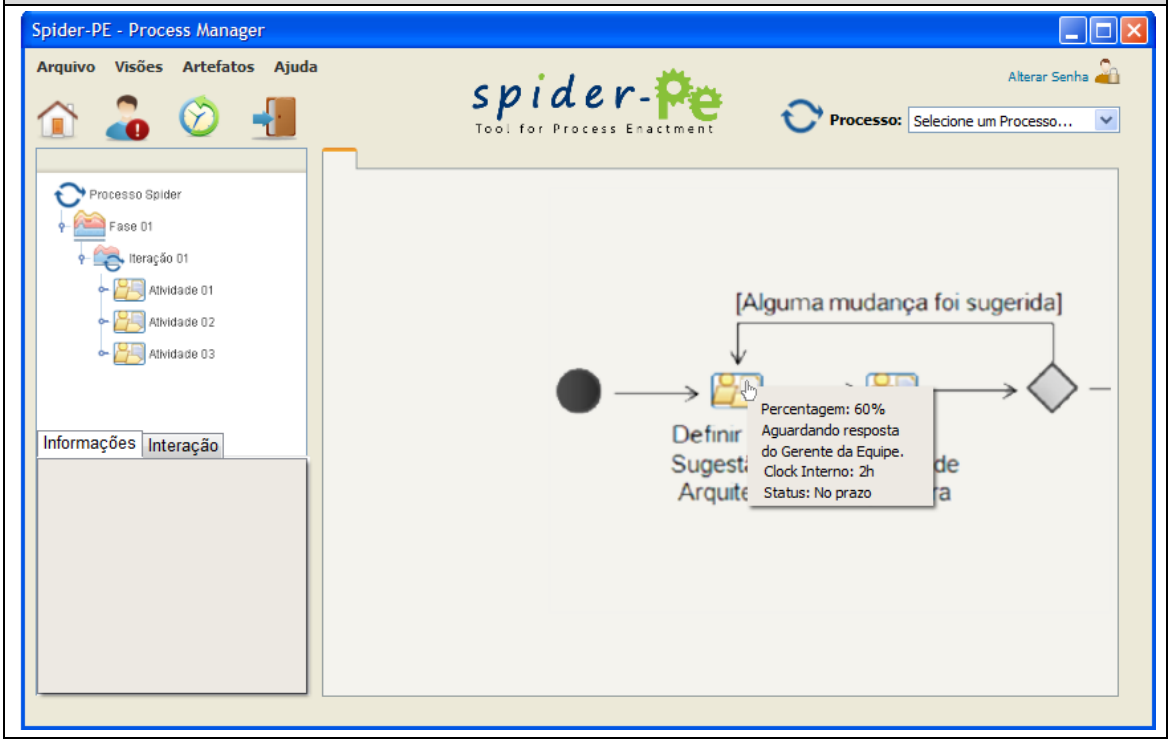
b) Aplicar Restrições do Processo

Itens dos Modelos:	Nível de Aderência:
RAP 1 / GP 1.1	Parcialmente Aderente.
Objetivo	

Especificar a relação das conexões entre as atividades para permitir a execução do processo.
Critérios de Entrada
Processo modelado.
Passos
• Consultar Especificação Técnica da xSPIDER_ML (disponível no site www.spider.ufpa.br); • Definir as regras de restrições entre as atividades do processo modelado;
Critérios de Saída
Fluxo de execução entre as atividades dos processos definidos.
Responsáveis/Pessoas Envolvidas
Ferramenta de Execução de Processos / Gerente de Processo.
Protótipo de Tela Correspondente

c) Controlar Variáveis do Projeto

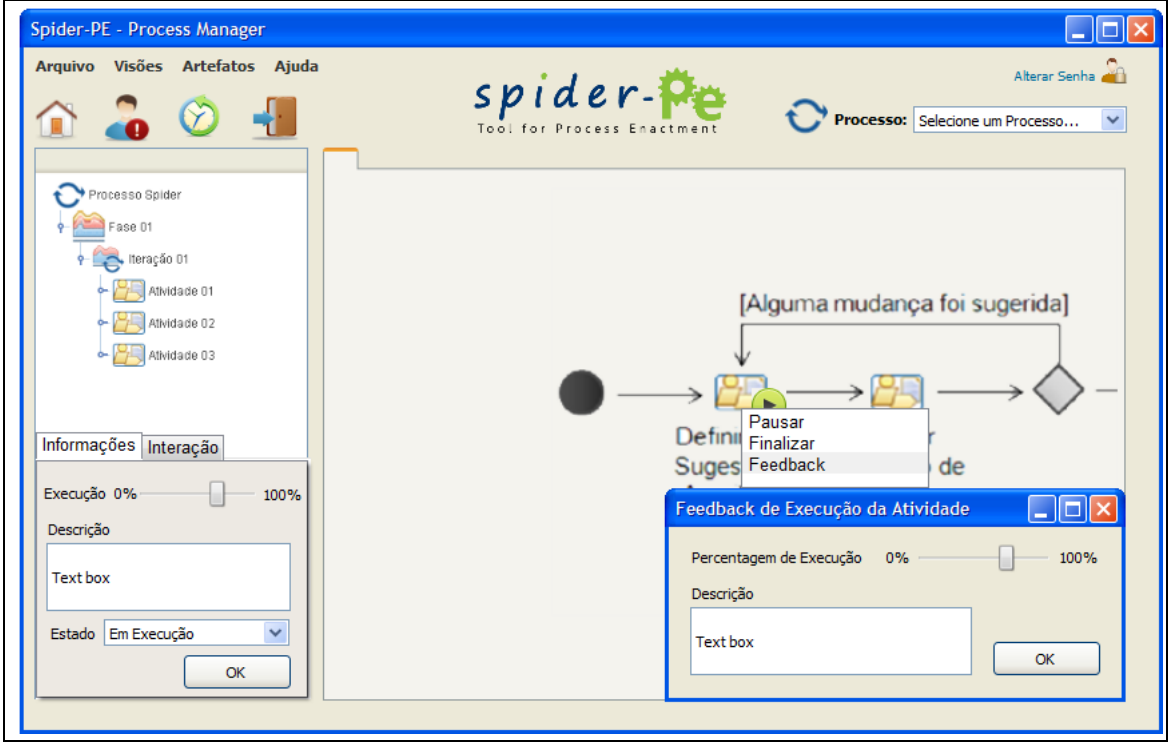
Itens dos Modelos:	**Nível de Aderência:**
RAP 1 / GP 1.1.	Parcialmente Aderente.
Objetivo	
Cadastrar e acompanhar as especificidades do projeto em relação a recursos (tipo), em relação às atividades (cronograma e carga horária).	
Critérios de Entrada	

Plano de Projeto.
Passos
• Consultar Especificação Técnica da xSPIDER_ML (disponível no site www.spider.ufpa.br); • Definir as variáveis para execução das atividades do processo modelado;
Critérios de Saída
Acompanhar a utilização de Recursos e Cronograma.
Responsáveis/Pessoas Envolvidas
Ferramenta de Execução de Processos.
Protótipo de Tela Correspondente


d) Monitorar Estado do Processo

Itens dos Modelos:	**Nível de Aderência:**
RAP 1 / GP 1.1.	Parcialmente Aderente.
Objetivo	
Efetuar o monitoramento dos estados dos elementos do processo de acordo com as regras de transição de estados.	
Critérios de Entrada	
Processo está em execução.	
Passos	
• Consultar Especificação Técnica da xSPIDER_ML (disponível no site	

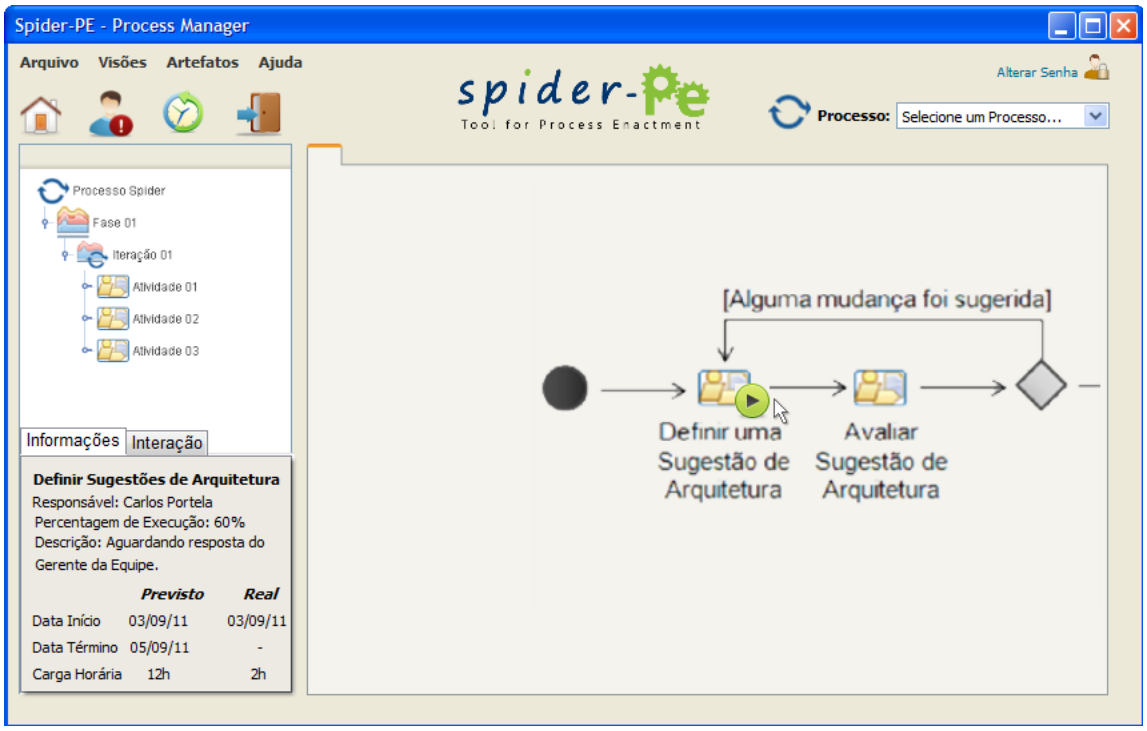
www.spider.ufpa.br); • Avaliar a mudança de estados nos elementos do processo de acordo com as regras de transição definidas na xSPIDER_ML;
Critérios de Saída
Monitoramento do estado dos elementos do Processo em execução até que o mesmo seja concluído.
Responsáveis/Pessoas Envolvidas
Ferramenta de Execução de Processos.
Protótipo de Tela Correspondente



e) Monitorar Tempos do Processo

Itens dos Modelos:	**Nível de Aderência:**
RAP 1 / GP 1.1.	Parcialmente Aderente.
Objetivo	
Monitorar o processo em relação ao tempo estimado e real de execução do mesmo.	
Critérios de Entrada	
Iniciar a execução do processo.	
Passos	
• Consultar Especificação Técnica da xSPIDER_ML (disponível no site www.spider.ufpa.br); • Contabilizar o tempo de execução de uma tarefa e comparar com o tempo estimado;	

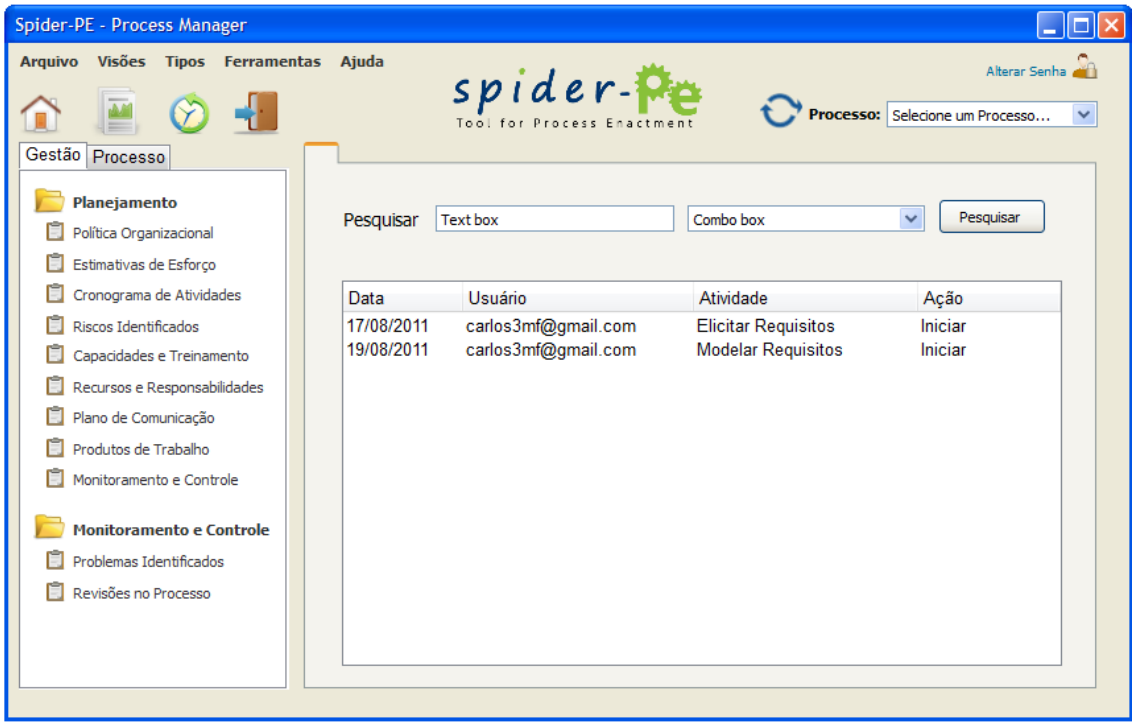
• Atualizar o status da tarefa de acordo com o tempo de execução e a carga horaria definida para a tarefa;
Critérios de Saída
Acompanhamento da execução do processo e informação dos tempos de execução de cada elemento desse processo.
Responsáveis/Pessoas Envolvidas
Ferramenta de Execução de Processos.
Protótipo de Tela Correspondente



f) Registar Histórico de Eventos

Itens dos Modelos:	**Nível de Aderência:**
RAP 1 / GP 1.1.	Parcialmente Aderente.
Objetivo	
Visa registrar as transições de estados e tempo como também os principais eventos que ocorrem durante a execução do processo.	
Critérios de Entrada	
Iniciar a execução do processo.	
Passos	
• Consultar Especificação Técnica da xSPIDER_ML (disponível no site www.spider.ufpa.br); • Cadastrar os eventos (transição de estados, status das tarefas, tempo de execução, entre outros) gerados a partir da execução de cada tarefa do processo;	

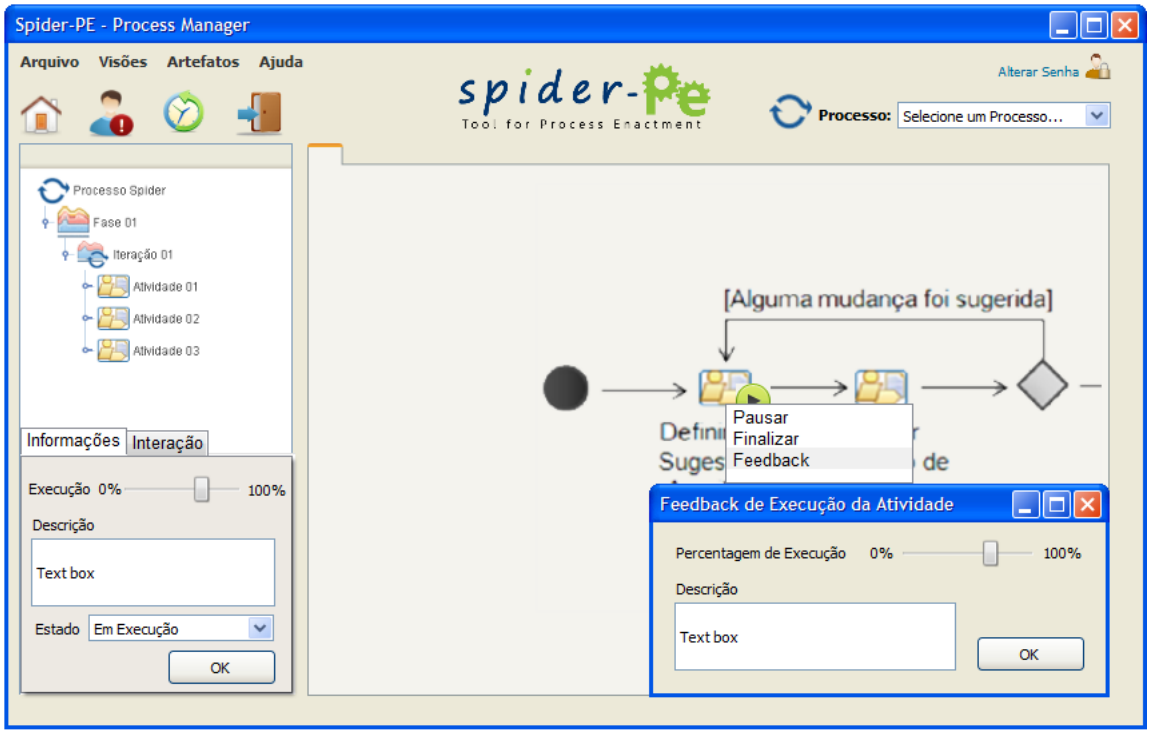
Critérios de Saída
O *Log* de eventos realizados deve ser salvo e está disponível para consulta.
Responsáveis/Pessoas Envolvidas
Ferramenta de Execução de Processos.
Protótipo de Tela Correspondente



g) Realizar *Feedback*

Itens dos Modelos:	Nível de Aderência:
RAP 1 / GP 1.1.	Parcialmente Aderente.
Objetivo	
Dar flexibilidade à execução do processo, permitindo retornar a realização de uma atividade do processo durante a sua execução.	
Critérios de Entrada	
Processo modelado.	
Passos	
• Consultar Especificação Técnica da xSPIDER_ML (disponível no site www.spider.ufpa.br); • Permitir, quando necessário, o retorno da execução de uma atividade do processo de acordo com as regras de hierarquia e restrições definidas ao processo;	
Critérios de Saída	
Processo poder ser alterado durante sua execução.	

Responsáveis/Pessoas Envolvidas
Ferramenta de Execução de Processos.
Protótipo de Tela Correspondente



3.4 APLICAÇÃO DO *FRAMEWORK*

Este *framework* apresenta as atividades consideradas necessárias para a execução genérica de qualquer processo de software, sendo estas atividades derivadas de recomendações de Modelos de Qualidade e de requisitos para execução de processo, como o Formalismo de Execução. Sendo assim, espera-se que este possa ser adotado como um referencial por qualquer organização desenvolvedora de software, no apoio à execução de seus processos, a partir da adoção de melhores práticas incorporadas neste *framework*, através da relação deste com os modelos de qualidade.

Espera-se, ainda, que a aplicação deste *framework* possa facilitar a adoção destes modelos de qualidade pelas organizações devido à visibilidade do andamento da execução das atividades dos projetos de software que este tipo de tecnologia permite e à geração de indicadores que explicitam o

atendimento às exigências e recomendações dos modelos CMMI-DEV e MR-MPS.

3.5 CONSIDERAÇÕES FINAIS

Neste capítulo foi apresentado o Projeto SPIDER e as pesquisas relacionadas a este trabalho que serviram de embasamento para que o mesmo fosse realizado. Também, destacou-se a aderência aos modelos de qualidade CMMI-DEV, que possui notoriedade internacional, e o Modelo de Referência do MPS.BR, que é voltado para a realidade brasileira de processo de software. Esta aderência se dá no contexto da capacidade do processo de software, que descreve o conjunto de resultados esperados que possam ser atingidos quando se executa o processo de software.

Apresentou-se o *Framework* para execução de processo de software Spider-PE, principal contribuição deste Trabalho de Conclusão de Curso, a partir de seus elementos estruturais, como fases, atores e descrição das atividades, bem como foram apresentados protótipos de tela que visam a validação deste *framework* e permite o desenvolvimento de uma ferramenta que aplica as atividades propostas. Por fim, mostrou-se como este *framework* pode ser aplicado nos processos das organizações desenvolvedoras de software.

4 CONCLUSÕES

Este capítulo apresenta as principais conclusões do trabalho realizado e contribuições do mesmo para a área da Engenharia de Software no que tange à Execução de Processo de Software.

No capítulo introdutório procurou-se contextualizar o trabalho, mostrando as justificativas e motivações deste, bem como os objetivos e metodologias aplicadas para a realização deste.

No segundo capítulo foi introduzido o conceito de processo de software, bem como suas etapas e componentes. Foram abordados aspectos relacionados à qualidade de software, dentro deste contexto foram introduzidos os modelos de qualidade de software bem como sua definição de maturidade e capacidade de processo de software. Ainda neste capítulo, foi dada a definição de execução de processo de software e qual o contexto em que esta se insere.

A partir do terceiro capítulo apresenta-se o *Framework* de execução de processo. Este *framework* é parte de um Ferramental de Apoio Execução Flexível de Processos de Software, denominado Spider-PE. Foram definidos para o *framework* três fases, cada uma delas está responsável por uma determinado conjunto de atividades com objetivos semelhantes: preparar e acompanhar a execução do processo; executar as atividades; e aplicar o formalismo de execução de processo nos projetos em execução. Para cada atividade do *framework*, apresenta-se um protótipo de tela que objetiva a validação do *framework* proposto.

4.1 RESULTADOS OBTIDOS

Através deste trabalho, obteve-se o mapeamento entre os modelos de qualidade de software, relacionando os perfis de capacidade entre o MPS.BR e o CMMI. Este mapeamento foi realizado com o intuito de permitir a aderência

do *framework à* estes modelos. Cada atividade deste *framework* possui equivalência com as recomendações presentes nestes modelos. Este *framework* está diretamente relacionado ao conceito de perfil de capacidade de processo de software em sua estrutura, uma vez que as organizações cumprindo com êxito as atividades nele contida estarão atendendo também os atributos de processo e objetivos genéricos destes modelos de qualidade.

Espera-se que o *Framework* apresentado neste trabalho auxilie uma maior institucionalização de boas práticas recomendadas pelos modelos de qualidade de software nas organizações desenvolvedoras de software.

Este *Framework* também pode servir de referência para o desenvolvimento de uma ferramenta, a partir dos protótipos apresentados no capítulo 3 na Seção 3.3.3.

4.2 TRABALHOS FUTUROS

Este *framework* serve de base para o desenvolvimento de um ferramental de apoio para execução de processo de software chamado Spider-PE, que é desenvolvido dentro do Projeto SPIDER. Sua contribuição pode auxiliar a condução de pesquisas relacionadas tanto à execução quanto a instanciação de processos, a partir do *framework* e demais trabalhos relacionados à aderência de tecnologias à modelos de qualidade de processo de software.

5 REFERÊNCIAS

BARROS, R. S. **Um Ferramental de Apoio para Modelagem de Processos de Software**. Trabalho de Conclusão de Curso - Curso de Bacharelado em Ciência da Computação - Universidade Federal do Pará, Belém, 2010.

CHAVES, R. *et al*. **A Software Process Simulator Machine for Software Engineering Simulation Games**.In ACM Computers in Entertainment. Vol. 8, No. 4. 2011.

GARG, P.; JAZAYERI, M. **Process-Centered Software Engineering Environments**. IEEE Computer Society Press. 1996.

HUMPHREY, W. S. **Managing the Software Process.** Addison Wesley, 1989.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION/ INTERNATIONAL ELECTROTECHNICAL COMISSION.**ISO/IEC 15504-2**: Information Technology - Process Assessment – Part 2 – Performing an Assessment. Geneve: ISO, 2003.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION/ INTERNATIONAL ELECTROTECHNICAL COMISSION. **ISO/IEC 12207**: Systems and Software Engineering– Software Life Cycle Processes. Geneve: ISO, 2008.

OLIVEIRA, S. R. B. **Processo de Software: Ambiente e Mecanismos de Execução**. Exame de Qualificação do Doutorado – Centro de Informática – Universidade Federal de Pernambuco, Recife, 2006.

OLIVEIRA, S. R. B. **ProDefiner: Uma Abordagem Progressiva para a Definição de Processos de Software no Contexto de um Ambiente Centrado no Processo**. Tese de Doutorado – Pós-Graduação em Ciência da Computação – Universidade Federal de Pernambuco, Recife, 2007.

OLIVEIRA, S. R. B. et al. 2011. **SPIDER: Uma Proposta de Solução Sistêmica de um SUITE de Ferramentas de Software Livre de Apoio à**

Implementação do Modelo MPS.BR. Revista do Programa Brasileiro da Qualidade e Produtividade em Software. pp. 103-1107. 2ª Ed. junho, 2011.

OLIVEIRA, S. R. B. et al. **Uma Proposta de um Ambiente de Implementação de Processo de Software**. Infocomp – Journal of Computer Science, 2005.

PORTELA, C. S., VASCONCELOS, A. M. L., OLIVEIRA, S. R. B. Spider-PE: **Um Ferramental de Apoio à Execução de Processos aderente a Modelos de Qualidade**. Workshop de Teses e Dissertações de Qualidade de Software, 2011, Curitiba, Brasil.

REIS, C. A. L. **Ambientes de Desenvolvimento de Software e seus Mecanismos de Execução de Processo de Software**. Exame de Qualificaçao - Programa de Pós-Graduação em Ciência da Computação - Universidade Federal do Rio Grande do Sul, Porto Alegre, Fevereiro 2000.

REIS, C. A. L. **Uma Abordagem Flexível para Execução de Processos de Software Evolutivos**. Tese de Doutorado – Programa de Pós-Graduação em Computação – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2003.

SEI - SOFTWARE ENGINEERING INSTITUTE.**CMMI for Development**. Carnegie Mellon University: 2010. Disponível em: <http://www.sei.cmu.edu/reports/10tr033.pdf>. Acesso em: 05Nov. 2011.

SOFTEX - ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO. **MPS.BR**: Guia Geral. 2011. Disponível em: <http://www.softex.br/mpsbr/_guias/guias/MPS.BR_Guia_Geral_2011.pdf >. Acesso em: 05 Nov. 2011.

SOUZA, J. N. S., OLIVEIRA, S. R. B. **Uma Proposta de Apoio Sistêmico à Avaliação de Processos com base no MA-MPS, SCAMPI e ISO/IEC 15504**. Anais do Workshop de Teses e Dissertações de Qualidade de Software - WTDQS, Belém, 2010.