

Características para a Definição de Processo de Software em um Ambiente de Implementação de Processo de Software

Sandro Ronaldo Bezerra Oliveira
Universidade da Amazônia – UNAMA
Centro de Ciências Exatas e Tecnologia
Av. Alcindo Cacela, 287, 66060-902,
Belém – PA – Brasil
Fone: (+55 91) 40093000
e-mail: sandro@cci.unama.br

Alexandre Marcos Lins de Vasconcelos
Universidade Federal de Pernambuco - UFPE
Centro de Informática
Caixa Postal 7851, 50732-970
Recife – PE – Brasil
Fone: (+55 91) 21268430
e-mail: amlv@cin.ufpe.br

RESUMO

Quando são definidos, processos devem levar em consideração especificidades do projeto em questão. Embora diferentes projetos requeiram processos com características específicas para atender às suas particularidades, é possível estabelecer um conjunto de ativos de processo de software comum a todos os processos de software de uma organização, constituindo um Processo Padrão da organização, a partir do qual podem ser definidos processos específicos de projetos, levando em conta suas características particulares. Desta forma, este trabalho apresenta uma sugestão de um conjunto de características organizacionais, de projetos e de produtos de software que possibilitam esta definição do processo.

PALAVRAS-CHAVE: Processo de Software, Qualidade de Processo de Software, Definição do Processo de Software, Características.

ABSTRACT

When processes are defined, they must take in consideration specificities of the project in course. Although different projects require processes with specific characteristics to take care to its particularities, it is possible to establish a set of software process assets common to all the software processes of an organization, being constituted a Standard Process of the organization, from which can be defined specific processes of projects, leading in account its particular characteristics. This way, this paper presents a suggestion of set of organizational, software projects and products characteristics that make possible this process definition.

KEYWORDS: Software Process, Software Process Quality, Software Process Definition, Characteristics.

1. INTRODUÇÃO

O desenvolvimento de um software de qualidade, com elevada produtividade, dentro do prazo estabelecido e sem necessitar de mais recursos que aqueles alocados tornou-se um desafio para as organizações [13]. O crescimento do tamanho e da complexidade dos produtos de software faz com que estes assumam papéis críticos nos negócios das organizações.

A experiência da indústria de software mostra que a principal razão para o insucesso dos projetos está na falta de um processo de software disciplinado, ou seja, na falta de um mecanismo que habilite o gerenciamento e controle da qualidade dos produtos. Seguindo a mesma tendência, já é amplamente aceito que a qualidade de um produto de software seja fortemente determinada pela qualidade do processo utilizado durante o seu desenvolvimento e manutenção [20].

Conhecer os processos significa conhecer como os produtos são planejados e produzidos. Cabe ressaltar que, a partir da definição do processo, é possível definir-se medições e coletar dados de execução. Isto dá visibilidade aos gerentes e técnicos sobre o andamento dos projetos, possibilitando ações para controlar as variações do projeto e dos processos por ele utilizados. Neste sentido, a definição de um processo de software é um requisito básico para a obtenção de produtos de software de qualidade.

No entanto, a definição de um processo de software não é uma atividade simples; exige experiência e envolve o conhecimento de muitos aspectos da engenharia de software. A dificuldade em definir processos encontra-se na ausência de um processo de software possível de ser genericamente aplicado. Os processos variam porque são diferentes os tipos de sistemas, os domínios de aplicação, as equipes, as organizações e as próprias restrições de negócio, tais como, cronograma, custo, qualidade e confiabilidade [11].

Neste contexto Oliveira propôs, em [19], a definição de um ambiente para a implementação de processo de software, o *ImPProS* (Ambiente de Implementação Progressiva de Processos de Software), com os objetivos de possibilitar: a especificação dos processos de acordo com o

domínio do projeto específico e das características da organização; a instanciação do processo de software de acordo com as propriedades de cada projeto; sua simulação a partir dos parâmetros de configuração (prazo, pressões, custo, recursos, etc.); uma execução (automação) mais próxima do que se espera para um processo organizacional; e uma avaliação a partir da coleta de métricas desta execução.

Este trabalho sugere um levantamento e análise das características de projetos software e organizacionais que influenciam na definição do processo de desenvolvimento. Como resultado será apresentado um conjunto de características de projeto e das organizações de desenvolvimento de software que irão servir de base de conhecimento para a definição de processos no ambiente de desenvolvimento de software *ImPProS*. Serão, ainda, apresentados que componentes do processo de software (atividades, técnicas, modelos de ciclo de vida, etc.) estas características influenciam no momento da definição deste.

Além desta seção introdutória, o artigo apresenta outras quatro seções. A seção 2 aborda as propriedades que compõem o ambiente de implementação de processo de software proposto por [19]. Na seção 3 tem-se o detalhamento das características de definição do processo de software. Na seção 4, para cada característica, serão feitas algumas considerações acerca do impacto da característica na definição do processo de software. Finalmente, a seção 5 apresenta as considerações finais.

2. *ImPProS*: UM AMBIENTE DE IMPLEMENTAÇÃO PROGRESSIVA DO PROCESSO DE SOFTWARE

Para ajudar uma organização na implementação progressiva de um processo de software, é útil fornecer apoio automatizado por meio de um ambiente capaz de suportar as fases que a literatura especializada propõe como necessárias. O termo “progressiva” decorre do fato de que a implementação do processo é aperfeiçoado com as experiências aprendidas na sua definição, simulação, execução e avaliação.

O *ImPProS* é um projeto de iniciativa do Centro de Informática da UFPE – Universidade Federal de Pernambuco com a parceria da UNAMA – Universidade da Amazônia, financiado pelo CNPq – Conselho Nacional de Desenvolvimento Científico e Tecnológico, que visa a criação de um ambiente de apoio à implementação de um processo de software em uma organização de forma progressiva. O termo “progressiva” decorre do fato de que a implementação do processo é aperfeiçoada com as experiências aprendidas na sua definição, simulação, execução e avaliação.

O *ImPProS* é composto de um ambiente cooperativo, formado por nove ferramentas principais, como pode ser visualizado na Figura 1:

- ***ProDefiner***: provê a definição do processo de software a partir da análise de características específicas e aprendizado adquirido com outras definições;
- ***ProSimulator***: possibilita a simulação do processo de software instanciado a partir de um plano de execução do processo e assim antever problemas;
- ***ProEnacter***: permite a execução automatizada e acompanhamento do processo de software pela equipe do projeto;
- ***ProEvaluator***: provê a avaliação da execução do processo de software a partir da análise de critérios qualitativos e quantitativos;
- ***ProImprove***: possibilita a execução sistemática das atividades de melhoria do processo de software, a partir do modelo IDEAL;
- ***ProAnalyser***: permite a análise e tomada de decisão acerca da avaliação de itens que compõe o processo **de software**;
- ***ProReuse***: provê a execução do reuso de processo de software a partir da definição do escopo do projeto e sua adaptação ao contexto de uso;
- ***ProKnowledge***: possibilita a coleta, análise e uso de conhecimentos aprendidos ao longo da execução do processo de software;

- **ProConverter:** provê a conversão dos componentes do processo de software a partir das estruturas especificadas por normas/modelos de qualidade.

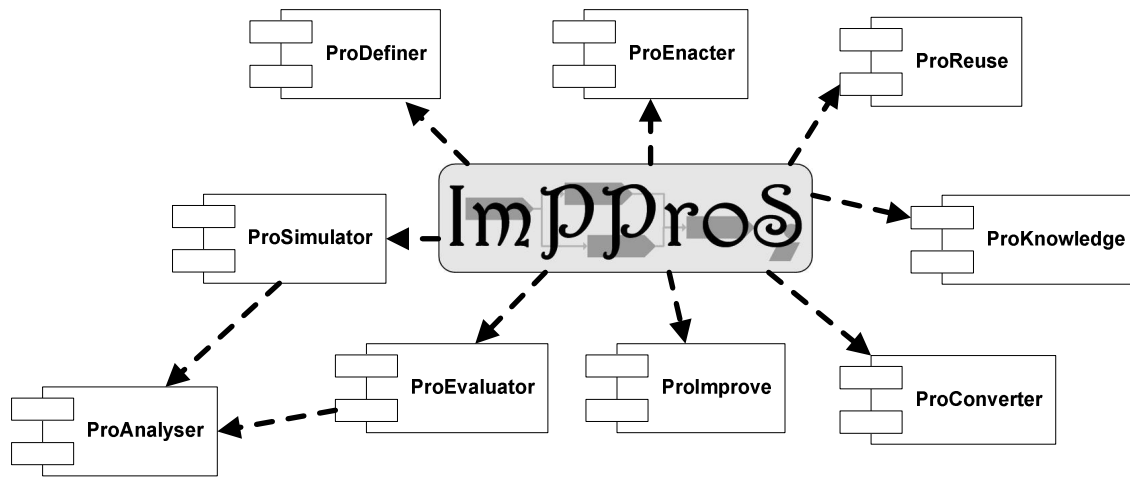


Figura 1. Ferramentas de Apoio ao Processo de Software do ImPProS

3. CARACTERÍSTICAS PARA A DEFINIÇÃO DO PROCESSO DE SOFTWARE NO *ImPProS*

Na definição de processos de software do *ImPProS*, como pode ser visualizado na Figura 2, adaptada do modelo definido por [22], inicialmente encontra-se o Meta-modelo de processo de software, composto de componentes e dos relacionamentos entre esses que são oriundos do mapeamento de algumas normas e modelos de qualidade para processo de software (CMMI [6], SPICE – ISO 15504 [10], ISO 12207 [11]). O objetivo deste meta-modelo é determinar uma terminologia única para a definição de processos de software no *ImPProS*.

Por sua vez, a definição de um processo padrão estabelece uma estrutura comum a ser utilizada pela organização nos seus projetos de software e constitui a base para a definição de todos os seus processos. Dessa forma, estabelece-se um processo básico que servirá como ponto de partida para a posterior definição dos processos de software adequados às diferentes características de cada projeto, permitindo economia de tempo e esforço na definição de novos processos.

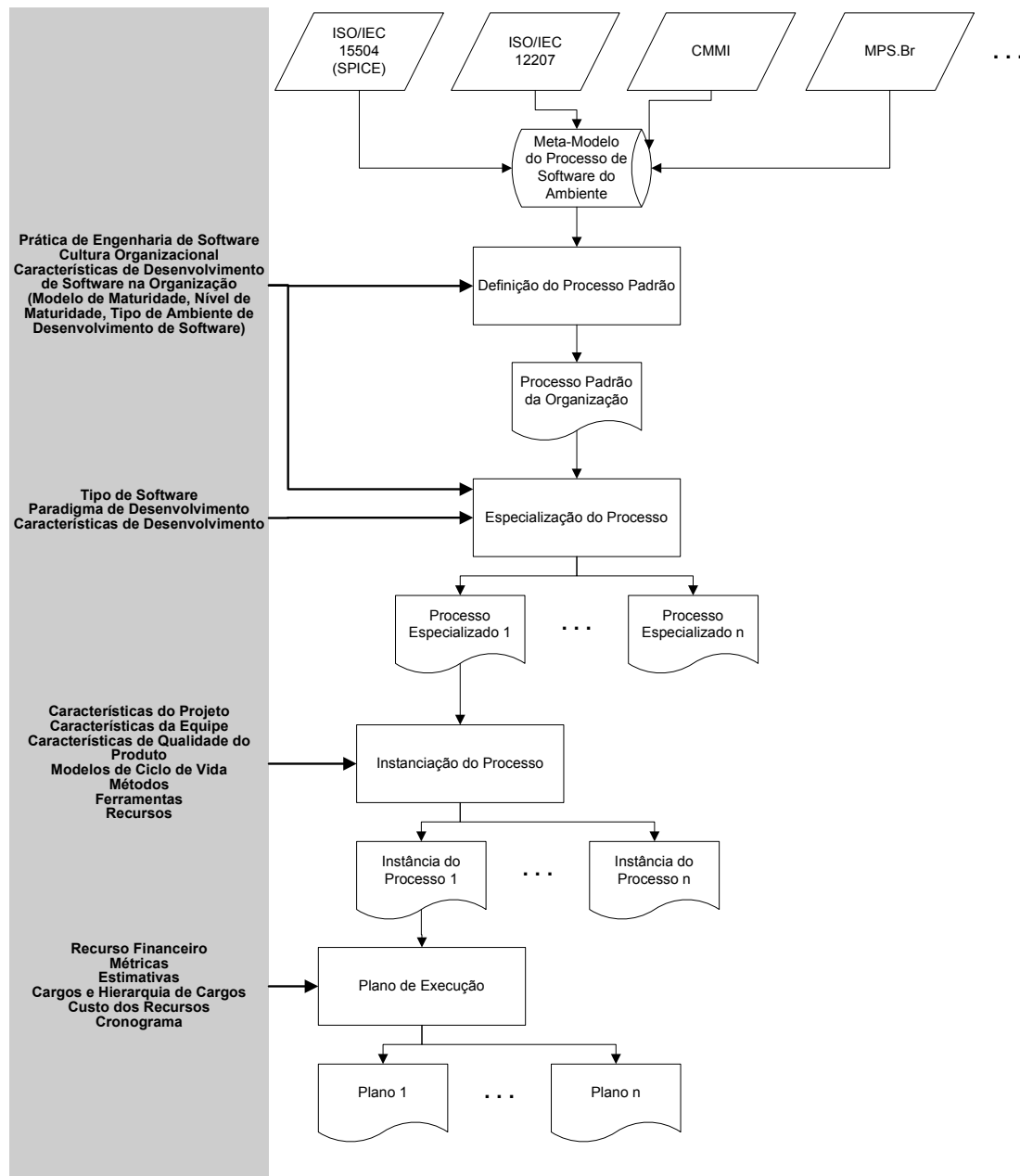


Figura 2. Estrutura de Definição do Processo de Software no ImPProS

Tendo em vista que tipos de software diferentes possuem características distintas e requerem diferentes abordagens de desenvolvimento, o processo de software padrão da organização deverá ser adaptado (especializado) considerando-se as características relacionadas ao tipo de software (por exemplo, sistemas de informação) e ao paradigma de desenvolvimento utilizado (por exemplo, orientação a objetos). Assim, durante a etapa de especialização do processo padrão, atividades poderão ser adicionadas ou modificadas, de acordo com o contexto para qual se está realizando a especialização.

A instanciação para projetos específicos consiste na adaptação de um processo especializado a um projeto, considerando-se as suas peculiaridades. Nesta etapa, são definidos o modelo de ciclo de vida, os métodos e as ferramentas que serão utilizadas no projeto, os recursos humanos e suas responsabilidades ao longo do processo e os artefatos (produtos) consumidos e gerados.

As três etapas de definição do processo consistem em adaptar o meta-modelo de processo para atender a objetivos específicos, através da análise das características organizacionais, de projetos e de produtos de software. Esta seção pretende fazer a sugestão de um levantamento das principais características organizacionais, de projetos e de produtos de software que devem ser observadas nas fases de definição do processo de software, as quais servem de base de conhecimento para a definição do processo no *ProDefiner*.

3.1. CARACTERÍSTICAS ORGANIZACIONAIS

No decorrer desta pesquisa foram identificados quatro grupos de Características Organizacionais que influenciam na definição do processo padrão. Esta pesquisa foi realizada mediante análise da literatura especializada e em entrevistas com grupos responsáveis pela definição do processo de software em organizações de Pernambuco, que relaciona estas possíveis características, assim como a sua influência na definição de um processo de software. Estes grupos de características estão relacionados a seguir.

a) Características do Processo de Software

Grupo que possibilita uma análise e uma especificação dos fatores relacionados ao processo de software da organização. Permite uma identificação da capacitação e maturidade da organização ao longo das atividades do ciclo de vida de software. São elas:

- **Modelo de Maturidade:** objetiva a definição de processos de software pelas organizações e a formulação de mecanismos visando melhorá-los continuamente.

Outro fator motivador está na importância da definição de processos de software como

base para garantir a qualidade dos produtos e melhorar a produtividade das equipes de desenvolvimento;

- **Nível de Maturidade:** é um platô bem definido em direção a um processo de software maduro, indicando uma capacitação de processo;
- **Institucionalização do Processo:** define se o processo especificado para uma organização encontra-se implantado seguindo seus mecanismos de implementação.

b) Características do Ambiente de Desenvolvimento

Este grupo possibilita uma análise dos parâmetros que definem o ambiente de desenvolvimento organizacional, ou seja, quão automatizado encontra-se o controle e o guia de desenvolvimento da organização entre os membros de sua equipe. São elas: Nível de contribuição na facilidade do gerenciamento do projeto; Nível de contribuição na facilidade do controle de versões; Nível de facilidade no uso do ambiente de desenvolvimento; Condição de suporte de uso a múltiplos usuários; Provimento de comunicação entre os membros da equipe de desenvolvimento; Suporte à cooperação da equipe de desenvolvimento; Suporta a múltiplas visões para diferentes usuários; Nível de automação da gerência do processo; Possibilidade de extensão; Possibilidade de integração entre as tecnologias utilizadas pela organização; Possibilidade de integração entre os paradigmas de desenvolvimento utilizados pela organização.

c) Características da Equipe

Aqui o foco está relacionado em identificar a experiência e adequação do conhecimento dos membros que compõem a equipe de desenvolvimento de software da organização no que tange a disciplina de Engenharia de Software, no contexto da definição e uso de padrões, procedimentos, ferramentas, atividades para um produto de software. São elas: Nível de conhecimento de Engenharia de Software; Nível de aplicação dos conhecimentos de Engenharia de Software; Adequação dos perfis dos membros às suas habilidades; Quantidade

de perfis agregados pelos membros no escopo do processo de software; Nível de treinamento nas tecnologias utilizadas; Nível de treinamento no ambiente de desenvolvimento utilizado; Nível de treinamento no processo de software institucionalizado.

d) Classificação da Organização de Desenvolvimento de Software

Uma vez definidas algumas das características que envolvem e especificam o contexto organizacional na atividade de desenvolvimento de software, faz-se necessário uma categorização destas organizações a fim de prover uma melhor caracterização do mapeamento dos processos. Este trabalho faz uso de quatro tipos de classificação para representar uma organização de desenvolvimento de software.

A primeira categorização diz respeito ao contexto de Fábricas de Software, onde analisando a literatura especializada [9], [8], percebem-se diferentes visões para sua definição. No entanto, a categorização definida por [8] foi adotada neste trabalho onde são apresentados quatro tipos de fábricas classificadas de acordo com o seu escopo de atuação ao longo das fases de desenvolvimento de um projeto de Software, a saber [18]:

- **Fábrica de Programas:** tem por objetivo principal codificar e testar programas de computador. No seu processo produtivo engloba praticamente as fases de construção e testes unitários;
- **Fábrica de Projetos de Software:** abrange além das atividades inerentes à fábrica de programas, fases como projeto conceitual, especificação lógica, projeto detalhado da solução, realização de testes de integração e de aceitação;
- **Fábrica de Projetos Físicos:** abrange as mesmas atividades encontradas na Fábrica de Projetos de Software exceto as atividades de projeto conceitual e especificação lógica;
- **Fábrica de Projetos Ampliada:** abrange além das atividades encontradas em uma Fábrica de Projetos de Software, soluções mais abrangentes de Tecnologia da

Informação como arquitetura da solução, modelagem do negócio, auditoria de qualidade e outras atividades de suporte ao desenvolvimento de projetos.

Além da classificação baseada no escopo de fornecimento de Fábricas de Software, essa pesquisa identificou outras classificações para as organizações que desenvolvem software. Baseado na pesquisa definida em [16] foram encontradas três formas de se caracterizar as organizações que desenvolvem software. Cada uma destas classificações possui um conjunto de variações. Baseado em [16] foram encontradas três formas de se caracterizar as organizações que desenvolvem software. Cada um dos itens dessas classificações foi definido e limitado de acordo com as necessidades do *ProDefiner*. Seguem, abaixo, essas classificações:

- **Classificação das empresas, segundo Atividades no Tratamento de Software:**
Pacote (packaged software); Sob encomenda (custom software); Embarcado; Para Internet; Para uso próprio.
- **Distribuição das empresas, segundo os principais tipos de software desenvolvidos:** Financeiro; Administração geral; Automação comercial; Contabilidade; Administração de recursos humanos; Página *Web*; Gestão integrada – ERP; Administração pública; Administração de serviços; Automação de escritórios; Automação industrial.
- **Atividades características das empresas em Tecnologia da Informação:**
Desenvolvimento de software; Consultoria e projetos; Treinamento; Distribuição ou Editoração de Software de terceiros; Manutenção e assistência técnica; Serviços de automação comercial; Distribuição ou revenda de produtos de hardware; Indústria de informática, telecomunicações ou automação; Serviços de processamento de dados; Serviços de automação industrial.

3.2 CARACTERÍSTICAS DE PROJETOS DE SOFTWARE

As características de projeto de software analisadas estão divididas em dois tipos:

a) Características do Desenvolvimento

As características de desenvolvimento caracterizam o projeto levando em consideração apenas os aspectos gerais ligados ao desenvolvimento do software, sem considerar fatores restritivos do projeto em questão e serão analisadas durante a fase de especialização do processo. São elas:

1. Tipo de Software

Tipos de softwares diferentes requerem que diferentes atividades sejam incorporadas ao processo de desenvolvimento a fim de garantir a qualidade do produto final. Um software interativo, por exemplo, requer que sejam incluídas no processo de desenvolvimento atividades que otimizem a interação homem-máquina. Com a ampla utilização da *World Wide Web*, e o crescimento do *e-business*, questões de segurança também passaram a ser prioritárias em sistemas interativos. Em outra abordagem, no desenvolvimento de sistemas de informação, torna-se fundamental garantir a adequação às regras de negócio por parte dos procedimentos implementados, além da precisão no registro das informações e questões de confiabilidade e recuperação após falhas [13]. A caracterização proposta neste trabalho foi adaptada de [13], que por sua vez se baseou na classificação proposta por [15]: Sistemas Operacionais; Sistemas de Missão-Crítica; Sistemas de Tempo-Real; Sistemas Interativos; Sistemas de Informação.

Mcmanus, em [15], estabeleceu recomendações (atividades) genéricas que devem ser incorporadas ao processo a fim de prover a garantia da qualidade do software em cada um dos tipos de software citados. No entanto o autor enfatiza que nem todas as recomendações devem ser aplicadas a qualquer projeto que envolva um determinado tipo de software. Uma análise prévia das características do projeto, e de fatores como estratégia da empresa e recursos disponíveis, deve ser realizada para adaptar as recomendações propostas.

2. Paradigma de Desenvolvimento

A escolha do paradigma está fortemente ligada ao tipo de software a ser desenvolvido. A escolha do paradigma implica em modificações na definição de algumas das atividades do modelo padrão, previamente definido. A atividade de análise, por exemplo, tem uma abordagem diferente do paradigma estrutural para o orientado a objetos. Os paradigmas de desenvolvimento suportados pelo *ProDefiner* são: Imperativo; Funcional; Orientado a Objetos; Lógico; Orientado a Aspectos; Orientado a Agentes.

A escolha do paradigma é feita em função da análise das restrições que devem ser atendidas em cada projeto e das vantagens oferecidas por cada paradigma. Um sistema operacional, por exemplo, exige eficiência e controle sobre os recursos de hardware, qualidades essas oferecidas pelo paradigma imperativo.

3. Tecnologia de Desenvolvimento

Representa a tecnologia a ser empregada no desenvolvimento do software, por exemplo tecnologia convencional de processamento de dados, tecnologia de sistemas baseados em conhecimento, etc. O domínio de aplicação e o propósito do sistema guiam a escolha da tecnologia de desenvolvimento a ser aplicada e tem grande impacto na escolha de um modelo de ciclo de vida e dos procedimentos a serem adotados no processo.

b) Características Relacionadas ao Projeto

As características do projeto caracterizam o projeto levando em consideração os aspectos ligados ao produto final que estará sendo desenvolvido bem como aos meios utilizados durante o desenvolvimento e serão analisadas durante a fase de instanciação do processo.

As características de projeto apresentadas a seguir são adaptados dos critérios propostos por [4], que por sua vez se baseou na abordagem proposta por [1], e servem como base na escolha do modelo de ciclo de vida apropriado a ser utilizado.

A seguir serão descritos os critérios citados e as características de projeto presentes em cada categoria. A classificação proposta pelo projeto para cada característica mencionada encontra-

se no Apêndice D. Vale ressaltar que cada uma dessas características, de forma particular, não influencia na definição do processo de software, elas devem ser analisadas de forma conjunta.

1. Critérios Relacionados aos Usuários: dizem respeito à experiência dos usuários e a qualidade da informação obtida através da comunicação com os mesmos. São eles:

- **Experiência dos usuários no domínio da aplicação:** nível de experiência do usuário acerca do problema a ser tratado pela aplicação desenvolvida, e sua familiaridade com a utilização de softwares que o auxiliem a resolvê-lo;
- **Facilidade dos usuários em expressar requisitos:** grau de precisão com que os usuários conseguem identificar as funcionalidades necessárias a uma aplicação para que a mesma seja satisfatória, atendendo as suas necessidades;
- **Grau de acesso aos usuários:** medida da disponibilidade dos usuários para interação e da qualidade do canal de comunicação utilizado;
- **Há nível de mudanças geradas no trabalho dos usuários:** aborda se há a possibilidade de as responsabilidades assumidas pelos usuários mudarem em função das habilidades requeridas nos projetos da organização, ou seja, se os usuários ocupam sempre o mesmo perfil.

2. Critérios Relacionados ao Problema: dizem respeito à complexidade quanto à especificação e resolução do problema e à experiência da organização. São eles:

- **Grau de maturidade do domínio da aplicação:** familiaridade da organização no desenvolvimento de projetos similares;
- **Complexidade do problema:** grau de dificuldade de resolução do problema, medido a partir da existência e de um plano de mitigação e utilizações anteriores do mesmo por parte da organização;
- **Frequência de mudanças nos requisitos:** aborda a frequência com que os requisitos mudam e as fases do ciclo de vida em que essas mudanças ocorrem;

- **Grau de magnitude das mudanças nos requisitos:** medida do impacto causado pela mudança dos requisitos na execução do projeto, principalmente quanto a cronograma e custo;
3. **Critérios Relacionados ao Produto:** abordam o tamanho e complexidade da aplicação a ser desenvolvida. São eles:
- **Tamanho da aplicação:** medida do tamanho da aplicação, dependendo da abordagem, ou seja, das métricas utilizadas no projeto em questão;
 - **Grau de complexidade da aplicação:** medida da complexidade da aplicação, dependendo de experiências obtidas e métodos usados para inferência;
 - **Grau de modularidade do produto:** medida do nível de reaproveitamento da aplicação desenvolvida.
4. **Critérios Relacionados aos Recursos:** trata da disponibilidade dos recursos necessários a execução do processo e a velocidade em que a demanda por novos recursos é atendida. O critério está dividido da seguinte forma: Disponibilidade de recursos humanos; Disponibilidade de recursos financeiros; Disponibilidade de recursos de software; Disponibilidade de recursos de hardware.
5. **Critérios Relacionados à Equipe de Desenvolvimento:** dizem respeito à composição da equipe de desenvolvimento, sua distribuição geográfica e experiência. São eles:
- **Tamanho da Equipe:** Número de integrantes que compõem a equipe de desenvolvimento;
 - **Localização Geográfica:** Distribuição espacial dos membros da equipe de desenvolvimento;
 - **Experiência da equipe de desenvolvimento no domínio da aplicação:** familiaridade dos membros da equipe em desenvolvimento de projetos no mesmo domínio de aplicação;

- **Experiência da equipe de desenvolvimento em engenharia de software:** nível experiência dos componentes da equipe em a aplicação/práticas de técnicas de engenharia de software;
 - **Nível de experiência da gerência:** grau de conhecimento acerca do gerenciamento de projetos, medido a partir do número de projetos gerenciados.
 - **Capacidade de gerenciamento de múltiplas equipes:** indica se a equipe de gerencia de projetos possui a habilidade e a experiência necessária para gerenciar, simultaneamente, múltiplas equipes de desenvolvimento.
6. **Critérios Relacionados ao Desenvolvimento:** estão relacionados às restrições e riscos técnicos identificados. São eles:
- **Há necessidade de entrega de produtos intermediários:** indica a necessidade de entrega de produtos a cada iteração do desenvolvimento;
 - **Há necessidade de o software ser colocado em uso rapidamente com funcionalidade total ou parcial:** indica se a aplicação será entregue iterativamente ou apenas quando da sua conclusão;
 - **Grau dos riscos técnicos:** mede a possibilidade de ocorrência de riscos relativos à utilização de tecnologias desconhecidas por parte da equipe de desenvolvimento;
 - **Necessidade de interface com sistemas existentes:** indica se existe a necessidade de comunicação com outros sistemas;
 - **Uso de tecnologia inovadora:** verifica se será necessária à utilização de tecnologias inovadores durante a execução do projeto;

3.3 CARACTERÍSTICAS DO PRODUTO DE SOFTWARE

Quanto ao produto de software as características baseiam-se em duas frentes:

a) Características de Qualidade da ISO/IEC 9126

A Norma ISO/IEC 9126 [12] define seis características de qualidade para produtos de software que, entretanto, podem não estar todas presentes em um produto ou, então não serem necessárias no mesmo grau. É, então, necessário determinar que características devem estar presentes no produto e em que grau. As características são: Funcionalidade; Confiabilidade; Usabilidade; Eficiência; Manutenibilidade; Portabilidade.

A avaliação do grau em que cada característica de qualidade deve estar presente em um produto de software deve ser determinada através de avaliações de seus usuários. Oliveira, em [17], definiu cinco graus de importância para as características de qualidade definidas pela Norma ISO/IEC 9126: Sem relevância, Pouco relevante, Relevante, Muito relevante e Imprescindível. A técnica usada para determinar este grau está descrita no Apêndice D, que toma como base o trabalho de [17].

b) Nível de Garantia de Qualidade do Produto

De acordo com Boegh et al., em [5], os produtos de software podem ser classificados em níveis considerando-se o dano causado por falhas e, portanto, o grau necessário de garantia da qualidade. Tais níveis variam de A a D, sendo A o nível mais alto e D o nível mais baixo (Tabela 1). Os níveis definem o grau de rigor com que a qualidade do produto deverá ser avaliada.

Tabela 1. Classificação do Projeto de acordo com o Nível de Avaliação [5]

Nível	Ambiente	Pessoas	Economia	Aplicação
D	Pequeno dano a propriedade	Sem risco para pessoas	Perda econômica desprezível	Lazer Uso doméstico
C	Dano a propriedades	Poucas pessoas mutiladas	Perda econômica significativa	Alarme de incêndio Controle de processos
B	Dano recuperável ao ambiente	Risco para vidas humanas	Grande perda econômica	Sistemas médicos Sistemas financeiros
A	Dano irrecuperável ao ambiente	Muitas pessoas mortas	Desastre financeiro	Controle de trens Sistemas nucleares

Boegh, em [5], propõe que com base na avaliação das características de qualidade previstas na norma ISO/IEC 9126 em relação ao produto a ser desenvolvido, técnicas de avaliação destas

características possam ser sugeridas. Tal seleção baseia-se no grau de importância atribuído a cada característica de qualidade pelo engenheiro de software e pelos usuários do produto final, o que irá influenciar no rigor das técnicas a serem aplicadas.

De acordo com o nível da aplicação, técnicas de avaliação para as características de qualidade da Norma ISO/IEC 9126 são sugeridas.

Quando o projeto é caracterizado, o software é classificado em um nível de acordo com o tipo de aplicação e impacto em caso de falha. Baseado neste nível, [3] propõe que sejam utilizadas técnicas de avaliação da qualidade apropriadas para o projeto. Quanto maior for o risco e o impacto de dano, mais rígidas são as técnicas de avaliação. A Tabela 2 apresenta as técnicas indicadas para avaliação de cada característica de qualidade, de acordo com o nível do produto. As técnicas são cumulativas, ou seja, um projeto classificado como sendo de nível A deverá contar com as técnicas indicadas para este nível e todos os níveis inferiores (B, C e D).

Tabela 2 Técnicas de Avaliação de acordo com o nível do projeto [5]

Característica	Nível D	Nível C	Nível B	Nível A
Funcionalidade	Teste Funcional	+ Inspeção de Documentos	+ Teste de Componentes	+ Prova Formal
Confiabilidade	Facilidades da Linguagem de Programação	+ Análise de Tolerância a Falhas	+ Modelos de Crescimento de Confiabilidade	+ Prova Formal
Usabilidade	Inspeção da Interface com o Usuário	+ Aderência a Padrões de Interface	+ Teste de Laboratório	+ Modelos Mentais do Usuário
Eficiência	Medição do Tempo de Execução	+ Benchmark	+ Análise da Complexidade de Algoritmos	+ Análise de Desempenho
Manutenibilidade	Inspeção de Documentos	+ Análise Estática	+ Análise do Processo de Desenvolvimento	+ Avaliação de Rastreabilidade
Portabilidade	Análise da Instalação	+ Aderência a Normas de Programação	+ Avaliação das Restrições do Ambiente	+ Avaliação do Projeto de Programas

4. INFLUÊNCIAS DAS CARACTERÍSTICAS NA DEFINIÇÃO DO PROCESSO DE SOFTWARE

O levantamento das características organizacionais e de projetos de software definidas na seção 3, como já mencionado, auxilia o grupo de SPEG na definição do processo de software

segundo os níveis de definição (definição do processo padrão, especialização e instanciação do processo) especificados na estrutura do *ImPProS*. Este auxílio é feito mediante sugestão dos componentes do processo de software (atividades, modelos de ciclo de vida, técnicas, etc.) que cada característica provê aos membros do SEPG. Baseado em pesquisas e em aprendizados obtidos na definição de processos de software, a Tabela 3 permite a visualização dessas sugestões.

Tabela 3 Sugestões de Componentes do Processo de Software a partir da Análise das Características Organizacionais, de Projetos e de Produtos de Software

Nível de Definição do Processo	Características	Sugestão dos Componentes do Processo de Software
Definição do Processo Padrão	Modelo de Maturidade.	Níveis de Maturidade
	Nível de Maturidade.	Processos do Ciclo de Vida de Software relacionados e mapeados ao Nível de Maturidade; Atividades dos Processos de Ciclo de Vida de Software relacionadas e mapeadas ao Nível de Maturidade.
	Institucionalização do Processo; Características do Ambiente de Desenvolvimento; Características da Equipe.	Tipo de Organização.
	Tipo de Organização.	Atividades dos Processos de Ciclo de Vida de Software relacionadas ao Tipo de Organização.
Especialização do Processo	Tipo de Software.	Paradigma de Desenvolvimento; Tecnologia de Desenvolvimento; Atividades dos Processos de Ciclo de Vida de Software relacionadas ao Tipo de Software.
Instanciação do Processo	Critérios relacionados aos Usuários; Critérios relacionados ao Problema; Critérios relacionados ao Produto; Critérios relacionados aos Recursos; Critérios relacionados à Equipe de Desenvolvimento; Critérios relacionados ao Desenvolvimento.	Modelo de Ciclo de Vida do Processo de Software.
	Atributos de Qualidade relevantes ao Produto segundo a ISO/IEC 9126.	Atividades dos Processos de Ciclo de Vida de Software relacionadas aos Atributos de Qualidade do Produto.
	Nível de Garantia de Qualidade definido ao Produto.	Técnicas de Avaliação de Qualidade para as Atividades do Processo de Ciclo de Vida de Software.
	Paradigma de Desenvolvimento; Tecnologia de Desenvolvimento.	Procedimentos das Atividades do Processo de Ciclo de Vida de Software

5. CONSIDERAÇÕES FINAIS

Durante a definição de processos de software, a análise e a especificação da formalidade das características adequadas a cada projeto de software e a organização, onde este será institucionalizado, é um fator determinante para se alcançar o sucesso em um projeto. Neste trabalho foram apresentadas as características de projeto de software e organizacionais que devem ser analisadas a fim de se definir um processo de desenvolvimento aderente às necessidades do que se espera.

Vale mencionar que em [3], [7], encontra-se especificado um modelo que auxilia a automação da definição de processos e fornece um mecanismo para incorporação de novas características ao ambiente *ImPProS*. Este modelo inicialmente foi concebido mediante especificação de um fluxo de atividades e posteriormente uma ferramenta foi desenvolvida e integrada à estrutura do *ImPProS*, para prover a incorporação de novas características, aprendizado de sugestões de componentes do processo de software, e a análise e a inferência de componentes condizentes às características definidas pelos membros do SEPG.

É fato, também, esclarecer que todas as características relatadas neste trabalho são frutos de pesquisas na literatura especializada e de lições aprendidas ao longo da definição de processos de software em organizações, o que valida os seus uso.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] ALEXANDER, L. C., DAVIS, A. M., **Criteria for Selecting Software Process Models**, In: Proceedings of the Fifteenth Annual International Computer Software & Applications Conference – COMPSAC 91, pp. 521-528, Tokyo, Japan, 1992.
- [2] BALDUINO, R., **Implementação de um processo de desenvolvimento de software: uma abordagem passo-a-passo**, Rational Software White Paper, 2002.
- [3] BARBOSA, I. M., **Análise de Características de Projetos de Software para a Definição de Processo de Software**, Trabalho de Graduação apresentado ao CIn/UFPE, orientador Prof. Alexandre Vasconcelos, Recife-PE, 2005.
- [4] BERGER, P. M., **Instanciação de Processos de Software em Ambientes Configurados na Estação TABA**, Orientadora Ana Regina Cavalcanti da Rocha. Dissertação de Mestrado, COPPE/UFRJ, Rio de Janeiro-RJ, 2003.
- [5] BOEGH, J., HAUSEN, H. L., WELZEL, D. **A Practitioners Guide to Evaluation of Software**, In: Software Engineering Standards Symposium, Brighton, Inglaterra, 1993.
- [6] CHRISSIS, M. B., KONRAD, M. and SHRUM, S., **CMMI Guidelines for Process Integration and Product Improvement**, Addison-Wesley, 2003.
- [7] CUNHA, M. B. F. L., **Análise das Características Organizacionais para a Definição**

- de Processo de Software**, Trabalho de Graduação apresentado ao CIn/UFPE, orientador Prof. Alexandre Vasconcelos, Recife-PE, 2005.
- [8] FERNANDES, A. A., TEIXEIRA, D. S., **Fábrica de Software: Implantação e gestão de Operações**, Atlas, São Paulo, 2004.
 - [9] GREENFIELD J., SHORT, K., **Software Factories – Assembling Applications with Patterns, Models, Frameworks and Tools**, OOPSLA'03, ACM, Califórnia, 2003.
 - [10] ISO/IEC TR 15504, Parts 1-9, **Information Technology – Software Process Assessment**, International Organization for Standardization, 1998.
 - [11] ISO/IEC TR 12207, **Amendment: Information Technology – Amendment to ISO/IEC 12207**, PDAM 3 version, 2000.
 - [12] ISO/IEC TR 9126, **Software engineering – product quality**, International Organization for Standardization, 2002.
 - [13] MACHADO, L. F., **Modelo para Definição de Processos de Software na Estação Taba**, Orientadora Ana Regina Cavalcanti Rocha, Tese de Mestrado, COPPE/UFRJ, 2000.
 - [14] MCFEELEY, B., **IDEALSM: A User's Guide for Software Process Improvement**, Software Engineering Institute Handbook. Carnegie Mellon University. CMU/SEI-96-HB-001, 1996.
 - [15] MCMANUS, J. I., **How does Software Quality Assurance Fit In?**, In: Shulmeyer, G. G., Mcmanus, J. I. (eds), Handbook of Software Quality Assurance, 3 ed., chapter 2, Prentice Hall PTR, 1999.
 - [16] MCT/SEPIN, **Qualidade no Setor de Software Brasileiro**, DSI/CGSA, Brasília, [ISSN 1518-112X], disponível em: <http://www.mct.gov.br/sepin>, 2000.
 - [17] OLIVEIRA, K. M., **Modelo para Construção de Ambientes de Desenvolvimento de Software Orientados a Domínio**, Tese de D. Sc. , COPPE/UFRJ, Rio de Janeiro, R.J., Brasil, 1999.
 - [18] OLIVEIRA, S. R. B., ROCHA, T. A., VASCONCELOS, A. M. L., **Adequação de Processos para Fábricas de Software**, VI SIMPROS – Simpósio Internacional de Melhoria de Processo de Software. São Paulo-SP, 2004.
 - [19] OLIVEIRA, S., Vasconcelos, A., Rouiller, A. C., **Uma Proposta de um Ambiente de Implementação de Processo de Software**, Artigo publicado na Revista InfoComp – Revista de Ciência da Computação da UFLA – vol. 4, n. 1, Lavras-MG, 2005
 - [20] PFLEEGER, S. L., **Software Engineering Theory and Practice**, Prentice Hall PTR, 1998.
 - [21] REIS, C. A. L., **Uma Abordagem Flexível para Execução de Processos de Software Evolutivos**, Orientador Daltro José Nunes. Tese de Doutorado, Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre-RS, 2003.
 - [22] ROCHA, A. R. C., MALDONADO, J. C. and WEBER, K. C., **Qualidade de software: teoria e prática**, São Paulo: Prentice-Hall, 2001.