A Experiência na Definição de um Processo Padrão Baseado no Processo Unificado

RICARDO DE ALMEIDA FALBO
Departamento de Informática - UFES
Av. Fernando Ferrari, s/n, Vitória – ES
falbo@inf.ufes.br

Abstract

This paper presents an Organizational Software Process for object-oriented development. This process was defined based on the Rational Unified Process (RUP). Since RUP does not address directly many quality issues, we included some activities considering ISO 12207 and ISO 9000-3 standards.

Palavras-chave: Processo de Software, Qualidade de Processo, Processo para Desenvolvimento Orientado a Objetos, Processo Unificado Rational (RUP).

1. Introdução

Na busca por qualidade e produtividade na construção de sistemas de software, profissionais têm se dado conta de que um de seus principais problemas é a falta de habilidade para gerenciar o processo de desenvolvimento. Um processo de software pode ser visto como o conjunto de atividades, métodos, práticas e transformações que guiam pessoas na produção de software. Um processo eficaz deve, claramente, considerar as relações entre as atividades, os artefatos produzidos no desenvolvimento, as ferramentas e os procedimentos necessários e a habilidade, o treinamento e a motivação do pessoal envolvido.

O processo de desenvolvimento de software não pode ser definido de forma universal. Para ser eficaz e conduzir à construção de produtos de boa qualidade, um processo deve ser adequado ao domínio da aplicação e ao projeto específico. Deste modo, processos devem ser definidos caso a caso, considerando-se as especificidades da aplicação, a tecnologia a ser adotada na sua construção, a organização onde o produto será desenvolvido e o grupo de desenvolvimento.

Ainda que diferentes projetos requeiram processos com características específicas para contemplar suas peculiaridades, conhecendo a tecnologia de desenvolvimento e a

organização, é possível estabelecer um modelo básico de processo, dito um processo padrão, a ser configurado e adaptado para os projetos.

Este trabalho discute a experiência na definição de um processo padrão para o desenvolvimento de software orientado a objetos no Departamento de Informática da Companhia Siderúrgica de Tubarão – CST. A seção 2 apresenta brevemente o histórico do trabalho. Na seção 3, são explorados os conceitos fundamentais que nortearam a definição do processo padrão, discutido na seção 4. A seção 5 trata de aspectos inerentes à avaliação da qualidade. Finalmente, na seção 6, são apresentadas as conclusões deste trabalho.

2. Histórico

A Companhia Siderúrgica de Tubarão – CST, localizada em Vitória – ES, é uma forte empresa do ramo siderúrgico, que tem procurado, cada vez mais, aumentar sua competitividade no mercado nacional e internacional. Dadas as características bastante peculiares de suas necessidades de informação e automação, possui um Departamento de Informática responsável pelo desenvolvimento e gestão de sistemas de software, congregando equipes atuando em áreas administrativas, tais como pessoal e gestão de contratos, e produtivas, tais como manutenção e produção.

Até meados de 1999, o desenvolvimento de sistemas na CST estava centrado em uma plataforma cliente-servidor, utilizando técnicas estruturadas. Com a decisão de se implantar uma nova linha produtiva – Laminador de Tiras a Quente (LTQ) – em parceria com uma empresa japonesa, surgiu a necessidade de uma migração para a tecnologia de objetos, utilizada por esta última.

Uma vez que, até então, o desenvolvimento de sistemas apresentava vários problemas decorrentes da falta de uma sistemática padrão para essa atividade, tais como estimativas irreais, falta de padronização e gasto excessivo com manutenção, decidiu-se que, para a implantação da nova tecnologia, seria adotada uma estratégia focada na qualidade. Assim, o primeiro passo consistiu na definição de um processo padrão para o desenvolvimento de aplicações orientadas a objetos. Esse processo, discutido na seção 4, foi elaborado tendo por base o Processo Unificado (Rational Unified Process – RUP) [1], já que este é um processo centrado na tecnologia de objetos e já relativamente experimentado na prática. Algumas práticas, contudo, foram adaptadas, considerando as particularidades da organização e sua cultura. A definição do processo durou aproximadamente seis meses e contou com intensa

participação de vários profissionais da CST, com o intuito de se chegar a um processo enxuto e factível, e não a um processo a ser engavetado e não utilizado. Como um segundo passo, um grande programa de treinamento foi estabelecido, tendo por base o processo padrão definido. Atualmente, o processo está sendo experimentado em vários projetos, com apoio de consultores, visando afiná-lo em um procedimento de melhoria contínua.

3. Processo de Desenvolvimento de Software

Um processo de desenvolvimento de software pode ser visto como um guia para o desenvolvimento e deve estabelecer: as atividades a serem realizadas durante o desenvolvimento de software, sua estrutura e organização (decomposição e precedência de atividades), incluindo a definição de um modelo de ciclo de vida; os artefatos requeridos e produzidos por cada uma das atividades do processo; os procedimentos (métodos, técnicas e normas) a serem adotados na realização das atividades; os recursos necessários para a realização das atividades, dentre eles recursos humanos, recursos de hardware e recursos de software, incluindo as ferramentas a serem utilizadas para apoiar a aplicação dos procedimentos na realização das atividades; e roteiros para a elaboração dos principais documentos (artefatos) gerados no desenvolvimento.

De maneira geral, o ciclo de vida de um software envolve, pelo menos, as atividades de planejamento, levantamento de requisitos, análise, projeto, implementação, testes, implantação, operação e manutenção.

Uma vez que o software é sempre parte de um sistema (ou negócio) maior, o trabalho começa pelo estabelecimento dos requisitos para todos os elementos do sistema e, na seqüência, procede-se a alocação para software de algum subconjunto destes requisitos. Esta etapa é a Engenharia de Sistemas e antecede a todas as demais relacionadas.

3.1 – O Processo Unificado

O Processo Unificado proposto pela Rational (Rational Unified Process – RUP) [1] foi criado para apoiar o desenvolvimento orientado a objetos, fornecendo uma forma sistemática para se obter reais vantagens no uso da Linguagem de Modelagem Unificada (*Unified Modeling Language* – UML). De fato, ele não é exatamente um processo: é uma infra-estrutura genérica de processo que pode ser especializada para uma ampla classe de sistemas

de software, para diferentes áreas de aplicação, tipos de organização, níveis de competência e tamanhos de projetos.

O RUP está fundamentado em três princípios básicos: orientação a casos de uso, arquitetura e iteração. Ele é dito *dirigido a casos de uso*, pois são os casos de uso que orientam todo o processo de desenvolvimento. Com base no modelo de casos de uso, são criados uma série de modelos de análise, projeto e implementação, que realizam estes casos de uso. É *centrado em arquitetura*, pois defende a definição de um esqueleto para a aplicação (a arquitetura), a ganhar corpo gradualmente ao longo do desenvolvimento. Finalmente, o RUP é iterativo e incremental, oferecendo uma abordagem para particionar o trabalho em porções menores ou mini-projetos. Esses três conceitos são igualmente importantes. A arquitetura provê a estrutura para guiar o desenvolvimento do sistema em iterações, enquanto os casos de uso definem as metas e conduzem o trabalho de cada iteração [1].

O ciclo de vida adotado no RUP é tipicamente evolutivo. Contudo, uma forma de organização em fases é adotada para comportar os ciclos de desenvolvimento, permitindo uma gerência mais efetiva de projetos complexos. Ao contrário do tradicionalmente definido como fases na maioria dos modelos de ciclo de vida – planejamento, levantamento de requisitos, análise, projeto, implementação e testes, são definidas fases ortogonais a estas, a saber [1]:

- Concepção: nesta fase, é estabelecido o escopo do projeto e suas fronteiras, determinando os principais casos de uso do sistema. Esses casos de uso devem ser elaborados com a precisão necessária para se proceder estimativas de prazos e custos. As estimativas devem ser globais para o projeto como um todo e detalhadas para a fase seguinte. Assim, a ênfase nesta etapa recai sobre o planejamento e, por conseguinte, é necessário levantar requisitos do sistema e preliminarmente analisá-los. Ao término dessa fase, são examinados os objetivos do projeto para se decidir sobre a continuidade do desenvolvimento;
- Elaboração: o propósito desta fase é analisar mais refinadamente o domínio do problema, estabelecer uma arquitetura de fundação sólida, desenvolver um plano de projeto para o sistema a ser construído e eliminar os elementos de projeto que oferecem maior risco. Embora o processo deva sempre acomodar alterações, as atividades da fase de elaboração asseguram que os requisitos, a arquitetura e os planos estão suficientemente estáveis e que os riscos estão suficientemente mitigados, de modo a se poder prever com precisão os custos e prazos para a conclusão do desenvolvimento.

- Construção: durante esta fase, um produto completo é desenvolvido de maneira iterativa e incremental, para que esteja pronto para a transição à comunidade usuária.
- Transição: nesta fase, o software é disponibilizado à comunidade usuária. Após o produto ter sido colocado em uso, naturalmente surgem novas considerações que vão demandar a construção de novas versões para permitir ajustes do sistema, corrigir problemas ou concluir algumas características que foram postergadas.

É importante realçar que dentro de cada fase, um conjunto de iterações, envolvendo planejamento, levantamento de requisitos, análise, projeto e implementação e testes, é realizado. Contudo, de uma iteração para outra e de uma fase para a próxima, a ênfase sobre as várias atividades muda, como mostra a figura 1, em que a cor preta indica grande ênfase, enquanto a cor branca indica muito pouca ênfase. Na fase de concepção, o foco principal recai sobre o entendimento dos requisitos e a determinação do escopo do projeto (planejamento e levantamento de requisitos). Na fase de elaboração, o enfoque está na captura e modelagem dos requisitos (levantamento de requisitos e análise), ainda que algum trabalho de projeto e implementação seja realizado para prototipar a arquitetura, evitando certos riscos técnicos. Na fase de construção, o enfoque concentra-se no projeto e na implementação, visando evoluir e rechear o protótipo inicial, até obter o primeiro produto operacional. Finalmente, a fase de transição concentra-se nos testes, visando garantir que o sistema possui o nível adequado de qualidade. Além disso, usuários devem ser treinados, características ajustadas e elementos esquecidos adicionados.

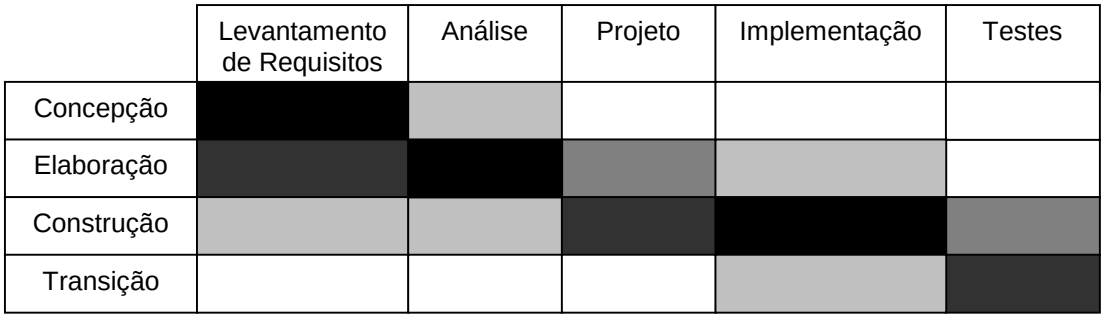

Figura 1 – A ênfase principal de cada uma das fases.

Além dos conjuntos de iterações em cada fase, as fases em si podem ocorrer de forma iterativa. Como mostra a figura 2, elas não necessariamente têm a mesma duração. O esforço realizado em cada uma varia fortemente em função das circunstâncias específicas do projeto.

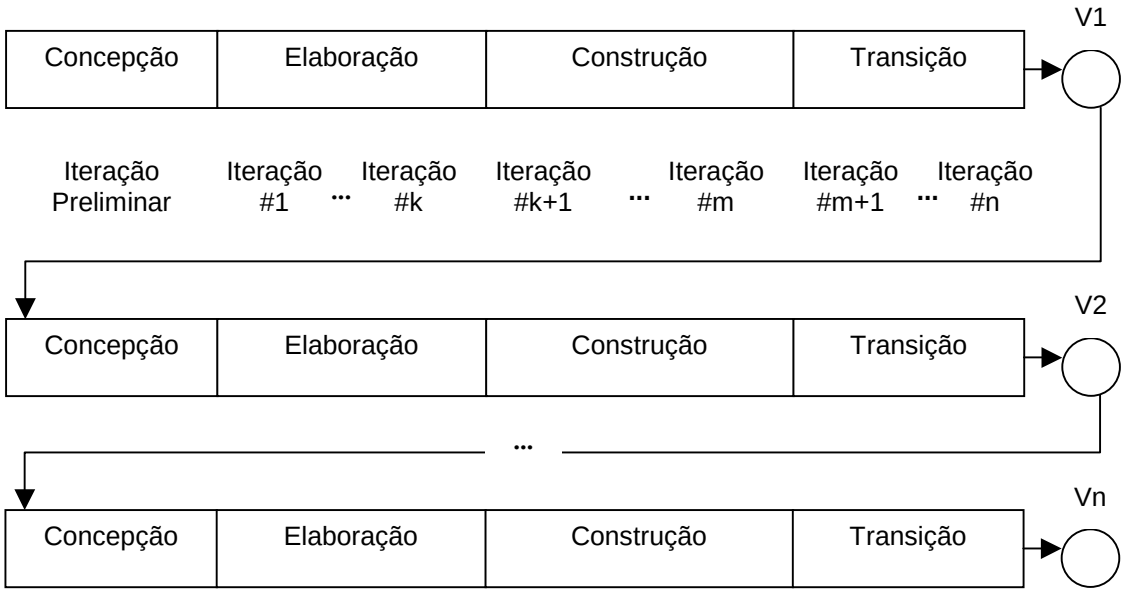

Figura 2 – O Modelo de Ciclo de Vida proposto no RUP.

4. Processo Padrão para Desenvolvimento Orientado a Objetos

O processo padrão proposto para o desenvolvimento de sistemas orientados a objetos na CST procurou incorporar as principais características do RUP, adequando-as a essa organização. O modelo de ciclo de vida foi adotado, assim como os princípios básicos. Entretanto, muitos aspectos relacionados a um processo de qualidade não são devidamente tratados pelo RUP e, portanto, foi necessário adaptá-lo, procurando adicionar atividades, principalmente, de gerência e de garantia da qualidade. Para tal, tomamos como base as normas ISO 12207 [2] e ISO 9000-3 [3].

Para cada uma das atividades do ciclo de vida, foram definidos sub-atividades, incluindo decomposição e precedência entre elas (fluxo de trabalho), métodos, técnicas e roteiros aplicáveis, recursos necessários, artefatos requeridos (insumos) e produzidos (produtos), e marcos de projeto e pontos de controle para acompanhamento do projeto e avaliação da qualidade. A tabela 1 apresenta um resumo das definições feitas para as atividades de levantamento de requisitos e análise. Escolhemos apresentar essas definições por estarem diretamente relacionadas à tecnologia de objetos. Na tabela, não constam os recursos de software definidos, mas, de maneira geral, são necessários um editor de textos, para todas as atividades em que há produção de documentos textuais e uma ferramenta CASE com suporte à UML, para as atividades de modelagem.

Tabela 1 – Principais características do Processo Padrão Definido.

Atividades e sub-atividades	**Métodos, Técnicas e Roteiros Aplicáveis**	**Recursos Humanos Necessários**	**Insumos**	**Produtos**
1. Levantamento de Requisitos	Modelagem de Casos de Uso		Plano de Projeto	Modelo de Casos de Uso
1.1 – Elicitação e Documentação de Requisitos	Técnicas de Levantamento de Informação	Analista de Sistemas, Especialistas Usuários	Plano de Projeto	Modelo de Casos de Uso
1.2 – Avaliação do Documento de Requisitos	Revisões Técnicas, Walkthrough	Gerente de Projeto, Analista de Sistemas, Especialistas Usuários	Modelo de Casos de Uso	Laudo de Revisão, Modelo de Casos de Uso Avaliado
1.3 – Avaliação da Especificação Final de Requisitos	Revisões Técnicas, Walkthrough	Gerente de Desenvolvimento, Gerente de Projeto, Analista de Sistemas, Gerente Usuário, Especialistas Usuários	Modelo de Casos de Uso Avaliado	Laudo de Revisão, Modelo de Casos de Uso Aprovado
2 – Análise	Método de Análise OO		Modelo de Casos de Uso	Modelo de Análise
2.1 – Identificação de Classes	Técnicas de Levantamento de Informação	Analista de Sistemas, Especialistas Usuários	Modelo de Casos de Uso	Lista de Potenciais Classes
2.2 – Definição de Hierarquias de Generalização / Especialização	Técnicas de Levantamento de Informação	Analista de Sistemas, Especialistas Usuários	Modelo de Casos de Uso, Lista de Potenciais Classes	Hierarquias de Generalização / Especialização
2.3 – Identificação de Subsistemas	Técnicas de Levantamento de Informação	Analista de Sistemas, Especialistas Usuários	Modelo de Casos de Uso, Classes e Hierarquias	Diagrama de Pacotes
2.4 – Identificação de Relacionamentos e Atributos	Técnicas de Levantamento de Informação	Analista de Sistemas, Especialistas Usuários	Modelo de Casos de Uso, Classes e Hierarquias	Diagrama de Classes (parte estática)
2.5 – Avaliação da Parte Estática do Modelo de Análise	Revisões Técnicas, Walkthrough	Gerente de Projeto, Analista de Sistemas, Especialistas Usuários	Modelo de Casos de Uso, Diagrama de Classes (parte estática)	Laudo de Revisão, Diagrama de Classes (parte estática) Avaliado
2.6 – Determinação do Comportamento	Técnicas de Modelagem	Analista de Sistemas, Especialistas Usuários	Casos de Uso, Diagrama de Classes (parte estática)	Diagramas de Interação, de Estados e de Classes
2.7 – Definição das operações		Analista de Sistemas	Diagrama de Classes, Diagrama de Interação	Operações Especificadas
2.8 – Avaliação da parte dinâmica do Modelo de Análise	Revisões Técnicas, Walkthrough	Gerente de Projeto, Analista de Sistemas, Especialistas Usuários	Modelo de Casos de Uso, Diagrama de Classes, de Interação e de Estado	Laudo de Revisão, Diagramas Avaliados
2.9 – Elaboração do Documento de Especificação de Análise	Roteiro de Documento de Especificação de Análise	Analista de Sistemas, Gerente de Projeto	Diagramas Avaliados	Documento de Especificação de Análise

2.10 – Avaliação do Documento de Especificação de Análise	Revisões Técnicas, Walkthrough	Gerente de Desenvolvimento, Gerente de Projeto, Analista de Sistemas, Gerente Usuário, Especialistas Usuários	Documento de Especificação de Análise	Documento de Especificação de Análise Aprovado (Modelo de Análise)

Além das atividades do ciclo de vida, foram definidas atividades de apoio [2,3], agrupadas em duas categorias: atividades de gerência e atividades de garantia da qualidade. As atividades de gerência incluem:

- **Definição de um Processo para o Projeto Corrente**: Todo projeto deve ser desenvolvido com base em um processo definido para ele. Uma vez que projetos têm características diferentes, não é possível definir um processo universal aplicável a qualquer projeto. Havendo um processo padrão, a tarefa de definição do processo pode ser vista como uma tarefa de adaptação do modelo às reais necessidades do projeto.
- **Acompanhamento e Controle do Projeto**: O acompanhamento pode ser realizado de muitas maneiras diferentes: realizando reuniões periódicas sobre a situação do projeto, avaliando os resultados de todas as revisões conduzidas ao longo do processo, determinando se os marcos do projeto foram atingidos até as datas programadas, etc. O controle é empregado pelo Gerente de Projeto para administrar os recursos do projeto, enfrentar problemas e dirigir o pessoal do projeto. Quando um problema for diagnosticado, recursos adicionais podem ser concentrados na área do problema, o pessoal pode ser redistribuído ou o cronograma do projeto redefinido.
- **Monitoração e Gerência de Riscos**: Da mesma forma que o projeto como um todo deve ser acompanhado e controlado, os riscos do projeto têm de ser monitorados e gerenciados. O conjunto de riscos deve ser revisado em intervalos regulares, reavaliando, para cada risco, as possíveis mudanças em sua probabilidade e em seu impacto. Pode ser necessário adicionar novos riscos, remover outros ou alterar a posição relativa de alguns. O gerente de projeto deve monitorar fatores que podem prover uma indicação de se o risco está se tornando mais ou menos provável, e a eficácia dos passos de abrandamento de riscos. Finalmente, quando os esforços de abrandamento falharem e o risco tornar-se uma realidade, é necessário gerenciá-lo de modo a resolver os problemas gerados.

- **Provisão Oportuna de Infra-estrutura**: Deve-se garantir que a infra-estrutura necessária para a realização de uma atividade estará disponíveis em tempo oportuno, de modo a não provocar atrasos. Essa infra-estrutura deve ser mantida, monitorada e modificada quando necessário, para garantir que ela continua a atender às demandas da atividade.
- **Treinamento**: Diversos tipos de treinamento podem ser necessários em um projeto de software, tais como treinamento da equipe de desenvolvimento em um método de construção ou técnica de avaliação da qualidade, treinamento dos usuários, etc.

As atividades de garantia da qualidade incluem:

- **Elaboração de um Plano de Qualidade para o Projeto**: As atividades de garantia da qualidade executadas pela equipe de engenharia de software e pelo grupo de Controle e Garantia da Qualidade de Software (CGQS) são governadas por este plano, que deve identificar: avaliações a serem executadas, auditorias e revisões a serem realizadas, padrões aplicáveis ao projeto, procedimentos para divulgação e depuração de erros e documentos a serem produzidos nas atividades de garantia da qualidade.
- **Revisões de software**: São aplicadas em vários pontos durante o processo de software e servem para descobrir erros e desvios em relação ao processo definido e aos padrões estabelecidos.
- **Gerência de Configuração do Software**: Busca identificar e controlar alterações, garantir que uma alteração está sendo adequadamente implementada e relatar a alteração às partes interessadas. Toda informação produzida como parte do processo de software torna-se parte da configuração do software. Um artefato só passa a fazer parte da configuração quando é aprovado.

Visando apoiar a adaptação do Processo Padrão a projetos reais, foram definidas diretrizes, principalmente, para apoiar a instanciação do modelo de ciclo de vida (escolha do número de passadas pelas quatro fases e do número de iterações em cada fase, entre outros), para proceder avaliações e para realizar medições. No que concerne à utilização do modelo de ciclo de vida, a tabela 2 apresenta o conjunto de artefatos a serem produzidos em cada uma das fases, o que, em última instância, favorece o entendimento da ênfase a ser dada em cada uma delas. Os aspectos relacionados à avaliação da qualidade ao longo do processo e às métricas a serem coletadas inicialmente são tratados na próxima seção.

Tabela 2 – Artefatos produzidos em cada fase do desenvolvimento.

Fase	Artefatos
Construção	Plano de Projeto (global para o projeto e detalhado para a fase de elaboração) Documento de Especificação de Requisitos (Modelo de Casos de Uso) Diagrama de Classes (esboço preliminar)
Elaboração	Plano de Projeto (revisado e detalhado para a fase de construção) Documento de Especificação de Requisitos revisado Documento de Especificação de Análise e Projeto da Arquitetura (Diagrama de Pacotes e esboço dos Diagramas de Classes para cada um dos pacotes)
Construção	Plano de Projeto (revisado e detalhado para a fase de transição) Documentos de Especificação de Requisitos e de Análise revisados Documento de Especificação de Projeto, Documento de Implementação Classes Codificadas e Bases de Dados Implementadas, Unidades e Integração Testadas
Transição	Plano de Projeto revisado Documentos de Especificação de Requisitos, de Análise e de Projeto revisados Documento de Implementação, Classes Codificadas e Bases de Dados revisados Sistema Testado e Implantado

5. Avaliação da Qualidade

Conforme discutido na seção 4, nos marcos do projeto (MP), que tipicamente ocorrem ao término de cada uma das atividades de desenvolvimento, e pontos de controle (PC), é necessário realizar uma avaliação da qualidade. A tabela 3 relaciona os documentos a serem revisados e avaliados ao término de cada uma das atividades do desenvolvimento.

Tabela 3 – Documentos a serem avaliados ao término de cada atividade.

Atividade	Documento a ser Avaliado
Planejamento (MP)	Plano de Projeto
Levantamento de Requisitos (PC)	Modelo de Casos de Uso
Levantamento de Requisitos (MP)	Documento de Especificação de Requisitos
Análise – Parte Estática (PC)	Diagrama de Classes / Pacotes Dicionário de Dados – Atributos
Análise – Parte Dinâmica (PC)	Diagramas de Seqüência Diagramas de Transição de Estados Definição das Operações
Análise (MP)	Documento de Especificação de Análise
Projeto Arquitetural (PC)	Diagrama de Pacotes da Arquitetura
Projeto Detalhado (PC)	Diagramas de Classes Dicionário de Dados Diagramas de Seqüência Diagramas de Transição de Estados Especificação dos Métodos
Projeto (MP)	Documento de Especificação de Projeto
Implementação – Codificação (PC)	Código Fonte
Implementação (MP)	Documento de Implementação
Implantação (MP)	Sistema

Uma vez que os referidos documentos tenham sido aprovados nos respectivos marcos, eles deverão ser submetidos à gerência de configuração. Enquanto um produto não tiver sido

aprovado em um marco de projeto, ele não deverá ser considerado um item de configuração.

Para que estimativas em desenvolvimentos futuros sejam mais precisas do que as realizadas atualmente, é preciso ter uma base de dados históricos. Assim, é fundamental coletar dados sobre o desenvolvimento e computar métricas capazes de guiar futuras estimativas. Tendo em mente este objetivo, em um primeiro momento, a intenção é concentrar esforços no cálculo apenas de métricas de projeto, isto é, aquelas métricas que possam ser usadas por gerentes de projeto para acompanhar mais de perto seus projetos. Métricas baseadas no tamanho do software têm se mostrado eficientes neste contexto. Uma boa medida de tamanho para software orientado a objetos é *número de casos de uso*.

Além de utilizar casos de uso para computar o tamanho do software, é útil classificá-los pelo tipo de funcionalidade oferecida, tal como, cadastro, consulta, emissão de relatório e negócio. Assim, é possível estabelecer médias de esforço, tempo e custo relativas a casos de uso de negócio, cadastro, consulta ou emissão de relatório.

É útil, ainda, computar o número de classes do domínio do problema (negócio) por caso de uso [4]. Essas classes têm pequena probabilidade de serem implementadas via reuso e, por esta razão, dão uma noção do esforço de desenvolvimento a ser realizado.

6. Conclusões

A definição de um processo para guiar o desenvolvimento é fundamental para se conseguir qualidade e produtividade. Neste trabalho, apresentamos um processo padrão para o desenvolvimento de software orientado a objetos, definido tendo por base o Processo Unificado. A escolha do RUP como ponto de partida deveu-se ao fato do mesmo ser direcionado à tecnologia de objetos, já estar sendo utilizado na prática e ter sido derivado a partir da experimentação, conforme indicado por seus autores [1]. Contudo, julgamos ser necessário adaptá-lo por duas razões principais: (i) considerar as características da organização, procurando adaptar o jargão e utilizar práticas já consagradas na mesma e (ii) considerar normas de qualidade, sobretudo as normas ISO 12207 e ISO 9000-3, uma vez que a qualidade de processo não é o foco direto do RUP.

O processo está em fase de implantação. Os primeiros projetos começam a fazer uso do mesmo e algumas dificuldades já podem ser sentidas, entre elas: (i) uma vez que, até então, a organização não realizava atividades de análise de riscos e gerência de configuração, há uma certa dificuldade por parte dos desenvolvedores na realização dessas atividades; (ii) as

estimativas eram realizadas através de pontos por função, ainda que com resultados pouco precisos, e, com a mudança de paradigma, os desenvolvedores não têm dados históricos, nem sentimento para realizá-las. Este problema está sendo contornado com o apoio de consultores na realização de estimativas e, com a coleta de dados de métricas, espera-se eliminar esta deficiência. À medida que o processo está sendo utilizado, novos modelos de documentos estão sendo propostos, visando padronizar a documentação gerada na organização e oferecer um guia para as equipes que estão iniciando na nova tecnologia. Esta foi a situação na elaboração de um padrão para descrição de casos de uso.

A médio prazo, com a utilização do processo padrão, espera-se que as estimativas tornem-se mais precisas, que os produtos gerados apresentem maior qualidade, e conseqüentemente menos tempo seja gasto com manutenção, e, finalmente, que a produtividade aumente em decorrência da introdução de uma abordagem sistemática de desenvolvimento de software. Dizemos a médio prazo, pois sabemos que, inicialmente, a tendência é a queda da produtividade, já que há novas práticas, não corriqueiras, foram introduzidas.

O processo está sendo colocado em prática e esta experimentação deve conduzir a ajustes. Assim, é necessário avaliar o processo e identificar pontos para melhoria. No que se refere à avaliação, nossa preocupação inicial está focada em dois atributos: desempenho e aderência [5]. Processos são projetados para produzir resultados. Assim, é necessário conhecer o desempenho do processo, isto é, os produtos apresentam a qualidade requerida e foram desenvolvidos dentro dos prazos e custos estimados? Além disso, a aderência ao processo definido deve ser avaliada, visando identificar práticas que não estão sendo seguidas e o impacto de sua não execução.

Referências

[1] I. Jacobson, G. Booch, J. Rumbaugh. *The Unified Software Development Process.* Addison-Wesley Object Technology Series, 1998.

[2] ISO 12207. Information technology – Software life cycle processes, 1995 (E).

[3] ISO 9000-3. Quality management and quality assurance standards, Part 3, 1991 (E).

[4] M. Lorenz, J. Kidd, *Object-Oriented Software Metrics*, Prentice Hall, 1994.

[5] W.A. Florac, A.D. Carleton. *Measuring the Software Process*, Addison-Wesley, The SEI Series in Software Engineering, 1999.