

# ***Myrup: uma Adaptação do RUP para Projetos de Pequeno e Médio Porte***

Jocelene de Oliveira Reis  
([jo.oliveira@terra.com.br](mailto:jo.oliveira@terra.com.br))

Arnaldo Dias Belchior  
([belchior@unifor.br](mailto:belchior@unifor.br))

Mestrado em Informática Aplicada – MIA  
Universidade de Fortaleza - UNIFOR  
Av Washington Soares, 1321  
60.811.341 – Fortaleza – CE – Brasil

## **Abstract**

This paper presents the Software Development Method MyRup. It is a customization of RUP (Rational Unified Process) for media and small projects, applying some CMMI-SW (Capability Maturity Model Integration for Software) and HCI (Human Computer Interface) concepts. This method purposes a sequence between activities in several RUP areas to be a guideline for development and management team. MyRup was successfully applied in some software projects. The learning lessons in these projects were used to adjust the proposed model.

**Key-words:** Development process, process quality, Method of development, RUP.

## **Resumo**

Este trabalho apresenta o método de desenvolvimento de software MyRup, que é uma adaptação do RUP (*Rational Unified Process*) voltado para projetos de pequeno e médio porte, com a utilização de conceitos do CMMI (*Capability Maturity Model Integration para Software*) e IHC (Interface Homem Computador). O método propõe uma seqüenciação entre as atividades a partir das diversas disciplinas do RUP, com a finalidade de guiar efetivamente os desenvolvedores desses projetos, sendo um guia simplificado e de fácil entendimento. O MyRup foi aplicado com êxito em alguns projetos de software, em uma pequena empresa de software. As lições aprendidas nesses projetos estão sendo utilizadas para refinarem o modelo proposto.

**Palavras-chave:** Processo de Desenvolvimento, qualidade do processo, método de desenvolvimento, RUP.

# 1 INTRODUÇÃO

Segundo estatísticas recentes, 74 % dos projetos de software não são bem sucedidos [12]. Assim sendo, a indústria de software, incluindo pequenas e médias empresas, buscam cada vez mais alternativas para mudar esta realidade.

Neste contexto, o RUP (*Rational Unified Process*) [2, 7] surgiu como uma alternativa, consolidando a utilização da UML (*Unified Modeling Language*) [1, 6], o conceito de arquitetura de software [10] e uma definição das atividades a serem executadas no desenvolvimento de um projeto de software.

O RUP [7, 11] é um processo com o objetivo de atender a uma grande classe de diversos projetos, podendo ser considerado como um *framework* genérico, necessitando de configurações para ser usado efetivamente. Assim, a sua implantação requer uma série de adaptações para a realidade particular da organização, dificultando a sua utilização em pequenas empresas, pois dificilmente estas têm um grupo de profissionais da área de qualidade aptos a fazerem tais adaptações.

Atuando neste problema, o MyRup propõe-se a ser um método baseado no RUP adicionando conceitos de IHC (Interface Homem Computador), adaptado a realidade de pequenos e médios projetos, oferecendo um guia simplificado, de fácil entendimento e passo a passo sobre as diversas atividades a serem executadas pela equipe de desenvolvimento durante o projeto.

O MyRup, ao invés de apresentar a seqüência das atividades por *disciplina* como o RUP faz, apresenta a seqüência destas atividades por fase, mostrando claramente o encadeamento destas em cada momento do projeto.

Este método utiliza um subconjunto das atividades propostas pelo RUP a fim de tornar o processo mais simples e acrescenta aspectos de usabilidade [9] através do uso do modelo do usuário [3]. O MyRup continua focado em casos de uso [14] e arquitetura [10], porém adiciona a preocupação com interface e de oferecer uma ligação clara entre as atividades das diversas disciplinas em todo o ciclo de desenvolvimento.

O artigo está organizado da seguinte forma: a seção 2 descreve em linha gerais o RUP; a seção 3 apresenta o MyRup com suas principais características e fases; a seção 4 mostra o estudo de caso onde o MyRup foi aplicado e apresenta os resultados preliminares obtidos; a seção 5 exibe as principais conclusões deste trabalho.

## 2 ENFOQUES SOBRE O RUP

O RUP (Figura 1) é um processo de desenvolvimento de software iterativo, centrado em arquitetura e dirigido por casos de uso. Oferece uma abordagem baseada em disciplinas (*workflows*) para atribuir tarefas e responsabilidades dentro de uma organização de desenvolvimento, tendo como principal meta garantir a produção de software de alta qualidade, atendendo às necessidades dos usuários dentro de um cronograma e orçamento previsíveis [11].

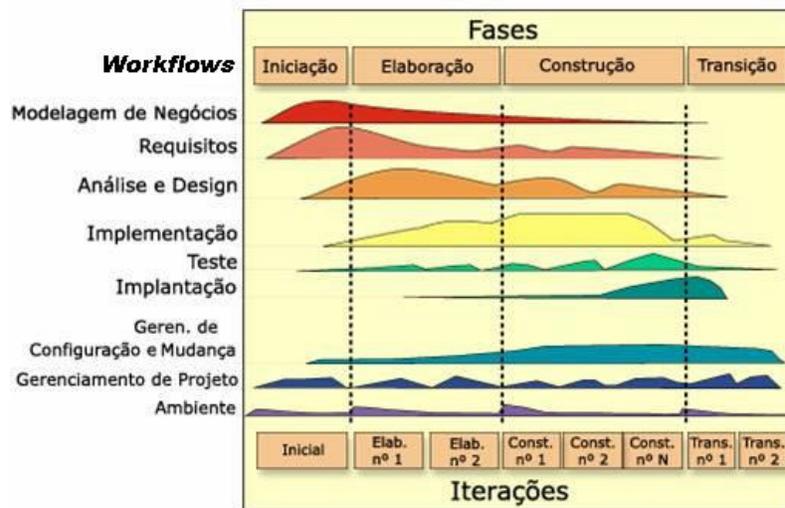


Figura 1: Fases e disciplinas do RUP

O RUP possui duas dimensões: horizontal e vertical. O eixo horizontal representa o aspecto dinâmico do processo, mostrando o tempo e os aspectos do ciclo de vida à medida que o projeto se desenvolve. O eixo vertical representa o aspecto estático do processo, mostrando as disciplinas que agrupam as atividades a serem desenvolvidas.

Uma disciplina envolve todas as atividades a serem realizadas para produzir um determinado conjunto de artefatos. O principal objetivo do agrupamento de atividades em disciplinas é ajudar a compreender o projeto a partir de uma perspectiva em cascata 'tradicional', como, por exemplo, ver todas as atividades a serem desenvolvidas de

requisitos. A separação dessas atividades em disciplinas distintas pode facilitar a compreensão, mas dificulta a programação, pois a seqüência das atividades do projeto como um todo não é clara.

O RUP divide o projeto de software em 4 fases distintas e com objetivos a serem alcançados: *concepção*, *elaboração*, *construção*, e *transição*.

A fase de concepção tem como objetivo fornecer uma visão inicial do projeto, definindo o escopo além de estimar preliminarmente os riscos e custos do projeto. A fase de elaboração tem por objetivo analisar o domínio do problema, de modo a propor uma arquitetura para os cenários da aplicação. O objetivo da fase de construção é esclarecer os requisitos restantes e concluir o desenvolvimento do sistema com base na arquitetura definida.

Finalmente, a fase de transição tem por objetivo assegurar que o software seja disponibilizado para seus usuários finais. A fase de transição inclui testar o produto e realizar pequenos ajustes com base no *feedback* do usuário. Ao final da fase de transição, os objetivos devem ter sido atendidos e o projeto deve estar em uma posição para fechamento.

A seguir, será apresentada a proposta do MyRup, que é um modelo de desenvolvimento de software para pequenos e médios projetos, baseado no RUP e em conceitos de IHC [9].

### 3 O MyRUP

O MyRup visa ser um guia simples e passo a passo para os integrantes da equipe de desenvolvimento de software, preservando a essência do RUP, porém trocando a forma de visualização das atividades (no RUP por disciplinas; no MyRup por fases), e acrescentando atividades com o enfoque de garantir a usabilidade do software. Assim, o método mostra uma forma de adaptação do RUP para pequenos projetos, já que o RUP é um framework genérico.

A característica essencial do MyRup e diferença em relação ao RUP é o agrupamento de atividades das diversas disciplinas por fase, mostrando uma visualização seqüencial dos grupos de atividades e tarefas a serem executadas em cada fase do processo (concepção, elaboração, construção e transição). Essa forma de visualização foi proposta para tirar a dificuldade existente no RUP de saber a ordem das atividades a serem executadas durante o projeto.

No RUP, a seqüência das atividades é mostrada por *disciplina* (exemplo: seqüência das atividades de requisitos do projeto e a seqüência de atividades de planejamento), não deixando clara a ligação das atividades das diferentes disciplinas, por exemplo, se a atividade de fazer o cronograma vem antes ou depois da atividade de especificação de requisitos.

Já a visualização do MyRup permite que o desenvolvedor possa conhecer a seqüência das atividades a serem executadas em cada fase do projeto, abstraindo-se da *disciplina* a que esta pertence. O desenvolvedor irá ver apenas a ordem que essas tarefas devem ser desempenhadas no decorrer do projeto (Ex: Especificar requisitos e depois fazer o cronograma).

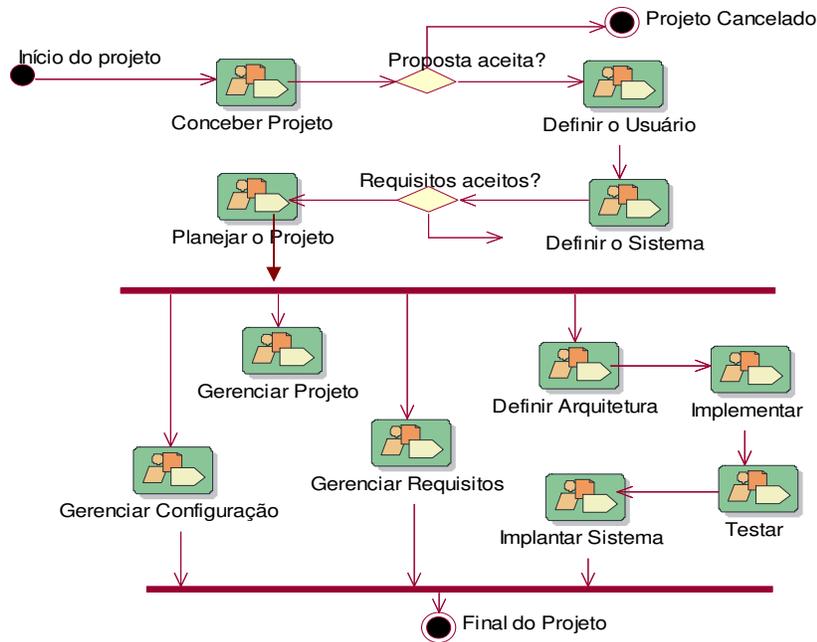
O MyRup retirou a disciplina de *modelagem de negócio* e várias atividades de outras disciplinas para tornar o processo mais leve para pequenos e médios projetos. Esse subconjunto visou ter o mínimo de atividades do RUP para um projeto garantir a qualidade de desenvolvimento. A outra diferença fundamental do método é o uso do modelo do usuário durante várias atividades ao longo do ciclo de vida, com o objetivo de adicionar aspectos de usabilidade na construção do software.

A Figura 2 apresenta o fluxo principal do MyRUP para o desenvolvimento de um projeto, com seus principais fluxos de trabalho, como se o ciclo de vida adotado para o projeto fosse em cascata. Cada fluxo de trabalho é detalhado em atividades envolvendo atores e gerando artefatos, além da exemplificação de artefatos.

Como indicado na Figura 2, o método procura guiar a equipe de desenvolvimento desde o início do projeto, com o fluxo *Conceber Projeto* (este contém as atividades de negociação com o cliente e de confecção de uma proposta), passando pela definição do sistema e planejamento macro.

Depois desta fase inicial, é proposta a definição e o refinamento da arquitetura a ser adotada (fluxo *Definir Arquitetura*), para só depois a implementação começar, seguida pelos testes e a implantação da versão final. Em conjunto com estes fluxos, ocorrem as atividades de gerenciamento do projeto com as atividades de planejamento e acompanhamento de cada iteração, além de controle dos riscos, gerência dos requisitos (com a finalidade de defini-los e controlar suas possíveis mudanças), e o gerenciamento da configuração dos artefatos dos projetos.

O MyRup indica que o projeto adote o ciclo de vida interativo, para minimização dos riscos e maiores chances de sucesso [13].



**Figura 2: Fluxo principal do MyRup**

Segue-se a apresentação de cada uma das fases do MyRup: *concepção, elaboração, construção, e transição.*

### 3.1. Concepção

A *Fase de Concepção* é a primeira fase do MyRup e tem como principal objetivo alcançar o entendimento entre todas as partes envolvidas (cliente, usuários, gerentes e equipe de desenvolvimento) sobre o produto a ser desenvolvido. Nessa fase, o escopo, os objetivos do sistema, o grau de usabilidade desejado e os principais marcos devem ser bem definidos e entendidos por todos. A fase de concepção é bastante crítica para todos os projetos, pois segundo Tom De Marco [5], 56% dos erros de um software podem ser rastreados na fase de requisitos.

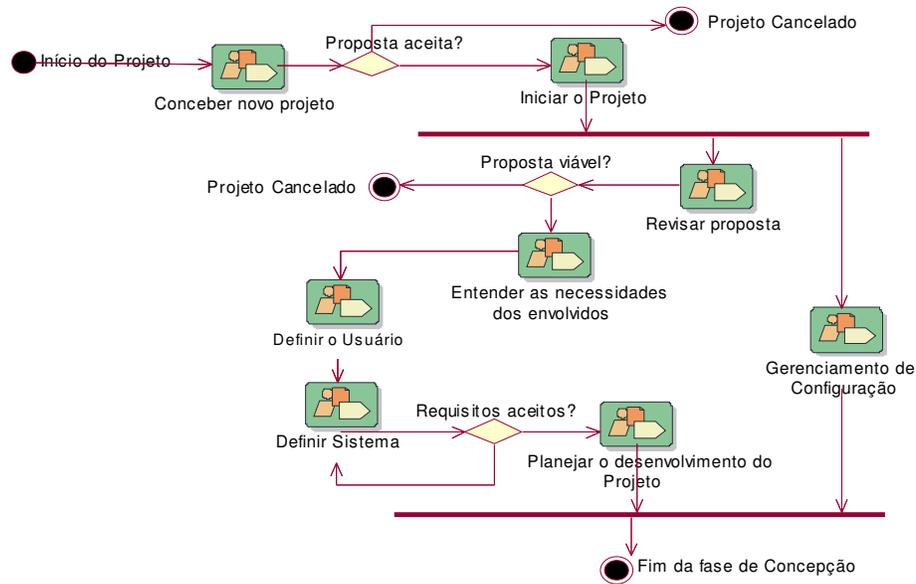
Nas atividades do fluxo de levantamento das necessidades dos usuários, foram incluídas atividades de modelagem do usuário, para que a equipe de desenvolvimento conheça o perfil destes e suas características, a fim de conseguir levantar os requisitos com maior facilidade além de influenciar no desenvolvimento do software.

A Figura 3 mostra os grupos de atividades (fluxos de trabalho) a serem executados dentro da fase de concepção, enfatizando a ordem em que estas ocorrerão. Com esta ordenação, o método tenta mostrar aos desenvolvedores tudo o que devem fazer na iteração da fase de Concepção.

Um exemplo de abstração das atividades das disciplinas é o fluxo *Planejar o desenvolvimento do Projeto* (pertencente à *disciplina* Gerenciar Projeto), ocorrendo depois do fluxo *Definir Sistema* (pertencente à *disciplina* Gerenciamento de requisitos). A equipe deve abstrair-se dos detalhes de organização do modelo, atendo-se apenas que precisa inicialmente definir o sistema, e depois planejar seu desenvolvimento.

No caso da fase de Concepção ter mais de uma iteração, o fluxo dessa fase repetir-se-á para cada uma delas. No contexto de pequenos e médios projetos, o MyRup recomenda que haja apenas uma iteração nessa fase.

Cada fluxo de trabalho apresentado na Figura 3 é descrito e diagramado, mostrando as atividades que os compõem, os artefatos gerados e os papéis desempenhados. Para exemplificar como o método é estruturado, será exibido a seguir um resumo do detalhamento de um fluxo e a descrição de uma de suas atividades, assim como uma parte de um dos artefatos a ser gerado.

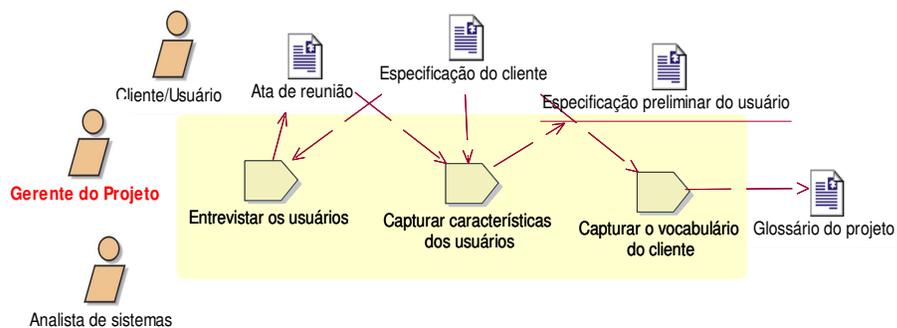


**Figura 3: Fase de Concepção do MyRup**

Na Figura 4, é apresentado um resumo do detalhamento do fluxo de trabalho *Entender as necessidades dos envolvidos* com suas atividades e o papel a ser desempenhado pelos integrantes da equipe de desenvolvimento. Esse fluxo apresenta as atividades de levantamento das informações com clientes ou usuários, para se obter a compreensão das reais necessidades do produto a ser desenvolvido pelo projeto (*Entrevistar os usuários*).

Inicialmente, pode-se imaginar essa atividade como um levantamento das expectativas de usuários e clientes, que poderão ser concretizadas em requisitos. A atividade *Capturar características dos usuários* visa levantar o perfil dos usuários assim como sua experiência no domínio do problema, para poder guiar desde o levantamento dos requisitos até a construção de uma interface com alto ou baixo grau de usabilidade.

Esse fluxo visa também *Capturar o vocabulário do cliente*, como um meio de uniformizar a linguagem que será utilizada pela equipe de desenvolvimento, pelos clientes e usuários.



**Figura 4: Fluxo *Entender as necessidades dos envolvidos***

A equipe do projeto, através de entrevistas com o cliente, ou em conversas técnicas internas, captura o vocabulário e elabora um glossário do projeto (artefato) com os termos e as definições necessárias. A definição do glossário é de suma importância para a consistência dos documentos ao longo do projeto (ex: especificação de requisitos, documento de arquitetura, etc.), a fácil inserção de novos membros à equipe, e o entendimento de documentos técnicos pelo cliente e de documentos de negócios pela equipe do projeto.

### 3.2. Elaboração

O MyRup propõe uma *Fase de Elaboração* mais simplificada que a do RUP, basicamente com atividades de *definir arquitetura* gerando o artefato Especificação de arquitetura e seu refinamento (fluxo “*refinar arquitetura*”) através dos diagramas de classe e de seqüência. A idéia dessa fase é diminuir os riscos da implementação do sistema através da definição de uma arquitetura sólida e já testada em protótipos evolutivos nas iterações. A Figura 5 mostra a visão geral da Fase de Elaboração do MyRup.

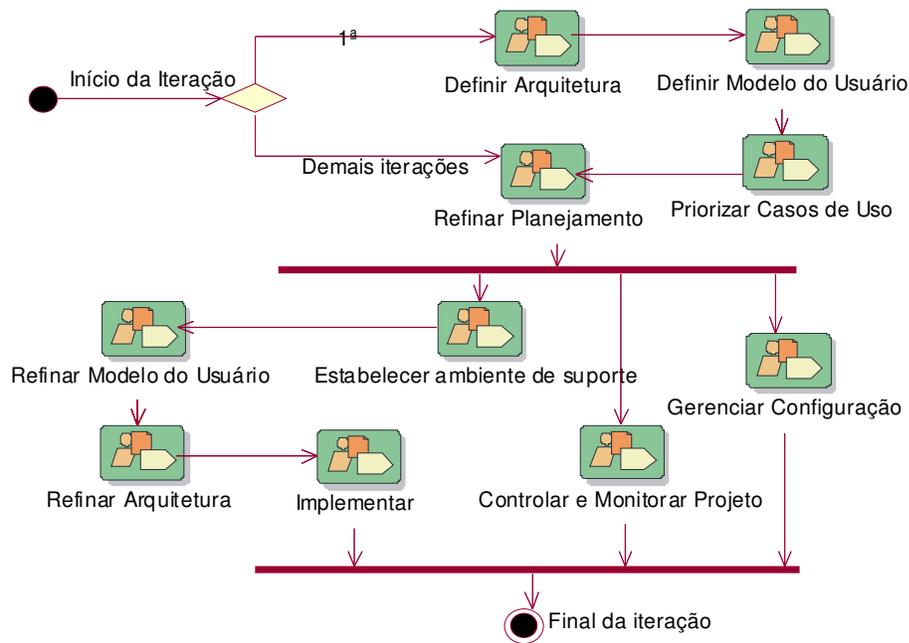


Figura 5: Fase de Elaboração do MyRup

Os principais fluxos de trabalho dessa fase são: *Definir arquitetura* e *Refinar a arquitetura* além de *Definir usuário*. O fluxo *Definir arquitetura* leva a equipe a estabelecer como o sistema irá funcionar em termos de componentes e tecnologias. A idéia nesse fluxo é de estabelecer uma arquitetura macro válida para todo o projeto visando identificar desde cedo as suas limitações e implicações (ex ao definir um banco de dados que o não suporta *store procedure*).

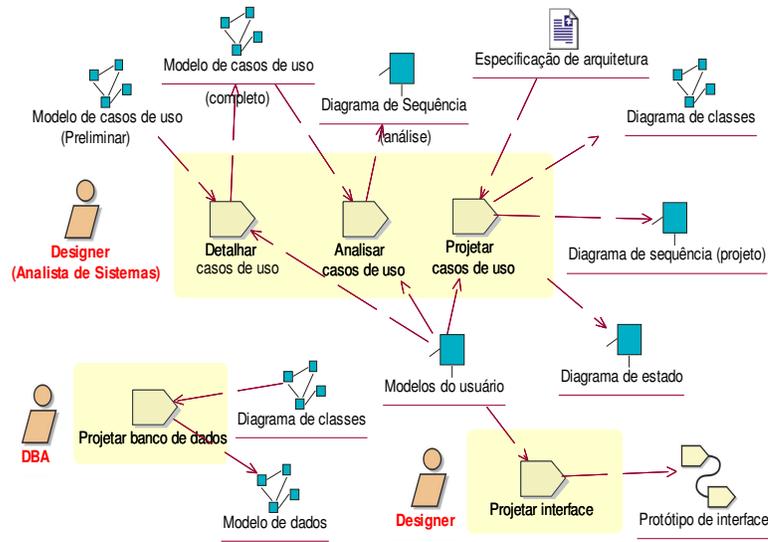
Após a definição da arquitetura, a equipe escolherá os casos de uso relevantes, no fluxo *Priorizar casos de uso*, para validar a arquitetura proposta. Esses casos de uso normalmente são considerados o núcleo do sistema, ou seja, o subconjunto capaz de validar se a arquitetura proposta irá funcionar.

O fluxo *Definir usuário* modela os perfis mais importantes do usuário, com suas diferentes expectativas do sistema em termos de funcionalidades e usabilidade. Essa definição acarretará num cuidado especial para funcionalidades ou interfaces que o usuário acha relevante, tanto por não conhecer ou por achar muito relevante.

O fluxo *Refinar arquitetura* (Figura 6) é um dos fluxos de maior valor percebido pelos implementadores, pois faz a interligação da documentação feita nos casos de uso na fase preliminar com o código a ser implementado. O MyRUP propõe apenas o uso dos diagramas de classe, de seqüência, e de estado (quando necessário para guiar o desenvolvimento).

Nestas atividades, o método mostra como chegar dos casos de uso ao código. Os desenvolvedores ou projetistas (*designers*) vão detalhar cada caso de uso (fazer a descrição destes e seus fluxos básico e alternativos), realizá-los através dos diagramas de classe e de seqüência e, quando a ferramenta o permitir, gerar o código a partir dos próprios diagramas desenvolvidos.

Por meio desta estrutura, objetiva-se mostrar aos desenvolvedores que a documentação é útil e está sendo feita como passo para a geração muitas vezes automática e com qualidade do código.



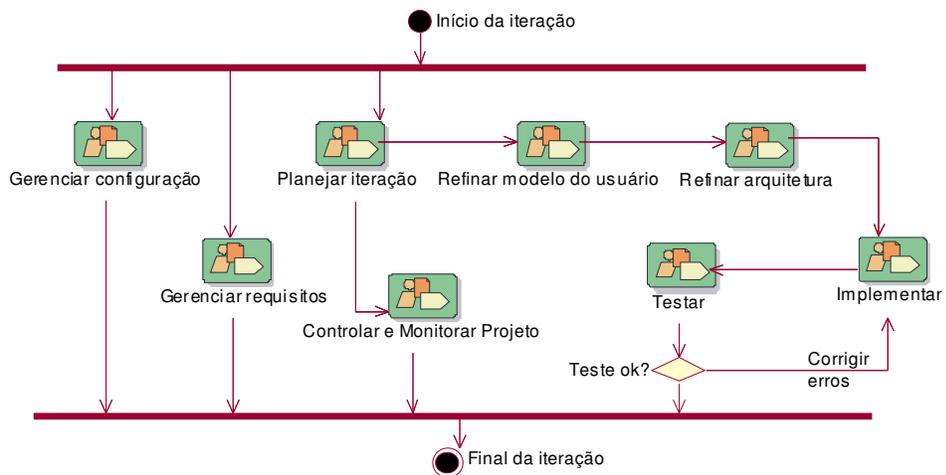
**Figura 6: Detalhamento do Fluxo Refinar Arquitetura**

O fluxo Implementar enfoca a geração do código dos casos de uso prioritizados a fim de validar a arquitetura proposta. Normalmente essa versão não é entregue ao cliente, porém gerando um protótipo evolutivo para os demais incrementos.

### 3.3. Construção

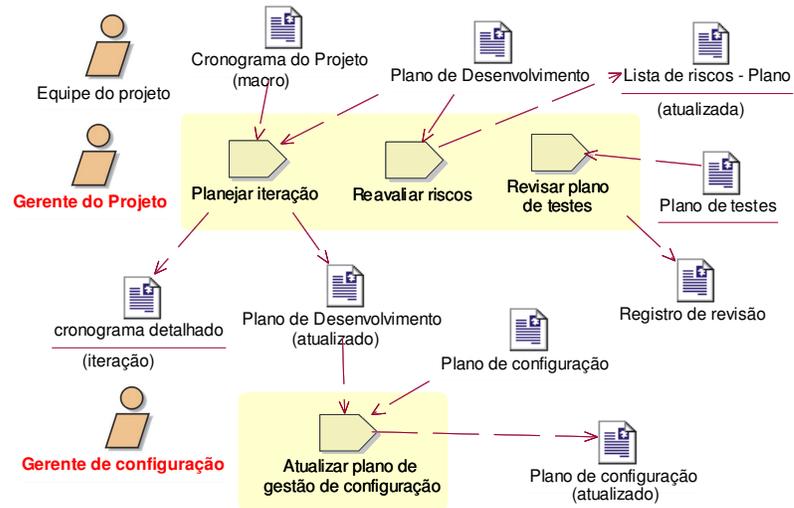
No MyRup, a *Fase de Construção* é voltada para um planejamento baseado em casos de uso e nas atividades voltadas para a implementação de uma versão passível de ser instalada no cliente. É interessante que o gerente do projeto planeje várias iterações para a fase de construção, e cada uma destas seguirá o fluxo apresentado na Figura 7.

Com a finalidade de preparar uma versão possível de ser instalada no cliente, aparece pela primeira vez, um fluxo de trabalho específico para testes a fim de garantir a qualidade de uma versão.



**Figura 7: Fase de Construção do MyRup**

A Figura 8 apresenta as atividades do fluxo *Planejar iteração* da fase de construção.



**Figura 8: Detalhamento do Fluxo *Planejar iteração***

No fluxo *Planejar iteração*, o Plano de desenvolvimento, que foi planejado a partir de conceitos do CMMI [4], contém os casos de uso de cada iteração, enquanto que o cronograma macro possui os marcos negociados com o cliente.

Com estas informações, o gerente do projeto organiza o cronograma detalhado da iteração, contendo todas as atividades a serem desenvolvidas. O cronograma proposto pelo MyRUP é guiado por casos de uso, conforme a Tabela 1 (parte do artefato gerado na seção 4).

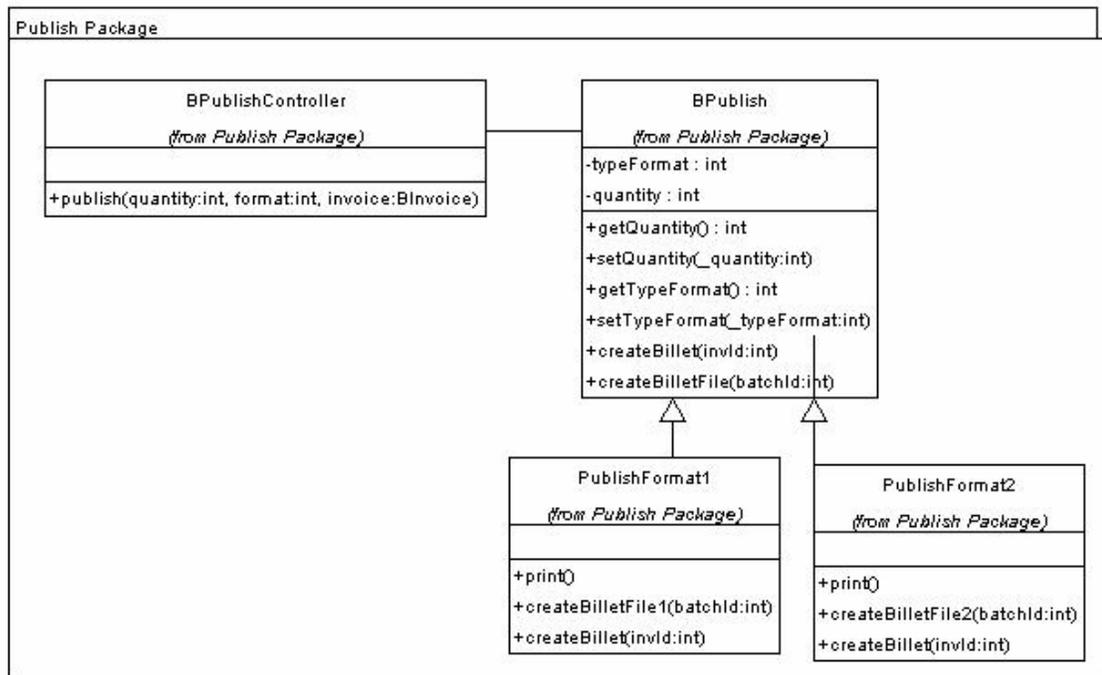
**Tabela 1: Exemplo de artefato - Cronograma detalhado do projeto (parcial)**

Tarefas	Nome da tarefa	Duração (dias)	Data Início	Responsável
	<b>PROJETO SISC</b>			
	<b>Fase de Construção – Iteração 1</b>			
1	Reunião de Planejamento	1	01/04/04	Todos
2	Atualizar Plano de Desenvolvimento	1	02/04/04	Sellaro
3	.... (outras atividades)	1	08/04/04	Ariluce e Fábio
6	<b>Detalhar Casos de Uso</b>			
7	Criar Fatura	1	10/04/04	Fábio
8	Inserir transações	0,5	11/04/04	Fábio
	... (demais casos de uso da iteração)			
16	<b>Projetar Casos de Uso</b>			
17	Criar Fatura	1,5	14/04/04	Ariluce
18	Inserir transações	2	16/04/04	Ariluce
	... (demais casos de uso da iteração)	2	16/04/04	Fábio
	<b>Implementar Caso de Uso</b>			
17	Criar Fatura	1,5	14/04/04	Ariluce
	... (demais casos de uso da iteração)			
21	<b>Testar Caso de Uso</b>	4	10/04/04	Sellaro
23	Criar Fatura	2	14/04/04	Fábio
	... (demais casos de uso da iteração)			

O fluxo *Refinar arquitetura* da fase de construção é semelhante ao da fase de elaboração com a diferença de que os diagramas de análise tornam-se opcionais, pois normalmente não são desenvolvidos devido à experiência que a equipe já adquiriu em confeccionar diagramas, e pelo conhecimento adquirido no domínio do problema.

O refinamento é feito apenas para o grupo de casos de uso planejados para a iteração corrente. A vantagem da implementação e modelagem em incrementos é o amadurecimento da equipe na técnica de confeccionar esses diagramas e no domínio do projeto, minimizando a quantidade de erros no projeto do software.

A Figura 9 apresenta um diagrama de classe construído em um dos projetos do estudo de caso (seção 4), que ilustra as classes utilizadas na realização de um dos casos de uso, e que já estão prontas para a geração de código.



**Figura 9: Exemplo de artefato - Diagrama de Classes do projeto (parcial)**

As atividades do fluxo *Gerenciar requisitos* nesta fase assumem uma importância bastante relevante, pois uma mudança em algum requisito, pode impactar desde uma simples mudança em um método de uma classe até uma alteração na arquitetura do sistema ou escopo do projeto.

Já o resultado da modelagem do usuário influencia fortemente na implementação das interfaces, pois estas devem ser projetadas de acordo com os perfis e conhecimentos dos usuários. A experiência do usuário e o tipo de componentes gráficos preferidos guiam o desenvolvimento para a construção do software que o usuário goste de usar e atenda suas expectativas de usabilidade.

### 3.4. Transição

A *Fase de Transição* do MyRup (Figura 10) é voltada para a implantação da versão final no cliente. Portanto, o gerente do projeto só deve dar por concluídas todas as iterações da Fase de Construção e passar para a primeira iteração da Fase de Transição, quando a implementação já tiver sido concluída.

O enfoque da Fase de Transição é realizar todas as atividades necessárias para a implantação e finalização do projeto, desde a documentação final do sistema ao teste de integração e treinamentos com os usuários. Depois da última iteração dessa fase, tem-se a conclusão do projeto.

A partir deste momento, as futuras manutenções devem ser tratadas como mini-projetos, com a elaboração de seu planejamento, e passando por todas as fases do MyRup.

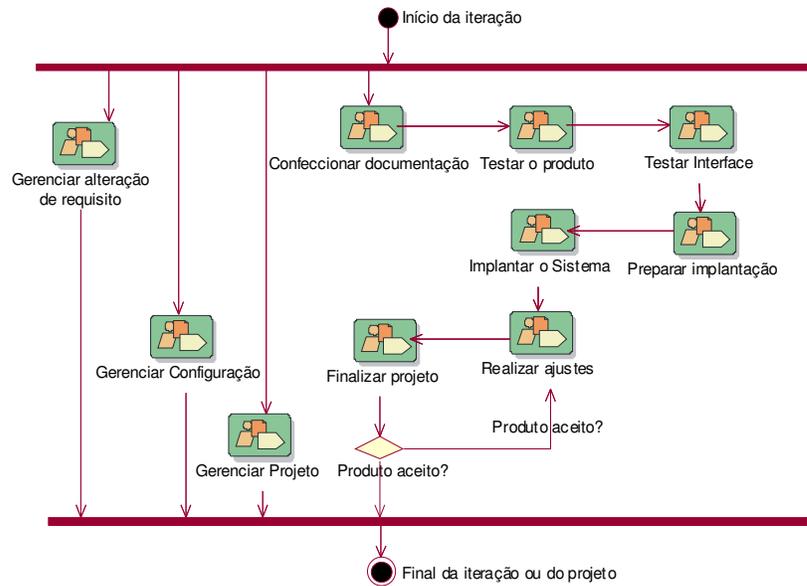


Figura 10: Fase de Transição do MyRup

O gerente do projeto deve ter um cuidado especial com as atividades do fluxo *Preparar implantação*, (Figura 11), contendo as atividades *Implementar correções*, *Criar arquivos de configuração* e *Fazer empacotamento do produto*, pois freqüentemente essas atividades não são tratadas devidamente pela equipe de desenvolvimento. Porém, uma falha nessas atividades pode gerar transtorno na implantação e, como consequência, deixar uma impressão ruim nos clientes e usuários do novo produto de software.

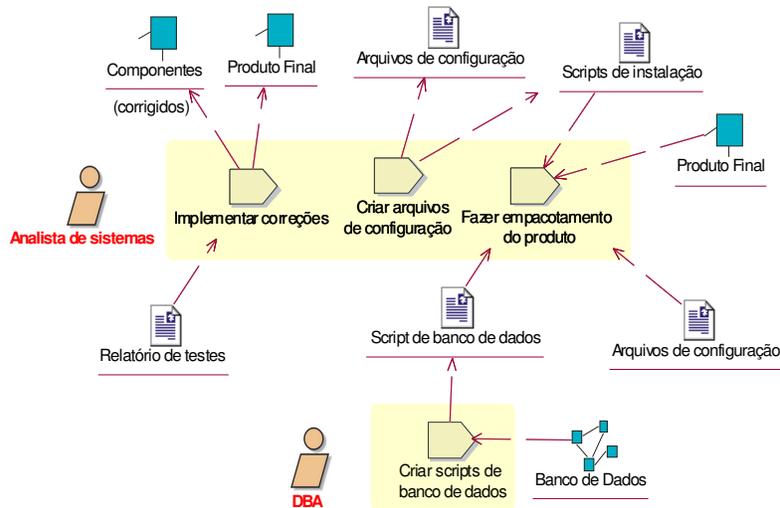


Figura 11: Detalhamento do Fluxo Preparar implantação

O fluxo *Gerenciar configuração* também é extremamente importante na fase de transição, visto que os erros mais freqüentes na implantação estão as versões erradas disponibilizadas aos clientes e usuários. Neste sentido, as *baselines* de patch devem ser rigorosamente controladas, para que o cliente e a equipe de implantação possam saber quais os erros encontrados naquela versão, e que correções já foram efetuadas.

O fluxo *Confeccionar Documentação* tem uma grande influência da modelagem dos usuários para gerar uma documentação fácil e de acordo com as características de cada perfil identificado. Como exemplo tem-se o Manual do Usuário desenvolvido de acordo com as premissas de cada perfil do usuário do sistema.

A conclusão do projeto é marcada pelo conjunto de atividades do fluxo *Finalizar projeto*, que, além de tratar da finalização interna do projeto, procura também a obtenção de um aceite formal de término do projeto emitido pelo cliente.

## 4 ESTUDO DE CASO

O MyRup está sendo aplicado na Fujitec, empresa de TI voltada para desenvolvimento de sistemas embutidos, soluções com *smartcards* e transações eletrônicas, com sede em Fortaleza (Brasil).

A Fujitec é certificada ISO 9001 (BVQI) e, atualmente, está implantando o MyRup, visando obter a qualidade necessária no processo de desenvolvimento de software, para continuar atuando competitivamente no mercado internacional. Atualmente, o MyRup está sendo utilizado em 3 projetos pilotos, visando o refinamento do mesmo e, posteriormente, ser estendido para toda a empresa.

Durante a implantação do MyRup nos projetos pilotos, foram colhidas algumas informações sobre o resultado da implantação do método, apesar de nenhum deles ainda ter sido finalizado para a coleta e a análise dos dados consolidados. A seguir, são apresentados alguns resultados mais relevantes obtidos com o uso do MyRup nesses projetos.

- **Interfaces adaptadas a diversos usuários**
  - A equipe de desenvolvimento conseguiu gerar interfaces adaptadas às necessidades de mais de um grupo a partir da modelagem do usuário. Nessa modelagem os desenvolvedores mapearam as principais exigências e preferências dos usuários como por exemplo a utilização de interfaces extremamente simples e diretas no caso das movimentações nos terminais de venda (POS).
  - Uma segunda consequência dessa modelagem e não prevista inicialmente como ganho pelo método (relato das equipes), foi a análise do impacto das decisões de interface no projeto e na construção do software.
- **Facilidade em seguir o processo**
  - Os desenvolvedores conseguiram seguir o MyRup sem uma equipe de qualidade na empresa e nem uma consultoria periódica. Os desenvolvedores depois de participaram de workshops (1 de apresentação e outro prático de estudo de caso) e terem acesso ao processo acharam-se aptos a seguirem as atividades propostas sem um suporte especializado.
- **Conhecimento das tarefas do projeto**
  - A equipe conseguiu visualizar nos fluxos de trabalho propostos e no seu detalhamento a seqüência das tarefas a serem realizadas assim como sua ordem.
  - A experiência com a implantação do MyRup indica que os desenvolvedores não sentiram necessidade de saber a qual o *workflow* pertencia cada atividade (característica do RUP perdida com a adaptação proposta pelo método) e sim sua ordem e o entendimento desta.
- **Melhoria da documentação e a utilização da mesma durante todo o projeto**
  - A Tabela 2 mostra a documentação dos projetos antes e depois do MyRup. A equipe de desenvolvimento atribuiu a efetiva utilização da documentação pelo fato de um documento estar relacionado a outro, pois, para confeccionar um artefato são utilizados como insumos artefatos produzidos anteriormente.
  - Clientes destes projetos reconheceram explicitamente a organização da empresa, e elogiaram, por exemplo, a qualidade da documentação da especificação de requisitos de seu projeto.

**Tabela 2: Comparação da documentação antes e depois da implantação do MyRup**

Documentação dos projetos antes do MyRup	Documentação atual dos projetos (Principais documentos)	
<ul style="list-style-type: none"> <li>▪ Documento de arquitetura</li> <li>▪ Modelo de Entidade e Relacionamento</li> <li>▪ Cronograma macro</li> <li>▪ Manual do usuário</li> </ul>	<ul style="list-style-type: none"> <li>▪ Especificação de Requisitos</li> <li>▪ Diagramas de classe</li> <li>▪ Modelo de Entidade e Relacionamento</li> <li>▪ Documento de arquitetura</li> <li>▪ Manual do usuário</li> </ul>	<ul style="list-style-type: none"> <li>▪ Modelo de casos de uso</li> <li>▪ Diagramas de seqüência</li> <li>▪ Plano de Desenvolvimento</li> <li>▪ Cronograma detalhado para cada iteração</li> <li>▪ ... outros artefatos do método</li> </ul>

- **Orientação a Objetos**
  - Com a utilização do MyRup, os projetos passam a trabalhar no paradigma orientado a objeto, usando a UML [8].

- **Facilidade na inserção de novos membros nas equipes de projeto**
  - Os gerentes de projeto constataram que a inserção de novos integrantes nas equipes dos dois projetos pilotos foi bem mais fácil em relação aos demais projetos da empresa. Isto porque logo na primeira semana, o novo integrante da equipe estudava todos os artefatos a partir da documentação do projeto gerados pelo MyRup, e tirava suas dúvidas com os demais membros da equipe. Em pouco tempo, o recém membro já se encontrava apto para executar suas tarefas no projeto.
  - O indicador de tempo para que um novo membro começasse a ser produtivo (desenvolver atividades sozinho) diminuiu de três para apenas uma semana.
- **Maior controle no gerenciamento do projeto e estabelecimento de estimativas mais realistas**
  - A utilização do cronograma detalhado por caso de uso fez com que os gerentes acompanhassem melhor as atividades de cada um dos envolvidos no projeto e conseguissem estabelecer estimativas mais próximas da realidade.
- **Controle das versões liberadas para os clientes**
  - Os dois projetos começaram a trabalhar com gestão de configuração e ter um controle mais efetivo nas versões internas e liberadas para o cliente. Para a gestão de configuração, a empresa adotou a ferramenta CVS (*Concurrent Versions System*), que uma ferramenta de software livre, largamente utilizada.

## 5 CONCLUSÕES

A principal motivação para o trabalho foi propor uma adaptação do RUP voltada para a realidade de pequenos e médios projetos, acrescentando aspectos de usabilidade e uma visualização mais clara da seqüência das atividades. O MyRUP selecionou um subconjunto das atividades do RUP com a finalidade de tornar o processo mais leve e possível de ser seguido com poucas pessoas.

A visualização por fase das diversas atividades dos *workflows* surgiu pela dificuldade em se saber a seqüência das atividades no RUP durante o projeto. Já as atividades de usabilidade foram acrescentadas ao método pelo fato de o RUP não possuir essa característica e ser um fator importante para o desenvolvimento de um produto com qualidade, principalmente na ótica do cliente.

Durante a implantação do método, os membros da equipe fornecem feedback sobre a experiência de utilizar o MyRup, sugestões de melhorias e dificuldades encontradas. Essas informações são utilizadas para a geração de novas versões e workshops para resolver dificuldades e problemas de interpretação.

Vale ressaltar que o método ainda está sendo melhorado e para as próximas versões já estão previstas maiores detalhamentos das atividades de configuração e continuidade com as atividades relacionadas à usabilidade.

Como trabalho futuro, vislumbra-se o desenvolvimento de uma ferramenta para apoiar o gerenciamento do projeto que utiliza o MyRUP.

## Referências

- [1] Booch, G. *The Unified Modeling Language User Guide*. Addison-Wesley, 1999.
- [2] Booch, G., Jacobson, I., Rumbaugh, J. *The Unified Software Development Process*. Addison-Wesley, 1999.
- [3] Brusilovsky, P. *User Modeling and User Adapted Interaction*. 1996.
- [4] CMMI Product Team. *CMMI for Systems Engineering/Software Engineering*, Version 1.1 Staged Representation (CMU/SEI-2002-TR-029, ESC-TR-2002-029). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, August, 2002.
- [5] Demarco, T. *Princípios e Conceitos de Engenharia de Software*. Compucenter, 1996.
- [6] Jacobson, I. *Object-Oriented Software Engineering: A Use Case Driven Approach*. Addison-Wesley, 1994.
- [7] Kruchten, P., Kroll, P. *The Rational Unified Process Made Easy*. Addison-Wesley. 2003.
- [8] Lairman, G. *Utilizando UML e padrões*. Bookman, 2001.
- [9] Mayhew, J. Deohah. *The Usability Engineering Lifecycle*. Morgan Kaufmann. 2001.
- [10] Mendes, A. *Arquitetura de Software - Desenvolvimento Baseado na Arquitetura*. Editora Campus. 2002.
- [11] Rational Software Corporation. *Rational Unified Process*. Version 2001.03.00. CD-ROM, *Rational Software*, Cupertino, California, 2001.
- [12] Standish Group. *CHAOS Report*. 2000.
- [13] Sommerville, I. *Software Engineering*. 1, Addison-Wesley, 1996.
- [14] Winters, J., Jacobson I. *Applying Use Cases*. Addison-Wesley. 1998.