

O Rational Unified Process – Um Ativador para a Maturidade de Processo Avançado

**Annie Kuntzmann–Combelles, Q-Labs
France**

**Philippe Kruchten, Rational Software
Canada**

A Rational Software and Q-Labs White Paper

TP 178, 2/01

Rational[®]
the software development company



Índice Analítico

Resumo	1
Você é Eficiente o Suficiente?	1
Esquemas para Aprimoramento	2
O que as Organizações Precisam para Iniciar um Programa SPI?	2
O Modelo CMM [6]	2
O Método IDEAL [4]	2
O Processo	2
O Rational Unified Process	3
Documentos ISO	3
Utilizando o RUP para SPI	3
O RUP Corresponde aos Requisitos do CMM no Nível do Projeto	4
Questões de Ciclo de Vida	4
Obter a Visão do Projeto e Desenvolver um Modelo de Negócios	6
Organizar e Planejar o Projeto	6
Implantar a Política Unified Change Management	7
Seleção de Processo e Adaptação	8
Gerenciamento de Riscos	9
Medida	9
O RUP Corresponde aos Requisitos do CMM no Nível da Organização	10
Aprimoramento do Processo	11
As Etapas para Implementar o Processo e Ferramentas em uma Organização	11
Um projeto de implementação de processo pode ser dividido em fases	12
Recursos e Habilidades	12
Pragmática: Como Iniciar o RUP	14
Conclusão	17
Informações de Contato dos Autores	18
Referências	19

Field Code Changed

Deleted: 5

Deleted: 8

Deleted: 10

Deleted: 18

Resumo

Essa nota destaca os conceitos principais que uma organização desenvolvida – uma unidade de desenvolvimento Nível 3 – precisa demonstrar e como os componentes do RUP (Rational Unified Process) correspondem a esses requisitos.

O vencimento principal do projeto e da organização são tratados. Além disso, a seção 4 fornece algumas idéias interessantes para iniciar o RUP e são relatados alguns dos benefícios principais observados pelos adotantes anteriores em vários contextos.

Você é eficiente o suficiente?

Hoje em dia, as empresas devem procurar continuamente formas de alcançar um grau mais alto de desempenho. A contestação do futuro é concluir em um nível global. Esses sintomas comuns tornam-se mais óbvios quando lançados no mercado mundial:

- **Necessário para desempenho:** muitos erros são observados na fase de validação e uma versão instável do produto é colocada no mercado, para atender o destino da entrega
- **Necessário para eficiência:** os projetos superam o orçamento e o planejamento
- **Perda de participação no mercado:** os concorrentes desempenham melhor e entregam melhor qualidade
- **Falta de recursos competentes:** o movimento é grande e/ou contratar novos engenheiros é muito difícil
- **Necessário para integrar as tecnologias em desenvolvimento com riscos mínimos:** as tecnologias que suportam os produtos precisam seguir as tendências; as equipes nem sempre têm conhecimento o suficiente para atender a contestação

Além do mais, geralmente, é possível dizer que há uma forte necessidade de otimizar todos os aspectos do produto desenvolvido.

Portanto, tornando a hipótese amplamente aceita de que o processo de desenvolvimento do produto impacta expressivamente nos resultados, um dos cenários mais comuns seria tratar do processo de software para aprimorar os produtos.

As organizações que enfrentam esse tipo de situação, provavelmente, passam pela solução a seguir para obter êxito.

A primeira parte da solução é entender o valor do software para o mercado. Se o software continuar a ser uma vantagem competitiva importante para os produtos e projetos, então, os processos para especificação e implementação dos mesmos precisarão ser definidos, documentados e otimizados com o passar do tempo com cuidado. Observa-se que o gerenciamento sênior ainda é, muitas vezes, convencido de que o software cria problemas e chega atrasado; a análise incluída no valor para o software não foi desenvolvida.

A segunda parte da solução é dedicar o investimento adequado para corrigir os problemas principais de seus processos:

- Avaliar o processo de identificação de oportunidades de aprimoramento
- Não reinventar o ciclo: aprenda a partir de experiências com terceiros e acelere o andamento
- Considerar o gerenciamento e a perspectiva técnica em qualquer projeto
- Gerenciar as habilidades necessárias no futuro: aprender, compartilhar e crescer

Existem vários esquemas para transformar uma organização em uma equipe altamente produtiva; uma das melhores decisões a ser tomada é mudar o que for possível, aceitar o que não pode ser mudado e emprestar tudo o que puder, que foi validado por uma comunidade de software bem-sucedida. O CMM (Capability Maturity Model) [6] e o RUP (Rational Unified Process) [11] são dois exemplos de ferramentas amplamente aceitas e fortes que podem acelerar o progresso sobre processos adequados para desenvolvimento de produtos. O termo ferramenta é utilizado porque, enquanto esses dois componentes são suportes efetivos para alteração e aprimoramento de práticas de software, eles permitem a definição das boas práticas para seu contexto organizacional.

A próxima seção descreve esses dois componentes—CMM e RUP—bem como, alguns outros componentes principais a serem considerados para aprimorar o processo e os produtos entregues.

Esquemas para Aprimoramento

Na última década, a comunidade de software tornou-se o processo consciente e foi constatado que os produtos de software não podem ser avaliados independentemente do processo que os produzem. Por isso, várias organizações criaram programas para aprimorar os processos de desenvolvimento de software conhecidos como iniciativas de SPI (Software Process Improvement). Na maioria dos casos, o desenvolvimento de software é concentrado em uma ou mais unidades ou departamentos de uma organização e é considerado uma disciplina independente. Essa situação é alterada quando o programa SPI é adequado e as equipes de software são melhores integradas a outras disciplinas da organização.

O que as Organizações Precisam para Iniciar um Programa SPI?

- Um estímulo para iniciar um programa e uma pessoa chave para coordenar as atividades de aprimoramento; em outras palavras, definir um projeto de aprimoramento com uma instrução de visão e objetivos.
- Um modelo de referência em oposição ao qual avaliar as práticas do projeto e a organização e identificar quaisquer partes que estão faltando que podem impedir o sucesso. O modelo CMM desenvolvido no SEI (Software Engineering Institute) é, de fato, uma referência utilizada por milhares de organizações e a estrutura SPICE (ISO 15504) [9] é uma alternativa que em breve pode ser um padrão oficial com o qual o CMM estará em conformidade.
- Um método para gerenciar o programa de aprimoramento para ser encerrado com resultados bem-sucedidos: a estrutura IDEAL [4] definida no SEI comprovou sua intensidade.
- Um processo de desenvolvimento que pode ser adaptado para o sucesso de cada projeto: o RUP é uma solução em circulação aceitável.

O Modelo CMM [6]

O CMM (Capability Maturity Model) do SEI (Software Engineering Institute) é uma estrutura que descreve os elementos de um processo de software efetivo [6]. O CMM descreve um caminho de aprimoramento evolutivo de um processo não desenvolvido específico para um processo desenvolvido e disciplinado. Apresenta os conjuntos de práticas recomendadas em várias áreas do processo chave, que foram mostradas para melhorar o desenvolvimento de software e a capacidade de manutenção. O CMM conduz os desenvolvedores à obtenção de controle de seus processos de desenvolvimento e de manutenção, e ao desenvolvimento por meio de uma cultura de destaque de engenharia e de gerenciamento de software.

O Método IDEAL [4]

O Software Process Improvement é um método sistemático, colaborativo e de longo alcance, para desenvolver a forma com que o trabalho de software é organizado e desempenhado. Os métodos de aprimoramento incluem o método IDEAL, o qual é uma abordagem integrada do SPI definida pelo SEI. O IDEAL identifica cinco fases: Iniciar, Diagnosticar, Estabelecer, Agir e Atuar sobre. Cada uma dessas fases é centralizada em uma atividade específica:

- Especifique as metas de negócios que serão realizadas ou suportadas (**Iniciar**).
- Identifique o estado atual da organização em relação a um modelo padrão ou de referência relacionado (**Diagnosticar**).
- Desenvolva planos para implementar a abordagem escolhida (**Estabelecer**).
- Reúna tudo o que é válido para criar uma solução com “melhor intenção” específica para necessidades organizacionais—por exemplo, ferramentas, processos, conhecimento e habilidades existentes—e determinar a solução adequada (**Agir**).
- Resuma as lições aprendidas sobre os processos utilizados para implementar o IDEAL (**Atuar sobre**).

O Processo

Algumas organizações não têm processo; algumas têm processos baseados em sua experiência. Há alguns processos off-the-shelf disponíveis, como por exemplo o OPEN e o RUP (Rational Unified Process). O OMG Group pode propor um modelo de processo genérico em breve. O termo “processo” é amplamente utilizado no Software Engineering Community, mas as premissas sobre os componentes do processo de software são muito heterogêneas. Vamos analisar a definição determinada pelo SEI: um conjunto de atividades, métodos, práticas e transformações que as pessoas utilizam para desenvolver e manter o

software e os produtos associados. O processo é um elemento chave que garante a correspondência de um projeto a seus objetivos definidos.

É importante observar que essa definição inclui atividades, tecnologias (isto é, métodos e ferramentas) e pessoas. Nenhum desses três componentes é mais crucial do que os outros.

O Rational Unified Process

O RUP é um exemplo de uma estrutura de processo preexistente que se beneficia da experiência de projeto longo [10, 11].

O RUP enfatiza as áreas de alto risco de endereçamento antecipadamente, pelo desenvolvimento rápido de uma versão inicial do sistema, que define sua arquitetura. Ele não aceita um conjunto fixo de requisitos estáveis na iniciação do projeto, mas permite o aperfeiçoamento dos requisitos conforme o desenvolvimento do projeto. Espera e acomoda mudanças. O processo não determina um foco intenso sobre documentos ou cerimônias, e serve para automação de muitas das tarefas cansativas associadas ao desenvolvimento do software. O foco principal fica com o próprio produto de software e sua qualidade, conforme medido até o ponto em que ele satisfaz seus usuários finais e atende completamente ao objetivo de investimento de rendimento.

É genérico o suficiente para ser direcionado a uma variedade ampla de produtos e projetos de software, no domínio de tamanho e de aplicativo e é centralizado ao redor de três áreas: pessoas, processo e ferramentas ou métodos [10,11]

Documentos ISO

Finalmente, para concluir esse grande painel, é necessário mencionar os documentos ISO, como por exemplo ISO 9001, ISO 12207, ISO 15504 (SPICE) [9], os quais são referências disponíveis, desta forma, as organizações podem comparar sua experiência em desenvolvimento de software com outras organizações. Eles formam a estrutura genérica com a qual os outros componentes anteriormente mencionados estão em conformidade.

ISO 15504 (também conhecido como SPICE) é outro modelo de referência para análise de processos de software. Ele considera que muitos modelos de avaliação completamente realizados (CMM é um deles) e muitos métodos de avaliação (o método definido pelo SEI é um deles) conseguirão ser mapeados para as partes normativas do ISO 15504.

No restante desse documento, é focalizado o CMM, o qual é, de fato, o padrão da comunidade de software. O CMM é um modelo completamente realizado que suporta milhares de iniciativas de aprimoramento do processo de software. Tudo o que é discutido está em conformidade com o SPICE.

Utilizando o RUP para SPI

Considerando os elementos chave listados na seção anterior—o modelo de referência do CMM, o método para gerenciar o programa de aprimoramento para encerramento (IDEAL) e o processo de desenvolvimento adaptável para aumentar a competição e atender a contestação global do software—são mapeados agora os conceitos do RUP para requisitos do CMM e refletidos no potencial do RUP, no alcance da capacidade do software específica.

A capacidade do software realmente significa qualquer um dos itens a seguir para uma organização:

- A decisão adequada para o sucesso do projeto na hora correta
- Obter sucesso e sobreviver nos negócios
- Assumir riscos e controlar a saída do projeto
- Obter a qualidade correta do produto
- Integrar as tecnologias de software mais avançadas nos produtos

A capacidade do software é totalmente uma mistura sofisticada dessas instruções anteriores, mas há alguns conceitos que devem ser considerados como parte da capacidade do software:

- No nível do projeto:
Questões de Ciclo de Vida: O ciclo de vida do projeto não é limitado à implementação e à manutenção dos produtos. Há uma perspectiva de gerenciamento que trata dos negócios, finanças e estratégias de um projeto.

Seleção e Adaptação do Processo: O processo padrão em circulação e/ou reflexo da experiência anterior não podem corresponder os objetivos para uma variedade de projetos. A eficiência resulta na definição de um processo de desenvolvimento adequado para um projeto específico.

Gerenciamento de Riscos: Qualquer projeto enfrenta riscos. Uma organização capaz antecipa os riscos e toma decisões para reconfigurar o projeto, a fim de minimizar o impacto de ocorrência de riscos.

Medidas: Um item importante para progresso a longo prazo em uma organização de software é coletar dados históricos, para analisar a qualidade e a produtividade do software. O processo de coleta de alguns dados históricos para cada projeto é demorado. Dados sobre esforço, planejamento, tamanho do programa e funções implementadas, e contagem de defeitos formam uma base sólida para planejamento de projetos futuros e aprimoramento de possibilidade de previsões. A previsão do desempenho é um sinal de maturidade.

- No nível da organização – visto como uma coleta de projetos:
Aprimoramento do Processo: A capacidade implica que uma organização consegue aprender com os erros do passado, especialmente de terceiros, e podem converter essas lições aprendidas em evoluções do processo. Os ciclos de iteração de aprimoramento apenas serão concluídos, se as medidas forem definidas para quantificar o aprimoramento.

Recursos e Habilidades: A capacidade de uma organização é rigidamente acoplada à capacidade das pessoas aplicarem os processos e desenvolvimento dos produtos.

O RUP Corresponde aos Requisitos do CMM no Nível do Projeto

O CMM descreve o status dos projetos em organizações que atingem um nível de maturidade específico.

Os projetos em organizações do Nível 2 têm controles de gerenciamento de software básicos instalados. As confirmações de projetos realistas são baseadas nos resultados observados em projetos anteriores e nos requisitos do projeto atual. Os coordenadores de projetos de software acompanham os custos, os planejamentos e a funcionalidade do software; os problemas no atendimento das confirmações são identificados conforme aparecem. Os requisitos de software e os produtos de trabalho desenvolvidos para satisfazê-los servem como base e sua integridade é controlada. Os padrões de projeto de software são definidos e a organização garante que eles são lealmente seguidos. O projeto de software trabalha com seus subcontratantes, se houver, para estabelecer um forte relacionamento entre o cliente e o fornecedor.

No Nível Definido (Nível 3), o processo padrão para desenvolvimento e manutenção de software é documentado através da organização. Os projetos direcionam o processo de software padrão da organização, para desenvolverem seus próprios processos de software definidos, os quais se responsabilizam pelas características exclusivas do projeto. Esse processo adaptado é referido no CMM como o processo de software definido do projeto. Um processo de software definido contém o conjunto coerente e integrado de processos de engenharia e de gerenciamento de software bem definidos. Um processo bem definido pode ser caracterizado, a medida que inclui critérios de disponibilidade, entradas, padrão e procedimentos para desempenho do trabalho, mecanismos de verificação, como por exemplo critérios iguais de revisões, saídas e conclusões. Como o processo de software é bem definido, o gerenciamento apresenta uma boa percepção no progresso técnico de todos os projetos.

Questões de Ciclo de Vida

No CMM, as questões de ciclo de vida são incluídas em mais de uma KPA (Key Process Area). Ao apresentar os conceitos do CMM pela primeira vez em uma organização, as pessoas geralmente não entendem o termo “processo de software” e o assimilam ao “ciclo de vida de software”. O ciclo de vida do projeto inclui questões de negócios e finanças, tanto quanto questões de desenvolvimento. Key Process Areas Nível 2 e algumas Key Process Areas Nível 3 são relacionadas ao ciclo de vida do projeto. São elas:

- Objetivo 1 de Gerenciamento de Requisitos: Os requisitos do sistema alocados para o software são controlados para estabelecer uma linha de base para uso de engenharia e gerenciamento de software.
- Objeto 1 de Planejamento de Projeto de Software: As estimativas de software são documentadas para planejar e acompanhar o projeto de software.

- Objetivo 2 de Planejamento de Projeto de Software: As atividades e confirmações de projetos de software são planejadas e documentadas.
- Objetivo 1 de Acompanhamento do Projeto de Software: Os resultados e desempenhos reais são acompanhados em oposição aos planos do software.
- Objetivo 2 de Acompanhamento do Projeto de Software: Mudanças para confirmações de software são ajustadas pelo grupo e pessoas afetadas.
- Objetivo 1 de Gerenciamento de Configuração de Software: Os produtos de trabalho de software selecionados são identificados, controlados e disponíveis.
- Objetivo 2 de Gerenciamento de Configuração de Software: Mudanças para produtos de trabalho de software identificados são controladas.
- Objetivo 3 de Gerenciamento de Configuração de Software: Os grupos e pessoas afetados são informados sobre o status e conteúdo da linha de base do software.
- Objetivo 1 de Engenharia de Produto de Software: As tarefas de engenharia de software são definidas, integradas e consistentemente desempenhadas para produzir o software.
- Objetivo 2 de Engenharia de Produto de Software: Os produtos de software são mantidos consistentes uns com os outros.

Em outras palavras, o CMM requer que:

1. Uma linha de base seja implicitamente definida para considerar o ambiente de negócios do aplicativo a ser desenvolvido, para definir as prioridades para os requisitos do projeto e tomar todas as decisões necessárias relacionadas aos conteúdos funcionais ou não-funcionais do projeto.
2. Os planos dos projetos são estabelecidos e uma lista de atividades, funções e responsabilidades, marcos e artefatos realistas e confiáveis precisa ser configurada. Os planos de projetos considera os riscos.
3. Os planos de projetos são monitorados e a equipe do projeto reage no caso de maiores desvios de conteúdo do recurso ou do ambiente/organização do projeto.
4. Quaisquer alterações no projeto são identificadas e analisadas. Decisões adicionais são tomadas e relatadas na linha de base e planos do projeto.
5. Os métodos e ferramentas são selecionados e aplicados, para garantir o melhor desempenho do projeto.

O Rational Unified Process preenche esses requisitos nas principais tarefas da fase de Iniciação, da seguinte maneira:

- Obtém a visão do projeto e desenvolve um modelo de negócios correspondendo o item 1 na lista anterior.
- Organiza e planeja o projeto
- Estima os riscos potenciais. (Esse item é relatado no item 2 na lista anterior.)
- Implanta a política do Unified Change Management, a qual é semelhante aos itens 3 e 4 nas listas anteriores.

A meta dominante da fase de Iniciação é atingir o consenso entre todos os investidores sobre os objetivos do ciclo de vida do projeto. A fase de Iniciação tem muita importância principalmente para os esforços de desenvolvimentos novos, nos quais há muitos riscos de negócios e de requisitos que devem ser tratados para que o projeto possa prosseguir. Para projetos que visam melhorias em um sistema existente, a fase de iniciação é mais rápida, mas ainda se concentra em assegurar que o projeto seja compensatório e que seja possível fazê-lo.

No final da fase de Iniciação, os objetivos de ciclo de vida do projeto são examinados e uma decisão de continuar com o projeto ou cancelá-lo é alcançada.

Os artefatos essenciais para revisão são:

- Visão do Projeto
- Caso de Negócios
- Lista de Riscos (consulte a seção em Gerenciamento de Riscos)
- Plano de Desenvolvimento de Software

- Plano de Iteração
- Caso de Desenvolvimento (consulte a seção em Seleção do Processo e Adaptação)

Esses artefatos satisfazem várias metas de Key Process Areas do CMM listadas no início dessa seção.

Obter a Visão do Projeto e Desenvolver um Modelo de Negócios

Um modelo de negócios define os casos de uso de negócios a partir do ponto de vista interno dos trabalhadores de negócios. O modelo define como as pessoas que trabalham no negócio e como os itens que elas manipulam e utilizam—"as classes e os objetos de negócios"—devem estar relacionados uns com os outros, estaticamente e dinamicamente, para produzir os resultados esperados. O modelo também dá ênfase nas funções desempenhadas na área de negócios e suas responsabilidades ativas. Juntos, os objetos das classes do modelo devem ser capazes de desempenhar todos os casos de uso de negócios.

Com base no modelo de negócios, o Rational Unified Process identifica uma tarefa nomeada Desenvolver Visão, a qual possui as seguintes finalidades:

- Estabelecer um acordo sobre quais problemas precisam ser resolvidos.
- Identificar investidores do sistema.
- Definir limites do sistema.
- Descrever os recursos principais do sistema.

Organizar e planejar o projeto.

Assim como o processo de software é influenciado pelas características do projeto, a organização do projeto também sofre essa influência. A estrutura padrão, aqui apresentada, e ilustrada na Figura 1, precisa ser adaptada para refletir os efeitos de fatores, como os que aparecem listados:

- Contexto de Negócios
- Tamanho do Esforço para Desenvolvimento do Software
- Grau de Inovação
- Tipo de Aplicativo
- Processo de Desenvolvimento Atual
- Fatores Organizacionais
- Complexidade Técnica e Gerencial

Esses fatores são considerados no RUP como discriminantes do processo, os quais causam impacto na opção de estrutura do projeto.

A estrutura do projeto é principalmente definida através do:

1. Comprimento de cada iteração
2. Número de iterações

Uma iteração é certamente um mini-projeto completo, que passa por todos os fluxos de trabalho principais e resulta, na maioria dos casos, em um sistema executável, porém incompleto, denominado release.

Para determinar o número de iterações, são possíveis diversas variações, dependendo dos riscos, do tamanho e da complexidade.

Se o produto for destinado a um domínio totalmente novo, poderá ser necessário incluir algumas iterações na fase de Iniciação para consolidar os conceitos, mostrar diversos modelos para uma seção cruzada de clientes ou usuários finais ou para construir uma resposta sólida a um pedido de proposta.

Se o produto for grande, complexo e desenvolvido durante um longo período de tempo, planeje três ou mais iterações na fase de Construção.

Durante a vida útil de um projeto, a organização se desenvolverá a fim de suportar a estrutura de divisão de trabalho capturada no plano de projeto. Isso é mostrado na Figura 1, [7].

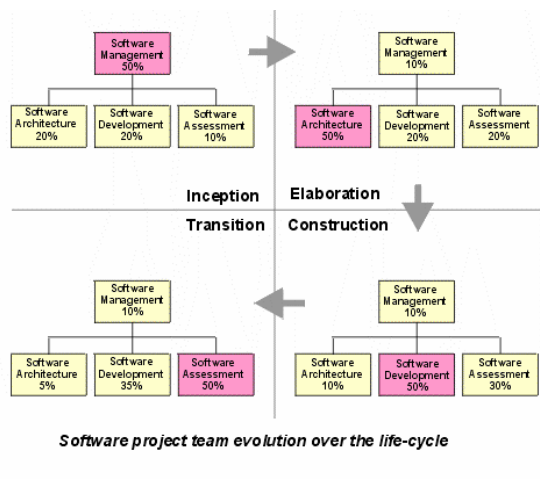


Figura 1: Evolução da Equipe do Projeto de Software

Essa evolução enfatiza um conjunto diferente de atividades em cada fase:

- Equipe de Iniciação: uma organização com foco no planejamento, com suporte suficiente das outras equipes para garantir um consenso dos planos de todas as perspectivas.
- Equipe de Elaboração: uma organização com foco na arquitetura, onde as forças incentivadoras do projeto residem na equipe de arquitetura de software, com suporte das equipes de desenvolvimento de software e de avaliação de software, quando necessário, para atingir uma linha de base de arquitetura estável.
- Equipe de Construção: uma organização equilibrada em que a maior parte das atividades reside nas equipes de desenvolvimento e de avaliação de software.
- Equipe de Transição: uma organização com foco no cliente, na qual o feedback de uso orienta as atividades de implantação.

A migração entre as equipes durante essa evolução, garante o conhecimento e a capacidade dentro do projeto.

Implantar a Política Unified Change Management

O UCM (Unified Change Management) é a abordagem da Rational Software para gerenciar as mudanças no desenvolvimento de sistemas de software, desde os requisitos até o lançamento. O UCM estende-se pelo ciclo de vida do desenvolvimento, definindo como gerenciar as mudanças nos requisitos, nos modelos de design, na documentação, nos componentes, nos casos de teste e no código-fonte.

Um aspecto principal do modelo do UCM, é que ele unifica as atividades utilizadas para planejar e controlar o progresso do projeto e os artefatos que sofrem mudanças. O modelo do UCM é realizado pelo processo e por ferramentas. Rational ClearCase® e Rational ClearQuest® são tecnologias de base do UCM. O ClearCase gerencia todos os artefatos produzidos por um projeto de software, incluindo artefatos do sistema e artefatos de gerenciamento de projeto. O ClearQuest gerencia as tarefas, os defeitos e os pedidos de melhoria do projeto (genericamente referidos como atividades) e fornece as ferramentas de elaboração de gráficos e de relatórios necessárias, para controlar o progresso do projeto.

Seleção de Processo e Adaptação

O Nível de Maturidade 3 inclui três KPAs que atingem a organização completa e quatro KPAs que atingem a organização, o gerenciamento e a engenharia do projeto. O processo de software para atividades de gerenciamento e engenharia é documentado, padronizado e integrado em um processo de software de organização considerável. Todos os projetos utilizam uma versão documentada e aprovada do processo da organização, para desenvolvimento e manutenção de software. As KPAs do OPF (Organization Process Focus), OPD (Organization Process Definition) e TP (Training Program) garantem que a organização identifica as boas práticas do projeto no Nível 2 e as documenta como um padrão para a organização. Também focalizam o gerenciamento de habilidades com base nas necessidades identificadas pelos projetos, nas lições aprendidas de projetos anteriores e na missão e visão da organização para o futuro. No entanto, a maturidade não satisfaz os requisitos do Nível 3, se não houver orientações para adaptar o processo padrão para um projeto específico, a fim de obter sucesso, gerenciar os riscos e melhorar o desempenho.

Portanto, para a maioria dos assessores de liderança e a comunidade do CMM, o Nível 3 é caracterizado pela KPA do ISM (Integrated Software Management), que possui a finalidade de integrar as atividades de engenharia e de gerenciamento de software em um processo de software coerente e definido. Esse processo definido é a customização mais adequada do processo padrão, que ajuda a satisfazer as necessidades e restrições do cliente, demanda de mercado e estratégia de negócios.

O ISM é definido por dois objetivos no modelo do CMM. O primeiro envolve a adaptação, enquanto que o segundo requer gerenciamento de projeto. Atividade 10 no ISM: “os riscos de software do projeto são identificados, avaliados, documentados e gerenciados, de acordo com um procedimento documentado” é o segundo indicador chave para uma organização de Nível 3. O gerenciamento de riscos é requerido no Nível 2, mas as organizações do Nível 3 têm obtido êxito em riscos de melhores expectativas e conseguem demonstrar um sistema de decisão empresarial. A seção Gerenciamento de Riscos a seguir, desenvolve esses aspectos.

No RUP, o conceito chave do Caso de Desenvolvimento e da Disciplina de Ambiente preenchem esses requisitos. Um Caso de Desenvolvimento é uma configuração estática do produto Rational Unified Process; ou seja, um processo de negócios destinado à engenharia de software adaptada a um projeto, produto e organização específicos. Ele é focalizado em o que e como fazer isso, e também fornece uma visão geral do processo a ser seguido, algo que seja compreendido por todos os participantes do projeto.

O Rational Unified Process lista os componentes de um processo que provavelmente serão modificados, personalizados, incluídos ou suprimidos de um determinado Caso de Desenvolvimento.

- **Disciplinas**
Raramente um projeto de software ignoraria uma das disciplinas principais—como Análise e Design, Implementação e outras—completamente
- **Artefatos**
É muito mais provável que os projetos se diferenciem pelos artefatos que eles têm que produzir, atualizar e liberar.
- **Tarefas**
As tarefas podem variar por pelo menos dois motivos: as tarefas que utilizam artefatos como entrada e que produzem, ou atualizam, artefatos como saída são afetadas pela modificação desses artefatos. Particularmente, se algum artefato ou algum elemento de informação em um artefato, não for mais necessário, as etapas correspondentes poderão ser suprimidas ou modificadas de forma significativa. As tarefas também são modificadas para introduzir determinadas técnicas, métodos e ferramentas pertencentes ao domínio específico do aplicativo ou a seus conhecimentos específicos de desenvolvimento, como por exemplo, etapas de design, linguagens de programação, ferramentas para geração automática de códigos, técnicas de medição e assim por diante.

O engenheiro de processo é responsável pela configuração do processo, decidindo qual será a aparência do processo de desenvolvimento, “instalando” o caso de desenvolvimento na organização de desenvolvimento (equipe, projeto ou empresa) e ensinando aos desenvolvedores como utilizá-lo.

Uma vez configurado o caso de desenvolvimento pelo engenheiro de processo, o gerente do projeto cria instâncias e o executa em um determinado projeto. Isso é geralmente chamado de execução do processo.

À medida que o processo se desenrola, são aprendidas muitas lições, que são usadas pelo engenheiro de processo como feedback para aprimorá-lo.

Gerenciamento de Riscos

O Gerenciamento de Riscos é amplamente tratado no CMM, principalmente, no Nível 2 e mais profundamente no Nível 3. As atividades de risco específicas são listadas para atender as metas de Software Project Planning, Software Project Tracking e Oversight, and Integrated Software Management de Key Process Areas. Os requisitos do CMM seguem as práticas observadas em muitas organizações: no Nível 2, os projetos geralmente identificam e avaliam riscos regularmente. Porém, existe pouca proatividade. No nível 2, o gerenciamento de riscos ainda não é bem tratado/entendido e há problemas com o acompanhamento de projetos e identificação antecipada de riscos.

Pelo contrário, em organizações no Nível 3, os riscos são identificados, avaliados e diminuídos, e as equipes demonstram uma expectativa muito boa. Na maioria das equipes avançadas, as medidas quantitativas são utilizadas para tomar decisões.

Nossa longa experiência em avaliações baseadas no CMM e Software Process Improvement, destaca as dificuldades enfrentadas pelas unidades de software no gerenciamento de riscos. Esse é um defeito ainda observado, apesar de todos os esforços dedicados para controle e medida do projeto.

O uso do RUP e da abordagem iterativa é muito útil para ter sucesso no gerenciamento de riscos do projeto. Todo o Rational Unified Process é orientado pelo Gerenciamento de Riscos.

O risco controla os planos de iteração; as iterações são planejadas considerando riscos específicos na tentativa de limitar o risco ou reduzi-lo. A Lista de Riscos é revista periodicamente para avaliar a eficácia das estratégias de diminuição de riscos e, conseqüentemente, orientar as revisões no plano de projeto e nos planos de iteração subseqüentes.

O segredo do gerenciamento de riscos é não esperar até que haja risco (e torne-se um problema ou defeito) para decidir o que fazer em relação a ele. Uma mudança de alguns graus no percurso de um vôo transcontinental produz um efeito significativo no local de aterrissagem do avião; de modo semelhante, gerenciar o risco antecipadamente é quase sempre menos dispendioso e penoso do que tentar solucioná-lo depois que virar um fato.

Quatro categorias de riscos são identificadas aqui:

1. Riscos de Recursos
 - Organização
 - Fundos
 - Pessoas
 - Tempo
2. Riscos do Negócio
3. Riscos Técnicos
 - Riscos de Escopo
 - Riscos Tecnológicos
 - Risco de Dependência Externa
4. Riscos de Planejamento

O Plano de Gerenciamento de Riscos e a Lista de Riscos são dois artefatos identificados pela disciplina de Gerenciamento do Projeto do RUP. Recomenda-se a identificação e a avaliação dos riscos.

Medida

É um recurso comum no CMM. Um recurso comum existe para cada Key Process Area do Nível 2 ao Nível 5 e indica quando uma prática é instituída. As métricas são tão importantes que na nova versão do SW-CMM—isto é, aquela incorporada na estrutura do CMMI, a engenharia do Integrated CMM System + engenharia do Software—há uma meta baseada na medida para cada Key Process Area.

O que é essencial no CMM é medir os processos para determinar sua adequação no Nível 2 e sua eficácia no Nível 3. A mudança de relatório para analisar e agir de acordo com as métricas, geralmente, é difícil, mas um sinal claro de que os projetos estão mais desenvolvidos. Os sucessos menores e iniciais que gerenciam um projeto de forma considerável, levam à aceitação e entendimento dos benefícios de medida.

Além desses requisitos gerais, algumas KPAs têm requisitos específicos para medidas, como por exemplo Software Project Planning, Software Project Tracking and Oversight e Integrated Software Management. Estes estão relacionados a dados de projetos que permitem estimativas e controles de projetos.

O RUP fornece orientação para o uso de métricas. Os artefatos Plano de Medidas e Medida precisam ser produzidos na fase de Iniciação.

O artefato Medidas de Projetos armazena os dados de métricas do projeto. Ele é atualizado conforme as medidas são ou tomam-se disponibilizadas. Além disso, ele contém as métricas derivadas que são calculadas a partir dos dados primitivos e deve armazenar informações, como por exemplo procedimentos e algoritmos, sobre como elas são obtidas. Os relatórios sobre o status do projeto—por exemplo, o progresso em relação às metas (funcionalidade, qualidade e assim por diante), despesas e outros consumos de recursos—são produzidos utilizando as medidas do projeto. Com bastante frequência ou até mesmo de modo aparentemente contínuo, as exibições do status do projeto são possíveis utilizando a abordagem do Rational Project Dashboard, por exemplo, em que os agentes de coleta de dados do software automatizado fornecem exibições em tempo real do status do projeto.

A documentação do Rational edita um conjunto inicial de métricas, o qual é simples o suficiente para ser iniciado. As métricas para determinados aspectos do projeto incluem:

- Progresso de tamanho e complexidade
- Estabilidade de taxa de mudança nos requisitos ou implementação, tamanho ou complexidade
- Modularidade do escopo de mudança
- Qualidade do número e do tipo de erros
- Maturidade da frequência de erros
- Recursos de despesas do projeto versus despesas planejadas

O RUP Corresponde aos Requisitos do CMM no Nível da Organização

As organizações de capacidade do processo de software de Nível 2 podem ser resumidas como disciplinadas, porque o planejamento e o controle do projeto de software são estáveis e sucessos anteriores podem ser repetidos. O processo do projeto está sob controle efetivo de um sistema de gerenciamento de projeto, seguindo planos realistas baseados no desempenho de projetos anteriores.

No Nível Definido (Nível 3), o processo padrão para desenvolvimento e manutenção de software é documentado através da organização, incluindo os processos de engenharia e de gerenciamento de software, e esses processos são integrados em um padrão coerente. O processo padrão é mencionado no decorrer do CMM como o Processo de Software Padrão da Organização. Os processos estabelecidos no Nível 3 são utilizados e alterados, conforme apropriado; para ajudar no desempenho de forma mais efetiva dos gerentes de software e da equipe técnica. A organização explora práticas de engenharia de software efetivas ao padronizar seus processos de software. Há um grupo responsável pelas atividades do processo de software da organização; por exemplo, uma engenharia de software ou SEPG. É implementado um programa de treinamento total da organização, para garantir que a equipe e os gerentes tenham o conhecimento e as habilidades necessárias para preencher suas funções designadas.

A capacidade de processo de software de organizações de Nível 3 pode ser resumida como padrão e consistente, porque as atividades de engenharia e de gerenciamento de software são estáveis e repetidas. Dentro das linhas de produtos estabelecidas, o custo, o planejamento e a funcionalidade estão sob controle e a qualidade do software é controlada. Essa capacidade do processo baseia-se em um entendimento comum em toda a organização das atividades, funções e responsabilidades em um processo de software definido.

Aprimoramento do Processo

A capacidade implica que uma organização consegue aprender com os erros do passado, especialmente de terceiros, e podem converter as lições aprendidas em evoluções do processo. Os ciclos de iteração de aprimoramento apenas serão concluídos, se as medidas forem definidas para quantificar o aprimoramento.

Quando a maturidade da organização é aprimorada, o processo padrão é mudado. A obtenção do Nível 2 para o Nível 3 significa que todas as boas práticas dentro dos projetos são institucionais e que há um processo de avaliação adequado, que ajudará na identificação de boas práticas através de projetos, as quais serão documentadas no OSSP (Organization Standard Software Process). Há os requisitos da Key Process Area do OPF (Organization Process Focus). Em seguida, o OSSP será aperfeiçoado na base de lições aprendidas dentro de projetos através da organização. Como os projetos utilizam o mesmo padrão, o repositório de experiências e lições aprendidas é facilmente enriquecido e o próprio OSSP pode se beneficiar de várias experiências.

A disciplina de Ambiente do RUP desenvolve uma abordagem semelhante. O conceito “Implementando um Processo em uma Organização” explica o que é feito no nível organizacional, para implementar o processo e as ferramentas em uma organização de desenvolvimento.

A implementação de um novo processo em uma organização de desenvolvimento de software pode ser descrita utilizando as quatro fases do RUP: Iniciação, Elaboração, Construção e Transição.

As Etapas para Implementar o Processo e Ferramentas em uma Organização

O OSSP do CMM é um novo processo na organização e sua definição pode seguir as quatro fases do RUP.

A decisão se desenvolver um ambiente de desenvolvimento total da organização, que cada projeto de desenvolvimento de software pode utilizar com as adaptações necessárias, é essencial, mas deve existir um determinado nível de maturidade.

Se você resolver desenvolver um ambiente total de organização, será necessário iniciar um projeto para desenvolver o ambiente de desenvolvimento da organização. E se resolver iniciar esse tipo de projeto, deve ser deixado claro que essa equipe do projeto funcionará muito rigorosamente com as equipes de projetos de desenvolvimento de software. O RUP também recomenda que você o considere como um projeto específico. Isso preenche os requisitos do CMM aqui novamente.

O projeto de implementação de processo é dividido em várias fases, onde todas as quatro etapas são desenvolvidas em cada fase, até que o projeto esteja pronto e o processo e as ferramentas sejam implantados e utilizados com êxito por toda a organização.

Um projeto de implementação de processo pode ser dividido em fases

As quatro fases consideram:

- Fase 1: Vender o projeto de implementação de processo para os responsáveis.
- Fase 2: Tratar dos riscos principais.
- Fase 3: Concluir tudo—gabaritos, orientações, exemplos de Casos de Desenvolvimento estão prontos e um currículo de treinamento é adequado.
- Fase 4: Implantá-los em toda a organização.

Essas fases podem ser nomeadas Iniciação, Elaboração, Construção e Transição respectivamente, conforme elas são destinadas a um projeto de desenvolvimento de software, utilizando o Rational Unified Process.

As fases do mesmo projeto serão definidas toda vez que o processo padrão precisar ser desenvolvido, a fim de aumentar seu desempenho com base em lições aprendidas em um projeto específico ou de acordo com a tecnologia específica.

O RUP também define o conceito Gerenciando Mudança Organizacional, o qual é o segundo plano global do Software Process Improvement. As recomendações para implementação bem-sucedida de uma mudança de processo servem para:

- Identificar os agentes de mudança em vários níveis nas organizações.
- Planejar a mudança em etapas pequenas, razoáveis e moderadas.
- Comunicar as mudanças utilizando o idioma de nível base apropriado para o nível da organização.

Essas recomendações são semelhantes àquelas fornecidas pela abordagem IDEAL, a partir das quais elas são inspiradas.

Finalmente, o RUP define um agente específico chamado “Mentor”. Um mentor é alguém que ensina e orienta as equipes de projetos sobre o que é necessário e quando isso é preciso. Formas típicas de aconselhamento são:

- Líder de workshop
Algumas atividades são melhor desempenhadas em um grupo; por exemplo, localizar agentes e utilizar casos durante a modelagem de caso de uso. No decorrer de tais atividades, é importante ter um líder de modelagem, que seja um especialista de processo.
- Especialista de Processo
O especialista de processo é uma pessoa de suporte on-site para o projeto. A tarefa do especialista do processo é ajudar os desenvolvedores a utilizar o processo e o modelo, assim que possível.
- Suporte do Coordenador de Projeto
Um especialista de processo pode ajudar o coordenador de projeto a planejar e conduzir o projeto. Às vezes, o coordenador de projeto tem pouca ou nenhuma experiência sobre o processo em questão.
- Revisor
Uma maneira com custo reduzido de transferir o conhecimento é precisar que um especialista de processo reveja os resultados de cada fase. Um revisor de processo também apresenta o valor na revisão de qualquer adaptação de processo desempenhada por um projeto.

Esse Mentor preenche uma função diferente do SEPG (Software Engineering Process Group) no CMM? Não, é o mesmo tipo de função.

Recursos e Habilidades

No nível de organização, o CMM identifica um “Programa de Treinamento” Key Process Area, o qual poderia ser melhor denominado como “gerenciamento de habilidades”, de acordo com a real preocupação aqui. Uma organização desenvolvida precisa identificar e planejar as necessidades para habilidades de médio a longo prazo, e gerenciá-las para que sejam obtidas no devido tempo.

A finalidade de um Programa de Treinamento é desenvolver as habilidades e conhecimento de pessoas, para que suas funções possam ser desempenhadas de forma efetiva e eficiente. O treinamento é uma responsabilidade organizacional, mas os projetos de software devem identificar as habilidades necessárias e fornecer o treinamento necessário, quando as necessidades do projeto são exclusivas. A disciplina de gerenciamento no RUP considera essas questões, ao nomear um projeto.

A Key Process Area do Programa de Treinamento possui três metas:

- Meta 1: As atividades de treinamento são planejadas.
- Meta 2: O treinamento é fornecido para desenvolvimento das habilidades e conhecimento necessário, para desempenhar o gerenciamento de software e funções técnicas.
- Meta 3: As pessoas no grupo de engenharia de software e em grupos relacionados ao software recebem o treinamento necessário, para desempenhar suas funções.

Vários aspectos do RUP atendem esses requisitos. As funções são precisamente identificadas e suas competências são definidas. Normalmente, uma função é desempenhada por uma pessoa ou um grupo de pessoas que trabalham juntas em equipe. Um membro da equipe do projeto geralmente desempenha muitas funções distintas; da mesma forma que uma pessoa pode desempenhar vários 'papéis', ela pode realizar muitas funções distintas.

Os papéis não são pessoas; e sim, eles descrevem como as pessoas devem se comportar no negócio e quais são suas responsabilidades.

Enquanto a maioria das funções é realizada por pessoas que fazem parte da organização, as pessoas de fora da organização de desenvolvimento desempenham uma função importante; por exemplo, aquela do investidor do projeto ou do produto sendo desenvolvido.

Cada tipo de função é definido com cuidado pelas habilidades e conhecimento que um membro da equipe deve demonstrar. Com base nessas definições, uma organização consegue derivar as habilidades ausentes e as características do treinamento necessário, para conseguir essa função desempenhada de forma eficiente.

As orientações do RUP formam um bom conjunto de regras a serem fornecidas como treinamento. O desenvolvedor do curso possui uma das categorias de função listadas.

O único aspecto que não é totalmente abrangido pelo RUP e é requerido pelo CMM é o treinamento de identificação, planejamento e entrega, para o desenvolvimento de novas habilidades no futuro. Imagine que a estratégia da organização é centralizar sua atividade no e-commerce um ano de agora em diante; esse objetivo precisa ser também decomposto e, provavelmente, irá impactar o tipo de desenvolvimentos desempenhados, bem como o tipo de ambientes utilizados. As equipes precisam ser preparadas para essa mudança e o treinamento necessário deve ser definido, desenvolvido ou adquirido externamente. A Key Process Area de Treinamento no Nível 3 requer que essas questões sejam gerenciadas, acompanhadas e registradas com cuidado.

- A seção três desse documento fornece um mapeamento detalhado de alguns perfis de maturidade com os recursos do RUP. Esse documento poderia ter criado um paralelo sistemático entre cada meta da referência do CMM e as orientações correspondentes do RUP; deliberadamente, isso não foi feito e, em vez disso, decidiu-se focalizar os sinais chave da alta maturidade.
- Por um lado, a qualidade do software é tratada dentro do RUP através da atividade de teste e por outro lado, através da tarefa Desenvolver Plano de Controle de Qualidade. No entanto, nossa demonstração destaca nossa experiência e observação sobre o que realmente significa a alta maturidade: o que é diferente em organizações do Nível 3 comparado ao Nível 1 ou em organizações inferiores ao Nível 2.
- A próxima seção é dedicada a algum feedback de adotantes iniciais dessas tecnologias combinadas.

Pragmática: Como Iniciar o RUP

A pergunta mais comum feita é: Alguém fez isso antes?

Provavelmente, essa pergunta será feita mais de uma vez, quando os conceitos anteriores são apresentados aos gerentes e profissionais liberais. Tornar-se mais desenvolvido, mais direcionado ao processo, gerenciar riscos e ser bem-sucedido é atrativo para qualquer negócio de software. No entanto, isso é realmente possível em um contexto de mudança, com requisitos que estão mudando rapidamente e equipes que precisam aplicar tecnologias nas quais não são treinadas?

Os relatos de clientes do Rational incluem a experiência de Computing Devices International: Computing Devices é uma empresa de eletrônicos e informações de defesa de liderança. As informações continuam sendo uma vantagem competitiva chave em quase todo o segmento de mercado. Para a Computing Devices International, ela não é apenas informativa—sua missão é fornecer soluções de ponta a ponta que entregam informações indispensáveis em qualquer hora e lugar on demand.

Precisam garantir a entrega consistente e no horário de soluções de alta qualidade. Os benefícios chave de sua experiência incluem o seguinte:

- Uma aceleração do processo de desenvolvimento através de uma abordagem de design iterativa
- Uma diminuição no tempo de desenvolvimento de 3 anos para 19 meses
- A capacidade de garantir a entrega confiável e no horário
- Um aumento na satisfação do cliente
- Uma redução de 33% nos custos de desenvolvimento

Os sistemas sofisticados que o Computing Devices desenvolve para seus clientes são altamente dependentes de software, o qual precisa ser projetado para requisitos rigorosos. Primeiro, o Computing Devices volta sua atenção para o processo de gravação de software crítico com segurança.

As unidades de negócios diferentes dentro da organização estavam utilizando uma variedade de ferramentas, com base no design estruturado ou métodos "em cascata". Conseqüentemente, o processo de desenvolvimento era lento e faltou uma abordagem consistente. Isso conduziu a prazos finais incorretos, custos mais altos e qualidade de software que nem sempre atendia aos altos padrões. O Computing Devices percebeu que precisava tomar uma atitude—rápida.

Skandia-IT é outro exemplo de portfólio do Rational. Nesta organização, nove sistemas de seguro extensos foram entregues a tempo ou antes do planejamento, em um único período de 12 meses.

Foi acelerado o recrutamento de novos talentos, atraindo os desenvolvedores e consultores com um processo de desenvolvimento de última geração, satisfação garantida do cliente pela utilização de casos de uso para solicitar requisitos, modelar sistemas e direcionar o desenvolvimento, e foi desenvolvido rapidamente uma arquitetura flexível de 3 camadas, agrupando os sistemas legados e utilizando-os como componentes. Esses eram alguns dos desafios conquistados do Skandia-IT.

Os gerentes no Skandia-IT explicaram, "Há vários motivos pelos quais todo o processo foi alterado. O mais importante era que todos os negócios foram reorganizados. O processo de desenvolvimento foi alterado em paralelo para a mudança de uma organização orientada pelo produto a uma organização orientada pelo cliente. Novos sistemas fornecem ao próprio cliente a possibilidade de tratar de muitas coisas, por telefone ou através da World-Wide Web. Os funcionários no Skandia podem se concentrar mais tempo em se tornarem especialistas em seguros, em vez de aprender como utilizar o sistema. O tempo economizado é investido em relações com o cliente."

O Skandia-IT investiu pesadamente na instalação do novo processo. É um investimento que foi pago. Melhor qualidade, tempo de entrega previsível e efeitos secundários, tal como o fato de que hoje é mais fácil recrutar chefes e desenvolvedores de projetos com experiência.

Existem outros exemplos de empresas que focalizam processos e iniciam com o RUP, a fim de acelerar a implantação. Q-Labs está trabalhando com a Lysis, uma empresa de rápido crescimento que desenvolve o software para difusão na Internet.

Em um ambiente de inicialização, as tecnologias e metodologias de processo focalizadas em níveis avançados de maturidade de processo podem não ser adequadas. Esse tipo de organização é principalmente rápida, reativa e inovativa, e o contraste entre a necessidade de acelerar a publicação dos produtos e a necessidade de maximizar o controle do processo, surge a pergunta: para qual extensão o tema de processo favorece essa diversidade de pontos de vista?

As características dessa empresa são:

- São uma unidade de software de crescimento rápido.
- Desenvolvem mais rápido (menos de 6 meses).
- Entregam melhores produtos utilizando tecnologias de objetos de liderança.
- Adaptam-se a mercados dinâmicos.

Esse ambiente realmente não é diferente de outros segmentos de mercado de software, exceto que a escala de tempo é completamente diferente, com possibilidades de previsões é equivalente à sobrevivência ou à decadência, e o trabalho em equipe precisa ser mudado radicalmente muito rápido. Tempo para aprendizagem é quase inexistente e a melhor opção é iniciar a partir de práticas existentes e bem comprovadas. Isso é exatamente o que a Lysis fez, quando eles adotaram ambos os CMM para levar ao crescimento e ao RUP como um segundo plano para seus processos de instalação.

A necessidade de definir práticas de software não é questionada. Atende ao entendimento, comunicação, execução e gerenciamento. Também forma uma base para formulação de relacionamentos com outras organizações, interna e externamente. No caso da Lysis, os relacionamentos com marketing e vendas e com subcontratantes de software potenciais são os objetivos. Os processos definidos precisam permanecer flexíveis: as mudanças podem ocorrer rapidamente em planejamentos pessoais, de infra-estrutura, de release e em tecnologias avançadas. Os processos de desenvolvimento precisam se adequar a elas. É requerida uma resposta rápida e na hora certa. São recomendados os processos iterativos com ciclos de iteração breves, portas direcionadas a riscos e gerenciamento construtivo. É necessária a capacidade de repetição de um processo através de componentes de linhas de produtos. As definições de processos ajudam a informar o pessoal novo que se une à empresa, como é o crescimento.

No CMM, a definição de processo (Nível 3 de maturidade) segue a repetitividade do processo (Nível 2 de maturidade), mas em tal empresa iniciada a partir de um processo padrão como o RUP, é útil promover e estabelecer a repetitividade, trazendo, através disso, resultados significativos.

A abordagem utilizada pela Lysis, começou com uma avaliação de práticas de desenvolvimento em oposição aos requisitos do CMM. Isso resultou em uma boa aquisição de orientação de processos pelas equipes que estavam mais interessadas na programação, tecnologias de objetos e ambiente de ferramentas. Com base nas observações, um plano de ação para definir os processos foi estabelecido com fortes restrições de tempo. As metas de negócios, configuradas pelo gerenciamento principal para suportar a SPI, incluem:

- Abreviar o processo de desenvolvimento para ser o primeiro nos mercados dirigidos
- Acelerar a transição de modo do projeto para modo de produção
- Gerenciar o crescimento da empresa
- Satisfazer as pessoas
- Manter a reatividade de inicialização
- Aprimorar a possibilidade de previsões e visibilidade (para IPO, resultados trimestrais necessários)
- Considerar aprimoramento da qualidade dos produtos
- Gerenciar custos e lucros prováveis
- Otimizar a satisfação do cliente
- Identificar e gerenciar a interface com os parceiros (para a parte subcontratada do trabalho)

O próprio programa SPI foi estabelecido como um projeto, utilizando a estrutura do RUP onde:

- Iniciação = Avaliação
- Elaboração = Planejamento do Projeto + Definição e Protótipo de Repositório baseado na Web
- Construção = Processos Definidos + Implementação e Ferramenta

- Transição = Projeto Piloto + Implantação

Os pacotes de trabalho são definidos para tratar de Key Process Areas do CMM. Cada uma delas possui objetivos, atividades a serem desempenhadas, produtos e um processo de revisão. Como exemplo, os objetivos para Planejamento de Software são:

Objetivo da ação:

- Documentar a prática adequada atual de planejamento de software
- Definir o conjunto de organizações genéricas para os projetos
- Definir funções e responsabilidades para planejamento
- Definir o processo de confirmação para nível de projeto
- Definir diretrizes de ciclo de vida e de adaptação do projeto genérico para um conjunto padrão de processos de projetos
- Definir um WBS (Work Breakdown Structure) padrão
- Relacionar lista padrão de produtos
- Definir o gabarito e as diretrizes para o Plano de Desenvolvimento
- Escrever o processo de planejamento do projeto
- Definir habilidades e treinamentos necessários para planejamento

Optar por iniciar com o RUP irá acelerar algumas dessas ações e garantirá os benefícios de uma organização a partir de uma forte experiência. A iniciativa SPI não está concluída, mas alguns fatores de sucesso visíveis já foram observados: o desenvolvimento de novos produtos é melhor estruturado; os planos existem e são regularmente revisados. Ações corretivas são tomadas e a visibilidade fora da equipe de software tem melhorado muito.

O Rational Software relata um caso semelhante com o E-corporation. O E-corporation é um European Interactive Architect, o qual está direcionando empresas de todo o mundo à fronteira de negócios digitais. O E-corporation combina habilidades estratégicas, artísticas e de software, para entender, criar e implementar aventuras na Internet e no e-business.

Os benefícios chave de implantação do RUP pelo E-corporation são:

- A melhor comunicação é fornecida entre os membros da equipe, o que aumenta sua eficiência.
- As expectativas do cliente são atendidas mais precisamente.
- É fornecida uma estrutura para atender demandas de alta qualidade e de alta velocidade.
- O risco com o desenvolvimento iterativo é reduzido e é utilizado um processo comprovado baseado em boas práticas de conhecimento do especialista e de confirmação de segmento de mercado.

A maior vantagem, O E-corporation teve experiências em trabalhar com um processo comum em todos os projetos, é a reutilização efetiva de componentes, experiências e atividades. O Rational Unified Process pode ser constantemente ajustado com precisão, quando a organização adquire experiência em vários projetos. Como todas as atividades são documentadas seguindo uma estrutura e idioma padrão, muita experiência pode ser adquirida de projetos anteriores aos novos projetos.

O E-corporation planeja estender o Rational Unified Process a todos os seus projetos em todos os sites, para atuar também sobre a unificação de suas equipes. O Rational Unified Process oferece às suas equipes de desenvolvimento o poder de aplicar princípios comprovados para otimizar todos os esforços, simplificar a comunicação e assegurar que as soluções de qualidade atendam a todos os requisitos.

Estes são exemplos de como várias organizações têm observado os benefícios, através do acoplamento das diretrizes do Rational Unified Process e do CMM. Atualmente, estamos trabalhando com outras organizações, onde o RUP é considerado a estrutura genérica que será populada com o recurso do processo de organização, na qual ele se ajustará.

Conclusão

O que tem sido descrito sobre o mapeamento do RUP e do CMM, é igualmente aplicável na ISO 15504. Os conceitos de base dessas duas referências são semelhantes e os fatores chave foram destacados, para explicar o suporte de perfil de maturidade da escala de capacidades para a ISO 15504.

Esse documento tem esclarecido a extensão na qual os conceitos do RUP correspondem às metas de uma organização altamente desenvolvida. Portanto, a adoção do RUP é uma forma de preencher com êxito os requisitos do CMM.

Um artigo recente publicado pela Telcordia Technologies, fornece alguns resultados quantitativos de alta maturidade nos negócios. A Telcordia, recentemente avaliada no Nível 5 de maturidade, relata que:

- A redução na densidade com falha do campo atingiu 94%, desde o início de sua iniciativa SPI.
- Mais de 98% de releases do software foi entregue a tempo desde 1995.
- O custo de teste de uma linha de código foi reduzido a 64%.

A jornada de qualidade da Telcordia levou seis anos para alcançar esse nível de maturidade.

Para empresas intensivas de software nos campos de e-business e de e-commerce, não é possível permitir um esforço de cinco ou seis anos no tempo de entrega, para otimizar os processos. Portanto, o único cenário possível para alcançar a maturidade necessária a fim de permanecer competitivo é:

- Entender o valor do processo de software
- Adotar uma estrutura existente, como por exemplo o RUP
- Adaptar essa estrutura às características do ambiente de mercado
- Medir seu impacto nos negócios
- Aprender e melhorar o modelo de processo customizado

Como o modelo do RUP é genérico, é importante não se perder. Conseqüentemente, o Rational identifica o “RUP Essencial”, que inclui esses aspectos a serem focalizados na adoção do modelo [11, 12]. O RUP padrão pode ser utilizado, mas nem tudo precisa ser implementado na primeira vez. Os recursos importantes são:

- Caso de Negócios
- Planejamento
- Documento de Visão
- Riscos
- Arquitetura
- Controles de Mudanças e Como Lidar com Defeitos
- Testes
- O Próprio Produto de Software
- Documentos de Suporte do Usuário
- Avaliações do Projeto

A própria execução do RUP não é uma meta; ela ajuda a resolver pontos difíceis em grupos e em equipes, e ajuda a atender os objetivos competitivos.

Informações de Contato dos Autores

Annie Kuntzmann-Combelles, Executive VP
Q-Labs France
28 Villa Baudran 94742 Arcueil cedex
tel +33 (0)1 49 08 58 00
akc@objectif.fr

Philippe Kruchten, Rational Fellow
Rational Software Canada
pbk@rational.com

Referências

- [1] K. Pulford, A. Kuntzmann-Combelles, and S. Shirlaw, 1995. *A Quantitative Approach to Software Management—The AMI Handbook*. Addison Wesley Longman.
- [2] Barry W. Boehm, 1996, “Anchoring the Software Process,” *IEEE Software*, July 1996, pp.73-82.
- [3] Philippe Kruchten, 1996. “A Rational Development Process,” *CrossTalk*, 9 (7), July 1996, p.11-16.
- [4] Robert McFeeley, 1996. *IDEAL: A User's Guide for Software Process Improvement*. Software Engineering Institute, Pittsburgh, PA, CMU/SEI-96-HB-001.
- [5] Steve McConnell, 1997. *Software Project Survival Guide*. Redmond, WA: Microsoft Press.
- [6] Mark Paulk, et al. 1993. *Capability Maturity Model for Software, Version 1.1*. Software Engineering Institute, Pittsburgh, PA SEI-93-TR-024.
- [7] Walker Royce, 1998. *Software Project Management: A Unified Framework*. Addison Wesley Longman.
- [8] Jas Madhur et al, 1998. *Reaching CMM Levels 2 and 3 with the Rational Unified Process* white paper, Rational Software.
- [9] Alec Dorling et al., 1999. *SPICE, the Theory & Practice of Software Process Improvement*, IEEE Computer Society.
- [10] Philippe Kruchten, 2000. *The Rational Unified Process—An Introduction, 2nd ed.*, Addison Wesley Longman.
- [11] *Rational Unified Process, version 2000.02.10*, Rational Software Corporation
- [12] Leslee Probasco, 2000. “Os Dez Itens Essenciais do RUP”, *The Rational Edge*, December 2000, <http://www.therationaledge.com>.

Field Code Changed

Rational[®]

the software development company

Sedes Corporativas:
Rational Software
18880 Homestead Road
Cupertino, CA 95014
Tel: (408) 863-9900

Sem custo: (800) 728-1212
E-mail: info@rational.com
Web: www.rational.com
Locais Internacionais: www.rational.com/worldwide

Rational, o logotipo Rational, Rational Unified Process e Rational Rose são marcas registradas da Rational Software Corporation nos Estados Unidos e/ou outros países. Microsoft, Microsoft Windows, Microsoft Visual Studio, Microsoft Word, Microsoft Project, Visual C++ e Visual Basic são marcas ou marcas registradas da Microsoft Corporation. Todos os outros nomes são usados apenas para fins de identificação e são marcas ou marcas registradas de suas respectivas empresas. **TODOS OS DIREITOS RESERVADOS.** Feito nos EUA.

© Copyright 2002 Rational Software Corporation.
Sujeito à mudanças sem aviso prévio.