

Atingindo os Níveis 2 e 3 do CMM com o Rational Unified Process

Rational Software White Paper

TP 174

Rational[®]
the software development company

Índice Analítico

Resumo.....	1
Introdução.....	1
Nível 2, Repetível	3
Gerenciamento de Requisitos	3
Planejamento de Projeto de Software	4
Rastreo e Supervisão do Projeto de Software	5
Gerenciamento de Subcontrato de Software	6
Garantia de Qualidade de Software	6
Gerenciamento de Configuração de Software	7
Nível 3, Definido	8
Foco no Processo da Organização	8
Definição do Processo da Organização.....	9
Programa de Treinamento	9
Gerenciamento de Software Integrado.....	10
Engenharia do Produto de Software	10
Coordenação Entre Grupos.....	10
Revisões de Pontos.....	11
Referências.....	11

Resumo

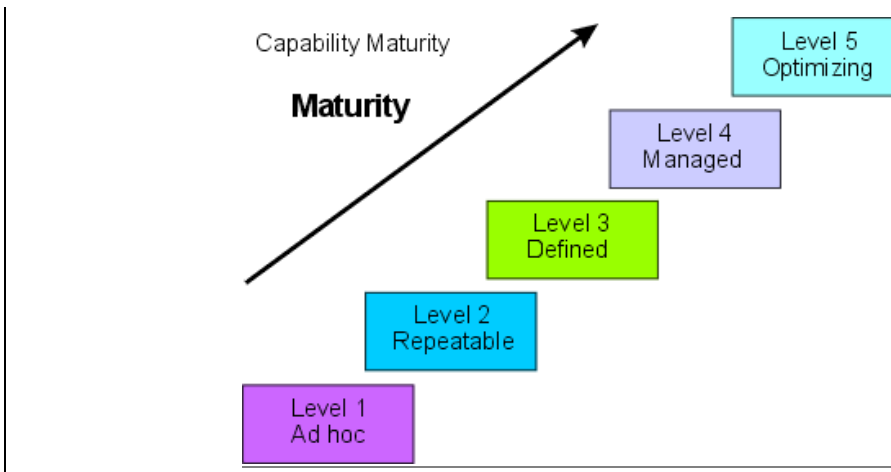
O CMM (Capability Maturity Model) do SEI (Software Engineering Institute) fornece um padrão de desempenho bem conhecido de maturidade de processo de software. O CMM tornou-se um veículo popular para avaliar a maturidade de um processo de software de uma organização em vários domínios. Este documento descreve como o Rational Unified Process pode suportar uma organização que está tentando atingir os níveis de maturidade do processo de software CMM Nível 2, Repetível, e Nível 3, Definido.

Introdução

O CMM (Capability Maturity Model) do SEI (Software Engineering Institute) é uma estrutura que descreve os elementos de um processo de software efetivo; consulte a seção intitulada *Referências*, [REF 1], no final deste documento. O CMM descreve um caminho de aprimoramento evolutivo de um processo especial, imaturo para um processo disciplinado maduro.

O CMM abrange práticas de planejamento, engenharia e gerenciamento de desenvolvimento e manutenção de software. Essas práticas-chave aprimoram a habilidade das organizações em atender aos objetivos de custo, planejamento, funcionalidade e qualidade do produto.

O CMM tem cinco níveis de maturidade: Nível 1 a Nível 5. Conforme ilustrado na seguinte figura, cada nível de maturidade é composto de KPAs (Áreas-chave do Processo) e cada KPA identifica um cluster de atividades relacionadas. Quando executadas coletivamente, essas atividades relacionadas obtêm um conjunto de objetivos considerados importantes para estabelecer a capacidade de processo nesse nível de maturidade.



No Nível 2, Nível Repetível, são estabelecidas as políticas para gerenciar um projeto de software e os procedimentos para implementá-las. O planejamento e o gerenciamento de novos projetos baseia-se na experiência com projetos similares. Um objetivo em atingir o Nível 2 é institucionalizar processos de gerenciamento efetivos para projetos de software, o que permite que as organizações repitam práticas bem-sucedidas desenvolvidas em projetos anteriores, embora os processos específicos implementados pelos projetos possam diferir. Um processo efetivo pode ser caracterizado como praticado, documentado, reforçado, treinado, medido e apto para aprimoramento.

Os projetos nas organizações de Nível 2 instalaram controles de gerenciamento de software básicos. Os compromissos realísticos de projetos baseiam-se nos resultados observados nos projetos anteriores e nos requisitos do projeto atual. Os gerentes de software de um projeto rastreiam custos de software, planejamentos e funcionalidade; os problemas em atender aos compromissos são identificados quando eles surgem. Os requisitos de software e os produtos de trabalho desenvolvidos para satisfazê-los servem como base e sua integridade é controlada. Os padrões do projeto de software são definidos e a

organização assegura que eles são fielmente seguidos. O projeto de software trabalha com suas subcontratadas, se houver alguma, para estabelecer um relacionamento forte com o fornecedor do cliente.

A capacidade do processo de software em organizações de Nível 2 pode ser resumida como disciplinada porque o planejamento e o rastreamento do projeto de software é estável e o sucesso anterior pode ser repetido. O processo do projeto está sob controle efetivo de um sistema de gerenciamento de projeto, seguindo planos realistas baseados no desempenho de projetos anteriores.

As KPAs de Nível 2 são:

- Gerenciamento de Requisitos
- Planejamento de Projeto de Software
- Rastreo e Supervisão do Projeto de Software
- Gerenciamento de Subcontrato de Software
- Garantia de Qualidade de Software
- Gerenciamento de Configuração de Software

No Nível 3, o Nível Definido, o processo padrão para desenvolver e manter o software através da organização é documentado, incluindo processos de engenharia e gerenciamento e esses processos são integrados em um todo coerente. O processo padrão é referido através do CMM como o processo de software padrão da organização. Os processos estabelecidos no Nível 3 são utilizados (e alterados, conforme apropriado) para ajudar os gerentes de software e a equipe técnica a executar mais efetivamente. A organização explora práticas de engenharia de software efetivas ao padronizar seus processos de software. Há um grupo que é responsável pelas atividades de processo de software da organização; por exemplo, engenharia de software ou SEPG. Um programa de treinamento em toda a organização é implementando para assegurar que a equipe e os gerentes tenham o conhecimento e as habilidades requeridas para preencher suas funções designadas.

Os projetos ajustam o processo de software padrão da organização para desenvolver seu próprio processo de software definido, que representa as únicas características do projeto. Esse processo adaptado é referido no CMM como o processo de software definido do projeto. Um processo de software definido contém um conjunto coerente e integrado de processos de engenharia e de gerenciamento de software bem definidos. Um processo bem definido pode ser caracterizado como incluindo critérios de prontidão, entradas, padrões e procedimentos para executar o trabalho; mecanismos de verificação, como revisões de pontos; e critérios de conclusão. Como o processo de software é bem definido, o gerenciamento apresenta uma boa percepção no progresso técnico de todos os projetos.

A capacidade do processo de software das organizações de Nível 3 pode ser resumida como padrão e consistente porque as atividades de engenharia e gerenciamento de software são estáveis e repetíveis. Dentro das linhas de produto estabelecidas, o custo, o planejamento e a funcionalidade estão sob controle e a qualidade do software é rastreada. 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.

As KPAs de Nível 3 são:

- Foco no Processo da Organização
- Definição do Processo da Organização
- Programa de Treinamento
- Gerenciamento de Software Integrado
- Engenharia do Produto de Software
- Coordenação Entre Grupos
- Revisões de Pontos

Cada seção deste documento descreve como os recursos, os métodos, os procedimentos e os artefatos do Rational Unified Process atendem aos objetivos da KPA.

Este documento é escrito para o pessoal de uma organização preocupado em atingir o Nível 2 e Nível 3 de maturidade organizacional dentro da estrutura do CMM.

Nível 2, Repetível

Gerenciamento de Requisitos

O objetivo do Gerenciamento de Requisitos é estabelecer um entendimento comum entre o cliente e o projeto de software dos requisitos do cliente que serão enfocados pelo projeto de software. Essa concordância com o cliente é a base do planejamento (conforme descrito em KPA de Planejamento do Projeto de Software) e do gerenciamento (conforme descrito em Rastreo e Supervisão do Projeto de Software) do projeto de software. O controle do relacionamento com o cliente depende de seguir um processo efetivo de controle de alterações, conforme descrito em Gerenciamento de Configuração de Software.

Um dos recursos-chave do Rational Unified Process é que ele é conduzido por *caso de uso*. Casos de uso representam uma abordagem sistemática de fazer levantamento, organizar e comunicar requisitos do usuário. Eles fornecem uma forma de documentar requisitos funcionais que servem como uma base para o desenvolvimento do projeto, teste e planejamento de iteração. No Rational Unified Process, os casos de uso são mantidos em um modelo de caso de uso e referidos consistentemente através do ciclo de vida do projeto, desde a análise até o teste de manutenção.

Os artefatos do Rational Unified Process que capturam requisitos no contexto de engenharia são:

- Modelo de Caso de Uso que consiste em Casos de Uso e Pacotes de Caso de Uso
- "Especificações Suplementares" não funcionais
- Relatório Sintético do Modelo de Caso de Uso
- Relatório do Caso de Uso
- Glossário

Os artefatos do Rational Unified Process que descrevem casos de uso e cenários (requisitos) que precisam ser desenvolvidos, conforme utilizados no contexto de gerenciamento, são:

- Plano de Iteração
- Plano de Integração do Build
- Plano de Projeto
- Plano de Desenvolvimento de Software

Todos esses artefatos utilizam linha de base e estão sujeitos a uma disciplina de gerenciamento de alterações.

Objetivo 1: Os requisitos do sistema que são alocados para o software são controlados para estabelecer uma linha de base para uso pela engenharia e gerenciamento de software.

O Rational Unified Process defende o controle de configuração contínuo de todos os artefatos em desenvolvimento, no entanto, as linhas de base 'formais' correspondem aos seguintes marcos.

- Marco dos Objetivos do Ciclo de Vida (Fase de Iniciação)
- Marco da Arquitetura do Ciclo de Vida (Fase de Elaboração)
- Marco da Capacidade Operacional Inicial (Fase de Construção)
- Marco de Liberação do Produto (Fase de Transição)

Como tal, o Rational Unified Process é compatível com o CMM para concordância em requisitos, seu gerenciamento e uso de linha de base.

Objetivo 2: Os planos de software, os produtos e as atividades são mantidos consistentes com os requisitos do sistema alocados para o software.

A ênfase desse objetivo do CMM é assegurar que os sistemas fornecidos atendam aos requisitos dos usuários. O Rational Unified Process ajuda as organizações a alcançarem esse objetivo de duas formas:

- A *Abordagem de Caso de Uso* assegura que os requisitos do usuário são entendidos e capturados. Depois de capturados, os requisitos fluem para os vários modelos "visuais" do Rational Unified Process (Caso de Uso, Design, Implementação e Teste) para assegurar consistência e aderência.
- A abordagem *Desenvolvimento Iterativo Controlado* é uma abordagem de mitigação de risco na qual os riscos do projeto são entendidos e explorados antecipadamente e, em seguida, revisitados com frequência. Cada iteração progressiva, através da integração contínua de funcionalidade adicionada, descobre os riscos antecipadamente. Utilizando metodologias em cascata tradicionais, esses riscos podem ter sido mantidos descobertos até um ponto avançado no ciclo de vida de desenvolvimento. A identificação antecipada de riscos representa um benefício direto ao gerenciamento do projeto através de novo escopo de requisitos ou da solicitação de outras alterações táticas.

Os documentos de gerenciamento do Rational Unified Process são:

- Caso de Negócios
- Plano de Desenvolvimento de Software
- Plano de Medidas
- Lista de Riscos
- Plano de Projeto
- Plano(s) de Iteração
- Avaliação(ões) de Iteração e de Status

O Controle e Gerenciamento Efetivo de Alterações é outro recurso do Rational Unified Process que assegura que o software seja desenvolvido para requisitos especificados, rastreados e alocados.

O Rational Unified Process defende que cada projeto estabeleça um CCB (Conselho de Controle de Alterações) que pode arbitrar no escopo e no impacto (orçamentário, técnico ou planejamento) das alterações propostas ou defeitos descobertos durante o curso do desenvolvimento. Para ajudar na operação do CCB, o Rational Unified Process recomenda o uso de um forte gerenciamento de configuração e uma ferramenta/ambiente de controle de versão.

Planejamento de Projeto de Software

O objetivo do Planejamento de Projeto de Software é estabelecer planos razoáveis para executar a engenharia de software e gerenciar o projeto de software. Esses planos são necessários para gerenciar o projeto de software, conforme descrito na seção intitulada *Rastreio e Supervisão do Projeto de Software*. Sem planos realísticos, o gerenciamento efetivo do projeto não pode ser implementado.

Objetivo 1: As estimativas de software são documentadas para uso no planejamento e rastreio do projeto de software.

Um dos objetivos do Rational Unified Process é assegurar que as expectativas de todas as partes estejam sincronizadas e consistentes. Isso é assegurado através de avaliações periódicas em todo o ciclo de vida do projeto e é documentado no *Relatório de Avaliação de Status*. O relatório busca o rastreio de dados em recursos (de equipe e financeiro), os dez principais riscos, o progresso técnico conforme medido através de métricas e os resultados de marcos principais.

O Rational Unified Process utiliza as seguintes classes de métricas:

- Progresso (linhas de código, números de classes, pontos de função por iteração e retrabalho)
- Estabilidade (tipo de retrabalho requisitos ou volatilidade de implementação)
- Adaptabilidade (custo de retrabalho)
- Modularidade (extensão do impacto de retrabalho)
- Qualidade (taxa de descoberta de defeitos, profundidade de herança e indicador de retrabalho)
- Maturidade (horas de teste por falha)
- Perfis de consumo em recursos (planejados versus reais)

Objetivo 2: As atividades e os compromissos do projeto de software são planejados e documentados.

Os documentos do Rational Unified Process que capturam os planos e compromissos do projeto são:

- Caso de Negócios
- Plano de Desenvolvimento de Software
- Plano de Medidas
- Lista de Riscos
- Plano de Projeto
- Plano(s) de Iteração
- Avaliação(ções) de Iteração
- Avaliação(ções) de Status

Objetivo 3: Os grupos e os indivíduos afetados concordam com seus compromissos relacionados ao projeto de software.

No Rational Unified Process, o Plano de Desenvolvimento de Software define o plano global para o projeto; o Plano de Iteração define em detalhes que deve ser realizado em uma iteração. A Revisão do Plano de Iteração, requerida pelo Rational Unified Process, expõe o Plano de Iteração para todos os envolvidos, permitindo que um consenso seja desenvolvido antes da iteração começar. A partir do Plano de Iteração aceito, o Coordenador de Projeto produz uma série de ordens de trabalho, que comunicam a intenção do Plano de Iteração em detalhes para as equipes e indivíduos do projeto afetado. O Coordenador de Projeto obtém a aceitação dessas ordens de trabalho com a equipe afetada para que a iteração possa continuar.

Rastreo e Supervisão do Projeto de Software

O objetivo do Rastreo e Supervisão do Projeto de Software é estabelecer visibilidade adequada para o progresso real a fim de que o gerenciamento possa executar ações efetivas quando o desempenho do projeto de software desvia-se significativamente dos planos de software.

Objetivo 1: Os resultados reais e o desempenho são rastreados nos planos de software.

Conforme descrito na seção *Planejamento do Projeto de Software*, o Rational Unified Process tem vários níveis de planos de projeto e um *Relatório de Avaliação de Status* que é gerado para avaliar desempenho planejado versus desempenho real. Esse relatório, gerado para marcos específicos, é de responsabilidade do Coordenador de Projeto.

Os marcos principais do Rational Unified Process correspondem ao final de uma fase (Iniciação, Elaboração, Construção ou Transição) e têm critérios de conclusão bem especificados. Existem oportunidades de revisão em marcos secundários no final de cada iteração dentro de uma fase e servem como pontos de decisão e lições aprendidas para orientação futura.

Por exemplo: Os objetivos da fase de elaboração são para analisar o domínio de problema, estabelecer uma base arquitetural segura, desenvolver o plano de projeto e eliminar os elementos de maior risco do projeto. As decisões arquiteturais devem ser feitas com um entendimento do sistema inteiro. Isso implica que a maioria dos casos de uso serão descritos, levando em conta algumas restrições: requisitos suplementares. Para verificar a arquitetura, é implementado um sistema que demonstra as opções arquiteturais e executa casos de uso significativos.

No final da fase de elaboração, objetivos detalhados do sistema e o escopo são examinados, bem como a opção de uma arquitetura e a resolução de riscos principais. As ações corretivas são executadas e gerenciadas para fechamento quando os resultados e desempenho reais se desviam significativamente dos planos de software.

A *Lista de Riscos* é um artefato do Rational Unified Process que fornece uma visão geral de todos os riscos conhecidos no projeto e serve como entrada para avaliações de planejamento e projeto. Cada risco é descrito em termos de seu impacto e o plano de contingência que deve ser chamado para mitigar o risco em questão. A Lista de Riscos é desenvolvida ao longo do Caso de Negócios para formar a base para uma decisão "ir" ou "não ir" no projeto. A Lista de Riscos é mantida em todo o ciclo de vida do projeto.

Objetivo 2: As ações corretivas são executadas e gerenciadas para fechamento quando os resultados e o desempenho reais se desviam significativamente dos planos de software.

O Rational Unified Process requer o rastreamento do desempenho em dois níveis: como parte do monitoramento e controle de rotina do projeto, o Coordenador de Projeto utilizará as métricas coletadas para produzir uma Avaliação de Status. Os problemas identificados na Avaliação de Status são tratados de acordo com o Plano de Resolução de Problemas para o projeto, diretamente pelo Coordenador de Projeto (através de Ordens de Trabalho) ou, para problemas de escopo mais amplo, através de Controles de Alterações. Além disso, os resultados de cada iteração estão sujeitos a uma Avaliação e Revisão de Iteração. Isso permite que a experiência de iterações anteriores guie o planejamento de iterações subsequentes, gerenciando ações corretivas através dos Controles de Alterações.

Objetivo 3: As alterações nos compromissos do software são aceitas pelo grupo e pelos indivíduos afetados.

O processo de desenvolvimento iterativo controlado, conforme descrito no Rational Unified Process, assegura que os envolvidos obtenham visibilidade regular do progresso do projeto e de todas as alterações que possam ser necessárias para acompanhar o curso do projeto. As alterações propostas são revisadas por um CCB (Conselho de Controle de Alterações) para assegurar que elas sejam realísticas e possam ser acomodadas no planejamento de projeto global.

Gerenciamento de Subcontrato de Software

O objetivo do Gerenciamento de Subcontrato é selecionar subcontratadas de software qualificadas e gerenciá-las efetivamente. Ele combina as preocupações de Gerenciamento de Requisitos, Planejamento de Projeto de Software e Rastreamento e Supervisão de Projeto de Software para controle básico de gerenciamento, juntamente com coordenação necessária de Garantia de Qualidade de Software e Gerenciamento de Configuração de Software e aplica esse controle à subcontratada conforme adequado.

Objetivo 1: A contratada principal seleciona as subcontratadas de software qualificadas.

Objetivo 2: A contratada principal e a subcontratada de software concordam com os compromissos estabelecidos entre elas.

Objetivo 3: A contratada principal e a subcontratada de software mantêm comunicações contínuas.

Objetivo 4: A contratada principal rastreia os resultados e desempenho reais da subcontratada de software em seus compromissos.

Esses objetivos vão além do escopo atual do Rational Unified Process e dependem da organização.

Enquanto a subcontratação não é especificamente enfocada no Rational Unified Process, assume-se que suas ferramentas, técnicas e mecanismos serão levados para as subcontratadas para que o processo permaneça homogêneo.

Todas as decisões de subcontratação devem ser documentadas no Caso de Negócios. As subcontratadas que seguem o mesmo plano de desenvolvimento da contratada principal também participarão dos mesmos intercâmbios de técnicas, marcos principais e avaliações de status.

Garantia de Qualidade de Software

O objetivo da Garantia de Qualidade de Software é fornecer gerenciamento com visibilidade adequada para o processo que está sendo utilizado pelo projeto de software e os produtos que estão sendo construídos. A Garantia de Qualidade de Software é uma parte integral da maioria dos processos de engenharia e gerenciamento de software.

O Rational Unified Process considera a qualidade como responsabilidade coletiva de toda a equipe do projeto e não expressa em qualquer organização especificada por si própria.

Objetivo 1: As atividades de garantia de qualidade de software são planejadas.

O planejamento das tarefas de garantia de qualidade de software é de responsabilidade organizacional. No entanto, o Rational Unified Process tem uma série de atributos que nos leva a preparar um programa efetivo de garantia de qualidade de projeto.

Cada marco do Rational Unified Process tem critérios de conclusão específicos que podem servir como uma base para auditoria. Dentro do Rational Unified Process existem tarefas de revisão separadas. Associado a cada revisão está um

conjunto de pontos de verificação que representa “portas” que precisam ser “passadas” antes de se entrar na atividade seguinte.

O Rational Unified Process fornece orientação sobre quem deve revisar os artefatos especificados. Por exemplo, os resultados da “Análise do Caso de Uso”, conforme executada por um Designer, precisam ser revisados por um Arquiteto, Designer, Designer de Caso de Uso e Revisor de Designer independentes. Dado o Rational Unified Process definido e os critérios de revisão de artefato, um corpo objetivo preocupado com a qualidade do produto deve facilmente poder avaliar a aderência ao processo e a conformidade com os padrões e orientações de desenvolvimento.

Objetivo 2: A aderência dos produtos de software a das atividades aos padrões, procedimentos e requisitos aplicáveis é verificada objetivamente.

Esse objetivo deve ser atendido pela adoção do pessoal de qualidade da organização. No entanto, o Rational Unified Process fornece as listas de verificação de revisão e os gabaritos de documentos necessários que podem ser aplicados como padrões de projeto.

Objetivo 3: Os grupos e os indivíduos afetados são informados sobre as atividades de garantia de qualidade de software e os resultados.

Conforme descrito em *Planejamento de Projeto de Software*, um dos objetivos do Rational Unified Process é assegurar que as expectativas de todas as partes sejam sincronizadas e estejam consistentes. À parte de qualquer entrada dos resultados de auditoria de qualidade, o Rational Unified Process busca relatórios sobre recursos (de equipe e financeiro), os dez principais riscos, o progresso técnico conforme medido através de métricas e os resultados de marcos principais. O programa de métricas do Rational Unified Process fornece diretrizes sobre a coleta das seguintes métricas:

- Progresso (Linhas de código, classes, pontos de função por iteração)
- Estabilidade (tipo de retrabalho, volatilidade)
- Adaptabilidade (custo de retrabalho)
- Modularidade (extensão do impacto de retrabalho)
- Qualidade (taxa de descoberta de defeitos, densidade, profundidade de herança)
- Maturidade (horas de teste por falha)
- Perfis de consumo (planejados versus reais)

Objetivo 4: Os problemas de não-conformidade que não podem ser resolvidos dentro do projeto de software são enfocados pelo gerenciamento sênior.

Isso está além do escopo do Rational Unified Process e é de responsabilidade da organização. No entanto, o Processo de Controle de Alterações descrito no Rational Unified Process ativar um mecanismo pelo qual as não-conformidades poderão ser documentadas e escaladas para resolução.

Gerenciamento de Configuração de Software

O objetivo do Gerenciamento de Configuração de Software é estabelecer e manter a integridade dos produtos do projeto de software em todo o ciclo de vida do software do projeto. O Gerenciamento de Configuração de Software é uma parte integral da maioria dos processos de engenharia e gerenciamento de software.

Objetivo 1: As atividades de gerenciamento de configuração de software são planejadas.

Conforme descrito no Rational Unified Process, o forte gerenciamento de configuração é um elemento essencial no método de desenvolvimento iterativo controlado. Como o software é desenvolvido em estágios, é vital que as versões de software do esforço de desenvolvimento precedente estejam disponíveis para desenvolvimento subsequente. Planejar como o software de trabalho demonstrável deve ser produzido em cada estágio está no núcleo do Rational Unified Process.

O Rational Unified Process tem dois instrumentos principais para definir como os recursos de desenvolvimento de software de um projeto são mantidos e como eles devem ser integrados:

- o Plano de Gerenciamento de Configuração;

- o Plano de Integração da Construção.

O Plano de Gerenciamento de Configuração, iniciado na Fase de Iniciação descreve o seguinte:

- gerenciamento do controle de versões e manipulação do software;
- salvamento dos modelos fornecidos do Rational Unified Process e divisão deles em itens de configuração;
- gerenciamento de alterações e releases utilizando o método de controle de alterações.

O Plano de Integração da Construção fornece os detalhes sobre itens de configuração que devem ser construídos e a ordem na qual eles devem ser integrados em uma determinada iteração.

Objetivo 2: Os produtos de trabalho de software selecionados são identificados, controlados e estão disponíveis.

O Plano de Gerenciamento de Configuração do Rational Unified Process busca uma descrição do processo de controle e gerenciamento de configuração para assegurar que os produtos de trabalho estejam realmente identificados, controlados e disponíveis.

Objetivo 3: As alterações nos produtos de trabalho de software identificados são controladas.

O Rational Unified Process defende que um projeto mantenha um CCB (Conselho de Controle de Alterações) e tenha um Sistema de Gerenciamento de Alterações para gerenciar, obter custos, rastrear e implementar adequadamente os controles de alterações.

Objetivo 4: Os grupos e os indivíduos afetados são informados sobre o status e o conteúdo das linhas de base de software.

O Rational Unified Process defende que os requisitos, o design e as linhas de base de implementação e a rastreabilidade entre eles sejam mantidos em formato eletrônico. Vários níveis do controle de projeto arbitram quaisquer alterações nas linhas de base. Por exemplo, o CCB (Conselho de Controle de Alterações) considera o impacto de alterações no nível do requisito. As alterações de design e implementação que são menores em escopo são revisadas no nível adequado da autoridade técnica. A aprovação, os níveis de controle e a forma como eles são comunicados estão descritos no *Plano de Gerenciamento de Configuração* e no *Plano de Desenvolvimento de Software*.

Nível 3, Definido

Foco no Processo da Organização

O objetivo do Foco no Processo da Organização é estabelecer a responsabilidade organizacional para atividades de processo de software que aprimorem a capacidade global do processo de software da organização. O resultado principal das atividades do Foco no Processo da Organização é um conjunto de recursos de processo de software, que estão descritos em Definição do Processo da Organização. Esses recursos são utilizados pelos projetos de software, conforme descrito em Gerenciamento de Software Integrado.

Objetivo 1: As atividades de desenvolvimento e aprimoramento do processo de software são coordenadas através da organização.

O Rational Unified Process é um processo iterativo que confia na re-representação do mesmo processo definido através de uma série de iterações. Essa natureza repetitiva de representação do processo e a avaliação das métricas de status e lições aprendidas em cada fase e iteração fornecem uma oportunidade para ajustar o processo em cada iteração sucessiva.

Objetivo 2: Os pontos fortes e fracos do processo de software são identificados com relação a um padrão de processo.

O Rational Unified Process representa um processo de desenvolvimento de software global que pode ser ajustado para uso efetivo em qualquer tipo de projeto determinado. A orientação sobre como ajustar o Rational Unified Process é fornecida na *Disciplina de Ambiente*. Algumas distinções do processo, que não a complexidade técnica e de gerenciamento, que determinarão a “forma” do processo a ser utilizado em um projeto são:

- contexto de negócios (contrato, especulativo ou interno);

- tamanho do esforço de desenvolvimento de software;
- grau de inovação;
- tipo de aplicativo.

Objetivo 3: As atividades de desenvolvimento e aprimoramento do processo no nível da organização são planejadas.

Esse objetivo de Nível 3 depende inteiramente da organização de adoção.

Definição do Processo da Organização

O objetivo da Definição do Processo da Organização é desenvolver e manter um conjunto utilizável de recursos de processo de software que aprimorem o desempenho do processo através dos projetos e forneça a base para benefícios acumulativos, de longo prazo, para a organização. Esses recursos fornecem uma base estável que pode ser institucionalizada através de mecanismos, como treinamento, descritos no Programa de Treinamento.

Objetivo 1: Um processo de software padrão para a organização é desenvolvido e mantido.

O Rational Unified Process pode fornecer um início decisivo e servir como um processo de linha de base de desenvolvimento de software da organização, que pode ser desenvolvido, ajustado e mantido.

Objetivo 2: As informações relacionadas ao uso do processo de software padrão da organização pelos projetos de software são coletadas, revisadas e disponibilizadas.

O objetivo precisaria ser suportado apenas pela organização que está adotando o Rational Unified Process.

Programa de Treinamento

O objetivo do Programa de Treinamento é desenvolver as habilidades e o conhecimento de indivíduos para que possam executar suas funções de forma efetiva e eficiente. O treinamento é uma responsabilidade organizacional, mas os projetos de software devem identificar suas habilidades necessárias e fornecer o treinamento necessário quando as necessidades do projeto são exclusivas.

Objetivo 1: As atividades de treinamento são planejadas.

Esse objetivo pode ser atendido apenas pela organização que está adotando o Rational Unified Process. No entanto, o Rational Unified Process é uma base de conhecimento de “boas práticas no segmento de mercado” que fornece diretrizes, conceitos e descrições passo a passo detalhadas sobre como executar várias atividades de desenvolvimento de software. Como tal, o Rational Unified Process é por si próprio uma boa fonte de material de treinamento.

Os cursos de suporte relacionados ao Rational Unified Process incluem:

- O Rational Unified Process Overview com módulos sobre Requisitos, Análise & Design, Implementação, Teste, Arquitetura, Configuração de Processo, Gerenciamento, Ferramentas e uma Introdução à Orientação de Objetos
- Requirements Management with Use Cases (RMUC)
- Object-Oriented Project Management (OOPM)
- Object-Oriented Design and Analysis (OOAD)
- Automating Software Quality
- Gerenciamento de Configuração
- Software Architecture and Iterative Process

Objetivo 2: O treinamento é fornecido para desenvolver as habilidades e o conhecimento necessários para executar funções de gerenciamento de software e técnicas.

Objetivo 3: Os indivíduos no grupo de engenharia de software e os grupos relacionados ao software recebem o treinamento necessário para executar suas funções.

Esses objetivos do programa de treinamento precisarão ser atendidos pela organização que está adotando o Rational Unified Process. No entanto, o Rational Unified Process fornece uma faixa de cursos, conforme descrito na seção anterior.

Gerenciamento de Software Integrado

O objetivo do Gerenciamento de Software Integrado é integrar as atividades de engenharia e gerenciamento de software em um processo de software definido, coerente, que é ajustado a partir do processo de software padrão da organização e de recursos do processo relacionados, que estão descritos na seção intitulada *Definição do Processo da Organização*. Esse ajuste baseia-se nas necessidades do ambiente de negócios e técnicas do projeto, conforme descrito em Engenharia do Produto de Software. O Gerenciamento de Software Integrado é desenvolvido a partir do Planejamento do Projeto de Software e do Rastreamento e Supervisão do Projeto de Software no Nível 2.

Objetivo 1: O processo de software definido do projeto é uma versão ajustada do processo de software padrão da organização.

De acordo com a Disciplina de Ambiente do Rational Unified Process, a entrega padrão do Rational Unified Process é configurável e pode ter um novo escopo para uso em vários tipos de projetos.

Objetivo 2: O projeto é planejado e gerenciado de acordo com o processo de software definido do projeto.

Esse objetivo precisará ser enfocado pela organização que está adotando o Rational Unified Process.

Engenharia do Produto de Software

O objetivo da Engenharia do Produto de Software é executar consistentemente um processo de engenharia bem definido que integre todas as atividades de engenharia de software para produzir produtos de software corretos e consistentes de forma efetiva e eficiente. A Engenharia do Produto de Software descreve as atividades técnicas do projeto; por exemplo, análise de requisitos, design, código e teste.

Objetivo 1: As tarefas de engenharia de software são definidas, integradas e executadas consistentemente para produzir o software.

As tarefas e definições do Rational Unified Process sobre o que é requerido em cada função, em uma cortina de fundo de artefatos requeridos para planejamento do projeto, asseguram que as tarefas sejam definidas, alocadas e concluídas. O processo de desenvolvimento iterativo inerente no Rational Unified Process serve para provar rapidamente a efetividade da equipe de desenvolvimento de software e fornece uma avaliação do produto final.

Objetivo 2: Os produtos de software são mantidos consistentes uns com os outros.

A rastreabilidade é mantida entre os modelos de engenharia, como modelos de caso de uso, modelos de design, código fonte e componentes executáveis pelo ambiente.

Coordenação Entre Grupos

O objetivo da Coordenação Entre Grupos é estabelecer um meio para o grupo de engenharia de software participar ativamente com outros grupos de engenharia para que o projeto satisfaça ainda mais as necessidades do cliente de forma eficiente e efetiva. A Coordenação Entre Grupos é o aspecto interdisciplinar do Gerenciamento de Software Integrado que se estende além da engenharia de software; o processo de software não deve apenas ser integrado, mas as interações do grupo de engenharia de software com outros grupos devem ser coordenadas e controladas.

Objetivo 1: Os requisitos do cliente são aceitos por todos os grupos afetados.

Um benefício substancial da utilização de casos de uso como uma base para captura e descrição de requisitos sobre outros métodos de requisitos “formais” é que os casos de uso são prontamente entendidos pelos envolvidos. Como tal, o método do Rational Unified Process de capturar requisitos de caso de uso significa que todos os envolvidos podem concordar com que

precisa ser feito. Isso é mais executado através do processo e é refletido nos modelos e revisões que são utilizados como uma base para o desenvolvimento de software.

Objetivo 2: Os compromissos entre os grupos de engenharia são aceitos pelos grupos afetados.

Esse objetivo precisará ser focado pela organização que está adotando o Rational Unified Process. No entanto, os modelos visuais do Rational Unified Process facilitam o entendimento do que é requerido em cada estágio de desenvolvimento do produto—da captura de requisitos à implantação. O processo de Gerenciamento de Alterações e Configuração do Rational Unified Process assegura que as alterações propostas estão adequadamente avaliadas e comunicadas a todos os envolvidos.

Os grupos de engenharia identificam, rastreiam e resolvem problemas entre grupos. O processo de desenvolvimento iterativo do Rational Unified Process facilita a detecção precoce de problemas de software através de integração contínua de todo o software desenvolvido. Os problemas de integração com o software que é desenvolvido por uma série de equipes pode servir como um “espaço comum” para gerar e resolver problemas de equipes cruzadas. Essa noção é suportada pelo processo de controle de alterações e defeitos do Rational Unified Process, que fornece um mecanismo formal para capturar, rastrear e resolver problemas de desenvolvimento de projeto.

Revisões de Pontos

O objetivo das Revisões de Pontos é remover defeitos dos produtos de trabalho de software de forma precoce e eficiente. Um efeito de consequência importante é desenvolver um melhor entendimento dos produtos de trabalho de software e dos defeitos que podem ser evitados. A revisão de pontos é um método de engenharia importante e efetivo que é chamado na Engenharia de Produto de Software.

Objetivo 1: As atividades de revisão de pontos são planejadas.

Conforme descrito nos objetivos de Garantia de Qualidade para Nível 2, cada tarefa dentro do Rational Unified Process tem uma tarefa de revisão separada.

Como a detecção precoce de problemas reduz os custos globais, o Rational Unified Process defende revisões de pontos “precoce e frequentes” de todos os artefatos, particularmente os críticos. O Rational Unified Process fornece listas de verificação de recursos salientes para revisar em cada estágio e dentro de cada modelo.

Objetivo 2: Os defeitos nos produtos de trabalho de software são identificados e removidos.

Os revisores de artefatos do Rational Unified Process devem determinar se o artefato está pronto para o próximo estágio de desenvolvimento. Se o artefato não atender aos critérios de “aprovação” da revisão, então, de acordo com o programa de métricas do Rational Unified Process, precisarão ser capturados detalhes sobre:

- Estabilidade (tipo de retrabalho, volatilidade)
- Adaptabilidade (custo de retrabalho)
- Modularidade (extensão do impacto de retrabalho)
- Qualidade (taxa de descoberta de defeitos, densidade, profundidade de herança)
- Maturidade (horas de teste por falha)
- Perfis de consumo (planejados versus reais)

Referências

[REF1] Mark C. Paulk et al, "Key Practices of the Capability Maturity Model - Version 1.1", Software Engineering Institute—Carnegie Mellon University.



Duas Sedes:

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

Rational Software
20 Maguire Road
Lexington, MA 02421
Tel: (781) 676-2400

Sem custo: (800) 728-1212

E-mail: info@rational.com

Web: www.rational.com

Localização Internacional: www.rational.com/worldwide

Field Code Changed

Field Code Changed

Field Code Changed

Rational, o logotipo Rational e Rational Unified Process 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.