

Os Dez Itens Essenciais do RUP— A Essência de um Processo de Desenvolvimento Efetivo

Leslee Probasco

Rational Software White Paper

TP177, 9/00

Rational
the software development company

Índice Analítico

Resumo	1	Deleted: 3
Um Pedido de Ajuda	1	Deleted: 3
Impressionado com os Detalhes	1	Deleted: 3
Foco nos Itens Essenciais	1	Deleted: 3
Começar com uma “Lista Abreviada”	2	Deleted: 4
O Que Deveria Estar em sua Lista? “Bem, isso depende...”	2	Deleted: 4
Construir Primeiro uma Estrutura	3	Deleted: 4
Os Dez Itens Essenciais do RUP	3	Deleted: 5
Visão — Desenvolver uma Visão.....	4	Deleted: 6
Plano — Gerenciar no Plano.....	4	Deleted: 6
Riscos — Identificar e Diminuir Riscos.....	5	Deleted: 6
Problemas — Designar e Acompanhar os Problemas.....	5	Deleted: 7
Caso de Negócios — Examinar o Caso de Negócios.....	5	Deleted: 7
Arquitetura — Projetar uma Arquitetura de Componente.....	5	Deleted: 7
Produto — Construir e Testar de Forma Incremental o Produto.....	6	Deleted: 7
Avaliação — Avaliar os Resultados Regularmente.....	6	Deleted: 7
Controles de Mudanças — Gerenciar e Controlar Mudanças.....	6	Deleted: 7
Suporte ao Usuário — Fornecer Assistência ao Usuário.....	6	Deleted: 7
E sobre os Requisitos?	6	Deleted: 8
E sobre o Teste?	7	Deleted: 8
Resumo: Aplicando os Dez Itens Essenciais	7	Deleted: 8
Para Projetos Menores.....	7	Deleted: 9
Para Projetos em Crescimento.....	7	Deleted: 9
Para Equipes de Projeto Desenvolvidas.....	8	Deleted: 9
		Deleted: 9
		Deleted: 9
		Deleted: 9

Resumo

Para aplicar efetivamente um processo de desenvolvimento de software, como por exemplo o Rational Unified Process¹ (carinhosamente conhecido como “RUP”), é importante primeiro entender seus objetivos principais, o motivo pelo qual cada um é importante e como eles funcionam em conjunto, a fim de ajudar a equipe de desenvolvimento a produzir um produto de qualidade que atenda às necessidades “reais” dos investidores.

Um Pedido de Ajuda

Outra noite, meu vizinho Randy veio pedir ajuda: Ele estava se preparando para uma viagem de acampamento e de caminhada no final-de-semana com seu grupo de igreja e estava tentando decidir quais equipamentos¹ levar para a viagem. Ele já tinha alguma roupa e equipamento e, sem dúvida alguma, estava indo comprar mais algumas. Poderia emprestar minha lista de equipamentos?

Minha lista de equipamentos? Sim. Parece que ele estava me visitando no dia em que eu estava me preparando para uma viagem como mochileiro e alpinista. Ele sabe que tenho muita experiência em conduzir e participar de viagens para regiões naturais e estava muito impressionado com minha rapidez e eficiência na determinação dos itens que se adequam ao meu espaço limitado de bagagem, referente à uma lista existente da maioria de meus equipamentos e roupas restritos. Ele poderia emprestar essa lista?

Bem, claro, ele estava à vontade para emprestar minha lista de equipamentos — mas eu estava com receio de que ela não seria muito útil. Por que não? Eu não achava que tudo o que ele fosse precisar estaria nesta lista?

Na verdade, eu tinha quase certeza de que tudo o que fosse necessário estaria pelo menos *representado* por algo em minha lista, porém, mais que provável, os itens exatos necessários para esse passeio específico poderia não estar, de forma alguma, na lista. Por exemplo, não apenas o número do sapato dele é *bem* maior do que o meu, mas, sem dúvida, ele prefere comidas (e quantidades) diferentes do que eu poderia escolher por um longo período.

Impressionado com os Detalhes

O fato é que tenho literalmente *centenas* de itens em minha lista de equipamentos para regiões naturais, incluindo vários tipos de viagens como mochileiro e alpinista, a esqui, raquete de neve, escalada no gelo e caiaque — e incluindo viagens longas, desde viagens de apenas um dia a vários dias. E tenho itens diferentes que eu levaria se estivesse conduzindo uma viagem para uma organização (como por exemplo, formulário de inscrição para viagens e formulários de desistências), do que se eu estivesse apenas saindo para as regiões naturais com alguns de meus amigos. Minha principal preocupação era que ele não conseguiria passar pelo grande número de itens na lista e calcular o que *ele* realmente precisava apenas referente à sua viagem.

Randy levantou as mãos exasperado! Ele nunca estaria corretamente preparado no devido tempo para sua longa caminhada. Como saberia o que levar? Ele tinha começado a arrumar as malas na semana passada — ele fez uma pesquisa abrangente na Internet, comprou botas, jaquetas e roupas novas e modernas, — sua mochila já estava cheia!

E ainda não havia colocado nenhum alimento! Ah, sim, e era necessário levar água também? E se acontecesse algo de errado? E se eles ficassem perdidos? Ou alguém se machucasse?

Foco nos Itens Essenciais

Conforme percorri os itens que Randy já havia colocado em sua mochila, pude ver facilmente que ele não tinha uma visão equilibrada dos itens essenciais necessários em uma viagem para regiões naturais.

Você tem os “dez itens essenciais”? Perguntei. Dez itens essenciais? Quais são eles?

¹ “Equipamentos”, uma gíria utilizada pelos escaladores de montanhas e outros entusiastas ao ar livre para denotar alguma forma de equipamento, ferramentas, roupas, calçados ou outros artefatos utilizados em seus vários e diferentes esportes. Se você acreditar que o RUP possui muitos artefatos, consulte: www.mgear.com.

Randy, aqui está apenas a lista que você precisa. Destaquei uma folha de papel em branco e escrevi no início da página: “Os Dez Itens Essenciais”. Nela, escrevi 10 itens:

1. Mapa
2. Bússola
3. Óculos de sol e filtro solar
4. Roupa extra
5. Comida e água extras
6. Farol
7. Estojo de primeiros socorros
8. Acendedor de fogo
9. Fósforo
10. Faca²

Formatted: Bullets and Numbering

Começar com uma “Lista Abreviada”

É isso? Sim! É isso mesmo! Se você começar com esses 10 elementos, certificando-se de ter pego cada área incluída, as outras coisas se adequarão conforme necessário. É claro, cada um desses “itens essenciais” será aumentado ou reduzido proporcionalmente, dependendo do passeio, mas começando com uma “lista abreviada” e expandindo-a, conforme necessário, é muito mais fácil do que começar de uma lista longa e tentando decidir o que *não* levar.

À medida que você adquire experiência, desenvolverá sua própria “lista longa”, a qual fará sentido com o passar do tempo. Nesse momento, pode ser útil para você examinar minha lista e outras, para ver como será possível expandir corretamente a sua própria lista. Porém, as listas de duas pessoas nem sempre serão exatamente as mesmas (a não ser que eles tenham trapaceado!).

Memorizei essa mesma lista muitos anos atrás quando iniciei a escalada em montanhas e ainda a consulto, independente do tipo de passeio que estou preparando ou por quanto tempo pretendo ficar fora. Se você perguntar para qualquer escalador de montanhas experiente, “Você tem os ‘dez itens essenciais?’”, a maioria deles saberá exatamente sobre o que você está falando. Na verdade, grupos ou pessoas diferentes podem ter uma lista diferente dos “dez itens essenciais”, mas, em essência, eles serão os mesmos. E para cada passeio, os itens atuais que eu levo podem ser totalmente diferentes, mas ainda posso dizer que tenho os “dez itens essenciais”.

O Que Deveria Estar em sua Lista? “Bem, isso depende...”

Portanto, de que maneira isso está relacionado com o RUP? Muitas vezes, como ajudo as equipes de projetos a percorrerem muitos elementos no RUP, ouço perguntas do tipo: “Como devo percorrer todos esses itens e determinar quais deles são necessários para o projeto?” “Preciso disto?” “Não é óbvio que o RUP se destine apenas a projetos grandes?”

Freqüentemente, a maioria das respostas é (uma de nossas favoritas): “Bem, isso depende...”

O que eu realmente queria, decidi, é uma lista de “Dez Itens Essenciais do RUP” que posso dar às pessoas para orientação do processo—tal como fiz para meu amigo Randy. Essa lista poderia servir como um ponto de partida razoável para determinação dos elementos corretos a serem incluídos em qualquer projeto e poderia se aplicada, se eles estivessem indo para apenas um dia de longa caminhada (um projeto bem pequeno); passeio de mochila noturno, passeio de esqui ou caiaque (um projeto de tamanho médio em domínios diferentes) ou uma expedição maior do tipo Everest (um projeto maior e crítico). A idéia é focalizar no que se chama “a essência” do RUP—ou, realmente, em qualquer processo de software efetivo. Isso também vincula nossos conceitos de “boas práticas”³ no desenvolvimento de software.

² Para obter uma análise completa dessa lista de “Dez Itens Essenciais”, recomendo a você um livro excelente, “Mountaineering – The Freedom of the Hills”, 6th Edition, The Mountaineers, Seattle, WA, 1997. pp. 35-41.

³ As “boas práticas” que o Rational têm observado nos últimos anos incluem: (1) Desenvolver de Forma Repetitiva, (2) Gerenciar Requisitos, (3) Utilizar Arquiteturas de Componentes, (4) Modelar Visualmente, (5) Verificar Qualidade e (6) Controlar Alterações.

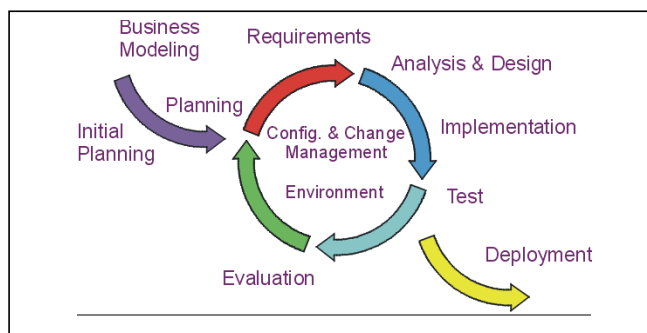
Construir Primeiro uma Estrutura

Um problema comum em muitos projetos, é que geralmente eles têm um foco rigoroso em uma área específica, aprofundando-se nos respectivos detalhes, sem a certeza de que haja conhecimento suficiente sobre os elementos "essenciais" que estão envolvidos no ciclo de vida total do processo, para gerar um produto de qualidade.

Então, quando o projeto é atrasado é dito, “Compreende, eu disse que o gerenciamento de requisitos apenas nos iria atrasar!”. Você pode substituir o “gerenciamento de requisitos” na instrução anterior por “utilizar casos”, “coletando métricas de projetos”, “utilizando o gerenciamento de configuração”, “utilizando uma ferramenta de trilha de defeitos”, “tendo reuniões de status” ou qualquer um dos números de frases com o qual estou certo de que você vai se familiarizar.

Acredito que é muito mais eficiente ter uma abordagem mais sistemática e holística, certificando-se de que os elementos principais de um processo sejam adequados (uma arquitetura, supostamente), antes de determinar o foco em qualquer uma das áreas de problemas específicos.

Uma vez que essa estrutura (ou arquitetura) para um processo de software de qualidade estiver no lugar certo, acredita-se que um projeto possa efetivamente focalizar uma área específica, a qual é identificada como sendo um contribuidor principal de seus problemas (e, geralmente, admite-se que, o gerenciamento de requisitos esteja, muitas vezes, exatamente no início da lista⁴). De qualquer forma, lembre-se que a abordagem do RUP deve basear essa seleção na identificação e priorização dos riscos do projeto e na determinação prévia de estratégias de mitigação para esses riscos identificados.



Os Dez Itens Essenciais do RUP

Portanto, quais são os “Dez Itens Essenciais do RUP”? A lista a seguir descreve qual o conjunto mínimo de itens de um projeto que eu acredito funcionar, se eles estiverem verdadeiramente seguindo a “essência” do Rational Unified Process:

1. Visão
2. Plano
3. Riscos
4. Problemas
5. Caso de Negócios
6. Arquitetura
7. Produto
8. Avaliação

← - - - Formatted: Bullets and Numbering

← - - - Formatted: Bullets and Numbering

⁴Como o Gerenciamento de Requisitos é uma das “Áreas de Processos Principais” iniciais do CMM (Capability Maturity Model), este não é de forma alguma um cenário improvável na determinação de melhoria do processo de desenvolvimento de software.

9. Controle de Alterações

10. Suporte ao Usuário

Formatted: Bullets and Numbering

Vamos examinar cada um desses itens individualmente, veja onde eles se encaixam no RUP e tente entender o motivo pelo qual cada um fez sua “lista abreviada” de itens essenciais.

Visão — Desenvolver uma Visão

Ter uma Visão clara é o principal para desenvolver um produto que atende as necessidades⁵ dos investidores.” — *Managing Software Requirements*, Dean Leffingwell & Don Widrig, Addison-Wesley Longman, January, 2000.

A Visão captura a “essência” da *disciplina de Requisitos* no RUP: analisar o problema, entender as necessidades dos investidores, definir o sistema e gerenciando os requisitos à medida que são alterados.

A Visão fornece uma base de alto nível, algumas vezes contratual, para os requisitos técnicos mais detalhados. Ela capta requisitos de nível alto e restrições de design, para que o leitor compreenda o sistema a ser desenvolvido. Ele fornece informações para o processo de aprovação do projeto e está, portanto, diretamente relacionado ao Caso de Negócios. Comunica os itens fundamentais “por que e o que” relacionados ao projeto e funciona como um calibre contra o qual todas as decisões futuras devem ser validadas.

O conteúdo da Visão, deve responder às perguntas a seguir, que podem ser divididas em artefatos separados e mais detalhados, conforme necessário:

- Quais são os termos-chave? (Glossário)
- Que problema estamos tentando resolver? (Declaração do Problema)
- Quem são os investidores? Quem são os usuários? Quais são as suas necessidades?
- Quais são as características do produto?
- Quais são os requisitos funcionais? (Casos de Uso)
- Quais são os requisitos não funcionais?
- Quais são as restrições de design?

Plano — Gerenciar no Plano

“O produto é simplesmente tão bom quanto o plano para o mesmo.”⁶

No RUP, o SDP (Plano de Desenvolvimento de Software) reúne todas as informações necessárias ao gerenciamento do projeto. Ele inclui vários artefatos separados, desenvolvidos durante a fase de Iniciação e é mantido durante todo o projeto.

É utilizado para planejar o projeto e as necessidades de recursos, e para acompanhar o progresso do planejamento. Trata de tais áreas como: Organização do Projeto, Planejamento (Plano do Projeto, Plano de Iteração, Recursos, Ferramentas), Plano de Gerenciamento de Requisitos, Plano de Gerenciamento de Configuração, Plano de Resolução de Problemas, Plano QA, Plano de Teste, Casos de Teste, Plano de Avaliação e Plano de Aceitação de Produto.

Em um projeto simples, tais áreas podem incluir apenas uma ou duas sentenças cada uma. Por exemplo, um Plano CM pode simplesmente declarar: “No final de cada dia, o conteúdo da estrutura de diretórios do projeto será compactado, copiado em um disco de zip datado e etiquetado, marcado com um número de versão e colocado no arquivamento central.”

O formato do próprio Plano de Desenvolvimento de Software não é tão importante quanto a atividade e o raciocínio de produção do mesmo. Portanto, não se preocupe com sua aparência – ou com quais ferramentas utilizará para construí-lo. Como disse Dwight D. Eisenhower, “O plano não é nada; o planejamento é tudo.”

⁵ *reais* dos investidores. “O objetivo é desenvolver um produto de qualidade, a tempo e no orçamento, que atenda as necessidades reais

⁶ Johnson Space Center Shuttle Software Group, “They Write the Right Stuff”, Charles Fishman, *Fastcompany*, Issue 6, p. 95, December, 1996.

Os números 2, 3, 4, 5 e 8 dos itens essenciais captam a “essência” da *Disciplina Gerenciamento de Projeto* no RUP: entender um novo projeto; avaliar o escopo e o risco; monitorar e controlar o projeto; planejar e avaliar cada iteração e fase.

Riscos — Identificar e Diminuir Riscos

Uma regra essencial do RUP é identificar e combater os itens de risco mais alto no início do projeto. A lista de riscos foi projetada para capturar os riscos aparentes para o sucesso do projeto. Ela identifica, em ordem decrescente de prioridade, os eventos que poderiam levar a um resultado negativo significativo.

Juntamente com cada risco, deve haver um plano para diminuí-lo. Isso serve como ponto focal para o planejamento de atividades do projeto e é a base para a organização das iterações.

Problemas — Designar e Acompanhar os Problemas

A comunicação aberta contínua com dados objetivos obtidos diretamente das atividades em andamento e das configurações do produto em desenvolvimento é importante em qualquer projeto. No RUP, isso é feito através das avaliações de status regulares, que fornecem um mecanismo para tratar, comunicar e resolver problemas de gerenciamento, problemas técnicos e riscos do projeto. Além de identificar os problemas, é necessário designar cada um deles a uma data de expiração e a uma pessoa responsável pela resolução. Isso deve ser regularmente acompanhado e atualizado conforme necessário.

Essas imagens fornecem informações ao gerente de projeto sobre o andamento. Embora o período possa variar, a função de determinação precisa capturar o histórico do projeto e tomar decisões para remover qualquer obstáculo ou gargalo que impeça o andamento.

Caso de Negócios — Examinar o Caso de Negócios

O Caso de Negócios fornece as informações necessárias, do ponto de vista de um negócio, para determinar se vale ou não a pena investir no projeto.

A finalidade principal do Caso de Negócios é desenvolver um plano econômico para realizar a Visão do projeto. Uma vez desenvolvido, o Caso de Negócios é usado para fazer uma avaliação precisa do Retorno do Investimento (ROI) fornecido pelo projeto. Ele fornece a justificativa para o projeto e estabelece suas restrições econômicas. Fornece informações para os tomadores de decisão econômica sobre o valor do projeto e é utilizado para determinar se o projeto deve continuar.

A descrição não deve se aprofundar nos itens específicos do problema, mas deve criar um argumento convincente a favor da necessidade do produto. Ela deve ser breve, no entanto, para que seja facilmente compreendida e lembrada por todos os membros da equipe. Em marcos críticos, o Caso de Negócios é reexaminado para verificar se as estimativas do retorno e do custo esperados ainda são precisas e se o projeto deve continuar.

Arquitetura — Projetar uma Arquitetura de Componente

No Rational Unified Process, a arquitetura de um sistema de software (em um determinado ponto) é a organização ou a estrutura dos componentes significativos do sistema que interagem por meio de interfaces, com elementos constituídos de componentes e interfaces sucessivamente menores. Quais são as partes principais? E como elas se ajustam?

O RUP fornece uma maneira metódica e sistemática de projetar, desenvolver e validar uma arquitetura de software. Essa é a “essência” da disciplina *Análise e Design* no RUP: definir uma arquitetura candidata, aperfeiçoar a arquitetura, analisar o comportamento e projetar componentes do sistema.

Para falar e tirar conclusões sobre a arquitetura do software, primeiro defina uma representação de arquitetura, uma forma de descrever aspectos importantes de uma arquitetura. No RUP, esta descrição é capturada no Documento de Arquitetura de Software, que apresenta a arquitetura em várias visualizações.

Cada visualização arquitetural trata de um conjunto específico de interesses, específicos dos investidores no processo de desenvolvimento: usuários finais, designers, gerentes, engenheiros de sistema, mantenedores e assim por diante. Isso serve como um meio de comunicação entre o arquiteto e outros membros da equipe do projeto, com relação a decisões significativas do ponto de vista da arquitetura, tomadas a respeito do projeto.

Produto — Construir e Testar de Forma Incremental o Produto

A “essência” das *disciplinas Implementação e Teste* no RUP serve para codificar, construir e testar de forma incremental os componentes do sistema, com liberações executáveis no final de cada iteração após a iniciação.

No final da fase de elaboração, um protótipo de arquitetura está disponível para avaliação, isso também pode incluir um protótipo de interface com o usuário, se necessário. No decorrer de cada iteração da fase de construção, os componentes são integrados em construções executáveis e testadas que se desenvolvem no produto final.

O importante para esse elemento essencial é um conjunto integrado de atividades de teste que acompanha a construção do produto – e também o Gerenciamento de Configuração em andamento e revê as atividades.

Avaliação — Avaliar os Resultados Regularmente

A Avaliação de Iteração captura os resultados de uma iteração, até que ponto os critérios de avaliação foram respeitados, as lições aprendidas e as mudanças de processos que devem ser implementadas.

A Avaliação de Iteração é um artefato essencial da abordagem iterativa. Dependendo do escopo e risco do projeto e da natureza da iteração, ela pode variar de um simples registro de demonstração e de resultados a um registro de revisão de teste formal e completo.

O importante aqui é focalizar os problemas do processo, bem como nos problemas do produto: "Quanto antes você alcançá-los, mais tempo terá para superá-los".

Controles de Mudanças — Gerenciar e Controlar Mudanças

A “essência” da *disciplina Configuração e Gerenciamento de Mudanças* é gerenciar e controlar o escopo do projeto, a medida que as mudanças ocorrem por todo o ciclo de vida do projeto, enquanto mantém o objetivo de considerar todas as necessidades dos envolvidos e atendê-las, até o quanto for possível.

Assim que o primeiro protótipo for colocado diante dos usuários (e, muitas vezes, antes disso), as alterações *serão* solicitadas. (Uma daquelas certezas da vida.) Para controlar essas mudanças e gerenciar, de forma eficaz, o escopo do projeto e as expectativas dos investidores, é importante que todas as mudanças em quaisquer artefatos de desenvolvimento sejam propostas por meio de Controles de Mudanças e gerenciadas com um processo consistente.

As Solicitações de Mudança são usadas para documentar e controlar defeitos, solicitações de melhorias e qualquer outro tipo de solicitação de mudança no produto. A vantagem das Solicitações de Mudança é que elas fornecem um registro de decisões e, devido a seu processo de avaliação, garantem que os impactos da possível mudança sejam compreendidos por todos os membros da equipe do projeto. As Solicitações de Mudança são essenciais para o gerenciamento do escopo do projeto, bem como para a avaliação do impacto das mudanças propostas.

Suporte ao Usuário — Fornecer Assistência ao Usuário

No mínimo, isso deveria incluir um Guia do Usuário, talvez implementado através da ajuda on-line e, também, pode incluir um Guia de Instalação e Notas sobre o Release. Dependendo da complexidade do produto, os materiais de treinamento também podem ser necessários, bem como uma lista de materiais juntamente com qualquer pacote do produto.

No RUP, a “essência” da *disciplina Implantação* é agrupar e entregar o produto, juntamente com quaisquer materiais necessários para ajudar o usuário final a aprender, utilizar, operar e manter o produto.

E sobre os Requisitos?

Algumas pessoas podem analisar minha lista de itens essenciais e discordar veementemente de minhas escolhas. Por exemplo, onde se encaixam os “Requisitos” neste painel? Não seria uma melhor opção em vez da “Visão”? Os requisitos são certamente essenciais, correto? Boa pergunta. Eu diria que basicamente é o primeiro item essencial – e para aqueles que preferem ter os “Requisitos” na lista, podem colocá-lo nesse momento em sua própria lista! (Lembre-se, cada equipe do projeto deve apresentar sua própria lista. Minha lista de dez itens essenciais apenas é destinada como um ponto de partida para discussão posterior.)

Entretanto, na prática, às vezes perguntaria a uma equipe do projeto (especialmente para um projeto interno), “Quais são seus Requisitos?” e receberia a resposta, “Na verdade, não temos nenhum requisito.” Primeiramente, isso me surpreendeu (vindo como uma bomba)! Como eles podiam não ter nenhum requisito? Então, descobri que o que eles lembram quando ouvem a palavra “requisitos” significa um conjunto de instruções “obrigatórias” externamente impostas, com as quais “devem” estar de acordo ou ter o projeto rejeitado – e, *de fato*, não possuem nenhuma delas! Eles podem estar envolvidos na pesquisa e no desenvolvimento, nos quais os requisitos do produto são desenvolvidos por todo o projeto.

Agora, se eu continuar com a resposta: O.K., qual é a visão do produto? Então, seus olhos brilhariam – continuaríamos facilmente cada uma das perguntas de A a G acima sem problemas – e, depois disso, os requisitos teriam um prosseguimento. Muitas vezes, isso é mais exato em um ambiente colaborativo, e não em um projeto contratual, onde os requisitos são “descobertos” mais do que impostos por decreto.

E sobre o Teste?

Sem dúvida, algumas de suas classificações apuradas, também informaram que não inclui o “Teste” como um dos “itens essenciais” do RUP. Por que isso? Ao contrário da Modelagem de Negócios, a qual sinceramente acredito ser opcional para um processo de desenvolvimento de software efetivo, em nenhum momento, acreditei que o teste seria opcional. Visualizo o teste como um conjunto integrado de atividades que acompanham o design e a construção do produto – grande parte da mesma forma que o Gerenciamento de Configuração em andamento e a revisão de atividades – e, coincidentemente, da mesma forma que a execução do padrão de processo de software IEEE 1074.

E, se você estivesse bem preparado, perceberia que o teste está, na verdade, incluído juntamente com a construção do produto (basicamente, o número sete), com os resultados do teste criticamente importantes para Verificação & Avaliação (essencialmente, número oito). A essência da abordagem iterativa do Rational Unified Process para desenvolvimento de software, é a criação, o de teste e a avaliação de forma contínua de versões executáveis do produto, a fim de afastar os problemas e resolver os riscos e as questões o mais cedo possível.

Resumo: Aplicando os Dez Itens Essenciais

Conforme foi mostrado, como a descoberta dos “Dez Itens Essenciais do RUP” pode fazer a diferença em minha vida?

Para Projetos Menores

Bem, primeiramente, se alguém como meu amigo, Randy, tivesse vindo, de fato, pedir conselho sobre como ele poderia utilizar o RUP para ele mesmo ou com uma equipe com pouca experiência construir um produto, apenas aprender sobre o processo, nestas circunstâncias, tenho minha própria lista de “dez itens essenciais” possível de ser compartilhada com ele, em vez de sobrecarregá-lo com a potência total do RUP e com o Rational Suites de ferramentas!

Na realidade, esses dez itens essenciais podem ser acompanhados sem suporte de ferramentas especificado! Um bloco de notas do projeto com uma seção dedicada para cada um dos dez itens essenciais, é realmente um ponto de partida muito bom para gerenciamento do projeto. E tenho encontrado Recados Adesivos inválidos ao gerenciar, dar prioridade e acompanhar inicialmente controles de mudanças para meu próprio aplicativo pessoal do RUP!

Para Projetos em Crescimento

Naturalmente, esses meios simples de aplicar rapidamente os dez itens essenciais tornam-se incontroláveis, conforme o crescimento do tamanho e da complexidade de um projeto; desta forma, muitos que procuram clientes potenciais para ferramentas do Rational, encontrarão perspectivas prontas neles, aplicando os “Dez Itens Essenciais do RUP”.

E, é claro, conseqüentemente, haverá a necessidade do RUP completo, também, bem como o suporte às ferramentas do Rational (se você fizer o trabalho correto!), uma vez que seja decidido aprofundar apenas o nível de superfície em cada área. Entretanto, eu o incentivaria a tratar dos “Dez Itens Essenciais do RUP”, ao longo das “Boas Práticas”, em vez de lidar com as ferramentas.

Para Equipes de Projeto Desenvolvidas

Para as equipes de projeto mais desenvolvidas, os “dez itens essenciais” podem ajudar no fornecimento de um método rápido de avaliação do equilíbrio dos elementos principais do processo e na identificação de áreas, nas quais o aprimoramento é mais vantajoso.

Em todos os projetos, é importante explorar o que ocorrerá se um desses elementos essenciais for ignorado – o que falhará se você não o tiver. Por exemplo:

- Sem visão? Você pode perder o rumo e ser facilmente confundido em desvios.
- Sem plano? Você não conseguirá acompanhar o andamento.
- Sem lista de riscos? Você pode estar se concentrando nas questões erradas atualmente e pode explodir em uma mina inesperada daqui a cinco meses.
- Sem lista de problemas? Sem responsabilidade, esses são os obstáculos para o sucesso.
- Sem caso de negócios? Você se arrisca a perder tempo e dinheiro no projeto; ele pode ser cancelado ou ir por água a baixo.
- Sem arquitetura? Você pode não conseguir lidar com questões de comunicação, de sincronização e de acesso a dados, conforme elas surgem. Pode haver problemas com a capacidade de ajuste e o desempenho.
- Sem produto (protótipo)? Introdução ao código de gravação; simplesmente acumular o trabalho de escritório não fará com que você tenha sucesso; apresentar itens em funcionamento na presença dos clientes.
- Sem avaliação? Não finja que não é com você. É importante encarar a verdade. Quão próximo você realmente está do prazo? Das metas em qualidade ou do orçamento?
- Não há solicitações de mudança? Como você as acompanha?
- Sem suporte ao usuário? O que acontece quando um usuário tem uma pergunta e não consegue entender como utilizar o produto? Com que facilidade se obtém ajuda?

Bem, aqui você encontrará!

Por que “dez” itens essenciais? Nenhuma razão específica, exceto que não quero *mais* de dez (não é preciso enumerar mais!). Eu teria ficado feliz por apresentar “Nove Itens Essenciais” ou um número menor, mas não foi possível tirar nenhum deles. Supõe-se que eles sejam qualificados como essenciais, certo? Pelo menos para mim! Você foi encorajado a utilizá-los como um ponto de partida em seu grupo de projetos: Quais são seus “10 itens essenciais”?

Se você for como eu, você talvez prefira propor uma pequena recordação ou acrônimo, para ajudá-lo a lembrar de sua lista dos “Dez Itens Essenciais”. Por exemplo, para diminuí-la para 4 sílabas, utilize V-PRI-BAPE-CU—e, certamente, é mantido o número de itens essenciais até 10. É possível contá-los, com astúcia, sem precisar consultar a lista no PalmPilot! (Agora, você sabe o meu segredo!)



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.