REVISTA DE TECNOLOGIA APLICADA (RTA)
v.5, n.3, Set-Dez 2016, p.46-52
ISSN: 2237-3713

http://dx.doi.org/10.21714/2237-3713rta2016v5n3p46

Comparação entre Metodologias de Desenvolvimento de Software baseadas nos métodos RUP e XP

Comparison between Software Development Methodologies based on RUP and XP methods

Denilson José Pereira
Programa de Pós Graduação em Governança e Gestão da Tecnologia da Informação – Senac – Jundiaí – SP.
denilsonjosepereira@gmail.com

Resumo
Este artigo compara as metodologias de desenvolvimento de software tradicionais em relação as metodologias denominadas ágeis. Foram escolhidos dois métodos dentre os mais difundidos para representar cada uma delas, sendo os dois métodos escolhidos o RUP e XP respectivamente. Os dois métodos de desenvolvimento escolhidos buscam a redução de riscos no desenvolvendo de software e utilizam técnicas de desenvolvimento incremental o que colabora para uma comparação mais detalhada de cada método. O artigo tem como foco o entendimento dos processos internos de cada um dos métodos utilizados para a comparação, concluindo com uma indicação de qual é a melhor metodologia a ser utilizada para determinados tipos de projetos.

Palavras-chave: Desenvolvimento de software; método RUP; método XP; redução de riscos

Abstract
This paper compares the methodologies of traditional software development methodologies in relation called agile. Two methods were chosen among the most widespread to represent each of them, and the two methods chosen RUP and XP respectively. The two methods chosen development seek to reduce risk in developing software and make use of incremental development which contributes to a more detailed comparison of each method. The article focuses on the understanding of the internal processes of each of the methods used for comparison, concluding with an indication of what is the best methodology to be used for certain types of projects

Keywords: Software development; RUP method; XP method; Risk reduction

INTRODUÇÃO

O sucesso de todo projeto de software começa sempre com um bom planejamento, com a escolha certa da arquitetura a ser utilizada para resolver o problema em questão, com profissionais bem capacitados e principalmente com a escolha de uma metodologia compatível com as características do projeto em questão.

É durante a etapa de planejamento que devemos estruturar todo o processo de desenvolvimento em relação a todos os recursos necessários e disponíveis para a execução do projeto. Esse planejamento deve sempre ter como foco e objetivo a entrega de um produto ou serviço de qualidade que atenda às necessidades do clientes dentro dos custos acordados e prazos previstos.

As metodologias tradicionais de desenvolvimento de software como a metodologia em cascata (waterfall), consiste em etapas que devem ser seguidas de forma sequencial, ou seja, o produto ou parte dele concluído em cada etapa é base para o início das próximas. Essa metodologia implica em um esforço enorme nas fases de levantamento de requisitos e de arquitetura pois para que seja possível iniciar as fases de desenvolvimento todos os requisitos já devem estar devidamente entendidos e documentados. Essa abordagem pode trazer grandes dispêndios de tempo e custo caso seja encontrado qualquer falha ou mudança nas fases posteriores como implementação ou testes, pois isso pode levar a grandes alterações de requisitos.

Os dois métodos escolhidos para a comparação (RUP e XP) utilizam uma abordagem que dividem o desenvolvimento de um projeto de software em ciclos, visando a redução de riscos e consequentemente custos e atrasos nos casos onde temos mudanças de requisitos.

O artigo tem como propósito analisar as semelhanças e diferenças presentes nas metodologias empregadas pelos processos RUP e XP. No capítulo 2 apresentaremos os tipos de metodologias de projetos mais utilizadas, no capítulo 3 detalharemos os processos em estudo (RUP e XP) respectivamente. No capítulo 4 compararemos as duas metodologias e por fim faremos uma síntese das ideias apresentadas através de uma conclusão.

TIPOS DE METODOLOGIAS DE PROJETO

Metodologias Iterativas

Metodologias iterativas foram propostas justamente para serem as respostas aos problemas encontrados no modelo em cascata e têm como objetivo principal o desenvolvimento de projetos de forma incremental. É um modelo de desenvolvimento que utiliza uma abordagem de dividir o desenvolvimento de um produto de software em ciclos sendo que em cada ciclo de desenvolvimento podem ser identificadas as fases de análise, projeto, implementação e testes. A cada iteração uma parte do sistema é desenvolvida, sendo o produto de cada nova iteração superior à da iteração anterior.

O desenvolvimento incremental possibilita aos desenvolvedores a aprenderem mais sobre o sistema em questão na medida em que vão construindo o software. Isso traz uma grande vantagem durante o projeto. Outra ponto positivo que podemos destacar é a capacidade de acomodar mudanças necessárias nos requisitos com um menor impacto no projeto.

Metodologias Ágeis

A Metodologias Ágil tornou-se conhecida em 2001, quando especialistas em processos de desenvolvimento de software estabeleceram os princípios e características comuns destes métodos. Assim foi criada a “Aliança Ágil” e efetuouse o estabelecimento do “Manifesto Ágil”.

Suas principais características são as pessoas e interações, ao contrário de processos e ferramentas, software executável ao invés de documentação extensa e confusa, colaboração do cliente e não

negociações estressantes baseados nos contratos e respostas rápidas as mudanças sem seguir planos previamente definidos.

As metodologias ágeis também dividem o desenvolvimento do software em iterações, buscando redução de riscos ao projeto e ao final de cada iteração, uma versão (release) funcional do produto é liberada ao cliente. Isso traz uma interação melhor entre o cliente e a equipe de desenvolvimento. Elas se destacam por priorizar os aspectos humanos no desenvolvimento do projeto, promovendo interação na equipe de desenvolvimento e o relacionamento de cooperação com o cliente. Comunicação face-a-face é preferida à documentação compreensiva.

Contrastes entre as Metodologias

As metodologias ágeis também utilizam a técnica de construção incremental de sistemas provenientes das metodologias iterativas, porém os períodos das iterações são geralmente menores, medidos em semanas, enquanto iterações de processos iterativos são geralmente medidos em meses ou até em alguns casos em anos dependendo do tipo e tamanho do projeto.

Metodologias de Desenvolvimento

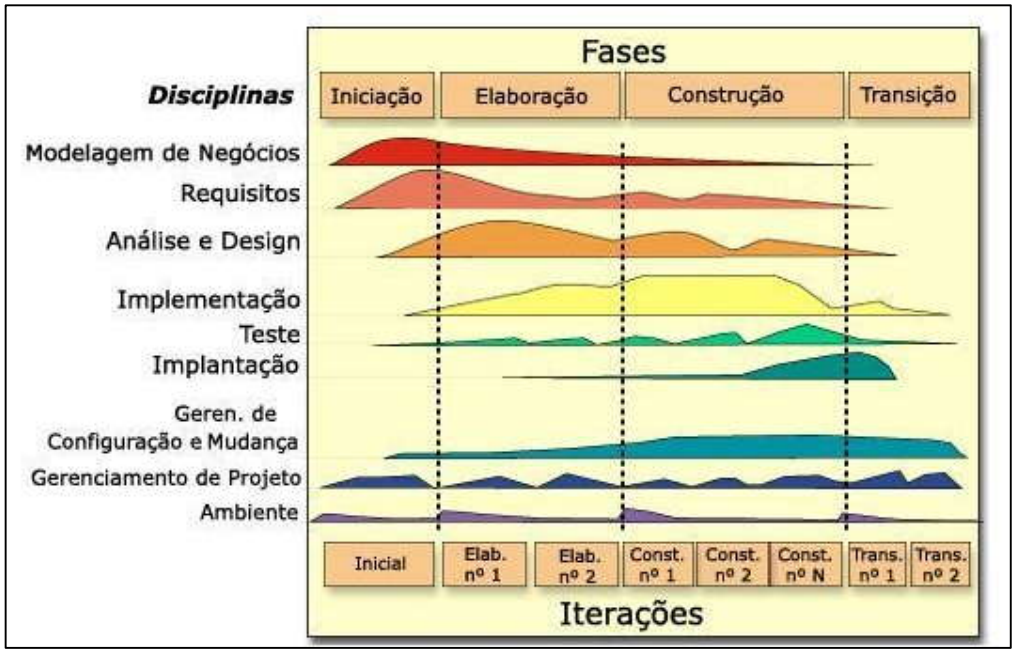
RUP – Rational Unified Process

RUP é uma metodologia iterativa de desenvolvimento. RUP é uma metodologia que pode ser configurável ou melhor adaptada aos seus propósitos de projeto conforme o seu tipo e tamanha. A Rational Software (atual divisão da IBM) desenvolveu e mantém o RUP.

A metodologia RUP identifica cada ciclo de desenvolvimento do projeto em quatro fases, cada uma com respectivos marcos de finalização definidos (chamados milestones). Os milestones são os indicadores de progresso do projeto, e são usados como base para decisões para continuar, abortar, ou mudar o rumo do projeto. As fases do RUP são:

1. Início (Inception): determina a fase onde se é levantado o escopo do desenvolvimento, sendo levantado uma visão do produto final a partir de um caso de uso (básico) definido.
2. Elaboração (Elaboration): fase onde é feito todo o planejamento das atividades e recursos necessários, onde são definidas funcionalidades e a arquitetura a ser desenvolvida.
3. Construção (Construction): implementação do software, é a fase onde o software é efetivamente construído. Em projetos grandes esta fase pode ser segmentada em várias iterações, visando à divisão em partes menores para um melhor gerenciamento.
4. Transição (Transition): é a fase onde o produto finalizado é passado aos usuários. Nesta fase ocorre treinamento dos usuários (e possíveis mantenedores) e é o momento da avaliação do produto.

A Figura abaixo apresenta a arquitetura de projetos que seguem a metodologia RUP. O eixo horizontal define aspectos dinâmicos, como ciclos, fases, iterações e marcos (milestones) e a vertical define os aspectos estáticos, como atividades, disciplinas, artefatos e papéis.



XP – Extreme Programming

XP é uma metodologia ágil, para o desenvolvimento de projetos de software cujas as suas características permitem que as especificações possam sofrer alterações no decorrer do projeto sem impactos profundos.

As iterações do XP costumam ser curtas, provendo constantes versões pequenas e funcionais do produto (releases) para o cliente, que por sua vez provê comentários e opiniões que realimentam a próxima iteração. O objetivo do XP é tornar o projeto flexível, diminuindo o custo a possíveis mudanças. O código produzido é tomado como indicador de progresso do projeto.

O ciclos da metodologia XP:

1. Exploração: é onde o cliente descreve o quais são as suas necessidades em relação ao sistema a ser construído na forma de escrita em cartões de estórias, cada um contendo uma funcionalidade desejada para o primeiro release.
2. Planejamento: é onde ocorre definição de prioridades das estórias escritas na fase de exploração. Nesta etapa os programadores estimam o esforço e o cronograma para cada uma das estórias.
3. Iterações para Release: nessa fase ocorrem diversas iterações até o primeiro release ser completado. Na primeira iteração é criado o sistema com toda a arquitetura, nas iterações seguintes serão adicionadas às funcionalidades de acordo com as prioridades estabelecidas.
4. Validação para Produção: é o momento onde o cliente faz os testes e verificações para validação do software para ser utilizado em ambientes de produção.
5. Manutenção: após o primeiro release para produção, há novas edições do sistema com novas funcionalidades.
6. Morte: quando não há mais estórias a serem implementadas, quando o cliente está satisfeito com o produto.

Um outro ponto a destacar em relação a metodologia XP são os seus valores: simplicidade, coragem, feedback, e comunicação.

COMPARAÇÃO ENTRE RUP E XP

As metodologias RUP e XP têm seu funcionamento baseado em iterações, são orientadas ao cliente e baseadas em papeis e responsabilidades. Uma análise inicial nos mostra que a dinâmica de desenvolvimento de software é mesma. Porém através da análise dos tópicos abaixo encontraremos diferenças na forma em que os artefatos são gerados, quantidade de papéis, etc.

Alocação de Tempo e Esforços

RUP segue 4 fases sequenciais constituindo um ciclo de desenvolvimento, produzindo uma nova versão de software. Em cada fase há um número de iterações, as quatro fases têm seu foco em diferentes atividades, podendo ocorrer em paralelo. A primeira fase, chamada de início foca o modelamento do negócio e a definição dos requisitos. A fase de elaboração foca em projetar (design), a de construção dá enfoque a implementação e aos testes e a última fase a de transição é onde a implantação e o gerenciamento de modificações são verificados.

O aspecto tempo em XP é diretamente relacionado com código produzido, ou seja, procura-se gastar o tempo primeiro nas funcionalidades básicas ou consideradas como núcleo do sistema e após essa base construída é que é focado nas funcionalidades extras. Porém como as liberações são definidas pelos clientes, os mesmos irão delimitar o tempo do projeto a partir de seu grau de satisfação com o software recebido.

Artefatos gerados

Os artefatos são informações que são produzidas, modificadas ou usadas por um processo [Kruchten 2000]. Como exemplos de artefatos têm-se modelos, código fonte e arquivos executáveis.

A comunicação dentro de um processo RUP é baseada em artefatos. Já em XP é baseada em comunicação oral, direta, o que restringe o uso de XP em projetos com grande distribuição geográfica ou em locais diferentes.

A pouca evidência de artefatos do XP, com foco em estórias de usuário e código, é visto como um fator que aumenta o risco do projeto, o conhecimento fica vinculado aos desenvolvedores e ao código já que pouca documentação é gerada.

Atividades, papéis e responsabilidades

O RUP define uma atividade como sendo o trabalho realizado por um papel, usando artefatos de entrada e produzindo artefatos de saída. Os papéis delimitados pelo RUP definem o comportamento e as responsabilidades dos envolvidos no projeto e estão agrupados em disciplinas:

1. Gerência de Projeto: tem o objetivo prover meios para a entrega do produto para o cliente que atenda às suas necessidades, gerenciando os riscos do projeto.
2. Modelamento de Negócio: faz o entendimento da estrutura onde o software será aplicado e os problemas atuais do cliente. Esta atividade deve assegurar que o cliente, os seus usuários e os desenvolvedores tenham um entendimento comum do produto a ser entregue.
3. Requerimentos: traduz as necessidades do sistema em forma de casos de uso, desenhando a interface com o usuário.

4. Analise e Desenho: especifica a forma de implementação dos requerimentos levantados através dos Use cases
5. Implementação: implementa as classes e os objetos em formas de componentes, os quais são individualmente testados.
6. Teste: testa e verifica se o produto funciona como o esperado, documentando falhas e problemas.
7. Deployment: tem como objetivo a distribuição, instalação e teste, provendo treinamento.
8. Configuração e Controle de Mudanças: concentram-se na garantia da rastreabilidade de versões dentro de um projeto. Através do definição de uma política adequada e de ferramentas específicas para
9. Ambiente: provê suporte adequado à organização do projeto, em ferramentas, métodos e processos.

Já a metodologia XP [Beck e Fowler 2000] apresenta apenas quatro atividades básicas: produção de código, testes, listening (escutar o cliente), e desenho. XP define os seguintes papéis:

1. Programador: escreve o código e realiza testes individuais.
2. Cliente: é o responsável por escrever as estórias e priorizá-las.
3. Testador: tem como objetivo auxiliar o cliente a criar testes funcionais. Os testes devem ser realizados regularmente e seus resultados divulgados.
4. Tracker: provê feedback no processo, comparando as estimativas com os resultados obtidos. Analisa o progresso de cada iteração e avalia se os objetivos podem ser alcançados com os recursos providos.
5. Coach: tem como responsabilidade o projeto como um todo, guiando os outros membros da equipe.
6. Consultor: membro externo que auxilia o time com os problemas técnicos específicos durante o projeto.
7. Big Boss: responsável pelo projeto, toma decisões e está em constante comunicação com a equipe para diagnosticar possíveis problemas ou falhas no processo.

Na comparação das definições das atividades e dos papéis e responsabilidades, verificamos que o RUP faz uma divisão de tarefas de forma específica, enquanto a divisão de papéis proposta pelo XP tem um caráter de uso-geral, sem atribuições específicas dentro das atividades.

CONCLUSÃO

Metodologias de desenvolvimento de software de uma forma geral buscam reduzir riscos e aumentar a qualidade do produto gerado. As metodologias RUP e XP têm esse objetivo, pois apresentam os mesmos valores como, envolvimento do cliente, iterações, testes contínuos e flexibilidade. Porém busca-se esses objetivos de forma diferente, através de implementações diferentes.

De forma geral o XP se apresenta como uma metodologia a ser utilizada em projetos onde os requisitos são voláteis e não claros sendo assim muito flexível, porém existe uma limitação de uso em equipes pequenas, pois não dá ênfase à documentação e sim a comunicação oral restringindo sua aplicação em projetos com equipes distribuídas geograficamente

O RUP possui uma estrutura melhor definida fazendo uma divisão melhor do projeto em relação as atividades, tarefas, papéis e responsabilidades e gera uma coleção de artefatos que são usados como produtos de entrada e de saída de processos. Essa estruturação permite que o RUP seja utilizado em projetos grandes, e com distribuição geográfica, com um custo adicional de gerência do projeto. Esse custo adicional pode não ser justificável em pequenas equipes ou em projetos menores que exigem uma maior agilidade.

REFERÊNCIAS BIBLIOGRÁFICAS

Kruchten, P. (2000) "The Rational Unified Process – An Introduction", AddisonWesley 2a edição.

Beck, K. and Fowler, M. (2000) "Planning Extreme Programming", Addison Wesley,1aedição.

Runeson, P. and Greberg, P. (2004) "Extreme Programming and Rational Unified Process – Contrasts or Synonyms?", Lund University, Sweden.