

Ontologia de um processo de desenvolvimento de software

Um processo de desenvolvimento está distribuído em *fases*, que possuem disciplinas *técnicas e gerenciais*, as quais expressam determinada área de conhecimento relevante ao desenvolvimento do projeto. Atividades são o bloco principal de construção de um processo e constituem-se de *tarefas*, que as detalham. Estas são descritas por *passos* e realizadas por determinados *papéis*, produzindo, como resultado final de sua realização, *produtos de trabalho* (artefatos, diagramas, documentos, dentre outros). A Figura 1 a seguir esquematiza as relações descritas.

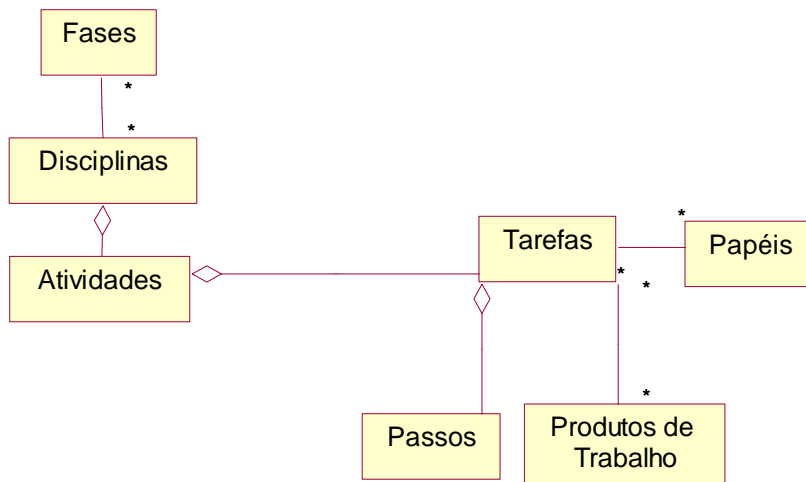


Figura 1.1-Ontologia de um processo de desenvolvimento

1.2-Estrutura das disciplinas no RUP

A Figura 1.2 abaixo exibe todas as disciplinas do RUP e a "intensidade" de sua realização ao longo de cada etapa.

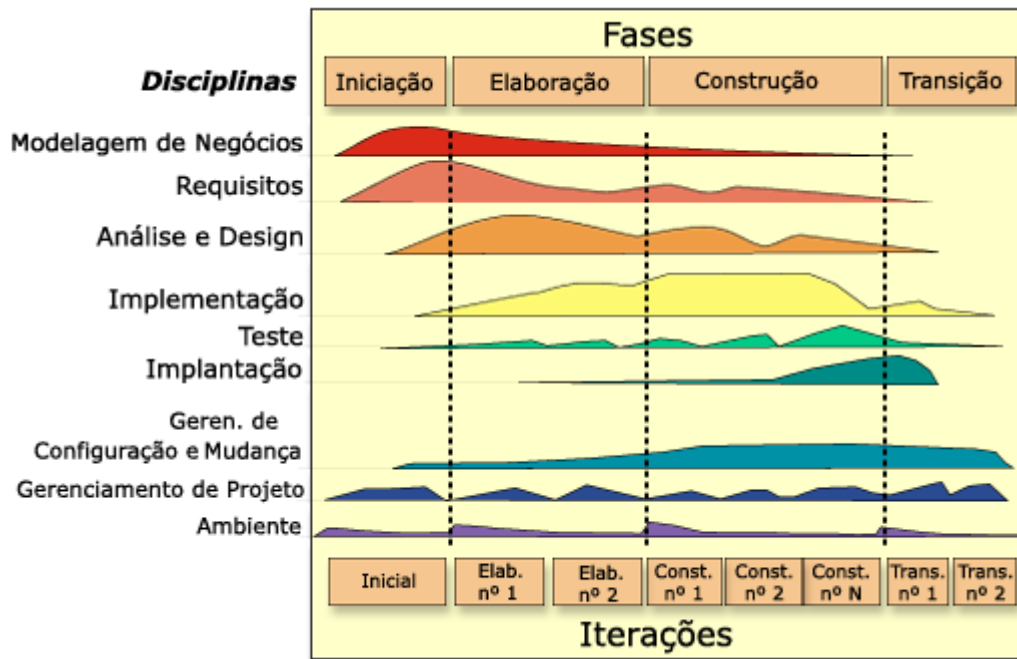


Figura 1.2-Estrutura de Disciplinas do RUP

As disciplinas são formalizadas por meio de fluxos de trabalho, que são diagramas de atividades que exibem a seqüência de atividades nos mesmos. Por exemplo, o fluxo de trabalho da disciplina Teste é o seguinte:

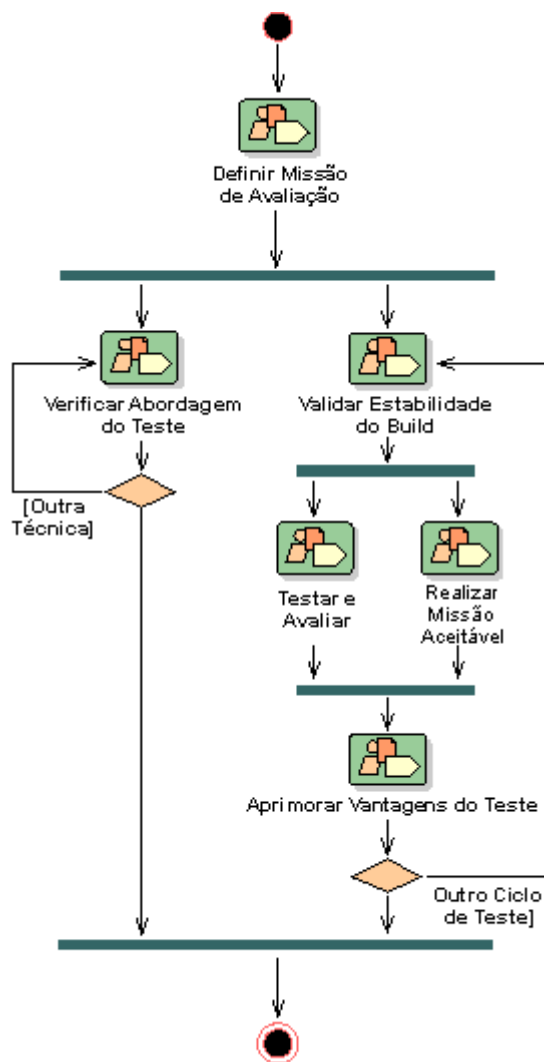


Figura 1.3-Sequência de atividades da atividade Teste

As atividades são “explodidas” em tarefas, papéis e artefatos. A figura abaixo exibe a tarefa Definir Missão da Avaliação:

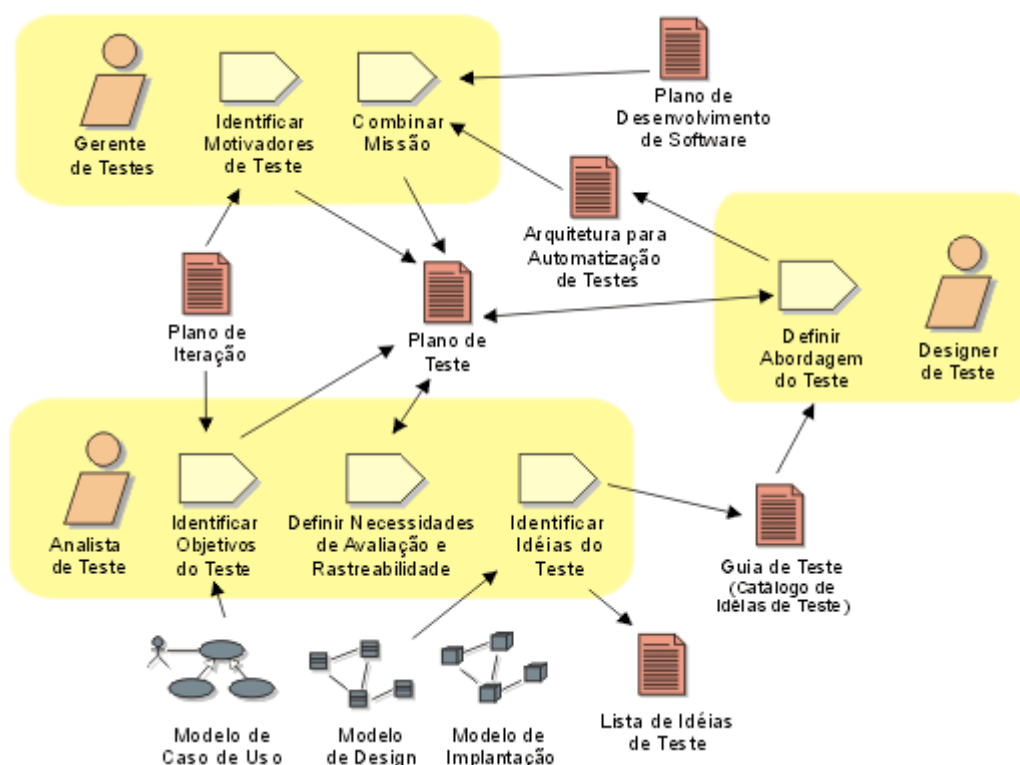


Figura 1.4- Detalhamento da atividade Definir Missão da Avaliação

Na Figura 1.4 temos:

Tarefas: Identificar Motivadores do Teste, Combinar missão, etc. Cada atividade possui passos que a descrevem.

Papéis: Gerente de Testes, Projetista de Teste, etc

Artefatos: Plano de teste, Plano de Iteração, Modelo de Caso de Uso, etc

Adaptação de um Processo

Analisar cada item (atividade, tarefa, passos da tarefa, artefato, itens de relatórios e papel) utilizando as planilhas abaixo, de forma a mantê-lo ou excluí-lo.

Nível: Disciplina		Nome: Teste
Tipo de Item a analisar: Atividade		
Item	Excluir/Manter	Justificativa
Definir Missão da Avaliação		
Verificar Abordagem do		

Teste		
Etc		

Para cada atividade mantida no nível disciplina:

Nível: Atividade		Nome: Definir Missão da Avaliação
Tipo de Item a analisar: Tarefa		
Item	Excluir/Manter	Justificativa
Identificar objetivos do teste	M	
Combinar Missão		
Etc		

Para cada tarefa mantida no nível atividade:

Nível: Tarefa		Nome: Identificar Objetivos de Teste
Tipo de Item a analisar: Passo		
Item	Excluir/Manter	Justificativa
Determinar software		
Identificar sugestões		
...		

Para cada atividade mantida no nível disciplina:

Nível: Atividade		Nome: Definir Missão da Avaliação
-------------------------	--	--

Tipo de Item a analisar: Papel		
Item	Excluir/Manter	Justificativa
Test manager		
Test designer		
Etc		

Para cada atividade mantida no nível disciplina:

Nível: Atividade		Nome: Definir Missão da Avaliação
Tipo de Item a analisar: Artefato		
Item	Excluir/Manter	Justificativa
Plano de teste		
Plano de Iteração		
Etc		

Para cada relatório mantido no nível atividade, fazer a tabela seguinte:

Nível: Relatório		Nome: Plano de Teste
Tipo de Item a analisar: Item do relatório		
Item	Excluir/Manter	Justificativa
Introdução		
Objetivos		
Etc		

1.3-Princípios da Abordagem Ágil

- Garantir a satisfação do consumidor entregando rapidamente e continuamente softwares funcionais;
- Softwares funcionais são entregues frequentemente (semanas, ao invés de meses);
- Softwares funcionais são a principal medida de progresso do projeto;
- Até mesmo mudanças tardias de escopo no projeto são bem-vindas.
- Cooperação constante entre pessoas que entendem do 'negócio' e desenvolvedores;
- Projetos surgem através de indivíduos motivados, e que deve existir uma relação de confiança.
- Design do software deve prezar pela excelência técnica;
- Simplicidade;
- Rápida adaptação às mudanças;
- Indivíduos e interações mais do que processos e ferramentas;
- *Software* funcional mais do que documentação extensa;
- Colaboração com clientes mais do que negociação de contratos;
- Responder a mudanças mais do que seguir um plano.

SCRUM

O **Scrum** é um processo de desenvolvimento iterativo e incremental para gerenciamento de projetos e desenvolvimento ágil de software. Apesar de a palavra não ser um acrônimo, algumas empresas que implementam o processo a soletram com letras maiúsculas como SCRUM. Isto pode ser devido aos primeiros artigos de Ken Schwaber, que capitalizava SCRUM no título.

Scrum não é um processo prescribente, ou seja, ele não descreve o que fazer em cada situação. Ele é usado para trabalhos complexos nos quais é impossível prever tudo o que irá ocorrer.

Apesar de Scrum ter sido destinado para gerenciamento de projetos de software, ele pode ser utilizado em equipes de manutenção de software ou como uma abordagem geral de gerenciamento de projetos/programas.

História

Inicialmente, o Scrum foi concebido como um estilo de gerenciamento de projetos em empresas de fabricação de automóveis e produtos de consumo, por Takeuchi e Nonaka no artigo "The New Product Development Game"

(Harvard Business Review, Janeiro-Fevereiro 1986). Eles notaram que projetos usando equipes pequenas e multidisciplinares (*cross-functional*) produziram os melhores resultados, e associaram estas equipes altamente eficazes à formação Scrum do Rugby (utilizada para reinício do jogo em certos casos). Jeff Sutherland, John Scumniotales e Jeff McKenna conceberam, documentaram e implementaram o Scrum, conforme descrito abaixo, na empresa Easel Corporation em 1993, incorporando os estilos de gerenciamento observados por Takeuchi e Nonaka. Em 1995, Ken Schwaber formalizou a definição de *Scrum* e ajudou a implantá-lo no desenvolvimento de softwares em todo o mundo.

Scrum junta conceitos de Lean, desenvolvimento iterativo e do estudo de Hirotaka Takeuchi e Ikujiro Nonaka.

A função primária do Scrum é ser utilizado para o gerenciamento de projetos de desenvolvimento de software. Ele tem sido usado com sucesso para isso, assim como Extreme Programming e outras metodologias de desenvolvimento. Porém, teoricamente pode ser aplicado em qualquer contexto no qual um grupo de pessoas necessitem trabalhar juntas para atingir um objetivo comum, como iniciar uma escola pequena, projetos de pesquisa científica, ou até mesmo o planejamento de um casamento.

Mesmo que idealizado para ser utilizado em gestão de projetos de desenvolvimento de software ele também pode ser usado para a gerência de equipes de manutenção, ou como uma abordagem para gestão de programas: *Scrum de Scrums*.

Características

Scrum é um esqueleto de processo que contém grupos de práticas e papéis pré-definidos. Os principais papéis são:

1. o **ScrumMaster**, que mantém os processos (normalmente no lugar de um gerente de projeto)
 2. o **Proprietário do Produto**, ou **Product Owner**, que representa os *stakeholders* e o negócio
 3. a **Equipe**, ou **Team**, um grupo multifuncional com cerca de 7 pessoas e que fazem a análise, projeto, implementação, teste etc.
- Cada *sprint* é uma iteração que segue um ciclo (PDCA) e entrega incremento de software pronto.
 - Um *backlog* é conjunto de requisitos, priorizado pelo Product Owner (responsável pelo ROI e por conhecer as necessidades do cliente);
 - Há entrega de conjunto fixo de itens do *backlog* em série de interações curtas ou *sprints*;
 - Breve reunião diária, ou *daily scrum*, em que cada participante fala sobre o progresso conseguido, o trabalho a ser realizado e/ou o que o impede de seguir avançando (também chamado de Standup Meeting ou Daily Meeting, já que os membros da equipe geralmente ficam em pé para não prolongar a reunião).

- Breve sessão de planejamento, na qual os itens do *backlog* para uma *sprint* (iteração) são definidos;
- Retrospectiva, na qual todos os membros da equipe refletem sobre a *sprint* passada.

O Scrum é facilitado por um *Scrum Master*, que tem como função primária remover qualquer impedimento à habilidade de uma equipe de entregar o objetivo do *sprint*. O Scrum Master não é o líder da equipe (já que as equipes são auto-organizadas), mas atua como um mediador entre a equipe e qualquer influência desestabilizadora. Outra função extremamente importante de um Scrum Master é o de assegurar que a equipe esteja utilizando corretamente as práticas de Scrum, motivando-os e mantendo o foco na meta da Sprint.

Scrum permite a criação de equipes auto-organizadas, encorajando a comunicação verbal entre todos os membros da equipe e entre todas as disciplinas que estão envolvidas no projeto.

Um princípio chave do Scrum é o reconhecimento de que desafios fundamentalmente empíricos não podem ser resolvidos com sucesso utilizando uma abordagem tradicional de "controle". Assim, o Scrum adota uma abordagem empírica, aceitando que o problema não pode ser totalmente entendido ou definido, focando na maximização da habilidade da equipe de responder de forma ágil aos desafios emergentes.

Uma das grandes vantagens do Scrum, porém, é que não tem abordagem "receita de bolo" do gerenciamento de projetos exemplificado no Project Management Body of Knowledge ou PRINCE2, que tem como objetivos atingir qualidade através da aplicação de uma série de processos prescritos.

Product backlog e Sprint backlog

Um *backlog* é uma lista de itens priorizados a serem desenvolvidos para um software. O *Product backlog* é mantido pelo *Product Owner* e é uma lista de requisitos que tipicamente vêm do cliente. O *Product Owner* pode alterá-lo a qualquer momento, desde que os itens alterados não estejam na *sprint*. O *Sprint backlog* é uma interpretação do *Product backlog* e contém tarefas concretas que serão realizadas durante o próximo *sprint* para implementar alguns dos itens principais no *Product backlog*. O *Product backlog* e o *sprint backlog* são, então, duas coisas totalmente diferentes, o primeiro contendo requisitos de alto-nível e o segundo contendo informações sobre como a equipe irá implementar os requisitos do produto.

Planejamento de sprint

Antes de todo *sprint*, o Product Owner, o Scrum Master e a Equipe decidem no que a equipe irá trabalhar durante o próximo *sprint*. O Product Owner mantém uma lista priorizada de itens de *backlog*, o *backlog* do produto, o que pode ser repriorizado durante o planejamento do *sprint*. A Equipe seleciona itens do topo do *backlog* do produto. Eles selecionam somente o quanto de trabalho eles podem executar para terminar. A Equipe então planeja a arquitetura e o design

de como o *backlog* do produto pode ser implementado. Os itens do *backlog* do produto são então destrinchados em tarefas que se tornam o *backlog* do *sprint*.

Reuniões

Daily Scrum

Cada dia durante o *sprint*, uma reunião de status do projeto ocorre. Isso é chamado de "scrum diário", ou "de pé o dia". Esta reunião tem diretrizes específicas:

A reunião começa precisamente no momento.

Todos são bem-vindos, mas apenas "porcos" podem falar.

O encontro é timebox e dura 15 minutos.

A reunião deve acontecer no mesmo local e mesma hora todos os dias

Durante a reunião, cada membro da equipe responde a três perguntas:

- O que você tem feito desde ontem?
- O que você está planejando fazer hoje?
- Você tem algum problema impedindo você de realizar seu objetivo?

É papel do ScrumMaster para facilitar a resolução desses impedimentos. Normalmente, isso deve ocorrer fora do contexto do Daily Scrum para que a reunião possa durar menos de 15 minutos.

Reunião de Planejamento da Sprint

No início do ciclo de *sprint* (a cada 7-30 dias), um "Sprint Planning Meeting" é realizado.

Selecione o trabalho que está a ser feito.

Prepare o Sprint Backlog que detalha o tempo que levará para fazer esse trabalho, com toda a equipe.

Identificar e comunicar o quanto o trabalho é susceptível de ser feito durante o *sprint* atual.