

Integrando MDSE no Processo de Desenvolvimento

Disciplina: Reúso de Software
Aluno: Daniel F. G. Feloni - 8434774

Agenda

- Introdução
- MDSE em um processo de desenvolvimento tradicional
- MDSE em Metodologias Ágeis
- Design Orientado a Domínio e MDSE
- Desenvolvimento Orientado a Teste e MDSE

Introdução

- Em qualquer mudança de tecnologia, aspectos organizacionais, gerenciais e sociais são os principais motivos do fracasso
- Apresentando MDSE sem considerar esses aspectos é um caminho certo para o fracasso
- Alguns conselhos de senso comum:
 - Primeiro projeto MDSE não deve ser crítico
 - Certifique-se que a gestão está comprometida com a decisão
 - Apoio ao longo dos problemas que aparecerão no projeto
 - Ter alguém com experiência na equipe
 - Comece pequeno, com um projeto piloto e crescer a partir deste

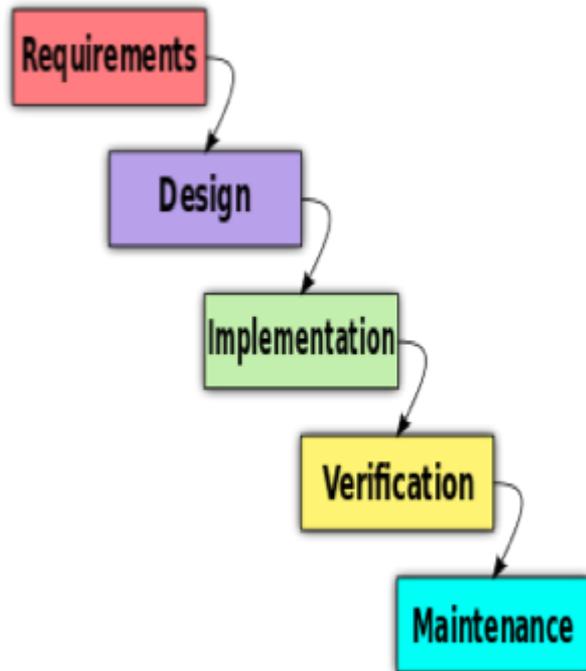
Introdução - Principais Desafios

- Perdas e ganhos da modelagem de software
 - Modelagem introduz novas tarefas e funções no desenv. processo
 - Algumas delas são a dor (agora há mais trabalho a ser feito)
 - Algumas outras obtêm o ganho (a manutenção é mais fácil com os modelos)
 - Se as pessoas que sentem a "dor" e as que sentem o "ganho" não fazem parte da mesma equipe
 - Cuidado com a motivação e problemas de percepção sobre o uso de modelagem
 - Reconheça o trabalho "doloroso"

Introdução - Principais Desafios

- Congruência sócio-técnico:
 - MDSE requer novas habilidades, funções e dependências na equipe de desenvolvimento
 - Sua organização deve ser capaz de corresponder a esses requisitos (por exemplo, se ninguém gosta / é bom em modelagem, que terá a cargo as tarefas de modelagem no processo?)

MSDE no processo tradicional

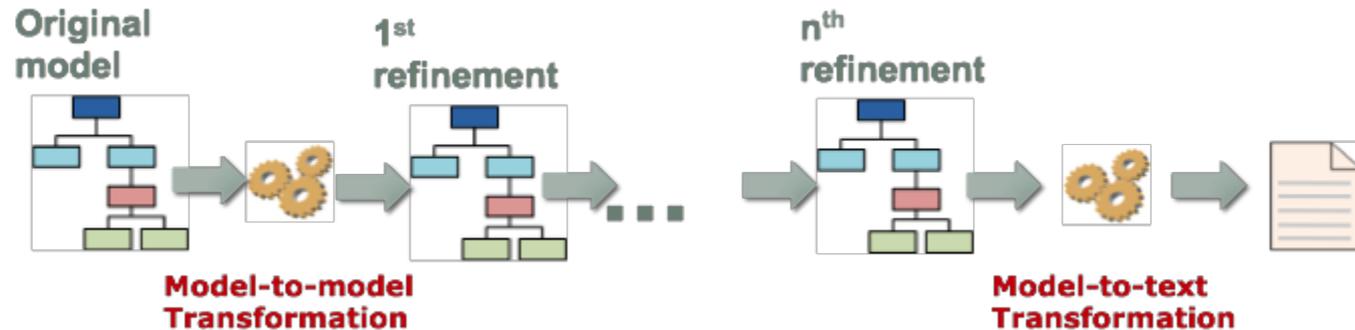


- Já é baseado em modelos
- Os modelos são tipicamente utilizados em cada fase do processo
 - modelos de requisitos
 - modelos de análise
 - modelos de design
 - modelos de implantação...
- Como MDSE contribui?

MSDE no processo tradicional

- Principal contribuição

- Transição entre baseado em modelos para orientado a modelos
- Oportunidade de (semi) automatizar as transições entre as diferentes fases do processo de desenvolvimento



MDSE em Metodologias Ágeis

- Manifesto Ágil propõe centro de desenvolvimento em torno de:
 - Indivíduos e interações sobre processos e ferramentas
 - Software funcionando mais que documentação abrangente
 - Colaboração do cliente sobre negociação de contratos
 - Responder a mudanças mais que seguir um plano
- MDSE tem um lugar neste manifesto? Críticas comuns:
 - Os modelos não são software de trabalho
 - Não pode ser testada
 - São apenas documentação
 - Trabalho extra para se adaptar às mudanças

Modelagem Ágil

- **Modelo com um propósito**
 - Identificar um propósito válido para a criação de um modelo e o público para este modelo
- **Travel Light**
 - Cada artefato que você cria, e depois decidir manter, terá que ser mantida ao longo do tempo.
- **Vários Modelos**
 - Você precisa usar vários modelos para o desenvolvimento de software, pois cada modelo descreve um único aspecto / visão do seu software.
- **Feedback rápido**
 - Ao trabalhar com outras pessoas em um modelo está a obtenção de feedback quase instantâneo de suas idéias.

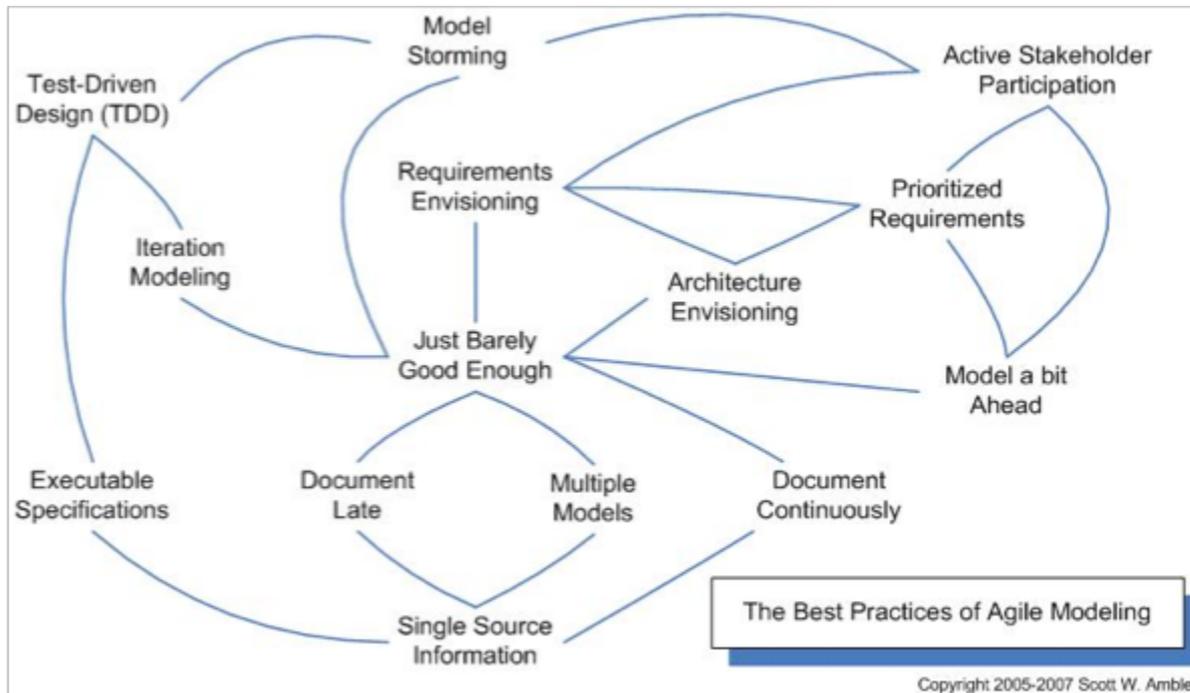
Modelagem Ágil

- **Assumir Simplicidade**
 - Mantenha seus modelos mais simples possível.
- **Aceitar a mudança**
 - Requisitos evoluem ao longo do tempo e por isso seus modelos também devem.
- **Mudança incremental**
 - Desenvolver bons modelos e suficientes. Evoluir modelos ao longo do tempo (ou simplesmente descartá-lo quando você não precisar mais dele) de forma incremental.

Modelagem Ágil

- Software de funcionando é seu objetivo principal
 - O objetivo principal não é produzir documentação alheia, artefatos de gerenciamento de estranhos, ou mesmo modelos. Qualquer (modelagem) atividade que não contribuem diretamente para esse objetivo deve ser questionada.
- Possibilitar o próximo esforço é o seu objetivo secundário
 - Para habilitá-lo você não vai apenas querer desenvolver software de qualidade, mas também criar apenas documentação suficiente e materiais de apoio para que as pessoas jogando o próximo jogo possam ser eficazes.

Modelagem Ágil - Práticas



MDSE Ágil

- Agile Modeling + modelos executáveis
- Modelagem efetiva de modelos executáveis para ir de modelos para software funcionando automaticamente da forma mais ágil possível.

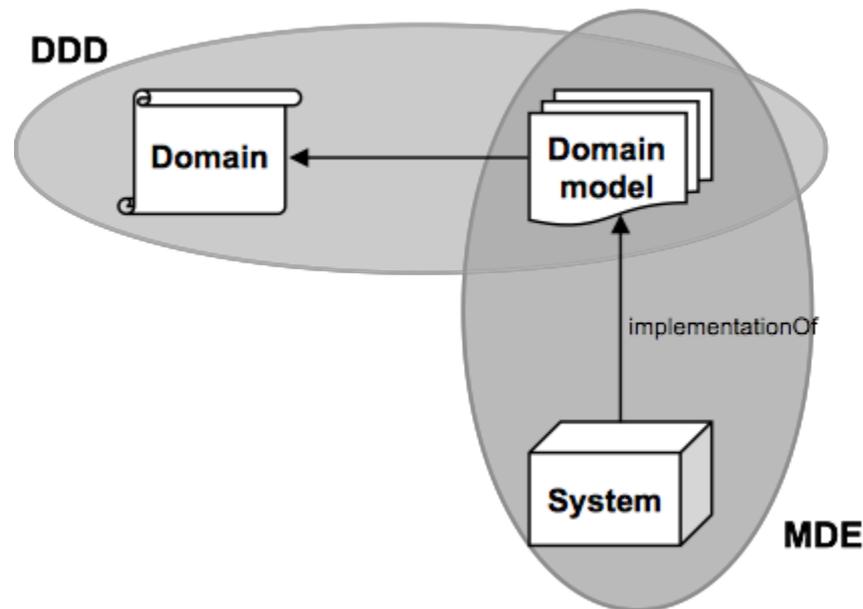
Design Orientado a Domínio e MDSE

- Domain-Driven Design (DDD) é baseado em duas ideias principais:
 - O foco principal do projeto de software deve ser o próprio domínio e não os detalhes técnicos.
 - Domínios complexos devem ser modeladas em primeiro lugar.
 - Um conjunto de práticas de design é fornecido para criar estes modelos.
- Assim, DDD enfatiza a importância de modelos de domínio.
- DDD e MDSE têm pontos em comum:
 - Necessidade do uso de modelos para representar o domínio do sistema
 - Concentre-se em aspectos independentes de plataforma

Design Orientado a Domínio e MDSE

- MDSE em DDD:

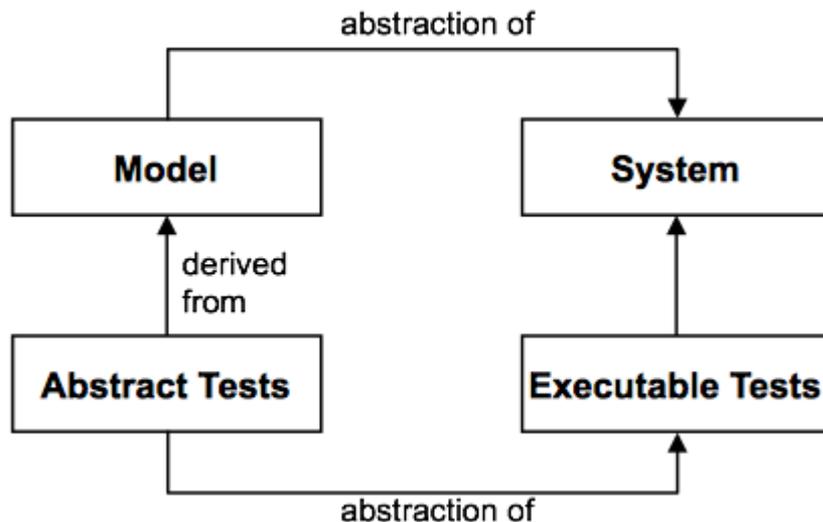
- Fornece um quadro para pôr em prática DDD (por exemplo, fornecendo linguagens de modelagem que podem ser usados em DDD)
- Maximiza o benefício que você pode sair dos modelos de domínio (por exemplo, transformando-os em código executável)



Desenvolvimento Orientado a Teste e MDSE

- Filosofia do teste primeiro:
 - Criar um teste executável para verificar a exatidão das novas funcionalidades-a-ser
 - Desenvolver o código para passar no teste
 - Refatorar o código e repita
- Integração de MDSE em TDD pode acontecer em dois níveis diferentes, dependendo do tipo de processo MDSE seguimos
 - Teste Orientado a Modelos
 - Modelagem Orientada a Teste

Desenvolvimento Orientado a Teste e MDSE



- Teste Orientado a Modelos
 - Se o sistema não é gerado automaticamente a partir dos modelos, é preciso verificar se a aplicação se comporta como se espera (ou seja, conforme definido nos modelos).
 - Os modelos podem ser usados para gerar os testes que a implementação terá de passar.

Desenvolvimento Orientado a Teste e MDSE

- Modelagem Orientada a Testes
 - Se o sistema é gerado automaticamente a partir dos modelos, então não há necessidade de se testar o sistema.
 - Modelos devem ser o foco de sua estratégia de testes.
 - Para cada novo modelo, escreva o primeiro teste do modelo, em seguida, escrever o modelo e verificar se o modelo passa no teste.

Fim

Dúvidas?