

Módulo 2 - Atividade Individual

Processo de Desenvolvimento de Software

Wellington Espindula

Janeiro de 2020

1. Defina modelo de processo de software. Escolha dois dos modelos de processo de software existentes, descreva-os, discuta suas vantagens e desvantagens de forma comparativa.

Dada a existência de diferentes processos de desenvolvimento de *software*, os modelos de *software* são uma forma de **agrupar os processos de desenvolvimento de forma a criar uma “família de processos”** de desenvolvimento de *software*. Dessa forma, modelos de processo de *software* são modelos criados com o intuito de representar de forma simples um processo de *software* e abstrair as características importantes de processos específicos. Dois exemplos de modelos de processo de *software* são:

- **Modelo em Cascata (*Waterfall*):** Um dos primeiros modelos utilizados no desenvolvimento de *software*, o modelo em cascata consiste em um processo de **desenvolvimento linear subdividido em fases**. Isto é, cada fase do projeto é realizada seguindo uma ordem. Inicialmente, realiza-se o levantamento dos requisitos. Só após esta fase estar completa, pode-se realizar o projeto do sistema, e assim consecutivamente. Por consequência, o **planejamento deve ser cauteloso**, visto que o modelo de desenvolvimento oferece **baixa flexibilidade à mudanças**.
- **Modelo Evolucionário (Prototipação): Rápido e barato**, o modelo evolucionário envolve a **prototipação**. A prototipação serve para testar ideias, viabilidade, demonstrar conceitos, etc. Dessa forma, a prototipação não necessariamente precisa de uma implementação visto que o **protótipo pode vir a ser descartado** e substituído por uma nova implementação de um sistema final.

Em relação à riscos e ao custo do erro, o modelo evolucionário se torna melhor, dado que o protótipo é rápido e barato e permite vislumbrar a viabilidade do produto. Já o modelo cascata necessidade de fases muito bem definidas e inflexíveis, então se houver erro, em fase de definição de requisitos por exemplo, ou o projeto for inviável será muito custoso, porque isso só será notado em fases mais avançadas do projeto - necessitando voltar às fases iniciais ou arruinando um projeto que já houve alto custo envolvido em diversas fases.

Ambos os projetos envolvem uma boa compreensão dos requisitos. Enquanto o evolucionário pode utilizar o protótipo para discutir modelos, ideias e chegar em uma maior compreensão dos requisitos do usuário, o método em cascata necessita de uma alta compreensão dos requisitos do sistema em fase inicial e estes se mantêm estáveis.

O modelo em cascata permite pouca participação do usuário e torna o projeto rígido à mudanças. Em contrapartida, o modelo evolucionário pode apresentar grande participação do usuário mas, ao mesmo tempo, pode apresentar mais variações nos requisitos ao passo que novos requisitos podem surgir a cada nova versão do protótipo avaliado pelo

usuário.

Um grande problema do uso do modelo evolucionar é que existe uma grande dificuldade em convencer o usuário da necessidade do descarte do protótipo. Bem como o processo evolucionar torna o processo do *software* pouco visível, visto que o progresso é mensurado pelos artefatos produzidos e outras preocupações mais abrangentes de *software* (como documentação e testes) podem ficar em segundo plano.

2. Descreva, com suas palavras, as principais atividades do processo de desenvolvimento de software e respectivos documentos gerados em cada atividade.

- **Especificação:** Atividade cujo objetivo é entender qual produto será criado, se é possível, quais são as restrições do produto, o que é essencial para este, quais são seus requisitos e como será realizado. Dessa forma a especificação é composta de atividades como estudo de viabilidade, análise, especificação e validação de requisitos. O maior documento gerado é a documentação da especificação de requisitos, que deve informar os requisitos de negócio, bem como requisitos de sistema. Ademais, a fase de Análise de requisitos também pode envolver protótipos, provas de conceito e modelos de sistema.
- **Projeto e Implementação:** Tornar a especificação em um sistema de *software*. Dessa forma, essa atividade envolve tornar as especificações em um projeto de *software* e implementá-lo. Dentre as etapas de projetos, tem-se o projeto de arquitetura, os projetos de componentes e o projeto de banco de dados. A programação envolve implementar o projeto em uma (ou mais) linguagem de programação. Na implementação, faz-se de extrema importância os testes e depuração.
- **Verificação e Validação:** As atividades de verificação e validação servem para verificar se o sistema está em conformidade com o que foi especificado e validar o *software* a partir do ponto de vista do cliente. Portanto, essa etapa envolve os testes de desenvolvimento, de integração e sistema e testes de aceitação.
- **Evolução:** Etapa responsável por evoluir o sistema envolvendo manutenção, correção de *bugs* e desenvolvimento de novas funcionalidades. Como o *software*, diferentemente de outros produtos, é flexível, essa etapa é de extrema importância porque surgem novas demandas e o *software* tem de se adaptar a isso. Ademais, é necessário detectar e corrigir os problemas existentes com a finalidade de diminuir a “deterioração” do sistema.

3. Modelo em Cascata. Em que tipos de projeto se poderia utilizar o modelo cascata?

O modelo em cascata consiste em um processo de desenvolvimento linear subdividido em fases. O modelo em cascata é **muito utilizado em projetos de grande porte que são desenvolvidos em lugares diversos**. Esse modelo é muito recomendado para projetos grandes com **requisitos bem definidos e pouco voláteis**.

4. Modelo Incremental. Descreva um exemplo de projeto de software suscetível ao uso de desenvolvimento incremental. Descreva as etapas de cada incremento.

O Modelo Incremental organiza as atividades em incrementos, aperfeiçoando o *software* paulatinamente. Um exemplo possível seria a de uma loja on-line, que tem-se uma ideia

inicial de como deve ser mas que deve ser melhorada ao longo do tempo com novos requisitos do usuário. As iterações poderiam ser (i) Desenvolver página inicial (estática) com produtos e preços; (ii) Desenvolver um CRUD de produtos; (iii) Interligar o página de produtos com os produtos cadastrados; (iii) Criar página de administrador com autenticação simples; (iv) Interligar página de administrador com CRUD de produtos; (v) Criar uma página de carrinho de compras; (vi) Criar um CRUD de pedidos; (vii) Integrar carrinho de compras com pedidos e com um *software* de pagamento on-line; (viii) Integrar pedidos de compras com página de administrador; ...

5. Considere o seguinte cenário. Você faz parte da equipe de um projeto de software e deve escolher um modelo de processo de desenvolvimento de software a ser seguido neste projeto. O software a ser desenvolvido visa auxiliar no gerenciamento de um hospital público, e tem as seguintes características.

1. O projeto consiste na criação de um software para dar suporte ao (a) agendamento de consultas; (b) realização de consultas; (c) cadastro de pacientes; e (d) verificação do seguimento, pelos médicos e enfermeiras, de protocolos estabelecidos pelo governo.
2. Nem o órgão do governo que solicitou o projeto, nem seus (perspectivos) usuários sabem exatamente todas as funcionalidades que o sistema deve oferecer.
3. O software deve entrar em produção o mais rápido possível.

Qual o modelo de processo de desenvolvimento de software que você escolheria para este projeto? Explique sua resposta, justificando-a com base no cenário acima descrito.

Para esse projeto utilizaria o Modelo Iterativo Incremental. Poderia, inclusive, ser utilizado metodologias ágeis para o desenvolvimento do *software* em questão. Visto que as funcionalidades mais básicas do *software* estão especificadas no item 1, poderia-se construir um sistema de forma ágil utilizando *Extreme Programming* (XP) por exemplo. Dessa forma, a primeira *release* entraria em produção o mais rápido possível com as funcionalidades mais básicas de acordo com os itens 1 e 3. Como o item 2 afirma que os usuários ainda não sabem exatamente todos as funcionalidades e requisitos, o modelo iterativo incremental usando metodologias ágeis ampliaria a participação dos usuários e tornaria o processo de aperfeiçoamento do *software* contínuo, assim os usuários poderão ao longo do processo de desenvolvimento ter cada vez mais claro o que desejam no *software*. Desse modo, o *software* teria *releases* mais frequentes envolvendo novas funcionalidades, *features* e correção de *bugs*.

6. Marque verdadeiro (V) ou falso (F). Os métodos ágeis...

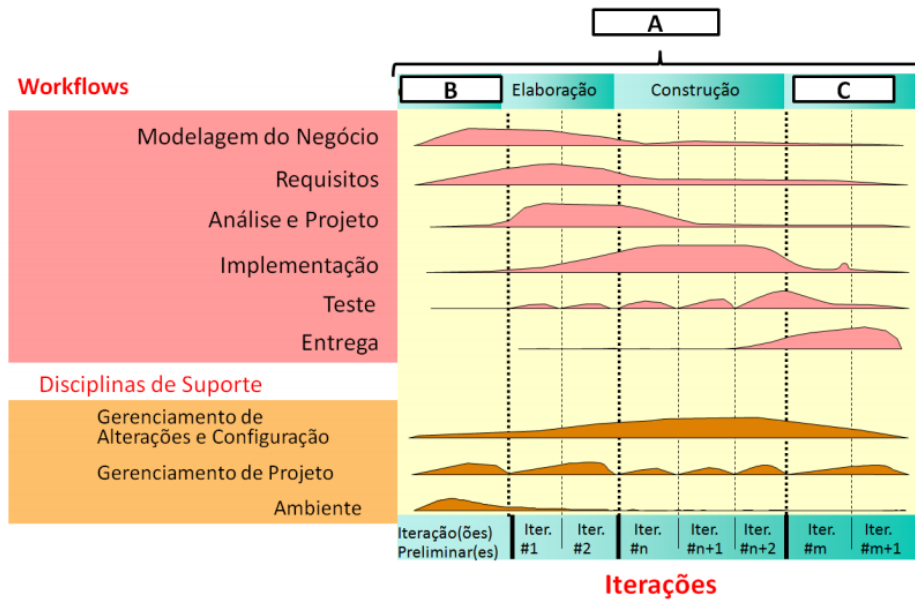
- (V) ... são um suplemento aos métodos existentes, e não uma metodologia completa.
- (F) ... são um processo prescritivo que detalha os passos que devem ser feitos para o desenvolvimento de um sistema de software.
- (V) ... são uma forma efetiva de se trabalhar em conjunto para atingir as necessidades das partes interessadas no projeto.
- (V) ... focam em simplicidade, sempre trabalhando para eliminar a complexidade do sistema.
- (V) ... têm como princípios a satisfação dos clientes, a estabilização de requisitos, e entregas de software operacional frequentemente.

- (V) ... funcionam na prática, não sendo teoria acadêmica.
- (V) ... são para o desenvolvedor médio, mas não são um substituto de pessoas competentes.
- (F) ... atacam a documentação dos sistemas de software, visto que eles aumentam significativamente o tempo e custo de desenvolvimento de software.
- (V) ... têm como dificuldade o envolvimento do cliente no desenvolvimento de software e que seja capaz de passar o tempo com a equipe de desenvolvimento.
- (F) ... pregam o não uso de ferramentas CASE, já que os documentos produzidos nelas são raramente utilizados e usualmente ficam desatualizados.

7. Escolha duas afirmações das apresentadas acima que você considera como falsas e explique o motivo, indicando o que deveria ser modificado para que elas se tornem verdadeiras.

- **Os métodos ágeis são um processo prescritivo que detalha os passos que devem ser feitos para o desenvolvimento de um sistema de software:** Os métodos ágeis são métodos mais adaptativos que prescritivos, dito isso, eles não criam uma documentação muito detalhada sobre o desenvolvimento de um sistema já que focam em se adaptar às mudanças de especificações e demandas.
- **Os métodos ágeis pregam o não uso de ferramentas CASE, já que os documentos produzidos nelas são raramente utilizados e usualmente ficam desatualizados.:** Por mais que os métodos ágeis foquem mais em um *software* funcional que em outros aspectos de *software* - como uma documentação detalhada -, estes não atacam as ferramentas CASE, mas, sim, envolvem a atenção contínua à excelência técnica. Então, testes, refatoração, uso de *Design Patterns* fazem parte do desenvolvimento ágil.

8. Com relação ao Rational Unified Process (RUP), responda as questões abaixo.



Considerando o diagrama do RUP acima, diga qual termo completaria cada um dos campos em branco no diagrama.

- **Campo A:** Fases
- **Campo B:** Concepção
- **Campo C:** Transição