

EXERCÍCIOS:

1. Introdução

1.1. Qual das afirmações abaixo é verdadeira em relação à assim chamada “crise do software”.

- a. Trata-se de um problema já plenamente resolvido pelas modernas “balas de prata” da engenharia de software.
- b. Trata exclusivamente da questão de projetos de desenvolvimento que sistematicamente atrasam.
- c. O termo surgiu nos anos 80 e tem a ver com a dificuldade inerente de implementar sistemas orientados a objetos.
- d. Trata-se de uma situação que perdura há tanto tempo que começa a ser considerada como uma dificuldade crônica inerente ao desenvolvimento de software.
- e. São dificuldades de desenvolvimento causadas exclusivamente pelo rápido incremento da complexidade do hardware.

1.2. Em relação a um dos mitos do software (Uma declaração geral de objetivos é suficiente para iniciar a fase de programação. Os detalhes podem ser adicionados depois.), pode-se afirmar que:

- a. É impossível para qualquer projeto conhecer com antecedência todos os requisitos.
- b. Não se pode esperar que a especificação inicial de um sistema esteja correta e completa, mas deve-se buscar isso como um objetivo.
- c. O analista não deve tentar descobrir todos os requisitos funcionais antes de iniciar a codificação do sistema.
- d. Uma boa ferramenta de análise de requisitos é suficiente para obter uma especificação completa e correta dos requisitos.
- e. Deve-se buscar fazer uma análise de requisitos suficientemente boa para que os requisitos encontrados sejam definitivos, ou seja, não mudem mais com o passar do tempo.

1.3. Um dos mitos administrativos de software diz que “Se o projeto estiver atrasado sempre é possível adicionar mais programadores para cumprir o cronograma.”. Porque isso é um mito?

- a. Porque adicionar mais pessoas a um projeto sempre causa mais atrasos.
- b. Porque as atividades de um projeto de software tipicamente não podem ser desenvolvidas em paralelo e, assim, de nada adianta adicionar mais pessoas.
- c. Porque é sempre difícil adicionar novas pessoas a um projeto já em andamento.
- d. Porque adicionar mais pessoas a um projeto sem um planejamento adequado para isso pode provocar mais atrasos ainda.
- e. Porque o ideal é que as funcionalidades não implementadas sejam deixadas para outro ciclo de desenvolvimento.

1.4. Dentre as atividades abaixo, qual a mais fortemente associada com o papel do engenheiro de software?

- a. Entrevistar o cliente e obter o melhor conjunto de requisitos possível.
- b. Escolher, definir e evoluir os processos relacionados ao desenvolvimento de software na empresa.
- c. Indicar qual a melhor tecnologia para desenvolver um determinado produto de software.
- d. Produzir código programado, testado e aceito pelo cliente.
- e. Definir e modelar a arquitetura do sistema a partir de seus casos de uso.

2. Processo

2.1. Em um processo, uma atividade pode ter itens opcionais e obrigatórios em sua descrição. Qual das opções abaixo apresenta *apenas* itens considerados obrigatórios?

- a. Saída(s), descrição, responsável(is), participante(s).
- b. Entrada(s), saída(s), descrição, responsável(is).
- c. Saída(s), descrição, responsável(is).
- d. Entrada(s), saída(s), descrição, responsável(is), participante(s).
- e. Saída(s), descrição, recursos(s) físico(s).

2.2. O primeiro ocorre em um tempo determinado consistindo na execução concreta de um conjunto de atividades que visam a criação de um produto específico. O segundo é um conjunto de regras que definem como o primeiro deve ser executado, sendo normalmente adotado por empresas como um conjunto de regras específicas que seus funcionários devem seguir sempre que trabalharem no primeiro. O terceiro é um conjunto de regras mais abstratas que especificam a forma geral do segundo, apresentando uma filosofia ou paradigma baseada na qual o segundo pode ser definido. *Primeiro, segundo e terceiro* são respectivamente:

- a. Processo, modelo de processo e projeto.
- b. Tarefa, atividade e procedimento.
- c. Atividade, procedimento e tarefa.
- d. Projeto, processo e modelo de processo.
- e. Procedimento, atividade e tarefa.

2.3. No contexto de processo de desenvolvimento de software, os objetivos de uma atividade podem ser descritos como:

- a. Promover a criação ou alteração de artefatos.
- b. Apenas criar novos artefatos.
- c. Apenas modificar artefatos existentes.
- d. Garantir que um conjunto de fases de projeto tenha sido executado.
- e. Criar código executável.

3. Modelos de Processo Prescritivo

3.1. Qual dos modelos de processo abaixo enfatiza o tratamento preliminar de riscos ao projeto?

- a. Modelo cascata.
- b. Sashimi.
- c. Espiral.
- d. Modelo V.
- e. Modelo orientado a cronograma.

3.2. Qual dos modelos de processo abaixo incorpora em sua filosofia a recomendação de Royce (1970): “Fazer duas vezes”?

- a. Sashimi
- b. Prototipação Evolucionária.
- c. Modelo W.
- d. Cascata com subprojetos.
- e. Modelo orientado a ferramentas.

3.3. Qual das características de ciclo de vida abaixo foi especialmente criada para diminuir os efeitos negativos da filosofia BDUF?

- a. Desenvolvimento iterativo.

- b. Desenvolvimento orientado a testes.
- c. Programação em pares.
- d. Reusabilidade.
- e. Processo de desenvolvimento gerenciado e otimizado.

3.4. Qual a diferença entre o modelo V e o modelo W que levou à criação deste último?

- a. O modelo W é iterativo e cíclico.
- b. O modelo W inclui o planejamento do teste durante a fase construtiva e atividades de correção durante a fase de verificação.
- c. O modelo W é mais adequado a equipes grandes e projetos complexos.
- d. O modelo W incluir a possibilidade de prototipação throw-away, o que ajuda a reduzir os riscos do projeto.
- e. O modelo W consiste em realizar todos os passos do modelo V duas vezes, sendo que o sistema construído na segunda vez será provavelmente melhor do que na primeira vez.

3.5. Qual das opções abaixo indica uma vantagem do modelo V em relação ao modelo Cascata original?

- a. O modelo provê uma substancial redução na quantidade de documentação, pois as equipes das diferentes fases trabalharão juntas boa parte do tempo.
- b. Este modelo é bem mais razoável de utilizar do que o modelo Cascata puro, visto que o fato de quebrar o sistema em subsistemas menores permite que subprojetos mais rápidos e fáceis de gerenciar sejam realizados.
- c. A principal vantagem está na sua ênfase nos testes e validações simétricos ao projeto.
- d. As primeiras iterações são as mais baratas do ponto de vista de investimento de tempo e recursos e ao mesmo tempo são as que resolvem os maiores problemas do projeto.
- e. Este modelo pode ser particularmente interessante se for difícil fazer o cliente entender os requisitos.

3.6. Qual característica do modelo Cascata com Subprojetos pode ser uma importante fonte de dificuldades?

- a. A integração contínua de pequenas partes do projeto.
- b. Uma grande integração de subsistemas desenvolvidos em separado.
- c. A possibilidade de dividir o projeto em subprojetos.
- d. A necessidade de executar um ciclo espiral de redução de risco antes de iniciar o projeto.
- e. A necessidade de planejar os testes detalhadamente antes de iniciar o desenvolvimento do código.

3.7. Assinale abaixo uma das desvantagens conhecidas do modelo Sashimi.

- a. O modelo é extremamente inflexível, com fases estanques que só terminam quando um marco (milestone) bem definido for produzido.
- b. Com este modelo é mais difícil definir marcos de final de fase, pois estas não são finalizadas de forma definitiva a não ser algum tempo depois de sua finalização relativa.
- c. A necessidade de existirem subprojetos paralelos no modelo pode dificultar a integração final do sistema.
- d. A grande quantidade de ciclos iterativos de curta duração pode tornar a tarefa de gerenciar o processo muito árdua.
- e. Não existem indicações claras sobre o que fazer em cada fase.

3.8. O modelo cascata é considerado o avô de muitos outros ciclos de vida de software. Os modelos mais novos, em alguns casos, adicionaram ao modelo cascata uma nova característica de trabalho predominante. Preencha os espaços em branco na segunda coluna com a letra correspondente da primeira coluna associando cada ciclo com a característica predominante que ele adicionou ao modelo cascata original.

Modelo	Característica predominante
--------	-----------------------------

a. Sashimi	() Entregar as funcionalidades possíveis em um prazo estrito pré-estipulado.
b. Cascata com subprojetos	() Divisão do projeto em partes que podem ser desenvolvidas em paralelo.
c. Cascata com redução de risco	() Disponibilizar várias versões aproveitáveis do sistema para o cliente ao longo do processo de desenvolvimento e não apenas no final.
d. Modelo V	() Fases que se entrelaçam no tempo.
e. Prototipação evolucionária	() Implementação preliminar de aspectos mais visíveis do sistema para compreensão de requisitos obscuros.
f. Entrega em estágios	() Ênfase no teste de software.
g. Modelo orientado a cronograma	() Mitigação dos potenciais problemas do projeto logo no início.

3.9. O ciclo de vida Espiral:

- a. Segue a filosofia *Big Design Up Front*.
- b. É mais um método para colocar pressão nos desenvolvedores do que um ciclo de vida planejado.
- c. Não usa análise de riscos nos primeiros ciclos da espiral.
- d. Cria um projeto preliminar a partir do primeiro conjunto de requisitos, que depois pode ser modificado.
- e. Também é conhecido como Waterfall com feedback.

3.10. O modelo “Entrega Evolucionária” é um meio termo entre outros dois modelos conhecidos. Qual característica permite determinar o grau em que este modelo se aproxima ou se afasta dos outros dois?

- a. Se a maioria dos requisitos solicitados após a apresentação do protótipo são acomodados no ciclo seguinte, então tende mais para Prototipação Evolucionária.
- b. Se a maioria dos requisitos solicitados após a apresentação do protótipo são acomodados no ciclo seguinte, então tende mais para Entrega em Estágios.
- c. Se as entregas continuarão sendo planejadas de acordo com o previsto e as novas funcionalidades acomodadas aos poucos nas entregas, então a abordagem se parece mais com Prototipação Evolucionária.
- d. Se as entregas continuarão sendo planejadas de acordo com o previsto e as novas funcionalidades acomodadas aos poucos nas entregas, então a abordagem se parece mais com o Modelo Espiral.
- e. Se a maioria dos requisitos solicitados após a apresentação do protótipo são acomodados no ciclo seguinte, então tende mais para Cascata com Redução de Risco.

3.11. Uma Linha de Produto de Software é especialmente recomendada quando a empresa...

- a. ... vai focar na produção de sistemas similares.
- b. ... deseja simplificar os processos de análise de requisitos e desenvolvimento.
- c. ... tem equipe mal preparada e precisa de normas para produzir melhor.
- d. ... vai produzir software para terceiros.
- e. ... deseja se tornar mais competitiva.

3.12. A implantação de uma Linha de Produto de Software tem como consequência:

- a. A análise de requisitos pode ser feita de maneira mais informal e rápida.
- b. Menor preocupação com reusabilidade.
- c. A adoção do ciclo de vida Waterfall ou uma de suas variantes.

- d. A necessidade de implantar uma estrutura eficiente para gerenciar partes comuns e variantes de famílias de produtos.
- e. A implantação de um sistema de fábrica de software.

4. Modelos Ágeis

4.1. Em relação aos modelos ágeis e prescritivos, é correto afirmar que:

- a. Modelos ágeis tem foco no controle e na descrição das atividades, enquanto modelos prescritivos tem foco nas pessoas e adaptação.
- b. Modelos ágeis tem foco nas pessoas e adaptação, enquanto modelos prescritivos tem foco no controle e na descrição das atividades.
- c. Modelos ágeis tem foco no controle e adaptação, enquanto modelos prescritivos tem foco nas pessoas e na descrição das atividades.
- d. Modelos ágeis tem foco nas pessoas e na descrição das atividades, enquanto modelos prescritivos tem foco no controle e adaptação.
- e. Modelos ágeis tem foco nas pessoas, enquanto modelos prescritivos tem foco no controle, na descrição das atividades e na adaptação.

4.2. No manifesto ágil, temos do lado esquerdo “indivíduos e interações”, “software funcionando”, “colaboração do cliente” e “responder às mudanças”. No lado direito temos: “processos e ferramentas”, “documentação compreensível”, “negociação de contrato” e “seguir um plano”. A filosofia do manifesto ágil diz que:

- a. Quanto mais cuidarmos do lado direito mais valor terão os itens à esquerda.
- b. Devemos valorizar o lado esquerdo em detrimento do lado direito.
- c. Devemos valorizar o lado direito em detrimento do lado esquerdo.
- d. Quanto mais cuidarmos do lado esquerdo mais valor terão os itens à direita.
- e. Não podemos valorizar tanto o lado direito quanto o esquerdo por serem incompatíveis.

4.3. No modelo SCRUM qual das responsabilidades abaixo é considerada típica do *product owner*?

- a. Introduzir novas funcionalidade durante o sprint.
- b. Planejar e gerenciar o processo de desenvolvimento.
- c. Indicar quais funcionalidades são mais importantes durante o planejamento de um sprint.
- d. Ser um resolvidor de conflitos e facilitador do processo.
- e. Explicar ao cliente porque o projeto está atrasado.

4.4. Em que situação espera-se que o *product owner* traga novas funcionalidades para dentro de um *sprint*?

- a. Quando o cronograma do projeto estiver atrasado.
- b. Quando o cronograma do projeto estiver adiantado.
- c. Quando a equipe for capaz de trabalhar horas extras.
- d. Quando a equipe for experiente e bem gerenciada.
- e. Nunca.

4.5. Qual o papel típico de um *Scrum Master*?

- a. É um desenvolvedor como qualquer outro.
- b. É o líder do projeto, ou seja, seu gerente.
- c. É o responsável pelos requisitos do projeto e pelo ROI (return of investment).
- d. É o planejador que define quais histórias de usuário serão tradas em cada sprint.
- e. É um facilitador para o processo de desenvolvimento e resolvidor de conflitos.

4.6. No contexto do modelo Scrum, o que é um *sprint backlog*?

- a. É uma visão dos requisitos de todo o projeto do ponto de vista da equipe de desenvolvimento.

- b. É uma visão dos requisitos de todo o projeto do ponto de vista do usuário.
- c. É uma visão dos requisitos de um ciclo de desenvolvimento do ponto de vista do usuário.
- d. É uma visão dos requisitos de todo o projeto do ponto de vista tanto do usuário quanto da equipe de desenvolvimento.
- e. É uma visão dos requisitos de um ciclo de desenvolvimento do ponto de vista da equipe de desenvolvimento.

4.7. Em relação à prática XP de refatoração, pode-se afirmar que:

- a. Deve sempre ser evitada, pois atrasa o desenvolvimento e o ciclo de tamanho fixo não pode atrasar.
- b. Deve ser usada apenas em situações em que decisões arquiteturais anteriores estejam prejudicando a performance do sistema.
- c. Deve ser usada apenas para remover redundâncias de código, simplificando sua compreensibilidade e browseabilidade.
- d. Deve ser feita diariamente, haja necessidade ou não.
- e. Deve ser feita sempre que for detectado algum problema de design, mesmo que este esteja funcionando.

4.8. Em relação à “propriedade do código”, do ponto de vista da equipe de desenvolvimento, pode-se afirmar que:

- a. XP recomenda a posse coletiva e FDD recomenda que cada artefato tenha um dono.
- b. XP e FDD recomendam que cada artefato tenha um dono.
- c. XP e FDD recomendam a posse coletiva.
- d. XP recomenda que cada artefato tenha um dono e FDD recomenda a posse coletiva.
- e. XP recomenda a posse coletiva e FDD nada afirma sobre isso.

4.9. No modelo XP, como se recomenda que seja feita a integração de código?

- a. Vários pares podem integrar pequenas unidades de código em paralelo.
- b. Apenas um par de programadores integra pequenas unidades de código de cada vez.
- c. Apenas um par de programadores integra grandes unidades de código de cada vez.
- d. Vários pares podem integrar grandes unidades de código em paralelo.
- e. Apenas um par de programadores integra o código produzido por todos os outros pares independentemente de seu tamanho.

4.10. No contexto do método XP, suponha que a equipe chegou à conclusão de que uma história de usuário necessitaria de 8 semanas para ser implementada. Neste caso a equipe deve:

- a. Dividir a história em pelo menos três histórias.
- b. Dividir a história em exatamente duas histórias.
- c. Seguir o projeto acomodando a história em um ciclo de oito semanas.
- d. Aumentar o tamanho da equipe para dar conta da história em menos tempo.
- e. Deixar esta história para o final do projeto, quando os elementos de maior risco já tenham sido tratados.

4.11. O que é um “spike” ou “solução afiada” no contexto do Método XP?

- a. Uma forma de avaliação que visa aumentar a produtividade da equipe.
- b. O nome que se dá às iterações ou ciclo que produzem entregas (deliveries).
- c. As reuniões diárias onde cada um fala sobre o que fez, o que vai fazer e o que o impede de continuar.
- d. Um desenvolvimento ou teste projetado especificamente para analisar e possivelmente resolver um risco.
- e. É um sinônimo para “simplicidade de projeto”.

4.12. No contexto do método XP, qual é a ação recomendada quando se percebe que o cronograma de uma iteração não vai ser cumprido, ou seja, quando se percebe que a quantidade de trabalho real será bem maior do que o previsto inicialmente?

- a. Deve-se aumentar o tamanho da equipe de forma a cumprir com os prazos estabelecidos.
- b. Deve-se postergar a data de entrega aumentando a duração do ciclo, o que pode ser compensado por ciclos mais curtos na continuação do projeto.
- c. Deve-se realizar uma reunião de planejamento de entregas e repassar algumas entregas para ciclos posteriores.
- d. Deve-se deixar algumas tarefas inacabadas e retomá-las no ciclo seguinte.
- e. Deve-se trabalhar horas extras de forma que o cronograma não seja prejudicado.

4.13. Servem ao mesmo propósito de casos de uso, mas não são a mesma coisa. São usados no lugar do documento de requisitos. Devem ser escritos pelos próprios interessados como sendo as coisas que eles precisam que o sistema faça para eles. Podem ser usados para definir os testes de aceitação. São:

- a. Histórias de usuário.
- b. Product backlog.
- c. Sprint backlog.
- d. Testes de aceitação (costumer tests).
- e. Plano de iteração.

5. UP – Processo Unificado

5.1. Qual o *milestone* (macro-objetivo) da fase de Concepção do processo unificado?

- a. A identificação dos requisitos funcionais e não funcionais e sua revisão
- b. A identificação dos casos de uso de alto nível, sua priorização e planejamento do projeto.
- c. O tratamento (análise detalhada) de todos os casos de uso identificados como processos de negócio.
- d. A entrega do sistema funcionando.
- e. O sistema funcionando em seu ambiente definitivo, com os dados finais e usuários devidamente treinados para seu uso.

5.2. Segundo a literatura a fase mais longa e trabalhosa do processo unificado é:

- a. Concepção
- b. Elaboração
- c. Construção
- d. Transição
- e. Implantação

5.3. Qual fase do processo unificado se caracteriza por produzir uma arquitetura estável ao seu final?

- a. Concepção
- b. Elaboração
- c. Construção
- d. Implantação
- e. Produção

5.4. No Processo Unificado (UP), quais seriam as três listas de elementos que fornecem as entradas para compor os objetivos de uma iteração no momento de seu planejamento?

- a. Casos de uso, modificações e riscos.
- b. Riscos, casos de uso e requisitos.
- c. Requisitos, riscos e artefatos.
- d. Modificações, casos de uso e artefatos.
- e. Riscos, requisitos e tarefas.

5.5. O Processo Unificado é dirigido por casos de uso e centrado na arquitetura. Na prática isso significa que:

- a. Os casos de uso devem ser refinados de forma estruturada e top-down de maneira que a arquitetura do sistema seja construída como um refinamento das funções representadas por estes casos de uso.
- b. A arquitetura é derivada de forma fragmentada para cada caso de uso, ou seja, cada caso de uso cria uma mini-arquitetura que serve exclusivamente a ele.
- c. Os casos de uso servem como referência para que se estude o problema e se defina a funcionalidade que o sistema deve ter, e a arquitetura deve ser construída incrementalmente de forma que essa funcionalidade se torne disponível.
- d. Os casos de uso representam o aspecto estático do sistema, enquanto a arquitetura representa o aspecto dinâmico.
- e. Antes de pensar nos casos de uso deve-se saber qual a arquitetura na qual o sistema será implementado.

5.6. Quando se diz que o Processo Unificado é focado em riscos, o que exatamente isso significa?

- a. Que existe uma fase do processo unificado cujo objetivo é mitigar riscos.
- b. Que o processo unificado tem uma disciplina de gerência de riscos bem definida.
- c. Que os casos de uso são priorizados de forma que aqueles que apresentam maior risco sejam tratados em primeiro lugar.
- d. Que as iterações devem considerar sempre o tratamento de riscos antes do desenvolvimento dos casos de uso.
- e. Que o processo de testes do processo unificado garante que o software esteja livre de problemas que são riscos para o usuário.

5.7. No contexto do RUP, além do plano da fase e do status do projeto, quais as listas priorizadas que devem servir de base para um plano de iteração?

- a. Casos de uso e riscos.
- b. Casos de uso, riscos e modificações.
- c. Casos de uso.
- d. Casos de uso, riscos, modificações e tarefas de programação.
- e. Casos de uso, riscos e tarefas de programação.

5.8. O Enterprise Unified Process (EUP), foi proposto como uma extensão do RUP e entre outras coisas adiciona duas novas fases. Quais são?

- a. Concepção e Aposentadoria.
- b. Gerenciamento de Riscos e Produção.
- c. Produção e Aposentadoria.
- d. Elaboração e Concepção.
- e. Elaboração e Transição.

5.9. Qual das disciplinas abaixo foi introduzida no modelo EUP, mas não existia no RUP?

- a. Modelagem de negócio.
- b. Análise e projeto.
- c. Gerenciamento de mudança e configuração.
- d. Gerenciamento de projeto.
- e. Gerenciamento de pessoas.

6. Respostas:

Questão	% respostas obtidas					Semestre	#alunos
	a	b	c	d	e		
1.1. Qual das afirmações abaixo é verdadeira em relação à assim chamada “crise do software”.1.1	8	4	4	71	13	2011.2	24

1.2	0	83	8	0	8	2011.2	24
1.3	0	0	3	97	0	2011.1	35
1.4	3	86	3	0	9	2011.1	35
Questão	a	b	c	d	e	Semestre	#alunos
2.1	6	6	74	14	0	2011.1	35
2.2	0	50	0	50	0	2011.1r	2
2.3	94	0	0	6	0	2010.1	36
Questão	a	b	c	d	e	Semestre	#alunos
3.1	0	4	96	0	0	2011.2	24
3.2	27	27	27	16	3	2010.2	37
3.3	83	8	0	8	0	2011.2	24
3.4	0	92	0	0	8	2011.2	24
3.5	0	0	100	0	0	2011.1r	2
3.6	0	96	0	4	0	2011.2	24
3.7	.	x	.	.	.	2011.2r	-
3.8	R.: gbfaedc Ac.: 63%					2011.1	35
3.9	5	0	0	76	19	2009.1	21
3.10	46	22	19	14	0	2010.2	37
3.11	81	8	0	8	3	2010.1	36
3.12	8	3	3	88	0	2009.2	40
Questão	a	b	c	d	e	Semestre	#alunos
4.1	0	96	0	0	4	2011.2	24
4.2	8	19	0	73	0	2010.2	37
4.3	0	8	79	8	4	2011.2	24
4.4	0	27	0	16	57	2010.2	37
4.5	0	0	3	11	86	2011.1	35
4.6	6	3	8	8	75	2010.1	36
4.7	0	0	17	21	63	2011.2	24
4.8	63	0	8	0	29	2011.2	24
4.9	20	66	0	3	11	2011.1	35
4.10	73	14	5	0	8	2010.2	37
4.11	0	3	8	81	8	2010.2	37
4.12	3	8	81	6	3	2010.1	36
4.13	100	0	0	0	0	2011.1r	2
Questão	a	b	c	d	e	Semestre	#alunos
5.1	42	54	4	0	0	2011.2	24
5.2	13	17	58	8	4	2011.2	24
5.3	8	81	0	8	3	2010.2	37
5.4	x	2011.2r	-
5.5	13	3	75	5	5	2009.2	40
5.6	50	0	50	0	0	2011.1.r	2
5.7	53	33	3	8	3	2010.1	36
5.8	14	3	69	14	0	2010.1	36
5.9	21	0	25	0	54	2011.2	24