



Discursivas

Engenharia de Software

Prof. Lúcio Camilo

Email: luciocamilo@gmail.com

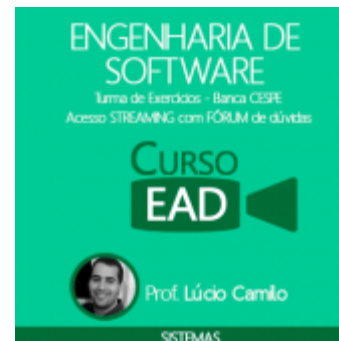
<http://www.itnerante.com.br/profile/LucioCamilo>



Lúcio Camilo

- Resumo – CV
- Analista de Sistemas TCE/RJ
- Pós Graduação em Engenharia de Software
- Autor do Livro “Android para Desenvolvedores”, Editora Brasport
- MBA Gerenciamento de Projetos
- Certificações Profissionais:
 - SCJP, OCWD, OCJA Part I
 - RHSA, Big IP Essentials e Advanced
- Professor de Concurso no Curso Multiplus RJ

Cursos já Lançados



Premissas

- Critérios de Correção e Tipo de Questão
- Conhecimento Prévio de Português
- Conhecimento do Edital



Conhecer o Edital

- ❑ Tipo de Prova:
 - ❑ Estudo de Caso;
 - ❑ Parecer;
 - ❑ Texto;
 - ❑ Perguntas Objetivas;



Edital BNDES 2012

- **Pontuação:**

- 7.1.3

- Após a 1ª Fase, os candidatos serão classificados por formação a partir do total de pontos obtidos, sendo **eliminado** o candidato que obtiver aproveitamento inferior a **60%** (sessenta por cento) do total da pontuação do conjunto das provas objetivas. Será, ainda, **eliminado** o candidato que **não** obtiver o **mínimo** de **3,0** (três) pontos na prova de **Língua Estrangeira**, de **10,0** (dez) pontos na prova de **Língua Portuguesa** e de **35,0** (trinta e cinco) pontos na prova de **Conhecimentos Específicos**.

- 7.1.4

- Será considerado **habilitado** à 2ª Fase o candidato não enquadrado nos critérios de eliminação detalhados no **subitem 7.1.3** e que esteja classificado entre os **660** (seiscentos e sessenta) mais bem colocados, por formação, respeitados os empates na última posição, incluindo nesse total os ocupantes das vagas destinadas a pessoas com deficiência em conformidade com o **subitem 5.1**.

Edital 2012

- **Discursiva:**

- 7.1.5

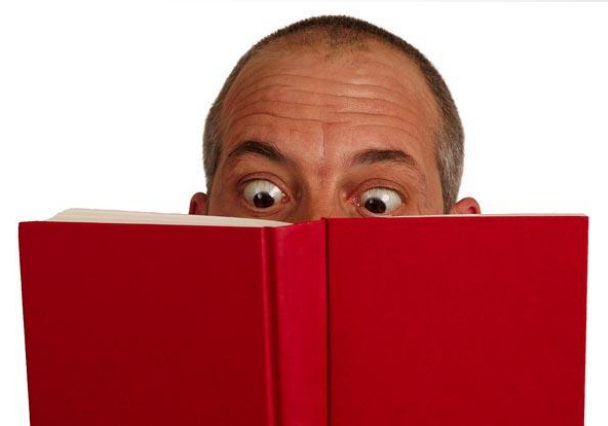
- 2ª Fase - prova discursiva – será composta de 5 (cinco) questões de **Conhecimentos Específicos** valendo, cada uma, 10 (dez) pontos, **totalizando 50,0** (cinquenta) pontos. Será **eliminado** o candidato que obtiver aproveitamento inferior a **60%** (sessenta por cento) do total da pontuação da prova discursiva.

- 7.1.6

- A prova discursiva destina-se a avaliar a capacidade de o candidato expor com clareza, concisão, precisão, coerência e objetividade assuntos ligados à formação acadêmica a que se candidatou. Levará, também, em conta a capacidade de organização do texto, de análise e síntese dos fatos e ideias examinados e, ainda, a correção gramatical com que foi redigida.

Conhecer o Edital

- Aspectos Técnicos (conteúdo);
- Aspectos Estruturais (Língua Portuguesa);
- Quantas serão corrigidas;
- Ordem de Desempate;
- Pode Rascunho? Lápis? Cor da caneta?
- Prova junta, separada?
- Onde irá almoçar?



Edital 2012

- **Desempate:**
 - 7.1.8.1 - Em caso de igualdade na nota final, para fins de classificação, na situação em que nenhum dos candidatos empatados possua idade igual ou superior a 60 (sessenta) anos, completados até o último dia de inscrição, o desempate se fará da seguinte forma:
 - 1º) maior pontuação na prova discursiva;
 - 2º) maior pontuação na prova objetiva de Conhecimentos Específicos;
 - 3º) maior pontuação na prova objetiva de Língua Portuguesa;
 - 4º) maior idade.

Edital 2012

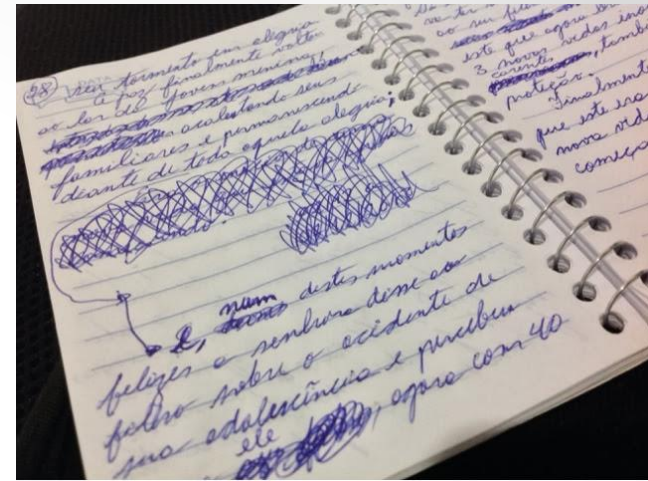
- 8.4 – O candidato deverá chegar ao local das provas com **uma hora de antecedência** do início das mesmas, munido de Cartão de Confirmação de Inscrição, recebido via postal ou impresso da página da FUNDAÇÃO CESGRANRIO na internet; do documento de identidade original com o qual se inscreveu e de caneta esferográfica de tinta preta fabricada em material transparente.

Edital 2012

- 8.9 – O candidato só poderá ausentar-se do recinto das provas após uma hora contada a partir do efetivo início das mesmas. Por motivos de segurança, o candidato não poderá levar o Caderno de Questões, a qualquer momento.

Escrita

- Legível;
- Diferenciar Maiúsculas e minúsculas;
- Preferencialmente letra cursiva;
- Evitar Falhas e Rasuras;
- Espaçamento do Texto;
- Número de Linhas



Estratégia



- Qual percentual representa a discursiva?
- Qual será o foco dos estudos?

FASE	TIPO DA PROVA	ASSUNTO	Nº DE QUESTÕES	PONTUAÇÃO		PERCENTUAL MÍNIMO PARA APROVAÇÃO	CARACTERÍSTICA
				MÁX.	MÍN.		
1ª	Objetivas	Língua Portuguesa	20	20	10	60%	Eliminatória e Classificatória
		Língua Estrangeira	10	10	03		
		Conhecimentos Específicos	40	70	35		
2ª	Discursiva	Conhecimentos Específicos	05	50	30	60%	Eliminatória e Classificatória

Tipos de Questões

- Questão Técnica (BNDES)
 - SQL / JAVA
 - Diretas e Objetivas
 - Tipo de Resposta: Diretas e Objetivas
- Parecer / Estudo de Caso / Relatório
 - Descrevem um cenário hipotético ou uma recomendação.
 - Tipo de Resposta: Deve apresentar problemas encontrados e abordar sugestões baseadas em normas, frameworks ou amparos técnicos para corrigir ou resolver a situação cobrada.

Espelhos de Correção

- CESGRANRIO – palavras-chave, avaliação do domínio do conteúdo prático. Respostas diretas, objetivas e depois detalhadas.
- Será considerado correto o código o query que obtiver o mesmo resultado do apresentado no espelho de correção.
- CESPE – palavras-chave, conhecimento do tema, escrita e uso de língua portuguesa;
- Erros são diluídos pela quantidade de linhas escritas.
- ESAF – Foco no português/gramática, capacidade de argumentação. Alinhamento do tema e estruturação
- FCC – avaliação do domínio do conteúdo prático.

Elementos Básicos do Texto

- Cobertura dos Tópicos:
 - Responder **TUDO** que foi pedido.
 - Utilizar palavras-chaves para identificar o que foi pedido.
 - Utilize termos originais propostos.
 - Qualquer argumentação é melhor do que nenhuma

Elementos Básicos do Texto

- Alinhar texto ao tema:
 - EVITE fuga do tema;
 - JAMAIS desvie do assunto;
 - Identifique tudo que foi pedido no seu texto;

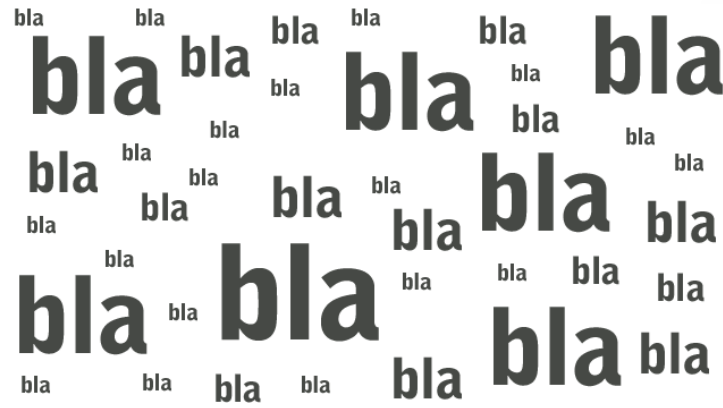
Elementos Básicos do Texto

- Sequenciamento das tarefas de acordo com o solicitado:
 - Siga o enunciado proposto utilizando conectores (assim, diante de, desta forma);
- FOCO
 - Vá direto ao ponto, mostre que entendeu o que está sendo pedido, e depois se aprofunde;

Elementos Básicos do Texto

- Equilíbrio:
 - Divida as atividades em quantidades de linhas semelhantes;
 - Balanceie os parágrafos;

- Não seja criativo!



Erros Comuns

- Português;
- Rasura (~~erro~~);
- Diagramação, parágrafos, separação entre páginas, separação de palavras;
- Acentuação, pontuação, trações em letras de forma incorreta.
- Espremer palavras;
- Extrapolar margem;
- Valorizar muito itens que domine mais;
- Marcações na prova;

Fluxo de Trabalho

- Análise do Enunciado:
 - Entenda o que está sendo pedido;
 - Separe as palavras-chaves;
 - Sequencie as informações;
 - Reescreva os itens sob a forma de uma pergunta direta;

Fluxo de Trabalho

- Brainstorming Solitário:
 - Coloque no papel todas as informações que você possui;
 - Utilize a própria prova para compor as frases;
 - Identifique as palavras que possam identificar os tópicos cobrados;

RUP
it*.me.*
fase*.disrup.*
boas praticas
evolucionarias



Fluxo de Trabalho

- Esqueleto:
 - Distribua de forma igualitária;
 - Divida a quantidade de linhas por tópico;
 - Margem de erro (quantidade de linhas);

3

10 linhas

30



Rascunho

- Resgate o termo rascunho para iniciar seus tópicos;
- Cada item deverá estar em um tópico separado;
- Siglas – Identifique-as na primeira vez;
- Aspas para termos em inglês;



Texto Final

- Dê um descanso entre o rascunho e o texto final;
- Texto deverá ser uma cópia do rascunho;

Na Hora do Desespero

- Fale sobre fundamentos e características;
- Mostre riqueza vocabular (parcimônia);
- Expandir siglas conhecidas;
- Reutilizar trechos do enunciado;



Dicas

- Controle o tempo;
- Leve 3 canetas;
- Leve um lanche;
- Planeje o local de almoço antecipadamente,
- Durma cedo;



Recursos

- Não conteste critérios de correção;
- Cite linhas;
- Cite referências associadas às linhas;
- Citar a proporcionalidade entre a pontuação recebida x o que foi feito.

Dicas de Recursos

- Algumas situações a vitória do recurso é mais uma questão de sorte do que de argumento;
- Sempre deixe claro o que está sendo solicitado;
- Não seja agressivo;
- Não use caixa alta;
- Não trate o examinador como um ignorante;
- Evitar qualquer dificuldade que faça com que o avaliador tenha que ter o mínimo de trabalho;
- Utilize linguagem ‘humilde’;
- Indique precisamente todas as linhas citadas;
- Evite palavras difíceis ou jargões;

Exemplo de Recursos

Esta requerente, baseada nos argumentos abaixo, pede que a banca reavalie sua posição inicial dada ao quesito 2.3.

Nas linhas 27 a 30 duas formas de automação de processos são citadas: workflow e BPEL. De acordo com a definição do autor Ian Sommerville, na obra "Software Engineering", nona edição, especificamente na página 527, último parágrafo da página, workflow é definido como:

"The sequence of steps is called a workflow – a set of activities ordered in time."

Desta forma, a incorporação deste conceito em um sistema de informação possibilita a automação de um processo de negócio (citado nas linhas 24 e 25), utilizando como fonte os requisitos de negócios, conforme apresentado nas linhas 25 a 27.

Ainda segundo o autor Ian Sommerville, precisamente no capítulo 19, página 510, item enumerado 3, o autor define BPEL como sendo:

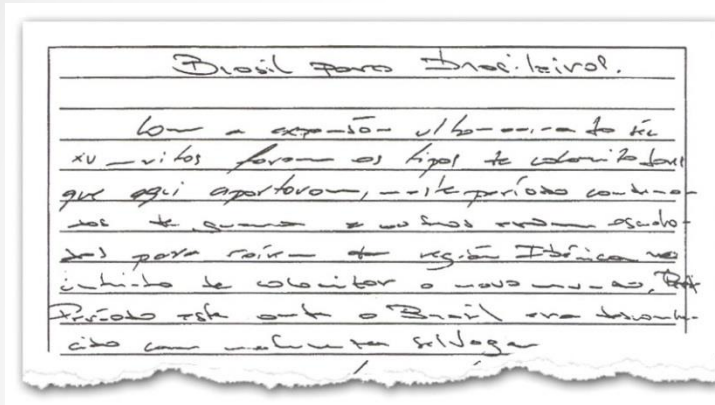
"standard for a workflow language that is used to define process programs involving several different services." o que é utilizado como argumento para a utilização do BPEL na automação de processos nas linhas 29 e 30.

De acordo com Philippe Kruchten, no livro Introdução ao RUP, segunda edição, na página 117, primeiro item da "Proposta" explicita que o fluxo Modelagem de Negócio visa "Entender a estrutura e a dinâmica da organização na qual um sistema será distribuído".

Ainda nas linhas 20 a 22 foi utilizado o argumento de utilização do RUP, sob a justificativa da disciplina de modelagem de negócio entender o cenário de negócio no qual o software será inserido e servir de base para a disciplina de modelagem de requisitos que utiliza casos de uso para especificar as interações entre usuário e sistema e assim obter os requisitos, conforme indicado nas linhas 20 a 22 e poder construir os workflows, citado na linha 28.

Diante da similaridade dos argumentos da recorrente com as obras citadas, solicito-vos uma reavaliação do presente item.

Jogo dos Erros



a) fuso horário de Pequim: +8 horas
fuso horário de Brasília (hora oficial do Brasil): -3 horas
diferença de horários: $-3 - 8 = -11$ horas
Como não está horário de verão em agosto, o horário do Brasil será 13 horas ~~mais cedo~~ do que o de Pequim, ~~pois~~ portanto, no Brasil será 07h00 min do dia 8 de agosto de 2008.

b) O ~~pr~~ fuso horário por acionadores de forma que a cada 15° haja 1 hora de diferença, pois a esfera por ser 360° e o dia 24 horas, $360 \div 24 = 15$, quando não há decisões políticas de fazer interferências na determinação do horário.

04) Qual a função do apóstrofo?

Apóstrofos são os amigos de Jesus, que se juntaram naquela partinha que Michelângelo fotografou.



Análise de Respostas





PROVA DISCURSIVA - FOLHA DE TEXTO DEFINITIVO

1 O fato de uma empresa gastar muito em TI
2 não é sinônimo de empresa competitiva.

3 Mesmo uma empresa global que invista altas
4 cifras em sua estrutura tecnológica, com todos os
5 índices de desempenho e relatórios cada vez mais
6 detalhados e mesmo que se tenha implementado um
7 modelo de gestão consagrado no mercado, ainda pode
8 ser que seja "esmagada" pelos concorrentes.

9 Há alguns anos atrás, na governança de TI, na
10 figura de seu gestor, a especialização técnica desse
11 gestor era praticamente suficiente para gerir a TI de uma
12 empresa. No entanto, vi no banco Itaú, onde
13 trabalhei, que os gestores de TI, além de suas especia-
14 lizações técnicas, o envolvimento com os negócios, estraté-
15 gias e planos futuros era cada vez mais nítido.

16 O desafio na governança de TI: devemos estar
17 preparados e devemos saber interpretar os padrões e tendências
18 na TI e nos negócios, agindo rapidamente e com
19 precisão, a um custo cada vez menor para a
20 empresa.

Questão Discursiva 03

19854930

78244161

13974295

O SMTP, protocolo que atua na porta 25, apresenta algumas falhas de segurança já conhecidas, como por exemplo a não comprovação da autenticidade do emissor e a não possibilidade de comprovação da integridade da informação.

a) estas falhas afetam a autenticidade e integridade da informação.

b) no serviço de correio estas falhas citadas não causam impacto.

c) utilização de funções hash para garantir a integridade e utilização de criptografia assimétrica para garantir a autenticidade.





Questão nº 1

Na construção de um sistema de software, a satisfação do usuário é alcançada quando o serviço ou produto desenvolvido está de acordo com as suas necessidades. Dentre os vários fatores diretamente relacionados à qualidade do software, tem-se a extensibilidade.

a) Explique, caracterizando, quando um produto de software apresenta extensibilidade.

(valor: 4,0 pontos)

b) O que deve ser considerado na construção de um produto de software para que a extensibilidade seja alcançada?

(valor: 6,0 pontos)

- **Análise do Enunciado:**
 - Entenda o que está sendo pedido;
 - Separe as palavras-chaves;
 - Sequencie as informações;
 - Coloque no papel todas as informações que você possui;
 - Utilize a própria prova para compor as frases;
 - Identifique as palavras que possam identificar os tópicos cobrados;
 - Resgate o termo enunciado para iniciar seus tópicos;
 - Cada item deverá estar em um tópico separado;

Questão nº 1

Na construção de um sistema de software, a satisfação do usuário é alcançada quando o serviço ou produto desenvolvido está de acordo com as suas necessidades. Dentre os vários fatores diretamente relacionados à qualidade do software, tem-se a extensibilidade.

a) Explique, caracterizando, quando um produto de software apresenta extensibilidade.

(valor: 4,0 pontos)

Fácil de mudar | modularizado | adicionar componentes
sem afetar o que já existe | Contratos/interfaces | adaptável
| flexibilidade | fácil acoplamento | mudanças não
causem erros

Questão nº 1

Na construção de um sistema de software, a satisfação do usuário é alcançada quando o serviço ou produto desenvolvido está de acordo com as suas necessidades. Dentre os vários fatores diretamente relacionados à qualidade do software, tem-se a extensibilidade.

a) Explique, caracterizando, quando um produto de software apresenta extensibilidade.

(valor: 4,0 pontos)

Um produto de software apresenta extensibilidade quando novas funcionalidades podem ser acrescentadas sem afetar o que já existe, facilitando as mudanças necessárias através de contratos/interfaces de serviços oferecidos. Adaptabilidade e flexibilidade são características de um software que apresente extensibilidade. Se um software possui baixo acoplamento, seus componentes serão modularizados, impedindo que uma mudança possa causar efeitos colaterais indesejados

Questão nº 1

Na construção de um sistema de software, a satisfação do usuário é alcançada quando o serviço ou produto desenvolvido está de acordo com as suas necessidades. Dentre os vários fatores diretamente relacionados à qualidade do software, tem-se a extensibilidade.

a) Explique, caracterizando, quando um produto de software apresenta extensibilidade. (valor: 4,0 pontos)

Um produto de software apresenta extensibilidade quando novas funcionalidades podem ser acrescentadas sem afetar o que já existe, facilitando as mudanças necessárias através de contratos/interfaces de serviços oferecidos. Adaptabilidade e flexibilidade são características de um software que apresente extensibilidade. Se um software possui baixo acoplamento, seus componentes serão modularizados, impedindo que uma mudança possa causar efeitos colaterais indesejados

Espelho de Resposta:

a) Um produto de software apresenta extensibilidade quando é flexível, capaz de ser alterado sem dificuldades, permitindo a introdução de novas rotinas, novos módulos e o enxugamento de seus processos sem gerar resultados indesejáveis, como bugs ou desvio de conduta.

(valor: 4,0 pontos)

b) O que deve ser considerado na construção de um produto de software para que a extensibilidade seja alcançada?

(valor: 6,0 pontos)

RASCUNHO

Simplicidade | coesão | Acoplamento | Modularização |
Fronteira dos serviços | evit^oar efeitos colaterais

b) O que deve ser considerado na construção de um produto de software para que a extensibilidade seja alcançada?

(valor: 6,0 pontos)

Para que a extensibilidade seja alcançada na construção de um software deverá ser levada em consideração a simplicidade no desenvolvimento e codificação do produto. Este deverá ser construído segundo os preceitos da alta coesão e baixo acoplamento, possibilitando assim uma maior modularização de seus componentes de forma que possua uma definição clara de seus serviços, expostos através de contratos por suas interfaces, delimitando assim o que poderá ser oferecido e impedindo que futuras mudanças possam provocar efeitos colaterais ou comportamento indesejado quando da modificação ou adição de novas funcionalidades.

b) O que deve ser considerado na construção de um produto de software para que a extensibilidade seja alcançada?

(valor: 6,0 pontos)

*Para que a extensibilidade seja alcançada na construção de um software deverá ser levada em consideração a **simplicidade** no desenvolvimento e codificação do produto. Este deverá ser construído segundo os preceitos da **alta coesão e baixo acoplamento**, possibilitando assim uma maior modularização de seus componentes de forma que possua uma definição clara de seus serviços, expostos através de contratos por suas interfaces, delimitando assim o que poderá ser oferecido e impedindo que futuras mudanças possam provocar **efeitos colaterais** quando da modificação ou adição de novas funcionalidades.*

b) Para que a extensibilidade do produto de software seja alcançada, deve-se considerar:

- a **simplicidade** de compreensão da arquitetura – a qual possibilita a realização das modificações que se fizerem necessárias após sua concepção.
- a **descentralização** de seus componentes – a **independência** dos diferentes componentes de software é fundamental para que, em caso de sua modificação ou exclusão, não **aconteça um conjunto de reações sucessivas, efeito dominó**, modificando todo o funcionamento do sistema e originando bugs ou erros antes não existentes.

(valor: 6,0 pontos)



BNDES - Ano: 2008 - Banca: CESGRANRIO

A estimativa de um sistema de informação hipotético, com base na análise de pontos de função, determinou que serão necessárias(os) para construção desse sistema: * 5 EE (Entrada Externa) * 6 SE (Saída Externa) * 4 CE (Consulta Externa) * 3 ALI (Arquivo Lógico Interno) * 7 AIE (Arquivo de Interface Externa), Todas as funções foram classificadas como sendo de complexidade baixa (nenhuma média e nenhuma alta). Considerando-se a tabela IFPUG abaixo e que o nível de influência geral é igual a 30, responda à pergunta a seguir. Qual a quantidade total de pontos de função não ajustados e ajustados do sistema? Indique os cálculos necessários.

Tipo de Função	Baixa	Média	Alta
EE	3	4	6
SE	4	5	7
CE	3	4	6
ALI	7	10	15
AIE	5	7	10

tabela IFPUG: pontos de função atribuídos a cada tipo de função

BNDES - Ano: 2008 - Banca: CESGRANRIO

A estimativa de um sistema de informação hipotético, com base na análise de pontos de função, determinou que serão necessárias(os) para construção desse sistema: * 5 EE (Entrada Externa) * 6 SE (Saída Externa) * 4 CE (Consulta Externa) * 3 ALI (Arquivo Lógico Interno) * 7 AIE (Arquivo de Interface Externa), Todas as funções foram classificadas como sendo de complexidade baixa (nenhuma média e nenhuma alta). Considerando-se a tabela IFPUG abaixo e que o nível de influência geral é igual a 30, responda à pergunta a seguir. Qual a quantidade total de pontos de função não ajustados e ajustados do sistema? Indique os cálculos necessários.

$$5 \times 3 + 6 \times 4 + 4 \times 3 + 3 \times 7 + 7 \times 5 \Rightarrow 15 + 24 + 12 + 21 + 35 \Rightarrow 107 \text{ PFNA}$$

$$VA = ((ni \times 0,01) + 0,65) \Rightarrow (30 \times 0,01) + 0,65 \Rightarrow 0,30 + 0,65 \Rightarrow 0,95 \text{ VA}$$

$$PFA = PFNA \times VA \Rightarrow 107 \times 0,95 \Rightarrow 101,65 \text{ PFA}$$

BNDES - Ano: 2008 - Banca: CESGRANRIO

ESPELHO

- Resposta: Para calcular o total, multiplica-se a quantidade de ocorrência de cada função pelo número de pontos de função não ajustados atribuídos segundo IFPUG. Soma-se então cada parcela multiplicada.
- Total de pontos de função não ajustados = $5*3 + 6*4 + 4*3 + 3*7 + 7*5 = 107$
(valor: 5,0 pontos)
- Nível de Influência Geral = 30
- Pontos de Função não ajustados = 107
- Valor do Fator de Ajuste = $(\text{Nível de Influência Geral} * 0,01) + 0,65 = (30 * 0,01) + 0,65 = 0,95$.
- Total de Pontos ajustados = pontos de função não ajustados * valor do fator de ajuste =
- $107 * 0,95 = 101,65 = 102$, aproximadamente.
(valor: 5,0 pontos)

BNDES - Ano: 2008 - Banca: CESGRANRIO

Questão nº 1

O ciclo de vida do software abrange uma especificação de alto nível para um grupo de atividades que constituem o processo de desenvolvimento de software, e se divide em modelos, entre os quais se incluem os modelos de Protótipo e de Desenvolvimento Automatizado.

- a) Descreva quais são os propósitos de um Protótipo, do ponto de vista do modelador e do ponto de vista do usuário.
(valor: 5,0 pontos)

RASCUNHO

BNDES - Ano: 2008 - Banca: CESGRANRIO

Questão nº 1

O ciclo de vida do software abrange uma especificação de alto nível para um grupo de atividades que constituem o processo de desenvolvimento de software, e se divide em modelos, entre os quais se incluem os modelos de Protótipo e de Desenvolvimento Automatizado.

- a) Descreva quais são os propósitos de um Protótipo, do ponto de vista do modelador e do ponto de vista do usuário.
(valor: 5,0 pontos)

Protótipo descartável e evolucionário

Modelador: Validar | evoluir | tornar clara | confirmar |
feedback | descobrir | modelagem | entender

Usuário: Entender | Facilidade | usabilidade

Pra que serve? Evitar erros posteriores. Garantir o que está sendo
feito.

RASCUNHO

BNDES - Ano: 2008 - Banca: CESGRANRIO

Questão nº 1

O ciclo de vida do software abrange uma especificação de alto nível para um grupo de atividades que constituem o processo de desenvolvimento de software, e se divide em modelos, entre os quais se incluem os modelos de Protótipo e de Desenvolvimento Automatizado.

- a) Descreva quais são os propósitos de um Protótipo, do ponto de vista do modelador e do ponto de vista do usuário.
(valor: 5,0 pontos)

Sob o ponto de vista do modelador de um sistema, o protótipo apresenta como propósito a validação de uma funcionalidade ou requisito que não esteja muito claro, possibilitando um entendimento maior sobre o que deverá ser feito antes que possa ser construído e possa refletir algo não esperado. Funciona como uma garantia de que o que estiver sendo feito é o que realmente foi solicitado. Sob o ponto de vista do usuário do sistema o protótipo apresenta outros propósitos, como por exemplo verificar se o que está sendo produzido pela equipe de construção terá utilidade para o que o usuário espera, bem como se terá a usabilidade esperada. Algumas vezes o protótipo irá permitir que tanto o modelador quanto o usuário tenham um maior entendimento.

BNDES - Ano: 2008 - Banca: CESGRANRIO

Questão nº 1

O ciclo de vida do software abrange uma especificação de alto nível para um grupo de atividades que constituem o processo de desenvolvimento de software, e se divide em modelos, entre os quais se incluem os modelos de Protótipo e de Desenvolvimento Automatizado.

- a) Descreva quais são os propósitos de um Protótipo, do ponto de vista do modelador e do ponto de vista do usuário.
(valor: 5,0 pontos)

Sob o ponto de vista do modelador de um sistema, o protótipo apresenta como propósito a validação de uma funcionalidade ou requisito que não esteja muito claro, possibilitando um entendimento maior sobre o que deverá ser feito antes que possa ser construído e possa refletir algo não esperado. Funciona como uma garantia de que o que estiver sendo feito é o que realmente foi solicitado. Sob o ponto de vista do usuário do sistema o protótipo apresenta outros propósitos, como por exemplo verificar se o que está sendo produzido pela equipe de construção terá utilidade para o que o usuário espera, bem como se terá a usabilidade esperada. Algumas vezes o protótipo irá permitir que tanto o modelador quanto o usuário tenham um maior entendimento.

BNDES - Ano: 2008 - Banca: CESGRANRIO

-Resposta:a) (valor: 5,0 pontos)

-1- propósito do ponto de vista do modelador: Fazer com que a estrutura conceitual de uma especificação de **requisitos seja mais concreta**, possibilitando, dessa forma, **verificar a funcionalidade dessa especificação** quanto à sua **consistência, correção, validade e à facilidade** de uso do produto real que será gerado.

-2-propósito do ponto de vista do usuário:Ser o instrumento que irá **validar** os aspectos funcionais que o software em construção deverá possuir, de forma a **atender as necessidades** de seu usuário e garantir que a interface humana trará **simplicidade** à sua operação.

BNDES - Ano: 2008 - Banca: CESGRANRIO

b) Cite duas ideias básicas do modelo de Desenvolvimento Automatizado e o que se considera nessas ideias.

(valor: 5,0 pontos)

RASCUNHO

BNDES - Ano: 2008 - Banca: CESGRANRIO

b) Cite duas ideias básicas do modelo de Desenvolvimento Automatizado e o que se considera nessas ideias.

(valor: 5,0 pontos)

Geração automática de código | modelagem de negócio |
definição de requisitos | manutenção é feita diretamente
no requisito |

áreas gestoras são responsáveis pelo negócio e não mais a TI
Consistência do modelo com a regra de negócio existente

BNDES - Ano: 2008 - Banca: CESGRANRIO

b) Cite duas ideias básicas do modelo de Desenvolvimento Automatizado e o que se considera nessas ideias.

(valor: 5,0 pontos)

Uma idéia muito importante do modelo de desenvolvimento automatizado é a geração automática do código. Esta idéia considera a geração do código fonte de um programa a partir da modelagem de negócio, e a responsabilidade sobre a definição das regras de negócio e da real forma de funcionamento sobre os requisitos é da área gestora e não mais da área de Tecnologia da Informação. Outra idéia é a consistência do modelo com a regra de negócio existente, uma vez que o código irá funcionar de acordo com a modelagem, caso a regra de negócio seja alterada então a mudança refletirá no código fonte do sistema que está sendo desenvolvido, uma vez que a alteração da funcionalidade será feita diretamente sobre o requisito.

BNDES - Ano: 2008 - Banca: CESGRANRIO

- b) (valor: 5,0 pontos) O candidato deverá apresentar duas das seguintes ideias básicas do modelo de Desenvolvimento Automatizado, com suas respectivas considerações:
- **Manutenção da Especificação de Requisitos ao invés do Código Fonte** – considera-se que a manutenção da especificação de requisitos modifica a forma mais próxima do modelo conceitual do usuário, que é mais facilmente alterável do que o código fonte.
- **Usuário como Analista de Sistemas** – considera-se que, como o nível de especificação de requisitos pode ser entendido pelo usuário, ou seja, que ele tem capacidade para tal entendimento, ele deve ser capaz de manipular essa especificação e, desde que haja uma linguagem de representação adequada, ser o provedor de sua manutenção.
- **Prototipação** – considera-se que um modo de validar a especificação de requisitos deve ser assegurado sem a necessidade de implementar o sistema para tal validação, ou seja, esse modo é a construção de um protótipo – uma especificação executável, mesmo que de forma ineficiente. **Assistente Automático** – considera-se a existência de um assistente com a função de gerar de forma automática o código fonte do sistema, partindo de uma especificação formal de requisitos, somada a um conjunto de decisões de otimização associadas a uma estratégia de desenvolvimento incremental acoplada à prototipação no nível da especificação de requisitos, à cooptação do usuário e à garantia de que será executada, automaticamente, parte significativa do processo.



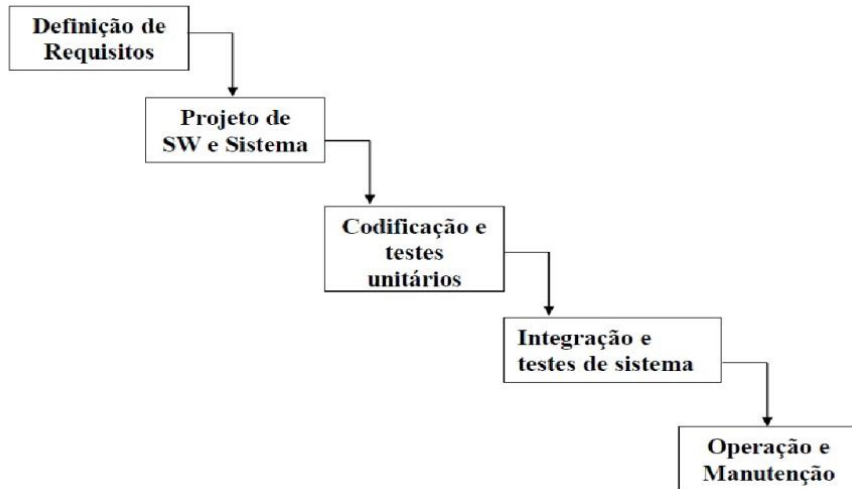
2011

FCC

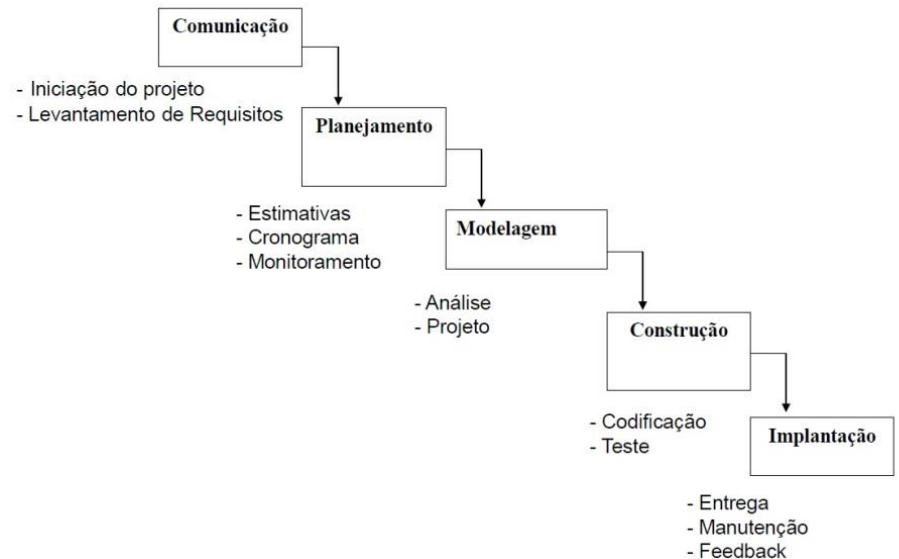
- No campo da Engenharia de Software, orientação a objetos e Governança de TI, analise e responda:
- 1. Com respeito ao modelo em cascata, descreva sucintamente as fases e os objetivos de cada uma delas;
- 2. Com respeito ao desenvolvimento em espiral, escreva a principal diferença entre este e os outros modelos do processo de software;
- 3. No modelo em espiral, cada loop na espiral está dividido em quatro setores. Descreva sucintamente o nome e o objetivo de apenas dois quaisquer destes setores, conforme definido por Barry Boehm

1. Com respeito ao modelo em cascata, descreva sucintamente as fases e os objetivos de cada uma delas;

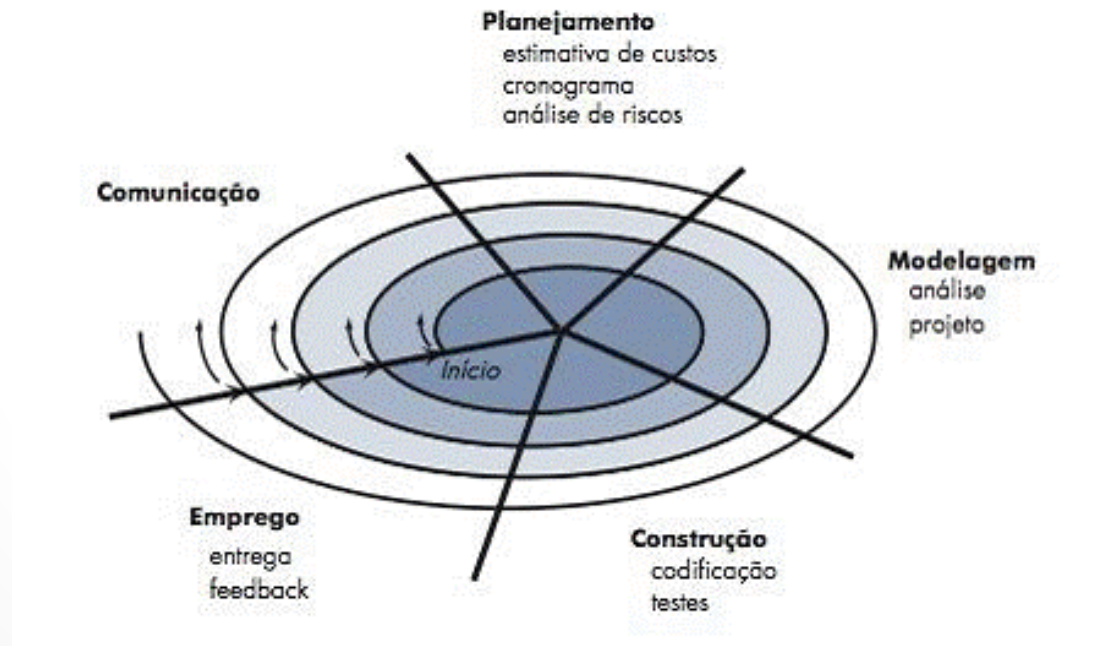
Sommerville



Pressman



2. Com respeito ao desenvolvimento em espiral, escreva a principal diferença entre este e os outros modelos do processo de software;
3. No modelo em espiral, cada loop na espiral está dividido em quatro setores. Descreva sucintamente o nome e o objetivo de apenas dois quaisquer destes setores, conforme definido por Barry Boehm





2012



BRDE - Ano: 2012 - Banca: AOCP -

De acordo com o autor Ian Sommerville, possuímos alguns modelos de processo de software, ainda de acordo com esse autor, descreva as atividades fundamentais do modelo em cascata.

BRDE - Ano: 2012 - Banca: AOCP -

De acordo com o autor Ian Sommerville, possuímos alguns modelos de processo de software, ainda de acordo com esse autor, descreva as atividades fundamentais do modelo em cascata.

Pressman :



Sommerville :



BRDE - Ano: 2012 - Banca: AOCP -

De acordo com o autor Ian Sommerville, possuímos alguns modelos de processo de software, ainda de acordo com esse autor, descreva as atividades fundamentais do modelo em cascata.

Definição de Requisitos: Levantamento de necessidades;

Projeto de SW e Sistemas: Pensar no problema e na solução deste;

Codificação e Testes Unitários: Desenvolvimento e testes necessários para o correto

funcionamento do sistema;

Integração e Testes de Sistemas: Compor os módulos do sistema e simular o funcionamento do sistema na forma que seria executado;

Operação e Manutenção: Entrega do sistema e acompanhamento sobre o funcionamento e evolução do software.



2012

Medicina - Ano: 2012 - Banca: FUNIVERSA -

As ferramentas CASE (Computer Aided Software Engineering) possuem um enorme potencial se usadas corretamente. Com o aumento de produtividade resultante do uso dessas ferramentas, o mercado de desenvolvimento de software começou a demandar cada vez mais ferramentas de apoio ao processo de produção, o que acarretou o surgimento de diversas opções limitadas e com problemas de integração. Considerando essas informações, redija um texto dissertativo que aborde, necessariamente, os seguintes tópicos:

- a) conceito das ferramentas CASE e uma categorização com os seus tipos;
- b) vantagens e desvantagens do uso das ferramentas CASE no desenvolvimento; e
- c) importância do uso dessas ferramentas para o desenvolvimento de sistemas.

Ferramentas Case

- Ferramentas que auxiliam o engenheiro de SW em cada atividade associada ao desenvolvimento de SW;
- Reduzem o esforço necessário para produzir artefatos e alcançar metas;
- Aumentam a qualidade do software;
- Devem ser utilizadas em conjunto com o modelo adotado.

Ferramentas Case não substituem uma metodologia de desenvolvimento de software.

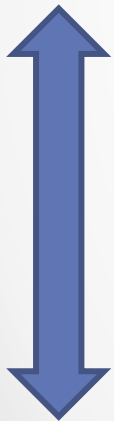
Categorização

❑ Horizontais:



- Utilizadas durante todo o processo de desenvolvimento;

Verticais:



- Utilizadas especificamente em algumas disciplinas de software;

Categorizaça

lowercase

UPPERCASE

MiXeDcAsE

<http://www.computerhope.com>

- Front-End ou Upper CASE
 - Apoiam as etapas iniciais da criação de sistemas: planejamento, análise e projeto;
- Back-End ou Lower CASE
 - Apoio à parte física: código, testes e manutenção;
- I-CASE ou Integrated CASE
 - Todo o ciclo de vida

Categorização

- ❑ **Ferramenta** - é um produto de software que auxilia em uma ou mais tarefas específicas de uma ou mais fases do processo (ciclo de vida) de desenvolvimento de software (ex: compilação, checagem de consistência de um projeto, edição de texto, etc.);
- ❑ **Workbench** - conjunto de ferramentas integradas que suporta uma ou mais fases do processo de desenvolvimento de software (ex: especificação, projeto, implementação, etc.).
- ❑ **Ambiente** - suporta todo ou uma grande parte do processo de desenvolvimento de software. Geralmente é um conjunto de workbenches integrados.
- ❑ **Meta-CASE Workbenches** - são sistemas que dão apoio ao processo de criação de outros workbenches.

Vantagens e Desvantagens



- Fazer a aplicação geral da engenharia de software;
Tornar prática a prototipação, acelerando o desenvolvimento e sistemas junto aos usuários;



- Agilizar o processo de desenvolvimento;
- Deixar disponível o reuso de componentes do sistemas;
- Documentação do sistema gerado pela própria ferramenta;
- Necessidade de uma mudança de atitude profissional de desenvolvimento de sistemas e aprendizado de novas metodologias que as ferramentas utilizam;



- Permite a construção de novos sistemas que atendam a todos os processos;
- Redução de custo da manutenção de sistemas.



Incompatibilidade de ferramentas;
Treinamento para utilização.



Medicina - Ano: 2012 - Banca: FUNIVERSA -

As ferramentas CASE (Computer Aided Software Engineering) possuem um enorme potencial se usadas corretamente. Com o aumento de produtividade resultante do uso dessas ferramentas, o mercado de desenvolvimento de software começou a demandar cada vez mais ferramentas de apoio ao processo de produção, o que acarretou o surgimento de diversas opções limitadas e com problemas de integração. Considerando essas informações, redija um texto dissertativo que aborde, necessariamente, os seguintes tópicos:

- a) conceito das ferramentas CASE e uma categorização com os seus tipos;
- b) vantagens e desvantagens do uso das ferramentas CASE no desenvolvimento; e
- c) importância do uso dessas ferramentas para o desenvolvimento de sistemas.



**SENADO
FEDERAL**

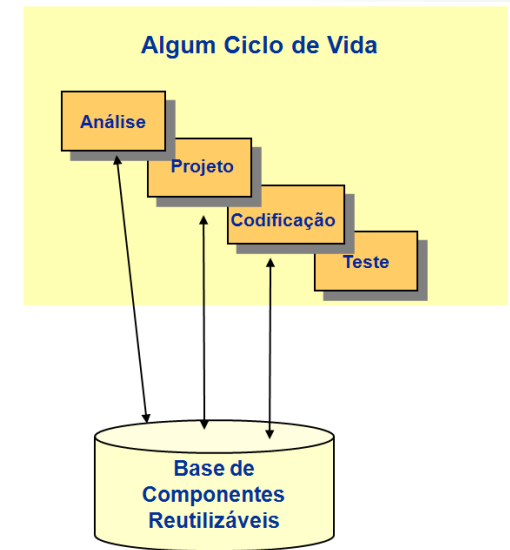


Senado Federal - Ano: 2008 - Banca: FGV -

Descreva o que é um desenvolvimento baseado em componentes e a relação dessa abordagem com o reúso de software.

Desenvolvimento Baseado em Componentes

- Enfatiza a reutilização de componentes
- Evolucionário
- Utiliza fortemente UML
- Vantagens:
 - Aumento da produtividade
 - Diminuição do ciclo de desenvolvimento
 - Aumento da qualidade



Desenvolvimento Baseado em Componentes

- *Surgiu como uma nova perspectiva para o desenvolvimento de software, utilizando ao invés de blocos monolíticos, componentes interoperáveis, reduzindo a complexidade tanto no desenvolvimento quanto nos custos, através da reutilização de componentes. O software passa a ter componentes com menor grau de dependência, ou seja, um fraco acoplamento, o que os torna substituíveis, reutilizáveis e interoperáveis. A utilização desta abordagem visa trazer a reusabilidade dos componentes, sendo desenvolvido para que possa ser utilizado, substituído ou modificado com uma redução dos efeitos colaterais possíveis.*



Disciplina: TI - Concurso: Tribunal Regional

Eleitoral - ES - Ano: 2012 - Banca: CESPE -

Na concepção de um software planejado, devem ser realizadas estimativas de esforço, prazo (cronograma) e custo (orçamento), relativas ao desenvolvimento e à manutenção do software. Erros nessa fase (de sub ou de superestimação) podem afetar seriamente a gestão do projeto e causar transtornos à organização no que se refere a recursos alocados, tempo e orçamento, ou, mesmo, tornar o projeto inviável. Uma estimativa importante nesse contexto é a do tamanho do software objeto do projeto, que pode ser obtida pela técnica de análise de pontos de função (APF) é uma das principais técnicas utilizadas para se obter essa estimativa. Em face dessas informações, que têm caráter meramente motivador, redija um texto dissertativo acerca da técnica APF, desenvolvendo, necessariamente, os seguintes tópicos:

- 1 – definição, características e objetivos da APF;
- 2 – descrição sucinta do processo de contagem de pontos de função;
- 3 – classificação das funções quanto ao tipo de funcionalidades proporcionadas ao usuário do software.

Métricas Orientadas a Função

- Análise de Pontos de Função (APF):
- Padrão mundial para medição funcional de um software;
- Independente de tecnologia;
- Mede as funcionalidades fornecidas do ponto de vista do usuário; (tamanho funcional)
- Medida da complexidade ou funcionalidade do software;
- Não mede diretamente esforço, custo ou produtividade (junto com outras variáveis poderia)
- Objetivo Principal: Medir a funcionalidade solicitada pelo usuário, de forma a estimar seu tamanho e seu custo.

3. Medir funções de Transação

- 3.1 Identificar cada processo elementar;
- 3.2 Classificar cada processo elementar (EE, SE ou CE)
- 3.3 Contar ALRs e DERs
- 3.4 Determinar a complexidade funcional de cada função de transação
- 3.5 Determinar o tamanho funcional de cada função de transação

3. Medir funções de Transação

3.1 Identificar cada processo elementar;

Processo Elementar é a menor unidade de atividade significativa para o usuário;

Exemplo: Incluir um Funcionário (deve ser uma transação completa, incluindo todas as informações necessárias para que a aplicação fique em um estado consistente)

Funções Transacionais

Funções transacionais representam as funcionalidades de processamento de dados do sistema fornecidas para o usuário;

- Entradas Externas (EEs)
 - Processos elementares que processam dados que entram pela fronteira da aplicação;
 - Objetivo de uma EE é manter um ou mais ALIs ou alterar o comportamento do sistema;
- Saídas Externas (SEs)
 - Processos elementares que enviam dados para fora da fronteira da aplicação;
 - Envolve cálculos ou criação de dados derivados, e não uma simples recuperação de dados;
- Consulta Externa (CEs)
 - Processo elementar que envia dados para fora da fronteira;
 - Não realiza nenhum cálculo e nem cria dados derivados;
 - Objetivo é apresentar informação por meio da recuperação simples;

Funções Transacionais

- Arquivo Lógico Referenciado (ALR)
 - ALI ou AIE que foi referenciado por uma função de transação;
 - Quanto mais ALRs maior a complexidade da minha transação;
- Dado Elementar Referenciado (DER)
 - Dados que atravessam a fronteira da aplicação;
 - Por exemplo: Atualização de cliente, envia o nome e cpf. Nome e CPF serão dados elementares referenciados;
 - Quanto mais DERs existirem, maior a complexidade da transação;

Entradas Externas

- Dados de transações utilizados na manutenção dos ALI alimentados através de arquivos;
- Tela de Entrada online. Caso existam opções de alteração, exclusão e inclusão, deverá ser considerada como três entradas externas.
- Tela de Entrada Batch. Entradas duplicadas serão computadas como uma única;

Saídas Externas

- Dados transferidos para outra aplicação. Processados e formatados para uso por outra aplicação;
- Relatórios;
- Relatórios Duplicados, porém, produzidos em diferentes meios para necessidades específicas devem ser consideradas saídas externas distintas, desde que a multiplicidade seja provida pela aplicação.

Consultas Externas

- Consultas implícitas: telas de alteração ou remoção que mostram o que será alterado ou removido antes da ação efetiva;
- Menus com consultas implícitas que além de permitir funcionalidade de seleção e navegação, permita informar parâmetros para consulta na tela;
- Telas de seleção de relatórios que permitam informar parâmetros para o relatório;
- Telas de Help;

Complexidade de Funções de Transação

Tabela 6 — Complexidade funcional das EE

		DERs		
		1 – 4	5 – 15	> 15
ALRs	0 – 1	Baixa	Baixa	Média
	2	Baixa	Média	Alta
	> 2	Média	Alta	Alta

Tabela 7 — Complexidade funcional das SE e CE

		DERs		
		1 – 5	6 – 19	> 19
ALRs	0 – 1	Baixa	Baixa	Média
	2 – 3	Baixa	Média	Alta
	> 3	Média	Alta	Alta

NOTA Uma CE tem no mínimo 1 ALR.

Tamanho Funcional - Transação

Tabela de Contribuição (EE, SE e CE)			
Funcionalidade	Baixa	Média	Alta
EE	3	4	6
SE	4	5	7
CE	3	4	6

4. Medir funções de Dados

- 4.1 Identificar funções de dados (ALI e AIE);
- 4.2 Contar DERs e RLRs para cada função de dados;
- 4.3 Determinar a complexidade funcional de cada função de dados;
- 4.4 Determinar o tamanho funcional de cada função de dados;

Funções de Dados

Funções de dados representam as funcionalidades relativas aos requisitos de dados internos e externos à aplicação.

- Arquivo Lógico Interno (ALI)
 - Mantido dentro da fronteira da aplicação;
 - Principal intenção é a de armazenar dados mantidos pela aplicação;
- Arquivo de Interface Externa (AIE)
 - Apenas referenciado pela aplicação, ou seja, mantido dentro da fronteira de outra aplicação;
 - Principal intenção é armazenar dados referenciados pela aplicação;
 - Um AIE sempre será um ALI em outra aplicação;

Funções de Dados

- Dado Elementar Referenciado (DER)
 - Atributo único, não repetido;
 - Análogo a campos de uma tabela ou atributos de um objeto;
 - Quanto mais DER, maior a complexidade;
- Registro Lógico Referenciado (RLR)
 - Subgrupo de dados elementares referenciados;
 - Por padrão toda função de dados tem um RLR, mas o usuário pode utilizar como mais de um subgrupo de dados;

Arquivo Lógico Interno

- Dados da aplicação (cadastros)
- Arquivos de dados de segurança
- Arquivos de dados de auditoria
- Arquivos de mensagens de auxílio
- Arquivos de mensagens de erro
- Arquivo de backup (apena se especificamente solicitado pelo usuário para atender requisitos da aplicação)

Arquivo Interface Externa

- Dados de referência (dados externos utilizados pela aplicação)
- Arquivos de mensagens de auxílio
- Arquivos de mensagens de erro

Complexidade de Funções de Dados

Tabela de Complexidade (ALI e AIE)			
Qtde Tipo de Registros (RLR)	Tipos de Dados (DET)		
	< 20	20-50	> 50
1	Baixa	Baixa	Média
De 2 até 5	Baixa	Média	Alta
Mais 5	Média	Alta	Alta

Tamanho Funcional - Dados

Tabela de Contribuição (ALI e AIE)			
Funcionalidade	Baixa	Média	Alta
ALI	7	10	15
AIE	5	7	10

Cálculo do Fator de Ajuste

- Após determinados os níveis de influência das 14 características gerais, o fator de ajuste é calculado com a seguinte fórmula:
- $VAF = (TDI \times 0.01) + 0,65$
- $TDI = \text{somatório dos níveis de influências das Características Gerais do Sistema (no máximo } 70 - 5 \times 14\text{)}$;
- Pode variar entre + 35% e -35%

Disciplina: TI - Concurso: Tribunal Regional

Eleitoral - ES - Ano: 2012 - Banca: CESPE -

Na concepção de um software planejado, devem ser realizadas estimativas de esforço, prazo (cronograma) e custo (orçamento), relativas ao desenvolvimento e à manutenção do software. Erros nessa fase (de sub ou de superestimação) podem afetar seriamente a gestão do projeto e causar transtornos à organização no que se refere a recursos alocados, tempo e orçamento, ou, mesmo, tornar o projeto inviável. Uma estimativa importante nesse contexto é a do tamanho do software objeto do projeto, que pode ser obtida pela técnica de análise de pontos de função (APF) é uma das principais técnicas utilizadas para se obter essa estimativa. Em face dessas informações, que têm caráter meramente motivador, redija um texto dissertativo acerca da técnica APF, desenvolvendo, necessariamente, os seguintes tópicos:

- 1 – definição, características e objetivos da APF;
- 2 – descrição sucinta do processo de contagem de pontos de função;
- 3 – classificação das funções quanto ao tipo de funcionalidades proporcionadas ao usuário do software.



2015

Parte I

Em uma aplicação solicitada por um usuário do Tribunal, um analista procedeu à contagem de Pontos de Função utilizando as seguintes tabelas e respectivas complexidades e referências ALI, AIE, SE, CE e EE (em que Baixa, Média e Alta referem-se às respectivas complexidades, alternativamente conhecidas também como Simples, Média e Complexa):

ALI e AIE	Tipos de Dados (TD)		
Tipos de Registro	<20	20–50	>50
1	Baixa	Baixa	Média
2–5	Baixa	Média	Alta
>5	Média	Alta	Alta

SE e CE	Tipos de Dados (TD)		
Arquivos Referenciados	<6	6–19	>19
<2	Baixa	Baixa	Média
2–3	Baixa	Média	Alta
>3	Média	Alta	Alta

EE	Tipos de Dados (TD)		
Arquivos Referenciados	<5	5–15	>15
<2	Baixa	Baixa	Média
2	Baixa	Média	Alta
>2	Média	Alta	Alta

	Complexidades		
	Baixa	Média	Alta
ALI	7	10	15
AIE	5	7	10
EE	3	4	6
SE	4	5	7
CE	3	4	6

No levantamento de requisitos do usuário ele registrou o seguinte:

Entradas Externas	
Tipos de Dados	Arquivos Referenciados
5	2
2	3
3	5

Saídas Externas	
Tipos de Dados	Arquivos Referenciados
3	1
1	3

Arquivos Lógicos Internos	
Tipos de Dados	Tipos de Registros
3	1
1	3

Neste contexto indique, fundamentadamente:

- a. As quantidades e respectivos tipos de funções EE, SE e ALI e a quantidade de Pontos de Função Brutos (PFB) obtida.

(Utilize as linhas abaixo para rascunho)

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

- b. Supondo-se que, ao tipo de sistema em questão, seja aplicável a seguinte tabela de características gerais do sistema utilizadas para o cálculo do fator de ajuste:

Características Gerais dos Sistemas	Justificativa	Nível de Influência (NI)
1. Comunicação de Dados	Somente um protocolo de comunicação.	4
2. Processamento Distribuído	Processamento distribuído e transferência <i>on-line</i> ambas direções.	4
3. Desempenho	Nenhuma exigência de performance.	0
4. Utilização do Equipamento	Não há restrições operacionais explícitas ou implícitas.	0
5. Volume de Transações	Não tem período de pico.	0
6. Entrada de Dados <i>On-Line</i>	Mais de 30% das transações são para entrada de dados interativas.	5
7. Eficiência do Usuário Final	Menu; help <i>on-line</i> ; movimento automático do cursor; teclas de função; utilização do mouse.	5
8. Atualização <i>On-Line</i>	Atualização <i>on-line</i> na maioria dos ALIs.	3
9. Processamento Complexo	Tem processamento lógico extensivo.	3
10. Reutilização do Código	Mais de 50% do código pode ser reutilizado.	4
11. Facilidade de Implantação	Nenhuma consideração especial sobre facilidade operacional; e processos de <i>start-up</i> , <i>backup</i> e recuperação com intervenção do operador.	0
12. Facilidade Operacional	Minimiza a operação de montagem de fita e formulários.	4
13. Múltiplos Locais	Ambientes similares de <i>hardware</i> e <i>software</i> .	3
14. Facilidade de Mudanças	Nenhum requerimento especial foi solicitado pelo usuário.	0

Indique a quantidade de Pontos de Função obtida após o ajuste (PFA).

(Utilize as linhas abaixo para rascunho)

1	
2	
3	
4	
5	

Espelho

QUESTÃO 2 (valor: 100,0 pontos)

Abordagem Esperada	Pontuação Máxima	Pontuação Atribuída																																																																		
<div>Parte I</div> <div> <table> <tr><th colspan="4">Entradas Externas</th></tr> <tr> <th>Tipos de Dados</th> <th>Arquivos Referenciados</th> <th></th> <th></th> </tr> <tr> <td>5</td> <td>2</td> <td>Média</td> <td>4</td> </tr> <tr> <td>2</td> <td>3</td> <td>Média</td> <td>4</td> </tr> <tr> <td>3</td> <td>5</td> <td>Média</td> <td>4</td> </tr> <tr> <td colspan="3">Subtotal</td> <td>12</td> </tr> </table> <div>3 EEs Médias</div> <table> <tr><th colspan="4">Saídas Externas</th></tr> <tr> <th>Tipos de Dados</th> <th>Arquivos Referenciados</th> <th></th> <th></th> </tr> <tr> <td>3</td> <td>1</td> <td>Baixa</td> <td>4</td> </tr> <tr> <td>1</td> <td>3</td> <td>Baixa</td> <td>4</td> </tr> <tr> <td colspan="3">Subtotal</td> <td>8</td> </tr> </table> <div>2 SEs Baixas</div> <table> <tr><th colspan="4">Arquivos Lógicos Internos</th></tr> <tr> <th>Tipos de Dados</th> <th>Tipos de Registros</th> <th></th> <th></th> </tr> <tr> <td>3</td> <td>1</td> <td>Baixa</td> <td>7</td> </tr> <tr> <td>1</td> <td>3</td> <td>Baixa</td> <td>7</td> </tr> <tr> <td colspan="3">Subtotal</td> <td>14</td> </tr> </table> <div>2 ALIs Baixas</div> <table> <tr> <td>Total de Pontos de Função Brutos (PFB)</td> <td>34</td> </tr> </table> </div>	Entradas Externas				Tipos de Dados	Arquivos Referenciados			5	2	Média	4	2	3	Média	4	3	5	Média	4	Subtotal			12	Saídas Externas				Tipos de Dados	Arquivos Referenciados			3	1	Baixa	4	1	3	Baixa	4	Subtotal			8	Arquivos Lógicos Internos				Tipos de Dados	Tipos de Registros			3	1	Baixa	7	1	3	Baixa	7	Subtotal			14	Total de Pontos de Função Brutos (PFB)	34	60,0	50,0
Entradas Externas																																																																				
Tipos de Dados	Arquivos Referenciados																																																																			
5	2	Média	4																																																																	
2	3	Média	4																																																																	
3	5	Média	4																																																																	
Subtotal			12																																																																	
Saídas Externas																																																																				
Tipos de Dados	Arquivos Referenciados																																																																			
3	1	Baixa	4																																																																	
1	3	Baixa	4																																																																	
Subtotal			8																																																																	
Arquivos Lógicos Internos																																																																				
Tipos de Dados	Tipos de Registros																																																																			
3	1	Baixa	7																																																																	
1	3	Baixa	7																																																																	
Subtotal			14																																																																	
Total de Pontos de Função Brutos (PFB)	34																																																																			
	50,0																																																																			

QUESTÃO 2 (valor: 100,0 pontos) (continuação)

Características Gerais dos Sistemas	Justificativa	Nível de Influência (NI)
1. Comunicação de Dados	Somente um protocolo de comunicação.	4
2. Processamento Distribuído	Processamento distribuído e transferência on-line ambas direções	4
3. Desempenho	Nenhuma exigência de performance	0
4. Utilização do Equipamento	Não há restrições operacionais explícitas ou implícitas	0
5. Volume de Transações	Não tem período de pico	0
6. Entrada de Dados On Line	Mais de 30% das transações são para entrada de dados interativas	5
7. Eficiência do Usuário Final	Menu; help on-line; movimento automático do cursor; teclas de função; utilização do mouse	5
8. Atualização On Line	Atualização on-line na maioria dos ALIs	3
9. Processamento Complexo	Tem processamento lógico extensivo	3
10. Reutilização do Código	Mais de 50% do código pode ser reutilizado	4
11. Facilidade de Implantação	Nenhuma consideração especial sobre facilidade operacional; e processos de start-up, backup e recuperação com intervenção do operador	0
12. Facilidade Operacional	Minimiza a operação de montagem de fita e formulários	4
13. Múltiplos Locais	Ambientes similares de hardware e software	3
14. Facilidade de Mudanças	Nenhum requerimento especial foi solicitado pelo usuário	0
Fator de Ajuste – FA		35

10,0

$$NIG = FA * 0,01 + 0,65 \Rightarrow NIG = 35 * 0,01 + 0,65 \Rightarrow$$
$$NIG = 0,35 + 0,65 \Rightarrow NIG = 1$$
$$PFA = PFB * NIG \Rightarrow PFA = 34 * 1 \Rightarrow PFA = 34$$

Total de Pontos de Função Ajustados (PFA)	34
---	----

DESENBAHIA



2014

DESENBAHIA - Ano: 2014 - Banca: ESSP

A Engenharia de Requisitos é um dos tópicos importantes na Engenharia de Software. Descreva suas principais características.

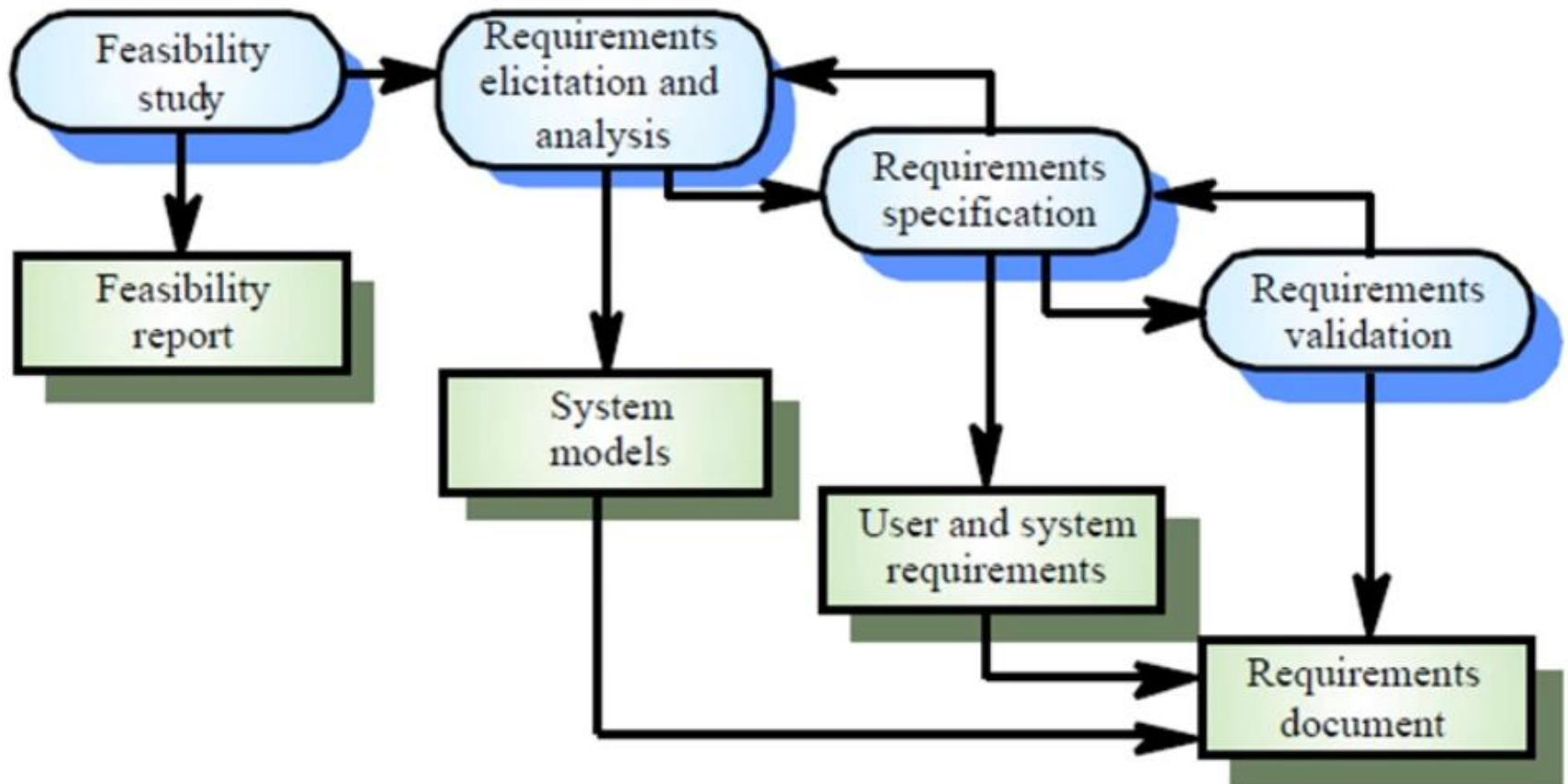
Engenharia de Requisitos

- Amplo espectro de tarefas e técnicas que levam a um entendimento dos requisitos;
- Ponte para o projeto e para a construção;
- Inicia durante a atividade de comunicação e continua na de modelagem;
- Fornece o mecanismo adequado para entender aquilo que é desejado pelo cliente;
- Analisa as necessidades, avalia a viabilidade e negocia uma solução;
- Especificar soluções sem ambiguidade;

Eng. Requisitos (PRESSMAN)

- Concepção;
- Levantamento;
- Elaboração
- Negociação;
- Especificação;
- Validação;
- Gestão de Requisitos;

Eng. Requisitos (Sommerville)



DESENBAHIA

- A engenharia de Requisitos é um dos tópicos importantes na Engenharia de Software. Descreva suas principais características.
- Requisitos claros, controlar mudanças, estudo de viabilidade, elicitação, especificação, negociação, validação, gestão de requisitos;

DESENBAHIA - Ano: 2014 - Banca: ESSP

A engenharia de requisitos é considerada um dos tópicos importantes na Engenharia de Software pois trata do conjunto de técnicas que levam a um entendimento dos requisitos de um sistema durante todo o seu ciclo de vida. Além de fornecer o mecanismo adequado para entender o que é realmente desejado pelo cliente, a engenharia de requisitos analisa as necessidades e avalia a viabilidade, e ao final, negocia uma solução. Seu principal objetivo é definir requisitos sem ambiguidades, de forma clara e bem definida.

De acordo com o Autor Ian Sommerville, a Engenharia de Requisitos possui as seguintes fases: Estudo de viabilidade (foco na análise do custo x benefício do requisito), Elicitação e Análise de Requisitos (levantamento de informações necessárias), Especificação de Requisitos (formalização dos requisitos que deverão ser desenvolvidos pelo sistema e Validação (responsável por garantir que não existem ambiguidades nos requisitos que foram definidos). Uma outra etapa que também faz parte da engenharia de requisitos é a Gestão de requisitos, fase que irá ocorrer durante todas as demais etapas, e possui como objetivo garantir a rastreabilidade entre os requisitos e as demais partes envolvidas (projetos, pessoas e requisitos dependentes) e garantir ainda um maior gerenciamento sobre a mudança dos requisitos.

– Resposta: A engenharia de requisitos (no contexto da engenharia de software) é um processo que engloba todas as atividades que contribuem para a produção de um documento de requisitos e sua manutenção ao longo do tempo. O processo de engenharia de requisitos é composto por quatro atividades de alto nível: identificação; análise e negociação; especificação e documentação; validação. Este processo deve ser precedido de estudos de viabilidade que, a partir das restrições do projeto, determinam se este é ou não viável e se deve prosseguir para a identificação dos requisitos. Uma outra atividade que se pode considerar que faz também parte deste processo, se incluirmos a fase posterior à produção do documento (isto é, a sua manutenção), é a gestão dos requisitos (change management), sendo que as alterações podem ser causadas pelos mais diversos fatores desde inovações tecnológicas a mudanças na natureza do negócio (e consequentemente nos requisitos), entre outras.

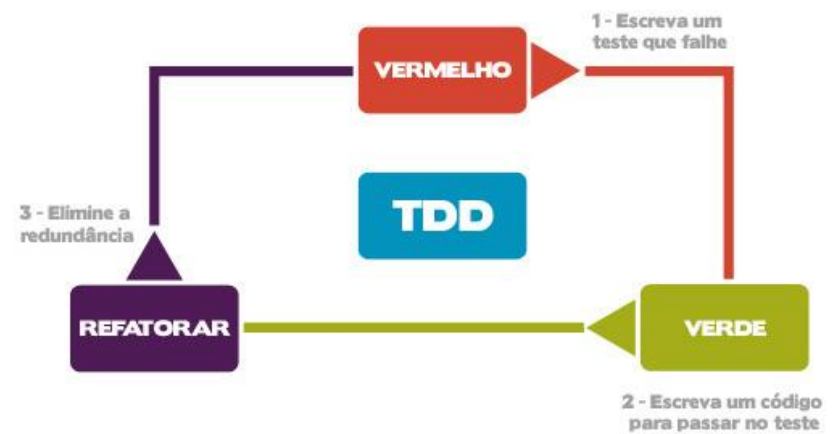


Banca: CESPE

O objetivo das metodologias ágeis é proporcionar maior rapidez e qualidade no desenvolvimento de um sistema de informação, aproximando a equipe de desenvolvimento dos usuários finais. As metodologias ágeis, como o SCRUM e o XP, apresentam, além de suas características próprias, alguns princípios em comum. As técnicas ou práticas envolvidas nos métodos ágeis podem ser complementares, sendo comum, por exemplo, associar o XP ao TDD (test driven development). Considerando que o texto acima tem caráter unicamente motivador, redija um texto dissertativo acerca do TDD. Ao elaborar seu texto, atenda, necessariamente, ao que se pede a seguir.

- 1 – Descreva a característica básica do TDD. [valor: 2,50 pontos]
- 2 – Apresente três vantagens do emprego do TDD em relação a outras metodologias ágeis. [valor: 3,00 pontos]
- 3 – Discorra sobre os princípios do XP que podem ser apoiados pelo TDD, explicando a associação entre esses procedimentos. [valor: 4,00 pontos]

Banca: CESPE



- TDD = Testar primeiro, codificar de
- Vantagens:
 - Garantir que as funcionalidades sejam testadas;
 - Promessa de maior qualidade no processo de desenvolvimento (já que o desenvolvedor irá pensar na forma que o usuário espera que funcione);
 - Pequenas unidades de código independentes (para que possam ser testadas com facilidade);
- Princípios do XP: Teste Primeiro, Refatoração

Assunto: Tecnologia da Informação -

– Resposta: Espera-se que o candidato redija texto dissertativo acerca do test driven development (TDD). Características básicas do TDD – A característica básica do TDD (test driven development) é que, no desenvolvimento de um software, os testes são criados antes da escrita dos códigos que permitirão o uso desses testes. É uma abordagem evolutiva na qual o desenvolvedor escreve o teste antes de escrever o código funcional para satisfazer esse teste. Vantagens do emprego do TDD em relação a outras metodologias ágeis (são listadas abaixo somente as três principais) – Entre as vantagens do emprego do TDD em relação a outras metodologias ágeis, incluem-se a oferta de programa de maior qualidade por estar direcionado às expectativas do cliente, já que o trabalho do desenvolvedor é feito a partir de considerações acerca de como as funcionalidades serão utilizadas por esse cliente; a possibilidade de se testar todo o código desenvolvido, o que dá maior confiabilidade ao sistema; e a geração, via de regra, de um código mais modularizado, flexível e extensível, pois a metodologia requer que os desenvolvedores pensem no software como pequenas unidades que podem ser reescritas, desenvolvidas e testadas de forma independente e integradas em momento posterior. Princípios do XP apoiados pelo TDD e associação entre os procedimentos – Um princípio do XP (extreme programming) a que o TDD está associado é o do Teste

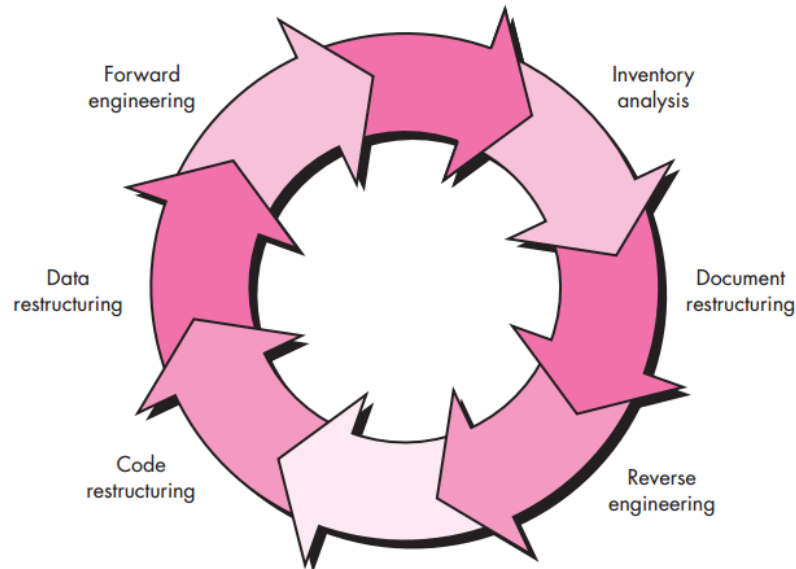
Primeiro. O TDD pode apoiar esse princípio por fornecer detalhes para a realização dos testes de unidade e de funcionalidade, que são importantes e necessários. Além disso, o TDD é útil no que diz respeito ao princípio do XP da Refatoração, pois confere ao programador maior segurança para identificar e remover o código duplicado, e permite, assim, a melhoria contínua do programa.



TI - Concurso: TCM-SP - Ano: 2011 - Banca: CETRO -
Disciplina: Tecnologia da Informação - Assunto:
Tecnologia da Informação - A Reengenharia de sistemas ocupa-se de reimplementar sistemas legados para que sua manutenção seja mais fácil. A reengenharia pode envolver, fazer uma nova documentação, organizar e reestruturar o sistema, traduzir o programa para uma linguagem de programação mais moderna, modificar e atualizar a estrutura e os valores dos dados do sistema. A funcionalidade do software não é modificada e, normalmente, a arquitetura do sistema permanece a mesma. Com base no conceito acima, responda: 1. Quais são as vantagens da reengenharia em relação ao desenvolvimento de um novo sistema?

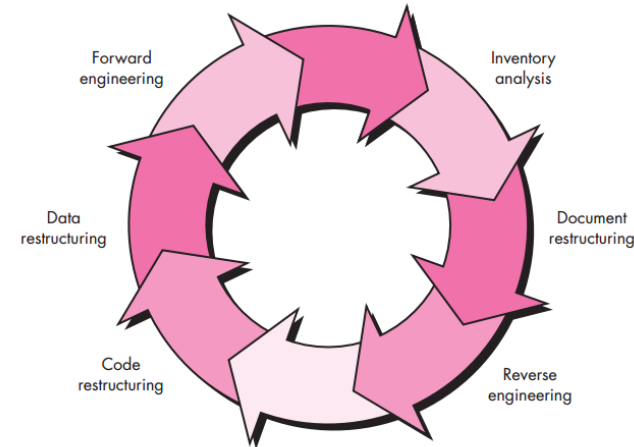
Reengenharia

- A reengenharia tem como principal objectivo melhorar um sistema de alguma maneira, através de alterações significativas que proporcionem melhoria, porém, sem alterar as suas funções (WARDEN, 2000).



Reengenharia

- Análise de inventário
 - Planilha com informações detalhadas
- Reestruturação dos Documentos
 - [Não]Criar/Atualizar Documentação
- Engenharia Reversa
- Reestruturação do Código
 - Reescrever em uma linguagem mais moderna
- Reestruturação dos Dados
 - Reescrever em uma estrutura mais moderna
- Engenharia Direta
 - “Motor de geração de aplicação”

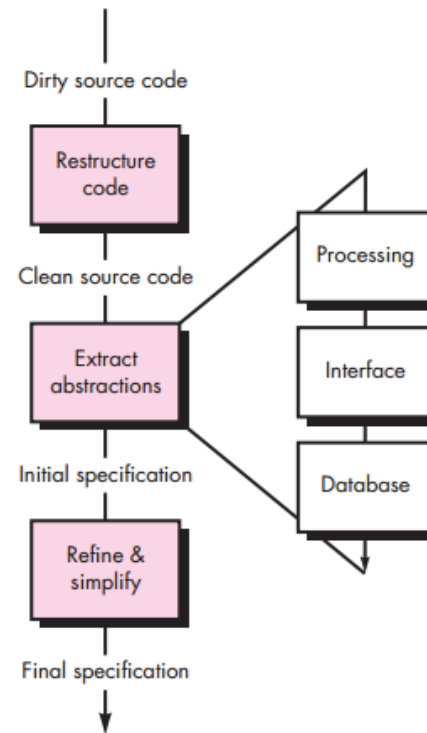


Reestruturação

- É o processo de transformação de um software de uma forma de representação não estruturada em uma forma estruturada preservando sua funcionalidade.
- Sua finalidade é fazer o software mais fácil de se entender e mudar, ou ficar menos suscetível a erros quando mudanças futuras forem feitas.
- Poder ser realizada através da utilização de ferramentas ou através da análise do Analista.

Engenharia Reversa

- Processo de analisar um programa na tentativa de criar uma representação do programa em um nível mais alto de abstração do que o código fonte.



Engenharia Direta

- Reimplementa a função do sistema existente e também acrescenta novas funções ou melhorias.
- Processo de gerar o código fonte à partir de um caso de uso ou requisito do sistema.

TI - Concurso: TCM-SP - Ano: 2011 - Banca: CETRO -
Disciplina: Tecnologia da Informação - Assunto:
Tecnologia da Informação - A Reengenharia de sistemas ocupa-se de reimplementar sistemas legados para que sua manutenção seja mais fácil. A reengenharia pode envolver, fazer uma nova documentação, organizar e reestruturar o sistema, traduzir o programa para uma linguagem de programação mais moderna, modificar e atualizar a estrutura e os valores dos dados do sistema. A funcionalidade do software não é modificada e, normalmente, a arquitetura do sistema permanece a mesma. Com base no conceito acima, responda: 1. Quais são as vantagens da reengenharia em relação ao desenvolvimento de um novo sistema?

BONUS

Banca: CESGRANRIO

Dois órgãos públicos federais, que utilizam plataformas de desenvolvimento distintas, desejam trocar informações entre si, por meio de Web Services. Explique como ocorre esse intercâmbio, citando as tecnologias SOAP, XML e WSDL.

ELETRONBRAS - Ano: 2010 - Banca: CESGRANRIO

Dois órgãos públicos federais, que utilizam plataformas de desenvolvimento distintas, desejam trocar informações entre si, por meio de Web Services. Explique como ocorre esse intercâmbio, citando as tecnologias SOAP, XML e WSDL.

Web Services

Garantir interoperabilidade entre sistemas;

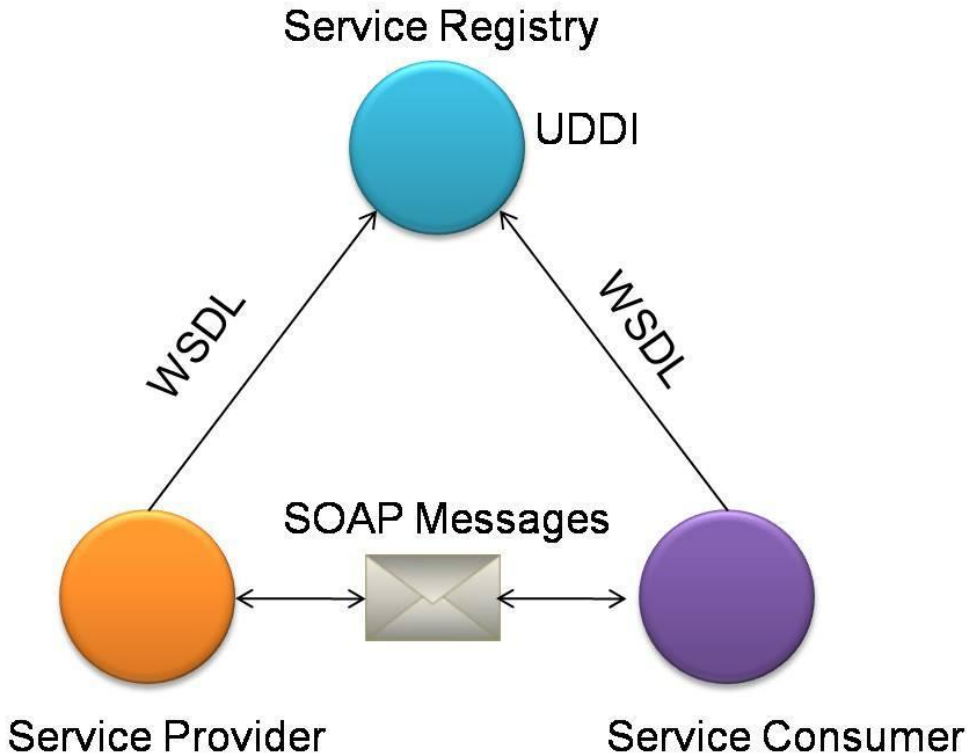
Enviar e receber dados em formato XML;

Requisição HTTP;

Corpo Padronizado;



Interoperabilidade



Registro dos Serviços – Diretório com as informações sobre os serviços.

Provedor de Serviço – Responsável pela descrição e publicação de um determinado serviço web no registro dos serviços.

Consumidor – Responsável por descobrir, obter a descrição e utilizar o serviço.

Características



XML- Based – a fim de garantir a interoperabilidade entre diferentes plataformas;

Baixo acomplamento – Web service pode mudar sem que cliente precise ser alterado;.

Síncrono/Assíncrono – Aplicação escolhe qual melhor estratégia;

SOAP (Simple Object Access Protocol)

Protocolo para a troca de informações estruturadas em uma plataforma descentralizada e distribuída.

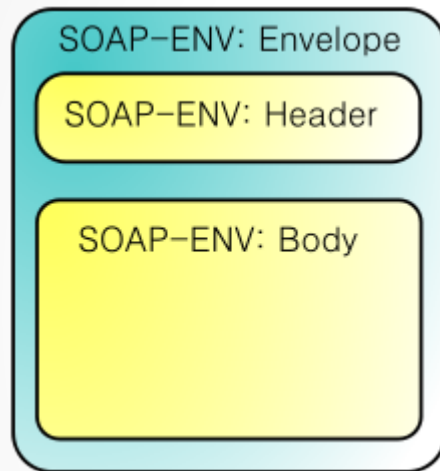
Independente de implementação específica.

Possibilita comunicação entre processos diferentes, construídos em linguagens diferentes.

Baseado em XML



ESTRUTURA SOAP



Mensagem SOAP é composta por:

Envelope = Elemento raiz. Contém declarações de namespaces e define o documento XML como uma mensagem SOAP

Header = contém informação de controle e processamento (autenticação por exemplo)

Body = contém informação da chamada ou da resposta da mensagem

A SOAP request:

```
POST /InStock HTTP/1.1
Host: www.example.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn

<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

  <soap:Body xmlns:m="http://www.example.org/stock">
    <m:GetStockPrice>
      <m:StockName>IBM</m:StockName>
    </m:GetStockPrice>
  </soap:Body>

</soap:Envelope>
```

The SOAP response:

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn

<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

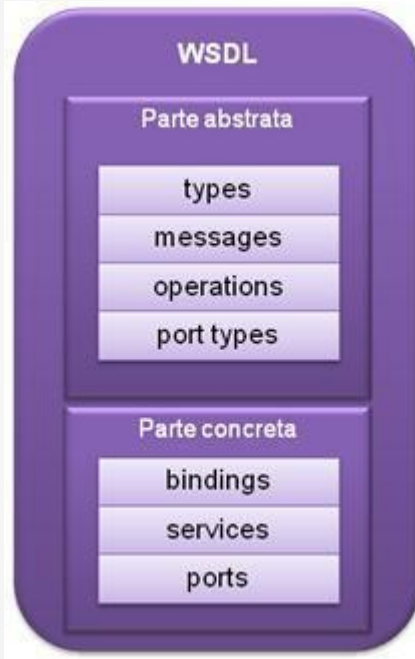
  <soap:Body xmlns:m="http://www.example.org/stock">
    <m:GetStockPriceResponse>
      <m:Price>34.5</m:Price>
    </m:GetStockPriceResponse>
  </soap:Body>

</soap:Envelope>
```

WSDL (Web Service Description Language)

- Linguagem baseada em XML utilizada para descrever Web Services.
- Funciona como um contrato do serviço.
- Além de descrever o serviço, especifica como acessá-lo e quais as operações ou métodos disponíveis.
- Operações, protocolo de comunicação e localização do serviço (Endpoint).

ESTRUTURA DO WSDL



Parte Abstrata – descreve a interface do serviço

Parte Concreta – define o protocolo e o endereço aonde o serviço será localizado

Elementos	Descrição
<types>	Especifica os tipos de dados trocados
<message>	Descreve as mensagens que são trocadas entre o serviço e o consumidor
<portType> <operations>	Conjunto de operações suportadas pelo serviço
<binding>	Detalhes de como as mensagens serão transmitidas (protocolo de rede para invocação)
<service> <port>	Referencia a atual localização do serviço

```

<?xml version="1.0"?>
<definitionsname="CustomerInfo"
  <types>
    <xsd schema targetNamespace=
      "http://www.customercommandservice.com/CustomerCommand"
      xmlns="http://www.w3.org/1999/XMLSchema"
      <xsd:complexType name="Customer">
        <xsd:element name="Num"type="xsd:string"/>
        ...
      </xsd:complexType>
    </xsd schema>
  </types>
  <message name="GetCustomerInfoInput">
    <part name="Customer"type="Customer"/>
  </message>
  ...
  <portType name="CustomerInfoPortType">
    <operation name="GetCustomerInfo">
      <input message="GetCustomerInfoInput"/>
      <output message="GetCustomerInfoOutput"/>
    </operation>
  </portType>

  <binding name="CustomerInfoConnectorBinding"type="CustomerInfoPortType">
    <format:typemapping style="COBOL"encoding="COBOL">
      <format:typemap typename="Customer"formattype="/CustomerInfo.ccp:CUSTINF"/>
    </format:typemapping>
    <operation name="GetCustomerInfo">
      <cics:operation functionName="GETCUST"/>
      <input>
        ...
      </input>
      <output>
        ...
      </output>
    </operation>
  </binding>
  <service name="CustomerServices">
    <port name="CICS_A"binding="CustomerInfoConnectorBinding">
      <cics:address connectionURL=".. "serverName="CICS_A"/>
    </port>
  </service>
</definition>

```

UDDI (Universal Description Discovery and Integration)

- Especificação que define um serviço de registro para Web Services
- Protocolo para descobrir, registrar e integrar web services
- Clientes utilizam o UDDI para obter informações sobre os serviços que lhe interessam e obter os metadados necessários para utilizar estes serviços
- Utilizado para buscar serviços de um determinado tipo;

REST (Representational State Transfer)

- Alternativa na utilização de Web Services
- Baseado em HTTP
- Utiliza um conjunto de operações bem definidas (POST, GET, PUT e DELETE)
- Stateless – favorece a escalabilidade
- Serviços são consumidos através de URLs, por exemplo:
- <http://www.thomas-bayer.com/sqlrest/CUSTOMER/>

ELETROBRAS - Ano: 2010 - Banca: CESGRANRIO -

– Resposta: Web Services é uma solução para integração de sistemas em ambientes heterogêneos, proporcionando um mecanismo de intercâmbio normalizado de dados. Para esse intercâmbio, cada órgão oferece serviços a serem consumidos e pode consumir serviços oferecidos pelo outro órgão. O protocolo utilizado é o HTTP (ou HTTPS), que simplifica a conectividade, principalmente por questões relacionadas a firewalls, proxies, etc. Os serviços oferecidos são descritos pela WSDL (Web Services DescriptionLanguage), e as mensagens de dados são trocadas utilizando o SOAP (SimpleObject Access Protocol), que utiliza o XML (ExtensibleMarkupLanguage) como formato de intercâmbio.

Simulado – Questão 1

O RUP é um processo de engenharia de software que aumenta a produtividade da equipe e oferece as melhores práticas relacionadas a software através de diretrizes, templates e orientações sobre ferramentas para todas as atividades críticas de desenvolvimento de software.

- a) Defina de forma sucinta quais os objetivos e a proposta desta metodologia. (5 linhas)
- b) Cite pelo menos duas características do RUP. (5 linhas)
- c) Descreva a organização do eixo horizontal e suas responsabilidades. (10 linhas)

Simulado – Questão 2

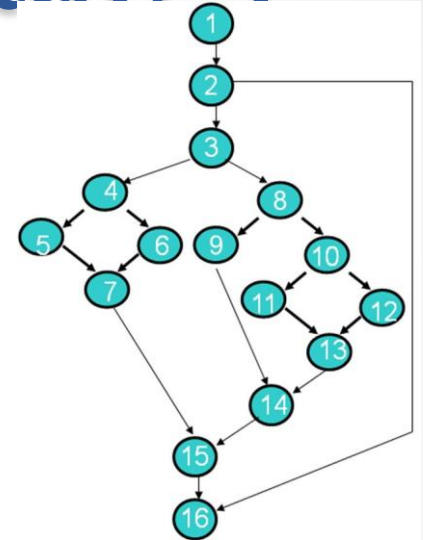
Na construção de um software utilizando o processo de desenvolvimento SCRUM os ciclos de desenvolvimento são curtos e constantes. Acerca deste assunto, descreva o ciclo de vida de um requisito no SCRUM. (15 linhas)

Simulado – Questão 3

Sobre o XP (Xtreme Programming), cite pelo menos 5 (cinco) práticas propostas por esta metodologia, bem como o objetivo de cada uma destas. (10 linhas)

Simulado – Questão 4

A aplicação cuja representação em grafo está na figura a seguir, foi desenvolvida para proceder com a automatização de um processo de contabilidade que era executado de forma manual. A atividade de teste é fundamental para auxiliar a cumprir os objetivos de qualidade para que o sistema possa ser entregue. Sobre testes,



- a) Defina 3 (três) técnicas de testes e suas características. (10 linhas)
- b) Qual a complexidade ciclomática de um software que possua a representação em grafo a seguir (demonstre os cálculos).