

Situação Finalizada**Iniciado** quarta-feira, 30 abr. 2025, 10:00**Concluído** quarta-feira, 30 abr. 2025, 12:22**Duração** 2 horas 21 minutos**Nota** 9,00 de um máximo de 10,00(90%)**Questão 1**

Completo

Atingiu 1,00 de 1,00

Assuma que em um projeto o seu orçamento (custo total) é de R\$ 100.200,00. Além disto, considere o método de Análise do Valor Agregado e os dados de cada mês listados abaixo. Apresente e demonstre os cálculos dos índices de desempenho de cronograma e de custo (SPI e CPI) de cada um dos meses.

Mês	P%C	A%C	AC
1	10%	8%	7.200
2	20%	14%	29.084
3	30%	16%	38.850
4	40%	30%	45.200
5	50%	48%	50.288
6	60%	54%	68.600
7	80%	69%	77.254
8	100%	83%	97.554



Formulas usadas para efetuar os cálculos:

$$BAC = 100200,00$$

$$PV = BAC * P\%C$$

$$EV = BAC * A\%C$$

$$CPI = EV/AC$$

$$SPI = EV/PV$$

Observação: tentei anexar o arquivo da tabela mas não obtive sucesso.

Tabela montada no Google Sheets:

Mês	P%C	A%C	AC (R\$)	PV (R\$)	EV (R\$)	CPI	SPI
1	10%	8%	7.200,00	10.020,00	8.016,00	1,11	0,8
2	20%	14%	29.084,00	20.040,00	14.028,00	0,48	0,7
3	30%	16%	38.850,00	30.060,00	16.032,00	0,41	0,533333
4	40%	30%	45.200,00	40.080,00	30.060,00	0,67	0,75
5	50%	48%	50.288,00	50.100,00	48.096,00	0,96	0,96
6	60%	54%	68.600,00	60.120,00	54.108,00	0,79	0,9
7	80%	69%	77.254,00	80.160,00	69.138,00	0,89	0,8625
8	100%	83%	97.554,00	100.200,00	83.166,00	0,85	0,83

Comentário:

ok

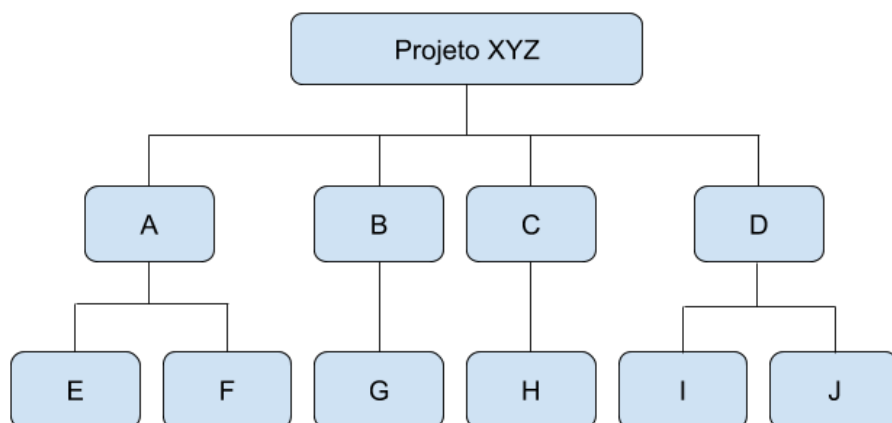
Questão 2

Completo

Atingiu 1,00 de 1,00

Considere a Estrutura Analítica do Projeto (EAP) da figura a seguir, bem como as seguintes informações:

- O banco de dados relacional deste projeto possuirá 15 tabelas.
- O sistemas tem interação com 3 outros softwares externos.
- Apenas os elementos folha do EAP são considerados funcionalidades a serem desenvolvidas neste projeto.
- O elemento H é uma funcionalidade que requer que o usuário informe dados de entrada para cadastro de informações no sistema.
- Os elementos E, F e G são funcionalidades que não requisitam nenhuma informação do usuário, e apenas apresentam dados já armazenados pelo sistema.
- Os demais elementos não citados correspondem a funcionalidades que requerem que o usuário informe algum dado, para o posterior processamento do sistema e visualização de dados.



Diante das informações fornecidas, utilize os conceitos da Análise de Pontos por Função para definir o quantitativo de cada um dos tipos de função abaixo:

- Entradas Externas
- Saídas Externas
- Consultas Externas
- Arquivos Lógicos Internos
- Arquivos de Interface Externa

- Entradas Externas: 1 (elemento H)
- Saídas Externas: 3
- Consultas Externas: 2
- Arquivos Lógicos Internos: 15 (tabelas)
- Arquivos de Interface Externa: 3

Comentário:

ok

Questão 3

Completo

Atingiu 1,00 de 1,00

Utilize os quantitativos das suas respostas na questão anterior para calcular os Pontos de Função Não-ajustados. Em seguida, informe quantos KLOCs terá o projeto, considerando que o mesmo será desenvolvido na Linguagem Java e que 1 PFNA corresponde a 53 linhas de código (LOCs).

Para o cálculo do PFNA considere as seguintes complexidades:

- Todos os Arquivos Lógicos Internos e de Interface Externa são de complexidade baixa
- Todas as funções do tipo Saída Externa e Consulta Externa possuem complexidade alta
- Todas as funções de Entrada Externa possuem complexidade média

Use as ponderações abaixo para o cálculo:

Elemento\Complexidade	Baixa	Média	Alta
Entradas Externas (EE)	3	4	6
Saídas Externas (SE)	4	5	7
Consultas Externas (CE)	3	4	6
Arquivos Lógicos Internos (ALI)	7	10	15
Arquivos de Interface Externos (AIE)	5	7	10

Entrada externa: $1 \times 4 = 4$

Saída externa: $3 \times 7 = 21$

Consulta externa: $2 \times 6 = 12$

Arquivos Lógicos Internos: $15 \times 7 = 105$

Arquivos de Interface Externa: $3 \times 5 = 15$

Total = PFNA = 157

Portanto, Klocks pode ser obtido através da relação:

$157 \times 53 = 8321 \text{ LOCs} = 8,321 \text{ KLOCS}$

Comentário:

ok



Questão 4

Completo

Atingiu 1,00 de 1,00

A partir dos KLOCs obtidos na questão anterior, calcule o Esforço e Duração do projeto utilizando o método paramétrico COCOMO. Considere que este projeto corresponde ao desenvolvimento de um software considerado crítico. Para o valor final da Duração, utilize a precisão de valores inteiros, arredondando para cima apenas quando a primeira casa decimal for maior que 5. (Observação: Para o cálculo do esforço, utilize a precisão de 2 casas após a vírgula).

COCOMO Básico				
Complexidade	α	β	ϵ	θ
Simples	2,4	1,05	2,5	0,38
Média	3,0	1,12	2,5	0,35
Alta	3,6	1,20	2,5	0,32

$$\text{Esforço (E)} = \alpha \cdot \text{KLOC}^\beta$$

$$\text{Tempo (T)} = \epsilon \cdot E^\theta$$

Esforço: $3,6 * (8,321^{1,20}) \Rightarrow 3,6 * 12,71 = 45,76$ pessoas/mês

Duração: $2,5 * (45,76^{0,32}) \Rightarrow 2,5 * 3,40 = 8,4 \Rightarrow 8$ meses



Comentário:

ok

Questão 5

Completo

Atingiu 0,70 de 1,00

Considere o valor da duração do projeto (T) calculada na questão anterior em conjunto com a tabela a seguir que esclarece a dependência entre as atividades, e a fórmula de cálculo da duração de cada atividade. Construa uma rede PERT-CPM e calcule os tempos mais cedo, mais tarde e a folga de cada evento. Por fim, indique qual é o caminho crítico. Todos os caminhos da rede devem terminar em um único evento. Nesta rede, **as atividades devem ser as arestas**. Considere que o prazo de entrega estabelecido para este projeto é igual ao tempo mais cedo do último evento da rede.

Atividade	Atividade Precedente	Duração
E	-	T/T
F	E	T/4
G	F	$T/(T*2)$
H	F	T/T
I	G	T/4
J	H,I	T/3

 [_questao5.pdf](#)

Comentário:

Uma atividade não pode ser representada 2 vezes (J) - não está correta essa composição do grafo. Vc também esqueceu da atividade E.

Questão 6

Completo

Atingiu 0,80 de 1,00



Suponha que um programador adote a seguinte estratégia: ao implementar qualquer nova funcionalidade ou corrigir um bug que implique na modificação de duas classes A e B localizadas em arquivos diferentes, ele conclui a tarefa movendo as classes para o mesmo arquivo. Por exemplo, após terminar a tarefa de programação que ficou sob sua responsabilidade, ele escolhe uma das classes, digamos a classe B, e a move para o mesmo arquivo da classe A. Agindo dessa maneira, ele estará melhorando qual propriedade de projeto? Por outro lado, qual propriedade de projeto estará sendo afetada de modo negativo? Justifique.

Ao mover as classes A e B para o mesmo arquivo após a modificação o desenvolvedor irá melhorar a localidade por referência, isso significa que quando uma tarefa futura exigir manutenção ou alteração em ambas as classes, todo o código estará próximo em um único arquivo, facilitando o ato de visualização e edição ao reduzir a necessidade de navegação entre múltiplos arquivos/diretórios. Porém essa abordagem prejudica a coesão e modularidade do código, visto que um arquivo ao possuir duas classes que diferem entre si perde seu foco e passa a ter múltiplas responsabilidades, ficando assim menos coeso e dificultando o seu entendimento além da reutilização independente de cada classe.

Comentário:

Vc não mencionou o nome da propriedade que melhora!

Questão 7

Completo

Atingiu 1,00 de 1,00

Qual princípio de projeto é violado pelo seguinte código? Como você poderia alterar o código do método para atender a esse princípio?

```
void sendMail(ContaBancaria conta, String msg) {  
    Cliente cliente = conta.getCliente();  
    String endereco = cliente.getEmailAddress();  
    "Envia mail"  
}
```

O código viola o princípio de Demeter, que recomenda que a implementação de um método deve invocar apenas os métodos de sua própria classe, objetos criados pelo próprio método e de objetos passados como parâmetro, além disso o código possui um acoplamento ruim com a classe ContaBancaria, pois embora sendMail seja dependente direto de ContaBancaria, a forma como ele obtém o endereço de email (usando os métodos) faz com que ele seja dependente da estrutura interna de ContaBancaria e da interface específica da classe Cliente. Essa é uma dependência que não ocorre unicamente através da interface direta de ContaBancaria para a atividade de obtenção do email. Mas para concertar essa violação de projeto basta transformar esse acoplamento ruim em um acoplamento aceitável, fazendo com que sendMail dependa apenas da interface direta de ContaBancaria para obter as informações que precisa, para isso irei adicionar um novo método à classe que encapsula a lógica de buscar o novo endereço de email do cliente, fazendo com que sendMail apenas solicite informação, sem precisar saber como ela está estruturada.

Código corrigido:

```
public class ContaBancaria{  
    private Cliente cliente;  
  
    public String getClientMailAddress(){  
        return this.cliente.getEmailAddress();  
    }  
  
}
```

```
void sendMail(ContaBancaria conta, String msg){  
    String endereco = conta.getClientMailAddress();  
    // faz o envio  
  
}
```

Comentário:

ok



Questão 8

Completo

Atingiu 1,00 de 1,00

Forneça um exemplo de código-fonte na linguagem Java que corresponda a uma violação do princípio de Segregação de Interfaces. Não pode ser utilizado o exemplo apresentado em sala de aula. Forneça explicações pertinentes para explicar a violação.

Observação inicial: Nessa questão tentei mexer na formatação do texto para exibir melhor o código e não consigo voltar para a formatação original, então creio que as letras estão maiores que o normal.

Na minha representação abaixo de um código fonte de um jogo é perceptível a violação do princípio de segregação de interfaces quando analisa-se a interface `GameObject`, nela há métodos que representam as ações de todos os possíveis objetos do jogo, porém não são todos os objetos/personagens do jogo que irão utilizar os métodos herdados. Por exemplo, no código abaixo eu illustrei um inimigo básico e uma porta, ambos herdam tudo da interface `GameObject` e isso viola o princípio, essas novas interfaces são forçadas a lidar com métodos que não correspondem ao seu propósito como o `InimigoBasico` que lidará com `abrirPorta()`, já a `Porta` irá lidar com `coletarItem()` sendo que não há necessidade. Sendo assim, é evidente que as interfaces `InimigoBasico` e `Porta` carregam métodos que não são universalmente aplicáveis aos conceitos que representam.

Pseudo-Código:

```
interface GameObject {  
    void updatePosicao();  
    void render();  
    void ataque();  
    void abrirPorta();  
    void coletarItem();  
}
```

```
interface InimigoBasico extends GameObject {  
    void setarAlvo(Jogador p1);  
    // metodos do inimigo basico  
}
```

```
interface Porta extends GameObject {  
    boolean trancada();  
    // metodos da porta  
}
```

Comentário:

ok



Questão 9

Completo

Atingiu 0,50 de 1,00

Explique o que diferencia os modelos de processo de software V e W.

Ambos os modelos são derivações do modelo Waterfall, ou seja, seguem uma ideia de processo de desenvolvimento linear e sequencial. O modelo V difere-se no que diz respeito a realização de testes após cada etapa de desenvolvimento, a parte de validação do que foi desenvolvido é efetuada após a conclusão da codificação. Já o modelo W possui um forte planejamento e execução de testes em paralelo ao desenvolvimento do projeto desde seu início até o fim, para tentar ao máximo detectar defeitos e aprimorar o sistema de forma gradual.

Comentário:

O W implementa totalmente o V. A diferença é que o W faz o planejamento dos testes paralelo ao desenvolvimento. Diferente do que vc escreveu, não faz testes em paralelo ao desenvolvimento, e sim o planejamento destes.

Questão 10

Completo

Atingiu 1,00 de 1,00

A empresa Spotify utiliza o modelo de desenvolvimento Scrum, entretanto, introduz novos conceitos ao modelo original Scrum. Quais são estes conceitos e o que eles significam?



A primeira adição ao modelo do método ágil foi o Squad, são equipes multifuncionais de pessoas assim como as equipes do Scrum porém no modelo Spotify podem ser compostas de 6 a 12 pessoas. Nos squads os membros tem PO dedicado e possuem uma grande autonomia dentro da empresa e tendem a desenvolver e planejar projetos para longo prazo. Devido ao tamanho do Spotify, torna-se complexo a interação entre as Squads, então foi desenvolvido as tribos, que são um arranjo de Squads que trabalham em áreas semelhantes como Frontend, interfaces, seleção de músicas e muito mais. As tribos são compostas por aproximadamente 100 pessoas, das quais há um líder, e devem trocar informações e trabalhos entre si para uma comunicação e produção eficientes. Há também as guildas, que são comunidades formadas apenas para troca de informação e interesse em comum, sem nenhuma formalidade ou com objetivo de otimizar a entrega do produto, existem apenas para fortalecer a cultura dos funcionários em um assunto específico. E por fim há os Capítulos, que são a união de pessoas com habilidades iguais porém que não necessariamente estão alocadas no mesmo squad, mas sim na mesma tribo, os capítulos de uma tribo se reúnem frequentemente para trocar informações e alinhar as produtos do momento e geralmente há um gerente para cada vertente dos capítulos.

Comentário:

ok