

Aula 00

*CEFET-MG (Técnico em Tecnologia da
Informação) Análise e Desenvolvimento -
2021 (Pós-Edital)*

Autor:

**Diego Carvalho, Equipe
Informática e TI, Fernando
Pedrosa Lopes**

11 de Julho de 2021

Sumário

| | |
|---|----|
| Rational Unified Process (RUP) | 3 |
| 1 – Conceitos Básicos | 3 |
| 2 – Princípios-Chave..... | 10 |
| 3 – Melhores Práticas..... | 13 |
| 4 – Fases do RUP..... | 15 |
| 4.1 – Iniciação (ou Concepção) | 15 |
| 4.2 – Elaboração..... | 15 |
| 4.3 – Construção | 16 |
| 4.4 – Transição | 16 |
| 5 – Disciplinas do RUP | 20 |
| 5.1 – Modelagem de Negócios | 20 |
| 5.2 – Requisitos..... | 21 |
| 5.3 – Análise e Projeto/Design..... | 22 |
| 5.4 – Implementação..... | 23 |
| 5.5 – Teste..... | 24 |
| 5.6 – Implantação..... | 25 |
| 5.7 – Gerenciamento de Configuração e Mudanças..... | 26 |
| 5.8 – Gerenciamento de Projetos | 27 |
| 5.9 – Ambiente..... | 28 |
| 6 – Esquema | 29 |
| Exercícios Comentados | 30 |
| Lista de Exercícios..... | 73 |
| Gabarito | 93 |



APRESENTAÇÃO

Seus lindos, essa aula é sobre um framework que é cada vez menos usado na prática, mas as bancas continuando adoraaaaaando esse assunto – é impressionante! Galera, esse é um assunto bem tranquilo de entender, tem alguns decorebas e é relativamente grande, **no entanto ele é bem interessante para você notar como funciona o desenvolvimento de um software.** Fechou? Então animação e altivez...

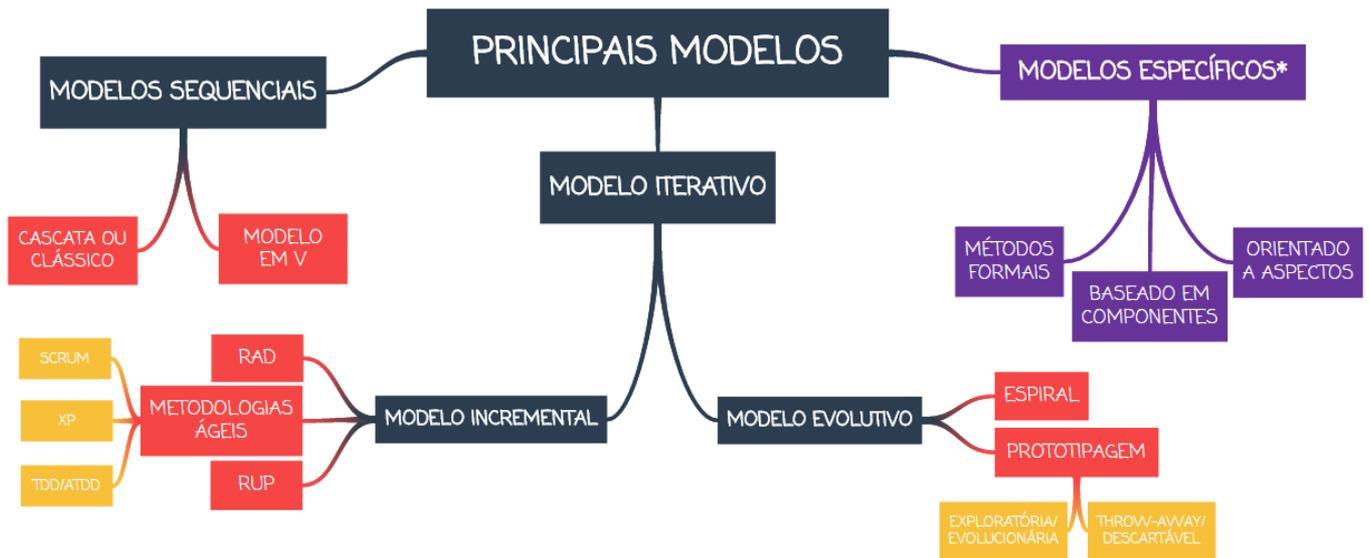


PROFESSOR DIEGO CARVALHO - [WWW.INSTAGRAM.COM/PROFESSORDIEGOCARVALHO](https://www.instagram.com/professordiegocarvalho)



RATIONAL UNIFIED PROCESS (RUP)

1 – Conceitos Básicos



Em 1995, havia uma empresa de desenvolvimento de software na Califórnia chamada **Rational Unified Corporation**. Nessa empresa, trabalharam três engenheiros de software fantásticos: o Grady Booch, o Jim Rumbaugh e o Ivar Jacobson (ficaram posteriormente conhecidos como The Three Amigos por conta de um filme de comédia que rolava na época).

O Grady Booch era especializado em Análise e Projeto Orientados a Objetos; o Jim Rumbaugh era especializado em Modelagem de Objetos; e o Ivar Jacobson era especializado em Processos de Software orientados a objetos. Esses dois últimos, inclusive, só por curiosidade, são os criadores da UML. **E esses caras trabalhavam lá na Rational, cada um em sua área, fazendo coisas diferentes.**

Até que eles começaram a interagir, viraram amigos, começaram a relatar os problemas que eles enfrentavam e como eles faziam para solucionar esses problemas – as melhores práticas que eles utilizavam. **Foi aí que eles tiveram a ideia de unificar essas melhores práticas de desenvolvimento de software em um único processo chamado Processo Unificado.**

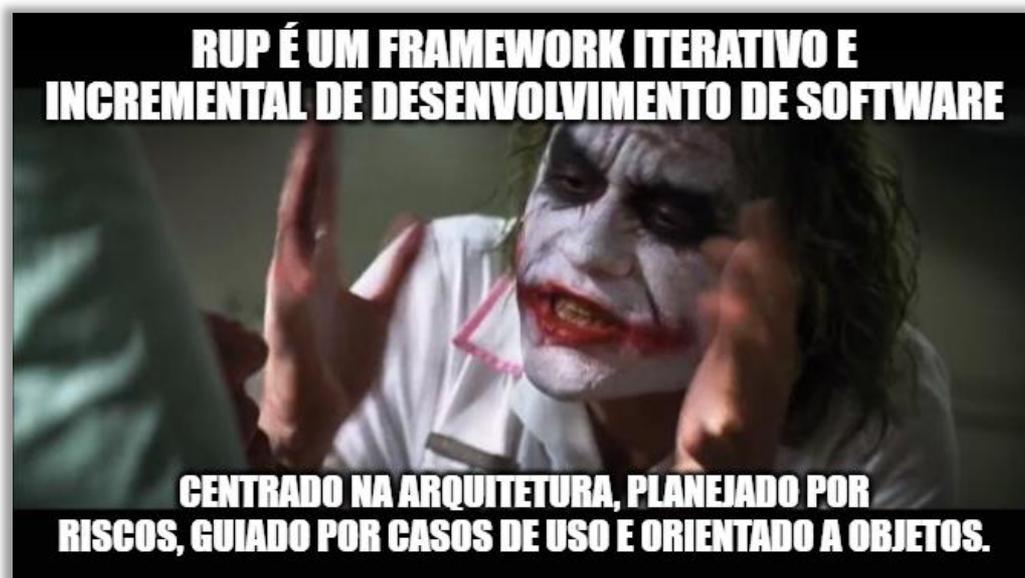
Como eles trabalhavam na Rational, ficou conhecido como Processo Unificado da Rational ou Rational Unified Process (RUP). E foi assim que nasceu o RUP! Em 2003, a IBM comprou a Rational e ela passou a manter, distribuir, evoluir, oferecer consultoria sobre RUP desde então. Alguns começaram a chamar o RUP de IRUP, que é IBM Rational Unified Process.



IMPORTANTE

RUP É O PROCESSO UNIFICADO DA RATIONAL! TRATA-SE DE UMA VERSÃO COMERCIAL DO PROCESSO UNIFICADO – ALGUNS DIZEM QUE SE TRATA DE UM REFINAMENTO COMERCIAL. EXISTEM OUTROS REFINAMENTOS, TAIS COMO SCOTT AMBLER'S AGILE UNIFIED PROCESS OU ECLIPSE FOUNDATION'S OPEN UNIFIED PROCESS. EM 2003, FOI ADQUIRIDO PELA IBM E O NOME FOI TROCADO PARA IRUP! SE CAIR EM PROVA “PROCESSO UNIFICADO”, PODEM CONSIDERAR COMO SINÔNIMO DE RUP OU IRUP! OK?

O Rational Unified Process (RUP) é um framework iterativo e incremental de desenvolvimento de software, centrado na arquitetura, planejado por riscos, guiado por casos de uso e orientado a objetos, criado pela Rational, empresa adquirida pela IBM. Vamos destrinchar essa definição, passo a passo, porque ela é importantíssima para o que virá a seguir.



O RUP é um framework! Um framework pode ser traduzido como um arcabouço ou plataforma! *Mas uma plataforma de que, professor?* Uma plataforma de melhores práticas, uma plataforma de processos. É muito simples! É só isso! **Nas provas, vocês vão encontrar questões falando que ele é uma metodologia, modelo ou processo de desenvolvimento de software.**

Galera, pelo amor de Deus, não marquem errado na questão por causa disso, mas o ideal, o mais correto, é chamá-lo de framework! *Beleza?* Além disso, ele é um framework adaptável. Isso quer dizer que você não é obrigado a utilizar tudo que existe no RUP na sua empresa, por exemplo. **Você pode utilizar aquilo que convém a sua realidade e ignorar aquilo que não lhe convém.** Por que?



Porque o RUP é configurável! *Sabe quando você compra o CD do Office e, na hora de instalar, você fala: eu vou usar Word, Excel, Outlook e não quero, vou usar o PowerPoint, Visual e não preciso, e assim por diante? RUP é a mesma coisa! Seguindo: RUP é iterativo e incremental, ou seja, grosso modo, ele é incremental porque eu desenvolvo o sistema em partes.*

E ele é iterativo, porque eu desenvolvo o sistema por meio de repetições, refinando essas partes. Agora uma pergunta para vocês: *RUP é um processo ágil? Não, RUP não é um processo ágil! Na verdade, todos os processos ágeis são iterativos e incrementais, mas nem todos os processos iterativos e incrementais são necessariamente ágeis. Certo?*

Prosseguindo: RUP é um framework de Desenvolvimento de Software! *Professor, é desenvolvimento de Hardware também? Não! É desenvolvimento de projeto? Não! É gerenciamento de projeto? Não! É desenvolvimento de software. Bacana? RUP é centrado na arquitetura! Professor, o que é a arquitetura de software? É uma macro-organização do software. É o core, o núcleo do software.*

Eu gosto de uma definição que diz que arquitetura é tudo aquilo que sobra quando você não pode tirar mais nada do software. Fazendo um paralelo com um prédio: eu posso tirar a pintura do prédio, os azulejos, portas, janelas, posso tirar várias coisas, mas eu não posso tirar uma coluna. Se eu tirar uma coluna, o prédio cai! *Então, o que quer dizer ser centrado na arquitetura?*

O RUP afirma que os projetos devem ter bases sólidas e estáveis, mas ao mesmo tempo essas bases devem ser flexíveis o suficiente para possibilitar incrementos em cada fase iterativa do processo. É como se eu construísse um prédio já pensando na parte hidráulica, na parte elétrica, na fiação de internet, de modo que tudo que venha depois seja feito com mais facilidade, já esteja tudo preparado.

O RUP é planejado por riscos! *Por que?* Porque ele busca controlar, mitigar ou reduzir os riscos mais críticos já no início do projeto. **Ou seja, pessoal, o planejamento é feito baseado nos riscos – aqueles casos de uso que tem um risco mais alto e que são mais complexos são feitos primeiro.** É por isso que se fala que o RUP é planejado por riscos.

RUP é guiado por casos de uso! *Por que?* **Porque a principal técnica utilizada para levantar e capturar requisitos de software é a técnica de casos de uso ou cenários de casos de uso.** A gente vai ver mais para frente que eles são produzidos lá nas primeiras fases e, a partir daí todas as outras disciplinas utilizam esses casos de uso em seu contexto específico.

Por fim, o RUP é orientado a objetos! *Por que?* **Porque esse é o paradigma de mundo em que ele se apoia. Ele vê o mundo como um conjunto de objetos, classes, colaborações, de interfaces, etc.** Acabamos de ver essa definição e agora vamos ver o que estimulou a sua criação. Ele se originou a partir de uma tentativa de acabar com uma série de problemas frequentes:

- **Necessidades não eram atendidas:** por conta de falhas na verificação e validação, os requisitos não atendiam as necessidades do negócio e do usuário.



- **Requisitos muito voláteis e instáveis:** partia-se do princípio de que os requisitos não se modificariam ao longo do desenvolvimento do projeto.
- **Módulos não se integravam:** projeto era modularizado, dividido entre equipes, entretanto ocorriam diversas falhas no momento de integrá-los.
- **Dificuldades de manutenção:** não havia assistência técnica nem padronização para escrita do software, prejudicando a manutenção.
- **Descoberta tardia de problemas:** requisitos eram entendidos de maneira errada, não eram validados e causavam altos prejuízos quando descobertos.
- **Qualidade ou experiência pobre dos usuários:** a validação dos requisitos era feita por usuários dispensáveis da organização, geralmente inexperientes.
- **Performance de carga sofrível:** a aplicação satisfaz todos os requisitos, no entanto tem uma performance sofrível.
- **Esforço descoordenado da equipe:** falta de coordenação adequada implica em retrabalho por parte das equipes.

O RUP é considerado por muitos uma metodologia pesada! *Pesada como, professor?* **Pesada no sentido de ser muito cerimonioso, muito formal, muito verboso, de possuir muita documentação.** *Comparado ao que?* Comparado com as metodologias ágeis, que seguem outro paradigma! No entanto, como eu disse, ele é altamente adaptável e configurável.

Dessa forma, se ele for usado na íntegra, ele realmente é pesado; caso contrário, você pode configurá-lo e ele fica mais leve. Além disso, o Rational Unified Process (RUP) afirma que sua meta é garantir a produção de software de alta qualidade que atenda às necessidades dos usuários dentro de um cronograma e de um orçamento previsíveis.

Percebam aqui o foco na qualidade, que nós vamos ver mais tarde que está contida em todas as fases e em todas as disciplinas do RUP. E percebam também que ele se preocupa com duas coisas: atender às necessidades dos usuários e entregar o software no cronograma e no orçamento – que eram alguns dos problemas mais grave que havia na época.

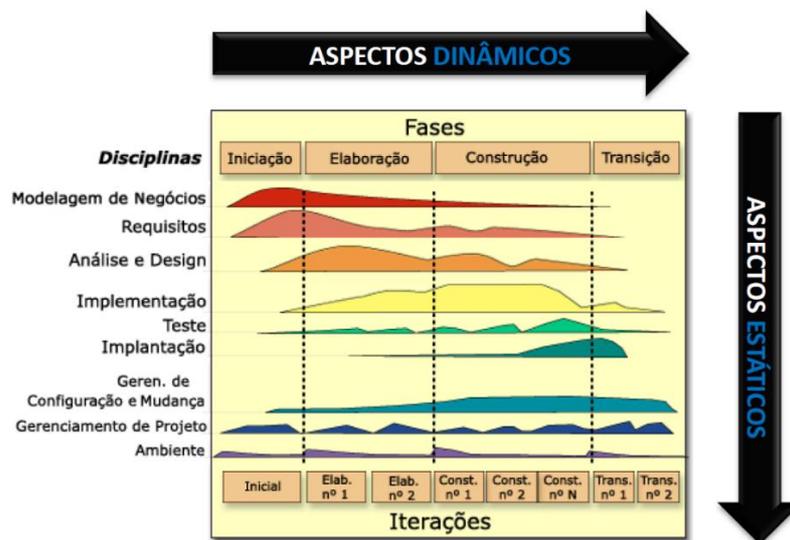
O RUP oferece uma abordagem baseada em disciplinas para atribuir tarefas e responsabilidades dentro de uma organização de desenvolvimento. **Sua meta é garantir a produção de software de alta qualidade que atenda às necessidades dos usuários dentro de um cronograma e de um orçamento previsíveis.** O RUP é geralmente descrito a partir de três perspectivas:

1. **Dinâmica:** também conhecido como temporal ou horizontal, mostra as fases do modelo ao longo do tempo, além de iterações e marcos do projeto.



- Estática:** também conhecido como funcional ou vertical, mostra as atividades realizadas no processo, componentes, disciplinas, artefatos e papéis de processo.
- Prática:** sugere boas práticas a serem usadas no processo. Galera, essa dimensão não é reconhecida por todos os autores.

A maior parte das descrições do RUP tenta combinar as perspectivas estática e dinâmica em um único diagrama, como é mostrado abaixo (Gráfico de Baleias):



Esse gráfico representa de maneira abstrata, ou seja, em alto nível, a arquitetura ou a organização do RUP. A perspectiva dinâmica representa aspectos da metodologia ao longo do tempo, por isso ela é chamada de dinâmica. *E que aspectos dinâmicos são esses?* São as fases, as iterações e os marcos. Percebam que o gráfico mostra as fases (parte superior), iterações (parte inferior) e marcos (Linhas pontilhadas).

Já a perspectiva estática representa aspectos da metodologia independentemente do tempo, por isso ela é chamada de estática. *E que aspectos estáticos são esses?* São as atividades, artefatos, papéis e fluxos de trabalho. *Professor, o que você quer dizer com "independente do tempo"?* Isso significa que esses aspectos (atividades, artefatos, papéis e fluxos de trabalho) são sempre os mesmos independentemente da fase.

Notem também que existem quatro fases e essas fases são discretas e sequenciais, ou seja, uma fase só começa após o término da fase anterior. Percebam também que eu falei que as fases são sequenciais e, não, que as iterações são sequencias – as iterações podem se sobrepor eventualmente. *E como eu sei que uma fase chegou ao fim, professor?* Por meio dos marcos de cada fase!

Por fim, mais três coisinhas: primeiro, percebam que pode haver mais de uma iteração por fase; segundo, essa quantidade de iterações do gráfico é só um exemplo – ela varia de projeto



para projeto; e terceiro: o gráfico de baleias trata de intensidade, então se uma questão disser, por exemplo, que não ocorre nenhuma atividade de implementação na iniciação, está errado! Ocorre pouco, mas ocorre...

Bem, antes de entrar em mais detalhes sobre as fases e as disciplinas, vamos ver alguns conceitos importantes do RUP! Esses conceitos não caem em prova diretamente, caem no contexto de outras questões, mas é importante para entender o que vem mais à frente. **Um processo, no RUP, define quem faz o quê, como e quando com o intuito de alcançar algum objetivo específico.**

Papel (Role): definição abstrata dos comportamentos e responsabilidades de um indivíduo (ou um conjunto de indivíduos) que trabalham como uma equipe em uma organização¹.

Atividades (Activities): é uma unidade de trabalho que fornece um resultado significativo dentro do contexto de um projeto. Tem um propósito claro e envolve criar ou atualizar artefatos.

Artefato (Artifact): trata-se de um pedaço de informação que é produzido e/ou utilizado durante a execução do processo – são os produtos tangíveis do processo.

Fluxo de Trabalho: uma sequência de atividades encadeadas e relacionadas a uma disciplina do projeto, tal como: Modelagem de Negócio, Implementação, Testes, etc.

Algumas observações: um papel não é uma pessoa! Na verdade, um papel pode ser desempenhado por várias pessoas, assim como uma pessoa pode desempenhar vários papéis – é um relacionamento muitos-para-muitos. **Para vocês terem uma ideia, existem 32 papéis no RUP. Existe, por exemplo, Analista de Testes, Especificador de Requisitos, Arquiteto de Software, etc.**

O que vocês precisam saber é que uma pessoa pode desempenhar mais de um papel na organização. **Ela pode ser Analista de Testes em um projeto e Especificadora de Requisitos em outro (ou até no mesmo projeto).** Da mesma forma, o papel de Analista de Testes, por exemplo, pode ser desempenhado por várias pessoas. Então, na minha organização, pode ser que eu tenha 5 analistas de testes.

Além disso, um artefato pode ser um documento, como um Caso de Negócio ou um Documento de Arquitetura de Software; um Modelo, como um Modelo de Casos de Uso ou Modelo de Design; ou pode ser um Elemento do Modelo, como uma Classe ou um Subsistema. **Galera, existem muito mais artefatos do que papéis – esses que eu mencionei são só alguns exemplos.**

¹ RUP possui 32 papéis! Na minha opinião, decorar quais são os papéis e as atividades de cada um é completamente inviável – decorá-los tem um péssimo custo/benefício.



Outra coisa importante é que um artefato também é chamado de Produto de Trabalho ou Work Product. Ok? Já as atividades estão fortemente relacionadas aos artefatos. Os artefatos geralmente são a entrada e/ou a saída das atividades. Então, por exemplo: uma das atividades do Papel Especificador de Requisitos é Detalhar um caso de Uso e o artefato produzido é o Caso de Uso em si.

| ARTEFATOS | DESCRIÇÃO |
|--------------------------------------|--|
| DOCUMENTO DE VISÃO | Explica o que é o projeto aos principais stakeholders. |
| CASOS DE NEGÓCIO | Fornecer informações necessárias para determinar se vale a pena ou não investir no projeto. |
| PLANO DE DESENVOLVIMENTO DE SOFTWARE | Reúne todas as informações necessárias ao gerenciamento do projeto. |
| MODELO DE CASOS DE USO | Modelo de funções futuras do sistema e do ambiente que serve como um contrato entre as partes. |
| GLOSSÁRIO | Define termos importantes usados pelo projeto. |
| PROTÓTIPOS | São usados de uma maneira direta para reduzir o risco. |
| LISTA | Classificada em ordem decrescente de importância e associada à contingência ou diminuição de riscos. |
| DOCUMENTO DE ARQUITETURA DE SOFTWARE | Visão geral de arquitetura do sistema, usando visões para descrever diferentes aspectos do sistema. |
| MODELO DE PROJETO | Modelo de objeto que descreve a realização dos casos de uso e serve como abstração do modelo. |
| MODELO DE DADOS | Subconjunto do modelo de implementação que descreve a representação lógica e física dos dados. |
| CONJUNTO DE TESTES | Testes implementados e executados para validar a estabilidade da versão de cada release executável. |
| NOTAS DE RELEASE | Identificam mudanças e erros em uma versão ou em uma unidade de implantação disponível para uso. |
| ARTEFATOS DE INSTALAÇÃO | Referem-se ao software e às instruções documentadas necessárias para instalar os produtos. |
| MATERIAIS DE TREINAMENTO | Material usado nos programas ou cursos de para ajudar os usuários finais a operação dos produtos. |



2 – Princípios-Chave

O RUP apresenta também seis princípios chaves para orientar o desenvolvimento de software. São estratégias altamente recomendadas! Percebam:

ADAPTAR PROCESSOS

O RUP é considerado um framework pesado, mas isso é meio injusto, porque o RUP pode ser adaptado. Ele pode ser configurado – existem, inclusive, configurações já prontas. O RUP tem uma versão mais leve chamada *RUP for Small Projects*. Há também outras implementações do RUP mais próximas de metodologias ágeis, como a *Open UP*.

O RUP serve tanto para desenvolver um sistema complexo de controle de tráfego aéreo, por exemplo, quanto para desenvolver o software de uma Padaria.

BALANCEAR PRIORIDADES DOS INTERESSADOS

Galera, na hora de levantar requisitos, vocês vão ouvir de tudo! Vai ter aquele usuário que vai pedir a coisa mais complexa do mundo e ele vai querer que tudo seja barato e que seja entregue o mais rápido possível. **O problema é que essas coisas são contraditórias: coisas complexas geralmente são mais caras e demoram mais para serem feitas – não tem mágica!**

Acontece muito também de o usuário querer algo personalizado só para ele ou para área dele. Isso também tem um custo alto, porque geralmente usar um componente já pronto, já padronizado, pode reduzir bastante os custos e o tempo de entrega. **Tem também sempre aquele cara pede algo que reduz a performance do sistema, por exemplo, apesar de ele não concordar.**

Outras vezes, você encontra dois caras da mesma área que têm entendimentos diferentes sobre um mesmo requisito ou sobre um mesmo problema. Quando você tem um sistema que permeia várias áreas de uma organização, acontece de o cara de uma área achar que o problema ou funcionalidade dele é mais importante que o problema ou funcionalidade do outro.

Vocês percebem como isso é complexo? O RUP diz que se deve chegar a um consenso ou algo próximo disso, balanceando essas prioridades. **Então, esse princípio é muito importante.** Deve-se balancear as prioridades dos interessados e, como eu disse, uma forma de conseguir isso é evitar personalizações e apostar em uma padronização do sistema.

COLABORAÇÃO ENTRE TIMES



A ideia aqui é dividir as equipes de acordo com as habilidades. **O intuito é aumentar a produtividade e a interação entre as equipes, trocando experiências, orientações, etc.** Às vezes, você encontra um cara que é mais sênior e que pode ajudar o cara que é mais júnior. Então a harmonia entre as pessoas e as equipes é essencial para o sucesso do projeto.

DEMONSTRAR O VALOR DA ITERATIVIDADE

A iteratividade (iteratividade, não!) proporciona a possibilidade de se adaptar a mudanças no escopo, tempo ou custo. Ao tratar os riscos mais prioritários no início do projeto, desenvolve-se uma arquitetura mais robusta, sendo possível prever o andamento do projeto. Com a iteratividade, o feedback é mais rápido e contínuo, a adaptação de planos ocorre de maneira mais fácil e as mudanças também, enfim...

ELEVAR O NÍVEL DE ABSTRAÇÃO

Muitas vezes, a complexidade é a maior inimiga do sucesso de um projeto. **Então, elevar o nível de abstração ajuda a reduzir a complexidade, assim como a reduzir a quantidade de documentação de um projeto.** Como conseguir isso? Reutilizando ativos e componentes, utilizando ferramentas, utilizando linguagens de mais alto nível – isso ajuda a reduzir a quantidade de documentação produzida.

Evitar documentar coisas de baixo nível, como estruturas de dados, detalhes de implementação, porque isso pode gerar ruído na comunicação, entre outras coisas. O ideal é reutilizar coisas que já existem em vez de fazer as coisas do zero. **É tentar sempre utilizar frameworks, ferramentas de geração de código, ferramentas de automação de software, middlewares, softwares open-source.**

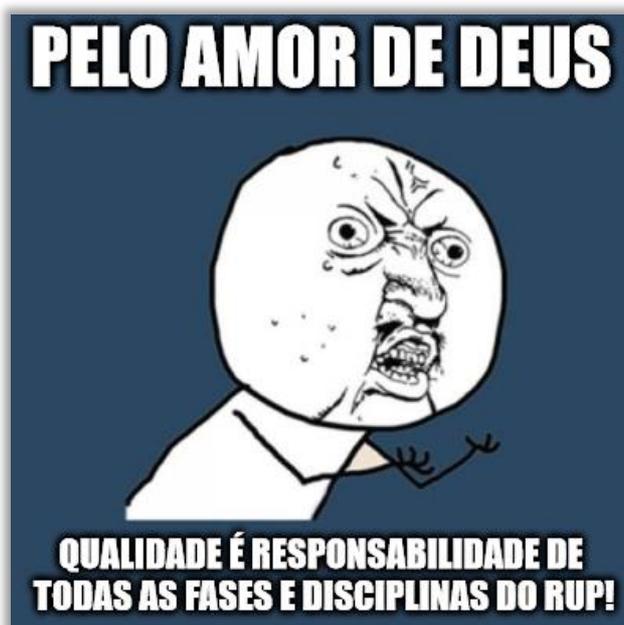
Não podemos esquecer também das linguagens de modelagem (UML), linguagens de desenvolvimento rápido de software – tudo que for possível para reduzir a complexidade do projeto. **Percebam que isso tudo envolve uma arquitetura de software bem escolhida e bem sólida.** Dessa forma, se você vai falar com o usuário final, você vai conseguir falar em um nível mais alto e ele poderá contribuir mais.

FOCO CONTÍNUO NA QUALIDADE

Vocês vão ver que não existe uma fase ou disciplina exclusiva de qualidade! Por que, professor? No PMBOK, por exemplo, existe uma área de conhecimento de gerenciamento de qualidade! Galera, não há uma fase ou disciplina de qualidade, porque considera-se que qualidade deve estar contida em todas as fases e em todas as disciplinas do RUP.



Você sempre tem que focar em qualidade, não importa o que você esteja fazendo. Eu até coloquei o meme abaixo. Uma banca pode cobrar em uma questão de prova, por exemplo, que a qualidade é responsabilidade da Disciplina de Testes ou da Fase de Transição. *Certo ou errado?* Errado, a qualidade é responsabilidade de todo mundo a todo momento.



3 – Melhores Práticas

Vocês se lembram algumas páginas atrás que eu falei que o RUP tinha duas perspectivas? Uma era a perspectiva estática e a outra era a perspectiva dinâmica. Certo? O Ian Sommerville afirma que essas seis melhores práticas formam uma terceira perspectiva do RUP: a perspectiva prática. **Isso é uma visão do Sommerville, então não levem isso como uma verdade universal.**

1. **Desenvolvimento Iterativo:** um projeto que usa o desenvolvimento iterativo tem um ciclo de vida que consiste em várias iterações, que incorporam um conjunto quase sequencial de atividades em modelagem de negócios, requisitos, análise e projeto, implementação, teste e implantação, dependendo do local em que ela está localizada no ciclo. **Este desenvolvimento gera um build que já pode interagir com os usuários gerando novos e melhores requisitos.**
2. **Gerenciamento de Requisitos:** condição ou capacidade com a qual o sistema deverá estar em conformidade, isto é, uma abordagem sistemática para levantar e documentar os requisitos e acompanhar todo o ciclo de vida desse requisito no processo de software. **Para tanto, deve-se estar preocupado com a Verificação e Validação destes.** Alguns requisitos dependem de outros, logo todos devem estar mapeados, para caso haja uma mudança, todos possam ser rastreados.
3. **Uso da Arquitetura de Componentes:** componentes são grupos de códigos coesos, na forma de código fonte ou executável, com interfaces bem definidas e comportamentos que fornecem forte encapsulamento do conteúdo e são, portanto, substituíveis. **De maneira informal, um componente pode ser descrito como algo que executa uma funcionalidade sem conhecer seus detalhes de implementação.**
4. **Modelagem Visual (UML):** é o uso de notações de design gráficas e textuais, semanticamente ricas, para capturar designs de software. **Uma notação, como a UML, permite que o nível de abstração seja aumentado, enquanto mantém sintaxe e semântica rígidas.** Assim, a comunicação na equipe melhora à medida que o design é formado e revisado, permitindo ao leitor raciocinar sobre ele e fornecendo uma base não ambígua para a implementação.
5. **Contínua Verificação da Qualidade:** garantia da qualidade de software é o ponto mais comum de falha nos projetos de software. **O RUP ajuda no planejamento do controle da qualidade e cuida da sua construção em todo processo, envolvendo todos os membros da equipe.** Nenhuma tarefa é especificamente direcionada para a qualidade e o RUP assume que cada membro da equipe é responsável pela qualidade durante todo o processo.
6. **Gerenciamento de Mudanças:** em todos os projetos de software, mudanças são inevitáveis. **RUP define métodos para controlar, rastrear e monitorar estas mudanças.** RUP também define espaços de trabalho, garantindo que um sistema de engenharia de software não será

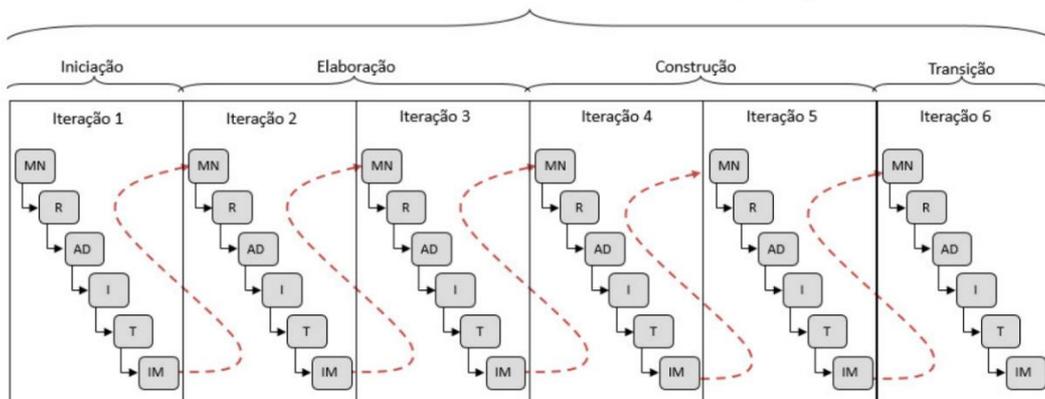


afetado por mudanças em outros sistemas. Este conceito é bem aderente com arquiteturas de software baseados em componentização.

Este conceito é bem aderente com arquiteturas de software baseados em componentização.



Sim! **Para o CESPE** é obrigatório adotar a UML no RUP! Isso não é opcional! Não há outra alternativa de modelagem visual. *Usou RUP?* UML é obrigatório!



Em suma, há três perspectivas: dinâmica, estática e prática. Há seis disciplinas de engenharia e mais três disciplinas de apoio/infraestrutura. Uma passagem pelas quatro fases se chama Ciclo de Desenvolvimento e uma passagem pelas nove disciplinas se chama Iteração, **podendo haver mais de uma iteração por fase**, como mostra a imagem acima (na Elaboração e na Construção).



4 – Fases do RUP

O RUP é constituído por quatro fases discretas e sequenciais. No entanto, ao contrário do modelo em cascata – no qual as fases coincidem com as atividades do processo –, **as fases do RUP estão relacionadas mais estritamente aos negócios do que a assuntos técnicos**. Cada fase consiste em um intervalo de tempo entre dois marcos principais. Podemos dividi-las em quatro fases:

4.1 – Iniciação (ou Concepção)

Tem o objetivo de estabelecer um caso de negócio para o sistema. **Deve-se identificar as entidades externas, que irão interagir com o sistema, e definir essas interações**. Depois você usa essas informações para avaliar a contribuição do sistema com o negócio. Se a contribuição for de pouca importância, o projeto pode ser cancelado depois dessa fase.

Atividades Básicas:

- Formular o escopo do projeto;
- Planejar e preparar um caso de negócio;
- Sintetizar uma possível arquitetura;
- Preparar o ambiente para o projeto;

Artefatos importantes: Documento de Visão; Casos de Negócio; Plano de Desenvolvimento do Software; Modelo de Casos de Uso; Glossário.

4.2 – Elaboração

Tem o objetivo de **desenvolver um entendimento do domínio do problema, estabelecer um framework de arquitetura para o sistema**, desenvolver o plano de projeto e identificar os riscos principais do projeto. Ao concluir, deve-se ter um modelo de requisitos para o sistema, uma descrição de arquitetura e um plano de desenvolvimento para o software.

Atividades Básicas:

- Definir, validar e criar a baseline da arquitetura;
- Refinar a Visão, estabelecendo uma compreensão sólida dos casos de uso;
- Criar planos de iteração detalhados e baselines para a fase de construção;
- Refinar o caso de desenvolvimento e posicionar o ambiente de suporte;
- Refinar a arquitetura e selecionar componentes.

Artefatos importantes: Protótipos; Lista de Riscos; Documento de Arquitetura de Software; Modelo de Projeto; Modelo de Dados.



4.3 – Construção

Tem o objetivo de desenvolver partes do sistema paralelamente e integradas durante esta fase. **Está essencialmente relacionada ao projeto, programação e teste de sistema.** Ao concluir esta fase, você deve ter um sistema de software em funcionamento e a documentação associada pronta para ser liberada para os usuários.

Atividades Básicas:

- Gerenciamento de recursos, otimização de controle e processo;
- Desenvolvimento completo do componente e teste dos critérios de avaliação;
- Avaliação dos releases do produto de acordo com os critérios de aceitação.

Artefatos importantes: o próprio sistema executável e suas bibliotecas; conjunto de testes do software; materiais de treinamento; material de suporte do usuário.

4.4 – Transição

Tem o objetivo de colocar o sistema em funcionamento no ambiente real de uso. **A fase final do RUP está relacionada à transferência do sistema da comunidade de desenvolvimento para a comunidade dos usuários.** Essa é uma atividade onerosa e, às vezes, problemática. Ao concluir esta fase, você deverá ter um sistema de software documentado.

Atividades Básicas:

- Executar planos de implantação;
- Finalizar o material de suporte para o usuário final;
- Testar o produto liberado no local do desenvolvimento;
- Criar um release do produto;
- Obter feedback do usuário;
- Ajustar o produto com base em feedback;
- Disponibilizar o produto para os usuários finais

Artefatos importantes: Notas de Release; Artefatos de Instalação; entre outros.

Vamos falar agora dos marcos de cada fase com mais detalhes! Um marco é um **ponto final reconhecível de uma atividade do processo de software**. A cada marco, deve existir uma saída formal e deve representar o fim de um estágio lógico e distinto do projeto. Para cada fase do projeto, há diversos marcos. No entanto, os principais de cada fase são apresentados abaixo:

| FASE | MARCO |
|------------|--|
| INICIAÇÃO | Escopo ou objetivos do ciclo de vida. |
| ELABORAÇÃO | Arquitetura estabilizada do ciclo de vida. |



CONSTRUÇÃO Capacidade operacional inicial.

TRANSIÇÃO Lançamento (ou Release) do produto.

O marco da iniciação é o escopo do projeto, definindo objetivos e necessidades e **delimitando sua área de aplicação**. Observem que já na iniciação, faz-se uma arquitetura **inicial** do projeto – apenas um esboço. Além disso, busca-se por riscos de projeto e decide-se sobre a viabilidade do projeto e sua continuidade. O marco da elaboração é uma arquitetura estabilizada. *De novo, o que é uma arquitetura?*

Arquitetura é uma macro-organização do software. É um conjunto de casos de uso críticos que foram implementados – similar a um esqueleto que sustenta um corpo. Um problema na arquitetura pode comprometer todo o projeto. Uma boa analogia é o plano de urbanização de Brasília (1960). A cidade não estava povoada, mas poder-se-ia prever como seria o processo de urbanização no decorrer dos anos.

Não se sabia detalhes, **mas se sabia os limites mínimos e máximos que a arquitetura poderia suportar**. Quando se chega à fase de construção, deve-se ter **software a baixo custo e alta qualidade**. Para tanto, faz-se necessário criar um paralelismo entre as equipes. O marco da construção é uma capacidade operacional inicial do software com parte de suas funcionalidades prontas.

É como uma versão beta. A partir da construção, as iterações começam a gerar builds, que são conjunto de casos de uso implementados. O marco da transição é o lançamento do produto, de fato. Agora, antes de ver as disciplinas, vamos ver algumas informações esparsas, mas importantes: Manual de Treinamento e Manual de Suporte do Usuário são planejados na Elaboração e refinados na Construção.

Ambos os artefatos pertencem à disciplina de Implantação. **Pessoal, perceba que o RUP descreve artefatos para as suas disciplinas e, não, para as suas fases**. Por conta disso, é um pouco complicado definir à qual fase pertence algum determinado artefato. *Como assim, professor?* Galera, existe um relacionamento 1:N entre artefatos e disciplinas.

Isso significa que uma disciplina pode ter vários artefatos, mas um artefato só é produzido por uma disciplina. No entanto, existe um relacionamento N:N entre artefatos e fases, isto é, uma fase pode apresentar vários artefatos e um artefato pode fazer parte de várias fases. Logo, não é possível – para qualquer artefato – definir à qual fase ele pertence.

Por que toda essa explicação? Porque já me perguntaram isso algumas vezes e porque mais acima eu falei os artefatos mais importantes de cada fase. **No entanto, isso não é possível para todos os artefatos. Existem alguns que são importantes em duas ou mais fases**. Ok? O RUP também apresenta alguns números mágicos: cada iteração dura em média de 2 a 6 semanas.



O número médio de iterações é entre 3 e 9 em um projeto de média complexidade. Outro dado importante é que a disciplina Modelagem de Negócios não é obrigatória! **O Rational Unified Process é um framework de processo de desenvolvimento de software completo, fornecido com várias instâncias iniciais configuráveis.**

Os processos derivados do RUP podem ser leves - voltados para as necessidades dos projetos pequenos com ciclos curtos de produto - ou processos mais abrangentes voltados para as necessidades mais amplas de equipes de projetos grandes, possivelmente distribuídas. **Projetos de todos os tipos e tamanhos já usaram o RUP com êxito.** Vamos falar um pouco mais sobre o RUP for Small Projects!

Trata-se de um exemplo de configuração do RUP para projetos de pequeno porte! *Vocês já imaginaram utilizar o RUP padrão para fazer um software simples de uma Farmácia, por exemplo?* Torna-se bastante inviável devido a sua complexidade, mas o RUP for Small Projects veio para tentar resolver esse problema. **Para tal, foi necessário que ele fizesse uma série de concessões.**

Os diagramas são diferentes; artefatos utilizados na maioria dos projetos pequenos foram removidos (Material de Treinamento, Arquitetura de Automação de Testes, etc); artefatos cobertos por outros artefatos foram removidos (Modelo de Análise, Modelo de Implantação, Especificação de Requisitos de Software, etc) – **eles são contemplados por outros artefatos de maneira simplificada.**

Planos detalhados separados foram removidos (Plano de Implantação, Plano de Gerenciamento de Configuração, Plano de Gerenciamento de Riscos, etc); a disciplina de Ambiente foi inteiramente removida; algumas revisões de projeto também foram removidas; entre outros. Então, percebam como o nosso framework é realmente bastante configurável.

Para finalizar, vamos falar um pouquinho sobre Riscos! Um projeto é sempre incerto. Os riscos devem ser identificados para que sejam previstos e diminuídos, se possível, ou para que sejam controlados quando houver poucas estratégias para a sua diminuição. O risco controla os planos de iteração; as iterações são planejadas considerando riscos específicos na tentativa de limitar o risco ou reduzi-lo.

A lista de riscos é revista periodicamente para avaliar a eficácia das estratégias de diminuição de riscos e, conseqüentemente, orientar as revisões no plano de projeto e nos planos de iteração subsequentes. O segredo do gerenciamento de risco é não esperar até que haja risco (e isso passe a ser um problema ou uma falha) para decidir o que fazer em relação a ele.

Uma mudança de alguns graus no percurso de um voo transcontinental produz um efeito significativo no local de aterrissagem do avião; de modo semelhante, **gerenciar o risco antecipadamente é quase sempre menos dispendioso e penoso do que tentar solucioná-lo depois que virar um fato.** Há três estratégias principais de gerenciamento de riscos.

- **Prevenção de Riscos:** reorganiza o projeto para que ele não possa ser afetado por um risco.



- **Transferência de Riscos:** reorganiza o projeto para que alguém ou algo assuma o risco, com intuito de prevenir os riscos.
- **Aceitação de Riscos:** aceita conviver com o risco como uma contingência, monitorando seus sintomas.

Ao aceitar um risco, deve-se tomar duas atitudes: primeiro, diminuir o risco, isto é, tomar alguma medida imediata e pró-ativa para reduzir a probabilidade ou o impacto do risco; segundo, definir um plano de contingência, isto é, que curso de ação tomar se o risco tornar-se um problema real.

Então, essas também podem ser consideradas estratégias de gerenciamento de riscos.

É importante fazer a distinção entre riscos diretos e indiretos. Em poucas palavras, um risco direto é aquele que permite um certo grau de controle e o indireto é o que não pode ser controlado. **Há quatro tipos de riscos:** riscos de recursos, riscos de negócio, riscos técnicos e riscos de programação. O gerenciamento de riscos incorpora-se à disciplina de Gerenciamento de Projeto.

Como é, professor? **O Gerenciamento de Riscos é responsabilidade do Gerente do Projeto, mas permeia todas as fases do desenvolvimento de software.** Ele deve desenvolver o plano de gerenciamento de riscos, identificar e avaliar riscos e monitorar o andamento do projeto. Os artefatos de entrada ou saída são: Documento de Visão Geral, Planos de Gerenciamento de Riscos e Lista de Riscos.



5 – Disciplinas do RUP

Dito isso, vamos finalmente para a última parte da aula em que falaremos sobre as seis disciplinas principais e três disciplinas de apoio ou infraestrutura. São elas:

5.1 – Modelagem de Negócios

Os processos de negócios são modelados usando casos de uso de negócios. Busca um entendimento comum da estrutura e da dinâmica da organização na qual um sistema deve ser implantado, além dos problemas atuais da organização-alvo e identificar as possibilidades de melhoria – está bastante ligada ao Plano Estratégico Institucional da organização.

FINALIDADES

ENTENDER A ESTRUTURA E A DINÂMICA DA ORGANIZAÇÃO NA QUAL UM SISTEMA DEVE SER IMPLANTADO (A ORGANIZAÇÃO-ALVO).

Entender os problemas atuais da organização-alvo e identificar as possibilidades de melhoria.

Assegurar que os clientes, usuários e desenvolvedores tenham um entendimento comum da organização-alvo.

Derivar os requisitos de sistema necessários para sustentar a organização-alvo.

Os artefatos da Modelagem de Negócio capturam e apresentam o contexto de negócios do sistema e funcionam como referência para os requisitos do sistema.



5.2 – Requisitos

Os agentes que interagem com o sistema são identificados e os casos de uso são desenvolvidos para modelar os requisitos. Ele estabelece e mantém concordância com os clientes e outros envolvidos sobre o que sistema deve fazer e define também fronteiras do sistema e fornece uma base para cálculo de custo e tempo.

| ESTABELECE E MANTER CONCORDÂNCIA COM OS CLIENTES E OUTROS ENVOLVIDOS SOBRE O QUE O SISTEMA DEVE FAZER. | |
|--|---|
| FINALIDADES | Oferecer aos desenvolvedores do sistema uma compreensão melhor dos requisitos do sistema. |
| | Definir as fronteiras do sistema (ou delimitar o sistema). |
| | Fornecer uma base para planejar o conteúdo técnico das iterações. |
| | Fornecer uma base para estimar o custo e o tempo de desenvolvimento do sistema. |
| | Definir uma interface de usuário para o sistema, focando nas necessidades e metas dos usuários. |

Os artefatos de Requisitos capturam e apresentam informações usadas para definir os recursos necessários do sistema.



5.3 – Análise e Projeto/Design

Um modelo de projeto é criado e documentado usando modelos de arquitetura, de componente, de objeto e também de sequência. Transforma os requisitos em um projeto do sistema a ser criado, desenvolvendo uma arquitetura básica para o sistema e adaptando o projeto ao seu ambiente.

| FINALIDADES | |
|-------------|--|
| | TRANSFORMAR OS REQUISITOS EM UM DESIGN DO SISTEMA A SER CRIADO. |
| | Desenvolver uma arquitetura sofisticada para o sistema. |
| | Adaptar o design para que corresponda ao ambiente de implementação, projetando-o para fins de desempenho |

Os artefatos de Análise e Design capturam e apresentam informações relacionadas à solução dos problemas propostos.



5.4 – Implementação

Os componentes de sistema são implementados e estruturados em subsistemas de implementação, organizados geralmente em camadas. Implementa classes e objetos em termos de componentes testados e desenvolvidos como unidades. Além disso, integra os resultados produzidos ao sistema executável.

| DEFINIR A ORGANIZAÇÃO DO CÓDIGO EM TERMOS DE SUBSISTEMAS DE IMPLEMENTAÇÃO ORGANIZADOS EM CAMADAS. | |
|---|---|
| FINALIDADES | Implementar classes e objetos em termos de componentes (arquivo-fonte, binários, executáveis e outros). |
| | Testar os componentes desenvolvidos como unidades. |
| | Integrar os resultados produzidos por implementadores individuais (ou equipes) ao sistema executável. |

Os artefatos de Implementação capturam e apresentam a realização da solução proposta no conjunto de Análise e Design.

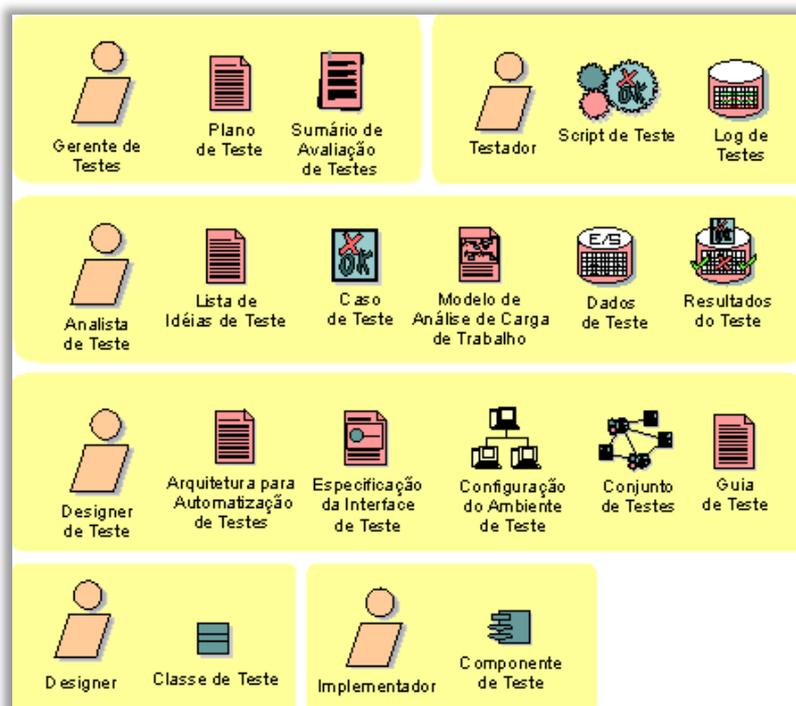


5.5 – Teste

O teste é um processo iterativo realizado em conjunto com a implementação. Localiza e documenta defeitos na qualidade do software, relatando a forma geral da qualidade observada no software. Valida suposições e funções, e também verifica se os requisitos foram implementados adequadamente.

| LOCALIZAR E DOCUMENTAR DEFEITOS NA QUALIDADE DO SOFTWARE. | |
|---|---|
| FINALIDADES | Avisar de forma geral sobre a qualidade observada no software. |
| | Validar as suposições feitas nas especificações de design e requisito através de demonstração concreta. |
| | Validar as funções do software conforme projetadas. |
| | Verificar se os requisitos foram implementados de forma adequada. |

Os artefatos desenvolvidos como produtos das atividades de Teste e avaliação agrupados por papel de responsabilidade.



5.6 – Implantação

Uma versão do produto é criada, distribuída aos usuários e instalada no local de trabalho. Garantem que o produto será disponibilizado aos usuários finais, por meio de uma instalação personalizada em formato compacto e com acesso correto. Galera, trocando em miúdos, é colocar o sistema em produção.

| FINALIDADES | COORDENAR E GERENCIAR OS TESTES BETA E TESTES DE ACEITAÇÃO. |
|-------------|---|
| | Desenvolver artefatos de instalação e materiais de treinamento. |
| | Liberar o sistema em produção para os usuários finais. |

Os artefatos de Implantação capturam e apresentam informações relacionadas à transição do sistema para o ambiente de produção.



5.7 – Gerenciamento de Configuração e Mudanças

Este workflow de apoio controla e gerencia mudanças e mantém a integridade entre eles e suas versões. Para tal, identificam-se os itens de configuração, definem-se restrições de mudança e realizam-se auditorias, evitando conflitos – além de criar baselines².

| IDENTIFICAR E CONTROLAR ITENS DE CONFIGURAÇÃO. | |
|--|--|
| FINALIDADES | Restringir as mudanças nesses itens de configuração. |
| | Auditar as mudanças nesses itens de configuração. |
| | Evitar confusões de: atualização simultânea; notificação limitada; e várias versões. |

O conjunto de artefatos dessa disciplina captura e apresenta informações relacionadas à disciplina de gerenciamento de configuração e mudança.



² A Baseline (ou Linha de Base) é uma imagem da versão de um artefato, isto é, ela é composta pela definição das versões dos artefatos que compõem o produto em dado momento. Essa não é a mesma *baseline* presente na atividade Desenvolvimento do Cronograma no Gerenciamento de Projetos (PMI) – mais especificamente situada na área de conhecimento Gerenciamento de Tempo, sob o processo de Planejamento. Este baseline (PMI) é um cronograma que considera a alocação dos recursos disponíveis, além do caminho crítico. Pelo fato dos termos serem iguais, a confusão pode ocorrer. O baseline do PMI é similar ao artefato Plano de Iteração. Já o baseline do RUP é associado à situação da versão dos artefatos em determinado momento, quando é disponibilizada uma versão para o cliente.

5.8 – Gerenciamento de Projetos

Este workflow de apoio fornece diretrizes práticas para planejar, montar equipes, executar e monitorar os projetos. Entretanto, não é tão amplo quanto o PMBOK, por exemplo. Apesar de incluir gerenciamento de riscos, não contém gerenciamento de pessoal, orçamento e contrato.

| FORNECER UM FRAMEWORK PARA GERENCIAR PROJETOS INTENSIVOS DE SOFTWARE. | |
|---|---|
| FINALIDADES | Fornecer diretrizes práticas para planejar, montar a equipe, executar e monitorar os projetos. |
| | Fornecer um framework de gerenciamento de risco. |
| | Não é finalidade cobrir Gestão de Pessoal (Contratação, Treinamento e Ensino); Gestão de Orçamento (Definição, Alocação, etc); e Gestão de Contratos (com fornecedores e clientes). |

Os artefatos de Gerenciamento de Projeto capturam os artefatos associados ao planejamento e à execução do projeto e do processo.



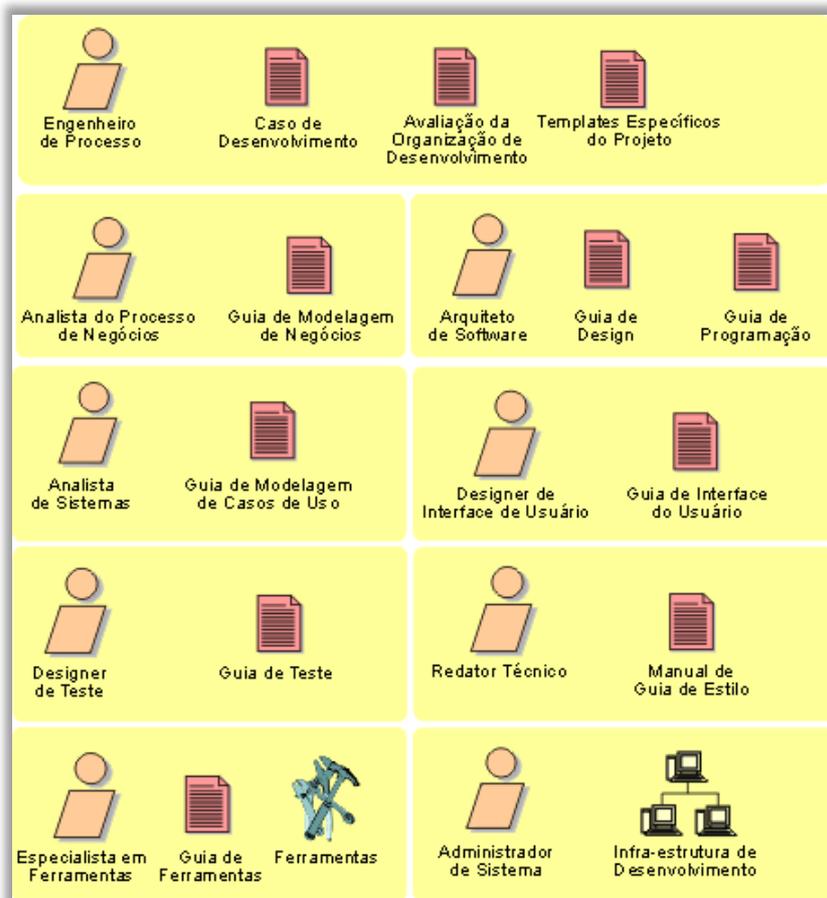
5.9 – Ambiente

Este workflow de apoio está relacionado à disponibilização de ferramentas apropriadas de software para a equipe de desenvolvimento. Ele concentra-se nas atividades necessárias à configuração (tanto de hardware quanto de software) do processo para um projeto.

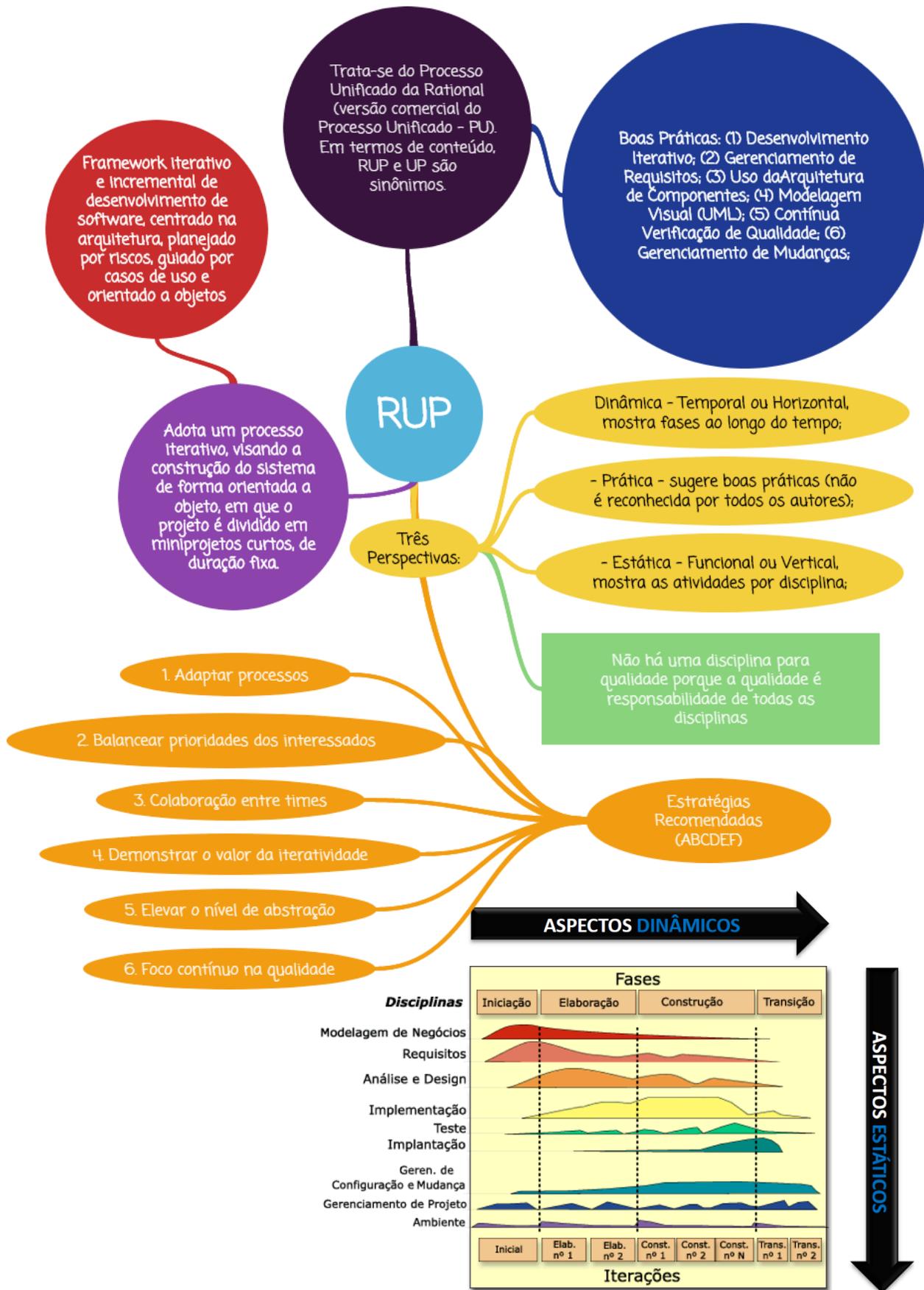
FINALIDADES

Oferecer processos e ferramentas que darão suporte à equipe de desenvolvimento.

O conjunto de artefatos de Ambiente apresenta artefatos usados como instruções durante o desenvolvimento, a fim de garantir a consistência dos artefatos.



6 – Esquema



EXERCÍCIOS COMENTADOS

1. **(CESPE – 2019 – MPC/AC – Assistente Ministerial)** O processo unificado (RUP) reúne boas práticas de especificação e de projeto de sistemas, sendo um modelo genérico de processo organizado em fases que podem gerar um conjunto de produtos de trabalho. Considerando isso, assinale a opção que identifica a fase do RUP na qual devem estar incluídos o refinamento e a expansão dos casos de uso preliminares, dos requisitos não funcionais e da descrição da arquitetura do software.

- a) concepção
- b) construção
- c) elaboração
- d) produção
- e) transição

Comentários:

Refinamento e expansão dos casos de uso preliminares? Descrição da arquitetura do software? Só pode estar tratando da fase de Elaboração! Nessa fase, é o momento de revisar e detalhar o que foi levantado na fase de concepção.

Gabarito: Letra C

2. **(CESPE – 2019 – SEMEF – Assistente Técnico)** Considerando o uso do RUP (Rational Unified Process), deve-se considerar que, segundo essa técnica,

- a) a disciplina Modelagem de Negócio apresenta maior atividade na fase de Construção.
- b) o número de iterações em cada uma de suas quatro fases é variável, conforme o projeto.
- c) a disciplina Requisitos apresenta menor atividade na fase de Concepção (Inception).
- d) a disciplina de Teste não é executada na fase de Elaboração.
- e) a disciplina Configuração e Gerenciamento de Mudanças não é executada na fase de Elaboração.

Comentários:

(a) Errado, na verdade é a fase de Concepção; (b) Correto, é possível haver mais de uma iteração por fase e esse número pode variar; (c) Errado, na verdade apresenta maior atividade; (d) Errado, essa disciplina é – sim – executada na fase de Elaboração; (e) Errado, essa disciplina é – sim – executada na fase de Elaboração.

Gabarito: Letra B



3. (CESPE – 2019 – SLU/DF – Analista de Gestão) No processo unificado, requisitos é a disciplina que demanda maior esforço nas fases de elaboração e construção de software.

Comentários:

Na verdade, pelo Gráfico das Baleias, a disciplina que demanda maior esforço é a de Análise e Design.

Gabarito: Errado

4. (UFPR – 2018 – COREN – Analista de TI) De acordo com o Processo Unificado (RUP), que resultado deve ser alcançado ao término da fase de elaboração?
- a) A integração do sistema nas plataformas definidas pelo cliente.
 - b) A descrição da arquitetura do sistema.
 - c) A disponibilização das versões finais dos manuais do usuário.
 - d) A disponibilização da versão inicial do documento de visão do sistema.
 - e) O encerramento dos testes para validar o sistema em relação às expectativas de seus usuários.

Comentários:

(a) Errado, fase de transição; (b) Correto, fase de elaboração; (c) Errado, fase de transição; (d) Errado, fase de concepção; (e) Errado, fase de construção.

Gabarito: Letra B

5. (FCC - 2009 - PGE-RJ - Técnico Superior de Análise de Sistemas e Métodos) O Processo Unificado se caracteriza por ser um:
- a) ciclo de desenvolvimento de software em cascata, centrado na arquitetura e guiado pela modelagem de negócio.
 - b) ciclo de desenvolvimento de software sequencial com todos os entregáveis produzidos em uma só fase.
 - c) processo de software específico para reengenharia, centrado em objetos e orientado a casos de uso.
 - d) processo de software iterativo e incremental, centrado na arquitetura e guiado por casos de uso.



e) processo de software interativo, centrado na temporalidade dos negócios e orientado a eventos.

Comentários:

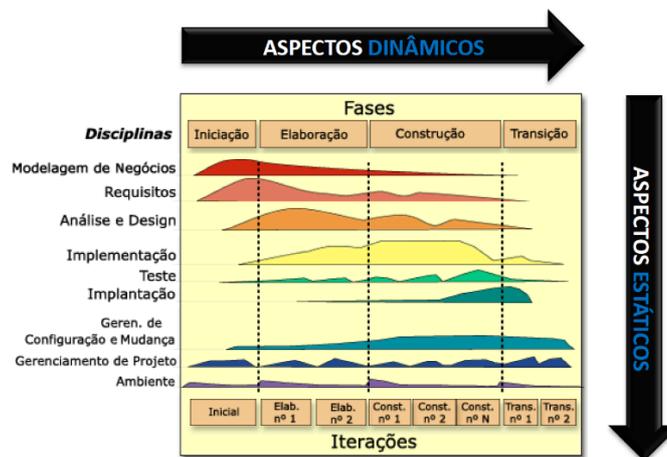
O Rational Unified Process (RUP) é um **framework iterativo** e incremental de desenvolvimento de software, **centrado na arquitetura, planejado por riscos, guiado por casos de uso e orientado a objetos**, criado pela Rational, empresa adquirida pela IBM. Infelizmente as questões continuam trocando iterativo com interativo :(

Gabarito: Letra D

6. (FCC - 2011 - TRE-AP - Analista Judiciário - Análise de Sistemas) Considerando a representação gráfica dos fluxos de processos essenciais e fases, Implementation tem participação mais significativa em:

- a) Construction.
- b) Elaboration.
- c) Requirements.
- d) Deployment.
- e) Transition.

Comentários:



Basta olhar o Gráfico das Baleias: na verdade, na Construção (Construction) que se tem participação mais significativa da Implementação (Implementation).

Gabarito: Letra A

7. (FCC - 2008 - TRF - 5ª REGIÃO - Analista Judiciário - Tecnologia da Informação – A) O RUP possibilita o desenvolvimento incremental e iterativo, guiado por casos de uso e centrado na arquitetura do sistema.



Comentários:

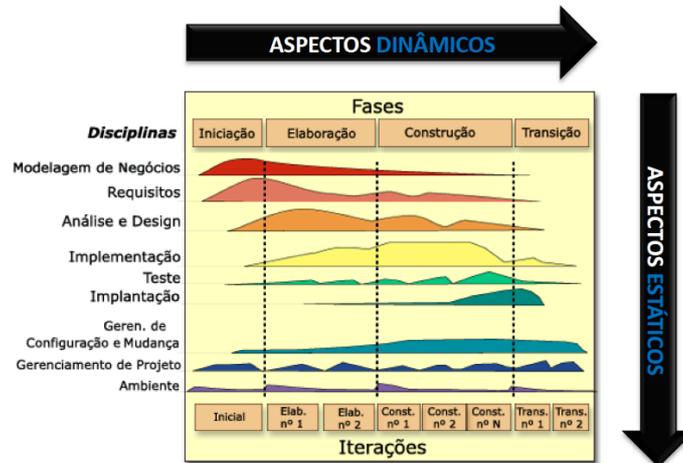
O Rational Unified Process (RUP) é um **framework iterativo e incremental** de desenvolvimento de software, **centrado na arquitetura**, planejado por riscos, **guiado por casos de uso** e orientado a objetos, criado pela Rational, empresa adquirida pela IBM. Mais uma vez: cansei se ver as bancas errando isso e não corrigindo, logo não liguem para esse erro ortográfico :(

Gabarito: Correto

8. (FCC - 2013 - DPE-SP - Agente de Defensoria - Analista de Sistemas) O modelo estabelecido para o RUP (Rational Unified Process) é composto por quatro fases, denominadas:

- a) Requisitos, Implantação, Testes e Ambiente.
- b) Análise, Projeto, Negócios e Comissionamento.
- c) Concepção, Elaboração, Construção e Transição.
- d) Partição, Integração, Testes e Operação.
- e) Planejamento, Codificação, Integração e Configuração.

Comentários:



As fases são: Concepção (ou Iniciação), Elaboração, Construção e Transição.

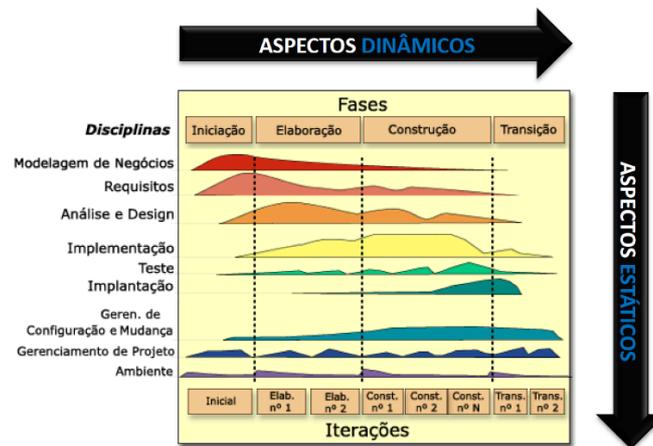
Gabarito: Letra C

9. (FCC - 2012 - MPE-PE - Analista Ministerial - Informática) A visão estática do RUP prioriza as atividades que ocorrem durante o processo de desenvolvimento. Na descrição do RUP, essas são chamadas de workflows. Existem seis workflows centrais, identificadas no processo e três de apoio, dentre os quais é possível citar os workflows de:



- a) Meio ambiente e Gerenciamento de projeto.
- b) Concepção e Construção.
- c) Transição e Iteração.
- d) Plano de desenvolvimento e Conceito de operação.
- e) Análise de Riscos e Operação e manutenção.

Comentários:



São três workflows de suporte: Ambiente (alguns traduzem como Meio Ambiente), Gerenciamento de Projeto e Gerenciamento de Configuração e Mudança.

Gabarito: Letra A

10. (FCC - 2012 - TRT - 6ª Região (PE) - Analista Judiciário - Tecnologia da Informação) A perspectiva prática sobre o RUP descreve as boas práticas da engenharia de software que são recomendadas para uso no desenvolvimento de sistemas. Dentre as práticas fundamentais recomendadas incluem-se:

- a) utilizar a arquitetura em cascata e efetuar programação em pares.
- b) definir a funcionalidade do protótipo e avaliar o protótipo.
- c) definir o esboço dos requisitos e estabelecer objetivos do protótipo.
- d) utilizar arquiteturas baseadas em componentes e modelar os softwares visualmente.
- e) desenvolver teste inicial a partir de cenários e utilizar frameworks de testes automatizados.

Comentários:

- **Uso da Arquitetura de Componentes:** componentes são grupos de códigos coesos, na forma de código fonte ou executável, com interfaces bem definidas e comportamentos que fornecem forte encapsulamento do conteúdo e são, portanto, substituíveis. De maneira informal, um componente pode ser descrito como algo que executa uma funcionalidade sem conhecer seus detalhes de implementação.



- **Modelagem Visual (UML):** é o uso de notações de design gráficas e textuais, semanticamente ricas, para capturar designs de software. Uma notação, como a UML, permite que o nível de abstração seja aumentado, enquanto mantém sintaxe e semântica rígidas. Assim, a comunicação na equipe melhora à medida que o design é formado e revisado, permitindo ao leitor raciocinar sobre ele e fornecendo uma base não ambígua para a implementação.

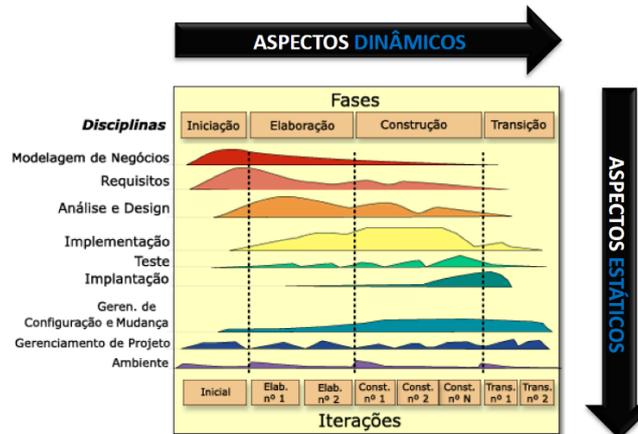
Lembram-se das melhores práticas? Modelagem Visual e Arquitetura de Componentes fazem parte delas!

Gabarito: Letra D

11. (FCC - 2011 - TCE-PR - Analista de Controle - Informática) A conclusão da análise, do design, do desenvolvimento e do teste de todas as funcionalidades necessárias ao sistema, no processo RUP, é um dos objetivos da fase de:

- a) iniciação.
- b) elaboração.
- c) integração.
- d) construção.
- e) transição.

Comentários:



Galera, questão chata! Observem que, conforme o gráfico, todas essas disciplinas acabam na Transição, no entanto isso não é obrigatório! Elas podem acabar na Fase de Construção. Na verdade, isso é o ideal, isto é, terminar toda análise, design, implementação e testes na Fase de Construção (é o objetivo dessa fase) e, por fim, só fazer a transição. No entanto, conforme mostra o gráfico, pode acontecer de esses fluxos terminarem apenas na Fase de Transição. Acredito que cerca de 70% das pessoas que fizeram essa questão caíram nessa pegadinha (eu, na minha época de concurseiro, também).

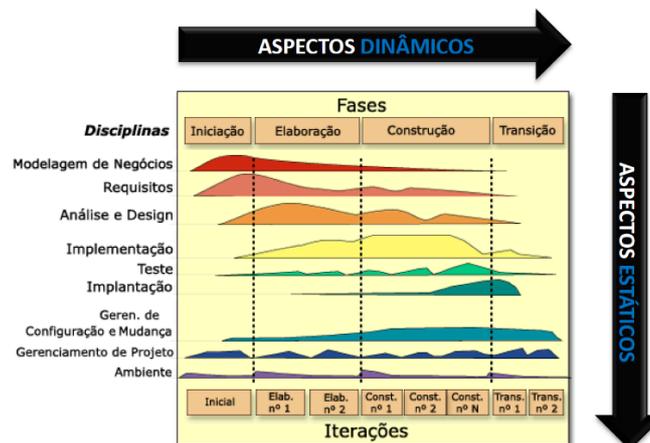
Gabarito: Letra D



12. (FCC - 2011 - TRE-AP - Analista Judiciário - Análise de Sistemas) Considerando a representação gráfica dos fluxos de processos essenciais e fases, Implementation tem participação mais significativa em:

- a) Construction.
- b) Elaboration.
- c) Requirements.
- d) Deployment.
- e) Transition.

Comentários:



Conforme o gráfico, a Implementação é mais concentrada na fase de Construção!

Gabarito: Letra A

13. (FCC - 2011 - INFRAERO - Analista de Sistemas - Gestão de TI) Uma disciplina do RUP que tem como uma de suas finalidades "assegurar que os clientes, usuários e desenvolvedores tenham um entendimento comum da organização-alvo", a qual se relaciona com a disciplina Ambiente. Trata-se de:

- a) Requisitos.
- b) Análise e Design.
- c) Modelagem de Negócios.
- d) Gerenciamento de Configuração e Mudança.
- e) Gerenciamento de Projetos.

Comentários:

- **Modelagem de Negócios:** os processos de negócios são modelados usando casos de uso de negócios. Busca um entendimento comum da estrutura e da dinâmica da organização na qual um sistema deve ser implantado, além dos problemas atuais da organização-alvo e identificar as



possibilidades de melhoria – está bastante ligada ao Plano Estratégico Institucional da organização.

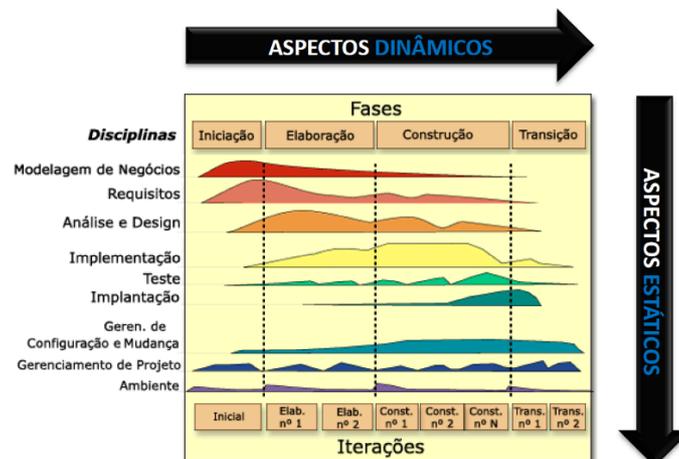
Galera, falou em **entendimento da organização** é Modelagem de Negócios!

Gabarito: Letra C

14. (FCC - 2011 - TRE-AP - Analista Judiciário - Análise de Sistemas) É um dos core “supporting” workflows, o:

- a) Test.
- b) Inception.
- c) Analysis & Design.
- d) Business modeling.
- e) Configuration and Change Management.

Comentários:



São três disciplinas de suporte! Entre elas, temos a *Configuration and Change Management* (Gerenciamento de Configuração e Mudanças).

Gabarito: Letra E

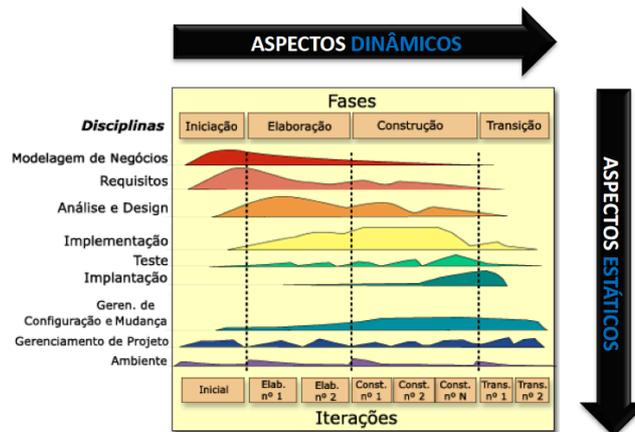
15. (FCC - 2011 - TRT - 1ª REGIÃO (RJ) - Analista Judiciário - Tecnologia da Informação) No Processo Unificado, a maior porção do core workflow denominado Analysis é executada na fase:

- a) Elaboration.
- b) Construction.
- c) Implementation.
- d) Inception.



e) Transition.

Comentários:



Observem que a Análise é mais focada na fase de Elaboração (*Elaboration*).

Gabarito: Letra A

16. (FCC - 2011 - TRE-RN - Analista Judiciário - Análise de Sistemas) São produtos da fase de elaboração do RUP:

- documento de visão e produto de software integrado.
- descrição da arquitetura do software e lista de riscos revisada.
- manual do usuário e base de dados operacionais convertidas.
- lista de riscos revisada e base de dados operacionais convertidas.
- produto de software integrado e descrição da arquitetura do software.

Comentários:

Elaboração: tem o objetivo de desenvolver um entendimento do domínio do problema, estabelecer um framework de arquitetura para o sistema, desenvolver o plano de projeto e identificar os riscos principais do projeto. Seus artefatos mais importantes são: Protótipos; **Lista de Riscos**; **Documento de Arquitetura de Software**; Modelo de Projeto; Modelo de Dados. Infelizmente, a questão foi bastante decorada – decorar esse tipo de coisa é inviável!

Gabarito: Letra B

17. (FCC - 2011 - TRE-RN - Técnico Judiciário - Programação de Sistemas) O RUP produz artefatos:

- na fase de Transição, apenas.
- em todas as suas fases.
- na fase de Concepção, apenas.



- d) na fase de Elaboração, apenas.
- e) na fase de Construção, apenas.

Comentários:

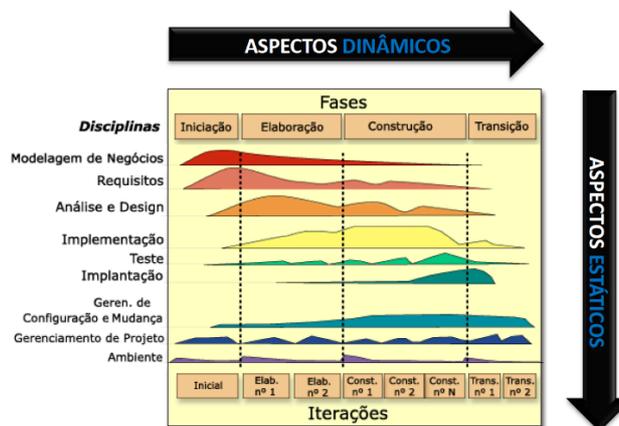
O RUP produz artefatos em todas as suas fases.

Gabarito: Letra B

18.(FCC - 2011 - CAIXA - Analista de Sistemas) No gráfico do modelo iterativo RUP, a modelagem do negócio e a implementação têm sua maior porção representada, respectivamente, nas fases:

- a) Inception e Elaboration.
- b) Inception e Construction.
- c) Elaboration e Construction.
- d) Requirements e Construction.
- e) Elaboration e Transition.

Comentários:



Observem que se trata de Modelagem de Negócio (Inception) e Implementação (Construction).

Gabarito: Letra B

19.(FCC - 2011 - TRE-RN - Técnico Judiciário - Programação de Sistemas) No Processo Unificado, uma descrição da arquitetura do software, um documento de visão e um modelo de projeto são aplicáveis, respectivamente, nas fases:

- a) elaboração, concepção e construção.
- b) concepção, concepção e elaboração.
- c) construção, transição e concepção.



- d) transição, construção e construção.
- e) concepção, elaboração e transição.

Comentários:

O Artefato *Modelo de Projetos* se encontra na fase de Elaboração! Na Construção, existe o *Template Específico de Projeto*, que nada tem a ver com o *Modelo de Projetos*. No entanto, a questão afirma que pode ser aplicada na fase de Construção. Apesar de não ser um artefato da fase de Construção, ela pode – sim – ser aplicada nessa fase. Em suma: Descrição da Arquitetura de Software (Elaboração); Documento de Visão (Concepção); e Modelo de Projeto (Construção).

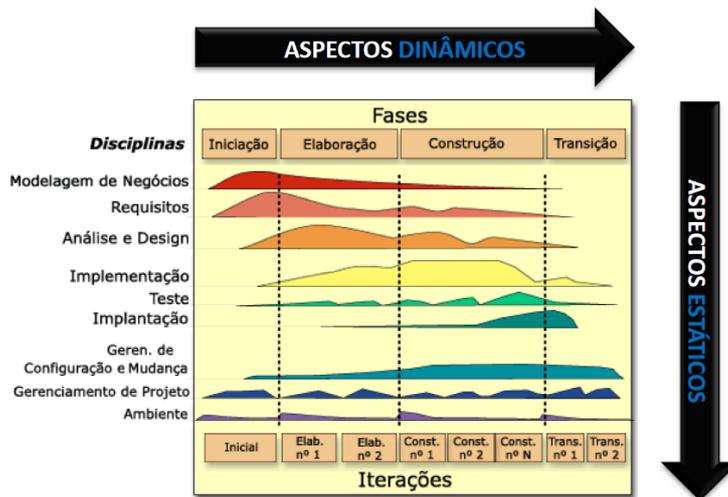
Gabarito: Letra A

20. (FCC - 2010 - MPE-RN - Analista de Tecnologia da Informação - Engenharia de Software)

São respectivamente disciplina (Core Process Workflow) e fase (Phase) do RUP:

- a) Concepção e Implantação.
- b) Implementação e Elaboração.
- c) Implantação e Requisitos.
- d) Requisitos e Modelagem de Negócios.
- e) Implementação e Teste.

Comentários:



A Disciplina é Implementação e a Fase é a Elaboração.

Gabarito: Letra B

21. (FCC - 2010 - MPE-RN - Analista de Tecnologia da Informação - Engenharia de Software)

No RUP, a ênfase no escopo do sistema está na fase de:

- a) Concepção.



- b) Implementação.
- c) Elaboração.
- d) Transição.
- e) Construção.

Comentários:

Em geral, o escopo do sistema é definido na disciplina de Requisitos, que está na fase de Concepção. Os agentes que interagem com o sistema são identificados e os casos de uso são desenvolvidos para modelar os requisitos. Eles estabelecem e mantêm concordância com os clientes e outros envolvidos sobre o que sistema deve fazer, além de definir fronteiras do sistema, fornecendo uma base para cálculo de custo e tempo.

Gabarito: Letra A

22. (FCC - 2010 - MPE-RN - Analista de Tecnologia da Informação - Engenharia de Software)

No RUP, a ênfase em arquitetura está na fase de:

- a) Transição.
- b) Modelagem de Negócio.
- c) Implantação.
- d) Implementação.
- e) Elaboração.

Comentários:

| FASE | MARCO |
|------------|--|
| INICIAÇÃO | Escopo ou objetivos do ciclo de vida. |
| ELABORAÇÃO | Arquitetura estabilizada do ciclo de vida. |
| CONSTRUÇÃO | Capacidade operacional inicial. |
| TRANSIÇÃO | Lançamento (ou Release) do produto. |

Observem que é o marco da fase de Elaboração, apesar de já haver um esboço na Concepção!

Gabarito: Letra E

23. (FCC - 2010 - MPE-RN - Analista de Tecnologia da Informação - Engenharia de Software)

No RUP, Project Management e Environment são:

- a) Phases.
- b) Core Process Workflows.
- c) Metrics.
- d) Core Supporting Workflows.



e) Analysis & Design Process.

Comentários:

São seis disciplinas principais e três disciplinas de apoio ou infraestrutura. As disciplinas de Apoio e Infraestrutura (ou *Core Supporting Workflows*) são: Gerenciamento de Projetos (*Project Management*), Ambiente (*Environment*) e Gerenciamento de Configuração e Mudança (*Software Configuration Management*).

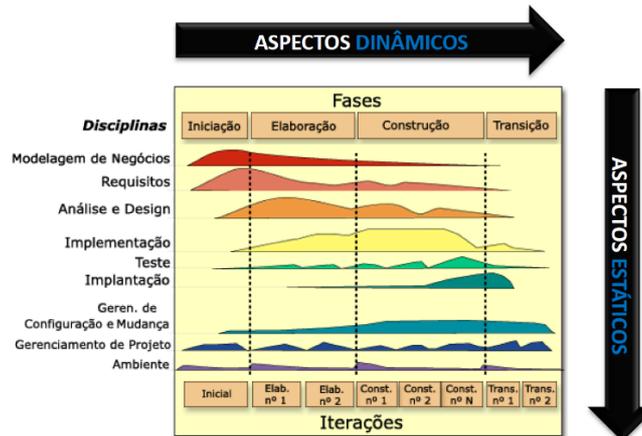
Gabarito: Letra D

24. (FCC - 2010 - MPE-RN - Analista de Tecnologia da Informação - Engenharia de Software)

No Processo Unificado (UP), o fluxo de trabalho Análise, tem forte concentração na fase de:

- a) Implementação.
- b) Elaboração.
- c) Construção.
- d) Concepção.
- e) Transição.

Comentários:



Observem pela imagem que o foco maior é na **Fase de Elaboração!**

Gabarito: Letra B

25. (FCC - 2010 - AL-SP - Agente Técnico Legislativo Especializado - Tecnologia da Informação) No Processo Unificado de desenvolvimento de software, Requisitos e Teste são:

- a) designações das fases do ciclo de vida.
- b) designações dos fluxos de trabalho.
- c) disciplinas consideradas apenas na quarta fase.



- d) disciplinas consideradas apenas na segunda fase.
- e) disciplinas consideradas apenas na terceira e na quarta fases.

Comentários:

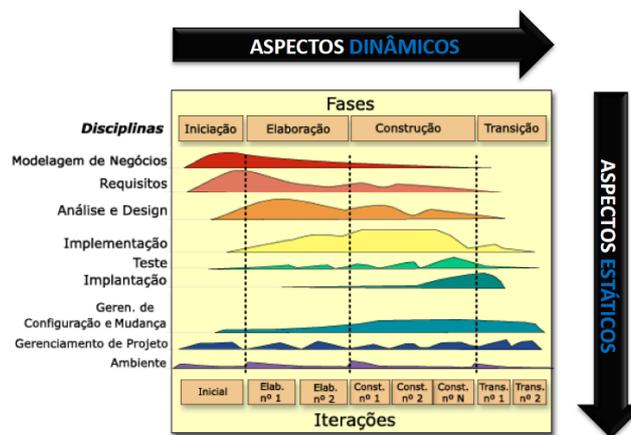
Requisitos e Teste são designações de **fluxos de trabalho** – também chamadas de disciplinas ou workflows.

Gabarito: Letra B

26.(FCC - 2010 - AL-SP - Agente Técnico Legislativo Especializado - Tecnologia da Informação) São fases do Processo Unificado de desenvolvimento de software:

- a) Concepção e Transição.
- b) Elaboração e Análise.
- c) Concepção e Implementação.
- d) Projeto e Implementação.
- e) Análise e Projeto.

Comentários:



Basta ver a figura: Concepção (ou Iniciação) e Transição são **fases** do Processo Unificado.

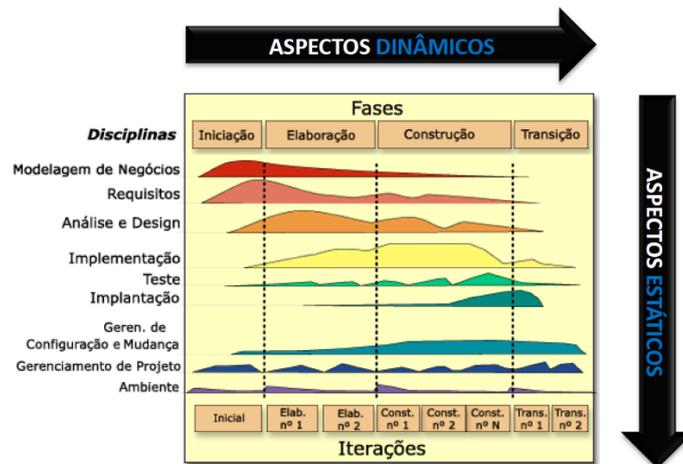
Gabarito: Letra A

27. (FCC - 2005 – TRT/13 – Analista de Sistemas) São quatro fases do RUP:

- a) Transaction, Implementation, Requirements e Configuration.
- b) Project Management, Transaction, Construction e Interaction.
- c) Deployment, Transition, Design e Requirements.
- d) Inception, Construction, Elaboration e Transition.
- e) Analysis, Design, Elaboration e Inception.



Comentários:



Observem a imagem: trata-se de *Inception, Construction, Elaboration* e *Transition*! Claro que não é nessa ordem, mas não foi isso que a questão perguntou!

Gabarito: Letra D

28. (FCC - 2006 – BACEN – Analista de Sistemas) O RUP (Rational Unified Process):

- a) tem sua arquitetura desenvolvida em paralelo aos use cases, englobando apenas os aspectos estáticos do software.
- b) é um processo estático que viabiliza a definição seqüencial de um problema e tem como única finalidade o teste do produto final.
- c) estabelece, em sua fase de concepção, um plano de projeto e uma arquitetura sólida, por meio do desenvolvimento estático de software, dentro do seu ciclo de vida.
- d) é o modelo de processo que estabelece a topologia do hardware necessária para executar o sistema.
- e) dá ênfase à criação e à manutenção de modelos e tem sua atividade de desenvolvimento orientada por use case como parte funcional do sistema que agrega valor ao resultado do software.

Comentários:

(a) Errado, tenta-se combinar as perspectivas estática e dinâmica do software; (b) Errado, é um processo dinâmico e tem diversas finalidades; (c) Errado, um esboço de arquitetura – a arquitetura sólida ocorre ao fim da elaboração. Ademais, é um processo dinâmico de



desenvolvimento de software; (d) Errado, não faz sentido algum; (e) Correto! Ele é um framework iterativo e incremental de desenvolvimento de software, centrado na arquitetura, planejado por riscos, guiado por casos de uso e orientado a objetos, criado pela Rational, empresa adquirida pela IBM.

Gabarito: Letra E

29. (FCC - 2006 – BACEN – Analista de Sistemas) No RUP,

- a) os artefatos são produzidos nas fases de Elaboração e Construção e os documentos nas fases de Requisitos e Implementação.
- b) podem existir mais de trinta papéis (analista de sistemas, projetista, projetista de banco de dados, entre outros), a serem representados pelos profissionais, sendo que cada pessoa pode representar apenas um papel.
- c) os workflows são descritos sumariamente, apresentando os aspectos gerais das tarefas e os principais profissionais envolvidos.
- d) a iteração pode ocorrer uma ou mais vezes, em cada uma de suas quatro fases básicas, ou até mesmo, em parte delas.
- e) a fase da elaboração visa estabelecer o escopo e a viabilidade econômica do projeto.

Comentários:

(a) Errado. Artefatos e documentos são produzidos em todas as fases. Além disso, Requisitos e Implementação são disciplinas; (b) Errado. Existem 32 papéis, mas pessoas não são papéis. Na verdade, uma pessoa pode ter vários papéis e um papel pode ser desempenhado por várias pessoas; (c) Errado. Workflows são descritos detalhadamente; (d) Correto. A iteração pode realmente ocorrer uma ou mais vezes; (e) Errado. Isso ocorre na fase de Iniciação/Concepção.

Gabarito: Letra D

30. (FCC - 2007 – TRE/SE – Analista de Sistemas) Considere as afirmativas abaixo.

- I. O RUP é um processo iterativo.
- II. Sob orientação do RUP, o desenvolvimento é centrado na arquitetura.
- III. Sob orientação do RUP, as atividades desenvolvidas são orientadas por casos de uso.

É correto o que se afirma em:

- a) I, II e III.



- b) I e III, apenas.
- c) I e II, apenas.
- d) III, apenas.
- e) I, apenas.

Comentários:

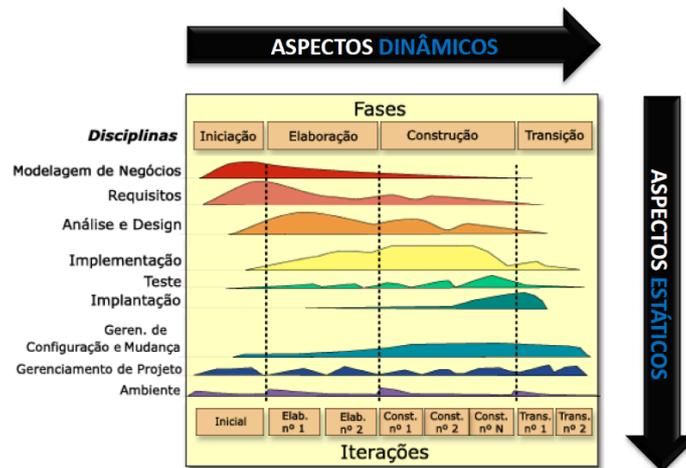
O Rational Unified Process (RUP) é um framework **iterativo** e incremental de desenvolvimento de software, **centrado na arquitetura**, planejado por riscos, **guiado por casos de uso** e orientado a objetos, criado pela Rational, empresa adquirida pela IBM. Logo, todos os itens estão corretos!

Gabarito: Letra A

31. (FCC - 2007 – TRE/SE – Analista de Sistemas) No RUP, a maior quantidade da disciplina Análise e Projeto é encontrada na fase de:

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

Comentários:



Vamos de cara eliminar a terceira opção, visto que se trata de uma disciplina e, não, uma fase. Daí, basta observar do gráfico das baleias e notar que a maior parte da disciplina Análise e Projeto se encontra na **Fase de Elaboração**.

Gabarito: Letra E

32. (FCC - 2007 – TRE/SE – Analista de Sistemas) Em um projeto baseado no processo RUP, a ênfase à arquitetura do sistema é dada na fase de:



- a) Requisitos.
- b) Construção.
- c) Transição.
- d) Concepção.
- e) Elaboração.

Comentários:

| FASE | MARCO |
|------------|--|
| INICIAÇÃO | Escopo ou objetivos do ciclo de vida. |
| ELABORAÇÃO | Arquitetura estabilizada do ciclo de vida. |
| CONSTRUÇÃO | Capacidade operacional inicial. |
| TRANSIÇÃO | Lançamento (ou Release) do produto. |

A ênfase à arquitetura do sistema é dada na fase de Elaboração! A arquitetura consolidada é, inclusive, o marco da fase de Elaboração.

Gabarito: Letra E

33. (FCC - 2007 – TRE/SE – Analista de Sistemas) Riscos reduzidos mais cedo porque os elementos são integrados progressivamente é uma vantagem, no processo RUP, da abordagem:

- a) Iterativa.
- b) Top-down.
- c) Incremental.
- d) Interativa.
- e) Em cascata.

Comentários:

A abordagem iterativa do RUP auxilia na redução de riscos do desenvolvimento de software.

Gabarito: Letra A

34. (FCC - 2007 – TRF – Analista de Sistemas) No RUP, a análise do domínio do problema, o desenvolvimento do plano do projeto, o estabelecimento de uma sólida base arquitetural e a eliminação dos elementos de mais alto risco do projeto são objetivos:

- a) da Elaboration Phase.
- b) da Construction Phase.
- c) da Inception Phase.



- d) da Transition Phase.
- e) do Deployment.

Comentários:

Galera, vocês devem ter notado que a FCC tem a mania de fazer questões em inglês! De todo modo, falou em **sólida base arquitetural**, trata-se da **Elaboration Phase**.

Gabarito: Letra A

35. (FCC - 2007 – TRF – Analista de Sistemas) Dos nove core process workflow do RUP, são, respectivamente, dois core engineering e dois core supporting workflows:

- a) Implementation, Test, Project Management e Environment.
- b) Requirements, Configuration and Change Management, Project Management e Test.
- c) Configuration and Change Management, Implementation, Requirements e Test.
- d) Project Management, Requirements e Implementation.
- e) Business modeling, Requirements, Analysis & Design e Implamentation.

Comentários:

Puro decoreba! **Core Engineering:** *Implementation e Test*; **Core Supporting:** *Project Management e Environment* (lembrem-se do respectivamente).

Gabarito: Letra A

36. (FCC - 2007 – TRF – Analista de Sistemas) A maior parcela do fluxo dos processos fundamentais do RUP, correspondente à modelagem de negócio,

- a) está distribuída entre as fases Inception e Construction.
- b) está distribuída entre as fases Elaboration e Construction.
- c) situa-se na fase Construction.
- d) está distribuída entre as fases Inception e Elaboration.
- e) situa-se na fase Transition.

Comentários:

Basta lembrar dos gráficos das baleias: está distribuída entre as fases de **Inception e Elaboration**.

Gabarito: Letra D



37. (FUNCAB – 2010 – PRODAM/AM - Analista de Sistemas) Assinale a alternativa que NÃO representa uma característica do Processo Unificado (UP) original, descrito no livro *The Unified Software Development Process* (1999).

- a) Iterativo e Incremental.
- b) Centrado na arquitetura.
- c) Ágil.
- d) Orientado por casos de uso.
- e) Focado na mitigação precoce dos riscos.

Comentários:

O Rational Unified Process (RUP) é um framework iterativo e incremental de desenvolvimento de software, centrado na arquitetura, planejado por riscos, guiado por casos de uso e orientado a objetos, criado pela Rational, empresa adquirida pela IBM. Ele não é uma metodologia ágil, trata-se de uma metodologia tradicional de desenvolvimento de software.

Gabarito: Letra C

38. (CESGRANRIO - 2008 - TJ-RO - Analista Judiciário - Tecnologia da Informação – A) Três dos principais artefatos da disciplina Requisitos previstos pelo RUP são: Glossário, Especificações Suplementares e Modelo de Casos de Uso.

Comentários:



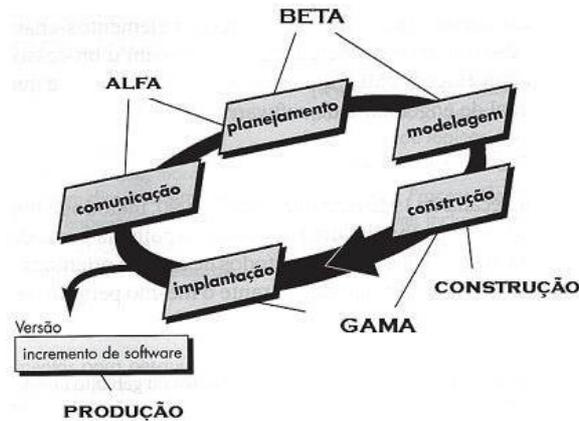
Infelizmente é uma questão decoreba! Vejam que esses três artefatos realmente pertencem à disciplina de Requisitos.

Gabarito: Correto

39. (FGV - 2010 – FIOCRUZ - Analista de Sistemas) O Processo Unificado, também chamado de Rational Unified Process (RUP), representa uma tentativa de apoiar-se nos melhores recursos e características dos modelos convencionais de processo de software, mas caracterizá-los de



um modo que implemente muitos dos melhores princípios de desenvolvimento ágeis. Ele enfatiza o importante papel da arquitetura de software e sugere um fluxo de processo iterativo e incremental, dando a sensação evolucionária que é essencial no desenvolvimento moderno de software. Nesse contexto, observe a figura abaixo.



As fases do processo unificado identificadas por ALFA, BETA e GAMA são denominados, respectivamente:

- requisitos, elaboração e implementação.
- requisitos, especificação e implantação.
- especificação, transição e implantação.
- concepção, elaboração e transição.
- concepção, projeto e implementação.

Comentários:

Primeiro, ele pede fases! *Requisitos, Implementação, Implantação, Especificação, Projeto* são fases do Processo Unificado? Não, as fases são Iniciação (ou Concepção), Elaboração, Construção e Transição. Logo, já eliminamos as letras A, B, C e E! Portanto sobrou apenas a quarta alternativa. *Vamos ver se ela está correta?* Comunicação e parte do Planejamento, de fato, fazem parte da Concepção; parte do Planejamento e Modelagem, de fato, fazem parte da Elaboração; por fim, parte da Construção e Implantação, de fato, fazem parte da Transição. *Bacana?*

Gabarito: Letra D

40. (FGV - 2009 - MEC - Analista de Sistemas - Especialista) O Rational Unified Process (RUP) é um exemplo de modelo de processo moderno que foi derivado do trabalho sobre a UML e do Processo Unificado de Desenvolvimento de Software associado. O RUP reconhece que os modelos convencionais de processo apresentam uma visão única de processo e é geralmente descrito a partir de três perspectivas:

- teórica, prática e funcional.



- b) dinâmica, estática e prática.
- c) teórica, prática e dinâmica.
- d) funcional, teórica e estática.
- e) teórica, prática e estática.

Comentários:

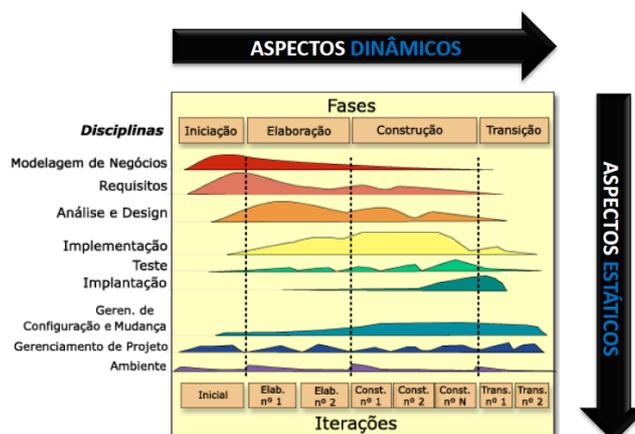
- **Dinâmica:** também conhecido como temporal ou horizontal, mostra as fases do modelo ao longo do tempo, além de iterações e marcos do projeto.
- **Estática:** também conhecido como funcional ou vertical, mostra as atividades realizadas no processo, componentes, disciplinas, artefatos e papéis de processo.
- **Prática:** sugere boas práticas a serem usadas no processo. Galera, essa dimensão não é reconhecida por todos os autores.

Gabarito: Letra B

41. (FGV - 2009 - MEC - Analista de Sistemas - Especialista) O RUP (Rational Unified Process) é uma metodologia de projeto de software criada pela Rational. O processo analítico do RUP divide o ciclo de vida de desenvolvimento de software em 4 fases. Assinale a alternativa que não indica uma fase do ciclo de vida de desenvolvimento de software, de acordo com o RUP.

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

Comentários:



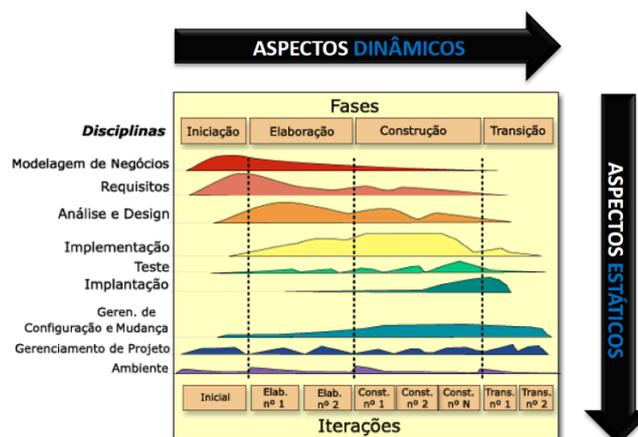
A questão pede qual NÃO indica uma fase do ciclo de vida. Logo, trata-se da Implantação.



42. (FGV - 2009 - MEC - Analista de Sistemas - Especialista) O RUP (Rational Unified Process) é um exemplo de modelo de processo moderno que foi derivado do trabalho sobre a UML e do Processo Unificado de Desenvolvimento de Software associado. O RUP é um modelo constituído por fases no processo de software. Ao contrário do modelo Cascata, no qual as fases coincidem com as atividades do processo, as fases do RUP estão relacionadas mais estritamente aos negócios do que aos assuntos técnicos. Assinale a alternativa que apresente as fases do modelo RUP.

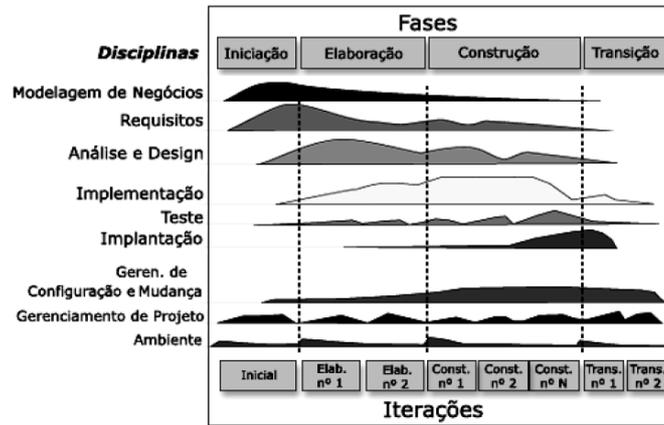
- a) Concepção, elaboração, construção e transição.
- b) Projeto, especificação, construção e transição.
- c) Especificação, construção e transição.
- d) Elaboração, construção e transição.
- e) Concepção, construção e Transição.

Comentários:



As fases são: Concepção (ou Iniciação), Elaboração, Construção e Transição.

43. (FGV - 2009 - MEC - Analista de Sistemas) A figura abaixo fornece uma visão geral do Rational Unified Process® – RUP, um processo de engenharia de software que representa uma abordagem baseada em disciplinas para atribuir tarefas e responsabilidades dentro de uma organização de desenvolvimento. Sua meta é garantir a produção de software de alta qualidade que atenda às necessidades dos usuários dentro de um cronograma e de um orçamento previsíveis.



Nessa abordagem, para fornecer uma base para o planejamento do conteúdo técnico de interações, uma visão de arquitetura chamada VISÃO DE CASOS DE USO é utilizada na disciplina denominada:

- a) Requisitos.
- b) Implantação.
- c) Análise e Design.
- d) Modelagem de Negócios.
- e) Gerenciamento de Projeto.

Comentários:

De acordo com o Wthreeex:

"Para fornecer uma base para o planejamento do conteúdo técnico de iterações, uma visão de arquitetura chamada visão de casos de uso é utilizada na disciplina Requisitos. Só existe uma visão de casos de uso do sistema, que ilustra os casos de uso e cenários que englobam o comportamento, as classes ou os riscos técnicos significativos do ponto de vista da arquitetura. A visão de casos de uso é refinada e considerada inicialmente em cada iteração."

Galera, falou em Casos de Uso, temos que pensar em cenários e comportamentos do sistema. Ok, mas isso se baseia em quê? Em Requisitos!

Gabarito: Letra A

44.(VUNESP - 2009 - CETESB - Analista de TI - Sistemas) Em RUP, a integração contínua, no contexto do ciclo de vida iterativo, significa:

- a) integração no fim de cada construção.
- b) integração no fim de cada iteração.
- c) integração no fim de cada transição.
- d) integração apenas no fim da elaboração e construção.



e) integrações apenas com as releases externas.

Comentários:

Galera, a Integração Contínua é uma prática típica da Metodologia Extreme Programming (XP)! Nesse contexto, ela busca integrar e criar o sistema muitas vezes ao dia, sempre que uma tarefa for concluída. No entanto, essa é uma prática compatível com o RUP: ele suporta essa prática através de builds no nível de sistema e de subsistema (dentro de uma iteração). Os componentes testados em unidades são integrados e testados no contexto do sistema emergente. Nem todas as práticas do XP são compatíveis com o RUP, por exemplo: Metáfora, Propriedade Coletiva, Refatoração e Pequenos Releases.

Gabarito: Letra B

45. (VUNESP - 2009 - CETESB - Analista de TI - Sistemas) Em RUP, a iteratividade ajuda no gerenciamento de recursos e custos porque:

- a) ajuda o gerente de projetos a alocar os requisitos por fase, com base na disponibilidade dos recursos durante o tempo de vida do projeto.
- b) ajuda o gerente de projetos a organizar recursos e custos por fases. Os artefatos do projeto evoluem conforme requerido por cada fase e aumenta-se a precisão na estimativa de custo fase a fase.
- c) permite mover os requisitos do projeto por entre as fases para garantir que os recursos possam ser mais bem gerenciados durante todo o ciclo de desenvolvimento.
- d) permite planejar as interações prévia e detalhadamente para todas as fases. Além disso, permite estabelecer custo e criar um perfil de uso de recurso para o projeto.
- e) permite que o gerente de projetos possa realizar orçamentos para os recursos de cada iteração.

Comentários:

(a) Errado. *Alocar os requisitos por fase? Requisitos?* Não! (b) Correto. A iteratividade reduz riscos, organizando recursos e custos. Como se repete diversas vezes, melhora a precisão de diversas estimativas. (c) Errado. Isso não tem relação com recursos e custos. (d) Errado. Interações não tem nada a ver com gerenciamento de recursos e custos. (e) Errado. Não vejo nada de errado nesse item, mas a Letra B é mais completa.

Gabarito: Letra B



46.(VUNESP - 2009 - CETESB - Analista de TI - Sistemas) Os planos de desenvolvimento de software cobertos em RUP são:

- a) Iteração, Construção e Publicação.
- b) Gerenciamento de Requisitos, Métricas de Produto e Garantia de Teste de Software.
- c) Gerenciamento de Recursos, Gerenciamento de Requisitos, Métricas e Gerenciamento de Risco.
- d) Gerenciamento de Requisitos, Validação Mestre, Teste de Qualidade e Gerenciamento de Risco.
- e) Resolução de Problemas, Aceitação do Produto, Métricas, Gerenciamento de Risco e Garantia de Qualidade.

Comentários:

Os Planos de Desenvolvimento de Software são: Plano de Infraestrutura, Plano de Aceitação do Produto, Plano de Gerenciamento de Configuração, Plano de Documentação, Plano de Garantia de Qualidade, Plano de Resolução de Problemas, Plano de Gerenciamento de Subfornecedores, Plano de Melhoria de Processo, Plano de Iteração, Plano de Gerenciamento de Requisitos, Plano de Métricas, Plano de Gerenciamento de Riscos.

Rapaziada, é um absurdo cobrar esse decoreba em provas! Infelizmente, as bancas às vezes perdem a mão =/

Gabarito: Letra E

47.(ESAF - 2013 – DNIT – Analista de Sistemas) São práticas recomendadas pelo Rational Unified Process:

- a) Desenvolver software paulatinamente. Eliminar requisitos. Usar arquiteturas baseadas em componentes. Modelar software sequencialmente. Verificar a qualidade do software continuamente. Controlar as mudanças de orientação.
- b) Adquirir software aplicativo. Gerenciar os requisitos. Usar arquiteturas baseadas em especificações de preço. Modelar software analiticamente. Verificar a atualidade do software continuamente. Controlar as pendências no software.
- c) Desenvolver problemas iterativamente. Gerenciar os repositórios de requisitos. Usar enfoques baseados em componentes. Modelar software visualmente. Verificar a qualidade do software continuamente. Eliminar as mudanças no software.



- d) Desenvolver software interativamente com os patrocinadores. Desconsiderar requisitos complexos. Usar arquiteturas baseadas em software. Modelar software visualmente. Verificar a origem do software continuamente. Controlar as variáveis no software.
- e) Desenvolver software iterativamente. Gerenciar os requisitos. Usar arquiteturas baseadas em componentes. Modelar software visualmente. Verificar a qualidade do software continuamente. Controlar as mudanças no software.

Comentários:

A perspectiva prática descreve boas práticas de engenharia de software recomendadas para uso em desenvolvimento de sistemas. São elas: Desenvolvimento Iterativo; Gerenciamento de Requisitos; Uso da Arquitetura de Componentes; Modelagem Visual (UML); Contínua Verificação da Qualidade; e Gerenciamento de Mudanças. Logo, trata-se da última opção.

Gabarito: Letra E

48.(ESAF - 2012 – CGU – Analista de Sistemas) São finalidades da Disciplina de Teste do RUP, verificar:

- a) Relações indevidas entre objetos. A integração adequada de todos os componentes de hardware com o software.
- b) A implantação das atividades. A documentação de todos os componentes de software.
- c) A interação entre atores. A integração adequada de todos os provedores de software.
- d) A implantação dos objetos. A dependência de componentes de software em relação a hardware.
- e) A interação entre objetos. A integração adequada de todos os componentes do software.

Comentários:

As finalidades são: verificar **a interação entre os objetos** componentes do sistema; verificar a integração correta de todos os componentes do sistema; verificar se todos os requisitos foram implementados de maneira correta; detectar o maior número possível de defeitos antes da fase de Implantação; retestar todas as correções de defeitos e garantir que outras partes do sistema não foram afetadas por tais correções.

Gabarito: Letra E



49. (ESAF - 2012 – CGU – Analista de Sistemas) O RUP usa a abordagem da orientação a objetos em sua concepção e é projetado e documentado utilizando a notação UML (Unified Modeling Language) para ilustrar os processos em ação. O objetivo da disciplina de análise e projeto é:

- a) mostrar como o sistema vai ser realizado.
- b) mostrar como o sistema pode estabelecer requisitos.
- c) controlar a execução do desenvolvimento.
- d) estabelecer metodologias de análise decorrentes de projetos.
- e) mostrar como o sistema pode especificar o projeto.

Comentários:

Mais numa questão decoreba! O objetivo da análise e projeto é mostrar como o sistema vai ser realizado. O objetivo é construir um sistema que: execute, em um ambiente de execução específico, as tarefas e funções especificadas nas descrições de casos de uso; cumpra todas as suas necessidades; seja fácil de manter quando ocorrerem mudanças de requisitos funcionais.

Gabarito: Letra A

50. (ESAF - 2012 – CGU – Analista de Sistemas) No RUP (Rational Unified Process), casos de uso são:

- a) casos de usuários unificados em processos de racionalização.
- b) cenários de utilização do sistema por usuários.
- c) cenários de racionalização de aplicações.
- d) casos de utilização do RUP para maior racionalidade na aplicação dos recursos.
- e) cenários de utilização compartilhada de soluções por usuários de maior racionalidade.

Comentários:

Galera, o que são casos de uso? São cenários de utilização do sistema por usuários, isto é, um processo compreendido do ponto de vista do usuário. Cada cenário descreve o comportamento do sistema por meio de casos de uso.

Gabarito: Letra B

51. (ESAF - 2012 – CGU – Analista de Sistemas) O RUP (Rational Unified Process) descreve:

- a) Perfis previstos ou perfis resultantes. Artefatos. Atividades modificadas durante o processo de racionalização. Ambientes de atividades.
- b) Papéis ou perfis de relações. Atributos. Atividades primárias e secundárias. Fluxos de atividades.



- c) Papéis presentes e futuros. Artefatos. Atividades executadas durante o processo de desenvolvimento. Fluxos de inter-relações.
- d) Papéis ou perfis de trabalho. Artefatos. Atividades executadas durante o processo de desenvolvimento. Fluxos de atividades.
- e) Papéis nos trabalhos modelados. Atividades compartilhadas. Processos de desenvolvimento e de implantação. Fluxos de atividades.

Comentários:

O RUP possui cinco elementos principais: papéis ou perfis de trabalho, atividades (executadas durante o processo de desenvolvimento), artefatos, fluxos de trabalho/atividades (Workflows) e disciplinas.

Gabarito: Letra D

52. (FGV – 2017 – ALERJ – Analista de Sistemas) Um sistema está sendo desenvolvido com a utilização do processo unificado, que contém diversas fases. Na fase atual do processo será feita a implantação do sistema e a análise de lições aprendidas. Os analistas de requisitos e de negócio, praticamente, já terminaram suas atividades. É necessário ainda analisar a possibilidade de se executar outro ciclo de desenvolvimento. O sistema está na fase de:

- a) produção;
- b) concepção;
- c) elaboração;
- d) transição;
- e) construção.

Comentários:

Observem que a questão afirma que “*na fase atual do processo será feita a implantação do sistema (...)*”. Ora, só pode estar na fase de Transição.

Gabarito: Letra D

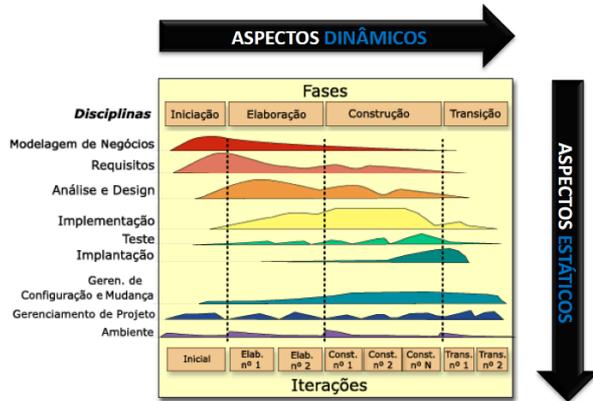
53. (IBFC /TJ-PE – 2017) O RUP possui duas dimensões, uma representando o aspecto dinâmico do processo e a outra o aspecto estático do processo. Para tanto, no eixo vertical ela é representada:

- a) pelas atividades
- b) pelas fases



- c) pelos artefatos
- d) pelos marcos
- e) pelas disciplinas

Comentários:



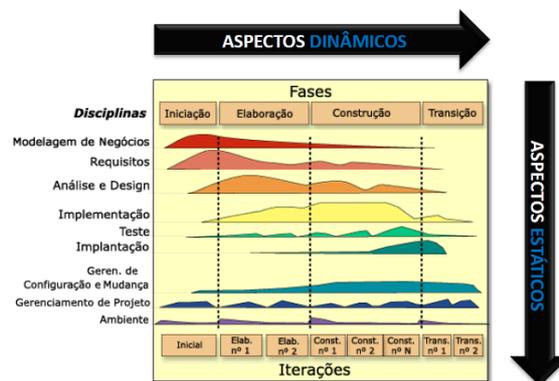
No eixo vertical, nós temos as disciplinas; no eixo horizontal, nós temos as fases.

Gabarito: Letra E

54. (IBFC /TJ-PE – 2017) Conforme o clássico “gráfico das baleias” do RUP (Rational Unified Process) especificamente a disciplina de Análise e Projeto (Design) estará em seu pico maior quando estiver na fase de:

- a) Implantação
- b) Construção
- c) Transição
- d) Elaboração
- e) Iniciação

Comentários:



A ênfase da disciplina de Análise e Projeto (Design) ocorre na fase de Elaboração (basta lembrar do gráfico das baleias).

Gabarito: Letra D

55. (IBFC / EBSERH – 2017) O RUP (Rational Unified Process) considera que existam nove disciplinas: seis de Engenharia de Software (Core Business) e três de Apoio/Suporte (Core Supporting Workflows). Assinale a alternativa que contenha somente as três disciplinas de Apoio/Suporte:

- a) Ambiente - Análise e Projeto - Gerência de Projeto
- b) Modelagem de Negócios - Configuração e Gerência de Mudança - Implementação
- c) Ambiente - Configuração e Gerência de Mudança - Implementação
- d) Modelagem de Negócios - Análise e Projeto - Gerência de Projeto
- e) Ambiente - Configuração e Gerência de Mudança - Gerência de Projeto.

Comentários:

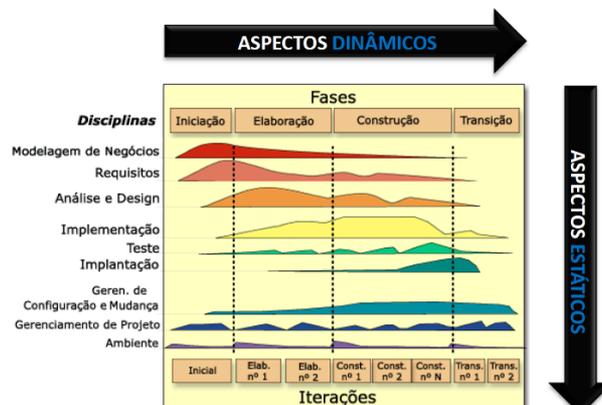
As três disciplinas de infraestrutura ou suporte são: Ambiente; Gerência de Configuração e Mudança (a banca misturou um pouco a ordem); e Gerência de Projeto.

Gabarito: Letra E

56. (IBFC / EBSERH – 2017) O RUP (Rational Unified Process) estabelece que as maiores cargas, ou fluxos de trabalho, da Implantação são realizadas durante as fases de:

- a) Iniciação e Elaboração
- b) Elaboração e Construção
- c) Iniciação e Transição
- d) Construção e Transição
- e) Iniciação e Construção.

Comentários:



A ênfase da disciplina de Implantação ocorre nas fases de Construção e Transição (basta lembrar do gráfico das baleias).

Gabarito: Letra D

57. (IBFC / EBSEH – 2017) “O RUP usa a abordagem da _____ em sua concepção, utiliza desenvolvimento _____ e é projetado e documentado utilizando a notação _____ para ilustrar os processos em ação”. Assinale a alternativa que complete correta e respectivamente as lacunas:

- a) programação procedural - orientado a aspecto - multiparadigma
- b) orientação a fluxo de dados - modular - UML (Unified Modeling Language)
- c) orientação a objetos - iterativo e incremental - UML (Unified Modeling Language)
- d) orientação a fluxo de dados - iterativo e incremental - modular
- e) orientação a objetos - modular - multiparadigma

Comentários:

O RUP usa a abordagem da orientada a objetos em sua concepção, utiliza desenvolvimento iterativo e incremental e é projetado e documentado utilizando a notação UML (Unified Modeling Language) para ilustrar os processos em ação.

Gabarito: Letra C

58. (IBFC / MGS – 2015) A linguagem padrão para modelagem orientada a objetos e que surgiu da fusão de três grandes métodos, do BOOCH, OMT (Rumbaugh) e OOSE (Jacobson) é denominada pela sigla em inglês:

- a) RUP
- b) UML
- c) CMMI
- d) SCRUM.

Comentários:

A linguagem padrão para modelagem orientada a objetos é a... UML (Unified Modeling Language). RUP e SCRUM são metodologias de desenvolvimento de software e CMMI é um modelo de boas práticas para maturidade e qualidade de software.

Gabarito: Letra B



59. (IBFC / TRE-AM – 2014) O processo unificado (Unified Process) de desenvolvimento de software é o conjunto de atividades necessárias para transformar requisitos do usuário em um sistema de software. Assinale a alternativa que NÃO faz parte das características típicas do processo unificado:

- a) Iterativo e Incremental.
- b) Dirigido por Casos de Uso.
- c) Focado no Risco.
- d) Centrado no Treinamento.

Comentários:

RUP é iterativo e incremental; é dirigido por casos de uso; é focado em risco; mas não é centrado em treinamentos.

Gabarito: Letra D

60. (IBFC / EBSE RH – 2013) Quanto as quatro fases do RUP relacione a fase da coluna da esquerda com a sua respectiva descrição na coluna da direita:

- | | |
|----------------|---------------------------------|
| (A) Iniciação | (E) ênfase na arquitetura |
| (B) Elaboração | (F) ênfase no escopo do sistema |
| (C) Construção | (G) ênfase no desenvolvimento |
| (D) Transição | (H) ênfase na implantação |

- a) AG - BE - CF - DH
- b) AF - BE - CG - DH
- c) AH - BF - CG - DH
- d) AE - BF - CH - DG.

Comentários:

(A) Iniciação tem (F) ênfase no escopo do sistema; (B) Elaboração tem (E) ênfase na arquitetura; (C) Construção tem (G) ênfase no desenvolvimento; (D) Transição tem (H) ênfase na implantação.

Gabarito: Letra B

61. (CESPE – 2013 – INPI) De acordo com a perspectiva de gerenciamento, o ciclo de vida de software do iRUP (IBM Rational Unified Process) divide-se em nove disciplinas sequenciais, sendo cada disciplina concluída por um artefato principal e consistida em um intervalo de tempo entre dois marcos principais, de modo que, ao final de cada ciclo, tem-se uma versão do produto.



Comentários:

Na verdade, o ciclo de vida de software que é dividido em quatro fases sequenciais e nove disciplinas sobrepostas. Ademais, cada fase consiste em um intervalo de tempo entre dois marcos principais e, não, disciplinas.

Gabarito: Errado

62. (CESPE – 2013 – INPI) No iRUP, o marco das fases de iniciação, elaboração, construção e transição são, respectivamente, objetivo do ciclo de vida, arquitetura do ciclo de vida, capacidade operacional inicial e release do produto.

Comentários:

Perfeito, percebam que bastava saber os marcos.

Gabarito: Correto

63. (CESPE – 2013 – ANP) Na fase de iniciação do RUP (Rational Unified Process), o projeto do sistema é elaborado com foco na arquitetura do sistema a ser implantado.

Comentários:

Na verdade, a fase de iniciação tem foco na definição do escopo e objetivos do projeto, apesar de já se construir um esboço de arquitetura do sistema – lembrem-se dos marcos.

Gabarito: Errado

64. (CESPE – 2011 – MEC) No RUP (Rational Unified Process), a qualidade de software é um quesito contemplado somente nas seguintes fases do ciclo de desenvolvimento: implementação, teste e entrega.

Comentários:

Um dos princípios básicos do processo unificado é o Foco Contínuo na Qualidade e uma das práticas recomendadas é a Contínua Verificação da Qualidade. Portanto, a qualidade é contemplada em todas as fases do ciclo de desenvolvimento e em todas as disciplinas.

Gabarito: Errado



65. (CESPE – 2011 – MEC) No RUP (Rational Unified Process), a qualidade de software é um quesito contemplado somente nas seguintes fases do ciclo de desenvolvimento: implementação, teste e entrega.

Comentários:

Um dos princípios básicos do processo unificado é o Foco Contínuo na Qualidade e uma das práticas recomendadas é a Contínua Verificação da Qualidade. Portanto, a qualidade é contemplada em todas as fases do ciclo de desenvolvimento e em todas as disciplinas.

Gabarito: Errado

66. (CESPE – 2011 – EBC) O RUP tem duas dimensões: o eixo horizontal e o eixo vertical. A primeira dimensão representa o aspecto estático do processo quando ele é aprovado e é expressa em termos de fases, iterações e marcos. A segunda dimensão representa o aspecto dinâmico do processo, como ele é descrito em termos de componentes, disciplinas, atividades, fluxos de trabalho, artefatos e papéis do processo.

Comentários:

A questão inverteu os conceitos: eixo horizontal representa aspectos dinâmicos e o eixo vertical representa aspectos estáticos. Tranquilo?! ;)

Gabarito: Errado

67. (CESPE – 2011 – ANAC) O RUP tem duas dimensões: o eixo horizontal e o eixo vertical. A primeira dimensão representa o aspecto estático do processo quando ele é aprovado e é expressa em termos de fases, iterações e marcos. A segunda dimensão representa o aspecto dinâmico do processo, como ele é descrito em termos de componentes, disciplinas, atividades, fluxos de trabalho, artefatos e papéis do processo.

Comentários:

A questão inverteu os conceitos: eixo horizontal representa aspectos dinâmicos e o eixo vertical representa aspectos estáticos. Tranquilo?! ;)

Gabarito: Errado

68. (CESPE – 2011 – ANAC) Ciclo de desenvolvimento em cascata, gerenciamento de requisitos, arquitetura e uso de componentes e gerência de configuração e mudanças estão entre as práticas que fundamentam o RUP (Rational Unified Process).

Comentários:



Ciclo de Desenvolvimento em Cascata? Não, Ciclo de Desenvolvimento Iterativo. São as Melhores Práticas do RUP: Desenvolvimento Iterativo, Gerenciamento de Requisitos, Uso da Arquitetura de Componentes, Modelagem Visual (UML), Contínua Verificação da Qualidade, e Gerenciamento de Mudança.

Gabarito: Errado

69. (CESPE – 2012 – ANAC) A disciplina Modelagem de Negócio do RUP descreve como elaborar o modelo de negócio da organização para a qual o sistema será construído. Tal modelo compreende o modelo de casos de uso de negócio e um modelo de objetos de negócio.

Comentários:

Perfeito, é exatamente isso! Dois artefatos importantes são o Modelo de Caso de Uso de Negócio e o Modelo de Objetos de Negócio.

Gabarito: Correto

70. (CESPE – 2011 – EBC) A metodologia RUP faz uso de UML (Unified Modeling Language) e procura reduzir riscos do projeto.

Comentários:

Uso de Modelagem Visual (UML) é uma das melhores práticas e o fato de ser um modelo iterativo reduz os riscos do projeto.

Gabarito: Correto

71. (CESPE – 2011 – TRE/ES) No desenvolvimento de software por meio do RUP, definem-se marcos de progresso do processo, com previsão de entrega de produtos e decisões nas passagens das fases.

Comentários:

Perfeito, há quatro marcos principais que marcam o progresso e ajudam nas decisões de passagem ou não de fases.

Gabarito: Correto

72. (CESPE – 2011 – STM) O RUP (Rational Unified Process) é um modelo de processo de desenvolvimento genérico e moderno, organizado em fases - concepção, elaboração, construção e implantação -, que separa as atividades em requisitos, análise e projeto.



Comentários:

Primeiro, as fases são: Concepção, Elaboração, Construção e Transição. Segundo, as disciplinas são: Modelagem de Negócio, Requisitos, Análise e Projeto, Implementação, Testes e Implantação – além das disciplinas de suporte.

Gabarito: Correto

73. (CESPE – 2011 – MPU) O processo unificado (PU) é um processo iterativo para a análise de projetos orientados a objetos, no qual o trabalho e as iterações são organizados em três fases principais: concepção, elaboração e construção.

Comentários:

Faltou uma fase: Transição.

Gabarito: Errado

74. (CESPE – 2010 – TCU) O processo unificado de software é centrado na arquitetura e orientado por casos de uso, o que sugere um fluxo de processo iterativo e incremental.

Comentários:

Perfeito! Lembram-se da definição de Processo Unificado? O RUP é um framework iterativo e incremental de desenvolvimento de software, centrado na arquitetura, planejado por riscos, guiado por casos de uso e orientado a objetos.

Gabarito: Correto

75. (CESPE – 2010 – TRE/BA) Uma falha comum em projetos de sistemas computacionais é não assegurar a qualidade do software. Normalmente, essa questão é discutida após o término dos projetos, ou a qualidade fica sob a responsabilidade de equipe diferente da equipe de desenvolvimento. O RUP, proposto pela IBM, é um processo que provê uma solução disciplinada sobre como assinalar tarefas e responsabilidades dentro de uma organização de desenvolvimento de software, porém, não auxilia no controle do planejamento e verificação da qualidade.

Comentários:

O RUP não possui uma disciplina Qualidade. Apesar disso, assume-se que cada membro da equipe é responsável pela qualidade durante todo o processo. O processo foca na descoberta do nível de qualidade esperado e provê testes nos processos para medir este nível.



Gabarito: Errado

76. (CESPE – 2010 – EMBASA) A primeira dimensão do RUP representa o aspecto dinâmico do processo quando ele é aprovado e é expressa em termos de fases, iterações e marcos.

Comentários:

Perfeito, o aspecto dinâmico é expresso em termos de fases, iterações e marcos.

Gabarito: Correto

77. (CESPE – 2010 – EMBASA) Sob a perspectiva de gerenciamento, o ciclo de vida de software do RUP é dividido em quatro fases sequenciais cujos finais são delimitados por marcos e avaliados para determinar se os objetivos da fase foram alcançados.

Comentários:

Perfeito, são quatro fases sequenciais e são delimitadas por marcos avaliados para determinar se os objetivos da fase foram alcançados e se é possível passar para a próxima fase.

Gabarito: Correto

78. (CESPE – 2010 – EMBASA) São atividades que se realizam no âmbito da disciplina de requisitos: identificar junto aos clientes o que o sistema deve fazer; definir escopo; e fornecer uma base para estimativas.

Comentários:

Sim, são todas atividades realizadas na disciplina de requisitos.

Gabarito: Correto

79. (CESPE – 2010 – TRE-MT) Uma das principais características do RUP é o uso da iteração, que, por meio de refinamentos sucessivos, melhora o entendimento do problema.

Comentários:

Perfeito, é exatamente isso!

Gabarito: Correto



80.(CESPE – 2010 – TRE-MT) O RUP fornece uma metodologia que utiliza um conjunto de ferramentas, modelos e entregáveis que interage diretamente com o código do sistema desenvolvido, agilizando o processo de compilação.

Comentários:

Agilizar o processo de compilação? Não existe essa relação!

Gabarito: Errado

81.(CESPE – 2010 – TRE-MT) Pelo fato de o RUP ser muito complexo, seu foco evita a redução dos riscos do projeto. Essa fase é tratada diretamente na UML.

Comentários:

O RUP é muito complexo? A questão não falou em relação a o quê! Seu foco evita a redução dos riscos? Não, a iteratividade ajuda a redução de riscos. Essa fase é tratada diretamente na UML? Que fase? Essa questão não faz o menor sentido!

Gabarito: Errado

82.(CESPE – 2010 – TRE-MT) O RUP reduz sensivelmente os requisitos de documentação de um projeto.

Comentários:

Pelo contrário, RUP preconiza documentação extensa!

Gabarito: Errado

83.(CESPE – 2010 – TRE-MT) O RUP tem dois modelos de comunicação: um para ambientes fora da equipe de desenvolvimento e outro exclusivo para a equipe de desenvolvimento.

Comentários:

Não, existe um único modelo com abstrações diferentes.

Gabarito: Errado

84.(CESPE – 2009 – SECONT-ES) O processo unificado é estruturado em duas dimensões. A dimensão horizontal representa o aspecto dinâmico do processo, onde estão representadas suas fases, às quais estão associados marcos que determinam sua finalização. Na outra



dimensão estão representadas as disciplinas, que agrupam logicamente as atividades. É possível haver disciplina que não esteja presente em todas as fases.

Comentários:

Primeiro, há autores que não consideram a perspectiva prática (somente horizontal e vertical). Ademais, é possível haver disciplina que não esteja presente em todas as fases. Basta visualizar o Gráfico de Baleias para perceber que a disciplina de Implantação, por exemplo, não está presente na fase de iniciação.

Gabarito: Correto

85. (CESPE – 2009 – INMETRO) O modelo de ciclo de vida empregado pelo RUP é mais formal que iterativo.

Comentários:

Pelo contrário, é mais iterativo que formal.

Gabarito: Errado

86. (CESPE – 2009 – MPE-RR) No Processo Unificado, a vida de um sistema é dividida em ciclos; cada ciclo, por sua vez, é dividido em fases e, entre as fases, tem-se a fase Construção, na qual as atividades visam capturar requisitos ainda não capturados na fase anterior e produzir uma arquitetura executável, a ser usada na fase Elaboração.

Comentários:

Na verdade, é na Elaboração que são capturados os requisitos ainda não capturados na fase anterior (Iniciação). Nesta fase, produz-se uma arquitetura executável a ser utilizada na Construção.

Gabarito: Errado

87. (CESPE – 2009 – TJ-ES) O modelo de processo incremental de desenvolvimento de software é iterativo, assim como o processo de prototipagem. Contudo, no processo incremental, diferentemente do que ocorre no de prototipagem, o objetivo consiste em apresentar um produto operacional a cada incremento.

Comentários:

Galera, essa é uma questão polêmica! Por que? Porque, de fato, o processo iterativo e incremental tem foco em entregas funcionais. Já o Processo de Prototipagem pode ocorrer de



duas maneiras: um protótipo evolucionário, que realiza entregas funcionais; e um protótipo descartável, que é utilizado para elicitare requisitos e não realiza entregas funcionais. Em geral, quando se fala apenas em prototipagem (sem especificar), devemos aceitar como prototipagem descartável e, não, evolucionária. Entenderam? ;)

Gabarito: Correto

88. (CESPE – 2009 – TJ-ES) Elaboração, no contexto do RUP, é uma fase que visa criar a baseline para a arquitetura do sistema a ser desenvolvido e, no contexto de engenharia de requisitos, a elaboração consiste em atividade cujo objetivo é o desenvolvimento de um modelo técnico refinado das funções, características e restrições do sistema.

Comentários:

Na Iniciação, já se cria um esboço da arquitetura do sistema. Na Elaboração, cria-se uma baseline para a arquitetura do sistema e a engenharia de requisitos ajuda a expandir e refinar os dados da Iniciação.

Gabarito: Correto

89. (CESPE – 2009 – TRE-ES) A metodologia Rational Unified Process (RUP) promove o envolvimento do cliente, bem como iterações e testes contínuos, o que torna o processo dependente de outros, apesar de reduzir os seus riscos. Já a metodologia Extreme Programming (XP) proporciona flexibilidade e agilidade, visto que, por meio dela, realiza-se a divisão de tarefas de forma específica.

Comentários:

O XP promove o envolvimento do cliente, bem como iterações e testes contínuos, o que torna o processo dependente de outros, apesar de reduzir os seus riscos. Ademais, proporciona flexibilidade e agilidade. Já o RUP realiza a divisão de tarefas de forma específica.

Gabarito: Errado

90.(CESPE – 2009 – TRE-ES) A metodologia Rational Unified Process (RUP) promove o envolvimento do cliente, bem como iterações e testes contínuos, o que torna o processo dependente de outros, apesar de reduzir os seus riscos. Já a metodologia Extreme Programming (XP) proporciona flexibilidade e agilidade, visto que, por meio dela, realiza-se a divisão de tarefas de forma específica.

Comentários:



O XP promove o envolvimento do cliente, bem como iterações e testes contínuos, o que torna o processo dependente de outros, apesar de reduzir os seus riscos. Ademais, proporciona flexibilidade e agilidade. Já o RUP realiza a divisão de tarefas de forma específica.

Gabarito: Errado

91. (CESPE – 2010 – MPU) Na fase de elaboração do RUP, são desenvolvidas as funcionalidades do sistema e implementados os requisitos identificados na fase de concepção.

Comentários:

Não! Na fase de construção do RUP, são desenvolvidas as funcionalidades do sistema e implementados os requisitos identificados na fase de elaboração.

Gabarito: Errado

92. (CESPE – 2010 – TRE-MT) O RUP divide todo o projeto de software em três fases diferentes: concepção (ou iniciação), com ênfase no escopo do sistema; elaboração, com ênfase na arquitetura; e construção, com ênfase no desenvolvimento.

Comentários:

Três fases? Não, quatro fases!

Gabarito: Errado

93. (CESPE – 2010 – TRE-MT) O RUP promove o uso de seis melhores práticas: desenvolva iterativamente; gerencie requisitos; use arquiteturas de componentes; modele visualmente (UML); verifique qualidade de software continuamente; e gerencie mudanças.

Comentários:

Perfeito, essas são as melhores práticas preconizadas pelo RUP!

Gabarito: Correto

94. (CESPE – 2010 – TRE-MT) As disciplinas de suporte (apoio) do RUP são: gerenciamento de classes; gerenciamento de produto; e ambiente.

Comentários:

Gerenciamento de Configuração e Mudança; Gerenciamento de Projetos; e Ambiente.



Gabarito: Errado

95. (CESPE – 2010 – TRE-MT) Um papel é uma definição abstrata de um conjunto de atividades executadas e dos respectivos artefatos. Exemplos de papéis no RUP são: analistas, desenvolvedores e testadores. Explicitamente, papéis de gerentes não fazem parte dos papéis possíveis no RUP.

Comentários:

Não, existem Gerentes, Arquitetos, Analistas, Desenvolvedores e Testadores.

Gabarito: Errado

96. (CESPE – 2010 – INMETRO) As disciplinas de engenharia do RUP são: modelagem de negócios; requisitos; análise e projeto; implementação; teste; qualidade; e implantação.

Comentários:

Não, Qualidade não é uma disciplina!

Gabarito: Errado

97. (CESPE – 2010 – INMETRO) As disciplinas de engenharia do RUP são: modelagem de negócios; requisitos; análise e projeto; implementação; teste; qualidade; e implantação.

Comentários:

Não, Qualidade não é uma disciplina!

Gabarito: Errado

98. (CESPE – 2010 – TRE-MG) RUP, uma metodologia adequada para grandes projetos de desenvolvimento, tem como características a facilidade de uso e a agilidade do projeto.

Comentários:

RUP não tem como características a facilidade de uso e agilidade do projeto.

Gabarito: Errado



LISTA DE EXERCÍCIOS

- (CESPE – 2019 – MPC/AC – Assistente Ministerial)** O processo unificado (RUP) reúne boas práticas de especificação e de projeto de sistemas, sendo um modelo genérico de processo organizado em fases que podem gerar um conjunto de produtos de trabalho. Considerando isso, assinale a opção que identifica a fase do RUP na qual devem estar incluídos o refinamento e a expansão dos casos de uso preliminares, dos requisitos não funcionais e da descrição da arquitetura do software.
 - concepção
 - construção
 - elaboração
 - produção
 - transição
- (CESPE – 2019 – SEMEF – Assistente Técnico)** Considerando o uso do RUP (Rational Unified Process), deve-se considerar que, segundo essa técnica,
 - a disciplina Modelagem de Negócio apresenta maior atividade na fase de Construção.
 - o número de iterações em cada uma de suas quatro fases é variável, conforme o projeto.
 - a disciplina Requisitos apresenta menor atividade na fase de Concepção (Inception).
 - a disciplina de Teste não é executada na fase de Elaboração.
 - a disciplina Configuração e Gerenciamento de Mudanças não é executada na fase de Elaboração.
- (CESPE – 2019 – SLU/DF – Analista de Gestão)** No processo unificado, requisitos é a disciplina que demanda maior esforço nas fases de elaboração e construção de software.
- (UFPR – 2018 – COREN – Analista de TI)** De acordo com o Processo Unificado (RUP), que resultado deve ser alcançado ao término da fase de elaboração?
 - A integração do sistema nas plataformas definidas pelo cliente.
 - A descrição da arquitetura do sistema.
 - A disponibilização das versões finais dos manuais do usuário.
 - A disponibilização da versão inicial do documento de visão do sistema.
 - O encerramento dos testes para validar o sistema em relação às expectativas de seus usuários.
- (FCC - 2009 - PGE-RJ - Técnico Superior de Análise de Sistemas e Métodos)** O Processo Unificado se caracteriza por ser um:



- a) ciclo de desenvolvimento de software em cascata, centrado na arquitetura e guiado pela modelagem de negócio.
 - b) ciclo de desenvolvimento de software sequencial com todos os entregáveis produzidos em uma só fase.
 - c) processo de software específico para reengenharia, centrado em objetos e orientado a casos de uso.
 - d) processo de software iterativo e incremental, centrado na arquitetura e guiado por casos de uso.
 - e) processo de software interativo, centrado na temporalidade dos negócios e orientado a eventos.
6. (FCC - 2011 - TRE-AP - Analista Judiciário - Análise de Sistemas) Considerando a representação gráfica dos fluxos de processos essenciais e fases, Implementation tem participação mais significativa em:
- a) Construction.
 - b) Elaboration.
 - c) Requirements.
 - d) Deployment.
 - e) Transition.
7. (FCC - 2008 - TRF - 5ª REGIÃO - Analista Judiciário - Tecnologia da Informação – A) O RUP possibilita o desenvolvimento incremental e interativo, guiado por casos de uso e centrado na arquitetura do sistema.
8. (FCC - 2013 - DPE-SP - Agente de Defensoria - Analista de Sistemas) O modelo estabelecido para o RUP (Rational Unified Process) é composto por quatro fases, denominadas:
- a) Requisitos, Implantação, Testes e Ambiente.
 - b) Análise, Projeto, Negócios e Comissionamento.
 - c) Concepção, Elaboração, Construção e Transição.
 - d) Partição, Integração, Testes e Operação.
 - e) Planejamento, Codificação, Integração e Configuração.
9. (FCC - 2012 - MPE-PE - Analista Ministerial - Informática) A visão estática do RUP prioriza as atividades que ocorrem durante o processo de desenvolvimento. Na descrição do RUP, essas são chamadas de workflows. Existem seis workflows centrais, identificadas no processo e três de apoio, dentre os quais é possível citar os workflows de:



- a) Meio ambiente e Gerenciamento de projeto.
- b) Concepção e Construção.
- c) Transição e Iteração.
- d) Plano de desenvolvimento e Conceito de operação.
- e) Análise de Riscos e Operação e manutenção.

10. (FCC - 2012 - TRT - 6ª Região (PE) - Analista Judiciário - Tecnologia da Informação) A perspectiva prática sobre o RUP descreve as boas práticas da engenharia de software que são recomendadas para uso no desenvolvimento de sistemas. Dentre as práticas fundamentais recomendadas incluem-se:

- a) utilizar a arquitetura em cascata e efetuar programação em pares.
- b) definir a funcionalidade do protótipo e avaliar o protótipo.
- c) definir o esboço dos requisitos e estabelecer objetivos do protótipo.
- d) utilizar arquiteturas baseadas em componentes e modelar os softwares visualmente.
- e) desenvolver teste inicial a partir de cenários e utilizar frameworks de testes automatizados.

11. (FCC - 2011 - TCE-PR - Analista de Controle - Informática) A conclusão da análise, do design, do desenvolvimento e do teste de todas as funcionalidades necessárias ao sistema, no processo RUP, é um dos objetivos da fase de:

- a) iniciação.
- b) elaboração.
- c) integração.
- d) construção.
- e) transição.

12. (FCC - 2011 - TRE-AP - Analista Judiciário - Análise de Sistemas) Considerando a representação gráfica dos fluxos de processos essenciais e fases, Implementation tem participação mais significativa em:

- a) Construction.
- b) Elaboration.
- c) Requirements.
- d) Deployment.
- e) Transition.

13. (FCC - 2011 - INFRAERO - Analista de Sistemas - Gestão de TI) Uma disciplina do RUP que tem como uma de suas finalidades "assegurar que os clientes, usuários e desenvolvedores



tenham um entendimento comum da organização-alvo”, a qual se relaciona com a disciplina Ambiente. Trata-se de:

- a) Requisitos.
- b) Análise e Design.
- c) Modelagem de Negócios.
- d) Gerenciamento de Configuração e Mudança.
- e) Gerenciamento de Projetos.

14. (FCC - 2011 - TRE-AP - Analista Judiciário - Análise de Sistemas) É um dos core “supporting” workflows, o:

- a) Test.
- b) Inception.
- c) Analysis & Design.
- d) Business modeling.
- e) Configuration and Change Management.

15. (FCC - 2011 - TRT - 1ª REGIÃO (RJ) - Analista Judiciário - Tecnologia da Informação) No Processo Unificado, a maior porção do core workflow denominado Analysis é executada na fase:

- a) Elaboration.
- b) Construction.
- c) Implementation.
- d) Inception.
- e) Transition.

16. (FCC - 2011 - TRE-RN - Analista Judiciário - Análise de Sistemas) São produtos da fase de elaboração do RUP:

- a) documento de visão e produto de software integrado.
- b) descrição da arquitetura do software e lista de riscos revisada.
- c) manual do usuário e base de dados operacionais convertidas.
- d) lista de riscos revisada e base de dados operacionais convertidas.
- e) produto de software integrado e descrição da arquitetura do software.

17. (FCC - 2011 - TRE-RN - Técnico Judiciário - Programação de Sistemas) O RUP produz artefatos:

- a) na fase de Transição, apenas.
- b) em todas as suas fases.
- c) na fase de Concepção, apenas.
- d) na fase de Elaboração, apenas.



e) na fase de Construção, apenas.

18.(FCC - 2011 - CAIXA - Analista de Sistemas) No gráfico do modelo iterativo RUP, a modelagem do negócio e a implementação têm sua maior porção representada, respectivamente, nas fases:

- a) Inception e Elaboration.
- b) Inception e Construction.
- c) Elaboration e Construction.
- d) Requirements e Construction.
- e) Elaboration e Transition.

19.(FCC - 2011 - TRE-RN - Técnico Judiciário - Programação de Sistemas) No Processo Unificado, uma descrição da arquitetura do software, um documento de visão e um modelo de projeto são aplicáveis, respectivamente, nas fases:

- a) elaboração, concepção e construção.
- b) concepção, concepção e elaboração.
- c) construção, transição e concepção.
- d) transição, construção e construção.
- e) concepção, elaboração e transição.

20.(FCC - 2010 - MPE-RN - Analista de Tecnologia da Informação - Engenharia de Software) São respectivamente disciplina (Core Process Workflow) e fase (Phase) do RUP:

- a) Concepção e Implantação.
- b) Implementação e Elaboração.
- c) Implantação e Requisitos.
- d) Requisitos e Modelagem de Negócios.
- e) Implementação e Teste.

21.(FCC - 2010 - MPE-RN - Analista de Tecnologia da Informação - Engenharia de Software) No RUP, a ênfase no escopo do sistema está na fase de:

- a) Concepção.
- b) Implementação.
- c) Elaboração.
- d) Transição.
- e) Construção.

22.(FCC - 2010 - MPE-RN - Analista de Tecnologia da Informação - Engenharia de Software) No RUP, a ênfase em arquitetura está na fase de:

- a) Transição.



- b) Modelagem de Negócio.
- c) Implantação.
- d) Implementação.
- e) Elaboração.

23. (FCC - 2010 - MPE-RN - Analista de Tecnologia da Informação - Engenharia de Software) No RUP, Project Management e Environment são:

- a) Phases.
- b) Core Process Workflows.
- c) Metrics.
- d) Core Supporting Workflows.
- e) Analysis & Design Process.

24. (FCC - 2010 - MPE-RN - Analista de Tecnologia da Informação - Engenharia de Software) No Processo Unificado (UP), o fluxo de trabalho Análise, tem forte concentração na fase de:

- a) Implementação.
- b) Elaboração.
- c) Construção.
- d) Concepção.
- e) Transição.

25. (FCC - 2010 - AL-SP - Agente Técnico Legislativo Especializado - Tecnologia da Informação) No Processo Unificado de desenvolvimento de software, Requisitos e Teste são:

- a) designações das fases do ciclo de vida.
- b) designações dos fluxos de trabalho.
- c) disciplinas consideradas apenas na quarta fase.
- d) disciplinas consideradas apenas na segunda fase.
- e) disciplinas consideradas apenas na terceira e na quarta fases.

26. (FCC - 2010 - AL-SP - Agente Técnico Legislativo Especializado - Tecnologia da Informação) São fases do Processo Unificado de desenvolvimento de software:

- a) Concepção e Transição.
- b) Elaboração e Análise.
- c) Concepção e Implementação.
- d) Projeto e Implementação.
- e) Análise e Projeto.

27. (FCC - 2005 – TRT/13 – Analista de Sistemas) São quatro fases do RUP:

- a) Transaction, Implementation, Requirements e Configuration.
- b) Project Management, Transaction, Construction e Interaction.



- c) Deployment, Transition, Design e Requirements.
- d) Inception, Construction, Elaboration e Transition.
- e) Analysis, Design, Elaboration e Inception.

28. (FCC - 2006 – BACEN – Analista de Sistemas) O RUP (Rational Unified Process):

- a) tem sua arquitetura desenvolvida em paralelo aos use cases, englobando apenas os aspectos estáticos do software.
- b) é um processo estático que viabiliza a definição seqüencial de um problema e tem como única finalidade o teste do produto final.
- c) estabelece, em sua fase de concepção, um plano de projeto e uma arquitetura sólida, por meio do desenvolvimento estático de software, dentro do seu ciclo de vida.
- d) é o modelo de processo que estabelece a topologia do hardware necessária para executar o sistema.
- e) dá ênfase à criação e à manutenção de modelos e tem sua atividade de desenvolvimento orientada por use case como parte funcional do sistema que agrega valor ao resultado do software.

29. (FCC - 2006 – BACEN – Analista de Sistemas) No RUP,

- a) os artefatos são produzidos nas fases de Elaboração e Construção e os documentos nas fases de Requisitos e Implementação.
- b) podem existir mais de trinta papéis (analista de sistemas, projetista, projetista de banco de dados, entre outros), a serem representados pelos profissionais, sendo que cada pessoa pode representar apenas um papel.
- c) os workflows são descritos sumariamente, apresentando os aspectos gerais das tarefas e os principais profissionais envolvidos.
- d) a iteração pode ocorrer uma ou mais vezes, em cada uma de suas quatro fases básicas, ou até mesmo, em parte delas.
- e) a fase da elaboração visa estabelecer o escopo e a viabilidade econômica do projeto.

30. (FCC - 2007 – TRE/SE – Analista de Sistemas) Considere as afirmativas abaixo.

- I. O RUP é um processo iterativo.
- II. Sob orientação do RUP, o desenvolvimento é centrado na arquitetura.
- III. Sob orientação do RUP, as atividades desenvolvidas são orientadas por casos de uso.



É correto o que se afirma em:

- a) I, II e III.
- b) I e III, apenas.
- c) I e II, apenas.
- d) III, apenas.
- e) I, apenas.

31. (FCC - 2007 – TRE/SE – Analista de Sistemas) No RUP, a maior quantidade da disciplina Análise e Projeto é encontrada na fase de:

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

32. (FCC - 2007 – TRE/SE – Analista de Sistemas) Em um projeto baseado no processo RUP, a ênfase à arquitetura do sistema é dada na fase de:

- a) Requisitos.
- b) Construção.
- c) Transição.
- d) Concepção.
- e) Elaboração.

33. (FCC - 2007 – TRE/SE – Analista de Sistemas) Riscos reduzidos mais cedo porque os elementos são integrados progressivamente é uma vantagem, no processo RUP, da abordagem:

- a) Iterativa.
- b) Top-down.
- c) Incremental.
- d) Interativa.
- e) Em cascata.

34. (FCC - 2007 – TRF – Analista de Sistemas) No RUP, a análise do domínio do problema, o desenvolvimento do plano do projeto, o estabelecimento de uma sólida base arquitetural e a eliminação dos elementos de mais alto risco do projeto são objetivos:

- a) da Elaboration Phase.
- b) da Construction Phase.
- c) da Inception Phase.



- d) da Transition Phase.
- e) do Deployment.

35. (FCC - 2007 – TRF – Analista de Sistemas) Dos nove core process workflow do RUP, são, respectivamente, dois core engineering e dois core supporting workflows:

- a) Implementation, Test, Project Management e Environment.
- b) Requirements, Configuration and Change Management, Project Management e Test.
- c) Configuration and Change Management, Implementation, Requirements e Test.
- d) Project Management, Requirements e Implementation.
- e) Business modeling, Requirements, Analysis & Design e Implamentation.

36. (FCC - 2007 – TRF – Analista de Sistemas) A maior parcela do fluxo dos processos fundamentais do RUP, correspondente à modelagem de negócio,

- a) está distribuída entre as fases Inception e Construction.
- b) está distribuída entre as fases Elaboration e Construction.
- c) situa-se na fase Construction.
- d) está distribuída entre as fases Inception e Elaboration.
- e) situa-se na fase Transition.

37. (FUNCAB – 2010 – PRODAM/AM - Analista de Sistemas) Assinale a alternativa que NÃO representa uma característica do Processo Unificado (UP) original, descrito no livro *The Unified Software Development Process* (1999).

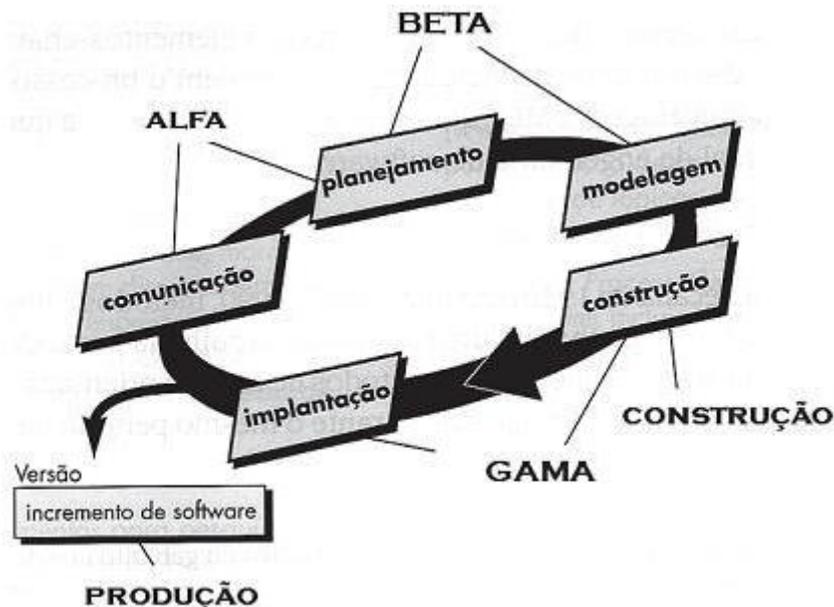
- a) Iterativo e Incremental.
- b) Centrado na arquitetura.
- c) Ágil.
- d) Orientado por casos de uso.
- e) Focado na mitigação precoce dos riscos.

38. (CESGRANRIO - 2008 - TJ-RO - Analista Judiciário - Tecnologia da Informação – A) Três dos principais artefatos da disciplina Requisitos previstos pelo RUP são: Glossário, Especificações Suplementares e Modelo de Casos de Uso.

39. (FGV - 2010 – FIOCRUZ - Analista de Sistemas) O Processo Unificado, também chamado de Rational Unified Process (RUP), representa uma tentativa de apoiar-se nos melhores recursos e características dos modelos convencionais de processo de software, mas caracterizá-los de um modo que implemente muitos dos melhores princípios de desenvolvimento ágeis. Ele enfatiza o importante papel da arquitetura de software e sugere um fluxo de processo



interativo e incremental, dando a sensação evolucionária que é essencial no desenvolvimento moderno de software. Nesse contexto, observe a figura abaixo.



As fases do processo unificado identificadas por ALFA, BETA e GAMA são denominadas, respectivamente:

- a) requisitos, elaboração e implementação.
- b) requisitos, especificação e implantação.
- c) especificação, transição e implantação.
- d) concepção, elaboração e transição.
- e) concepção, projeto e implementação.

40. (FGV - 2009 - MEC - Analista de Sistemas - Especialista) O Rational Unified Process (RUP) é um exemplo de modelo de processo moderno que foi derivado do trabalho sobre a UML e do Processo Unificado de Desenvolvimento de Software associado. O RUP reconhece que os modelos convencionais de processo apresentam uma visão única de processo e é geralmente descrito a partir de três perspectivas:

- a) teórica, prática e funcional.
- b) dinâmica, estática e prática.
- c) teórica, prática e dinâmica.
- d) funcional, teórica e estática.
- e) teórica, prática e estática.

41. (FGV - 2009 - MEC - Analista de Sistemas - Especialista) O RUP (Rational Unified Process) é uma metodologia de projeto de software criada pela Rational. O processo analítico do RUP divide o ciclo de vida de desenvolvimento de software em 4 fases. Assinale a alternativa que não indica uma fase do ciclo de vida de desenvolvimento de software, de acordo com o RUP.

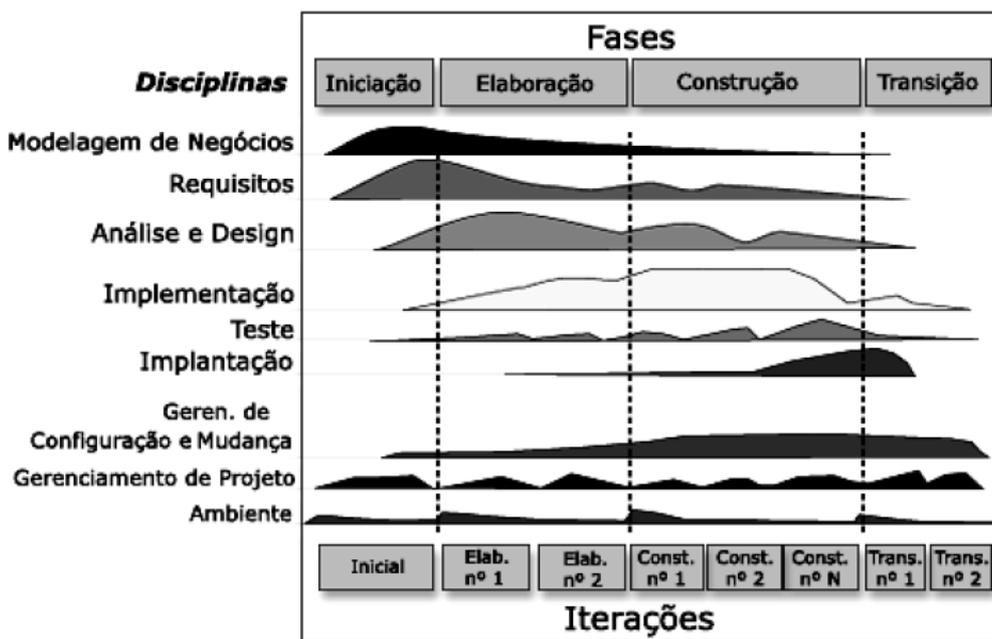


- a) Transição.
- b) Concepção.
- c) Elaboração.
- d) Implantação.
- e) Construção.

42. (FGV - 2009 - MEC - Analista de Sistemas - Especialista) O RUP (Rational Unified Process) é um exemplo de modelo de processo moderno que foi derivado do trabalho sobre a UML e do Processo Unificado de Desenvolvimento de Software associado. O RUP é um modelo constituído por fases no processo de software. Ao contrário do modelo Cascata, no qual as fases coincidem com as atividades do processo, as fases do RUP estão relacionadas mais estritamente aos negócios do que aos assuntos técnicos. Assinale a alternativa que apresente as fases do modelo RUP.

- a) Concepção, elaboração, construção e transição.
- b) Projeto, especificação, construção e transição.
- c) Especificação, construção e transição.
- d) Elaboração, construção e transição.
- e) Concepção, construção e Transição.

43. (FGV - 2009 - MEC - Analista de Sistemas) A figura abaixo fornece uma visão geral do Rational Unified Process® – RUP, um processo de engenharia de software que representa uma abordagem baseada em disciplinas para atribuir tarefas e responsabilidades dentro de uma organização de desenvolvimento. Sua meta é garantir a produção de software de alta qualidade que atenda às necessidades dos usuários dentro de um cronograma e de um orçamento previsíveis.



Nessa abordagem, para fornecer uma base para o planejamento do conteúdo técnico de interações, uma visão de arquitetura chamada VISÃO DE CASOS DE USO é utilizada na disciplina denominada:

- a) Requisitos.
- b) Implantação.
- c) Análise e Design.
- d) Modelagem de Negócios.
- e) Gerenciamento de Projeto.

44.(VUNESP - 2009 - CETESB - Analista de TI - Sistemas) Em RUP, a integração contínua, no contexto do ciclo de vida iterativo, significa:

- a) integração no fim de cada construção.
- b) integração no fim de cada iteração.
- c) integração no fim de cada transição.
- d) integração apenas no fim da elaboração e construção.
- e) integrações apenas com as releases externas.

45.(VUNESP - 2009 - CETESB - Analista de TI - Sistemas) Em RUP, a iteratividade ajuda no gerenciamento de recursos e custos porque:

- a) ajuda o gerente de projetos a alocar os requisitos por fase, com base na disponibilidade dos recursos durante o tempo de vida do projeto.
- b) ajuda o gerente de projetos a organizar recursos e custos por fases. Os artefatos do projeto evoluem conforme requerido por cada fase e aumenta-se a precisão na estimativa de custo fase a fase.
- c) permite mover os requisitos do projeto por entre as fases para garantir que os recursos possam ser mais bem gerenciados durante todo o ciclo de desenvolvimento.
- d) permite planejar as interações prévia e detalhadamente para todas as fases. Além disso, permite estabelecer custo e criar um perfil de uso de recurso para o projeto.
- e) permite que o gerente de projetos possa realizar orçamentos para os recursos de cada iteração.

46.(VUNESP - 2009 - CETESB - Analista de TI - Sistemas) Os planos de desenvolvimento de software cobertos em RUP são:

- a) Iteração, Construção e Publicação.



- b) Gerenciamento de Requisitos, Métricas de Produto e Garantia de Teste de Software.
- c) Gerenciamento de Recursos, Gerenciamento de Requisitos, Métricas e Gerenciamento de Risco.
- d) Gerenciamento de Requisitos, Validação Mestre, Teste de Qualidade e Gerenciamento de Risco.
- e) Resolução de Problemas, Aceitação do Produto, Métricas, Gerenciamento de Risco e Garantia de Qualidade.

47. (ESAF - 2013 – DNIT – Analista de Sistemas) São práticas recomendadas pelo Rational Unified Process:

- a) Desenvolver software paulatinamente. Eliminar requisitos. Usar arquiteturas baseadas em componentes. Modelar software sequencialmente. Verificar a qualidade do software continuamente. Controlar as mudanças de orientação.
- b) Adquirir software aplicativo. Gerenciar os requisitos. Usar arquiteturas baseadas em especificações de preço. Modelar software analiticamente. Verificar a atualidade do software continuamente. Controlar as pendências no software.
- c) Desenvolver problemas iterativamente. Gerenciar os repositórios de requisitos. Usar enfoques baseados em componentes. Modelar software visualmente. Verificar a qualidade do software continuamente. Eliminar as mudanças no software.
- d) Desenvolver software interativamente com os patrocinadores. Desconsiderar requisitos complexos. Usar arquiteturas baseadas em software. Modelar software visualmente. Verificar a origem do software continuamente. Controlar as variáveis no software.
- e) Desenvolver software iterativamente. Gerenciar os requisitos. Usar arquiteturas baseadas em componentes. Modelar software visualmente. Verificar a qualidade do software continuamente. Controlar as mudanças no software.

48. (ESAF - 2012 – CGU – Analista de Sistemas) São finalidades da Disciplina de Teste do RUP, verificar:

- a) Relações indevidas entre objetos. A integração adequada de todos os componentes de hardware com o software.
- b) A implantação das atividades. A documentação de todos os componentes de software.
- c) A interação entre atores. A integração adequada de todos os provedores de software.



d) A implantação dos objetos. A dependência de componentes de software em relação a hardware.

e) A interação entre objetos. A integração adequada de todos os componentes do software.

49. (ESAF - 2012 – CGU – Analista de Sistemas) O RUP usa a abordagem da orientação a objetos em sua concepção e é projetado e documentado utilizando a notação UML (Unified Modeling Language) para ilustrar os processos em ação. O objetivo da disciplina de análise e projeto é:

- a) mostrar como o sistema vai ser realizado.
- b) mostrar como o sistema pode estabelecer requisitos.
- c) controlar a execução do desenvolvimento.
- d) estabelecer metodologias de análise decorrentes de projetos.
- e) mostrar como o sistema pode especificar o projeto.

50. (ESAF - 2012 – CGU – Analista de Sistemas) No RUP (Rational Unified Process), casos de uso são:

- a) casos de usuários unificados em processos de racionalização.
- b) cenários de utilização do sistema por usuários.
- c) cenários de racionalização de aplicações.
- d) casos de utilização do RUP para maior racionalidade na aplicação dos recursos.
- e) cenários de utilização compartilhada de soluções por usuários de maior racionalidade.

51. (ESAF - 2012 – CGU – Analista de Sistemas) O RUP (Rational Unified Process) descreve:

- a) Perfis previstos ou perfis resultantes. Artefatos. Atividades modificadas durante o processo de racionalização. Ambientes de atividades.
- b) Papéis ou perfis de relações. Atributos. Atividades primárias e secundárias. Fluxos de atividades.
- c) Papéis presentes e futuros. Artefatos. Atividades executadas durante o processo de desenvolvimento. Fluxos de inter-relações.
- d) Papéis ou perfis de trabalho. Artefatos. Atividades executadas durante o processo de desenvolvimento. Fluxos de atividades.
- e) Papéis nos trabalhos modelados. Atividades compartilhadas. Processos de desenvolvimento e de implantação. Fluxos de atividades.



- 52. (FGV – 2017 – ALERJ – Analista de Sistemas)** Um sistema está sendo desenvolvido com a utilização do processo unificado, que contém diversas fases. Na fase atual do processo será feita a implantação do sistema e a análise de lições aprendidas. Os analistas de requisitos e de negócio, praticamente, já terminaram suas atividades. É necessário ainda analisar a possibilidade de se executar outro ciclo de desenvolvimento. O sistema está na fase de:
- a) produção;
 - b) concepção;
 - c) elaboração;
 - d) transição;
 - e) construção.
- 53. (IBFC /TJ-PE – 2017)** O RUP possui duas dimensões, uma representando o aspecto dinâmico do processo e a outra o aspecto estático do processo. Para tanto, no eixo vertical ela é representada:
- a) pelas atividades
 - b) pelas fases
 - c) pelos artefatos
 - d) pelos marcos
 - e) pelas disciplinas
- 54. (IBFC /TJ-PE – 2017)** Conforme o clássico “gráfico das baleias” do RUP (Rational Unified Process) especificamente a disciplina de Análise e Projeto (Design) estará em seu pico maior quando estiver na fase de:
- a) Implantação
 - b) Construção
 - c) Transição
 - d) Elaboração
 - e) Iniciação
- 55. (IBFC / EBSERH – 2017)** O RUP (Rational Unified Process) considera que existam nove disciplinas: seis de Engenharia de Software (Core Business) e três de Apoio/Suporte (Core Supporting Workflows). Assinale a alternativa que contenha somente as três disciplinas de Apoio/Suporte:
- a) Ambiente - Análise e Projeto - Gerência de Projeto
 - b) Modelagem de Negócios - Configuração e Gerência de Mudança - Implementação
 - c) Ambiente - Configuração e Gerência de Mudança - Implementação
 - d) Modelagem de Negócios - Análise e Projeto - Gerência de Projeto
 - e) Ambiente - Configuração e Gerência de Mudança - Gerência de Projeto.



56. (IBFC / EBSE RH – 2017) O RUP (Rational Unified Process) estabelece que as maiores cargas, ou fluxos de trabalho, da Implantação são realizadas durante as fases de:
- a) Iniciação e Elaboração
 - b) Elaboração e Construção
 - c) Iniciação e Transição
 - d) Construção e Transição
 - e) Iniciação e Construção.
57. (IBFC / EBSE RH – 2017) “O RUP usa a abordagem da _____ em sua concepção, utiliza desenvolvimento _____ e é projetado e documentado utilizando a notação _____ para ilustrar os processos em ação”. Assinale a alternativa que complete correta e respectivamente as lacunas:
- a) programação procedural - orientado a aspecto - multiparadigma
 - b) orientação a fluxo de dados - modular - UML (Unified Modeling Language)
 - c) orientação a objetos - iterativo e incremental - UML (Unified Modeling Language)
 - d) orientação a fluxo de dados - iterativo e incremental - modular
 - e) orientação a objetos - modular - multiparadigma
58. (IBFC / MGS – 2015) A linguagem padrão para modelagem orientada a objetos e que surgiu da fusão de três grandes métodos, do BOOCH, OMT (Rumbaugh) e OOSE (Jacobson) é denominada pela sigla em inglês:
- a) RUP
 - b) UML
 - c) CMMI
 - d) SCRUM.
59. (IBFC / TRE-AM – 2014) O processo unificado (Unified Process) de desenvolvimento de software é o conjunto de atividades necessárias para transformar requisitos do usuário em um sistema de software. Assinale a alternativa que NÃO faz parte das características típicas do processo unificado:
- a) Iterativo e Incremental.
 - b) Dirigido por Casos de Uso.
 - c) Focado no Risco.
 - d) Centrado no Treinamento.
60. (IBFC / EBSE RH – 2013) Quanto as quatro fases do RUP relacione a fase da coluna da esquerda com a sua respectiva descrição na coluna da direita:
- | | |
|----------------|---------------------------------|
| (A) Iniciação | (E) ênfase na arquitetura |
| (B) Elaboração | (F) ênfase no escopo do sistema |



(C) Construção (G) ênfase no desenvolvimento
(D) Transição (H) ênfase na implantação

- a) AG - BE - CF - DH
- b) AF - BE - CG - DH
- c) AH - BF - CG - DH
- d) AE - BF - CH - DG.

- 61. (CESPE – 2013 – INPI)** De acordo com a perspectiva de gerenciamento, o ciclo de vida de software do iRUP (IBM Rational Unified Process) divide-se em nove disciplinas sequenciais, sendo cada disciplina concluída por um artefato principal e consistida em um intervalo de tempo entre dois marcos principais, de modo que, ao final de cada ciclo, tem-se uma versão do produto.
- 62. (CESPE – 2013 – INPI)** No iRUP, o marco das fases de iniciação, elaboração, construção e transição são, respectivamente, objetivo do ciclo de vida, arquitetura do ciclo de vida, capacidade operacional inicial e release do produto.
- 63. (CESPE – 2013 – ANP)** Na fase de iniciação do RUP (Rational Unified Process), o projeto do sistema é elaborado com foco na arquitetura do sistema a ser implantado.
- 64. (CESPE – 2011 – MEC)** No RUP (Rational Unified Process), a qualidade de software é um quesito contemplado somente nas seguintes fases do ciclo de desenvolvimento: implementação, teste e entrega.
- 65. (CESPE – 2011 – MEC)** No RUP (Rational Unified Process), a qualidade de software é um quesito contemplado somente nas seguintes fases do ciclo de desenvolvimento: implementação, teste e entrega.
- 66. (CESPE – 2011 – EBC)** O RUP tem duas dimensões: o eixo horizontal e o eixo vertical. A primeira dimensão representa o aspecto estático do processo quando ele é aprovado e é expressa em termos de fases, iterações e marcos. A segunda dimensão representa o aspecto dinâmico do processo, como ele é descrito em termos de componentes, disciplinas, atividades, fluxos de trabalho, artefatos e papéis do processo.
- 67. (CESPE – 2011 – ANAC)** O RUP tem duas dimensões: o eixo horizontal e o eixo vertical. A primeira dimensão representa o aspecto estático do processo quando ele é aprovado e é expressa em termos de fases, iterações e marcos. A segunda dimensão representa o aspecto dinâmico do processo, como ele é descrito em termos de componentes, disciplinas, atividades, fluxos de trabalho, artefatos e papéis do processo.
- 68. (CESPE – 2011 – ANAC)** Ciclo de desenvolvimento em cascata, gerenciamento de requisitos, arquitetura e uso de componentes e gerência de configuração e mudanças estão entre as práticas que fundamentam o RUP (Rational Unified Process).



69. **(CESPE – 2012 – ANAC)** A disciplina Modelagem de Negócio do RUP descreve como elaborar o modelo de negócio da organização para a qual o sistema será construído. Tal modelo compreende o modelo de casos de uso de negócio e um modelo de objetos de negócio.
70. **(CESPE – 2011 – EBC)** A metodologia RUP faz uso de UML (Unified Modeling Language) e procura reduzir riscos do projeto.
71. **(CESPE – 2011 – TRE/ES)** No desenvolvimento de software por meio do RUP, definem-se marcos de progresso do processo, com previsão de entrega de produtos e decisões nas passagens das fases.
72. **(CESPE – 2011 – STM)** O RUP (Rational Unified Process) é um modelo de processo de desenvolvimento genérico e moderno, organizado em fases - concepção, elaboração, construção e implantação -, que separa as atividades em requisitos, análise e projeto.
73. **(CESPE – 2011 – MPU)** O processo unificado (PU) é um processo iterativo para a análise de projetos orientados a objetos, no qual o trabalho e as iterações são organizados em três fases principais: concepção, elaboração e construção.
74. **(CESPE – 2010 – TCU)** O processo unificado de software é centrado na arquitetura e orientado por casos de uso, o que sugere um fluxo de processo iterativo e incremental.
75. **(CESPE – 2010 – TRE/BA)** Uma falha comum em projetos de sistemas computacionais é não assegurar a qualidade do software. Normalmente, essa questão é discutida após o término dos projetos, ou a qualidade fica sob a responsabilidade de equipe diferente da equipe de desenvolvimento. O RUP, proposto pela IBM, é um processo que provê uma solução disciplinada sobre como assinalar tarefas e responsabilidades dentro de uma organização de desenvolvimento de software, porém, não auxilia no controle do planejamento e verificação da qualidade.
76. **(CESPE – 2010 – EMBASA)** A primeira dimensão do RUP representa o aspecto dinâmico do processo quando ele é aprovado e é expressa em termos de fases, iterações e marcos.
77. **(CESPE – 2010 – EMBASA)** Sob a perspectiva de gerenciamento, o ciclo de vida de software do RUP é dividido em quatro fases sequenciais cujos finais são delimitados por marcos e avaliados para determinar se os objetivos da fase foram alcançados.
78. **(CESPE – 2010 – EMBASA)** São atividades que se realizam no âmbito da disciplina de requisitos: identificar junto aos clientes o que o sistema deve fazer; definir escopo; e fornecer uma base para estimativas.
79. **(CESPE – 2010 – TRE-MT)** Uma das principais características do RUP é o uso da iteração, que, por meio de refinamentos sucessivos, melhora o entendimento do problema.



80. **(CESPE – 2010 – TRE-MT)** O RUP fornece uma metodologia que utiliza um conjunto de ferramentas, modelos e entregáveis que interage diretamente com o código do sistema desenvolvido, agilizando o processo de compilação.
81. **(CESPE – 2010 – TRE-MT)** Pelo fato de o RUP ser muito complexo, seu foco evita a redução dos riscos do projeto. Essa fase é tratada diretamente na UML.
82. **(CESPE – 2010 – TRE-MT)** O RUP reduz sensivelmente os requisitos de documentação de um projeto.
83. **(CESPE – 2010 – TRE-MT)** O RUP tem dois modelos de comunicação: um para ambientes fora da equipe de desenvolvimento e outro exclusivo para a equipe de desenvolvimento.
84. **(CESPE – 2009 – SECONT-ES)** O processo unificado é estruturado em duas dimensões. A dimensão horizontal representa o aspecto dinâmico do processo, onde estão representadas suas fases, às quais estão associados marcos que determinam sua finalização. Na outra dimensão estão representadas as disciplinas, que agrupam logicamente as atividades. É possível haver disciplina que não esteja presente em todas as fases.
85. **(CESPE – 2009 – INMETRO)** O modelo de ciclo de vida empregado pelo RUP é mais formal que iterativo.
86. **(CESPE – 2009 – MPE-RR)** No Processo Unificado, a vida de um sistema é dividida em ciclos; cada ciclo, por sua vez, é dividido em fases e, entre as fases, tem-se a fase Construção, na qual as atividades visam capturar requisitos ainda não capturados na fase anterior e produzir uma arquitetura executável, a ser usada na fase Elaboração.
87. **(CESPE – 2009 – TJ-ES)** O modelo de processo incremental de desenvolvimento de software é iterativo, assim como o processo de prototipagem. Contudo, no processo incremental, diferentemente do que ocorre no de prototipagem, o objetivo consiste em apresentar um produto operacional a cada incremento.
88. **(CESPE – 2009 – TJ-ES)** Elaboração, no contexto do RUP, é uma fase que visa criar a baseline para a arquitetura do sistema a ser desenvolvido e, no contexto de engenharia de requisitos, a elaboração consiste em atividade cujo objetivo é o desenvolvimento de um modelo técnico refinado das funções, características e restrições do sistema.
89. **(CESPE – 2009 – TRE-ES)** A metodologia Rational Unified Process (RUP) promove o envolvimento do cliente, bem como iterações e testes contínuos, o que torna o processo dependente de outros, apesar de reduzir os seus riscos. Já a metodologia Extreme Programming (XP) proporciona flexibilidade e agilidade, visto que, por meio dela, realiza-se a divisão de tarefas de forma específica.



90. **(CESPE – 2009 – TRE-ES)** A metodologia Rational Unified Process (RUP) promove o envolvimento do cliente, bem como iterações e testes contínuos, o que torna o processo dependente de outros, apesar de reduzir os seus riscos. Já a metodologia Extreme Programming (XP) proporciona flexibilidade e agilidade, visto que, por meio dela, realiza-se a divisão de tarefas de forma específica.
91. **(CESPE – 2010 – MPU)** Na fase de elaboração do RUP, são desenvolvidas as funcionalidades do sistema e implementados os requisitos identificados na fase de concepção.
92. **(CESPE – 2010 – TRE-MT)** O RUP divide todo o projeto de software em três fases diferentes: concepção (ou iniciação), com ênfase no escopo do sistema; elaboração, com ênfase na arquitetura; e construção, com ênfase no desenvolvimento.
93. **(CESPE – 2010 – TRE-MT)** O RUP promove o uso de seis melhores práticas: desenvolva iterativamente; gerencie requisitos; use arquiteturas de componentes; modele visualmente (UML); verifique qualidade de software continuamente; e gerencie mudanças.
94. **(CESPE – 2010 – TRE-MT)** As disciplinas de suporte (apoio) do RUP são: gerenciamento de classes; gerenciamento de produto; e ambiente.
95. **(CESPE – 2010 – TRE-MT)** Um papel é uma definição abstrata de um conjunto de atividades executadas e dos respectivos artefatos. Exemplos de papéis no RUP são: analistas, desenvolvedores e testadores. Explicitamente, papéis de gerentes não fazem parte dos papéis possíveis no RUP.
96. **(CESPE – 2010 – INMETRO)** As disciplinas de engenharia do RUP são: modelagem de negócios; requisitos; análise e projeto; implementação; teste; qualidade; e implantação.
97. **(CESPE – 2010 – INMETRO)** As disciplinas de engenharia do RUP são: modelagem de negócios; requisitos; análise e projeto; implementação; teste; qualidade; e implantação.
98. **(CESPE – 2010 – TRE-MG)** RUP, uma metodologia adequada para grandes projetos de desenvolvimento, tem como características a facilidade de uso e a agilidade do projeto.



GABARITO

1. LETRA C
2. LETRA B
3. ERRADO
4. LETRA B
5. LETRA D
6. LETRA A
7. CORRETO
8. LETRA C
9. LETRA A
10. LETRA D
11. LETRA D
12. LETRA A
13. LETRA C
14. LETRA E
15. LETRA A
16. LETRA B
17. LETRA B
18. LETRA B
19. LETRA A
20. LETRA B
21. LETRA A
22. LETRA E
23. LETRA D
24. LETRA B
25. LETRA B
26. LETRA A
27. LETRA D
28. LETRA E
29. LETRA D
30. LETRA A
31. LETRA E
32. LETRA E
33. LETRA A
34. LETRA A
35. LETRA A
36. LETRA D
37. LETRA C
38. CORRETO
39. LETRA D
40. LETRA B
41. LETRA A
42. LETRA D
43. LETRA D
44. LETRA B
45. LETRA B
46. LETRA E
47. LETRA E
48. LETRA E
49. LETRA A
50. LETRA B
51. LETRA D
52. LETRA D
53. LETRA E
54. LETRA D
55. LETRA E
56. LETRA D
57. LETRA C
58. LETRA B
59. LETRA D
60. LETRA B
61. ERRADO
62. CORRETO
63. ERRADO
64. ERRADO
65. ERRADO
66. ERRADO
67. ERRADO
68. ERRADO
69. CORRETO
70. CORRETO
71. CORRETO
72. CORRETO
73. ERRADO
74. CORRETO
75. ERRADO
76. CORRETO
77. CORRETO
78. CORRETO
79. CORRETO
80. ERRADO
81. ERRADO
82. ERRADO
83. ERRADO
84. CORRETO
85. ERRADO
86. ERRADO
87. CORRETO
88. CORRETO
89. ERRADO
90. ERRADO
91. ERRADO
92. ERRADO
93. CORRETO
94. ERRADO
95. ERRADO
96. ERRADO
97. ERRADO
98. ERRADO



ESSA LEI TODO MUNDO CONHECE: PIRATARIA É CRIME.

Mas é sempre bom revisar o porquê e como você pode ser prejudicado com essa prática.



1 Professor investe seu tempo para elaborar os cursos e o site os coloca à venda.



2 Pirata divulga ilicitamente (grupos de rateio), utilizando-se do anonimato, nomes falsos ou laranjas (geralmente o pirata se anuncia como formador de "grupos solidários" de rateio que não visam lucro).



3 Pirata cria alunos fake praticando falsidade ideológica, comprando cursos do site em nome de pessoas aleatórias (usando nome, CPF, endereço e telefone de terceiros sem autorização).



4 Pirata compra, muitas vezes, clonando cartões de crédito (por vezes o sistema anti-fraude não consegue identificar o golpe a tempo).



5 Pirata fere os Termos de Uso, adultera as aulas e retira a identificação dos arquivos PDF (justamente porque a atividade é ilegal e ele não quer que seus fakes sejam identificados).



6 Pirata revende as aulas protegidas por direitos autorais, praticando concorrência desleal e em flagrante desrespeito à Lei de Direitos Autorais (Lei 9.610/98).



7 Concurseiro(a) desinformado participa de rateio, achando que nada disso está acontecendo e esperando se tornar servidor público para exigir o cumprimento das leis.



8 O professor que elaborou o curso não ganha nada, o site não recebe nada, e a pessoa que praticou todos os ilícitos anteriores (pirata) fica com o lucro.



Deixando de lado esse mar de sujeira, aproveitamos para agradecer a todos que adquirem os cursos honestamente e permitem que o site continue existindo.