

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.....	9
3 – Melhores Práticas.....	11
4 – Fases do RUP	13
4.1 – Iniciação (ou Concepção)	14
4.2 – Elaboração.....	14
4.3 – Construção	14
4.4 – Transição	15
5 – Disciplinas do RUP	18
5.1 – Modelagem de Negócios	18
5.2 – Requisitos.....	19
5.3 – Análise e Projeto/Design.....	19
5.4 – Implementação.....	20
5.5 – Teste.....	21
5.6 – Implantação.....	21
5.7 – Gerenciamento de Configuração e Mudanças.....	22
5.8 – Gerenciamento de Projetos	23
5.9 – Ambiente.....	24
Questões Comentadas.....	25
Lista de Questões	38
Gabarito	44



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...

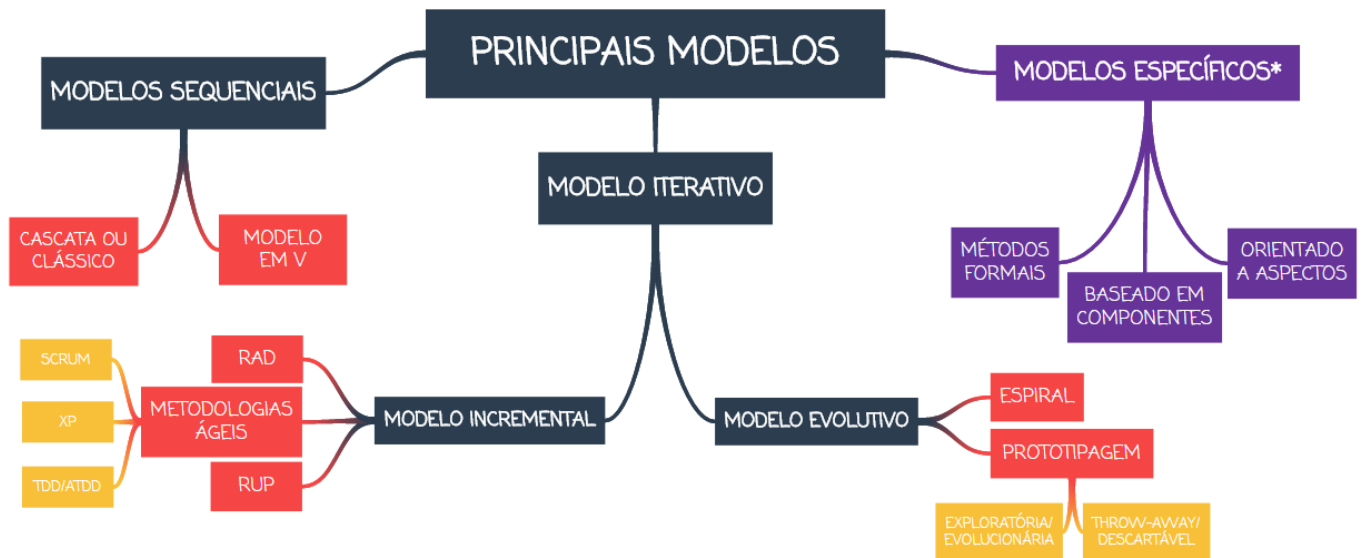


PROFESSOR DIEGO CARVALHO - [WWW.INSTAGRAM.COM/PROFESSORDIEGOCARVALHO](https://www.instagram.com/professordiego-carvalho)



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.**

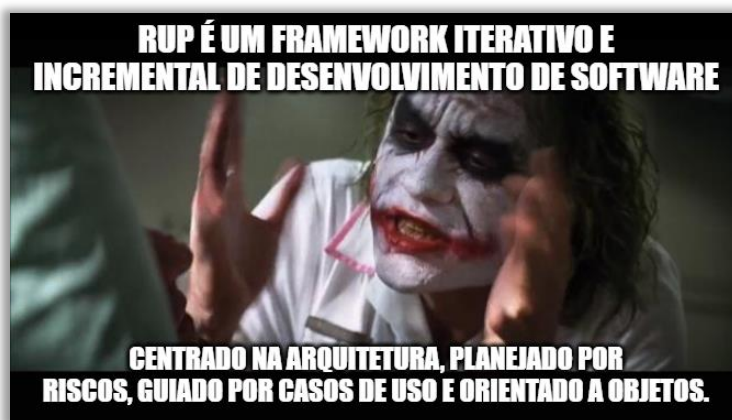
Como eles trabalhavam na Rational, ficou conhecido como Processo Unificado da Rational ou Rational Unified Process (RUP). E foi assim que nasceu o RUP! Em 2003, a IBM comprou a Rational e ela passou a manter, distribuir, evoluir, oferecer consultoria sobre RUP desde então. Alguns começaram a chamar o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

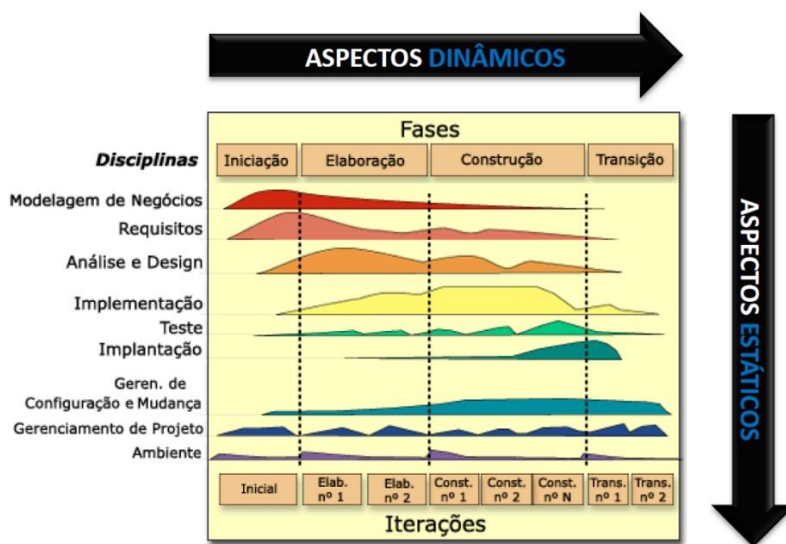
O RUP oferece uma abordagem baseada em disciplinas para atribuir tarefas e responsabilidades dentro de uma organização de desenvolvimento. **Sua meta é garantir a produção de software de alta qualidade que atenda às necessidades dos usuários dentro de um cronograma e de um orçamento previsíveis.** O 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.
2. **Estática:** também conhecido como funcional ou vertical, mostra as atividades realizadas no processo, componentes, disciplinas, artefatos e papéis de processo.



3. **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 independente 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.**

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

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.



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.**



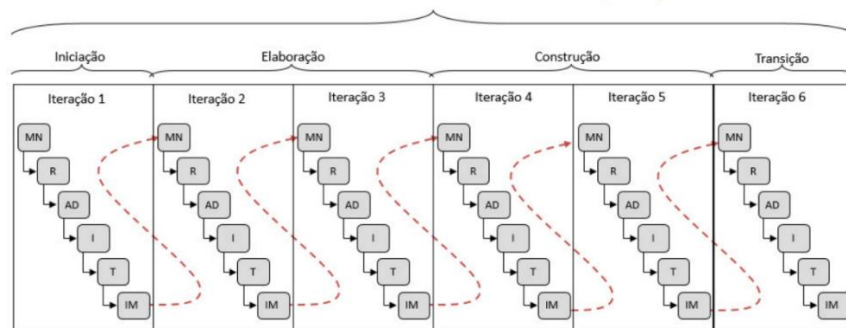
1. **Desenvolvimento Iterativo:** um projeto que usa o desenvolvimento iterativo tem um ciclo de vida que consiste em várias iterações, que incorporam um conjunto quase sequencial de atividades em modelagem de negócios, requisitos, análise e projeto, implementação, teste e implantação, dependendo do 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 conjuntos 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.



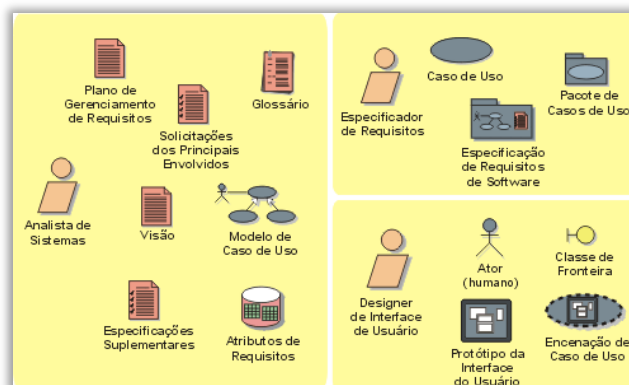


5.2 – Requisitos

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.

FINALIDADES	ESTABELECER E MANTER CONCORDÂNCIA COM OS CLIENTES E OUTROS ENVOLVIDOS SOBRE O QUE O SISTEMA DEVE FAZER.
	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.

FINALIDADES

DEFINIR A ORGANIZAÇÃO DO CÓDIGO EM TERMOS DE SUBSISTEMAS DE IMPLEMENTAÇÃO ORGANIZADOS EM CAMADAS.

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.



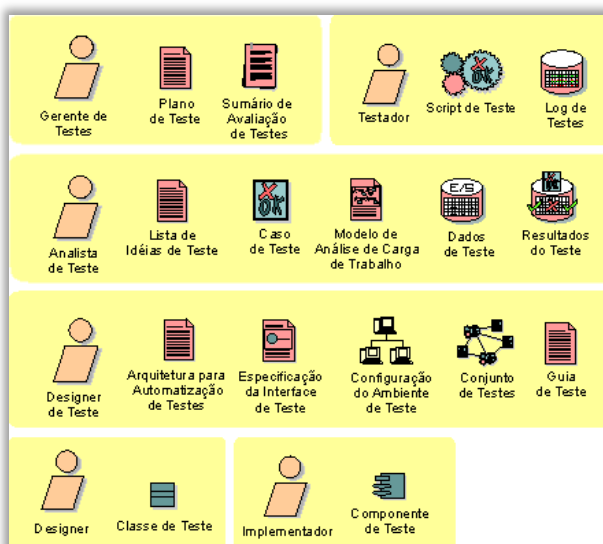


5.5 – Teste

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.

FINALIDADES	LOCALIZAR E DOCUMENTAR DEFEITOS NA QUALIDADE DO SOFTWARE.
	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.

FINALIDADES

IDENTIFICAR E CONTROLAR ITENS DE CONFIGURAÇÃO.

Restringir as mudanças nesses itens de configuração.

Auditar as mudanças nesses itens de configuração.

Evitar confusões de: atualização simultânea; notificação limitada; e várias versões.

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.



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



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.

FINALIDADES	FORNECER UM FRAMEWORK PARA GERENCIAR PROJETOS INTENSIVOS DE SOFTWARE.
	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.



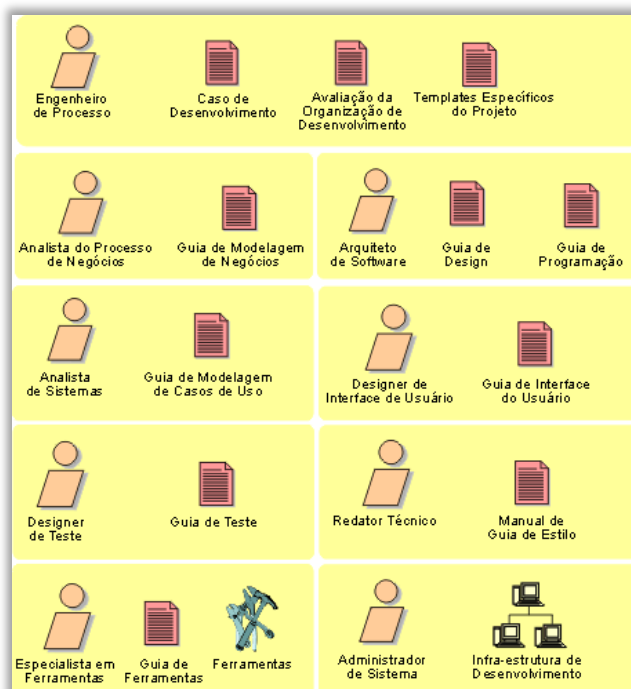
5.9 – Ambiente

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.



QUESTÕES COMENTADAS

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:

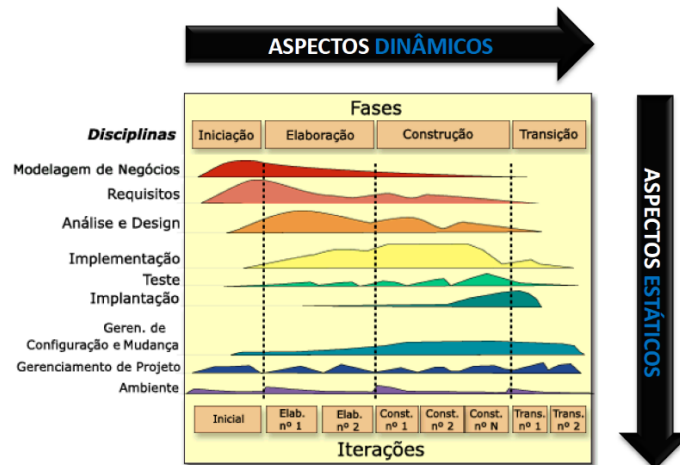
(a) Errado, fase de transição; (b) Correto, fase de elaboração; (c) Errado, fase de transição; (d) Errado, fase de concepção; (e) Errado, fase de construção.

Gabarito: Letra B

5. (FCC - 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:





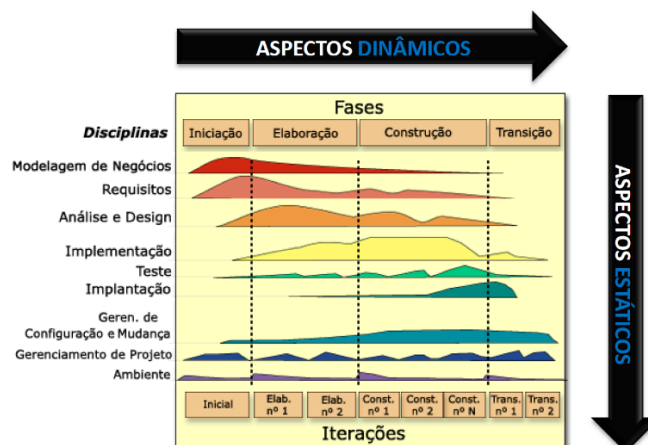
As fases são: Concepção (ou Iniciação), Elaboração, Construção e Transição.

Gabarito: Letra C

6. (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:

- Meio ambiente e Gerenciamento de projeto.
- Concepção e Construção.
- Transição e Iteração.
- Plano de desenvolvimento e Conceito de operação.
- 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.



7. (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!

8. (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 iterativamente 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

9. (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

- 10. (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

- 11. (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

- 12. (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

13. (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.

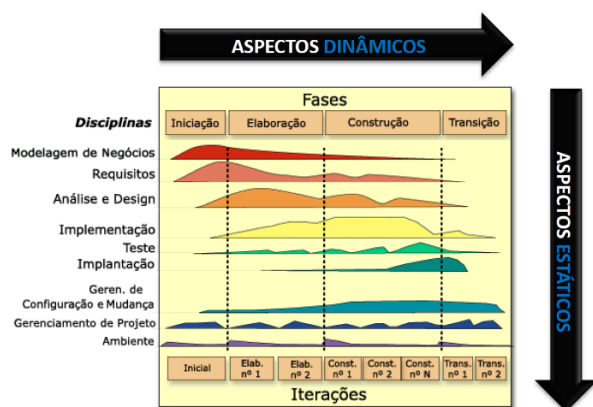
Gabarito: Letra D

14. (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.

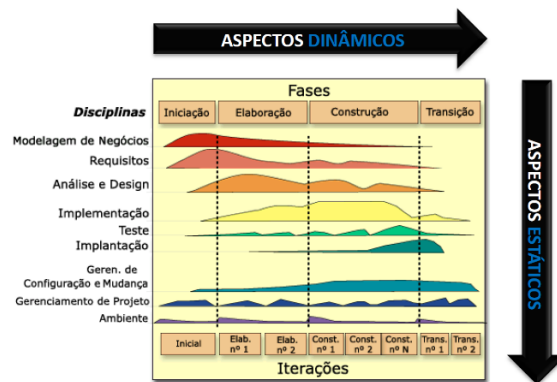
Gabarito: Letra E

15. (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

16. (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.

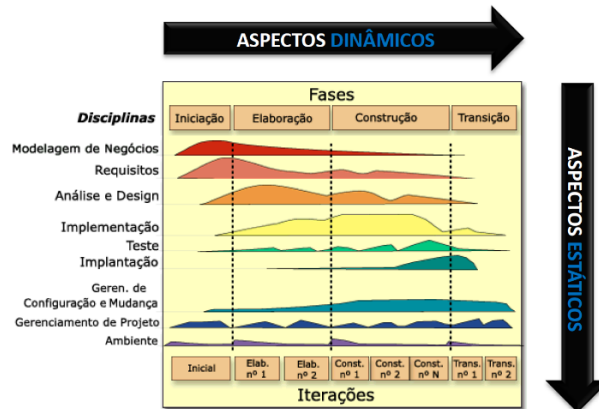
Gabarito: Letra E

17. (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

18. (IBFC / EBSERH – 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

19. (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

20. (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

21. (IBFC / EBSERH – 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

22. (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

23. (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

24. (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

25. (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



LISTA DE QUESTÕES

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
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.
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.
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.
5. **(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.

6. (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.

7. (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.

8. (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 iterativamente 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.

9. (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.

10. (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.

11. (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.

12. (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.

13. (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.

14. (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

15. (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

16. (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.

17. (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.

18. (IBFC / EBSERH – 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

19. (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.



- 20. (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.
- 21. (IBFC / EBSERH – 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.
- 22. (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.
- 23. (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.
- 24. (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.
- 25. (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.



GABARITO

1. LETRA C
2. LETRA B
3. ERRADO
4. LETRA B
5. LETRA C
6. LETRA A
7. LETRA D
8. LETRA E
9. LETRA E
10. LETRA A
11. LETRA B
12. LETRA D
13. LETRA D
14. LETRA E
15. LETRA D
16. LETRA E
17. LETRA D
18. LETRA C
19. LETRA B
20. LETRA D
21. LETRA B
22. ERRADO
23. CORRETO
24. ERRADO
25. CORRETO



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.