

Análise e Modelagem de Sistemas

Modelo do processo unificado

Você sabia que seu material didático é interativo e multimídia? Isso significa que você pode interagir com o conteúdo de diversas formas, a qualquer hora e lugar. Na versão impressa, porém, alguns conteúdos interativos ficam desabilitados. Por essa razão, fique atento: sempre que possível, opte pela versão digital. Bons estudos!

Nesta webaula vamos conhecer as características do Processo Unificado (PU), suas fases e atividades.

Processo Unificado (PU)

O Processo Unificado (PU) surgiu para melhorar o desenvolvimento de softwares com foco na A/POO (Análise e Projeto Orientados a Objetos). Este modelo de desenvolvimento de software é iterativo e adaptativo, permitindo produzir um sistema de grande porte como se fossem vários pequenos sistemas, o que diminui o risco do projeto.

Foco

Segundo Jacobson, Booch e Rumbaugh (2000), o PU é definido por três aspectos chaves:

I. Dirigido por caso de uso

O caso de uso é o fio condutor do PU, como afirmam Jacobson, Booch e Rumbaugh (2000). Nesta fase devemos conhecer **o quê** os futuros usuários necessitam e desejam. Devemos nos atentar para que os casos de usos respondam o que *cada usuário necessita* e não apenas as funções que o sistema precisa ter. Desta maneira o modelo de caso de uso guiará o desenvolvimento no seu projeto (**design**), na implementação e nos testes, avançando através de uma série de fluxo de trabalho (**workflow**).

II. Centrado na arquitetura

Visa dar ao engenheiro de software uma visão abrangente do sistema, como no caso do projeto de um carro, no qual há o design, a mecânica, o projeto elétrico, aerodinâmico etc. A arquitetura do software também deverá prover ao desenvolvedor estas visões generalizadas, tais como as necessidades dos usuários e objetivos estratégicos da empresa.

III. Iterativo e incremental

A iteração consiste em dividir o projeto em subprojetos menores e o resultado de uma iteração produz um incremento. O processo iterativo no PU é controlado, isso reduz os riscos de aumento de custos com incrementos específicos, os prazos são melhor controlados e também acelera o ritmo de desenvolvimento, pois reconhece um aspecto ignorado: dificilmente em um sistema complexo, no início, é possível definir totalmente os requisitos e necessidades do usuário.

Ciclo de vida

O ciclo de vida do PU é uma série de repetições ao longo da vida do sistema, sendo que cada ciclo completo resulta em uma versão do software, por sua vez cada ciclo é composto por 4 fases:

Concepção

Definirá a visão geral do projeto, o escopo e os requisitos

Elaboração

É uma visão mais refinada dos requisitos e da

Construção

É o momento de desenvolvimento do sistema, começando

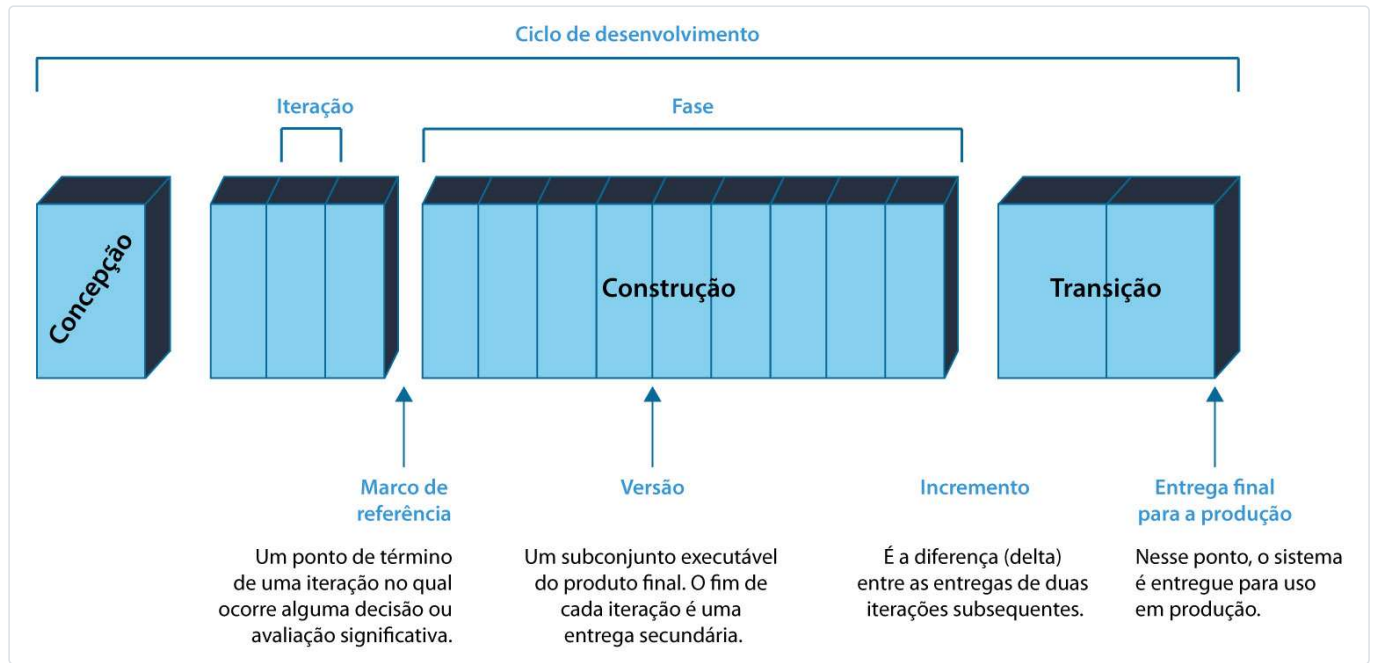
Transição

É a fase de implantação do sistema, ou seja, a

iniciais.	arquitetura, análise de riscos e estimativas.	pelos elementos mais fáceis, e inicia-se a preparação para a implantação.	entrega.
-----------	---	---	----------

Na figura a seguir o ciclo de desenvolvimento (concepção + elaboração + construção + transição) termina com a entrega de uma versão do sistema, pronta para ser implementada em produção. Todo esse ciclo é composto de muitas iterações, segundo LARMAN (2007).

Ciclo de vida do PU



Fonte: Larman (2007, p. 62)

Elementos de um Processo

O PU descreve um processo por quatro elementos básicos: papel (**QUEM**), artefato (**O QUÊ**), atividade (**COMO**) e disciplina (**QUANDO**):

<p>Papel</p> <p>(<i>worker</i>, trabalhador)</p> <p>É a identificação das responsabilidades de cada indivíduo (quem faz o quê), o papel do trabalhador naquele momento, o ator da cena.</p> <p>+</p>	<p>Artefato</p> <p>É o termo utilizado para identificar qualquer produto de trabalho, seja código, esquema de banco de dados, diagramas, modelos etc. É o produto que o <i>worker</i> gera.</p>	<p>Atividade</p> <p>É a tarefa executada pelo <i>worker</i> com o objetivo de produzir ou alterar um artefato.</p>	<p>Disciplina</p> <p>(<i>workflow</i>, fluxo de trabalho)</p> <p>É a sequência de atividades que gera um resultado significativo.</p> <p>+</p>
---	--	---	---

Workflow (Disciplinas)

Segundo Jacobson, Booch e Rumbaugh (2000), as **disciplinas fundamentais**, também chamadas fluxo de trabalho do PU são:

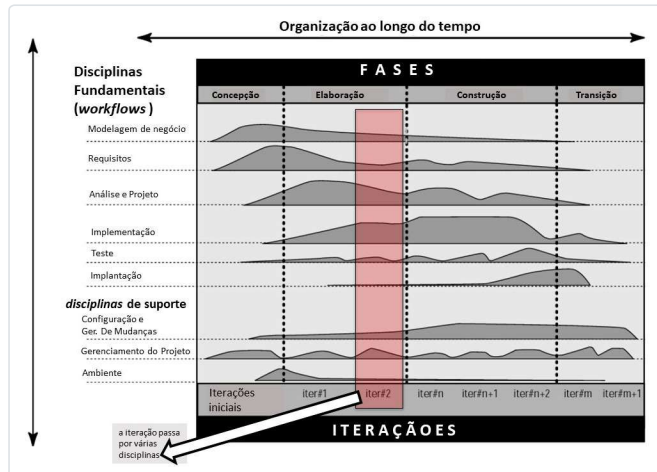
Modelagem de negócio	▼
O objetivo é documentar quais os objetivos de negócio estão envolvidos no processo e são analisados, buscando entender como o negócio deve apoiar os processos associados. Muitos projetos podem optar por não fazer modelagem de negócios. Os casos de uso de negócios são utilizados nessa fase.	
Requisitos	▼
Tem por objetivo descrever o que o sistema deve fazer. Tem como base os casos de uso para identificar os requisitos. Nessa fase surge um importante elemento na engenharia de software, os atores. (os atores envolvidos devem ser identificados, todos os quais representam os usuários e qualquer outro sistema que venha a interagir com o sistema em desenvolvimento. O foco está nas funcionalidades do sistema e tanto desenvolvedores como o cliente deverão concordar com a descrição dos requisitos.	
Análise e Projeto	▼
Estas disciplinas geram os modelos do sistema objetivando as implementações futuras. Em conjunto, essas fases melhoram a compreensão dos requisitos funcionais, definido nos casos de uso. Segundo Larman, "a análise enfatiza a investigação do problema e dos requisitos, em vez de uma solução" (LARMAN, 2005, p. 34), ou seja, como será usado e quais serão as suas funções, deixando para a implementação o modo como fazer isso. O projeto dá ênfase à solução conceitual, ou seja, cria os modelos para satisfazer os requisitos. O projeto é para o desenvolvedor e não para o usuário. Consiste em subsistemas bem definidos, com classes estruturadas em pacotes e deverá representar os componentes a serem implementados.	
Implementação	▼
O seu foco está direcionado à construção do software, o desenvolvimento do código, a famosa "mão na massa". Também prepara os primeiros testes, chamados beta teste.	
Testes	▼
Nesta etapa deve ser descrito quais os procedimentos a serem avaliados. Segundo as recomendações do documento <i>RUP: boas práticas para o desenvolvimento de software</i> (IBM,2005), os objetivos do teste são: verificar a interação entre objetos e componentes, verificar se todos os requisitos foram implementados corretamente e identificar e garantir que os defeitos sejam resolvidos antes da implantação do software.	
Implantação	▼
Nesta etapa deve ser descrito quais os procedimentos a serem avaliados. O objetivo do fluxo de trabalho da implantação é entregar o produto com êxito para os usuários, e envolve a aceitação formal do software pelo cliente. Entre as atividades do fluxo de trabalho da implantação temos a distribuição, instalação, migração dos dados e de software já existente.	

As demais disciplinas estão associadas ao **suporte** e foram incluídas no Processo Unificado Racional (RUP) para suprir disciplinas não contempladas pelo PU, são elas: **gerenciamento de mudanças e configurações**, **gerenciamento de projeto** e **ambiente**. Basicamente visam a maturidade do projeto.

Na figura a seguir você pode observar como as disciplinas são desenvolvidas ao longo das fases do ciclo de vida do desenvolvimento e nas diversas iterações (minissistemas). Cada ponto de iteração é um marco no PU e finaliza com a entrega de um incremento. Neste ponto é avaliado se os objetivos foram alcançados e, se necessário, ajustes são

Fluxo das iterações ao longo do tempo e das disciplinas

realizados. Embora para cada iteração todas as disciplinas podem estar presentes, o tempo dedicado a cada uma, muda de iteração para iteração.



Fonte: IBM (2005, p. 3, tradução nossa).

O processo unificado foi um marco na engenharia de software, e voltado para o paradigma *orientado a objeto*, sendo uma característica importante o fato de ser iterativo e incremental, assim o usuário não mais aguarda a conclusão total do software para iniciar o uso do sistema e, ao término do desenvolvimento, praticamente não há erros.

Pesquise mais!

O principal objeto do PU ou do RUP é o desenvolvimento de um produto de qualidade.

Vale a pena você conhecer o relator da Standish Group sobre alguns fatores críticos de sucesso para projetos de TI. Seguem duas referências para você consultar:

D'ÁVILA, M. Sucessos e falhas em projeto de TI. **Blog do Márcio**, 2010.

Ou diretamente na fonte em inglês (use o tradutor do browser) para acessar as pesquisas atuais do Standish Group (2020).

STANDISH GROUP. Página inicial. 2020.

A indústria de software cresceu e está em todos os negócios e as empresas de softwares precisaram obter alguma certificação para que seus produtos fossem aceitos pelos clientes ou para exportação de seus softwares. O CMMI (Capability Maturity Model Integration) é uma destas certificações e está associado diretamente a melhorar o setor de desenvolvimento. Seus resultados foram tão bons que se expandiu para outros negócios, não somente de software. Para saber mais sobre CMMI, pesquise por CMMI Institute (2020).

CMMI INSTITUTE. Página inicial. 2020.

A certificação CMMI era apenas empresarial, mas há pouco tempo o CMMI Institute criou uma certificação pessoal, que passou a ser necessária aos membros das empresas que desejam ter os níveis mais altos do padrão CMMI, logo, é bom você conhecer esse tema.

Como essa certificação é muito cara, a Softex (Associação promoção da Excelência do Software Brasileiro), um órgão do MCTIC (Ministério da Ciência, Tecnologia, Inovação e Comunicações) do governo brasileiro, desenvolveu o MPS-BR, uma certificação para a melhoria do processo de software brasileiro. É uma certificação reconhecida internacionalmente com um custo viável para as empresas nacionais. Pesquise esse programa de melhoria para saber mais.

SOFTEX. **MpsBr**. Melhoria do Processo de Software Brasileiro. 2020.