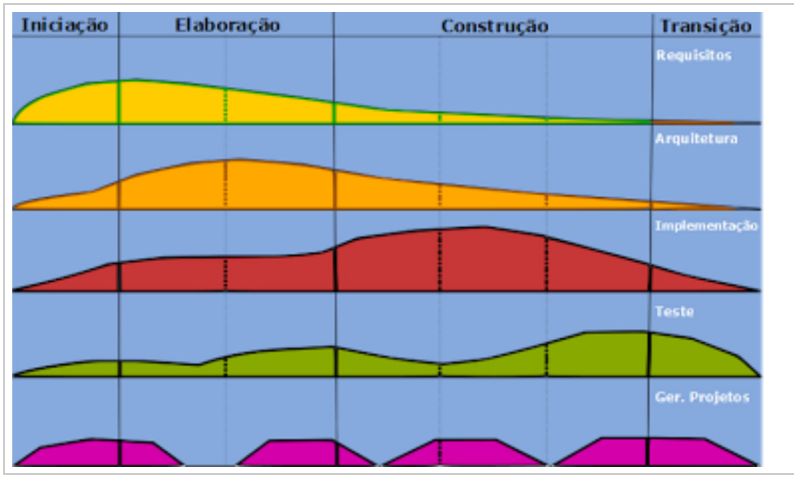
OPEN UP - PROCESSO UNIFICADO ABERTO

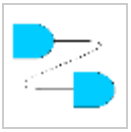
ESTE BLOG É DEDICADO A APRESENTAR OS PRINCIPAIS CONCEITOS DO PROCESSO UNIFICADO ABERTO - OPEN UP, UMA DAS METODOLOGIAS ÁGEIS QUE FAZEM PARTE DA ECLIPSE PROCESS FRAMEWORK, DA COMUNIDADE ECLIPSE. ESPERO QUE POSSA TE AJUDAR.

4. As Disciplinas do Open UP

No Processo Unificado, as Tarefas são agrupadas logicamente nas diversas disciplinas, que são distribuídas entre as fases e são executadas a cada iteração, em maior ou menor escala.
As disciplinas recebem maior ou menor ênfase de acordo com a fase na qual a iteração corrente está inserida.

Observe na figura abaixo a distribuição das disciplinas de acordo com as fases no Open UP.





Disciplina de Requisitos

A disciplina de Requisitos agrupa todas as tarefas relativas ao processo de Análise de Negócios e de Análise e Especificação de Requisitos do Open UP, tendo maior ênfase na fase de Iniciação do Processo.

Os principais propósitos desta disciplina são:

- Compreender o problema a ser resolvido;
- Compreender as necessidades dos Stakeholders (o que os envolvidos precisam - ou esperam - realmente do sistema);
- Definir os requisitos para a solução proposta (atendendo às necessidades dos envolvidos);
- Definir o escopo[1] do sistema (suas fronteiras);
- Identificar as interfaces do sistema;
- Identificar as restrições técnicas do sistema (incluindo requisitos não funcionais);
- Prover a base para o planejamento das iterações;
- Prover a base inicial para a definição do cronograma e dos custos.

Para atingir essas metas, é importante compreender a definição de escopo do problema que a solução proposta visa atender. Além disso, identificar corretamente todos os Stakeholders e suas reais necessidades é uma tarefa vital para o bom andamento do processo de desenvolvimento. A última página desse blog é dedicada exclusivamente ao processo de Análise e Levantamento de Requisitos utilizando Casos de Uso. Só lembrando, é importante destacar que hoje, um bom analista de negócios e requisitos não deve compreender e levantar apenas os requisitos que os Stakeholders querem. É preciso identificar corretamente, o que o negócio analisado demanda e o que os Stakeholders "precisam".

Tendo sido obtida a concordância dos Stakeholders e da equipe de desenvolvimento sobre quais são as reais necessidades do negócio, então os Requisitos para o sistema podem ser elicitados, organizados, analisados, validados e especificados.

Vale a pena destacar também que temos cinco classificações básicas para os requisitos que são:

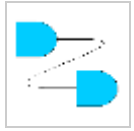
- [**F**]uncionality (Funcionalidade)
- [**U**]sability (Usabilidade)
- [**R**]eliability (Confiança)
- [**P**]erformance (Performance)
- [**S**]uportability (Suportabilidade)

Os requisitos funcionais [**F**] devem ser especificados no Modelo de Caso de Uso e os não funcionais [**URPS**] devem ser especificados no documento de Requisitos Suplementares. Não podemos esquecer, enquanto analistas ou desenvolvedores, que requisitos são dinâmicos; eles mudam, eles evoluem, eles são excluídos, novos requisitos surgem. A mudança dos requisitos deve ser adequadaente gerenciada pela equipe de desenvolvimento, evitando que requisitos sejam erroneamente implementados, que deixem de ser implementados ou que o escopo do sistema seja ultrapassado.

A disciplina de Requisitos tem uma ligação muito forte com as outras disciplinas. A arquitetura do sistema será proposta baseada nos requisitos definidos. A implementação será elaborada com base na arquitetura e nos requisitos especificados. Os testes serão realizados e analisados comparando os resultados obtidos com os requisitos especificados.

O Gerente de Projeto elabora todo o planejamento de desenvolvimento baseando-se nos requisitos. Em conjunto com os analistas de requisitos, o Gerente de Projeto deve planejar as iterações iniciais, definir os riscos e priorizar os casos de uso para iniciar o processo de desenvolvimento.

A disciplina de Requisitos é guiada por um conjunto de Tarefas que podem ser estudadas com mais detalhes **aqui** (em construção).



Disciplina de Arquitetura

A disciplina de Arquitetura tem como principal objetivo apresentar uma arquitetura estável para o desenvolvimento do sistema, baseando-se nos requisitos especificados. A disciplina de Arquitetura tem maior ênfase na fase de Elaboração.

Os principais objetivos da disciplina de Arquitetura são:

- Transformar os requisitos no *design* do futuro sistema;
- Evoluir uma arquitetura robusta do sistema;
- Adaptar o design para adequá-lo ao ambiente de implementação;
- O principal Produto de Trabalho gerado é a Arquitetura.

A disciplina de Arquitetura gera um modelo de objetos que descreve a realização de casos de uso, servindo como abstração do modelo de implementação e seu código-fonte e sendo utilizado como insumo

essencial nas atividades de implementação e teste.

A disciplina de Arquitetura acaba por transformar Requisitos em Classes, Componentes e Subsistemas, além de elaborar o modelo de dados do sistema a ser implementado. Além disso, a disciplina de Arquitetura oferece uma base muito importante para o conceito de Reuso, considerando que a mesma apresenta como resultado final, o projeto dos Componentes que podem ser devidamente reutilizados na construção de outros softwares.

A disciplina de arquitetura oferece a base necessária para que o software seja devidamente implementado. A disciplina de Testes deve verificar e validar a arquitetura.

O *design* do sistema, gerado pela disciplina de Arquitetura, tem dois objetivos principais:

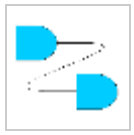
- Definir a estrutura do software;
- Definir o comportamento do software.

Isso pode ser feito utilizando modelagem visual. A forma mais indicada para tal é através a **UML** [UML2010] e seus diagramas. Para que o processo não se torne burocrático, é importante definir uma gama de diagramas a se utilizar. Por exemplo. O [RUP2007] indica o uso de um diagrama de sequência para cada cenário de caso de uso existente. Para o Open UP isso já não é indicado, considerando a quantidade de diagramas que podem surgir, dependendo do número de casos de uso. Uma indicação minha para produtos de trabalho de *design* são:

- Digrama de Atividades;
- Diagrama de Classe (MVC);
- Diagrama de Sequência (Fluxo Básico);
- Diagrama de Comunicação;
- Diagrama de Componentes;
- Digrama de Implantação.

Essa é uma indicação minha. O Open UP não trata isso de forma mais detalhada. Os diagramas podem ser alterados de acordo com a necessidade da equipe de desenvolvimento.

A disciplina de Requisitos é guiada por um conjunto de Tarefas que podem ser estudadas com mais detalhes **aqui** (em construção).



Disciplina de Implementação

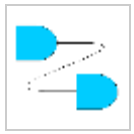
A Disciplina de Implementação organiza as tarefas que irão transformar a arquitetura proposta na implementação do sistema, buscando atender os requisitos definidos pelos Stakeholders. Os principais objetivos da disciplina de Implementação são:

- Transformar os requisitos em classes e objetos em termos de componentes;
- Definir a organização de componentes em termos da implementação de subsistemas;
- Testar os componentes desenvolvidos como unidades;
- Criar um sistema executável.

A cada Micro-Incremento e a cada Iteração que evolui, seu respectivo executável (*build*) se torna mais completo e estável. Um *build* é uma versão operacional de um sistema ou parte de um sistema e que demonstra um subconjunto de capacidades fornecidas ao produto final. Um *build* é uma parte integrante do ciclo de vida iterativo, considerando que a proposta de cada iteração é entregar como produto final, um executável estável. Outro ponto importante da entrega dos *builds* é que o mesmo fornece pontos de revisão para a equipe de desenvolvimento, estabelecendo o *feedback* constante dos *Stakeholders*. A entrega dos *builds* também permite a execução constante dos testes de integração, permitindo que o *Release*[2] obtenha estabilidade e robustez durante toda o Ciclo de Vida de Projeto.

A disciplina de Implementação é guiada por um conjunto de Tarefas

que podem ser estudadas com mais detalhes **aqui** (em construção).



Disciplina de Teste

Esta disciplina agrupa as tarefas relacionadas a teste, que se preocupam em prover feedback sobre a maturidade do sistema, projetando, implementando, executando e avaliando testes. Os principais objetivos da disciplina de Teste são:

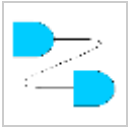
- Demonstrar se a solução arquitetada ou implementada satisfaz aos requisitos especificados;
- Medir a evolução da solução através dos Micro Incrementos;
- Melhorar o processo de desenvolvimento, descobrindo pontos controversos nos documentos de requisitos, arquitetura e implementação o mais cedo possível.
- A disciplina de Teste aplica a estratégia "testar prematuramente e testar sempre" para eliminar os riscos o mais cedo possível no Ciclo de Vida de Projeto.

Os testes ocorrem a cada iteração no Ciclo de Vida de Projeto, iniciando com os primeiros artefatos arquiteturais da solução proposta. Dependendo da fase na qual se encontra a iteração, a mesma pode ter diversos Micro Incrementos voltados para teste, dependendo da frequência com que os *builds* são liberados.
As tarefas de Teste buscam responder à seguinte questão: "O que a solução deve oferecer para considerarmos um requisito implementado e estável?".
Esta disciplina desafia as suposições, riscos e incertezas inerentes ao processo de desenvolvimento de artefatos que demandam alto conhecimento técnico.
A disciplina de Teste interagem com as demais disciplinas das seguintes formas:

- Verifica como a solução se comporta em relação aos requisitos implementados;

- Avalia os *builds* gerados na disciplina de Implementação;
- Oferece um *feedback* objetivo a cada iteração;
- Habilita os desenvolvedores a focar na implementação de novas funcionalidades e aos gerentes de projeto a elaborar o planejamento de novas iterações e/ou Micro Incrementos;
- Permite a validação dos planos de iteração desenvolvidos pelos gerentes de projeto.

A disciplina de Testes é guiada por um conjunto de Tarefas que podem ser estudadas com mais detalhes **aqui** (em construção).



Disciplina de Gerência de Projetos

Esta disciplina tem como objetivo apresentar técnicas para que o gerente de projetos possa liderar, facilitar e oferecer suporte à sua equipe, auxiliando-a a lidar com os riscos e obstáculos encontrados durante o processo de desenvolvimento de software.

O Gerenciamento de Projeto de Software é a arte de confrontar os objetivos dos *Stakeholders*, gerenciar riscos e superar obstáculos, para assim, liberar com êxito um produto que atenda às necessidades dos clientes (que pagaram por ele) e dos seus usuários. O fato de que tão poucos projetos sejam indiscutivelmente bem-sucedidos é prova suficiente da dificuldade desta tarefa.

Existem duas grandes prioridades para o gerente de projetos:

- A arquitetura da solução proposta;
- A gerência dos riscos.

Podemos definir Risco como sendo qualquer obstáculo que possa estar no caminho do sucesso do desenvolvimento de software, ou no caminho para atingir os objetivos principais do processo de

desenvolvimento de software. Por Sucesso podemos compreender como sendo o ato de satisfazer o conjunto total dos requisitos e restrições e satisfazer as expectativas dos *Stakeholders*.

Os principais objetivos da disciplina de Gerência de Projeto são:

- Encorajar os *Stakeholders* a atingir o consenso na especificação das prioridades das suas necessidades;
- Estimular a colaboração entre os diversos membros da equipe de desenvolvimento da solução proposta;
- Focar a equipe de desenvolvimento da solução na contínua entrega de *Builds* testados e estáveis para a avaliação dos *Stakeholders*;
- Auxiliar a criar um ambiente de trabalho amigável e com o senso de colaboração entre seus membros visando maximizar a produtividade da equipe;
- Manter os *Stakeholders* e a equipe de desenvolvimento devidamente informados do progresso da solução em desenvolvimento;
- Manter uma estrutura tal que permita o gerenciamento dos riscos e a capacidade de adaptação às mudanças.

A Gerência de Projetos é uma disciplina que envolve todas as demais disciplinas, impactando e sofrendo os impactos de todas elas. A disciplina de Gerência de Projetos adiciona valor às atividades de desenvolvimento da solução proposta quando:

- Os Stakeholders confiam nas habilidades da equipe para atingir o sucesso no processo de desenvolvimento da solução;
- Os membros da equipe de desenvolvimento compreendem as intenções do Gerente de Projeto e confirmam essa compreensão gerando continuadamente os builds para avalição e integração,

gerando o produto desejado.

A disciplina de Gerência de Projetos é guiada por um conjunto de Tarefas que podem ser estudadas com mais detalhes **aqui** (em construção).

---

[1] **Escopo do Sistema** - Por Escopo do Sistema podemos entender o conjunto de funcionalidades especificadas nos requisitos definidos para atender às necessidades dos Stakeholders, nesse caso, mais necessariamente os requisitos Funcionais.
Uma boa forma de se observar o escopo do sistema é observar o diagrama de caso de uso do mesmo. O conjunto de todos os casos de uso compõem o Conjunto de Funcionalidades do Sistema, respectivamente, seu Escopo.

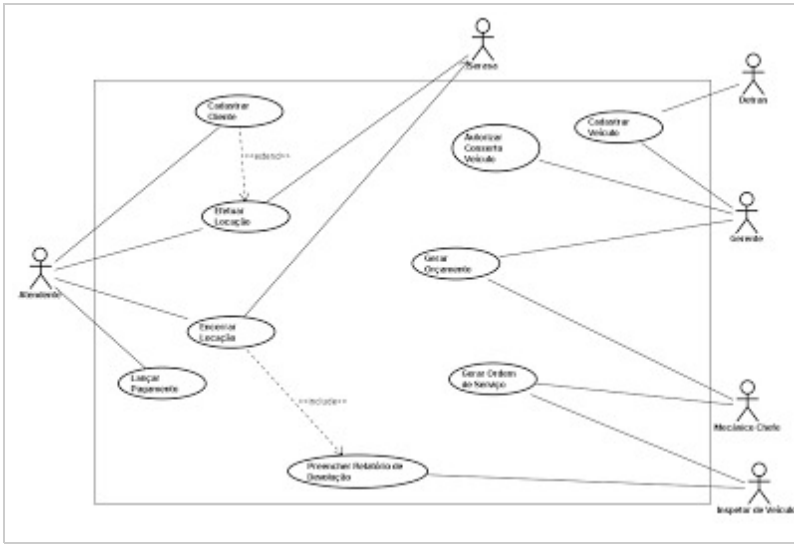


Diagrama de Caso de Uso - Locadora de Veículos

[2]***Release*** - Um *Release* é a versão de uma solução proposta que é

entregue durante a fase de Transição e que será utilizada para o teste Beta do software. A versão final, após as mudanças solicitadas e realizadas também pode ser tratada como um *Release*.

Referências

[RUP2007] Rational Unified Process; Direitos Autorais (C) IBM Corporation 2000, 2007.

[UML2010] Site oficial da Linguagem de Modelagem Unificada, em www.uml.org acessada em 04/05/2010.

Página inicial

Assinar: Postagens (Atom)