O que é UML ?

UML (Unified modeling language – Linguagem unificada de modelagem) é uma família de notações gráficas, apoiado por um meta modelo único, que ajuda na descrição e no projeto de sistemas de software, particularmente daqueles construídos utilizando o paradigma orientado a objetos(OO).

Modelagem de software

A modelagem de um software implica em criar modelos de software. A função de um modelo é capturar uma visão de um sistema físico, é uma abstração do sistema, com propósito de descrever aspectos estruturais e comportamentais do software. Este propósito determina o que deve ser incluído no modelo e o que é considerado irrelevante.

Assim, um modelo de caso de uso fornecerá uma visão dos requisitos necessários ao

sistema. Identificando as funcionalidades do software e os atores que poderão utilizá-las.

Porque modelar software?

A realização de projetos envolve cálculos que devem ser corretos e precisos. Além disso é necessário fornecer estimativas de custos, tempo de construção, material necessário, etc.

Os sistemas são dinâmicos, pois necessitam de mudanças devido a diversos fatores, como:

Os clientes necessitam de modificações ou melhorias em seus sistemas.

O mercado está sempre mudando(evoluindo).

O governo promulga novas leis, cria novos impostos e alíquotas ou modifica leis, que implica em mudanças nos sistemas

Assim, um sistema precisa ter uma documentação detalhada, precisa e atualizada para que possa ser mantido com facilidade, rapidez e correção, sem produzir novos erros ao corrigir antigos. Modelar um sistema é uma forma bastante eficiente de documenta-lo.

Fases de Desenvolvimento de software (abordagem Unified process – Processo unificado)

No processo de desenvolvimento de software temos quatro fases, que descrevemos brevemente.

Levantamento e analise de requisito

Levantamento do que o usuário deseja que o software faça.

Concepção: Onde é feito o levantamento dos requisitos

Elaboração: Onde feita a analise e o projeto do software

Construção: Onde o software é implementado e testado.

Levantamento e análise de requisito

Esta fase com o domínio do problema e tenta determinar o que o software deve fazer e se realmente é possível desenvolver o software solicitado. Na etapa de analise de requisitos, o engenheiro de software busca compreender as necessidades do usuário e o que ele deseja que o sistema realize. Isso é feito por meio de entrevistas com o usuário. São realizadas tantas entrevistas, quanto forem necessárias para compreensão do software.

Esta fase deve identificar dois tipos de requisitos:

Requisitos Funcionais e requisitos não funcionais.

Os requisitos funcionais correspondem ao que o cliente quer que o sistema realize, ou seja, as funcionalidades do sistema.

OS requisitos não funcionais correspondem as restrições, consistências e validações.

Por exemplo: em um sistema bancário: Abrir uma conta, é um requisito funcional.

Já determinar que somente pessoas maiores de 18 anos possam abrir conta corrente é um requisito não funcional.

Podem existir diversos tipos de requisitos não funcionais, como usabilidade, desempenho, confiabilidade, segurança ou interface.

Alguns requisitos não funcionais identificam regras de negócio, ou seja políticas, normas e condições estabelecidas pela empresa.

Por ex. estabelecer que após abrir uma conta, é necessário um valor mínimo de depósito é uma regra de negócio.

A UML é composta de vários diagramas, cujo objetivo é fornecer múltiplas visões do sistema a ser modelado, analisando-o e modelando-o sobre diversos aspectos, procurando-se, assim atingir a completude da modelagem, permitindo que cada diagrama complemente os outros.

Cada Diagrama da UML analisa o sistema, ou parte dele, sobre determinada ótica.

O.O – Orientação a objeto

A UML está totalmente inserida no paradigma de orientação a objetos.

Vamos ver os conceitos de O.O.

As crianças aprendem conceitos simples, tais como Pessoa, Carro e Casa, por exemplo ao fazerem isso definem Classes, ou seja grupos de objetos, sendo que cada objeto é um exemplo de determinado grupo, tendo as mesmas características e comportamento dos objetos do grupo em questão.

Vamos definir um conceito muito usado em OO, que é conceito de **Abstração**:

Abstrair é considerar isoladamente a parte do todo.

No processo de analisar as classes está envolvido um grande esforço de abstração. Por exemplo, os carros apresentam diferentes formatos e estilos. A criança deve se sentir confusa no começo ao descobrir, que o objeto amarelo e o objeto vermelho tem ambos a mesma classificação: Carro. A Partir desse momento, a criança precisa abstrair o conceito de carro, para chegar a conclusão que “carro” é um termo geral que se refere a muitos objetos.No entanto, cada um dos objetos Carro tem características semelhantes entre si. No momento em que a criança compreende este conceito, ela abstraiu uma classe: A classe Carro.

Sempre que precisamos compreender um novo conceito, criamos uma nova classe para esse conceito, muitas vezes derivando de classes mais simples.

Quando instanciamos um objeto de uma classe, um novo item representado por essa Classe, as mesmas características e comportamento de todos os objetos já instanciados.

Classe de Objetos.

A Classe representa uma categoria, e os objetos são membros dessa categoria.