



programação orientada a objetos

Elementos Históricos da UML

Luiz Felipe Neves dos Santos Siqueira – Tads Turma A

Professor: Carlos Veríssimo

Introdução:

Esta Ado tem como objetivo ampliar o conhecimento sobre os elementos da UML e a evolução ao decorrer dos anos, apontando as principais causas.

Resumindo um pouco mais, sobre o UML, que fornece uma maneira poderosa de modelar sistemas de software, permitindo aos desenvolvedores criar representações visuais que facilitam a compreensão, comunicação e documentação dos sistemas, desde sua concepção até a implementação e manutenção.

Sobre a UML

A Linguagem de Modelagem Unificada, mais conhecida como UML (é uma linguagem visual amplamente utilizada na engenharia de software para modelar sistemas de software. Essa linguagem desempenha um papel crucial no desenvolvimento de software orientado a objetos e na engenharia de sistemas.

As linguagens de modelagem orientadas a objetos surgiram entre a metade da década de 1970 e o final da década de 1980, à medida que o pessoal envolvido com metodologia, diante de um novo gênero de linguagens de programação orientadas a objeto e de aplicações cada vez mais complexas, começou a experimentar métodos alternativos de análise e projeto. A quantidade de métodos orientados a objetos aumentou de pouco mais de 10 para mais de 50 durante o período de 1989 a 1994. Muitos usuários desses métodos tiveram dificuldades para encontrar uma linguagem de modelagem capaz de atender inteiramente às suas necessidades. Basicamente, a UML permite que desenvolvedores visualizem os produtos de seus trabalhos em diagramas padronizados. Junto com uma notação gráfica, a UML também especifica significados, isto é, semântica.

As linguagens fornecem um vocabulário e as regras para combinação de palavras desse vocabulário com a finalidade de comunicar uma linguagem de modelagem é a linguagem que tem seu foco voltado para a representação conceitual e física de um sistema. O vocabulário e as regras de uma linguagem como a UML, indicam como criar e ler modelos bem-formados, mas não apontam quais modelos deverão ser criados, nem quando você deverá criá-los.

A UML É UMA LINGUAGEM DESTINADA A:

- VISUALIZAR
- ESPECIFICAR
- CONSTRUIR
- DOCUMENTAR

... ARTEFATOS DE UM SISTEMA DE SOFTWARE.

Artefato é um conjunto de informações utilizado ou produzido por um processo de desenvolvimento de software

Utilizando os Blocos de Construção:

Diagramas: O UML oferece vários tipos de diagramas para representar diferentes perspectivas de um sistema, como diagramas de classe, diagramas de sequência, diagramas de atividade e muitos outros. Cada tipo de diagrama se concentra em aspectos específicos do sistema

Classes: As classes são a base da modelagem de objetos no UML. Elas representam entidades ou conceitos no sistema e contêm atributos e métodos que definem suas características e comportamento.

Objetos: Os objetos são instâncias de classes e representam entidades específicas no sistema. Eles são usados em diagramas de sequência e diagramas de colaboração para representar o comportamento do sistema em tempo de execução.

Relacionamentos: Os relacionamentos descrevem as associações entre classes e objetos. Alguns tipos de relacionamentos comuns incluem associações, agregações, composições e heranças, que definem como as classes estão conectadas e interagem umas com as outras.

Atividades: Diagramas de atividade são usados para modelar o fluxo de trabalho e o comportamento do sistema em termos de ações, decisões e condições.

Casos de Uso: Os casos de uso descrevem os diferentes cenários de interação entre o sistema e seus atores (usuários ou sistemas externos). Eles são usados para definir os requisitos funcionais do sistema.

Estado: Os diagramas de estado são usados para modelar o comportamento de objetos em diferentes estados e como eles transitam entre esses estados em resposta a eventos.

Componentes e Pacotes: Esses elementos são usados para organizar o sistema em módulos ou componentes, o que ajuda na gerência de complexidade e na separação de responsabilidades.

Interfaces: As interfaces definem os contratos que as classes ou componentes devem seguir. Elas especificam os métodos que uma classe ou componente deve implementar.

em UML 2.0

Metodologias de desenvolvimento de software como agile foram incorporadas e o escopo da especificação UML original foi ampliado.

Originalmente, a UML especificou 9 diagramas. UML 2.x aumentou o número de diagramas de 9 para 13. Os quatro diagramas que foram adicionados são: diagrama de tempo, diagrama de comunicação, diagrama de visão geral de interação e diagrama de estrutura composta. A UML 2.x renomeou diagramas de gráfico de estado para diagramas de máquina de estado.

UML 2.x adicionou a capacidade de decompor o sistema de software em componentes e subcomponentes.

Evolução da UML com os principais tópicos:

Diversidade de Métodos (1990): Nas décadas de 1980 e 1990, havia uma proliferação de diferentes métodos de desenvolvimento de software, cada um com suas próprias notações e abordagens. Isso tornou difícil a colaboração e a comunicação entre equipes de desenvolvimento que usavam métodos diferentes.

Início da Padronização (1994-1995): A necessidade de uma linguagem de modelagem comum levou à formação de um consórcio chamado "UML Partners," que incluiu empresas de software líderes, como Rational Software, IBM, e outras. Eles começaram a trabalhar em uma linguagem de modelagem unificada que combinaria as melhores práticas de vários métodos existentes.

Lançamento da UML 1.0 (1997): A UML 1.0 foi a primeira versão oficial da linguagem e incluiu uma variedade de diagramas, como diagramas de classe, diagramas de caso de uso, diagramas de sequência e muitos outros. Foi um marco importante na padronização da modelagem de software.

Revisões e Expansões (2000): Durante a década de 2000, várias revisões da UML foram lançadas, incluindo a UML 1.3, 1.4 e 2.0. Essas versões introduziram melhorias e refinamentos na linguagem.

OMG e Padronização (2000): A UML foi adotada como uma especificação pela Object Management Group (OMG), uma organização de padronização reconhecida internacionalmente. Isso solidificou ainda mais a posição da UML como uma linguagem de modelagem padrão para a indústria de software.

UML 2.0 e UML 2.1 (2005): A UML 2.0 trouxe uma reformulação significativa da linguagem, tornando-a mais extensível e capaz de modelar sistemas mais complexos. A UML 2.1 introduziu pequenas melhorias e correções. **UML 2.4 (2011):** Esta versão trouxe estabilidade à linguagem, com foco na correção de erros e ambiguidades e na garantia de compatibilidade com versões anteriores. **UML 2.5 (2015):** A UML 2.5 introduziu melhorias significativas na documentação e na simplificação de alguns conceitos. Também permitiu que a UML fosse usada em um ambiente mais ágil e de desenvolvimento baseado em modelos.

(Após 2015): Após a versão 2.5, a UML continuou a evoluir com pequenas atualizações e refinamentos. A linguagem ainda é amplamente usada na indústria de desenvolvimento de software e está incorporada em várias ferramentas de modelagem e engenharia de software.

Concluindo

A Uml desempenha um papel fundamental na comunicação e na documentação de sistemas, facilitando a colaboração entre equipes multidisciplinares e a compreensão dos sistemas por parte de todas as partes interessadas. Ela ajuda a identificar requisitos, visualizar a arquitetura, detectar problemas de design e muito mais. pode ser usada para modelar várias fases de um sistema, desde os primeiros contatos até a geração do código, continuando a evoluir para atender às necessidades em constante mudança das indústria.

Referência bibliográfica:

Site de artigos como, Ufu, UFMG, google acadêmico, e Usp.

E com uma breve revisão em Livros digitais como:

Uml - Guia do Usuário

UML Ferramentas CASE.