OTES12 – Tópicos Avançados em Engenharia de Software

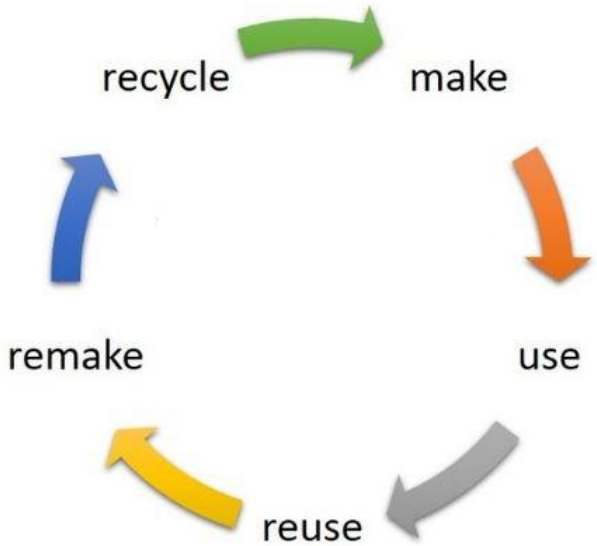
Universidade do Estado de Santa Catarina
Centro de Ciências Tecnológicas – DCC

Prof. Dr. William Alberto Cruz Castañeda

2020/2

[Processos de Reutilização]

Como a reutilização deve ser praticada?



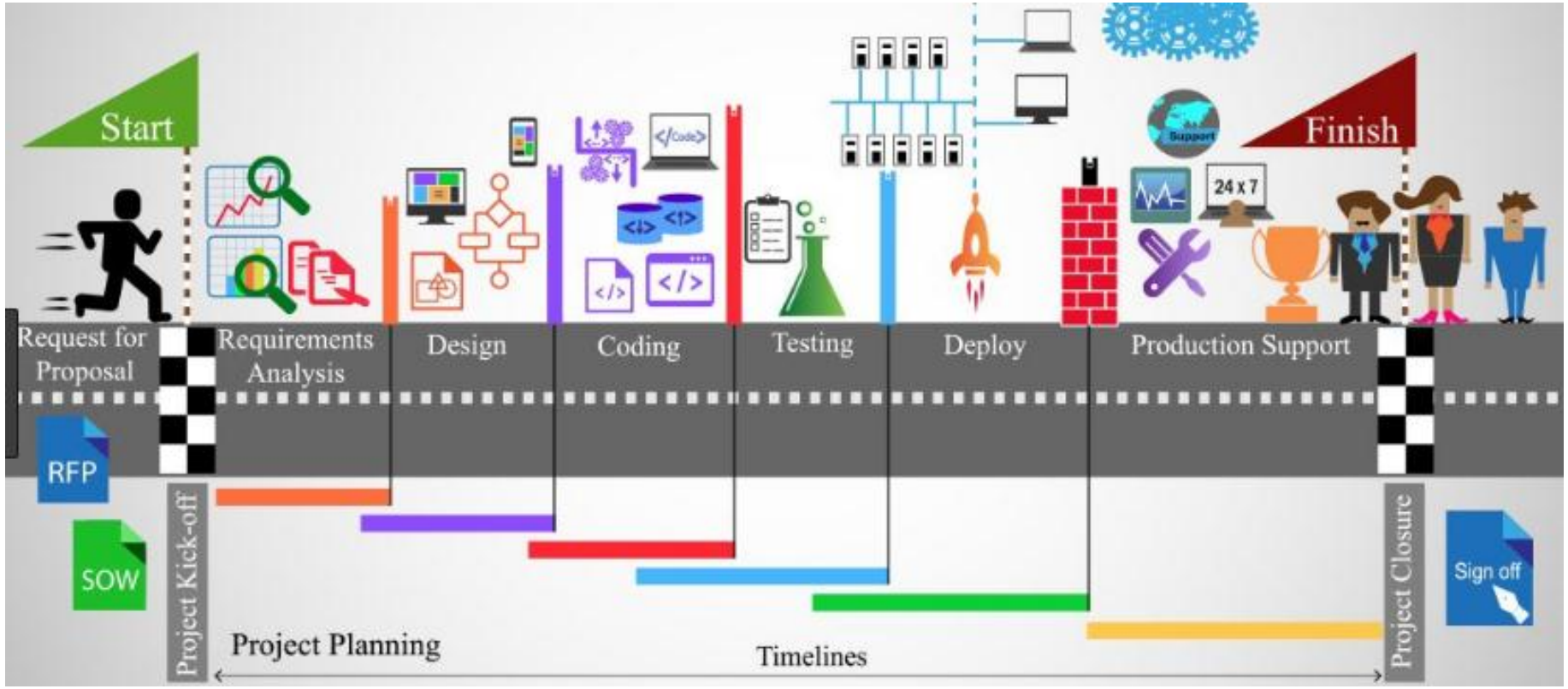
Que processos precisamos?

O que é um processo?

Coleção de tarefas relacionadas que levam a um produto;

Aninhados e podem ser divididos em subprocessos até atingir tarefas atômicas;

Processos de software se referem a todas as tarefas necessárias para produzir e gerenciar software, enquanto os **processos de reutilização** são o subconjunto de tarefas necessárias para uma reutilização bem-sucedida de software dentro de uma empresa.



Exemplos de processo de reutilização → produção de componentes reutilizáveis, identificação de componentes reutilizáveis; desenvolvimento de componentes reutilizáveis; validação de componentes reutilizáveis;

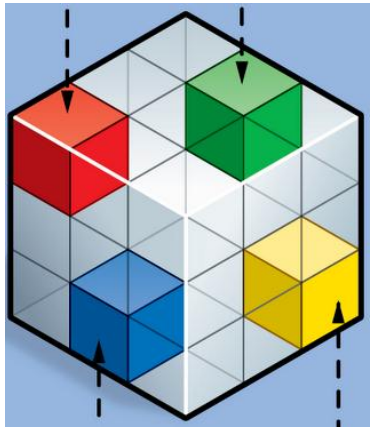
Desenvolvimento de aplicativos → não é um processo de reutilização puro, mas sim um processo tradicional de software que é obviamente impactado pela reutilização;

Analisar quais processos são necessários para que a reutilização seja executada com eficiência.

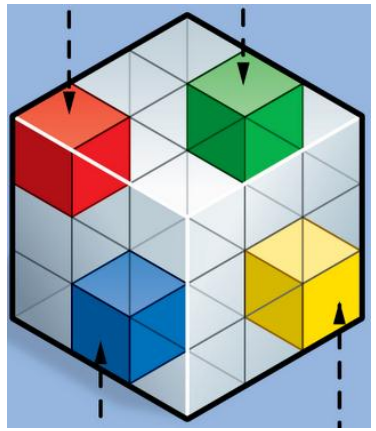
[Processo de Reutilização]

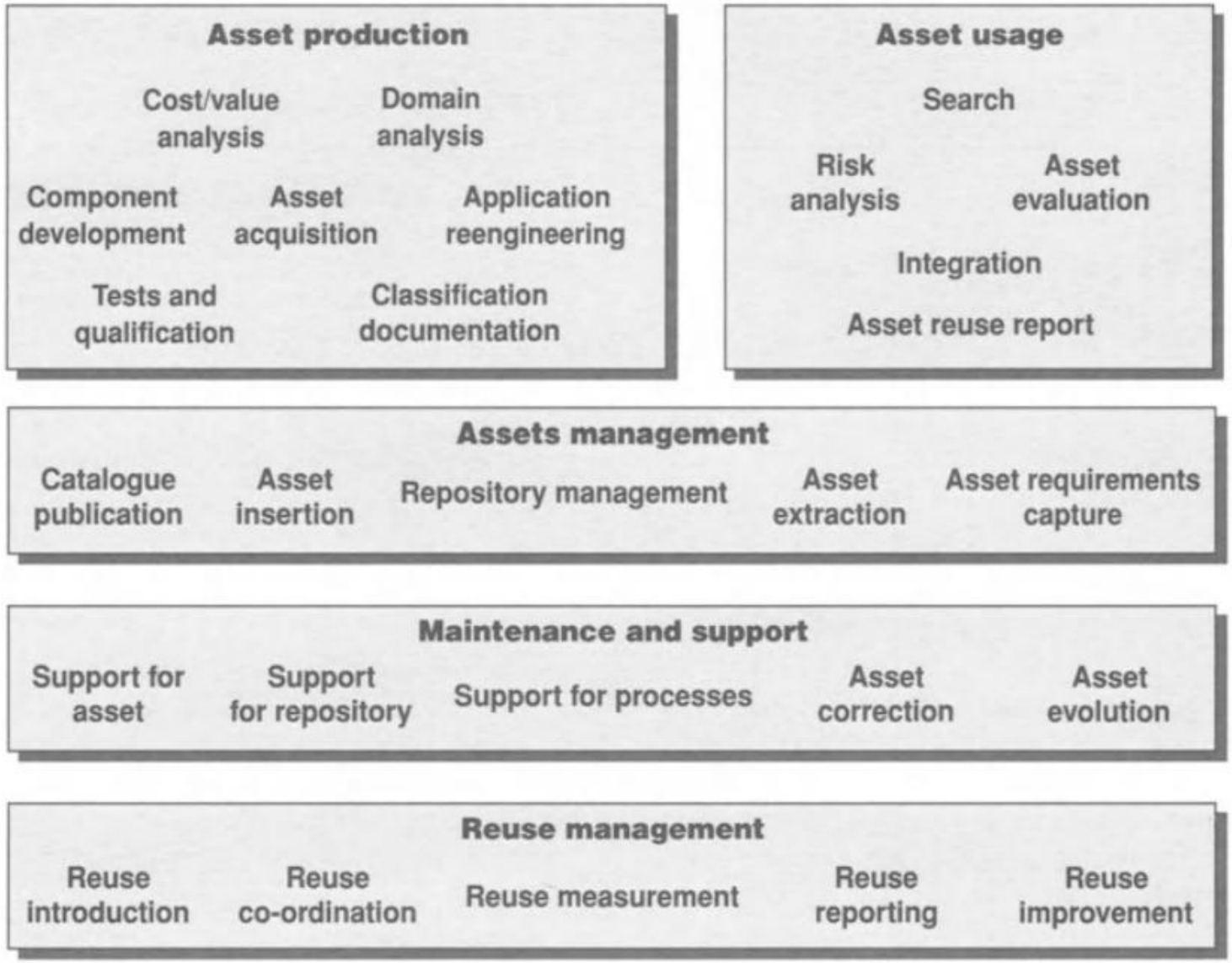
Principais Processos de Reutilização

- **Produção de componentes →** identificação, desenvolvimento e classificação de componentes;
- **Uso de componentes →** localização e avaliação assim como obtenção da reutilização real, integrando-os aos aplicativos que estão sendo desenvolvidos. Essa é a parte do processo tradicional de desenvolvimento de aplicativos que é impactado pela reutilização;
- **Gerenciamento de componentes →** armazenamento, gerenciamento de repositório e disseminação;



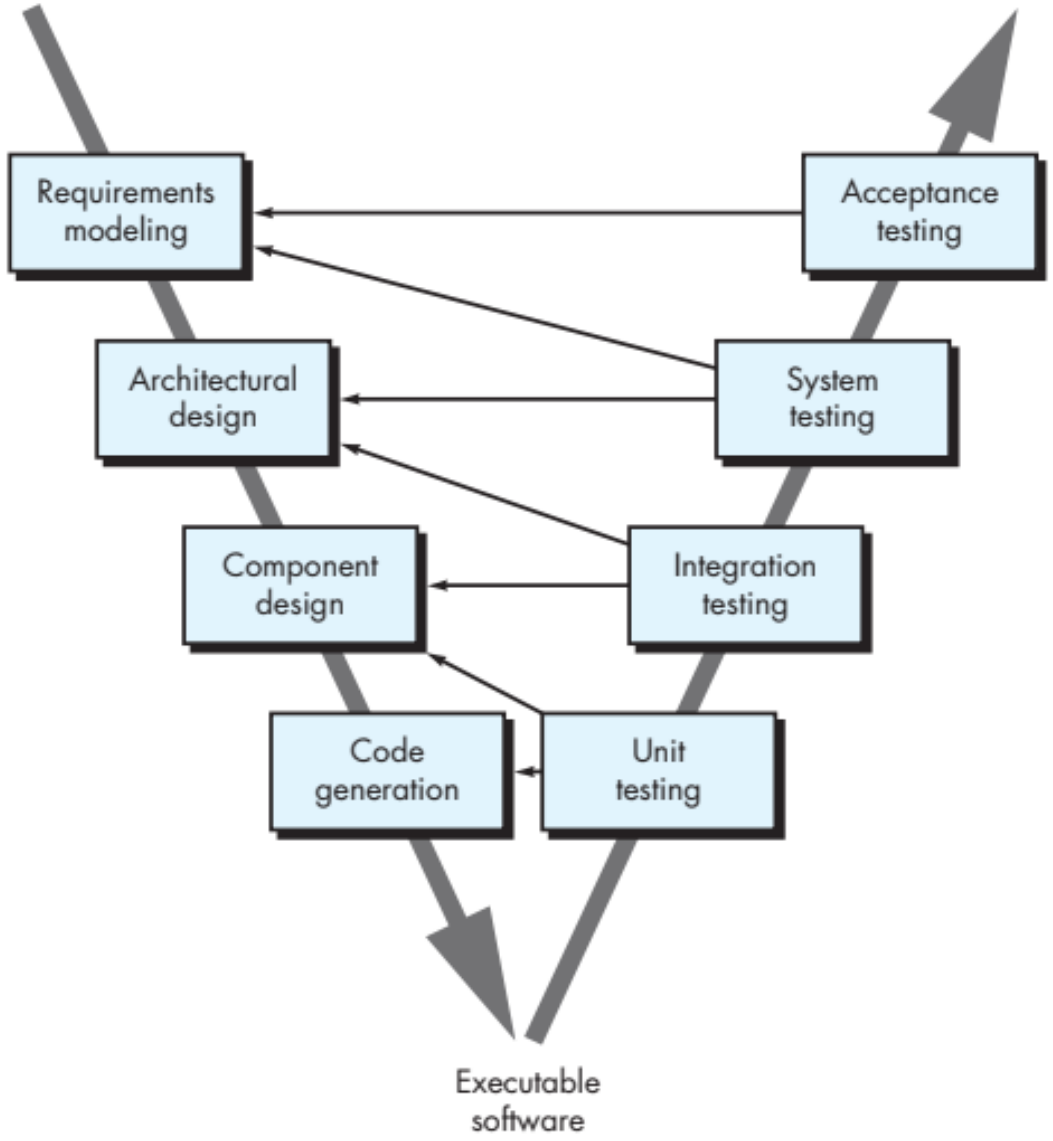
- **Manutenção e suporte** → suporte para uso, suporte metodológico e manutenção corretiva e evolutiva;
- **Gerenciamento de reutilização** → monitoramento da reutilização em uma empresa ou departamento.
  - Inclui o rastreamento dos resultados da reutilização e melhoria dos processos de reutilização;
  - Afeta os processos de software existentes;
  - Define o processo para desenvolver um aplicativo a partir dos componentes existentes.




Asset production
Cost/value analysis
Domain analysis
Component development
Asset acquisition
Application reengineering
Tests and qualification
Classification documentation
Asset usage
Search
Risk analysis
Asset evaluation
Integration
Asset reuse report
Assets management
Catalogue publication
Asset insertion
Repository management
Asset extraction
Asset requirements capture
Maintenance and support
Support for asset
Support for repository
Support for processes
Asset correction
Asset evolution
Reuse management
Reuse introduction
Reuse co-ordination
Reuse measurement
Reuse reporting
Reuse improvement

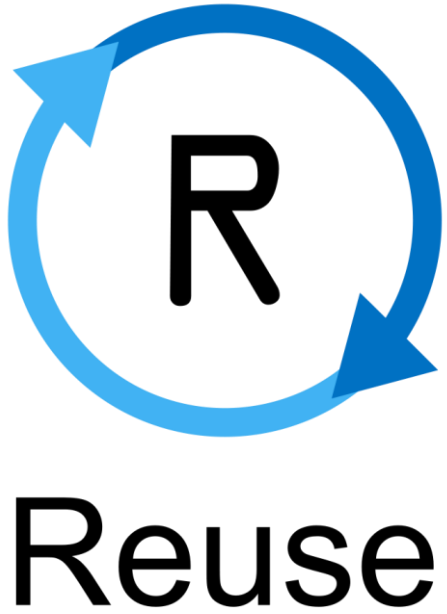
Modelo em V

- Descreve ações de garantia de qualidade associadas às atividades de comunicação, modelagem e construção inicial;
- Requisitos básicos do problema refinados para representações progressivamente mais detalhadas e técnicas do problema e de sua solução;
- Código gerado → executa série de testes que validam cada um dos modelos;


Requirements modeling
Acceptance testing
Architectural design
System testing
Component design
Integration testing
Code generation
Unit testing
Executable software

E o reuso?

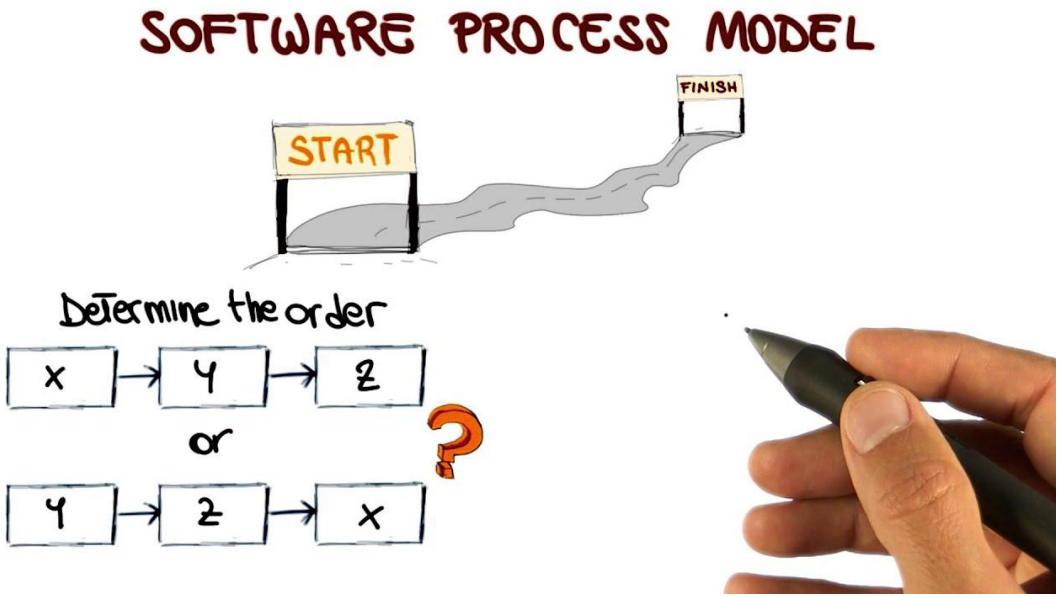
- Reutilização na fase de design quando uma arquitetura usada em outro projeto é utilizada;
- Reutilização na fase de projeto quando algoritmos comprovados de outros projetos são utilizados;
- Reutilização na fase de codificação quando bibliotecas são usadas;



Modelo de Processo REBOOT

- Concentra-se na cadeia de produtores e consumidores;
- Como em um modelo V tradicional, mapeia etapas de produção para as etapas de validação;
- Componentes desenvolvidos pelo projeto X para serem reutilizados pelo projeto Y;
- Componentes qualificados pelo projeto X para serem avaliados pelo projeto Y;
- Componentes classificados e inseridos em um repositório do projeto X para serem recuperados e extraídos no repositório do projeto Y;

- Fácil de entender e introduz;
- Principal desvantagem → pode ser interpretado como centrado no projeto e ajustado a componentes de baixa granularidade;
- Aplicados com sucesso nos domínios de sistema embarcado e controle (industrial e telecomunicações);



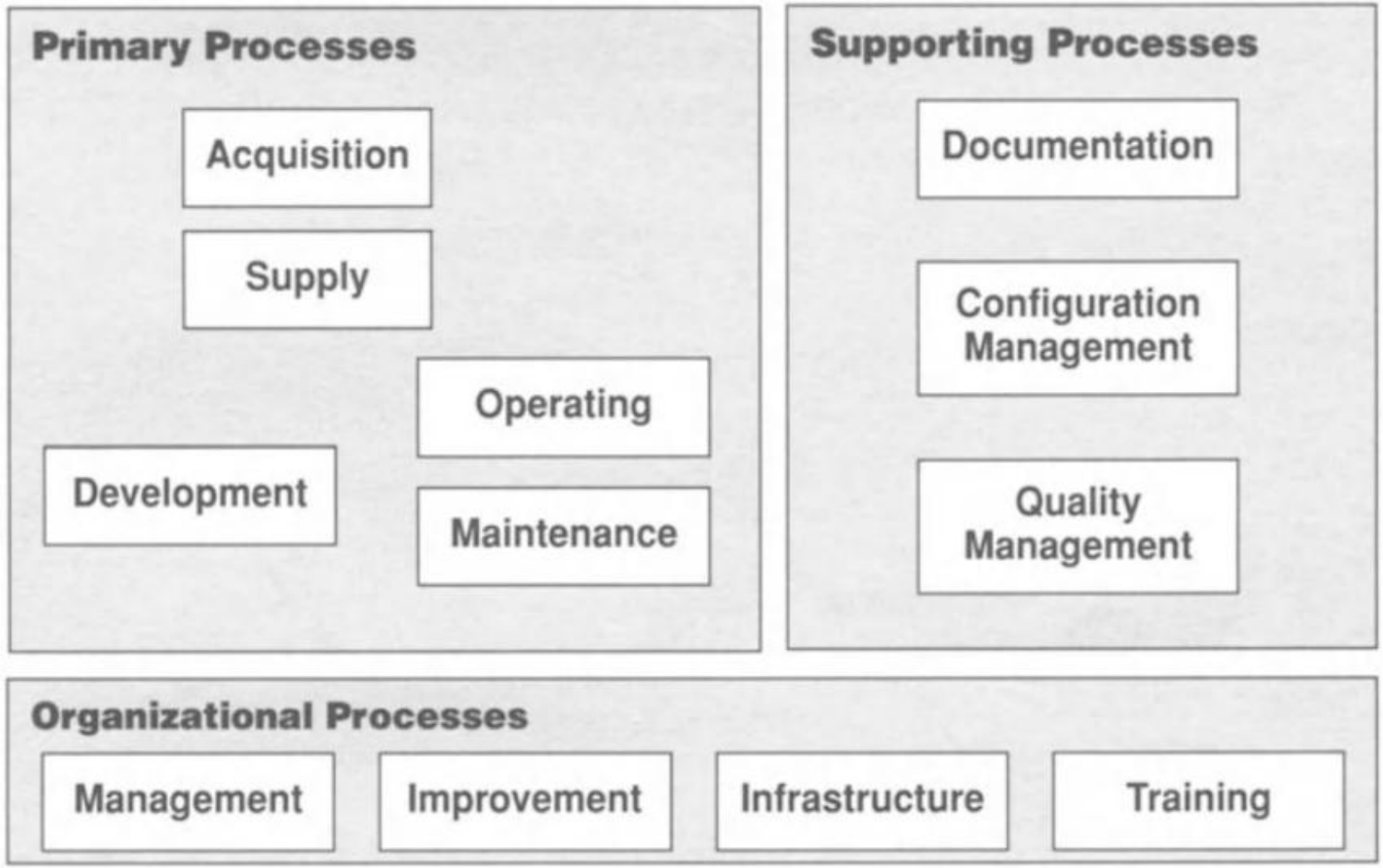
Modelo de Processo de Jacobson, Griss e Jonsson

- ***Application Family Engineering*** – projeta e implementa uma arquitetura funcional ajustada a uma família de aplicativos ou a uma linha de produtos. Arquitetura comum e estável é o ponto de partida para o desenvolvimento de aplicativos e componentes;

- ***Component System Engineering*** – desenvolve componentes reutilizáveis, analisando os requisitos e sua variabilidade através das diferentes aplicações de uma família;

- ***Application System Engineering*** – processo de construção de aplicativos com base na arquitetura comum e no uso de componentes existentes. Fornece um processo de modelagem amplo e claro para reutilização

Como a reutilização afeta os processos tradicionais de software?



Traditional Software Process (according to ISO 12207)	Variation introduced by reuse practices
	Primary processes
Acquisition	Must include not only turnkey application/system acquisition, but also software asset acquisition (see the Section in Chapter 2 on acquiring software assets).
Supply	This process is mostly unchanged if the final customer is not aware of reuse. Otherwise, reuse may lead to a trade-off between costs and functionality.
Development	This is traditionally an application or system development process. • Each step of this process (analysis, architecture, design . . .) is impacted by reuse: reusable assets must be searched for, evaluated and integrated. • A new 'domain engineering' process must be introduced (domain analysis and vertical reusable asset development are part of it).
Operating	Reuse has no impact on this process.
Maintenance	Application or system maintenance must be differentiated from asset maintenance. This aspect has been discussed above.

Supporting processes	
Documentation	System documentation (both user documentation and engineering documentation) must be differentiated from asset documentation (mainly asset user documentation).
Configuration management	Assets may be managed in the same CM repository as applications, but asset changes should be controlled separately from application changes. In both cases a new asset management process is necessary.
Quality management	The quality system applies both to systems development and asset development. A potential difference is that assets are internal work products, not delivered directly to external customers.
Organizational processes	
Management	Should include reuse scoping, reuse planning, reuse control.
Infrastructure	Should include the management of reuse tools, and in particular the technical infrastructure necessary to give access to and manage an asset repository.
Training	Should include training in reuse, for managers and practitioners.
Improvement	Should include reuse process definition, assessment and

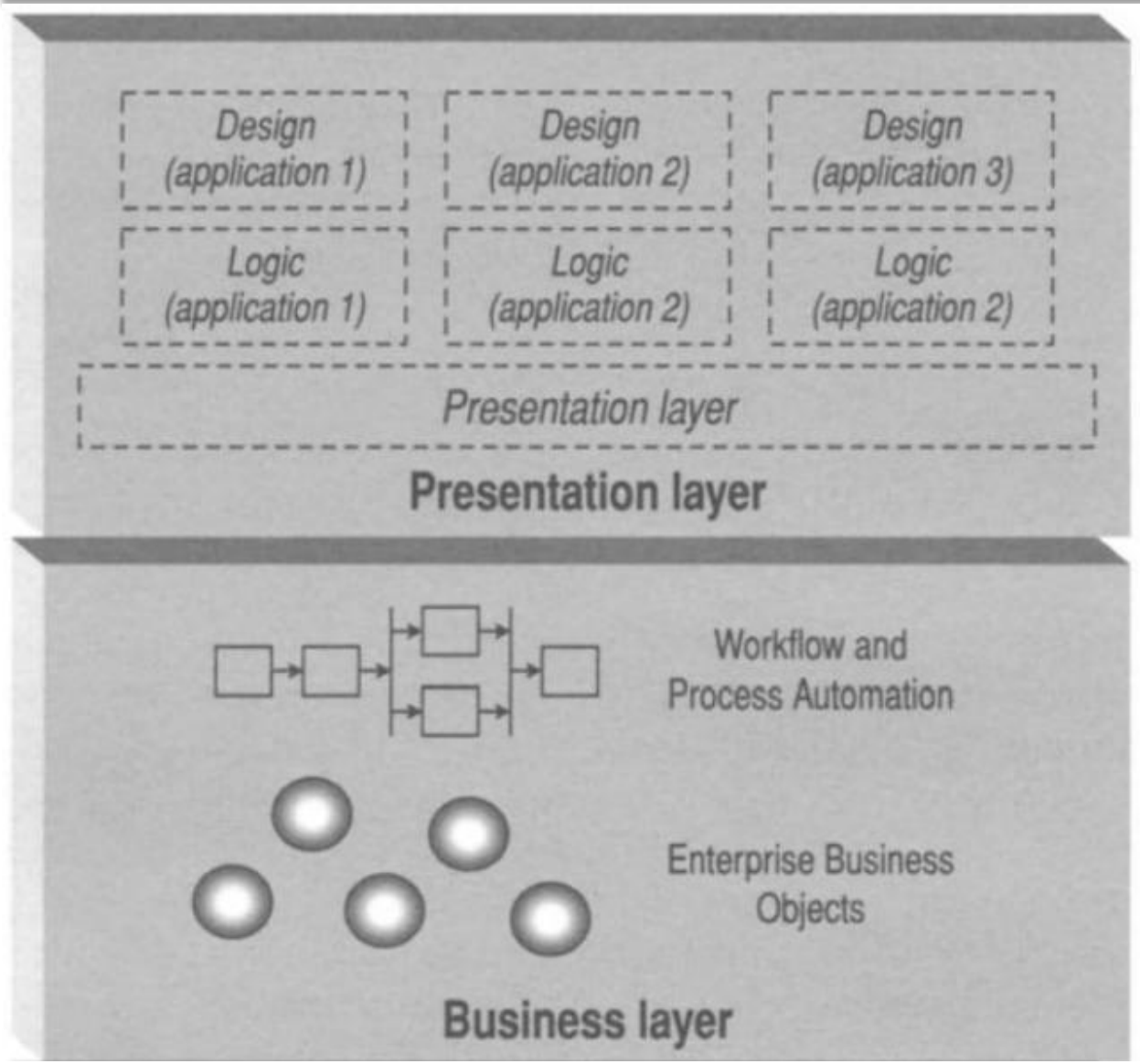
[Técnicas e tecnologias de reutilização]

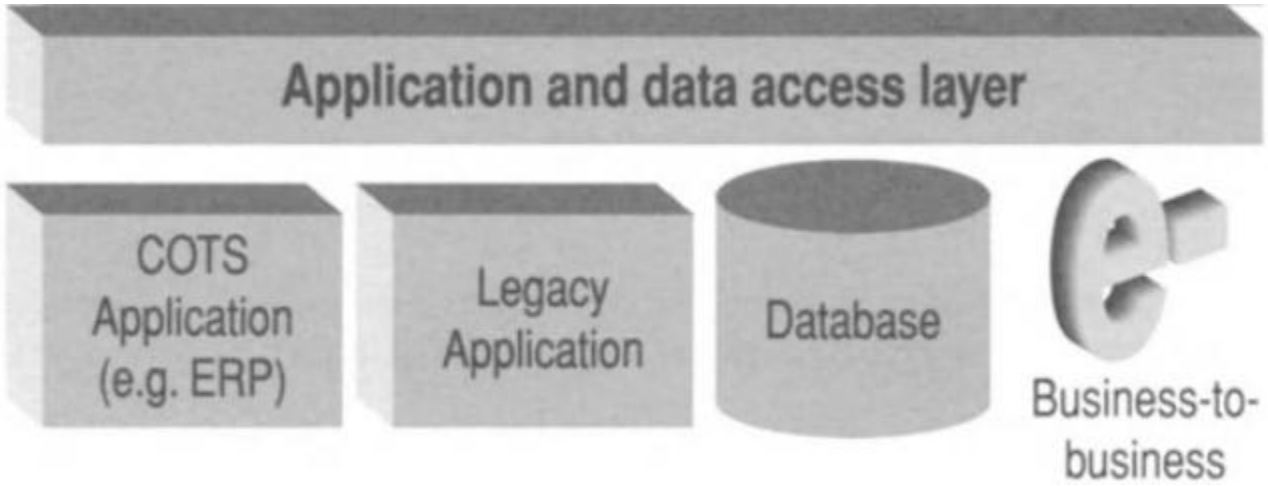
Arquiteturas de Reuso

- Arquitetura → definida como componentes de sistema identificados, conectados e a natureza dessas conexões (protocolos de comunicação, sincronização e acesso a dados);
- Determinam os tipos de tecnologias usadas para construir sistemas de software, os tipos de componentes envolvidos e os processos para criar e reutilizar componentes;

Arquiteturas de Três Camadas

- Evolução do modelo tradicional cliente/servidor de duas camadas;
- Adequado para grandes aplicativos, onde vários clientes acessam um ou mais bancos de dados distribuídos;
- Parte da lógica de negócios não faz parte do aplicativo cliente e não é implementada no banco de dados, mas é separada em uma camada intermediária chamada servidor de negócios;


Design (application 1)
Design (application 2)
Design (application 3)
Logic (application 1)
Logic (application 2)
Logic (application 2)
Presentation layer
Presentation layer
Workflow and Process Automation
Enterprise Business Objects
Business layer


Application and data access layer
COTS Application (e.g. ERP)
Legacy Application
Database
Business-to-business

Arquitetura Multi Camadas e habilitada para a Web

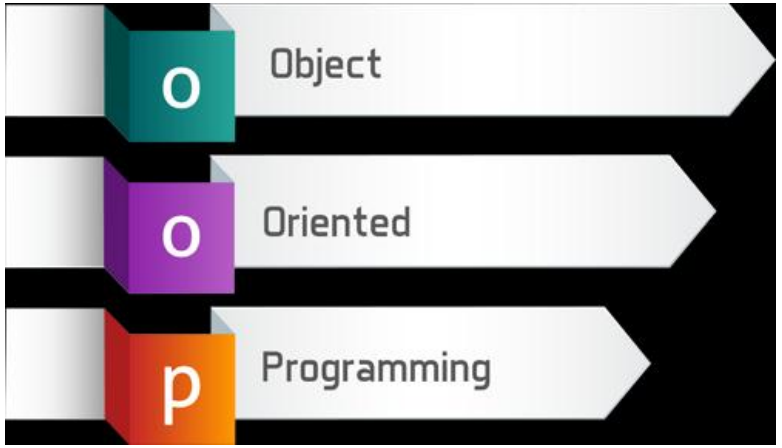
- Arquiteturas de três camadas podem ser desenvolvidas em arquiteturas de várias camadas subdividindo cada camada em várias subcamadas

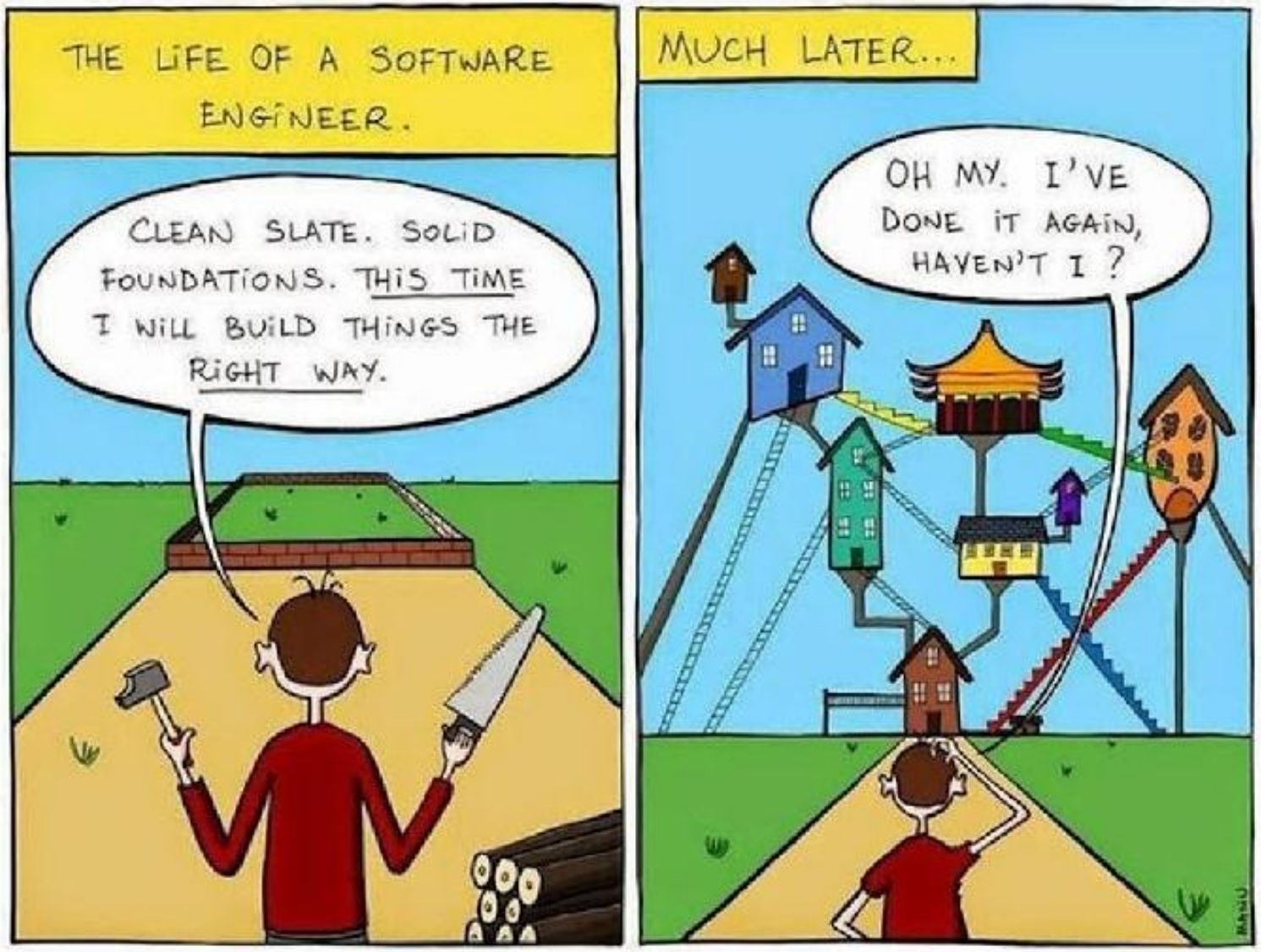
Arquiteturas de Sistemas da Internet

- A Internet e padrões associados forneceram um modelo aberto e flexível;
- Ampla disponibilidade de fontes torna possível observar a arquitetura dos sistemas de maneira diferente, concentrando-se na integração de fontes de informações em vez de focar na integração de back-end;
- Introduzem um novo paradigma, de construção de sistemas e componentes altamente genéricos e reutilizáveis que processam informações em formato padrão e a integração desses sistemas;

[Técnica Orientada Objetos]

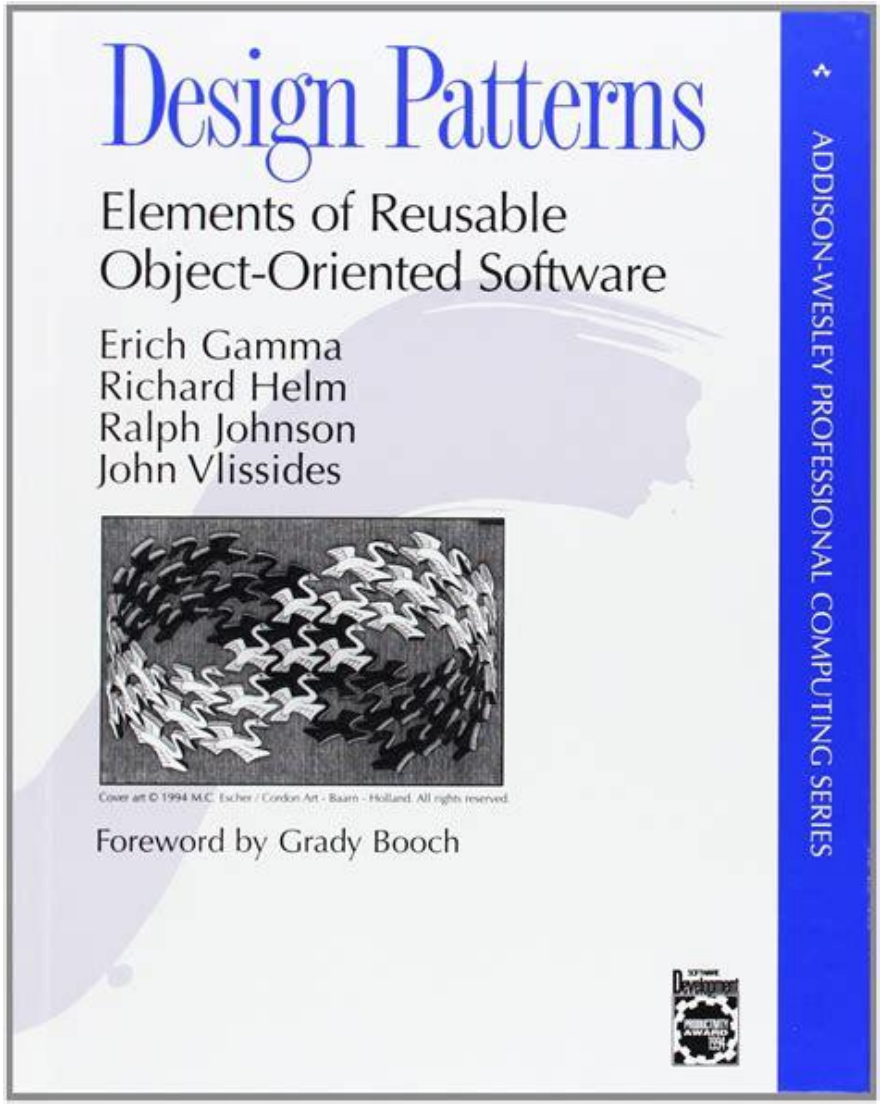
- Fornecem métodos e mecanismos para estruturar modelos e código de programa para corresponder aos objetos encontrados no domínio do problema;
- Permitir o desenvolvimento de sistemas altamente modulares (pilares da OOP);
- Consiste em análise (OOA), design (OOD) e implementação (OOP);
- OOA – analisa um sistema usando técnicas orientadas a objetos, fornecem elementos de modelagem e relacionamentos que permitem aos analistas capturar tudo de importância em um sistema;
- OOD – transforma modelos OOA em modelos de design com propriedades de design (UML)




THE LIFE OF A SOFTWARE ENGINEER.
CLEAN SLATE. SOLID FOUNDATIONS. THIS TIME I WILL BUILD THINGS THE RIGHT WAY.
MUCH LATER...
OH MY. I'VE DONE IT AGAIN, HAVEN'T I ?

Padrões de Projeto

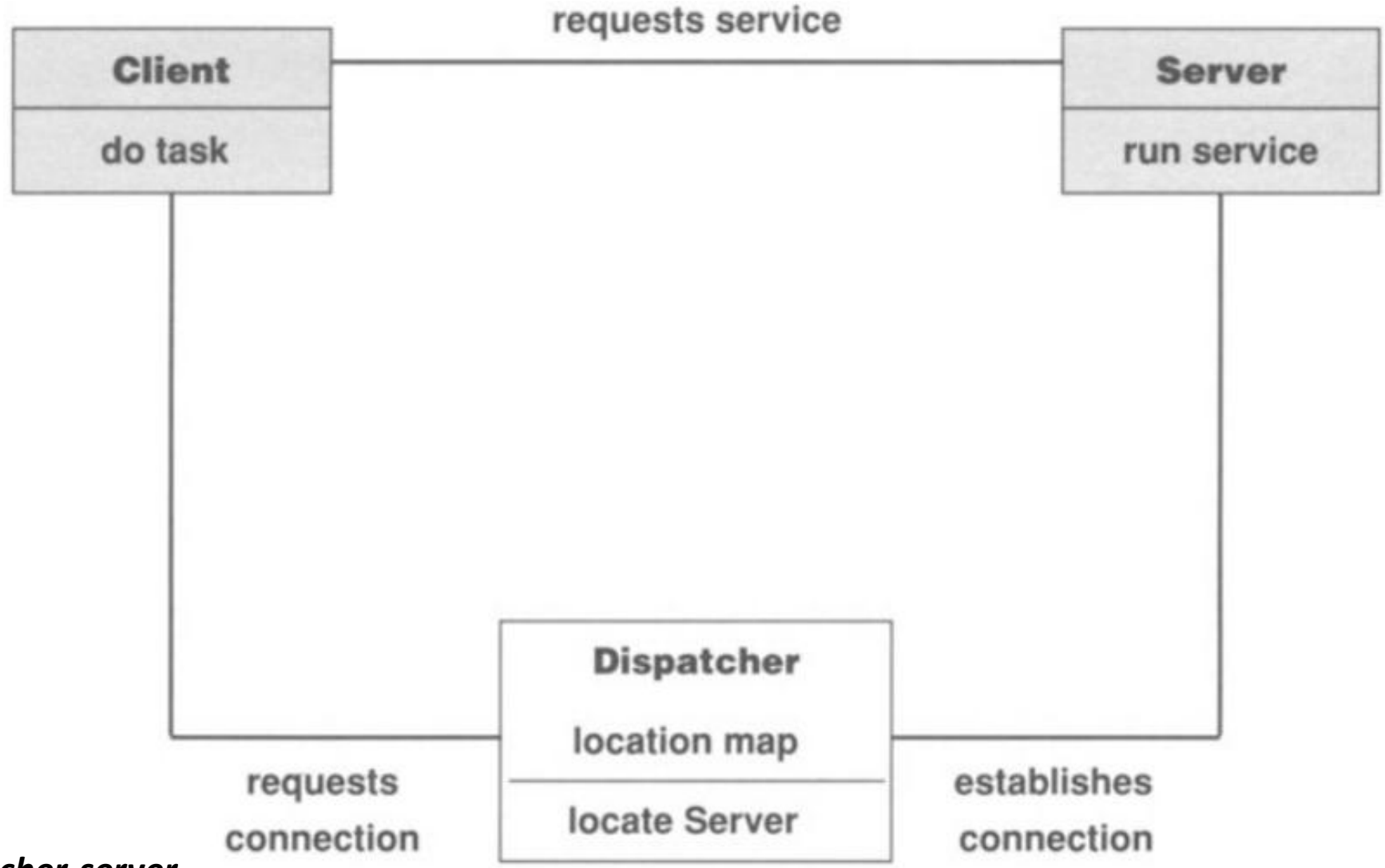
- Descreve um contexto da aplicação, um problema de design que ocorre nesse contexto e uma solução para o problema que foi provado pela experiência como eficaz;
- Documenta uma solução concreta e reutilizável para um problema arquitetural recorrente;
- Biblioteca de padrões de projetos → biblioteca significa um livro que descreve um ou mais padrões de projetos;



- Abrangem desde análises e arquitetura geral até detalhes específicos de design ou implementação;

- Coleção estruturada de padrões de projetos para um domínio específico → linguagem de padrões;

- Não suportam a reutilização de código de maneira sistemática;



Pattern client-dispatcher-server

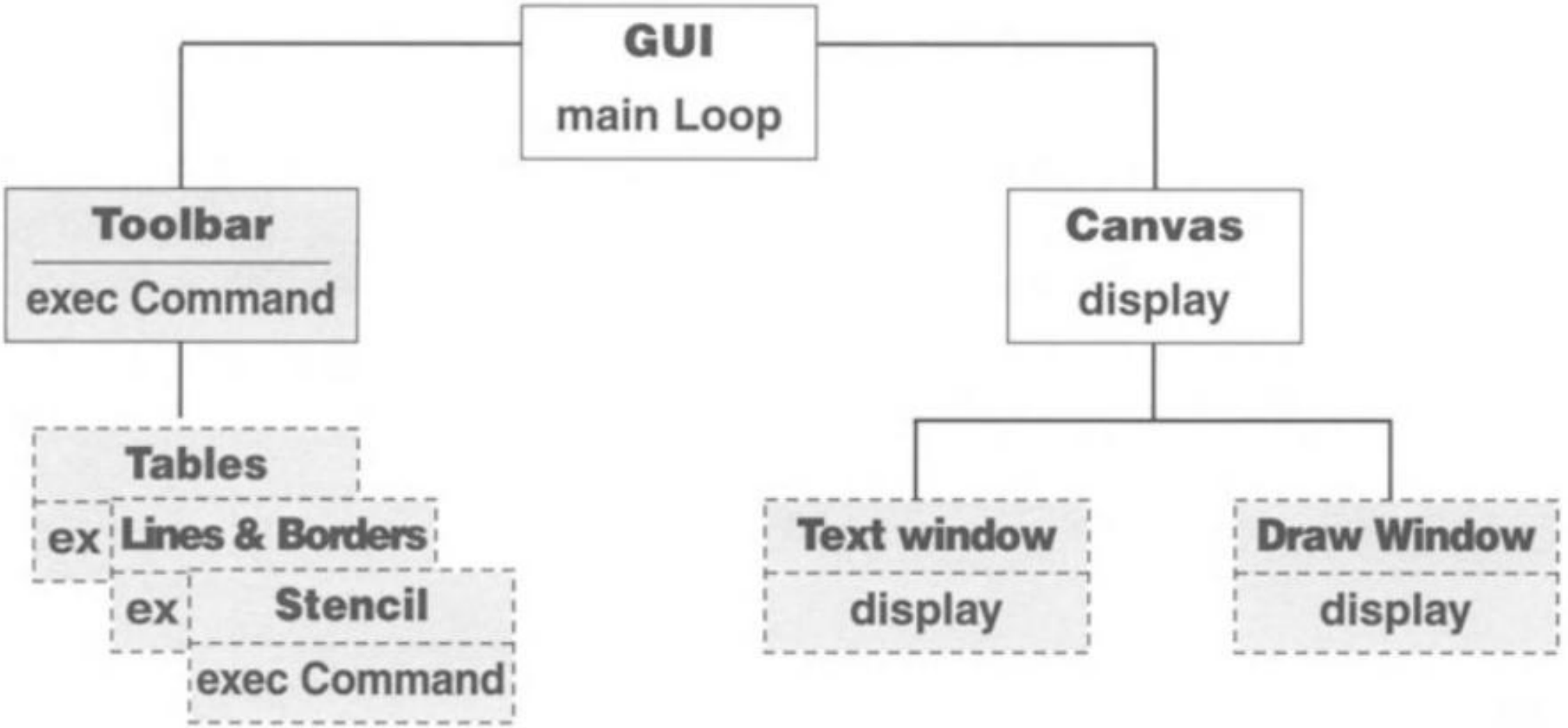
Frameworks Orientados a Objetos

- Conjunto integrado de artefatos de software reutilizáveis e extensíveis para um domínio específico;
- OOP → conjunto de classes e relações que descrevem estruturas de objetos cooperantes;
- Aplicações geradas personalizando aspectos variáveis da estrutura;

Tipos de personalização

- ***Black-box customization*** – fornece uma implementação específica de cada aspecto variável. Usuário gera um aplicativo selecionando os componentes específicos a serem conectados;

- ***White-box customization*** – fornece uma implementação genérica de cada aspecto variável. Especializado pelo usuário para aplicações concretas. Flexível, mas requer uma compreensão mais profunda do design e implementação do framework.

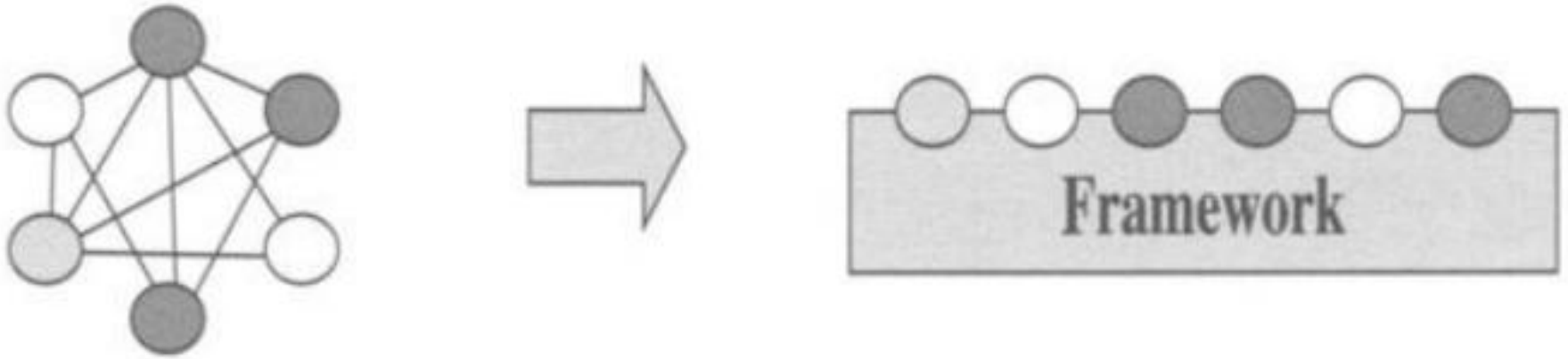

GUI
main Loop
Toolbar
exec Command
Tables
ex
Lines & Borders
ex
Stencil
exec Command
Canvas
display
Text window
display
Draw Window
display

- Framework consiste em código reutilizável e design reutilizável;
- Aplicativos criados nunca são projetados do zero. O design do framework é usado como ponto de partida;
- Frameworks OO classificados em duas categorias:

1. Aplicativos (ou comerciais ou verticais);
2. Middleware (ou horizontal);

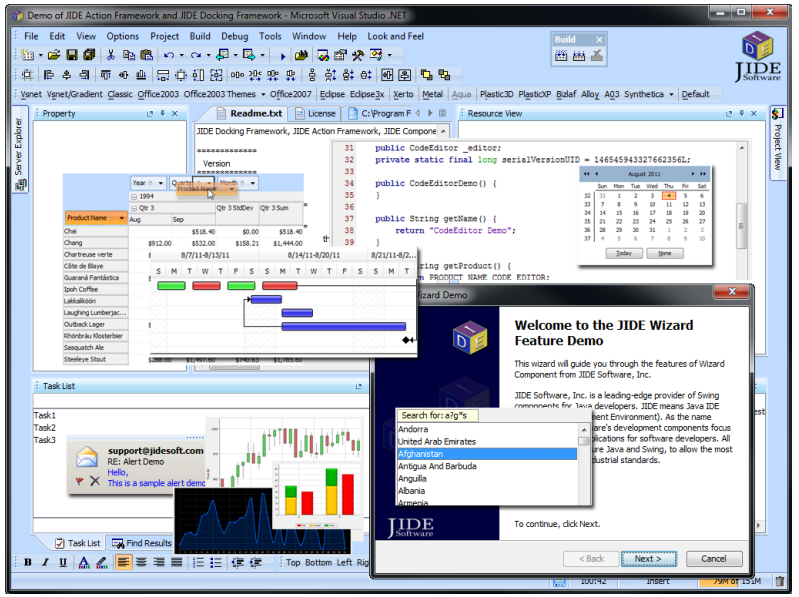
[Desenvolvimento baseado em componentes]

- Consiste na montagem de componentes de software existentes para construir sistemas complexos;
- Componentes são unidades de programa disponíveis em formato binário;
- Desenvolvedor fornece interface bem definida, mas não o código fonte.
- Desenvolvimento baseado em componentes permite estabelecer uma fronteira entre o desenvolvimento de reutilizáveis e sua reutilização;



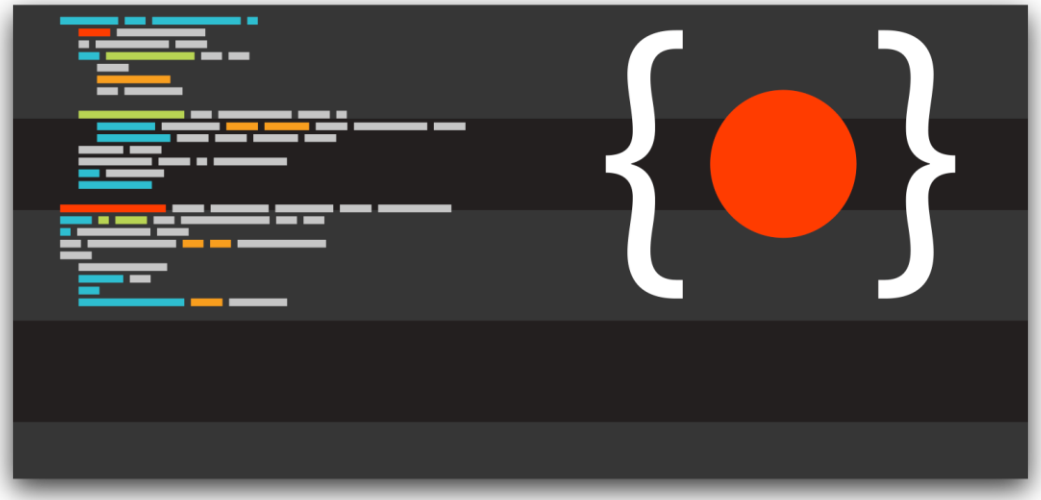
Componentes do lado do Cliente

- Implementam parte do design da apresentação e das camadas lógicas;
- A lógica da aplicação define o fluxo de controle da aplicação e a maneira como os componentes interagem. Não definido pelos componentes, isso reduziria sua reutilização e introduz alto acoplamento;



Componentes do lado do Servidor

- Objetos de negócios implementados como componentes em execução na camada de negócios;
- Implementa a estrutura e o comportamento de entidades comerciais ou processos de negócios de nível superior que envolvem vários componentes;
- Usados pela camada de apresentação para chamar serviços comerciais e exibir informações comerciais;

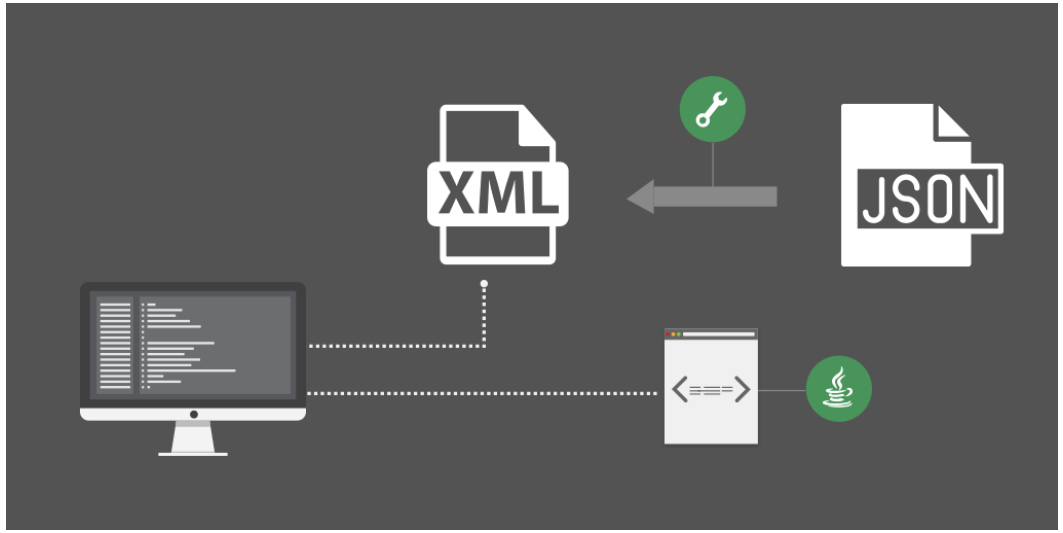


Componentes da Internet

- Componentes gráficos montados em uma página da Web, geralmente como parte de um portal horizontal ou vertical;
- Modularidade e fácil integração;

Da reutilização de componentes à reutilização de informações

- Componentes da Internet usados para acessar, processar e exibir informações em formato padrão → reutilizar as informações;
- Vocabulário comum XML, JSON, Ontologias;
- Uso de protocolos para troca de dados entre sites



	Strength	Weakness
OOT	Provides basic reuse enablers like modularity, information hiding, aggregation and specialization.	Requires significant modelling effort. Usage of OOT does not guarantee reuse.
Design patterns	Facilitate retrieval of design solutions, provide guidelines for the development process, improve communication among designers.	Implementation from scratch.
Frameworks	Reuse of object model (design, code) plus architecture.	Framework development requires high domain expertise and deep understanding of object design. Usage of a framework imposes an architecture.
Components	Development of external markets. Easy to integrate.	Less customizable.

[Métricas de Reutilização]

Objetivos

- **G1** trata da necessidade de um gestor de reuso para monitorar a atividade de reutilização através da utilização do repositório;
- **G2** entende como um projeto reutiliza componentes de um repositório. Interesse do gerente de reutilização e do gerente de projeto. Corresponde um ponto de vista técnico. A medida nível de reutilização (RL) definido como **RL = reused software / total software;**
- **G3** representa a necessidade de um gerente de reutilização saber o custo dos componentes no repositório. **Custo relativo de reutilização (RCR)**→ mede o custo de reutilização de um componente **Custo relativo de escrita para reuso (RCWR)** → mede o custo do desenvolvimento de um componente;

- **G4** ponto de vista da alta gerencia, interessada no aspecto econômico da reutilização. Às vezes, o fator determinante para a reutilização não é a redução de custos, mas a redução do tempo de colocação no mercado ou o aumento da qualidade. Número de reutilizações necessárias para pagar o custo inicial do seu desenvolvimento. **LP = RCWR / (1-RCR)**
- **G5** representa o ponto de vista do desenvolvedor e o que ele espera dele. Relação entre a qualidade de um componente e sua reuso. Qualidade = confiabilidade;
- **G6** mede o potencial de reutilização de um componente. Representa o ponto de vista do gerente de reuso ao decidir se aceita ou não o componente no repositório. Dois aspectos devem ser avaliados.
- Internos – relacionados à qualidade;
- Externos – integridade funcional e portabilidade;

Goal	Purpose	Attribute	Entity	Viewpoint	Context
G1	Characterize	The utilisation	Of the repository	From the point of view of the reuse manager	In the context of a division of the company
G2	Evaluate	The extent of reuse	In a project	From the point of view of the project manager and reuse manager	In the context of a division of the company
G3	Evaluate	The variation in cost	Of reusable assets	From the point of view of the reuse manager	In the context of a division of the company

G4	Evaluate	The economic aspects	Of the software process including reuse	From the point of view of senior management	In the context of the whole company
G5	Evaluate	The quality	Of reusable assets	From the point of view of reusers	In the context of a project
G6	Predict	The reusability	of reusable assets	From the point of view of the reuse manager	In the context of a division of the company

OTES12 – Tópicos Avançados em Engenharia de Software

Universidade do Estado de Santa Catarina
Centro de Ciências Tecnológicas – DCC

Prof. Dr. William Alberto Cruz Castañeda

2020/2