

ARQUITETURA DE SOFTWARE

**LINHAS DE PRODUTO
DE SOFTWARE**

Bryan Keven Rocha Brauna

Luis Felipe Pereira Garcia

Lara Amanny Ramos de Sousa

Marcelo Albuquerque Queiroz



Sumário

Introdução	5
Conceitos	12
Definições	14
Exemplos	16
Padrões Arquiteturais	20
Arquiteturas de referências	25
Abordagens de apoio	32
Design Arquitetural	33
Desafios na arquitetura de software	91

Definições & Conceitos

!



Introdução

Introdução

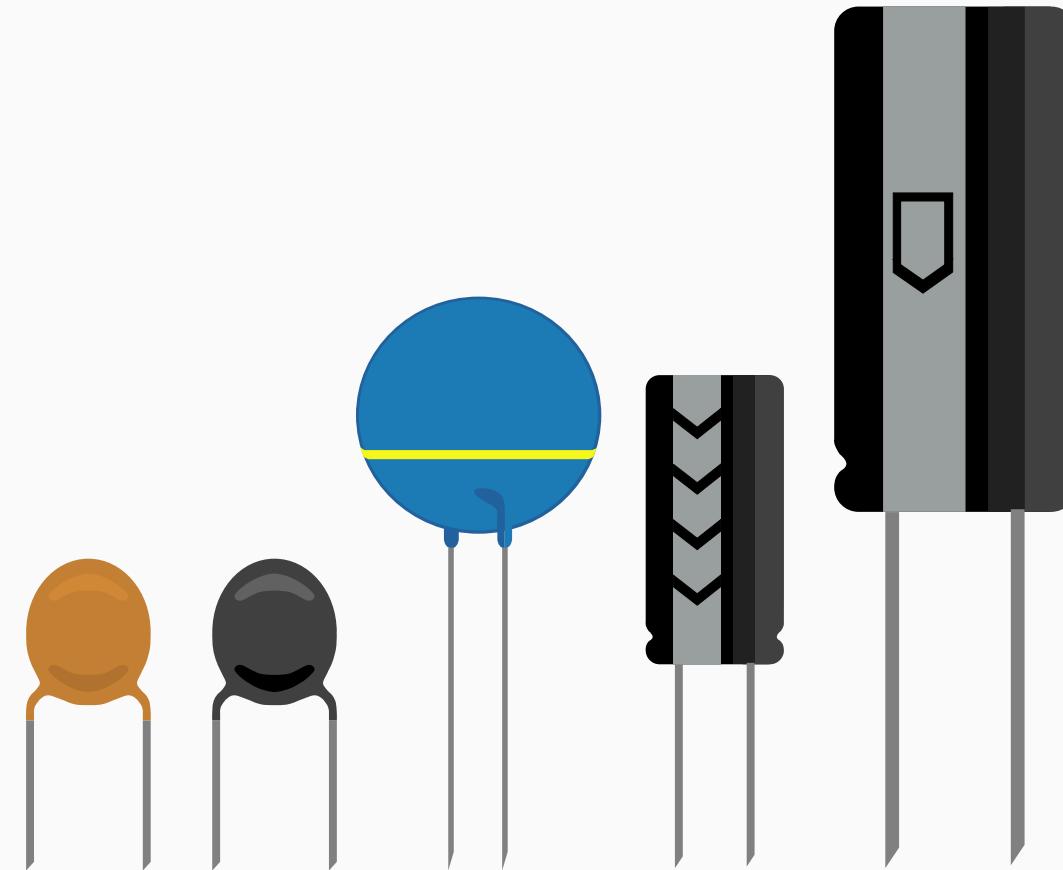


Economizar

Introdução



Economizar

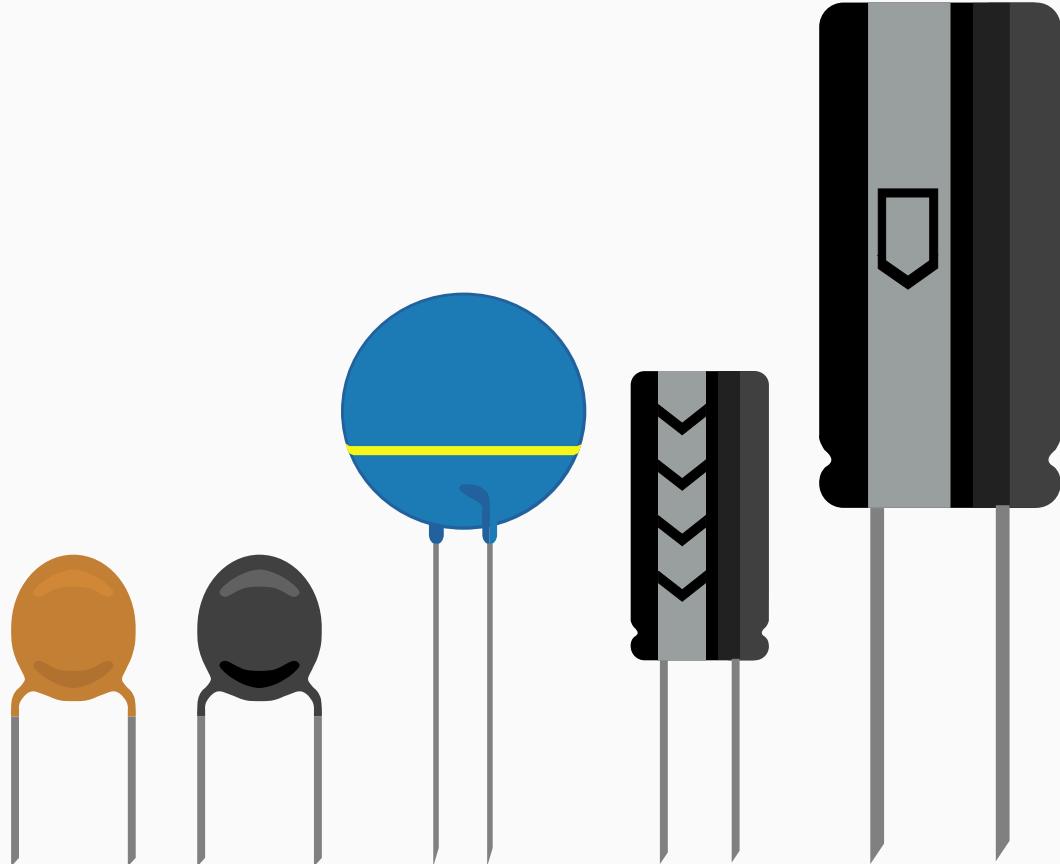


Componentes

Introdução



Economizar



Componentes

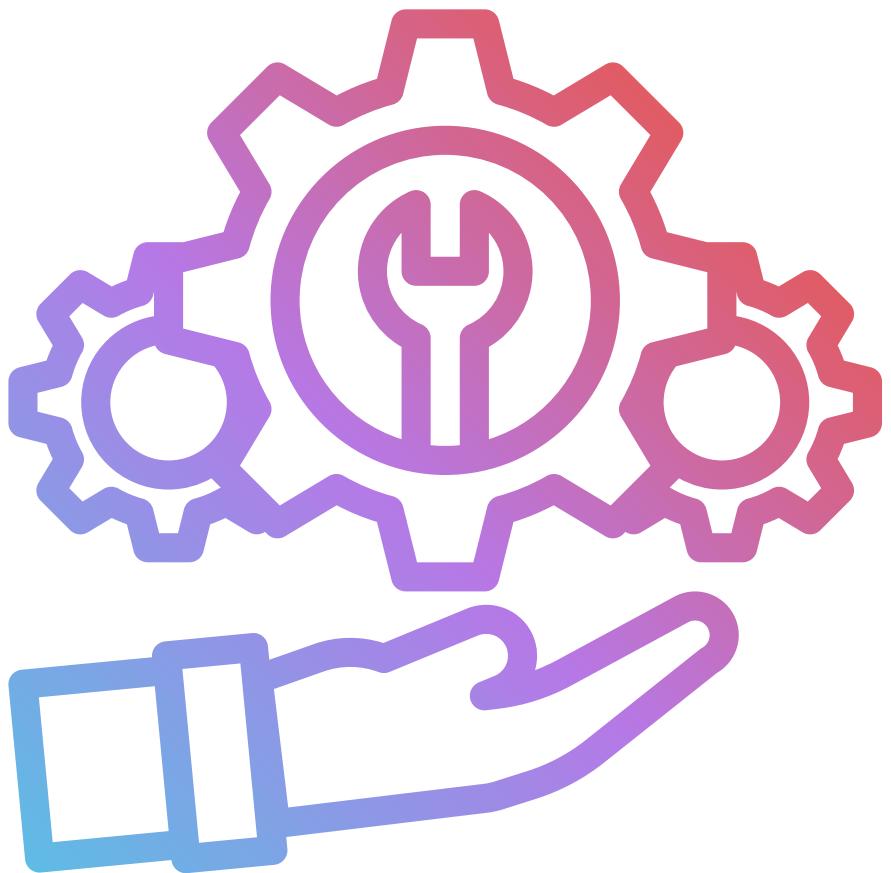


Reutilização

Benefícios

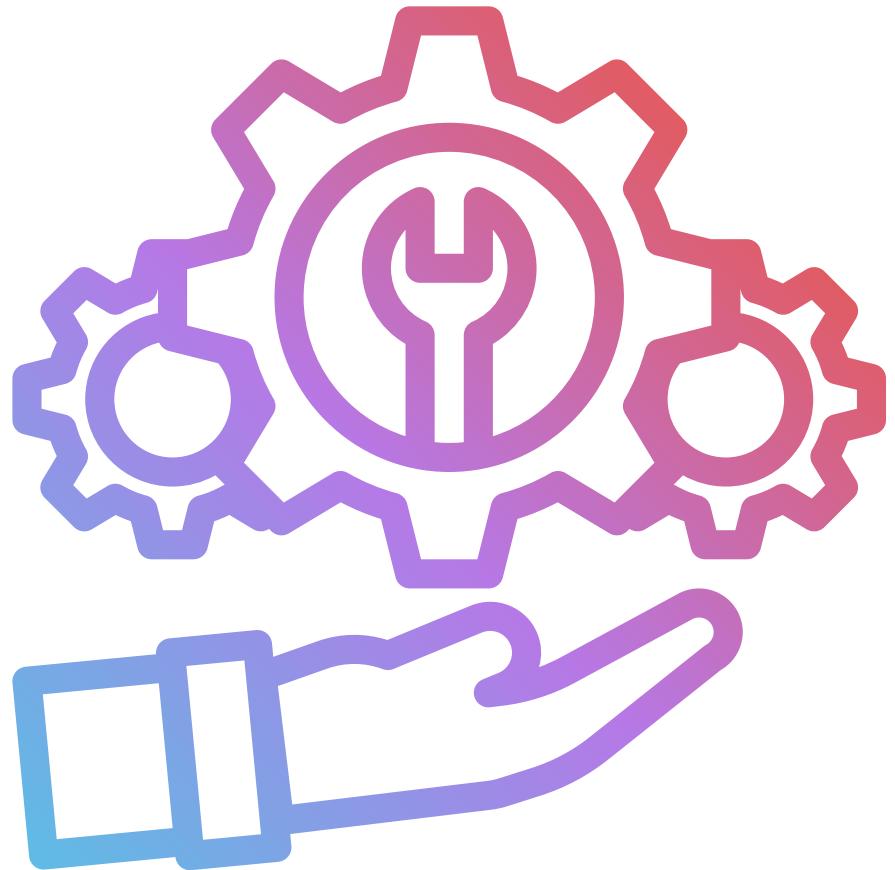
Benefícios

Manutenção e Evolução

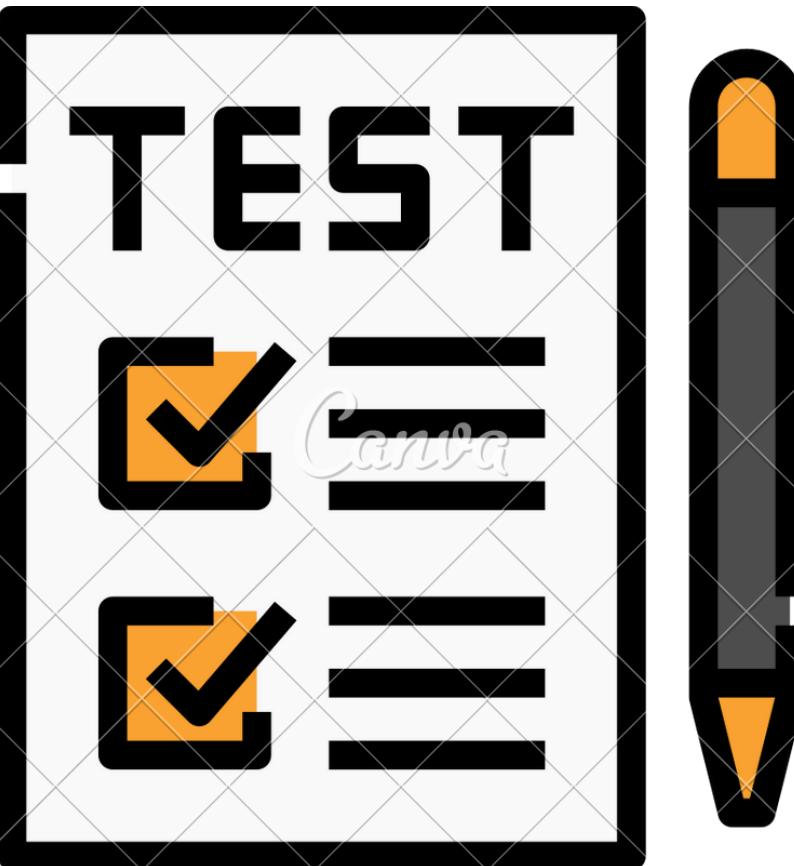


Benefícios

Manutenção e Evolução

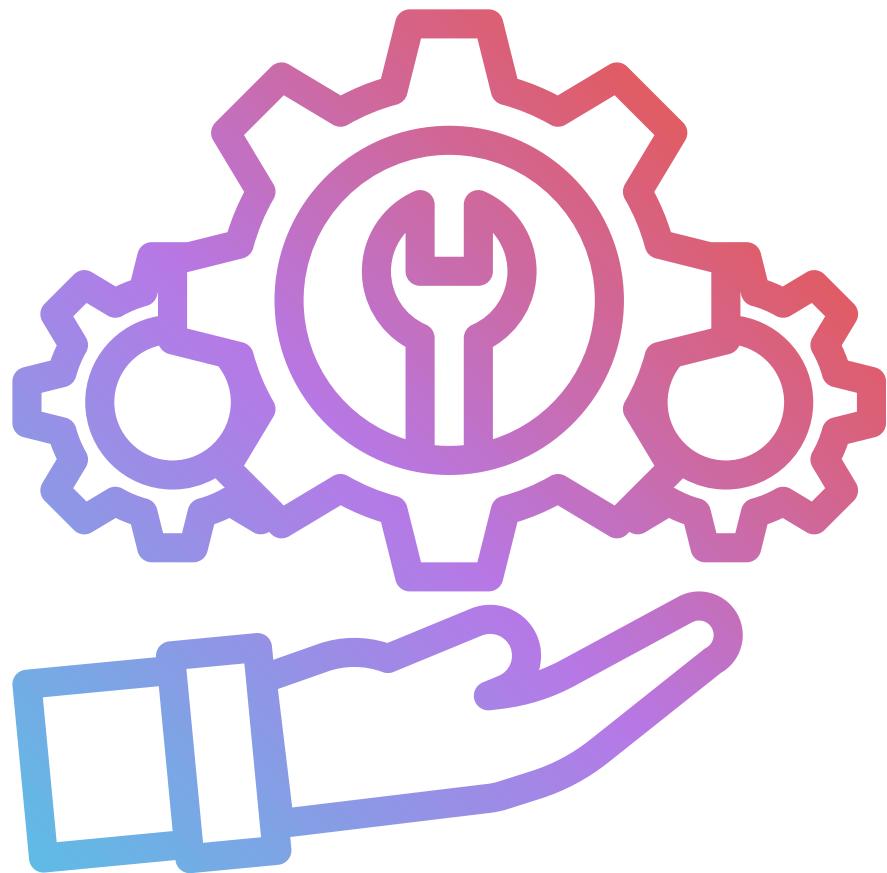


Testes

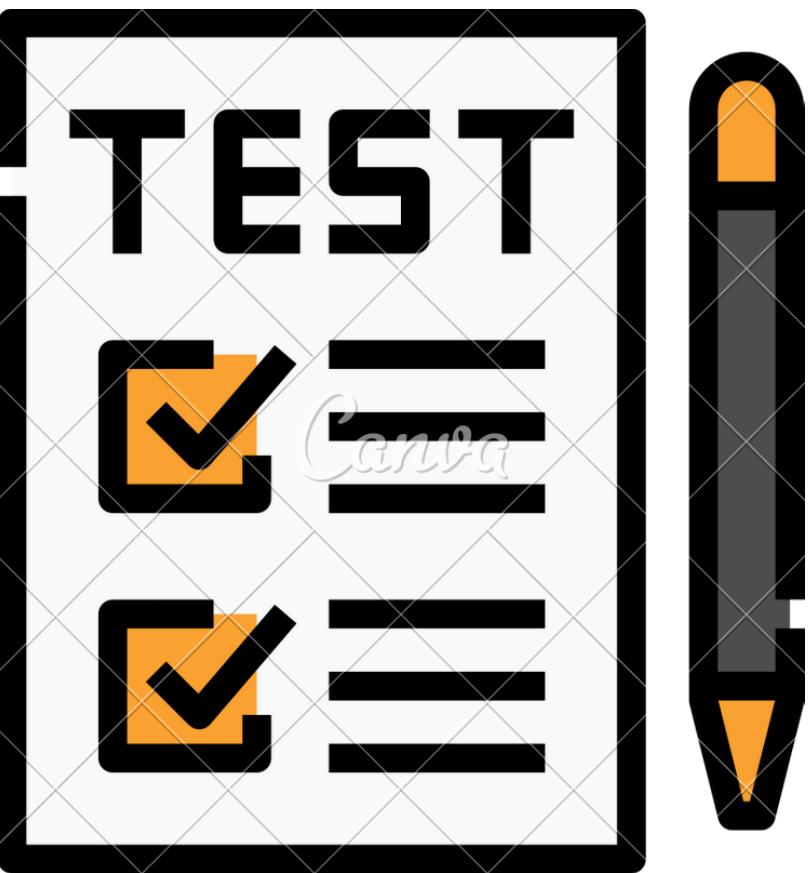


Benefícios

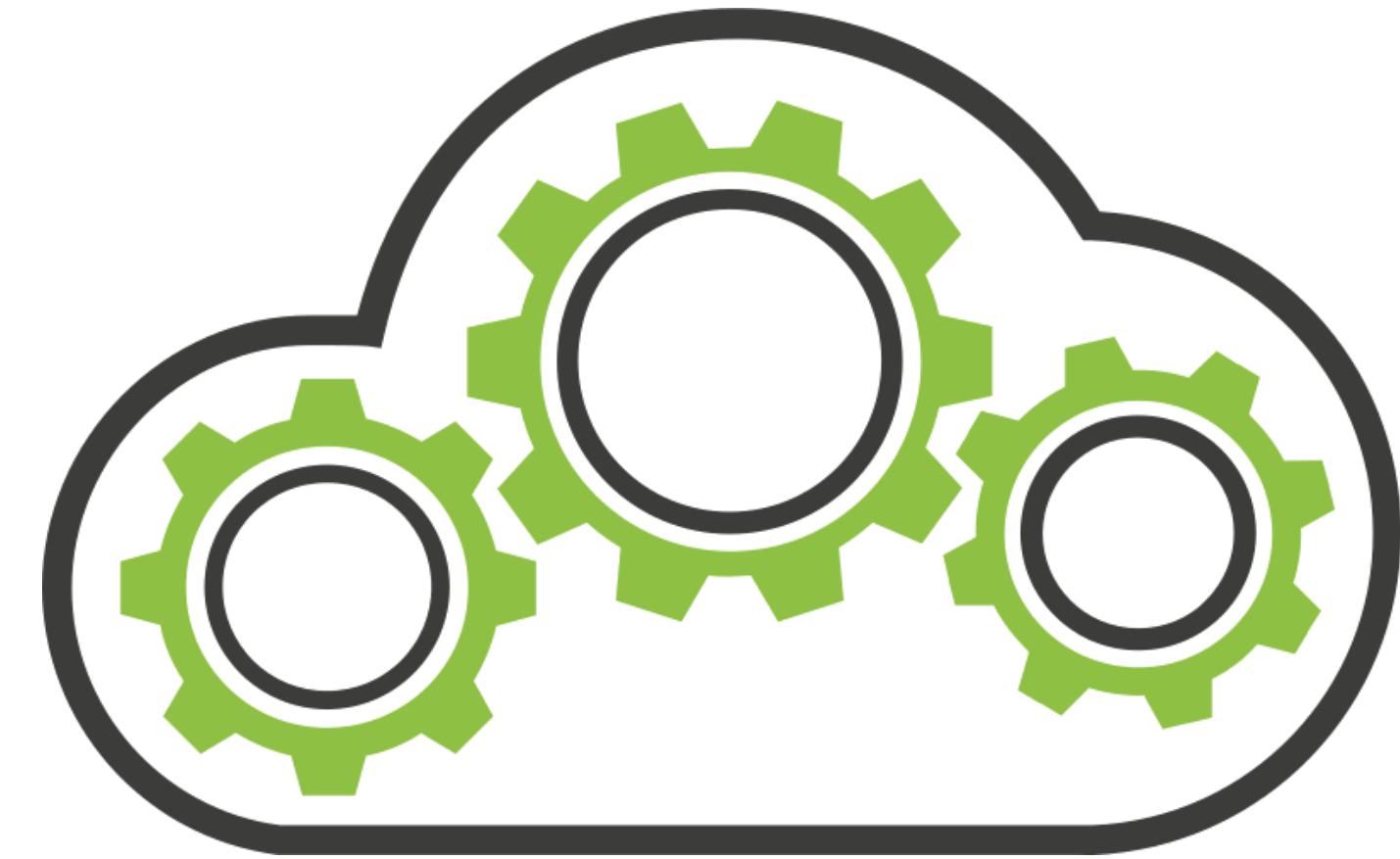
Manutenção e Evolução



Testes



Reuso de software



Reuso de Software

"Reuso de software é uma abordagem que procura reusar partes do processo de construção de um software que já foi realizado, não somente o código, mas todas as abordagens utilizadas no processo de criação de um software."

Organização do reuso de software

Reuso externo

Software é reusado para construção de outros sistemas;

Reuso interno

Software é reusado para construção dele mesmo como uma nova versão evolutiva

Definições

Linhas de produtos de software (LPS)

“SOFTWARE PRODUCT LINE ENGINEERING IS A PARADIGM TO DEVELOP SOFTWARE APPLICATIONS (SOFTWARE-INTENSIVE SYSTEMS AND SOFTWARE PRODUCTS) USING PLATFORMS AND MASS CUSTOMISATION”

[POHL ET. AL, 2005].

“A SOFTWARE PRODUCT LINE IS A SET OF SOFTWARE-INTENSIVE SYSTEMS THAT SHARE A COMMON, MANAGED SET OF FEATURES SATISFYING THE SPECIFIC NEEDS OF A PARTICULAR MARKET SEGMENT OR MISSION AND THAT ARE DEVELOPED FROM A COMMON SET OF CORE ASSETS IN A PRESCRIBED WAY”

[CLEMENTS, NORTHROP, 2002].

Definições

Engenharia de Domínio

Processo responsável por estabelecer a plataforma de reutilização definindo o que é comum e o que é variável da linha de produtos.

A plataforma consiste:

- Requisitos
- Testes
- Design

Engenharia de Aplicação

Processo responsável por derivar aplicações concretas a partir da plataforma estabelecida na engenharia de domínio.

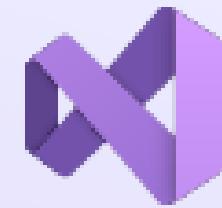
Ela explora a variabilidade da linha de produtos e assegura sua correta instanciação de acordo com as necessidades específicas das aplicações finais.

Exemplos de linhas de produtos de software



Visual Studio IDE

- *Versão Community*
- *Versão Professional*
- *Versão Enterprise*



Visual Studio |

O IDE mais abrangente para desenvolvedores .NET e C++ no Windows. Totalmente empacotado com uma bela matriz de ferramentas e recursos para elevar e aprimorar cada estágio de desenvolvimento de software.

[Saiba mais →](#)

Baixar o Visual
Studio

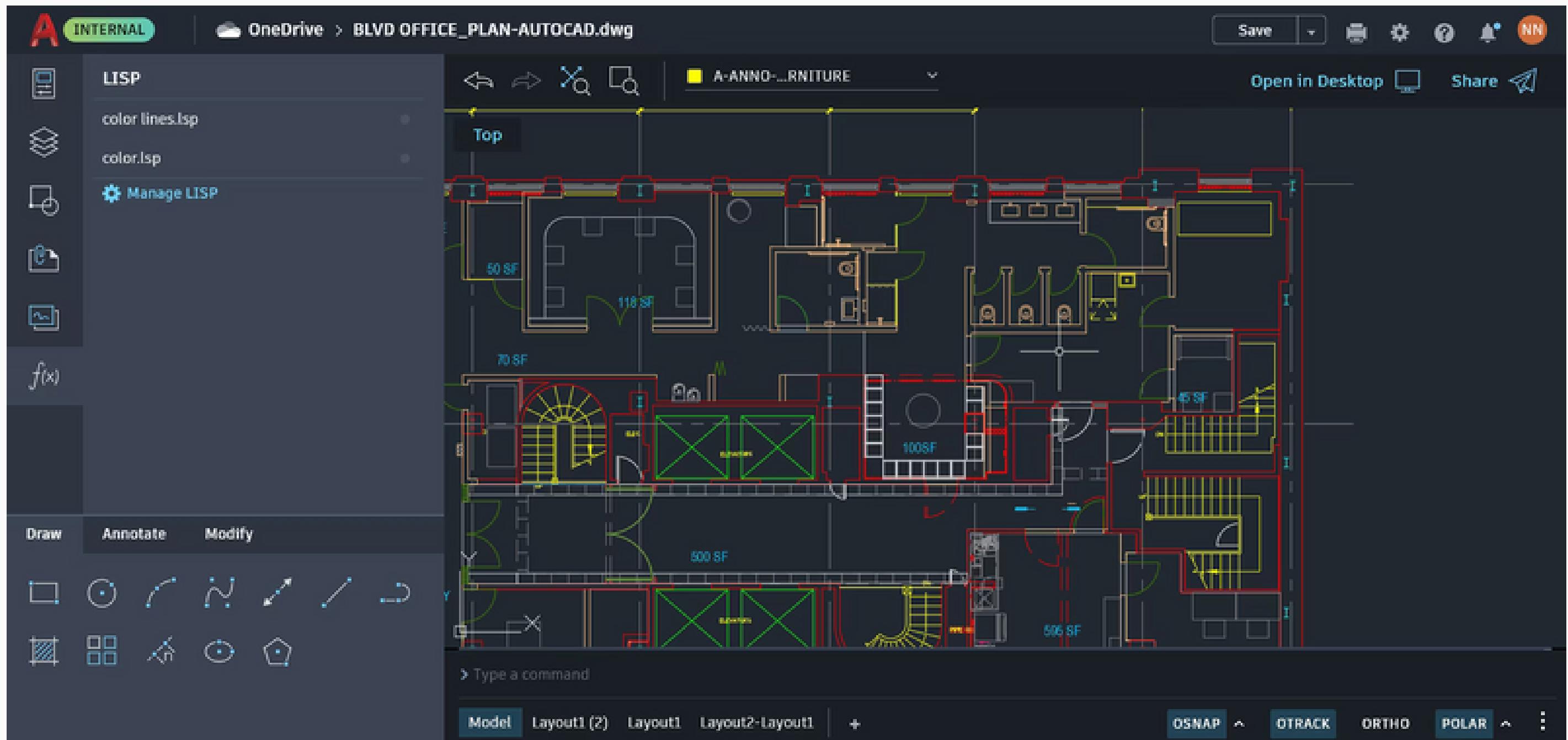
Community 2022

Professional 2022

Enterprise 2022

AUTOCAD

- *Versão Estudante*
- *Versão Profissional*



OS Windows



Microsoft

Em destaque

	Windows 10 Home	Windows 10 Pro
Windows Hello ¹ ▾	✓	✓
Criptografia de dispositivos ² ▾	✓	✓
Firewall e proteção de rede ▾	✓	✓
Proteção da Internet ▾	✓	✓
Controles / proteção dos pais ▾	✓	✓
Inicialização Segura ▾	✓	✓
Criptografia de dispositivo BitLocker ³ ▾		✓
WIP (Proteção de informações do Windows) ^{4 5} ▾		✓
Windows Defender Antivírus ▾	✓	✓

PADRÕES ARQUITETURAIS

PADRÕES ARQUITETURAIS

FEATURE-BASED ARCHITECTURE

Esse padrão arquitetural define uma arquitetura de software que suporta a implementação de diferentes características em uma linha de produto. Cada característica é implementada em um módulo separado, e esses módulos podem ser combinados para produzir diferentes produtos.

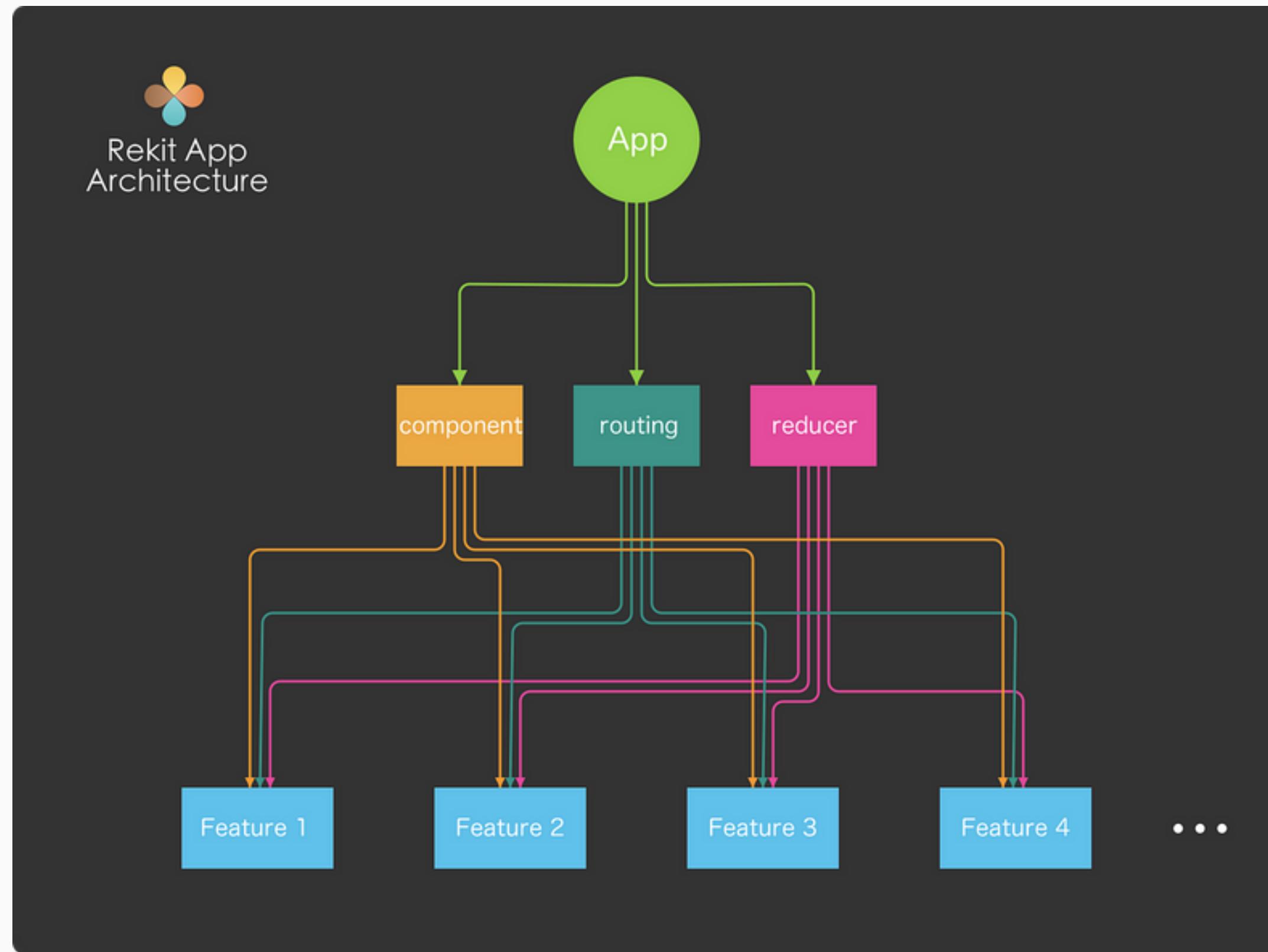
SERVICE-ORIENTED ARCHITECTURE - SOA

Nesse padrão arquitetural, os serviços são a unidade básica de funcionalidade. Dessa forma, cada serviço é independente e pode ser reutilizado em diferentes contextos. A SOA é especialmente útil para linhas de produto que precisam suportar integrações com outros sistemas.

COMPONENT-BASED ARCHITECTURE

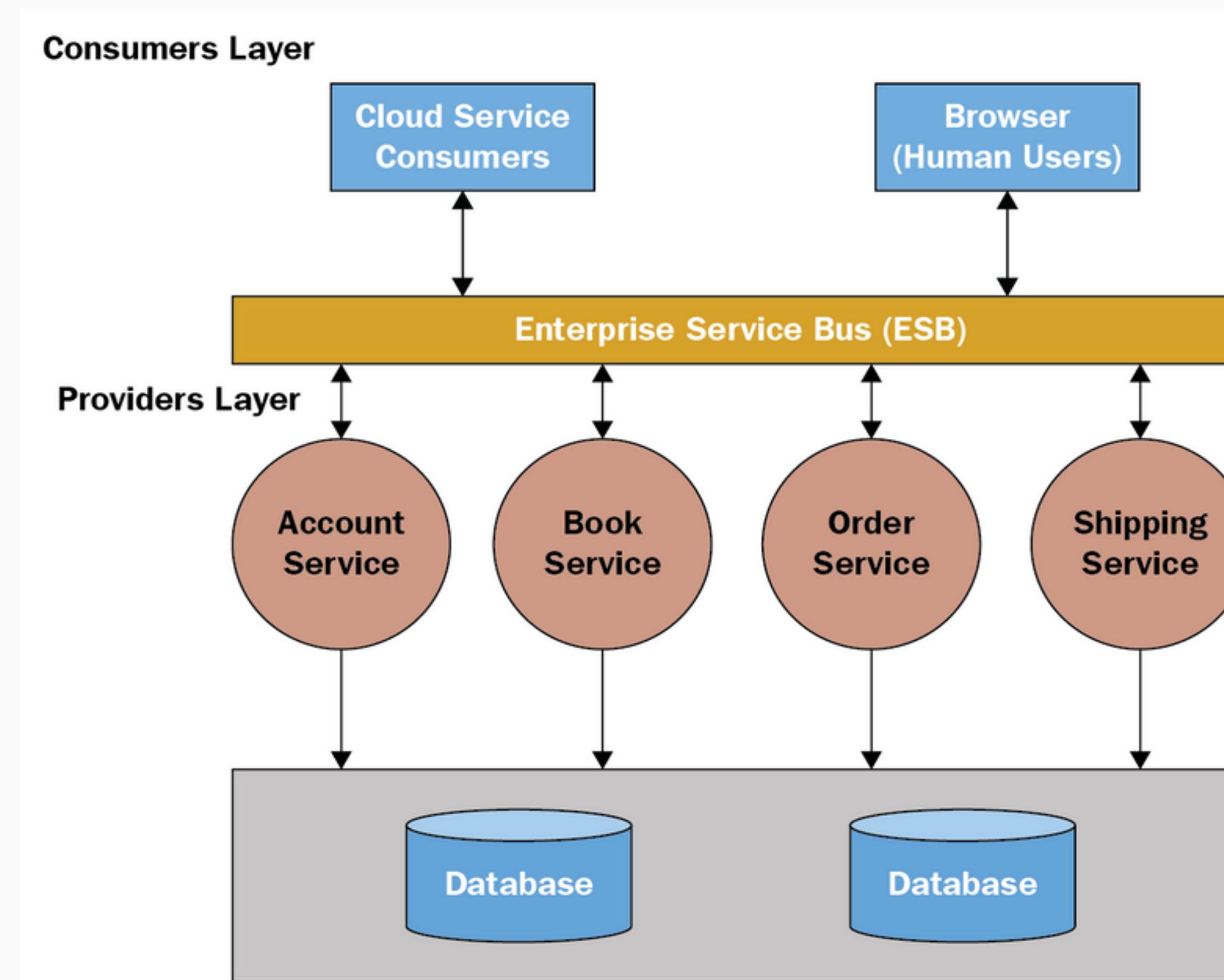
Essa arquitetura baseia-se na reutilização de componentes de software. Cada componente é responsável por uma funcionalidade específica e pode ser reutilizado em diferentes contextos. Essa arquitetura é particularmente útil para linhas de produto que compartilham um conjunto comum de componentes.

Feature-Based Architecture

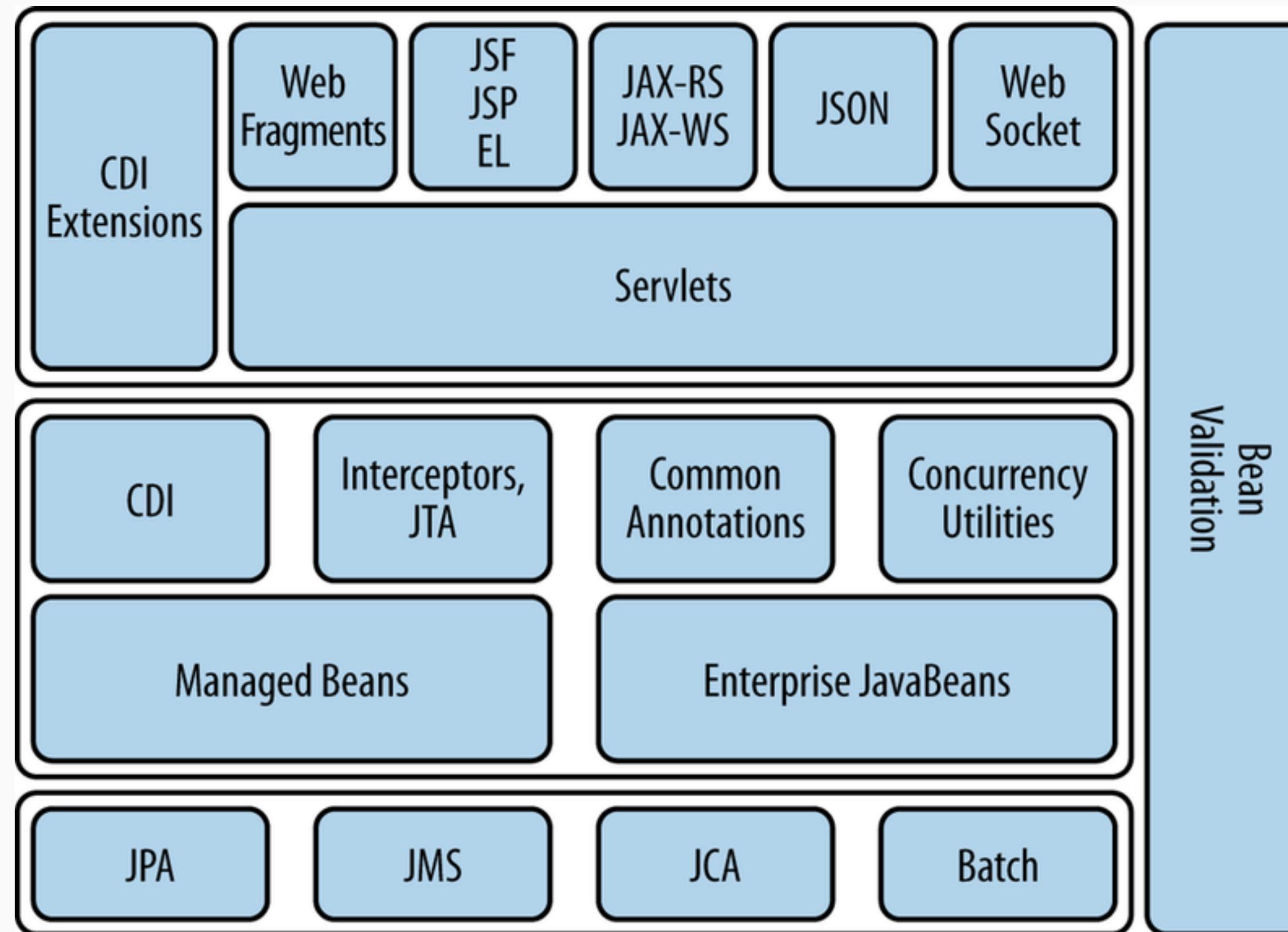


THIS ARCHITECTURE IS ALSO THE KEY CONCEPT OF REKIT, A TOOLKIT FOR BUILDING WEB APPLICATIONS WITH REACT, REDUX AND REACT ROUTER

Service-Oriented Architecture - SOA



Component-Based Architecture



ARQUITETURAS DE REFERÊNCIAS

FEATURE-ORIENTED
DOMAIN ANALYSIS
(FODA)

GENERATIVE
SOFTWARE
DEVELOPMENT (GSD)

ARCHITECTURE-
CENTRIC DESIGN
METHOD (ACDM)

ARQUITETURAS DE REFERÊNCIAS

FEATURE-ORIENTED DOMAIN ANALYSIS (FODA)

- Feature-Oriented Domain Analysis (FODA) é um método utilizado no desenvolvimento de SPLs que visa identificar as características comuns e as variações em um conjunto de produtos relacionados.
- O objetivo da análise de domínio orientada a recursos é fornecer uma base para o desenvolvimento de produtos de software que atendam às necessidades específicas de diferentes clientes ou segmentos de mercado.

ARQUITETURAS DE REFERÊNCIAS

FEATURE-ORIENTED DOMAIN ANALYSIS (FODA)

- Um exemplo é a família de sistemas operacionais Microsoft Windows. A Microsoft desenvolveu o Windows como uma linha de produtos de software, com diferentes versões do sistema operacional direcionadas a diferentes mercados e usuários.
- O FODA foi utilizado no desenvolvimento do Windows para analisar as necessidades dos usuários e definir as características que cada versão do sistema operacional deveria ter.
- Ao usar o FODA, a Microsoft foi capaz de criar uma linha de produtos de software eficiente e personalizada, que atendesse às necessidades específicas de cada mercado e usuário.

ARQUITETURAS DE REFERÊNCIAS

ARCHITECTURE-CENTRIC DESIGN METHOD (ACDM)

- Embora o Architecture-Centric Design Method (ACDM) não seja necessariamente uma arquitetura de referência para linhas de produto de software, ele pode ser utilizado como uma abordagem para projetar uma arquitetura comum em uma linha de produtos de software.
- O Architecture-Centric Design Method é uma abordagem estruturada e sistemática para projetar sistemas de software baseados em uma arquitetura sólida e coerente, que é implementada e refinada iterativamente.
- Ele enfatiza a modularidade, a flexibilidade e a reutilização de componentes para criar sistemas escaláveis e sustentáveis.

ARQUITETURAS DE REFERÊNCIAS

ARCHITECTURE-CENTRIC DESIGN METHOD (ACDM)

- Um exemplo de software de linha de produtos de software que utiliza o Architecture-Centric Design Method (ACDM) é o Rational Software Architect (RSA) da IBM. O RSA é uma ferramenta de modelagem e design de software que permite aos usuários criar arquiteturas de software para diferentes linhas de produtos.
- O RSA é um exemplo de como o Architecture-Centric Design Method pode ser utilizado em uma ferramenta de modelagem de software para criar arquiteturas consistentes e modulares que podem ser facilmente reutilizadas em diferentes linhas de produtos.

ARQUITETURAS DE REFERÊNCIAS

GENERATIVE SOFTWARE DEVELOPMENT (GSD)

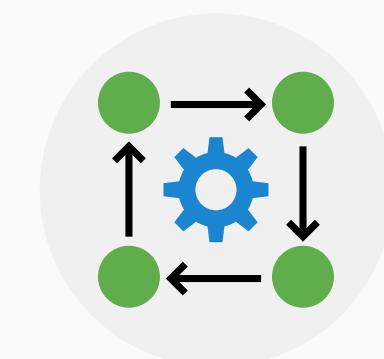
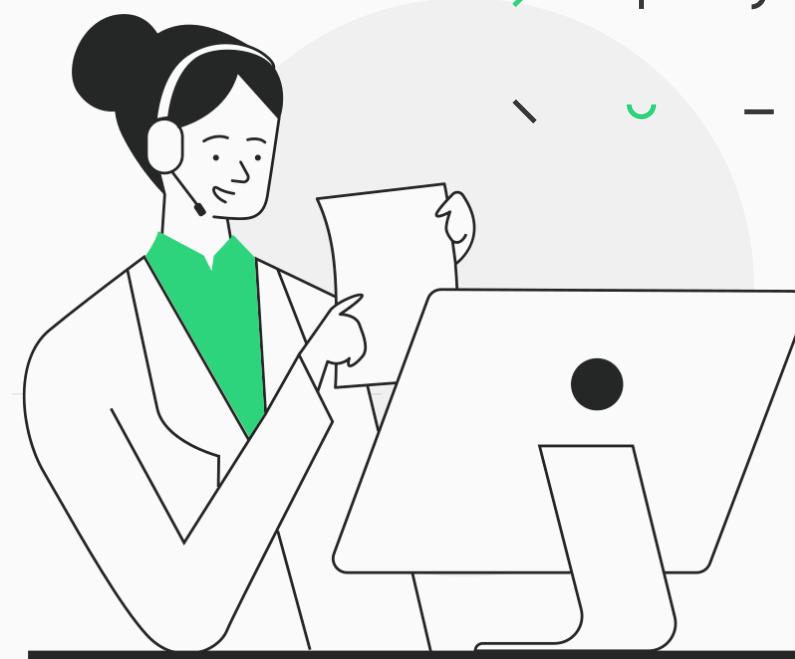
- O Generative Software Development (GSD) é uma abordagem de desenvolvimento de software que utiliza modelos e técnicas de programação gerativa para produzir software automaticamente.
- Ele pode ser usado como uma arquitetura de referência para linhas de produto de software, pois permite que os desenvolvedores criem uma plataforma comum de componentes de software reutilizáveis e adaptem-na para atender a requisitos específicos de produtos dentro da linha.
- Ao criar uma plataforma comum de componentes de software, os desenvolvedores podem implementar novos produtos dentro da linha mais rapidamente e com maior qualidade.

ARQUITETURAS DE REFERÊNCIAS

GENERATIVE SOFTWARE DEVELOPMENT (GSD) - EXEMPLO

- Um exemplo de software real que utiliza Generative Software Development (GSD) é o ThingWorx, uma plataforma de IoT (Internet das Coisas) criada pela PTC. O ThingWorx permite que os usuários criem aplicativos de IoT de maneira mais rápida e eficiente usando a modelagem e a programação gerativa.
- Com o ThingWorx, os desenvolvedores podem criar modelos de componentes de IoT, como sensores, atuadores e interfaces de usuário, e definir regras que geram automaticamente o código-fonte do aplicativo de IoT. Isso permite que os usuários criem aplicativos de IoT personalizados rapidamente e com menos erros, reduzindo o tempo e os custos de desenvolvimento.

Abordagens de Apoio



PROCESSOS



FERRAMENTAS



MÉTODOS

Métodos de Apoio ao Design Arquitetural

FEATURE-ORIENTED
DOMAIN ANALYSIS
(FODA)

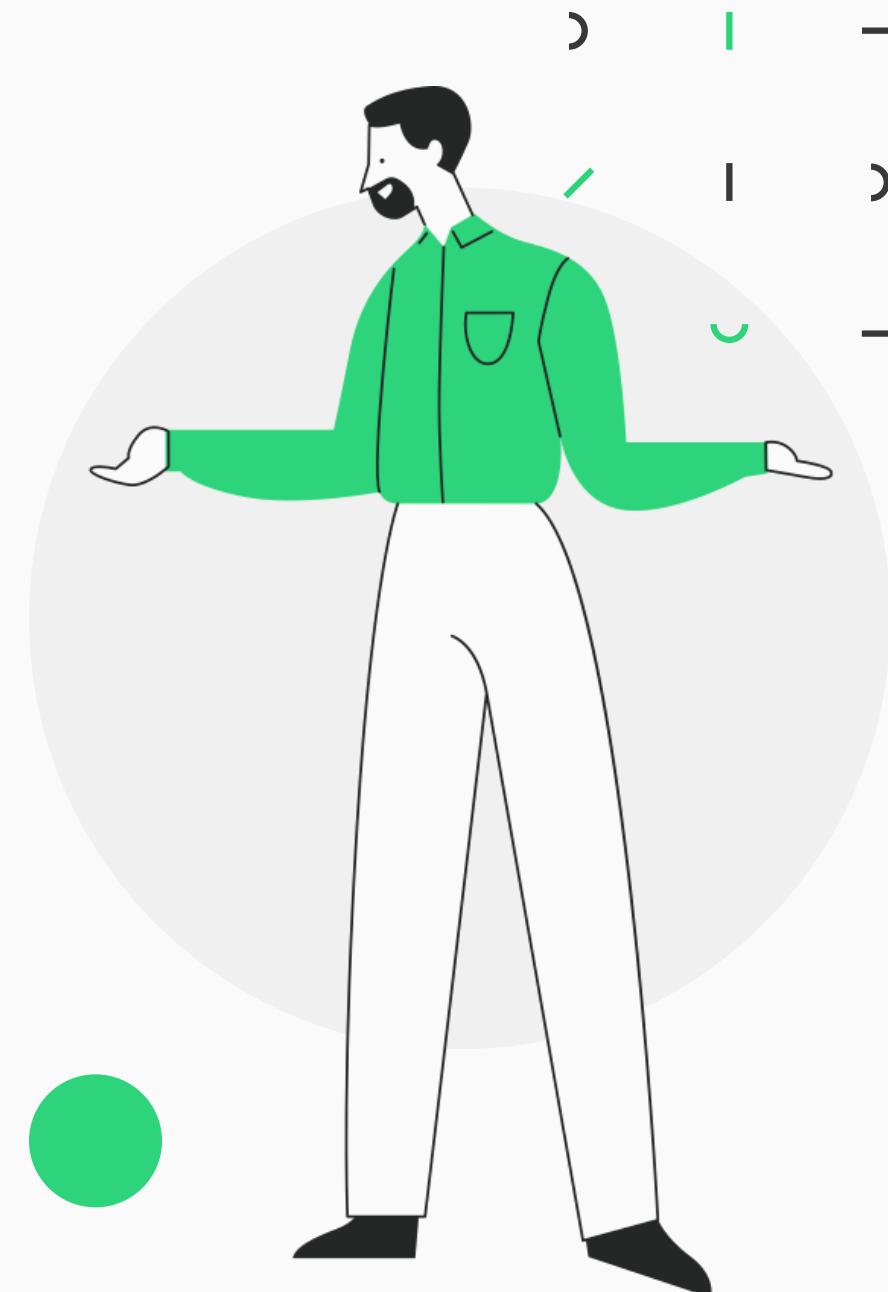
FEATURE ORIENTED
REUSE METHOD
(FORM)

DOMAIN SPECIFIC
SOFTWARE
ARCHITECTURES
(DSSA)

Conceitos

Feature Model

- Um feature model consiste de um ou mais diagramas de features, que organizam os features em hierarquias.
- Usada para capturar o que há de comum ou discriminar entre produtos de uma mesma linha
- Modelagem de features foi proposta como parte do método Feature Oriented Domain Analysis (FODA)



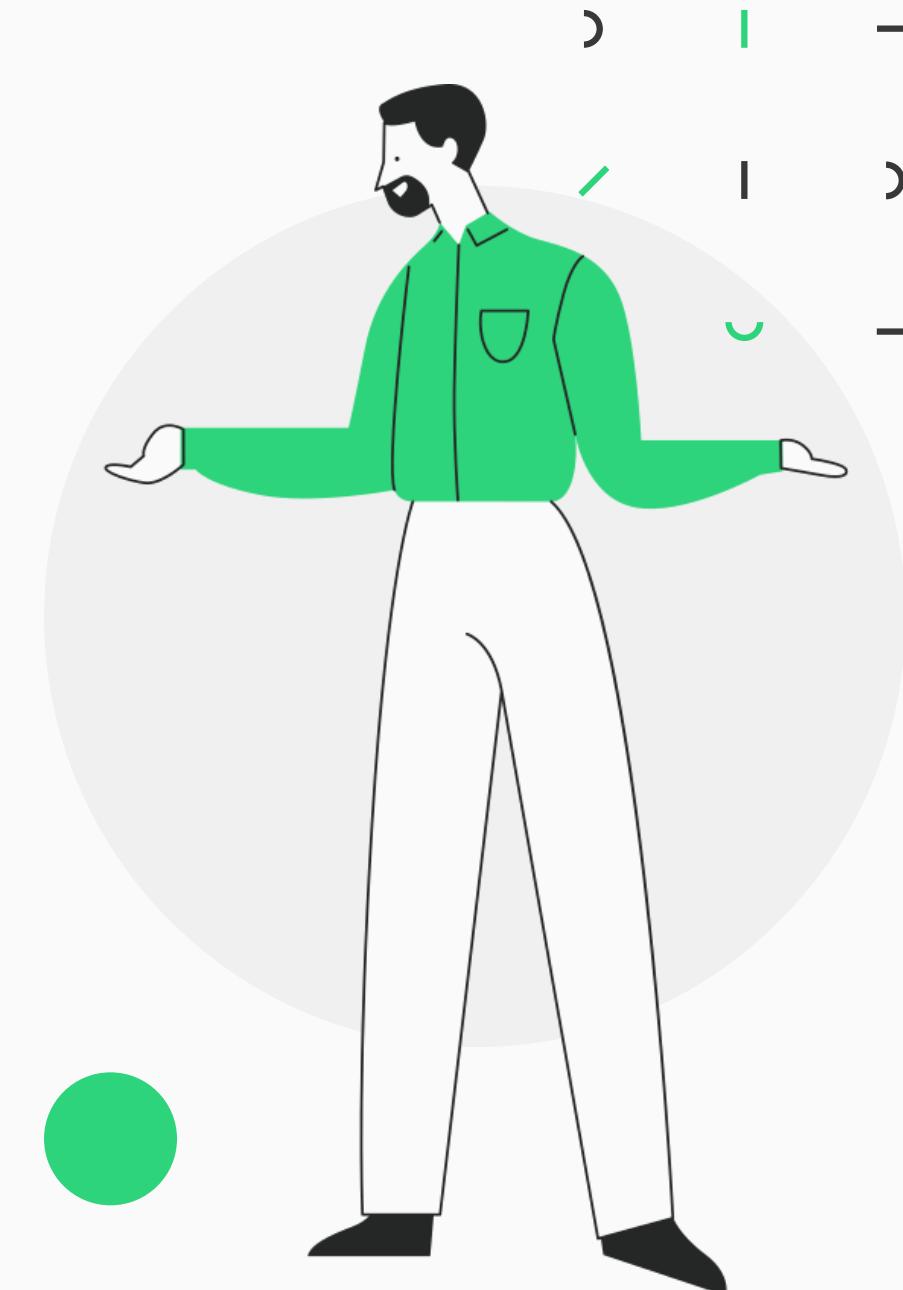
Configuration Knowledge

- O mapeamento entre o feature model e os artefatos de implementação
- Associação dos artefatos de um modelo com as features do feature model
- Conjunto de regras que definem que artefatos de implementação (classes, arquivos de recursos, etc) entram em cada produto da linha

Conceitos

Feature Analysis

- Identificação de todas as características relevantes do domínio de aplicação, bem como sua modelagem em termos de requisitos, restrições e dependências.
- Permite o desenvolvimento de uma arquitetura flexível e reutilizável para uma linha de produto de software.



Modelagem de Domínio

- Processo de análise e especificação das características e requisitos comuns que definem um conjunto de produtos relacionados.
- Geralmente realizada por meio de técnicas de modelagem conceitual, como diagramas de classes, diagramas de entidade-relacionamento, entre outras.

INFORMAÇÕES

"O FODA utiliza-se das seguintes fontes de informação para prover um modelo de domínio: livros texto, padrões, aplicações existentes e especialistas de domínio [KANG et al. 1990]."

"O processo de análise de domínio proposto em FODA é dividido em três fases descritas em Cordet (2007)"

INFORMAÇÕES

"O FODA utiliza-se das seguintes fontes de informação para prover um modelo de domínio: livros texto, padrões, aplicações existentes e especialistas de domínio [KANG et al. 1990]."

"O processo de análise de domínio proposto em FODA é dividido em três fases descritas em Cordet (2007)"

Fases

ANÁLISE DE CONTEXTO:

- ESTABELECER AS FRONTEIRAS DO DOMÍNIO E RELAÇÃO COM OUTROS DOMÍNIOS --> ESCOPO DA ANÁLISE
- RESULTADO FINAL --> DIAGRAMAS (ESTRUTURADO E FLUXO DE DADOS)

MODELAGEM DE DOMÍNIO:

- MODELO DE CONTEXTO --> MODELOS DE DOMÍNIO

MODELAGEM DA ARQUITETURA:

- MODELOS DE DOMÍNIO --> ARQUITETURA DE ALTO NÍVEL

INFORMAÇÕES

"O FODA utiliza-se das seguintes fontes de informação para prover um modelo de domínio: livros texto, padrões, aplicações existentes e especialistas de domínio [KANG et al. 1990]."

"O processo de análise de domínio proposto em FODA é dividido em três fases descritas em Cordet (2007)"

PROBLEMA

Não aborda o desenvolvimento baseado em componentes.

"Essa metodologia também não possui nenhum processo definido para especificação, validação e gerenciamento de requisitos [CORDET 2007]."

Fases

ANÁLISE DE CONTEXTO:

- ESTABELECER AS FRONTEIRAS DO DOMÍNIO E RELAÇÃO COM OUTROS DOMÍNIOS --> ESCOPO DA ANÁLISE
- RESULTADO FINAL --> DIAGRAMAS (ESTRUTURADO E FLUXO DE DADOS)

MODELAGEM DE DOMÍNIO:

- MODELO DE CONTEXTO --> MODELOS DE DOMÍNIO

MODELAGEM DA ARQUITETURA:

- MODELOS DE DOMÍNIO --> ARQUITETURA DE ALTO NÍVEL

Phase	Inputs	Activities	Products
Context Analysis	Operating environments, Standards	Context analysis	Context model
Domain Modeling	Features, Context model	Features analysis	Features model
	Application domain knowledge	Information modeling	Information model
	Domain technology, Context model, Features model, Information model, Requirements	Functional analysis	Functional model Behavioral model
Architectural Modeling	Implementation technology, Context model, Features model, Information model, Design information	Architectural modeling	Structured executive Subsystems model(s)

TABELA 1. SUMÁRIO DO MÉTODO FODA

FEATURE ORIENTED REUSE METHOD (FORM)

INFORMAÇÕES

Método sistemático que extende o FODA para propor as atividades de design e de implementação.

"O FORM é dividido em três fases, descritas em Cordet (2007)"

INFORMAÇÕES

Método sistemático que extende o FODA para propor as atividades de design e de implementação.

"O FORM é dividido em três fases, descritas em Cordet (2007)"

FEATURE ORIENTED REUSE METHOD (FORM)

Fases

ANÁLISE DE CONTEXTO:

- ESTABELECER AS FRONTEIRAS DO DOMÍNIO E RELAÇÃO COM OUTROS DOMÍNIOS --> ESCOPO DA ANÁLISE

MODELAGEM DE CARACTERÍSTICAS:

- ANÁLISE DOS ASPECTOS COMUNS E VARIANTES
- IDENTIFICAÇÃO DOS ASPECTOS OBRIGATÓRIOS E OPCIONAIS DAS APLICAÇÕES DE DOMÍNIO
- DIAGRAMA DE FEATURES

MODELAGEM DA ARQUITETURA:

- MODELOS DE CARACTERÍSTICAS --> CONJUNTO DE ARQUITETURA DE REFERÊNCIAS
 - Subsystem Model: arquitetura geral do sistema
 - Process Model: comportamento dinâmico do sistema
 - Module Model: modelo com componentes reusáveis

FEATURE ORIENTED REUSE METHOD (FORM)

INFORMAÇÕES

Método sistemático que extende o FODA para propor as atividades de design e de implementação.

"O FORM é dividido em três fases, descritas em Cordet (2007)"

PROBLEMA

Inflexibilidade da arquitetura: pode ser difícil de adaptar a novas demandas ou alterações nos requisitos do cliente

Dependência de um número significativo de recursos: a implementação eficaz do FORM requer uma equipe de desenvolvimento com conhecimentos especializados em modelagem de domínio, design arquitetural, desenvolvimento de software e gerenciamento de linha de produto de software

Fases

ANÁLISE DE CONTEXTO:

- ESTABELECER AS FRONTEIRAS DO DOMÍNIO E RELAÇÃO COM OUTROS DOMÍNIOS --> ESCOPO DA ANÁLISE

MODELAGEM DE CARACTERÍSTICAS:

- ANÁLISE DOS ASPECTOS COMUNS E VARIANTES
- IDENTIFICAÇÃO DOS ASPECTOS OBRIGATÓRIOS E OPCIONAIS DAS APLICAÇÕES DE DOMÍNIO
- DIAGRAMA DE FEATURES

MODELAGEM DA ARQUITETURA:

- MODELOS DE CARACTERÍSTICAS --> CONJUNTO DE ARQUITETURA DE REFERÊNCIAS
 - Subsystem Model: arquitetura geral do sistema
 - Process Model: comportamento dinâmico do sistema
 - Module Model: modelo com componentes reusáveis

INFORMAÇÕES

"O DSSA representa uma arquitetura para um domínio específico demonstrando as características comuns e variantes entre os sistemas do domínio, podendo ser geral o suficiente para suportar uma infinidade de aplicações."

"Essa abordagem foca em como prover características comuns e variáveis e como derivar a arquitetura definida para cada aplicação [CORDET 2007]"

"Um documento de requisitos é gerado a partir desse processo [CORDET 2007]."

INFORMAÇÕES

"O DSSA representa uma arquitetura para um domínio específico demonstrando as características comuns e variantes entre os sistemas do domínio, podendo ser geral o suficiente para suportar uma infinidade de aplicações."

"Essa abordagem foca em como prover características comuns e variáveis e como derivar a arquitetura definida para cada aplicação [CORDET 2007]"

"Um documento de requisitos é gerado a partir desse processo [CORDET 2007]."

PROBLEMA

Possibilidade de criar arquiteturas de software excessivamente complexas e inflexíveis, que podem ser difíceis de manter e evoluir

Dificuldade de balancear a flexibilidade e a reutilização com as necessidades específicas de cada aplicação dentro do domínio.

Outros métodos de apoio

PRODUCT LINE
SOFTWARE
ARCHITECTURE
(PLSA)

ARCHITECTURE-CENTRIC DESIGN METHOD (ACDM)

GENERATIVE
SOFTWARE
DEVELOPMENT
(GSD)

Ferramentas de Apoio

Estudo de caso - EShop

O EShop consiste numa família de produtos de lojas eletrônicas, que faz venda de produtos on-line

FORMA DE PAGAMENTO

- CRÉDITO
- DÉBITO
- ORDEM DE COMPRA

DETECÇÃO DE FRAUDES

ENVIO

- PADRÃO OU GRATUITO
- GATEWAY DE ENVIO (FedEx / UPS / USPC)

POLÍTICA DE SENHA

- EXPIRAÇÃO:
 - NÚMERO DE DIAS
 - NUNCA EXPIRAR
- CARACTERES NECESSÁRIOS:
 - LETRAS MAIÚSCULAS
 - NÚMEROS
 - CARACTERES ESPECIAIS,
 - LETRAS MINÚSCULAS

Principais Ferramentas

FEATURE MODEL PLUGIN - FMP

Plugin para o Eclipse desenvolvido pelo Generative Software Development Lab, da University of Waterloo.

XFEATURE

Plugin para o Eclipse desenvolvido pela P&P Software, que é uma empresa que se originou no Institute of Automatic Control of ETH.

PURE::VARIANTS

Plugin para o Eclipse desenvolvido pela pure-systems GmbH uma empresa que se originou no Institute Otto-von-Guericke-Universität Magdeburg e no Fraunhofer Instituts Rechnerarchitektur und Softwaretechnik.

GEARS

Ferramenta completamente stand-alone, desenvolvida pela BigLever Software Inc..

1.

Feature Model Plugin - fmp

1.

Feature Model Plugin - fmp



Background

- acadêmico

1.

Feature Model Plugin - fmp



Background



Custo

- acadêmico
- open source

1.

Feature Model Plugin - fmp



Background

- acadêmico



Custo

- open source



Implementação

- baseado na versão 3.2 do Eclipse

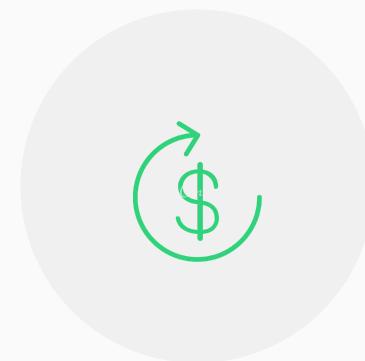
1.

Feature Model Plugin - fmp



Background

- acadêmico



Custo

- open source



Implementação

- baseado na versão 3.2 do Eclipse



Técnica de modelagem e notação utilizada

- modelagem de features baseada em cardinalidades

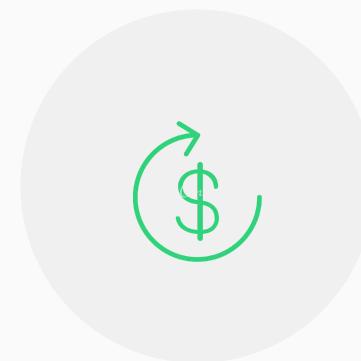
1.

Feature Model Plugin - fmp



Background

- acadêmico



Custo

- open source



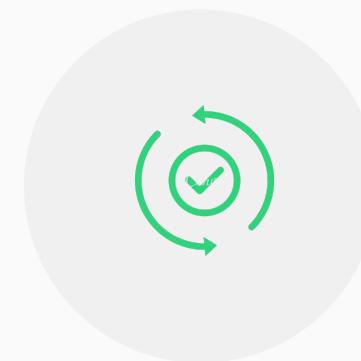
Implementação

- baseado na versão 3.2 do Eclipse



Técnica de modelagem e notação utilizada

- modelagem de features baseada em cardinalidades



Facilidade de uso

- confortável para desenvolvedores familiarizados com o Eclipse
- notação utilizada é simples, porém não intuitiva

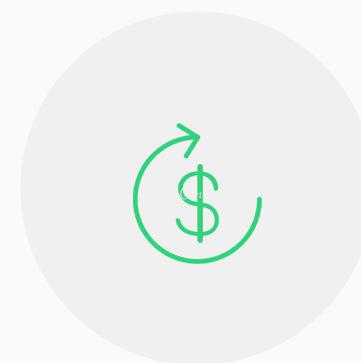
1.

Feature Model Plugin - fmp



Background

- acadêmico



Custo

- open source



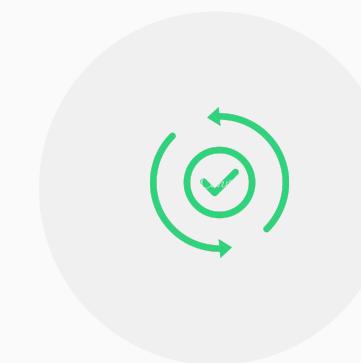
Implementação

- baseado na versão 3.2 do Eclipse



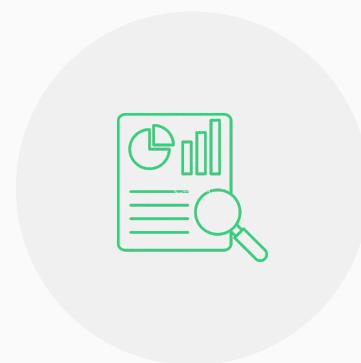
Técnica de modelagem e notação utilizada

- modelagem de features baseada em cardinalidades



Facilidade de uso

- confortável para desenvolvedores familiarizados com o Eclipse
- notação utilizada é simples, porém não intuitiva



Estudo de caso

- originalmente do fmp
- diversos recursos do fmp foram utilizados

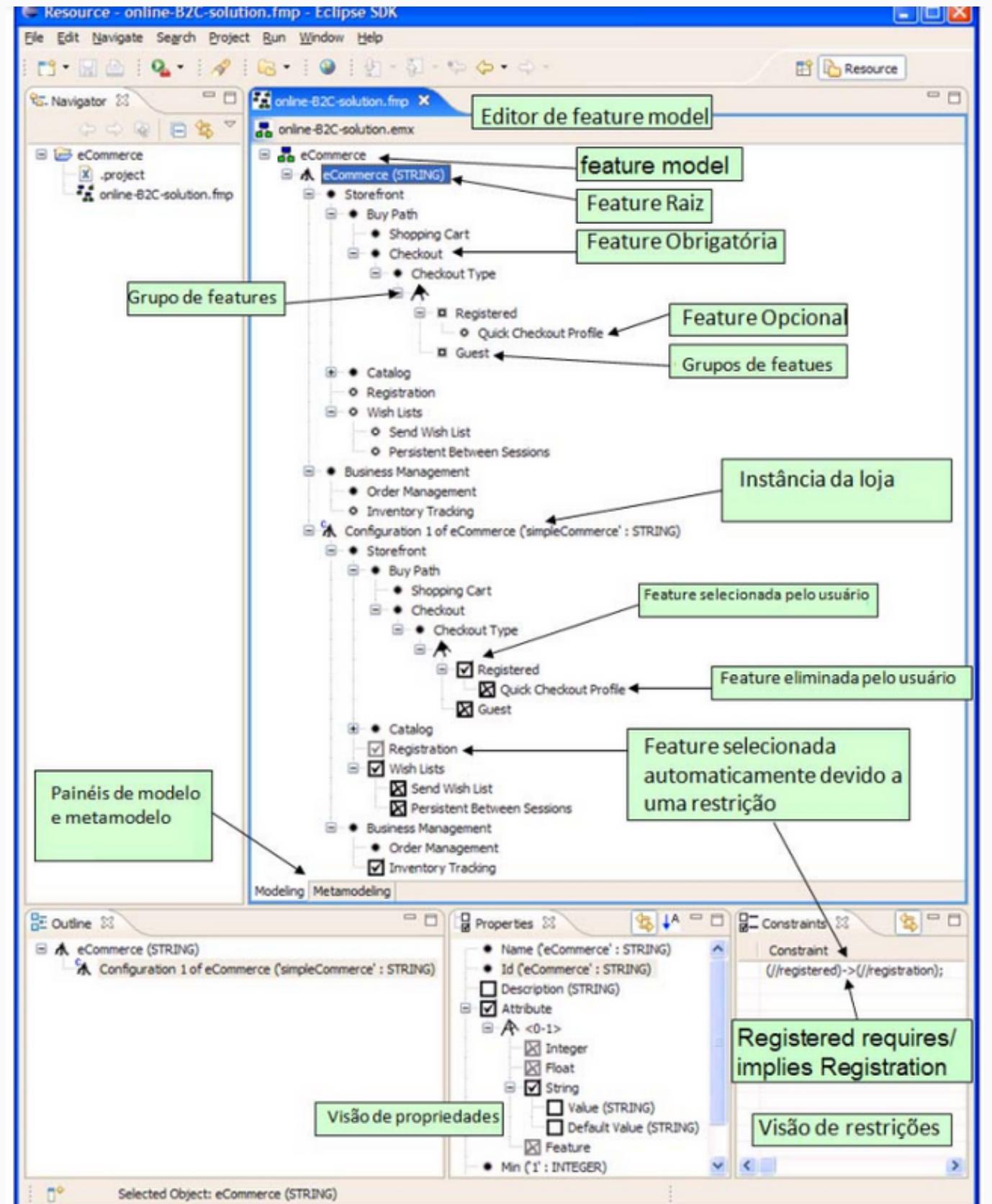


FIGURA 1. INTERFACE DO FMP

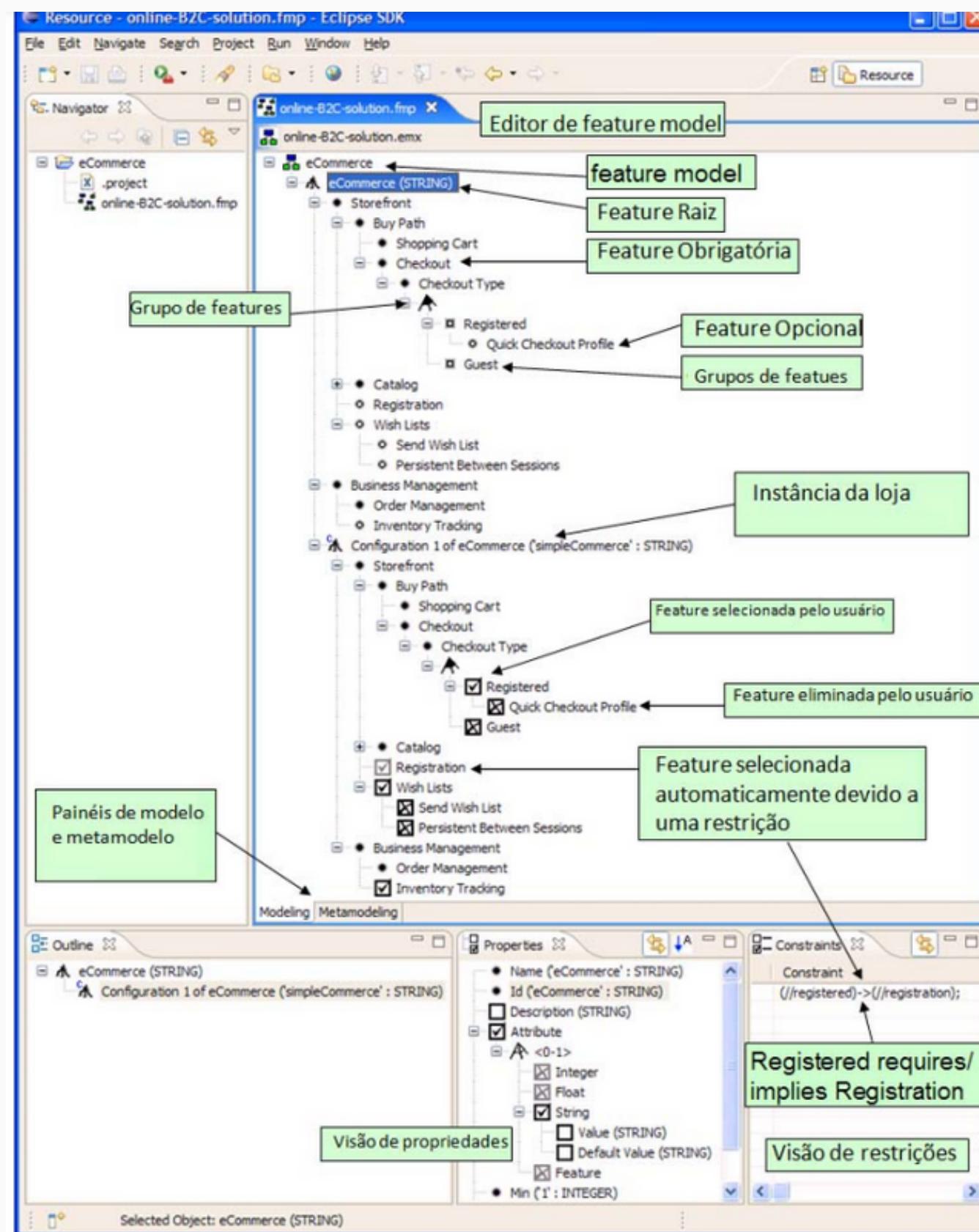


FIGURA 1. INTERFACE DO FMP

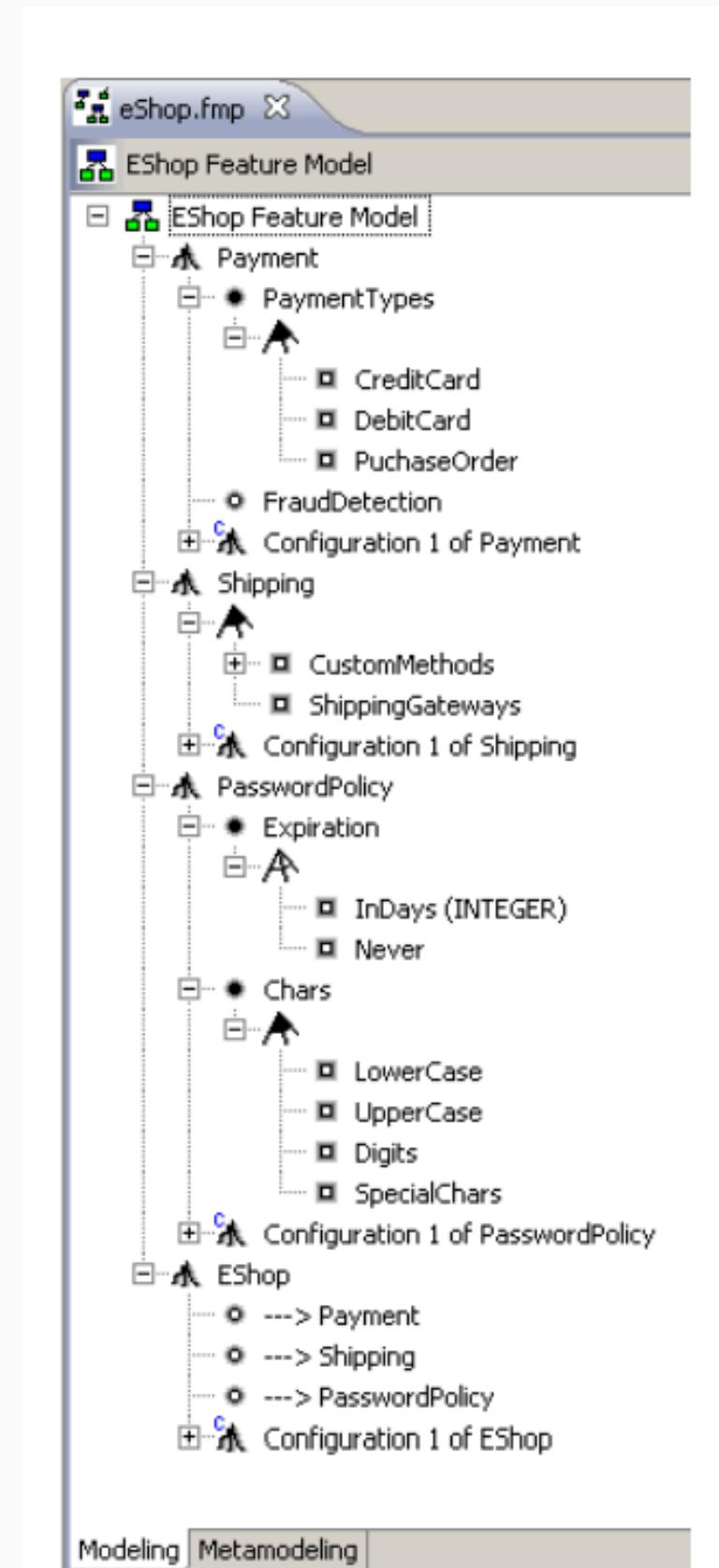


FIGURA 2. ESHOP NO FMP

2. XFeature

2. XFeature



Background

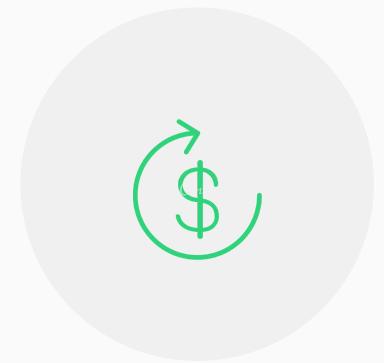
- acadêmico e corporativo
- uso em aplicações espaciais (sistemas de controle embutidos para naves espaciais, por exemplo)

2. XFeature



Background

- acadêmico e corporativo
- uso em aplicações espaciais (sistemas de controle embutidos para naves espaciais, por exemplo)



Custo

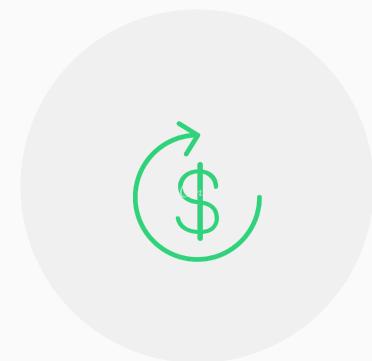
- open source licenciado sob a GNU General Public Licence

2. XFeature



Background

- acadêmico e corporativo
- uso em aplicações espaciais (sistemas de controle embutidos para naves espaciais, por exemplo)



Custo

- open source licenciado sob a GNU General Public Licence



Implementação

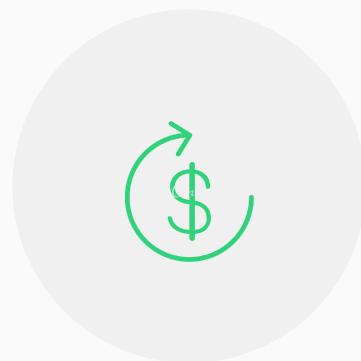
- baseado na versão 3.3 do Eclipse
- editor é baseado no GEF (Graphical Editing Framework) do Eclipse

2. XFeature



Background

- acadêmico e corporativo
- uso em aplicações espaciais (sistemas de controle embutidos para naves espaciais, por exemplo)



Custo

- open source licenciado sob a GNU General Public Licence



Implementação

- baseado na versão 3.3 do Eclipse
- editor é baseado no GEF (Graphical Editing Framework) do Eclipse



Técnica de modelagem e notação utilizada

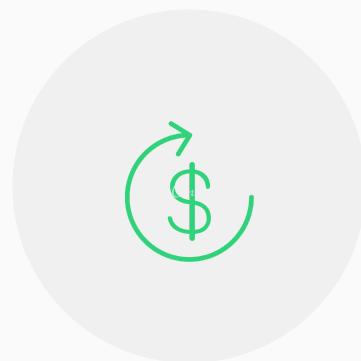
- extremamente flexível
- são fornecidas 4 configurações: FD, FMP, ICSR e SimplifiedICSR

2. XFeature



Background

- acadêmico e corporativo
- uso em aplicações espaciais (sistemas de controle embutidos para naves espaciais, por exemplo)



Custo

- open source licenciado sob a GNU General Public Licence



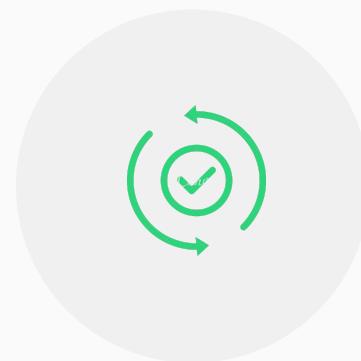
Implementação

- baseado na versão 3.3 do Eclipse
- editor é baseado no GEF (Graphical Editing Framework) do Eclipse



Técnica de modelagem e notação utilizada

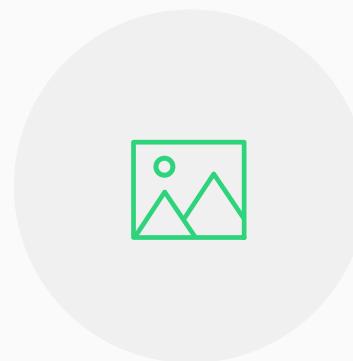
- extremamente flexível
- são fornecidas 4 configurações: FD, FMP, ICSR e SimplifiedICSR



Facilidade de uso

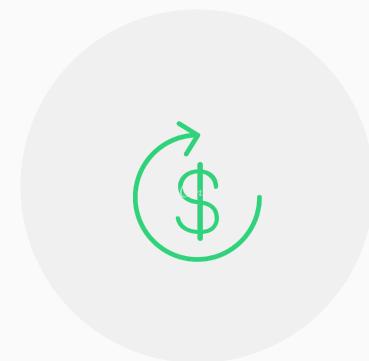
- menos intuitiva
- pior usabilidade

2. XFeature



Background

- acadêmico e corporativo
- uso em aplicações espaciais (sistemas de controle embutidos para naves espaciais, por exemplo)



Custo

- open source licenciado sob a GNU General Public Licence



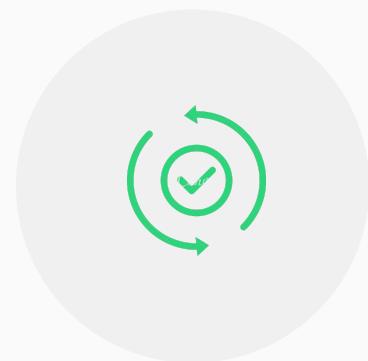
Implementação

- baseado na versão 3.3 do Eclipse
- editor é baseado no GEF (Graphical Editing Framework) do Eclipse



Técnica de modelagem e notação utilizada

- extremamente flexível
- são fornecidas 4 configurações: FD, FMP, ICSR e SimplifiedICSR



Facilidade de uso

- menos intuitiva
- pior usabilidade



Estudo de caso

- configuração FMP
- A cardinalidade é representada visualmente como um range entre os sinais de menor '<' e maior '>'

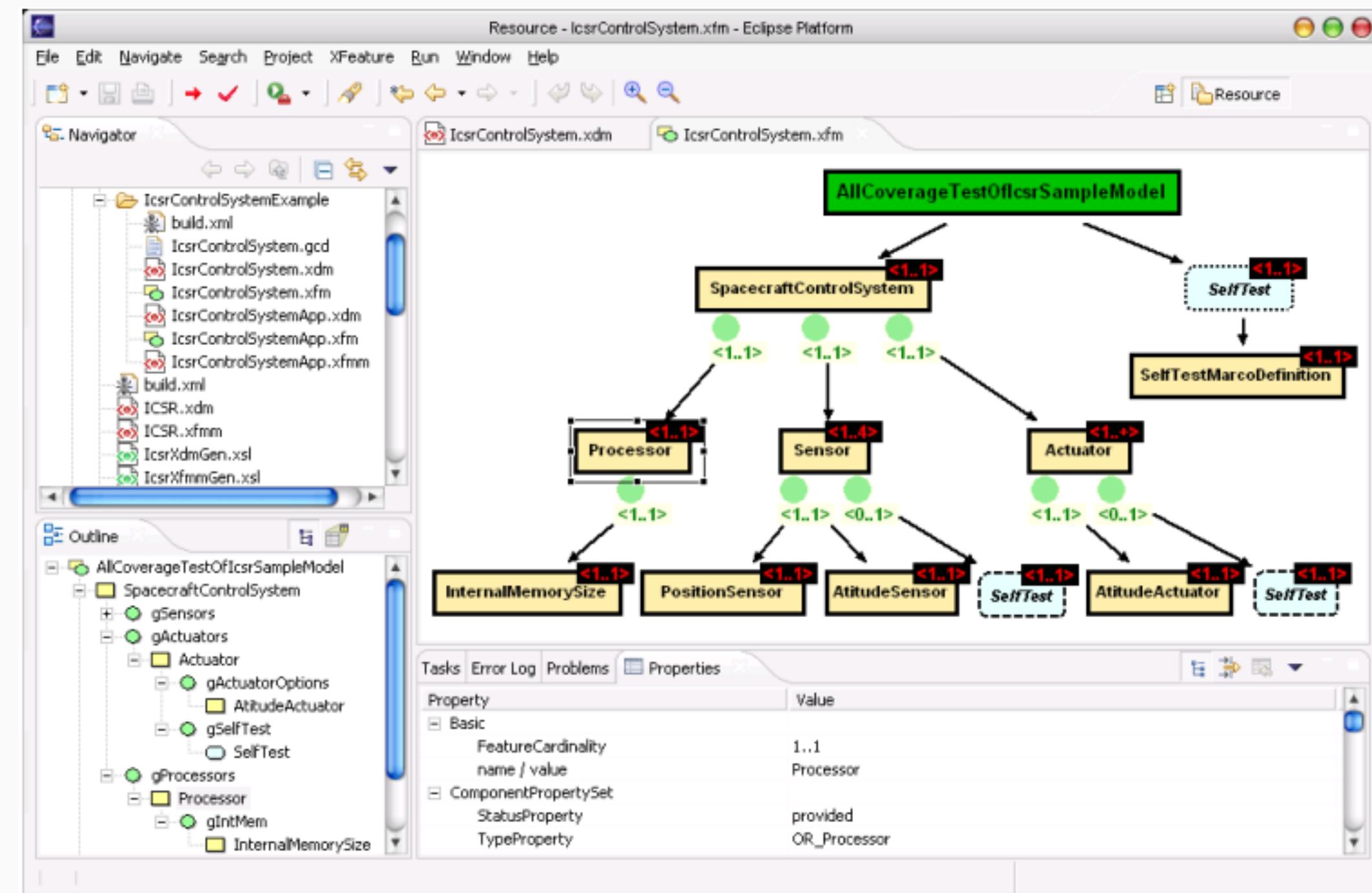


FIGURA 3. INTERFACE DO XFEATURE

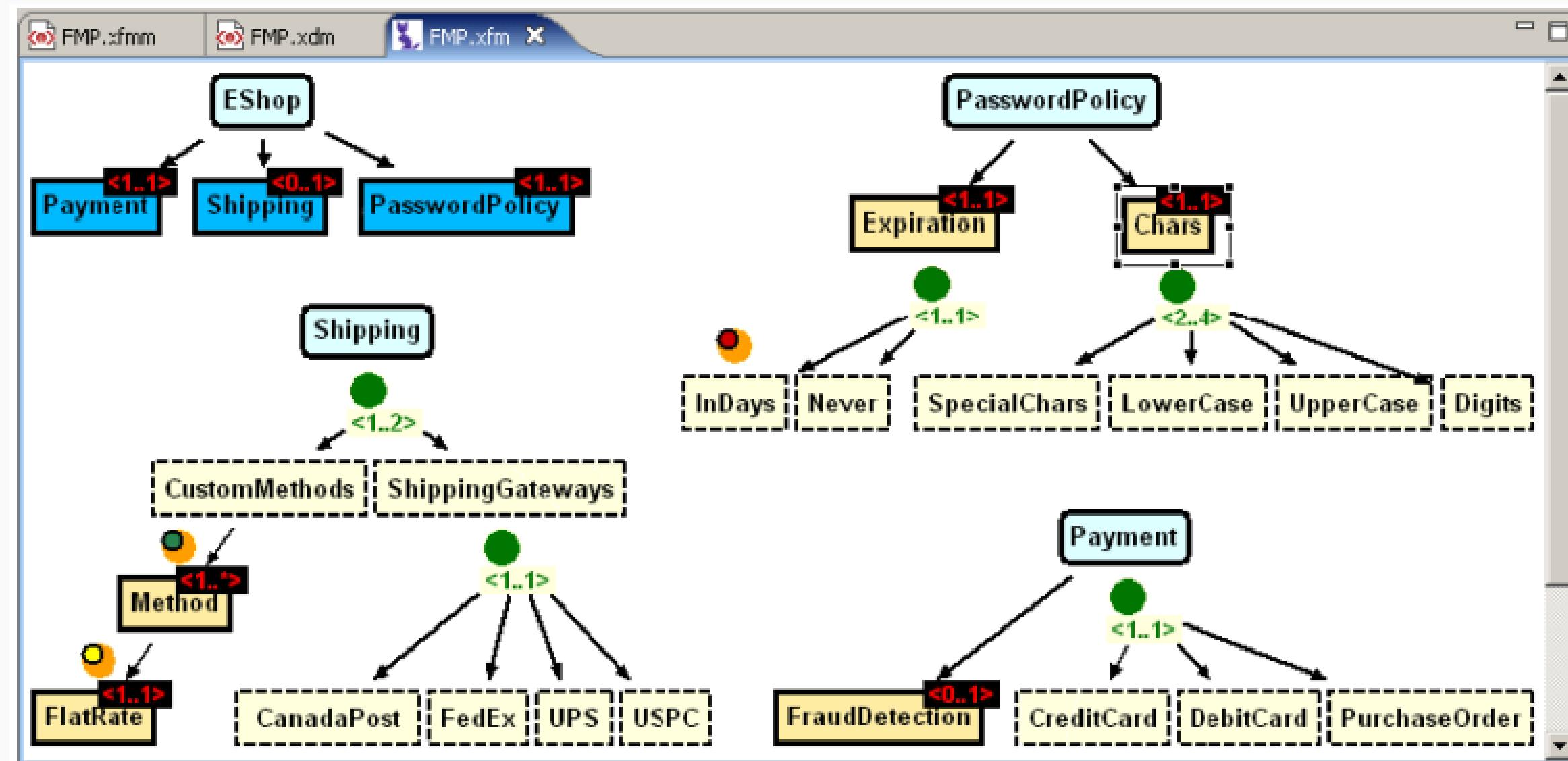


FIGURA 4. FEATURE MODEL DO ESHOP NO XFEATURE

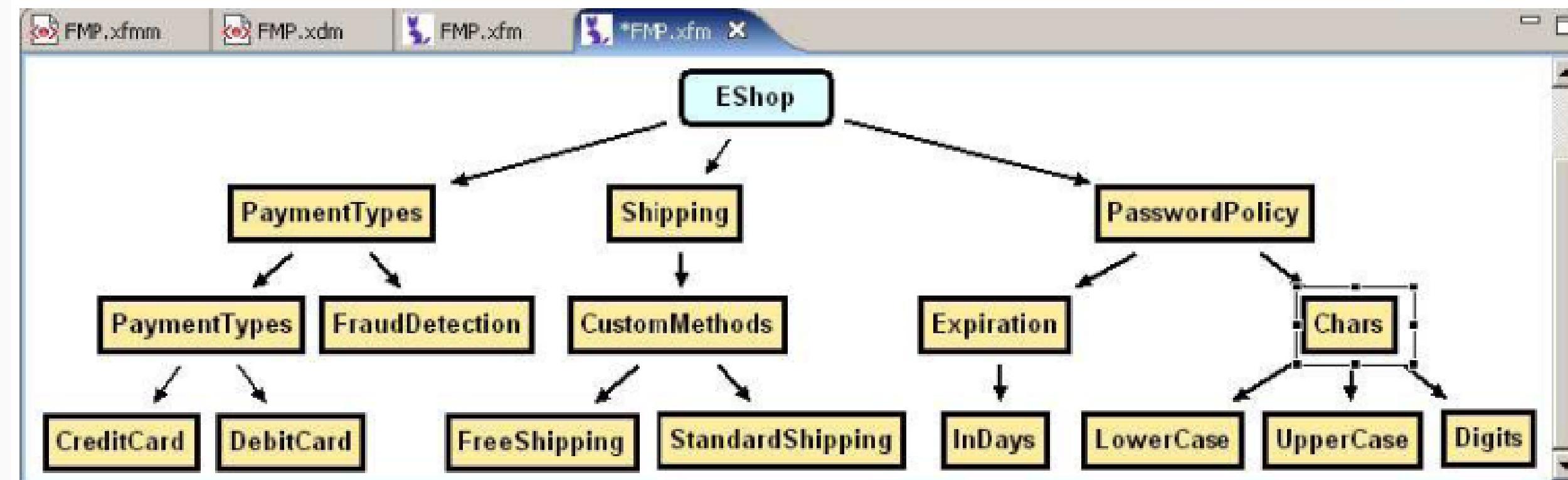


FIGURA 5. INSTÂNCIA DO ESHOP NO XFEATURE

3. pure::variants

3. pure::variants



Background

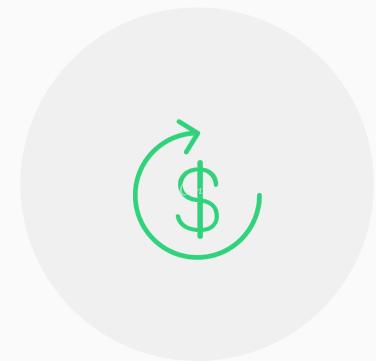
- início acadêmico
- ferramenta comercial rica
- criado visando o mercado de sistemas embarcados

3. pure::variants



Background

- início acadêmico
- ferramenta comercial rica
- criado visando o mercado de sistemas embarcados



Custo

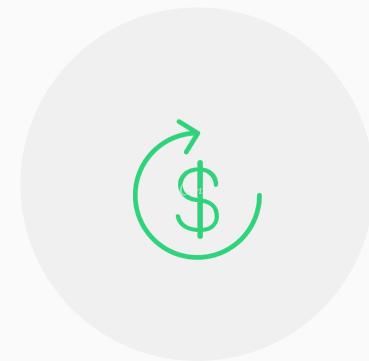
- versão gratuita para testes
- varia dependendo das licenças

3. pure::variants



Background

- início acadêmico
- ferramenta comercial rica
- criado visando o mercado de sistemas embarcados



Custo

- versão gratuita para testes varia dependendo das licenças



Implementação

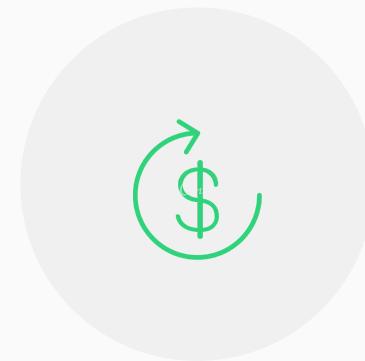
- funciona com a versão 3.3 do Eclipse, mas não se utiliza dos frameworks de modelagem e editores gráficos
- arquitetura cliente-servidor

3. pure::variants



Background

- início acadêmico
- ferramenta comercial rica
- criado visando o mercado de sistemas embarcados



Custo

- versão gratuita para testes varia dependendo das licenças



Implementação

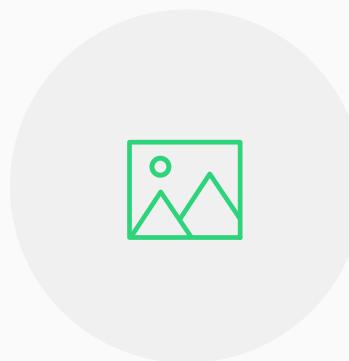
- funciona com a versão 3.3 do Eclipse, mas não se utiliza dos frameworks de modelagem e editores gráficos
- arquitetura cliente-servidor



Técnica de modelagem e notação utilizada

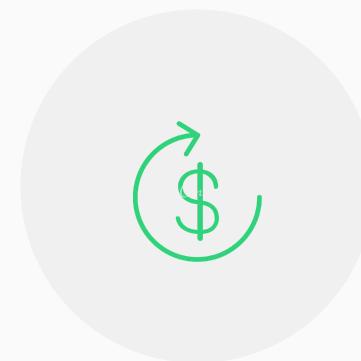
- notação baseada em FODA.
- 3 tipos de modelos diferentes

3. pure::variants



Background

- início acadêmico
- ferramenta comercial rica
- criado visando o mercado de sistemas embarcados



Custo

- versão gratuita para testes varia dependendo das licenças



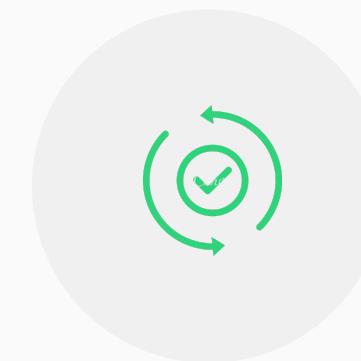
Implementação

- funciona com a versão 3.3 do Eclipse, mas não se utiliza dos frameworks de modelagem e editores gráficos
- arquitetura cliente-servidor



Técnica de modelagem e notação utilizada

- notação baseada em FODA.
- 3 tipos de modelos diferentes



Facilidade de uso

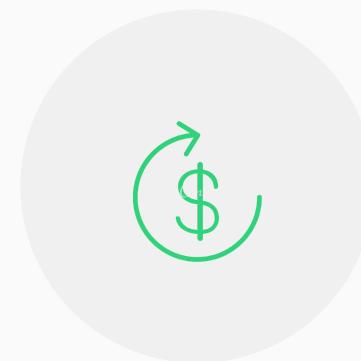
- vários editores , mais conceitos dificultam o uso inicial
- linguagens para as restrições: prolog e pvscl

3. pure::variants



Background

- início acadêmico
- ferramenta comercial rica
- criado visando o mercado de sistemas embarcados



Custo

- versão gratuita para testes
- varia dependendo das licenças



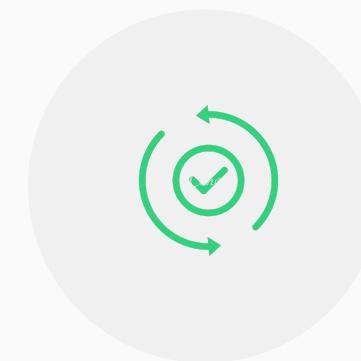
Implementação

- funciona com a versão 3.3 do Eclipse, mas não se utiliza dos frameworks de modelagem e editores gráficos
- arquitetura cliente-servidor



Técnica de modelagem e notação utilizada

- notação baseada em FODA.
- 3 tipos de modelos diferentes



Facilidade de uso

- vários editores, mais conceitos dificultam o uso inicial
- linguagens para as restrições: prolog e pvscl



Estudo de caso

- configuração FMP
- A cardinalidade é representada visualmente como um range entre os sinais de menor '<' e maior '>'

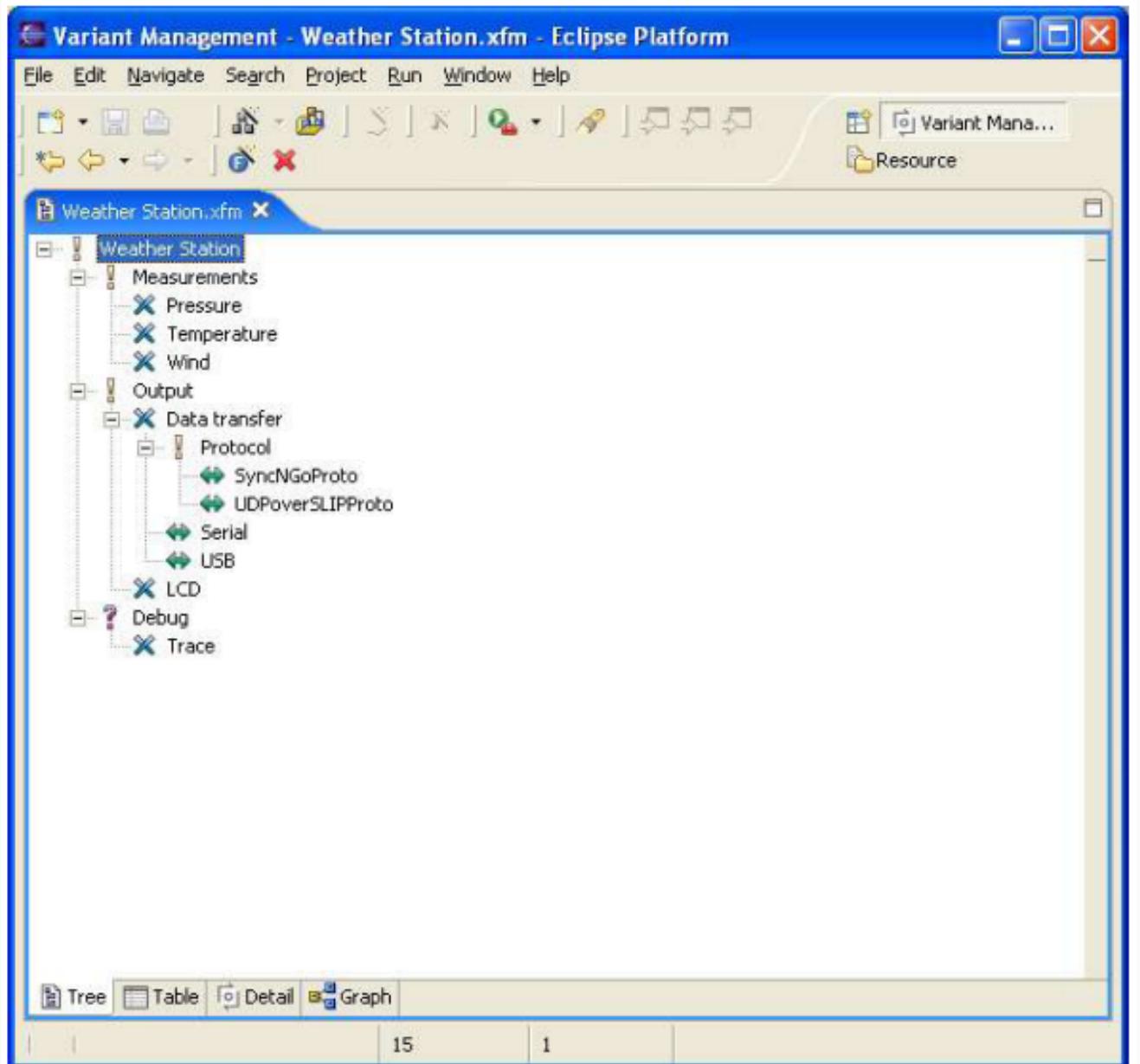


FIGURA 6. INTERFACE DE EDIÇÃO DO FEATURE MODEL EM MODO ÁRVORE DO PURE::VARIANTS

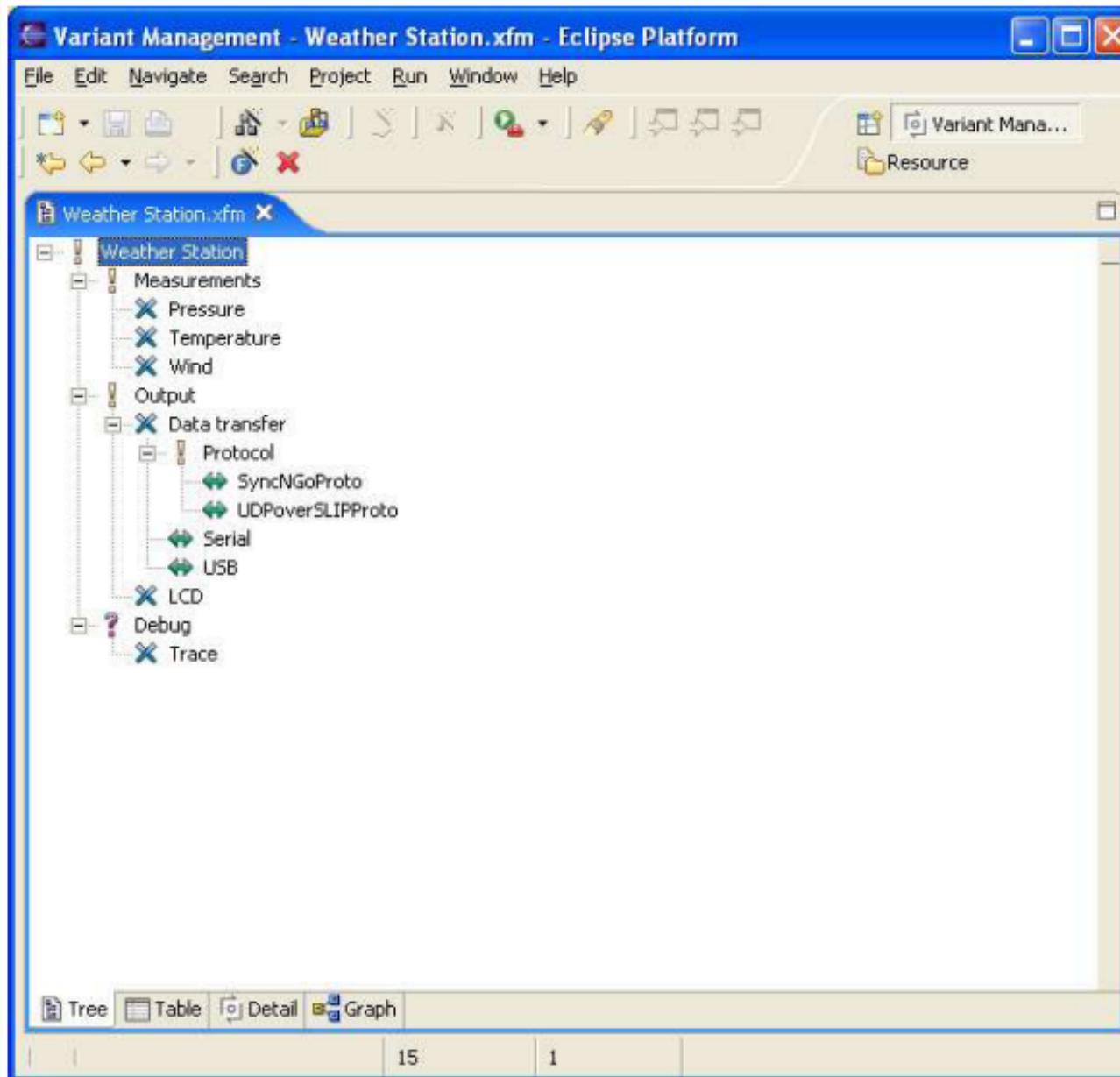


FIGURA 6. INTERFACE DE EDIÇÃO DO FEATURE MODEL EM MODO ÁRVORE DO PURE::VARIANTS

The screenshot shows the Variant Management interface in Table mode. The main window displays a table of features for the "Weather Station.xfm" model. The columns are "Unique ID", "Unique Name", "Visible Name", "Type", "Changed", and "Created". The table lists 15 features, including "Output", "UDPoverSLIPProto", "PCConnection", "USBLine", "Sensors", "SyncNGoProto", "Display", "WindSpeedSensor", "DebuggingSupport", "RS232Line", "TemperatureSensor", "AirPressureSensor", "Protocol", "WeatherMon", and "Trace". The "Trace" feature is currently selected, and its details are shown in a tooltip below the table:

Unique ID	Unique Name	Visible Name	Type	Changed	Created
22158	Output	Output	ps:mandatory		
522984859	UDPoverSLIPProto	UDPoverSLIPProto	ps:alternative		
522984853	PCConnection	Data transfer	ps:or		
522984860	USBLine	USB	ps:alternative		
522984855	Sensors	Measurements	ps:mandatory		
522984856	SyncNGoProto	SyncNGoProto	ps:alternative		
522984852	Display	LCD	ps:or		
522984862	WindSpeedSensor	Wind	ps:or		
522984851	DebuggingSupport	Debug	ps:optional		
522984854	RS232Line	Serial	ps:alternative		
522984857	TemperatureSensor	Temperature	ps:or		
522984850	AirPressureSensor	Pressure	ps:or		
22159	Protocol	Protocol	ps:mandatory		
522984861	WeatherMon	Weather Station	ps:mandatory		
522984858	Trace	Trace	ps:or		

FIGURA 7. INTERFACE DE EDIÇÃO DO FEATURE MODEL EM MODO TABELA DO PURE::VARIANTS

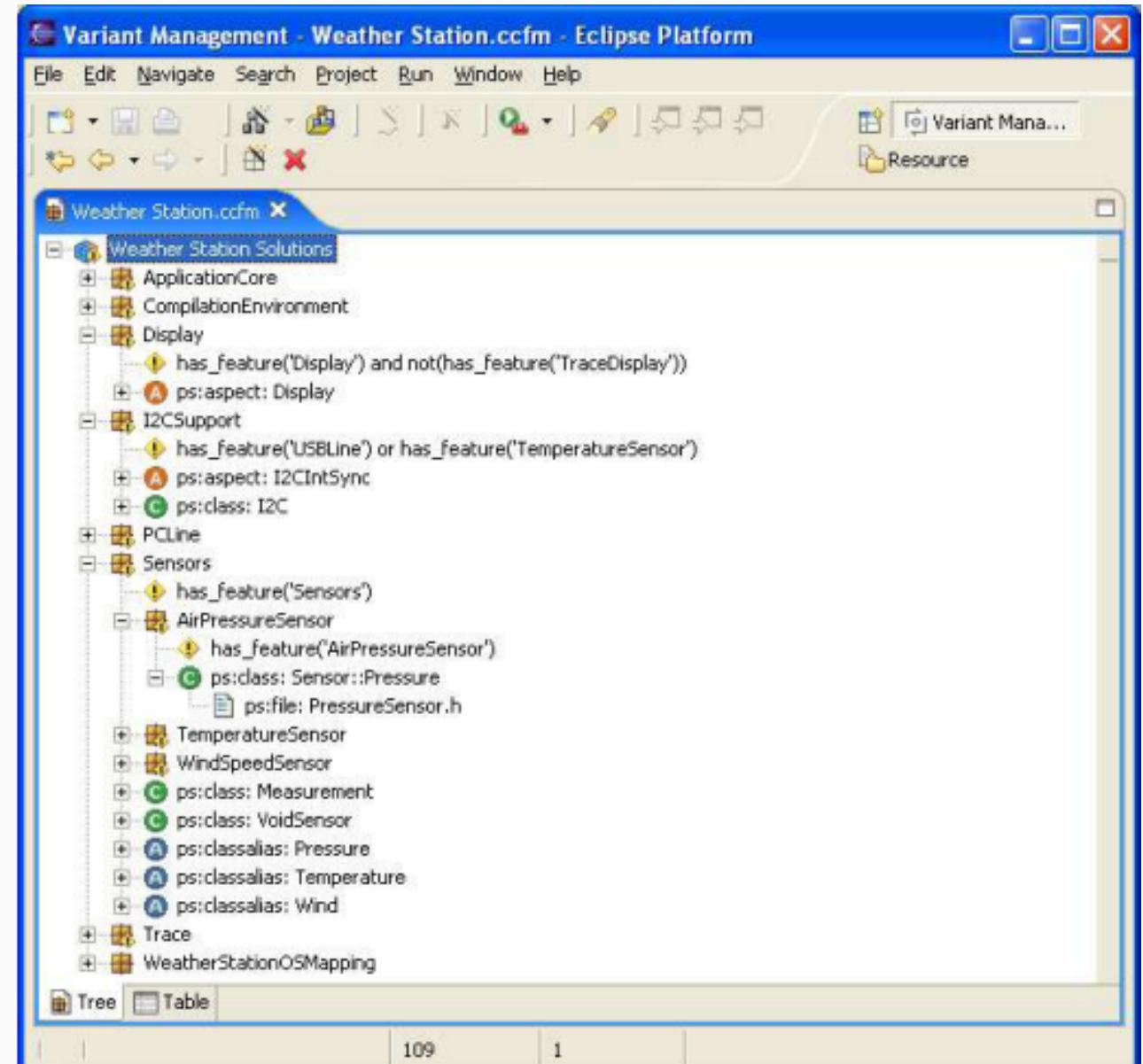


FIGURA 8. INTERFACE DE EDIÇÃO DO FAMILY MODEL DO PURE::VARIANTS

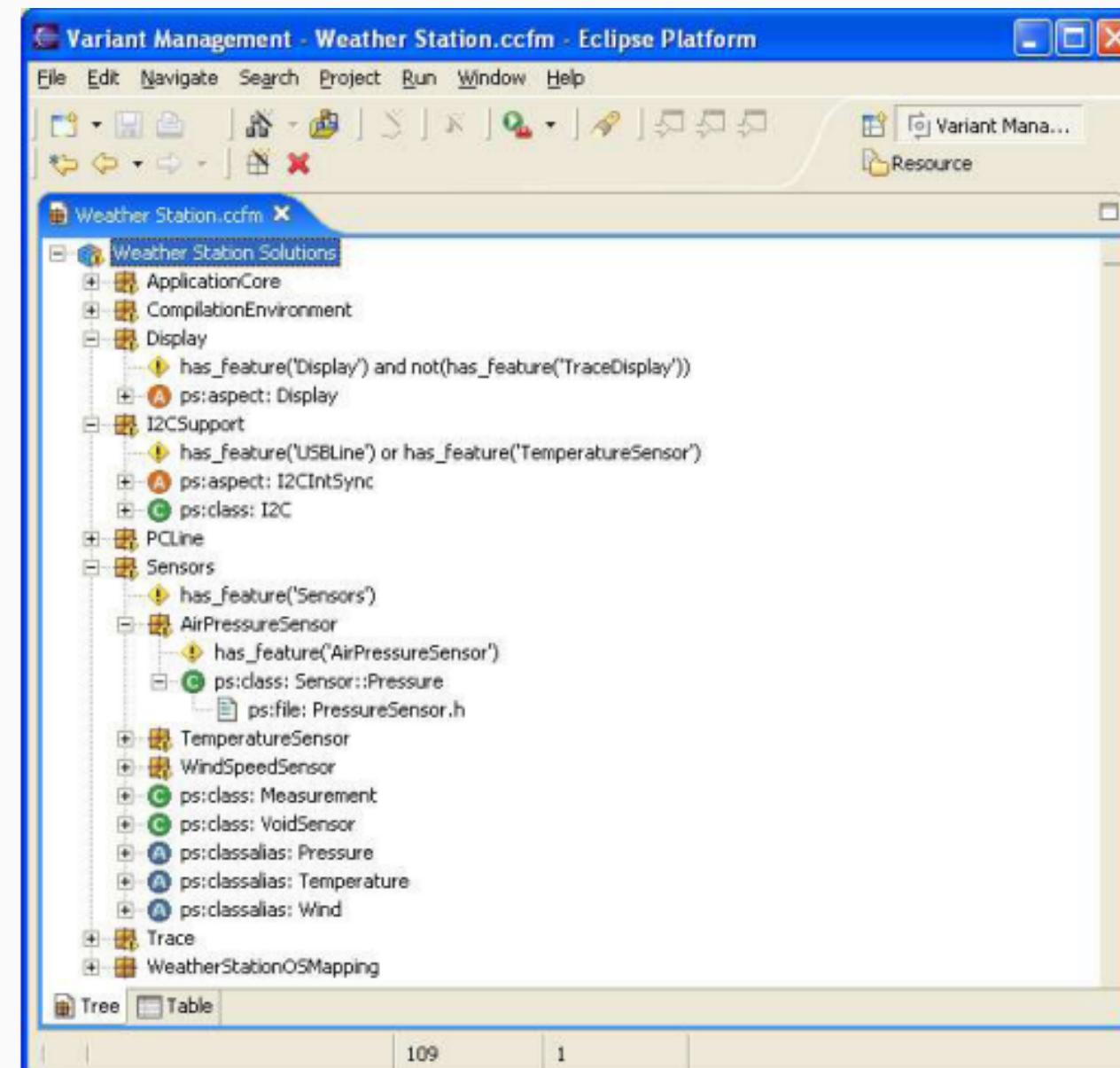


FIGURA 8. INTERFACE DE EDIÇÃO DO FAMILY MODEL DO PURE::VARIANTS

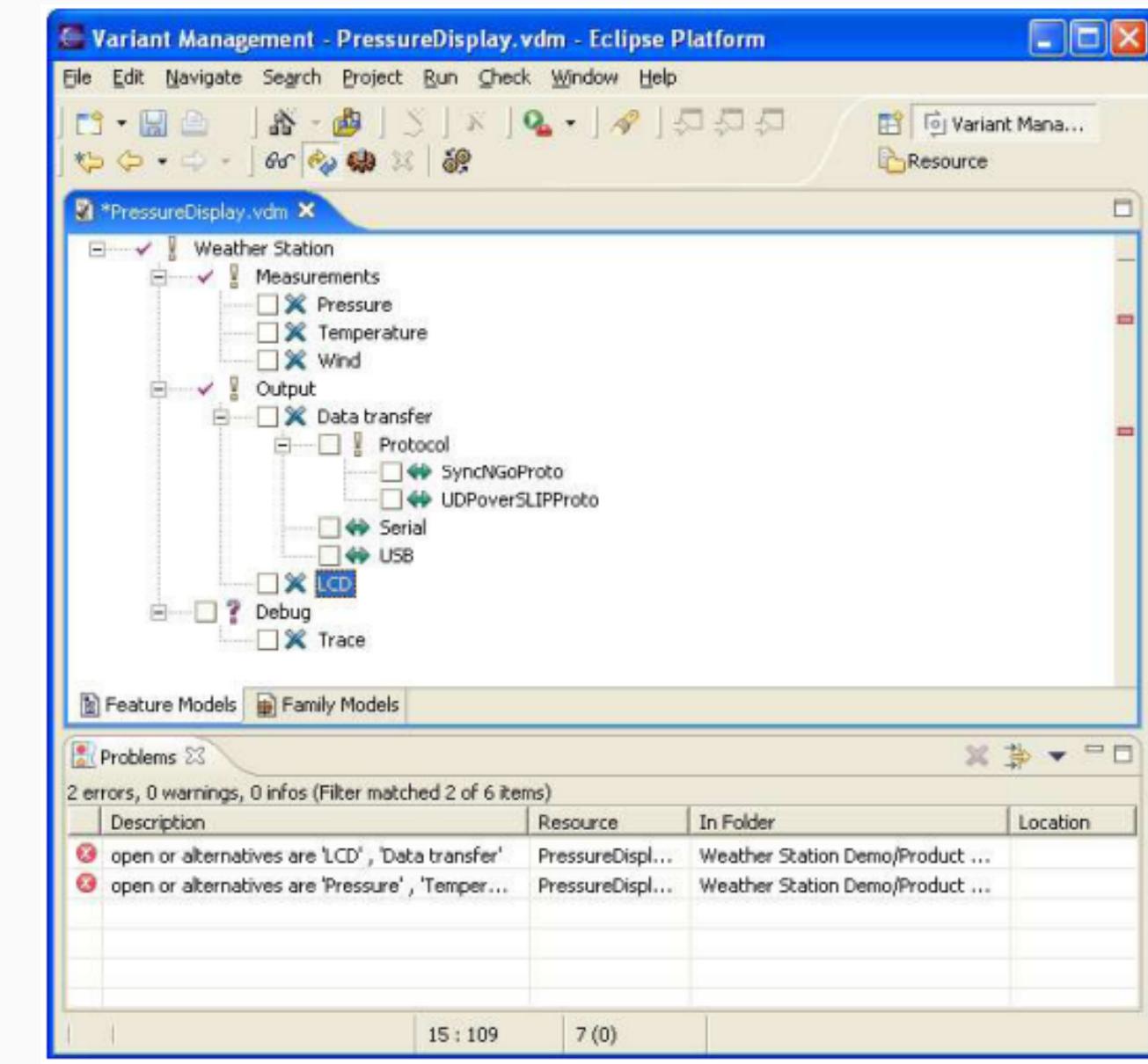


FIGURA 9. INTERFACE DE EDIÇÃO DE UM VARIANT MODEL DO PURE::VARIANTS

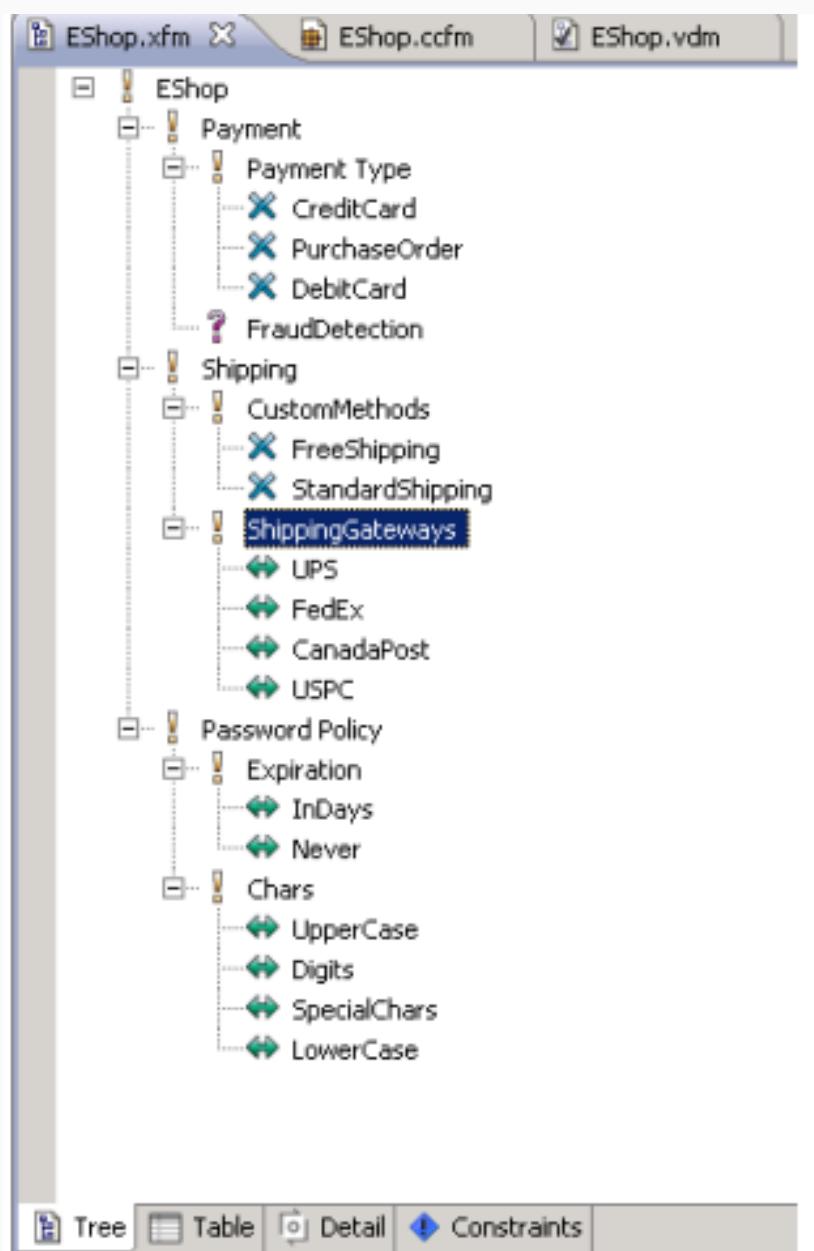


FIGURA 10. FEATURE MODEL DO
ESHOP NO PURE::VARIANTS

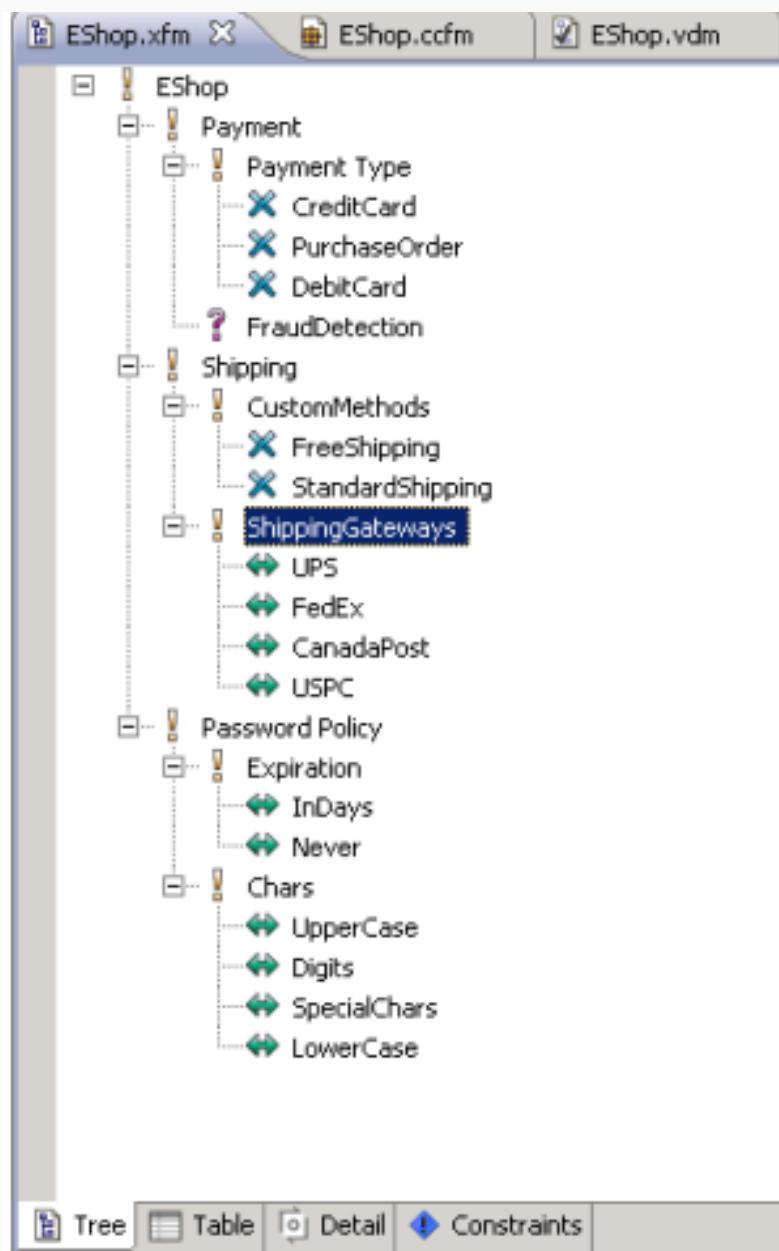


FIGURA 10. FEATURE MODEL DO ESHOP NO PURE::VARIANTS

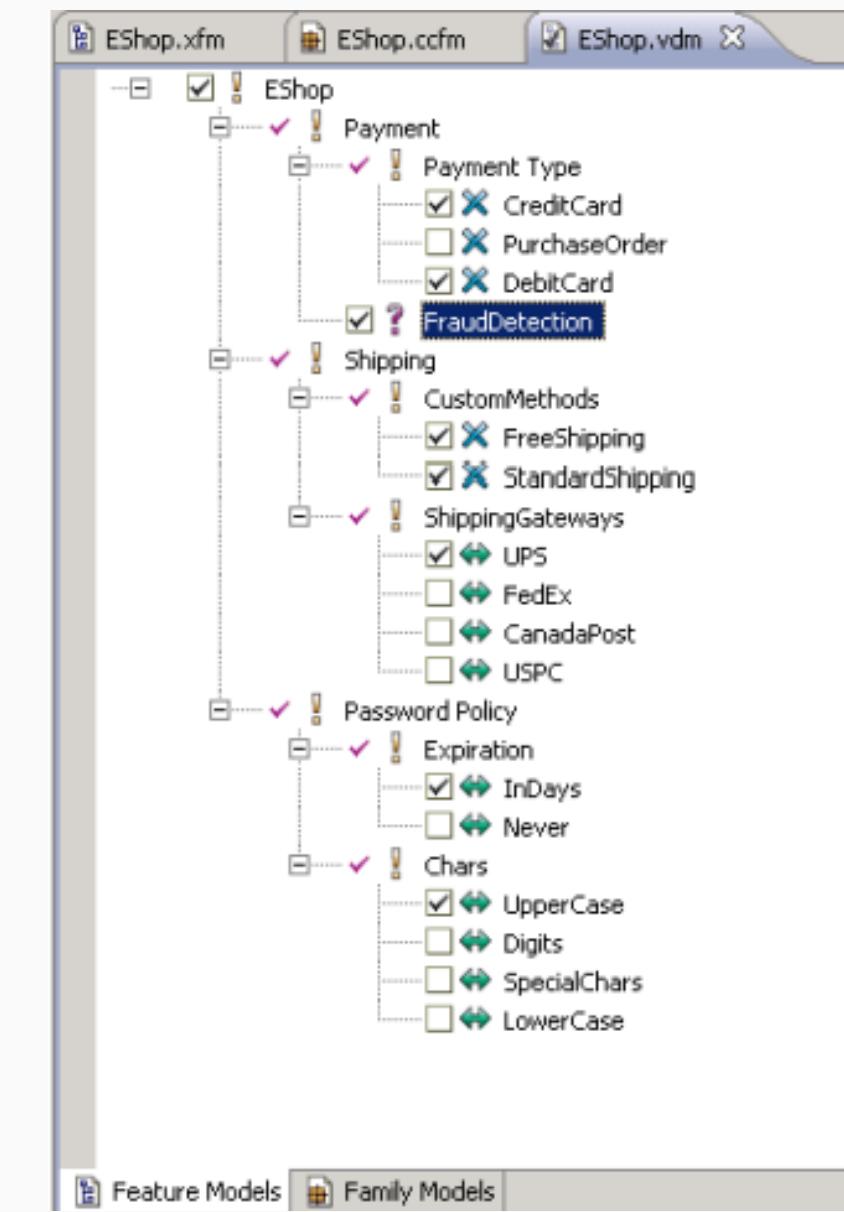


FIGURA 11. INSTÂNCIA DO ESHOP NO PURE::VARIANTS

4. Gears

4. Gears



Background

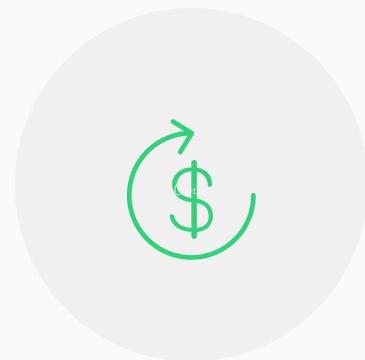
- ferramenta com fundo comercial

4. Gears



Background

- ferramenta com fundo comercial



Custo

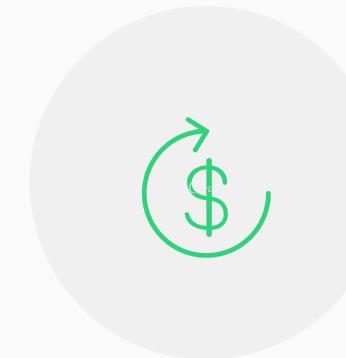
- custo exato é desconhecido
- pacote de avaliação custa 5 mil dólares

4. Gears



Background

- ferramenta com fundo comercial



Custo

- custo exato é desconhecido
- pacote de avaliação custa 5 mil dólares



Implementação

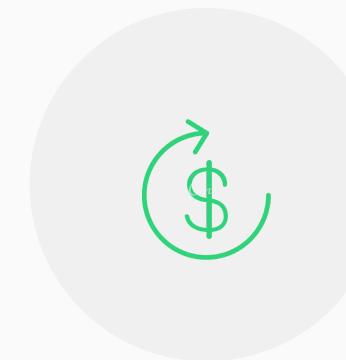
- informações escassas
- utiliza uma metodologia de linhas de produtos chamada 3-Tiered Software Product Line Methodology

4. Gears



Background

- ferramenta com fundo comercial



Custo

- custo exato é desconhecido
- pacote de avaliação custa 5 mil dólares



Implementação

- informações escassas
- utiliza uma metodologia de linhas de produtos chamada 3-Tiered Software Product Line Methodology



Técnica de modelagem
e notação utilizada

- utiliza uma notação própria

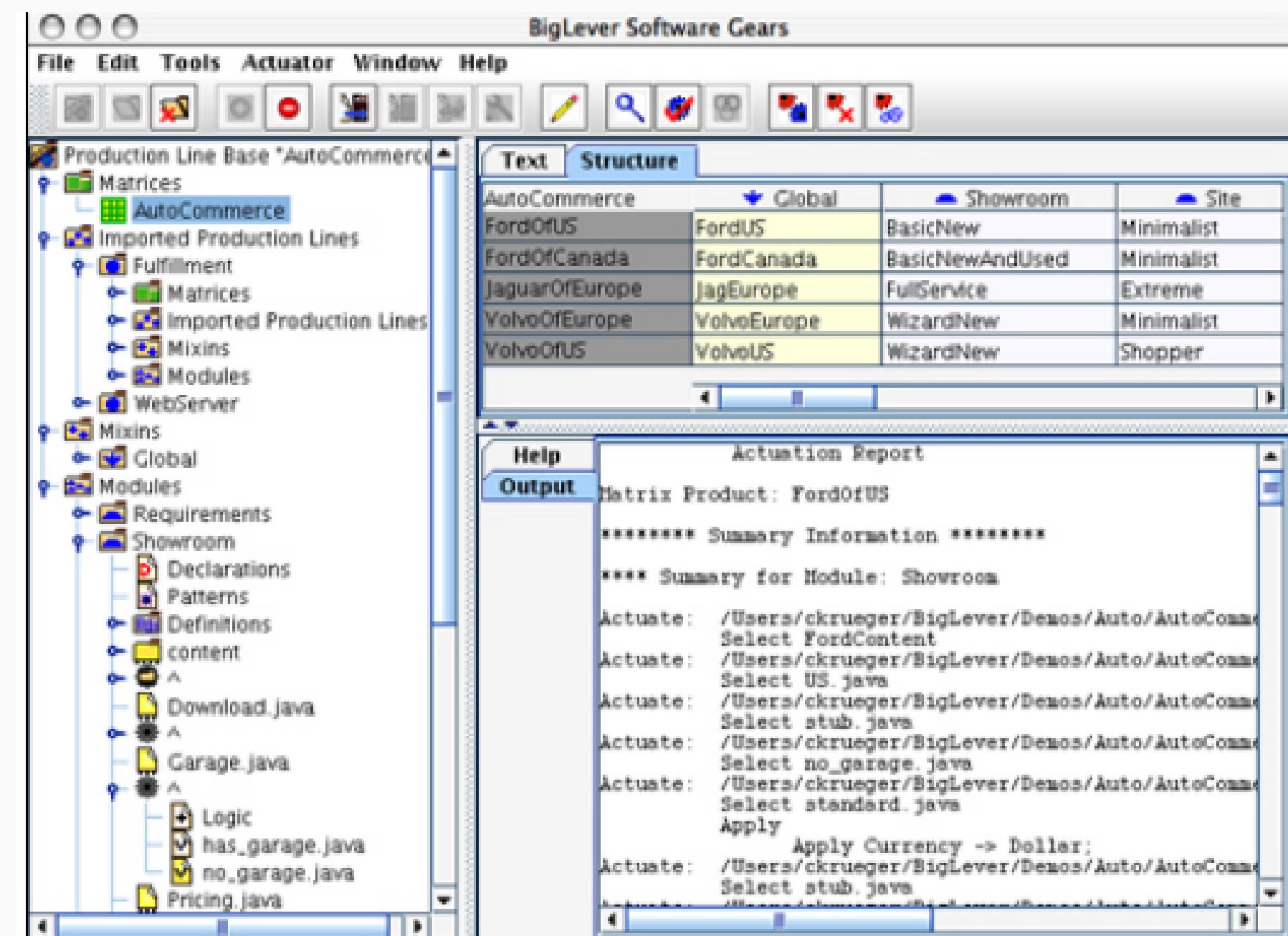


FIGURA 11. INTERFACE DO GEARS

Comparativo entre as ferramentas

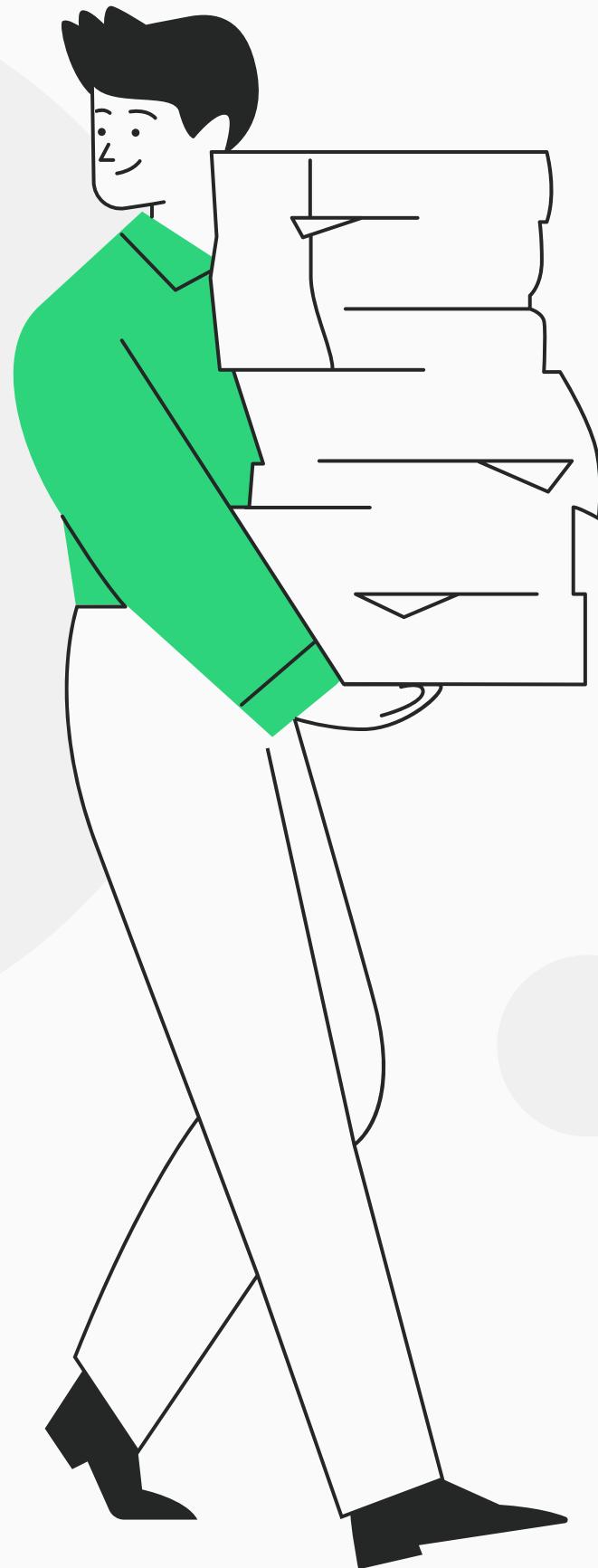
Critério	Ferramentas			
	<i>Fmp</i>	<i>XFeature</i>	<i>pure::variants</i>	<i>Gears</i>
<i>Background</i>	Acadêmico	Acadêmico	Industrial	Industrial
Suporte a processo	Não	Não	Sim	Sim
Notação utilizada	Árvore	Árvore e Diagrama	Árvore, Diagrama e Tabela	Desconhecido
Implementação	<i>Plugin</i>	<i>Plugin</i>	<i>Plugin</i>	<i>Stand-alone</i>
Tipo de licença	<i>Open Source</i>	<i>Open Source</i>	Gratuito e Comercial	Comercial
Código fonte	Disponível	Disponível	Não disponível	Não disponível
Técnicas de modelagem	Baseada em cardinalidade	Variável	Derivada de <i>FODA</i>	Desconhecido

TABELA 2. PRINCIPAIS CARACTERÍSTICAS DAS FERRAMENTAS

Critério	Ferramentas		
	<i>fmp</i>	<i>XFeature</i>	<i>pure::variants</i>
Representação de dados em forma tabular	Não	Não	Sim
Checagem de consistência	Sim	Sim	Sim
Documentação das <i>features</i>	Sim	Sim	Sim
Controle de Versão	Não	Não	Sim
Possui <i>configuration knowledge</i>	Não	Não	Sim
Derivação de produtos	Não	Não	Sim
Restrições	XPath e linguagem própria	XSL	Prolog / pvSCL
Cardinalidade em relacionamentos	Sim	Sim	Não

TABELA 3. RECURSOS DAS FERRAMENTAS

Outras ferramentas existentes



FAMA-FW

framework de análises automatizadas de modelos de features

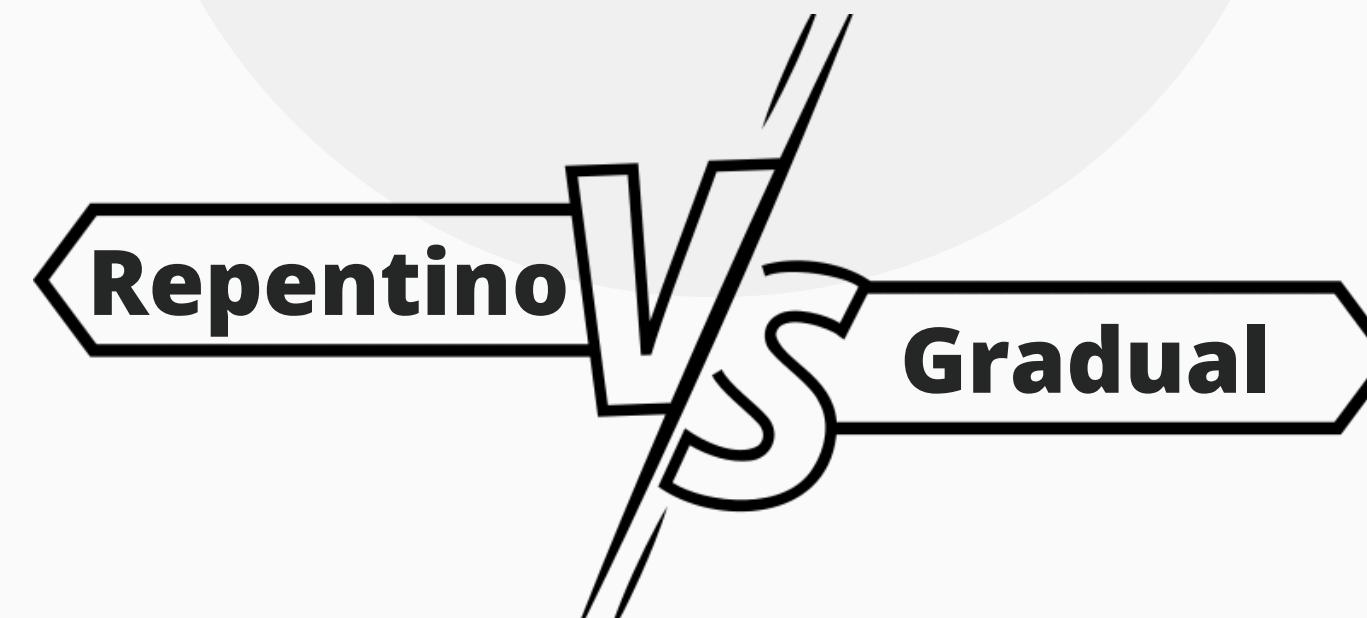
KUMBANG TOOLS

conjunto de ferramentas para configuração de famílias de produtos de software

ODYSSEY DOMAIN SDE

infra-estrutura de reutilização de software baseada em modelos de domínio que fornece ferramentas automatizadas para apoiar as fases distintas de um processo de reutilização

Desafios da arquitetura de software / engenharia de software



Falta de liderança

PARA O PROCESSO DE ADOÇÃO SUGERE-SE A ALOCAÇÃO DE UM LÍDER, UMA PESSOA QUE ACREDITA NOS PRINCÍPIOS DO MODELO E QUE ESTARÁ SUPERVISIONANDO O PROCESSO E MOTIVANDO AS PESSOAS.



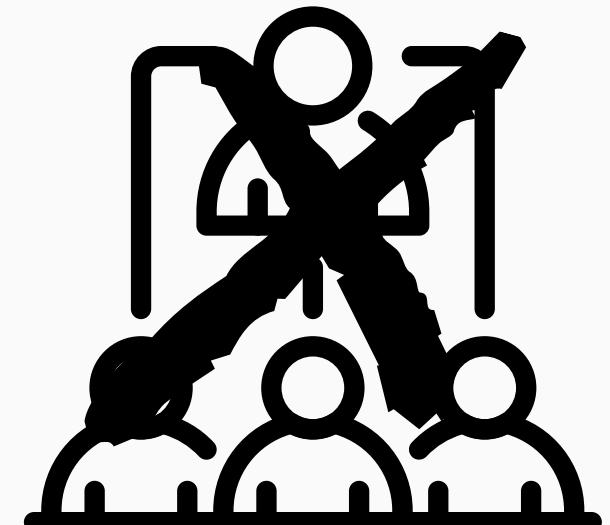
Falta de liderança

PARA O PROCESSO DE ADOÇÃO SUGERE-SE A ALOCAÇÃO DE UM LÍDER, UMA PESSOA QUE ACREDITA NOS PRINCÍPIOS DO MODELO E QUE ESTARÁ SUPERVISIONANDO O PROCESSO E MOTIVANDO AS PESSOAS.



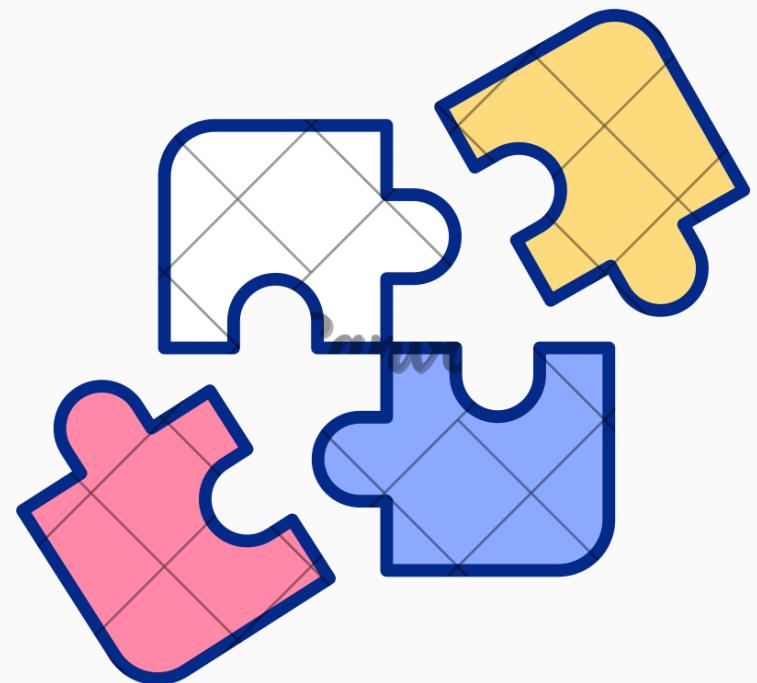
Falta de compromisso da gerencia

A GERÊNCIA DEVE ESTAR CONVENCIDA DA VIABILIDADE E DAS VANTAGENS QUE O PROJETO VAI TRAZER PARA A ORGANIZAÇÃO.



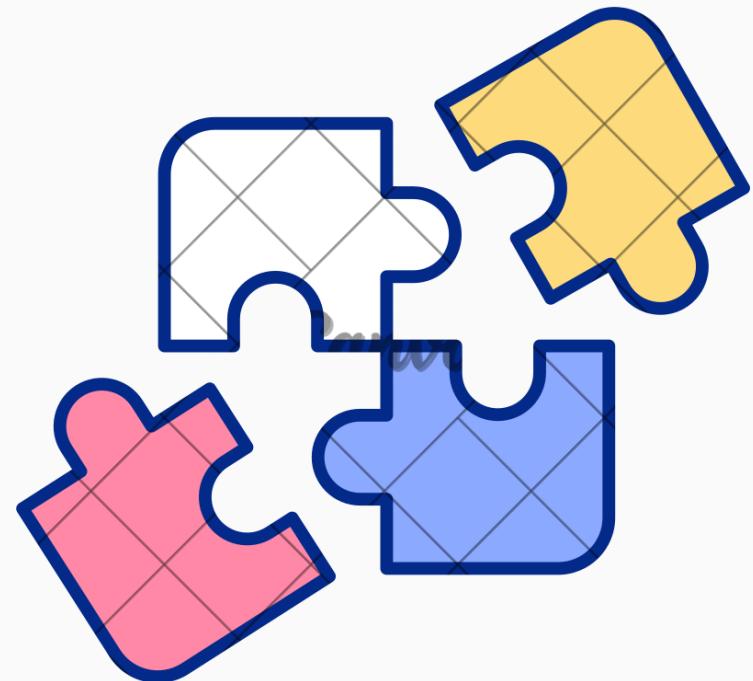
Abordagem inadequada

A LINHA DE PRODUTO DEVE SER UM PROJETO ESTRATÉGICO
PARA ATINGIR OBJETIVOS CORPORATIVOS.



Abordagem inadequada

A LINHA DE PRODUTO DEVE SER UM PROJETO ESTRATÉGICO
PARA ATINGIR OBJETIVOS CORPORATIVOS.



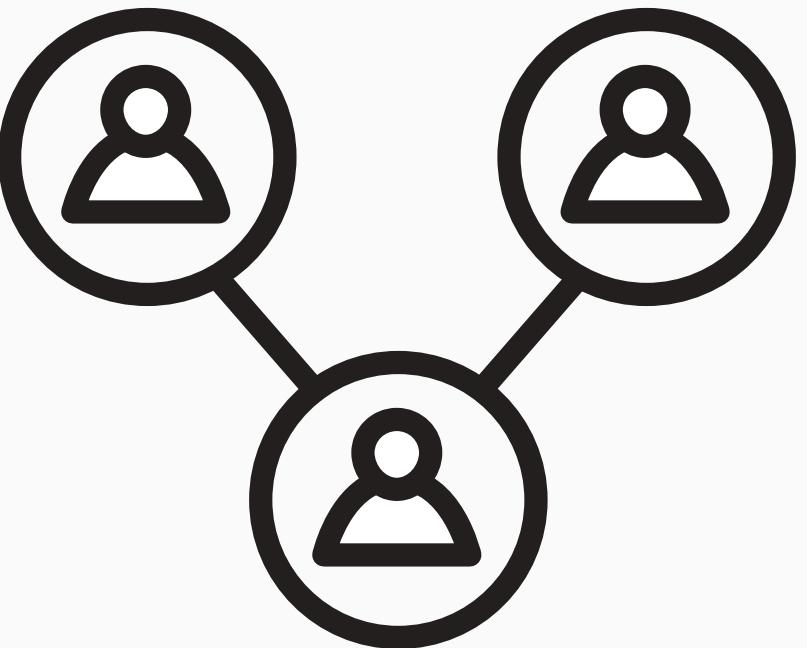
Compromisso da equipe

O LÍDER DA LINHA DEVE OBTER O COMPROMISSO DA EQUIPE
DESENVOLVEDORA COM OS PRINCÍPIOS E COM O PROCESSO DE
DESENVOLVIMENTO DA LINHA.



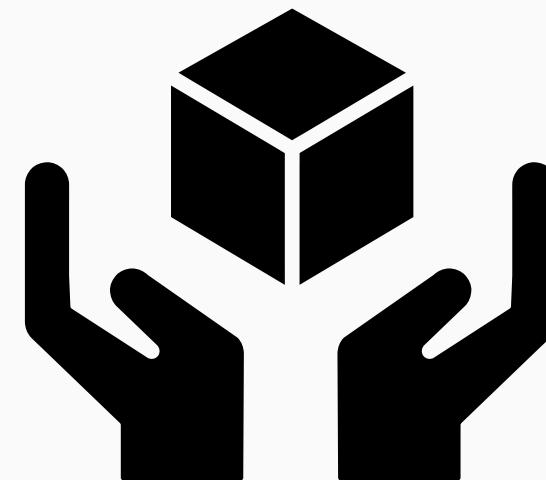
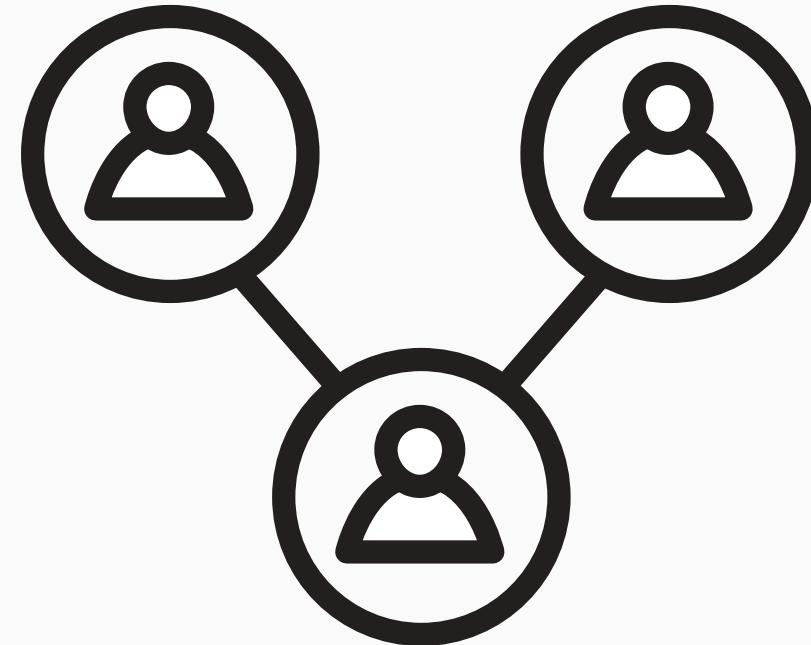
Falta de interação

A INICIATIVA DE UMA LINHA DE PRODUTO TRANSCENDE OS LIMITES TRADICIONAIS DA ORGANIZAÇÃO. UMA FALTA DE INTERAÇÃO E COLABORAÇÃO ENTRE ESSAS EQUIPES PODE IMPEDIR A OBTENÇÃO DOS OBJETIVOS PLANEJADOS.



Falta de interação

A INICIATIVA DE UMA LINHA DE PRODUTO TRANSCENDE OS LIMITES TRADICIONAIS DA ORGANIZAÇÃO. UMA FALTA DE INTERAÇÃO E COLABORAÇÃO ENTRE ESSAS EQUIPES PODE IMPEDIR A OBTENÇÃO DOS OBJETIVOS PLANEJADOS.

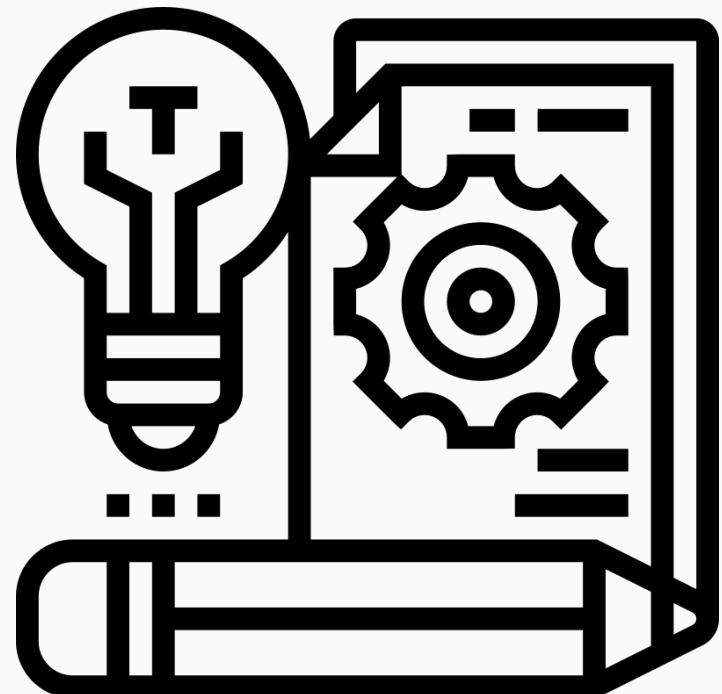


Padronização inapropriada

A INSTITUCIONALIZAÇÃO PREMATURA E IMPENSADA DE PADRÕES, MUITAS VEZES OBSOLETOS, PODE LIMITAR AS OPÇÕES DE TECNOLOGIA DA LINHA, PREJUDICANDO O ESCOPO DA MESMA.

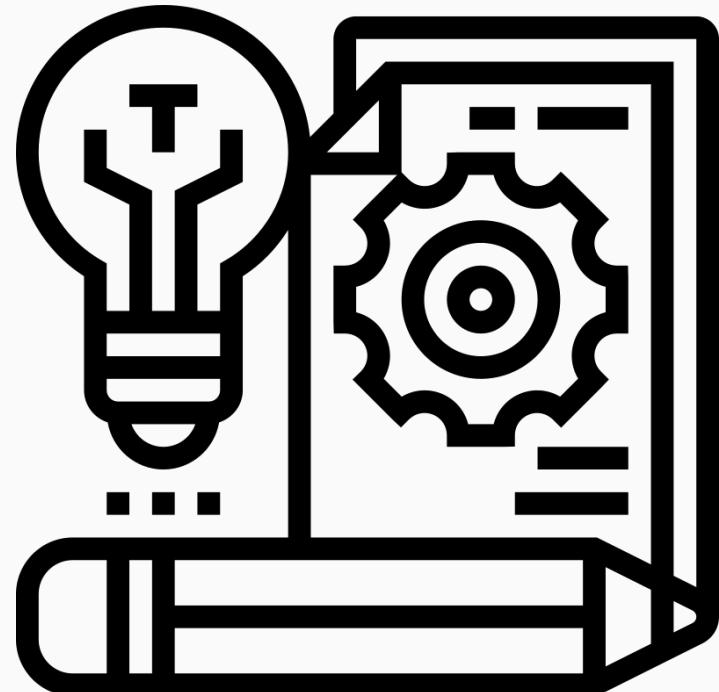
Adaptação insuficiente

AS PRÁTICAS ORGANIZACIONAIS (PADRONIZADAS OU NÃO) DEVEM PERMITIR UM GRAU DE ADAPTAÇÃO, OU MELHOR, PERSONALIZAÇÃO VOLTADAS PARA AS CARACTERÍSTICAS DA LINHA.



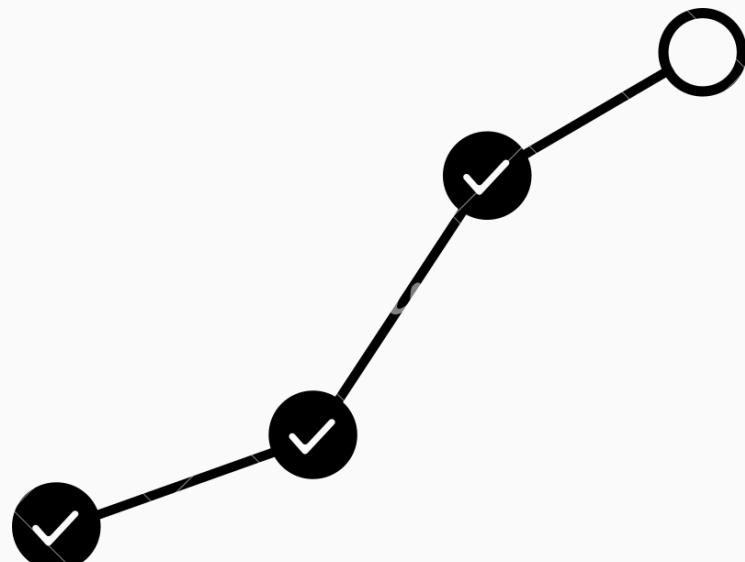
Adaptação insuficiente

AS PRÁTICAS ORGANIZACIONAIS (PADRONIZADAS OU NÃO) DEVEM PERMITIR UM GRAU DE ADAPTAÇÃO, OU MELHOR, PERSONALIZAÇÃO VOLTADAS PARA AS CARACTERÍSTICAS DA LINHA.



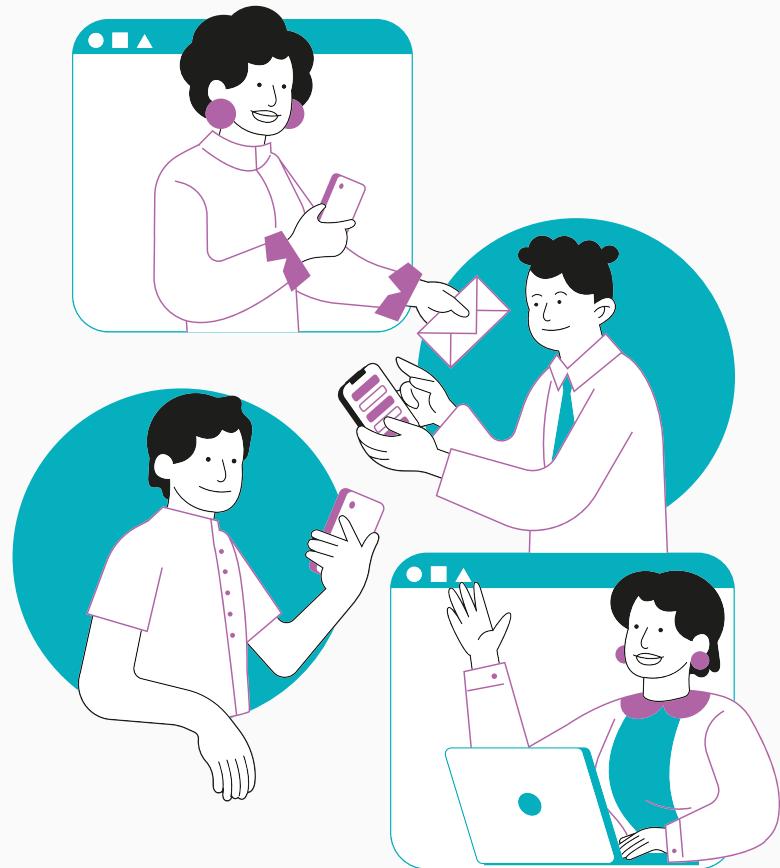
Evolução da abordagem

CASO O PROCESSO DA LINHA DE PRODUTO NÃO SEJA REVISADO E ATUALIZADO PERIODICAMENTE, SUAS PRÁTICAS IRÃO TORNAR-SE OBSOLETAS E INAPROPRIADAS.



Falha de disseminação

O LÍDER DA LINHA DEVE TOMAR A RESPONSABILIDADE DE DESENVOLVER E DISTRIBUIR AS DOCUMENTAÇÕES EM NÍVEL E TIPO APROPRIADOS PARA A ORGANIZAÇÃO.



Referências

- N. WANG, "**FEATURE ORIENTED ARCHITECTURE FOR WEB APPLICATIONS - REACT WEEKLY - MEDIUM,**" MEDIUM, JUL. 04, 2017.
<HTTPS://MEDIUM.COM/REACT-WEEKLY/FEATURE-ORIENTED-ARCHITECTURE-FOR-WEB-APPLICATIONS-2B48E358AFB0> (ACCESSED APR. 07, 2023).
- A. GUPTA, **JAVA EE 7 ESSENTIALS: ENTERPRISE DEVELOPER HANDBOOK.** BEIJING: O'REILLY, 2013.
- R. C. DURSCKI, M. DE, R. BURNETT, AND IMACULADA CONCEIÇÃO, "**LINHAS DE PRODUTO DE SOFTWARE: RISCOS E VANTAGENS DE SUA IMPLANTAÇÃO,**" RESEARCHGATE, 2004.
HTTPS://WWW.RESEARCHGATE.NET/PUBLICATION/242761922_LINHAS_DE_PRODUTO_DE_SOFTWARE_RISCOS_E_VANTAGENS_DE_SUA_IMPLANTACAO (ACCESSED APR. 07, 2023).
- T. ERL, "**SERVICE-ORIENTED ARCHITECTURE: CONCEPTS, TECHNOLOGY, AND DESIGN**", PRENTICE HALL, 2005.
- G. T. HEINEMAN AND W. T. COUNCILL, "**COMPONENT-BASED SOFTWARE ENGINEERING: PUTTING THE PIECES TOGETHER**", ADDISON-WESLEY PROFESSIONAL, 2001.
- JÚNIOR, ROGÉRIO. "**COMPARAÇÃO ENTRE FERRAMENTAS PARA LINHA DE PRODUTOS DE SOFTWARE**". ORIENTADOR: SÉRGIO SOARES. 2008. 46 F. TCC (GRADUAÇÃO) – CURSO DE ENGENHARIA DA COMPUTAÇÃO, SISTEMAS E COMPUTAÇÃO, UNIVERSIDADE DE PERNAMBUCO, RECIFE. 2008. DISPONÍVEL EM: <HTTP://TCC.ECOMP.POLI.BR/20081/ROGERIOMONOGRAFIAFINAL.PDF>. ACESSO EM: 8 ABRIL 2023.
- DURSCKI ROBERTO, SPINOLA MAURO, SPINOLA ROBERT, REINEHR SHEILA, "**LINHAS DE PRODUTO DE SOFTWARE: RISCOS E VANTAGENS DE SUA IMPLANTAÇÃO**". DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO – ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO (USP), PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ - RUA: IMACULADA CONCEIÇÃO.
- SOUZA, RAMON, JUCA, PAULYNE. "**METODOLOGIAS PARA ANÁLISE DE DOMÍNIO**". DISPONÍVEL EM:
<HTTP://ROMEIRAO.QUIXADA.UFC.BR/PORTAL/WP-CONTENT/UPLOADS/2014/04/METODOLANALISDOMIN.175.PDF>. ACESSO EM: 21 ABRIL 2023.