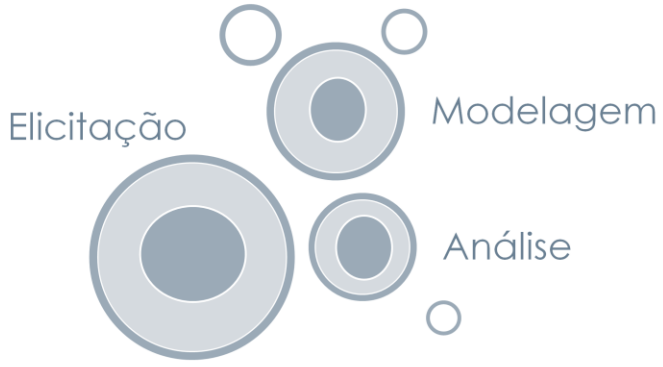


Requisitos – Aula 13

Professores: Milene Serrano e Maurício Serrano

Agenda

- Considerações Iniciais
- Modelagem de Requisitos
  - Casos de Uso
    - FOCO: Requisitos Funcionais
  - Especificação Suplementar
    - FOCO: Requisitos Não-Funcionais
- Típicos Processos/Metodologias
- Considerações Finais

Considerações Iniciais

Considerações Iniciais

Quando começamos a desenvolver um software...

- É comum pensarmos na parte comportamental do mesmo...
- Portanto, pensamos em comportamentos como cadastrar, editar, visualizar, encontrar, registrar, armazenar, inserir, remover e assim vai...
- O que são esses aspectos?
  - **FUNCIONALIDADES!**

Considerações Iniciais

Mas, como modelar esse tipo de aspecto usando uma abordagem de modelagem tradicional?

- Primeiramente, o que é uma abordagem de modelagem tradicional?
  - Depende do paradigma de programação...
  - Se Estruturado, então: Diagrama de Fluxo de Dados (DFD) e Dicionário de Dados (DD) são boas alternativas.
  - Mas, o mais comum é estarmos usando Orientação a Objetos ou mesmo um paradigma que tem fronteiras com a Orientação a Objetos. Portanto, uma notação que ganha força é a (*Unified Modeling Language*) UML.

DFD - Exemplo


Atividade Essencial
(realização de uma
tarefa específica)
Agente Externo
pedido
CLIENTE
Registrar
Pedido
clientes
dados_cliente
pedidos
comprovante
dados_pedido
Fluxo de Dados
Depósito de D

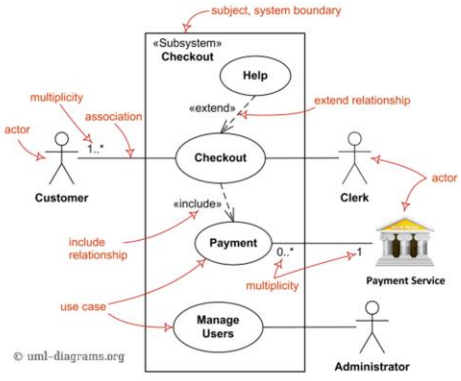

Vendedor
solicitação
confirmacão
Cadastrar
Solicitação
cliente
lançamento
registro
CLIENTES
FATURAS
PEDIDOS

Dicionário de Dados Exemplos

= é composto por
+ e
() opcional
{} iteração
[] selecionar uma das várias alternativas
** comentário
@ chave de um depósito
| separa alternativas quando se usa [ ]

Nome = título + primeiro-nome + apelido
título = [Sr. | Sra.| Prof.| Dr.| Eng.]
primeiro-nome = 1{caracter-válido}
apelido = 1{caracter-válido}
caracter-válido = [A-Z | a-z | ' | - |]
endereço = * ainda não definido*

UML – Diagrama de Casos de Uso



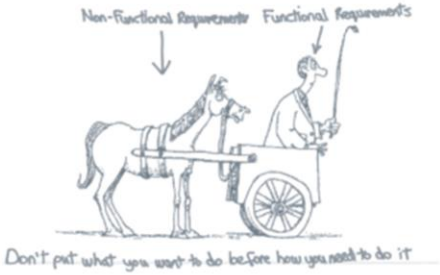
Vamos conhecer!

Diagrama de Casos de Uso – Principal diagrama da UML para modelar requisitos, com foco em requisitos funcionais.

Considerações Iniciais

Um software não deve ser pensado somente em termos de funcionalidades.

- Pelo contrário, sabemos que não tratar os requisitos não funcionais é uma causa clara de insucesso de produtos de software. Vide caso da Ambulância de Londres. Conhecem? Lembram?
- Mas, lidar com requisitos não funcionais é algo complexo, abstrato.
- Uma funcionalidade é possível dizer se ela foi realizada ou não. Certo?
- Mas, um requisito não funcional, tipo privacidade, pode ser parcialmente satisfeito. Pode ser 80% satisfeito, na opinião de um interessado; ou 50% na opinião de outro; ou ainda menos de 20% na opinião de vários.



Considerações Iniciais

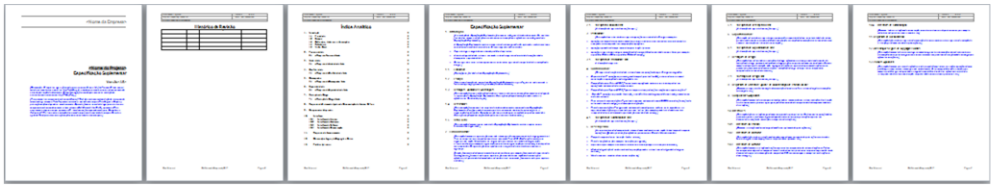
Portanto, como lidar com os requisitos não funcionais?

Existem várias propostas para especificar e modelar esse tipo de requisitos:

- Algumas mais tradicionais, como a Especificação Suplementar, e
- Algumas mais emergentes, como NFR Framework.

Especificação Suplementar

Nessa aula, iremos nos concentrar nessa especificação...



OBS: NFR Framework será visto mais adiante no curso de Requisitos de Software. **Aguardem**! :)

NFR Framework será visto mais adiante no curso de Requisitos de Software...
Aguardem! :)

Modelagem de Requisitos

Casos de Uso

Casos de Uso

Também chamados de diagramas comportamentais, na notação da UML.

Usados para descrever um conjunto de ações (*use cases* – casos de uso) que um sistema ou um conjunto de sistemas (*subject* - sujeito) deve desempenhar em colaboração com um ou mais usuários externos ao sistema (*actors* - atores).

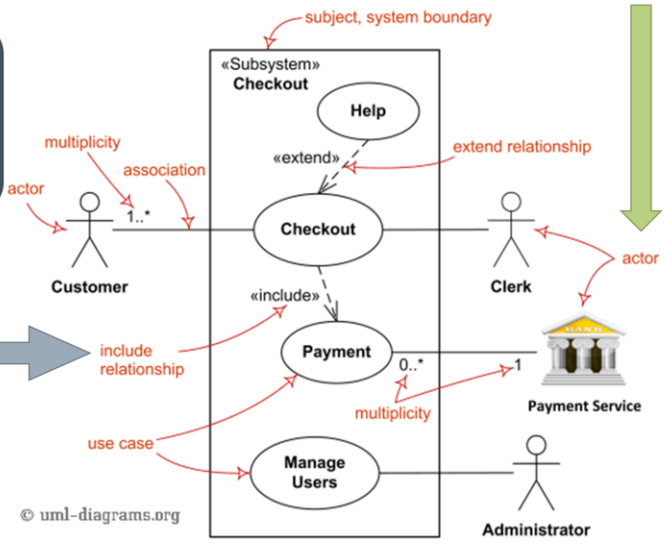
Cada caso de uso deverá prover algum resultado observável e de valor para os atores ou outros interessados do sistema.

Casos de Uso - Notação

Os atores podem ser atores humanos – pessoas; ou podem ser sistemas, subsistemas, componentes, chamados atores sistêmicos.

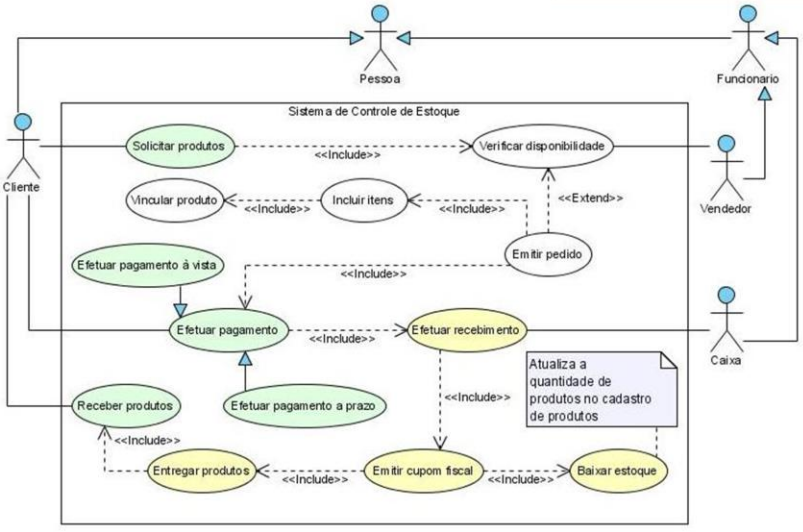
As elipses, as quais representam as ações - casos de uso – constituem, internamente, fluxos / cenários.

Payment é um substantivo! Evitem! Usem sempre **verbos no infinitivo**, indicando que são ações! Ok?



Casos de Uso - Exemplo

Os típicos relacionamentos são: *extend* (extensão), *include* (inclusão) e *generalization* (herança)



Especificação de Casos de Uso

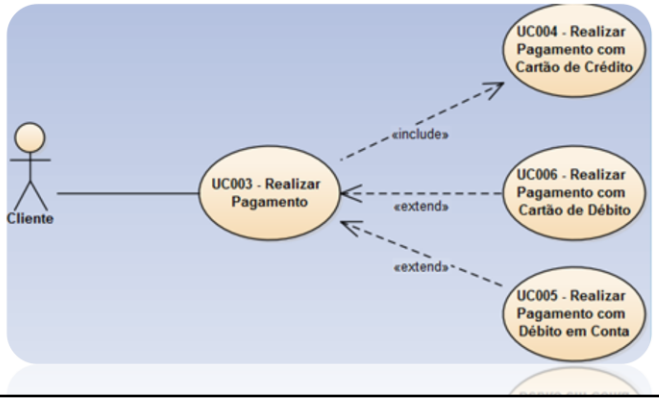
Como recomendamos, desde a primeira aula sobre modelagem, um modelo deve ser claro e objetivo.

Portanto, caso seja necessário um detalhamento maior, é recomendado apelar para uma especificação em linguagem natural.

Para a notação de Casos de Uso, existe uma especificação, já estabelecida na comunidade, a qual complementa a visão do Diagrama de Casos de Uso. Chama-se: Especificação de Casos de Uso.

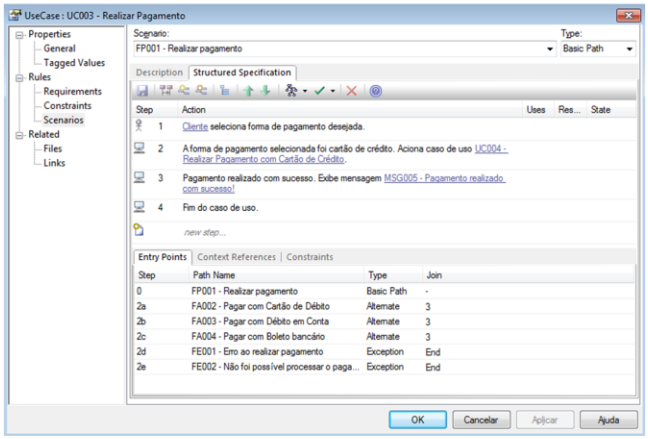
Especificação de Casos de Uso

Para explicar essa especificação, é adequado considerarmos um exemplo-base. Considerem:



Especificação de Casos de Uso

Vamos focar no UC003, especificando o fluxo: FP001 Realizar pagamento



Especificação de Casos de Uso

Mas, podemos ter vários tipos de fluxo...

Podemos ter:

- Fluxo Principal,
- Fluxo Alternativo, e
- Fluxo de Exceção

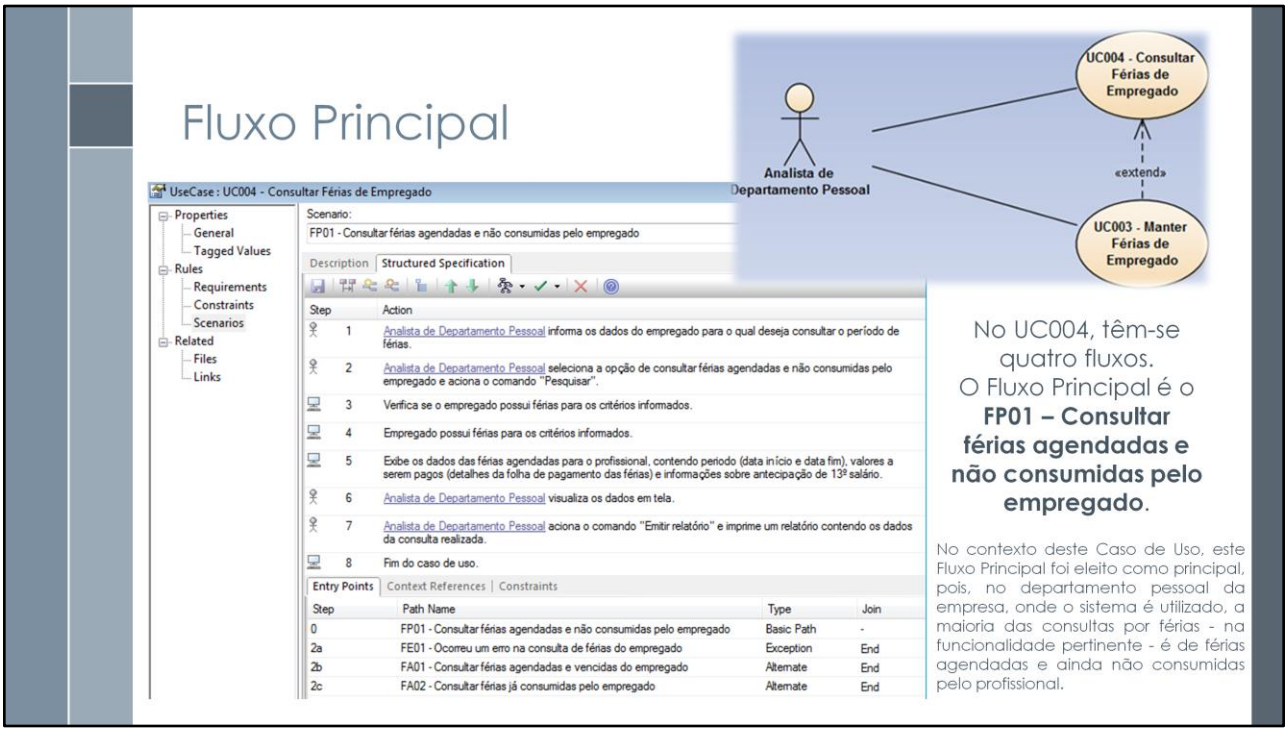
Fluxo Principal

Entendendo o Fluxo Principal...

Cada caso de uso tem somente um fluxo principal.

Trata-se da maneira "default" que o ator utilizará a funcionalidade, ou seja, é o que ele tentará realizar, primariamente, sempre que utilizar a funcionalidade.

Também chamado: Caminho Feliz, Fluxo Básico, Fluxo Ótimo ou Fluxo de Sucesso.



No contexto deste Caso de Uso, este Fluxo Principal foi eleito como principal, pois, no departamento pessoal da empresa, onde o sistema é utilizado, a maioria das consultas por férias - na funcionalidade pertinente - é de férias agendadas e ainda não consumidas pelo profissional.

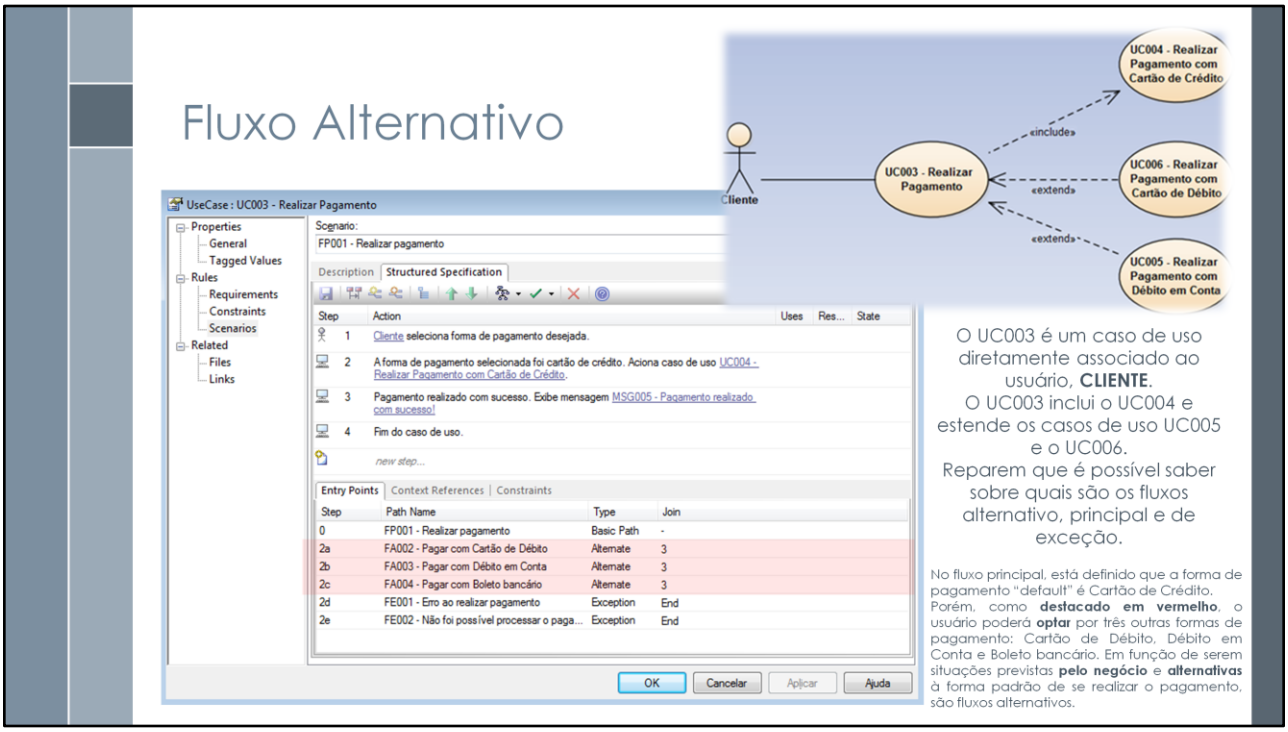
Fluxo Alternativo

Entendendo o Fluxo Alternativo...

Quando pensamos em Fluxos Alternativos, estamos falando realmente de escolhas que o usuário poderá fazer na execução de uma funcionalidade, escolhas que alterarão o comportamento da funcionalidade.

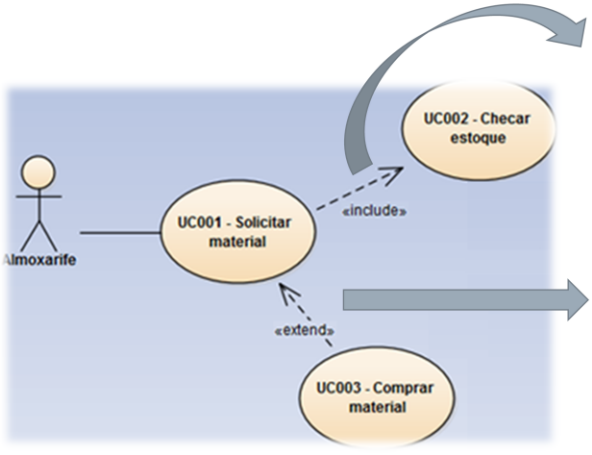
É bem diferente de Fluxo de Exceção... **Analogia**: O "caminho principal" era seguir reto. Como "caminhos alternativos", temos as opções de seguir à direita e seguir à esquerda. Já como "exceção", havia a possibilidade de uma pessoa, ao andar por algum caminho, cair no buraco.

Portanto, Fluxos Alternativos são fluxos que podem ser executados numa funcionalidade a partir da escolha do usuário, e não a partir de erros ou exceções do sistema.



No fluxo principal, está definido que a forma de pagamento "default" é Cartão de Crédito.
Porém, como **destacado em vermelho**, o usuário poderá **optar** por três outras formas de pagamento: Cartão de Débito, Débito em Conta e Boleto bancário. Em função de serem situações previstas **pelo negócio** e **alternativas** à forma padrão de se realizar o pagamento, são fluxos alternativos.

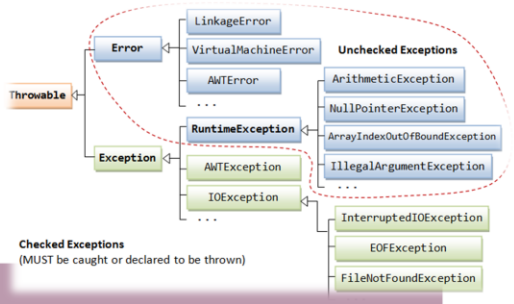
Relacionamentos *Include* e *Extend*



O caso de uso "Solicitar Material" faz *include* no caso de uso "Checar Estoque".
Assim, **sempre** que houver a solicitação de material, haverá a consulta ao estoque para saber se o material está disponível.

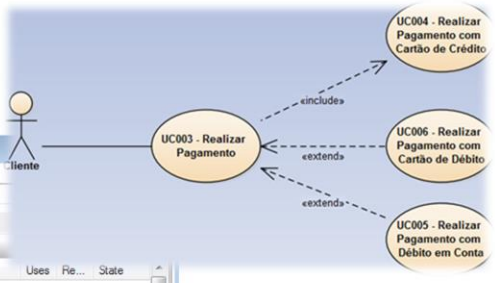
O caso de uso "Comprar Material" estende o caso de uso "Solicitar Material".
Quando houver a solicitação de material, **caso o material não exista em estoque** (após consulta via o caso de uso "Checar estoque"), **poderá** ser solicitada a compra do item.

Fluxo de Exceção



- Entendendo o Fluxo de Exceção...
- Exceções devem ser previstas e tratadas.
- É muito difícil um software tratar todas as possibilidades de exceção.
- Entretanto, quanto mais possibilidades forem previstas e tratadas, melhor.
- Isso colabora com a qualidade do produto final do projeto.

Fluxo de Exceção



UseCase : UC004 - Realizar Pagamento com Cartão de Crédito

- Properties
  - General
  - Tagged Values
- Rules
  - Requirements
  - Constraints
  - Scenarios
- Related
  - Files
  - Links

Scenario: FP001 - Realizar pagamento com cartão de crédito

Description | Structured Specification

Step	Action	Uses	Re...	State
1	Caso de uso chamador informa dados do cartão de crédito (número, código de segurança, data de validade e valor a ser pago).			
2	Valida dados do pagamento.			
3	Processa pagamento.			
4	Retorna sucesso ao chamador.			
5	Fim do case de uso.			

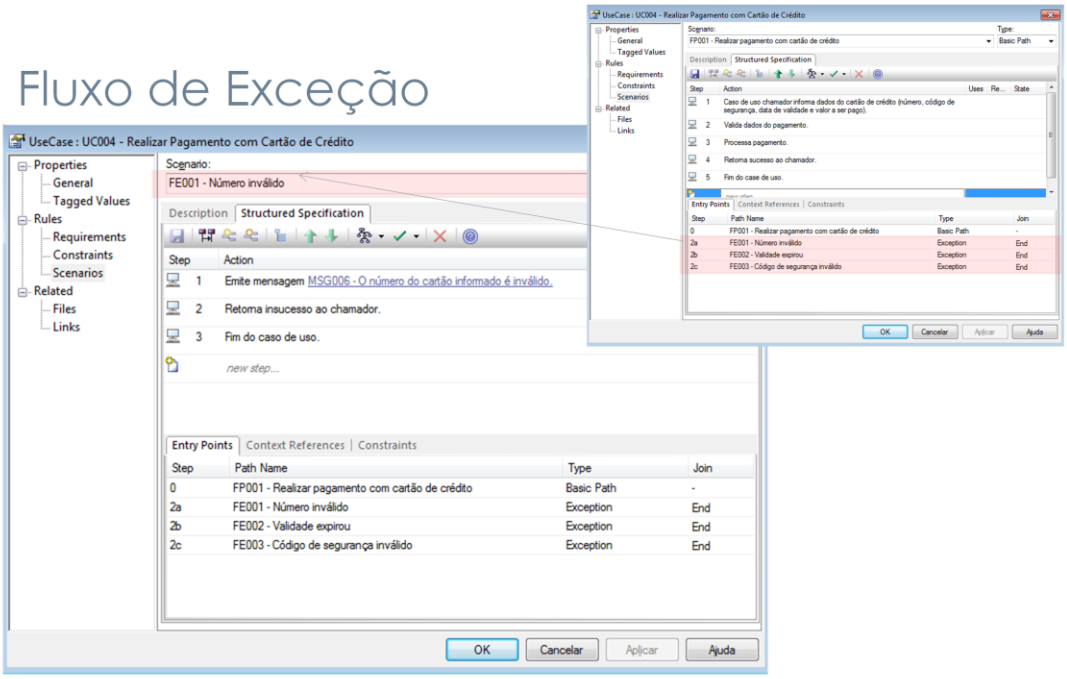
Entry Points | Context References | Constraints

Step	Path Name	Type	Join
0	FP001 - Realizar pagamento com cartão de crédito	Basic Path	-
2a	FE001 - Número inválido	Exception	End
2b	FE002 - Validade expirou	Exception	End
2c	FE003 - Código de segurança inválido	Exception	End

OK | Cancelar | Aplicar | Ajuda

Três Fluxos de Exceção!

Fluxo de Exceção



Modelagem de Requisitos

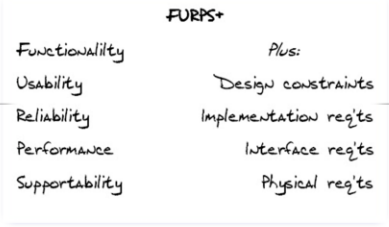
Especificação Suplementar

Especificação Suplementar

Trata-se de um documento em linguagem natural, no qual são descritos os requisitos não funcionais. Outros detalhes, acessem:

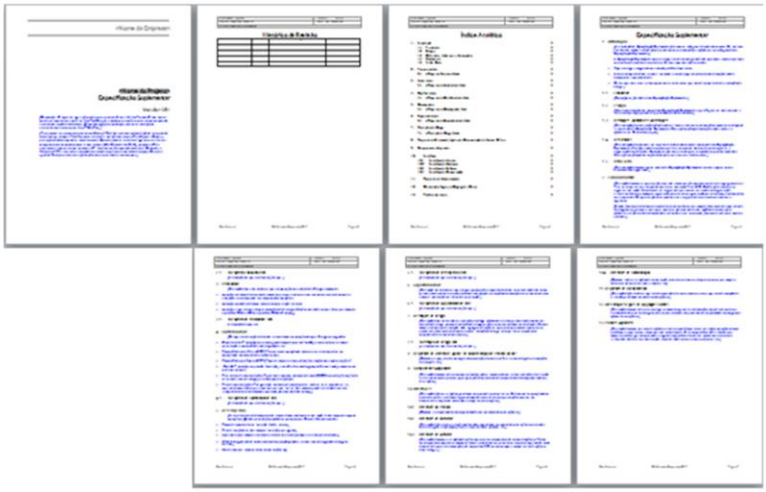
http://www.funpar.ufpr.br:8080/rup/process/artifact/ar_sspec.htm

Visando auxiliar os engenheiros de software, esse documento orienta-se pelo FURPS+



Especificação Suplementar

Especificação Suplementar



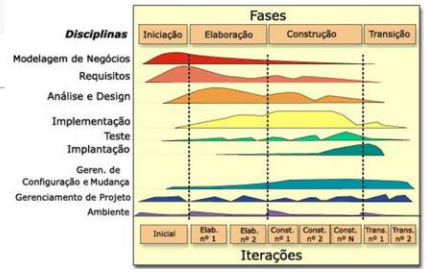
<<Mostrar Template>>

Típicos Processos/Metodologias

Principal Processo: RUP

O principal processo de desenvolvimento, conhecido por usar em suas especificações artefatos na notação UML.

- RUP (**Rational Unified Process**)



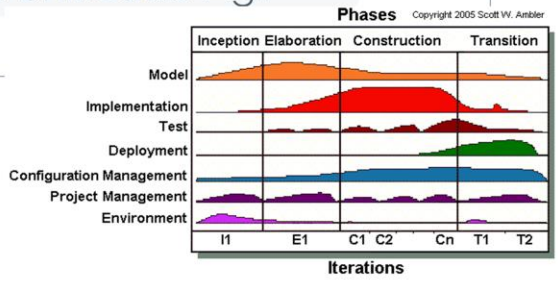
ftp://public.dhe.ibm.com//software/pdf/br/RUP_DS.pdf

Primeira Variação do RUP...

Agile Unified Process

Outros processos, baseados no RUP, procuraram incorporar princípios ágeis, visando aproximar o RUP às necessidades de desenvolvimento mais atuais.

- AUP - *Agile Unified Process*, ou
- PUÁgil – Processo Unificado Ágil



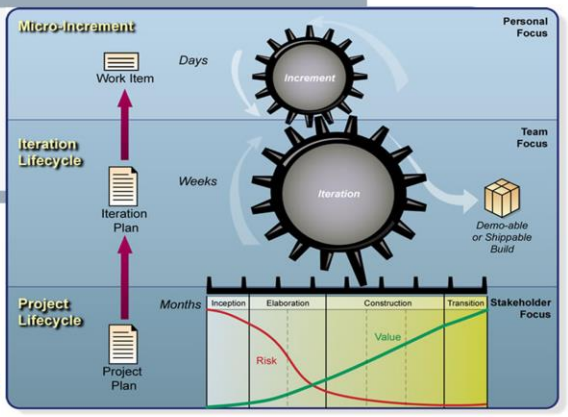
http://www.ambysoft.com/unifiedprocess/agileUP.html
http://www.craiglarman.com/wiki/index.php?title=Main_Page

Evoluindo mais um pouco...

OpenUP

Mais uma iniciativa nesse sentido...

- OpenUp



http://epf.eclipse.org/wikis/openup/

https://www.ibm.com/developerworks/br/rational/local/open_up/index.html?ca=dat

Considerações Finais

Considerações Finais

- Nessa aula, foi apresentada a atividade de modelagem de requisitos funcionais e não-funcionais com base em artefatos de abordagens mais tradicionais. No caso, focou-se em:
  - Casos de Uso
    - Diagrama, e
    - Especificação com fluxos principal, alternativos e de exceção.
  - Especificação Suplementar, usando o FURPS+ como orientação.

- Continuem os estudos!

Referências

Referências

Bibliografia Básica

1. **[Ebrary] Young, Ralph. Requirements Engineering Handbook. Norwood, US: Artech House Books, 2003.**

2. **[Open Access] Leite, Julio Cesar Sampaio do Prado. Livro Vivo - Engenharia de Requisitos. http://livrodeengenhariaderequisitos.blogspot.com.br/ (último acesso: 2017)**

3. **[Ebrary] Chemuturi, Murali. Mastering Software Quality Assurance : Best Practices, Tools and Technique for Software Developers. Ft. Lauderdale, US: J. Ross Publishing Inc., 2010.**

4. Software & Systems Requirements Engineering: In Practice - Brian Berenbach, Daniel Paulish, Juergen Kazmeier, Arnold Rudorfer (Livro bem completo mas, não tem exemplar físico na biblioteca, nem mesmo consta na Ebrary)

5. Requirements Engineering and Management for Software Development Projects - Murali Chemuturi (Livro bem completo mas, não tem exemplar físico na biblioteca, nem mesmo consta na Ebrary)

Referências

Bibliografia Complementar

1. [BIBLIOTECA – 15 exemplares] Pfleeger, Shari Lawrence. Engenharia de Software: Teoria e Prática. 2ª. Edição. São Paulo: Prentice Hall, c2004. xix, 535 p. ISBN 978858791831

2. [BIBLIOTECA – 3 exemplares] Withall, Stephen. Software Requirement Patterns. Redmond: Microsoft Press, c2007. xvi, 366 p. ISBN 978735623989.

3. [BIBLIOTECA - vários exemplares] Leffingwell, 2011, Agile Software Requirements, http://www.scaledagileframework.com/ (último acesso: 2017)

4. [Ebrary] Evans, Isabel. Achieving Software Quality Through Teamwork. Norwood, US: Artech House Books, 2004.

5. [Ebrary] Yu, Eric, Giorgini, Paolo, and Maiden, Neil, eds. Cooperative Information Systems: Social Modeling for Requirements Engineering. Cambridge, US: MIT Press, 2010.

6. [Open Access] Slides disponíveis em: https://www.wou.edu/~eltonm/Marketing/PP%20Slides/ (último acesso: 2017)


UnB
GAMA
Dúvidas?
Orientações?
Sugestões?
FIM
mileneserrano@unb.br ou mileneserrano@gmail.com
serrano@unb.br ou serr.mau@gmail.com