

Elicitação

Modelagem

Análise

# Requisitos – Aula 04

Professores: Milene Serrano e Maurício Serrano

## Agenda

- › Considerações Iniciais
- › Pré-Rastreabilidade
  - RichPicture
  - Argumentação
- › Considerações Finais

# Considerações Iniciais

## Considerações Iniciais

Antes de focarmos em Pré-Rastreabilidade, cabe definir:  
O que é Rastreabilidade?

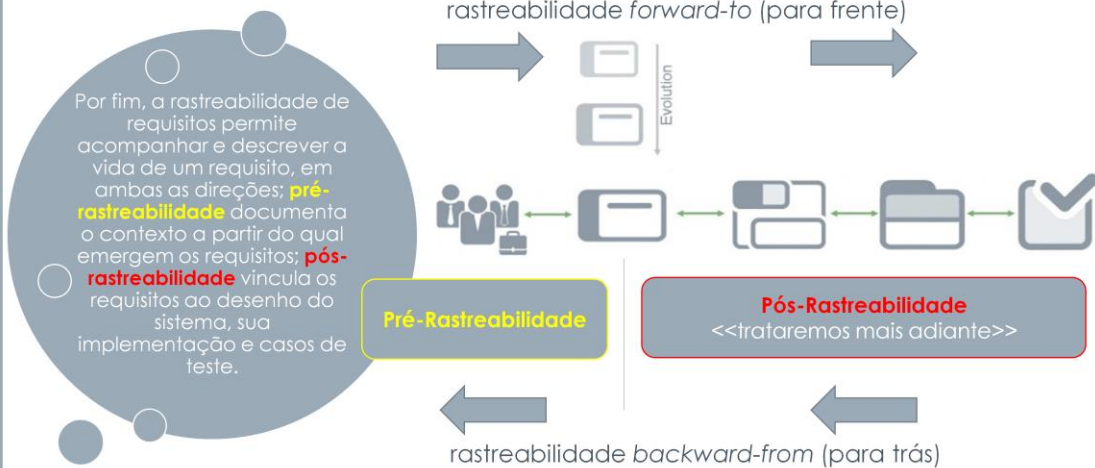
*"Um requisito é rastreável se é possível descobrir : Quem sugeriu o requisito (a fonte)?; Por que o requisito existe (rationale)? Que outros requisitos estão relacionados a ele (dependência entre requisitos)?, e Como o requisito se relaciona com outras informações, tais como desenho do sistema, implementação e documentação do usuário? Fonte: adaptado de Sommerville (1998)*



Por que é necessária?

Requisitos não podem ser efetivamente gerenciados sem rastreabilidade. A rastreabilidade, portanto, pode ser obtida por um conjunto de elos ou ligações (*links*) entre requisitos inter-relacionados, entre requisitos e suas fontes, e entre requisitos e os componentes que os implementam.

## Considerações Iniciais

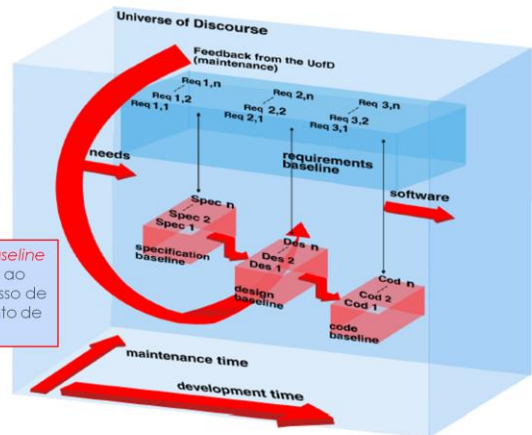


## Considerações Iniciais

A **baseline** de requisitos, construída durante a Engenharia de Requisitos, evolui durante o processo de desenvolvimento. Portanto, alterações devem ser incorporadas à **baseline** para que esta reflita sempre o conjunto de requisitos estabelecido entre clientes e desenvolvedores.

O modelo - registrado na **baseline** de requisitos - deve evoluir em consonância com o sistema e o Universo de Informações. A **baseline** é referência para todo o processo de desenvolvimento, guiando a equipe nas várias atividades relacionadas à geração do sistema desejado.

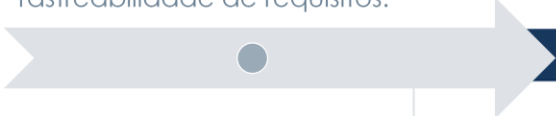
Evolução da **Baseline** de Requisitos ao longo do Processo de Desenvolvimento de Software



Uma das principais fontes geradoras de alterações é o próprio conjunto de interessados, que inicialmente não tem muita clareza nos objetivos a alcançar, e à medida que o desenvolvedor evolui o produto de software vai descobrindo novas possibilidades ou funcionalidades a incluir no projeto.

## Considerações Iniciais

Uma forma de registrar esses rastros é construir uma matriz de rastreabilidade de requisitos.



Matriz de rastreabilidade de requisitos - versão 1.1					
	Funcionalidade 1	Funcionalidade 2	Funcionalidade 3	Funcionalidade 4	
Requisito 1	x				
Requisito 2		x	x		
Requisito 3				x	
Requisito 4				x	
Projeto A					

# Considerações Iniciais

## FOCO: Pré-Rastreabilidade?

- Nosso olhar estará voltado em como manter o rastro nos primeiros capítulos da Engenharia de Requisitos.

## Duas preocupações chave:

- **Reuniões longas**, em domínio novo, com interessados expondo suas necessidades e seus desejos sobre o produto de software a ser desenvolvido.
- **Resolução de requisitos em conflito**, principalmente, nos primeiros passos da Engenharia de Requisitos. É comum que diferentes representantes do cliente ou usuário – interessados – interajam com a equipe de desenvolvimento. Essa interação gera requisitos, os quais podem ser conflitantes entre os interessados e/ou entre o desejado e o possível computacionalmente falando.
- **Portanto, são necessários recursos capazes de auxiliar nesses primeiros passos da Engenharia de Requisitos bem como na resolução desses conflitos.**





# Pré-Rastreabilidade

RichPicture

## RichPicture

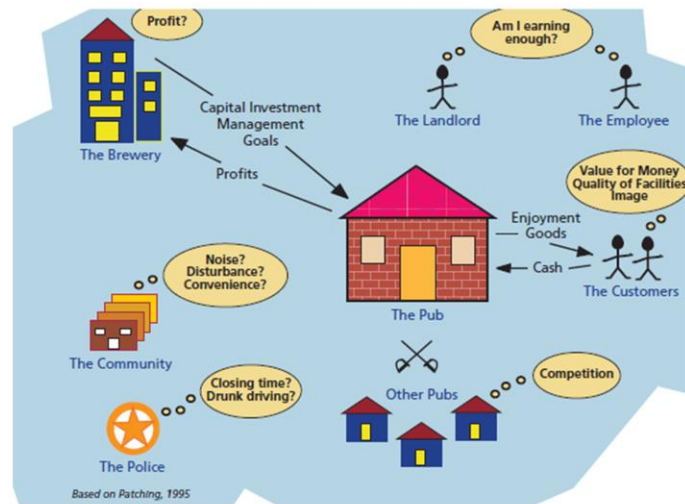
Trata-se de uma modelagem cuja notação permite analisar problemas e expressar ideias.

É um modelo informal, bem fácil de entender, e que pode ser construído colaborativamente com o cliente.

Pode auxiliar na identificação: de processos de negócio e seus requisitos; de atores envolvidos nos processos de negócio e suas responsabilidades; dos relacionamentos entre processos e atores, e de potenciais problemas e conflitos.

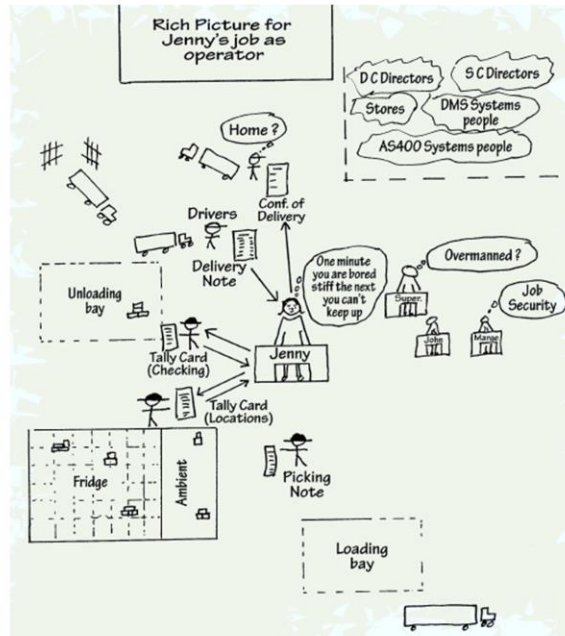
Para desenhar, basta começar com um problema central, bem no centro de uma página em branco, e acrescentar tópicos relacionados no entorno dessa problemática.

## RichPicture



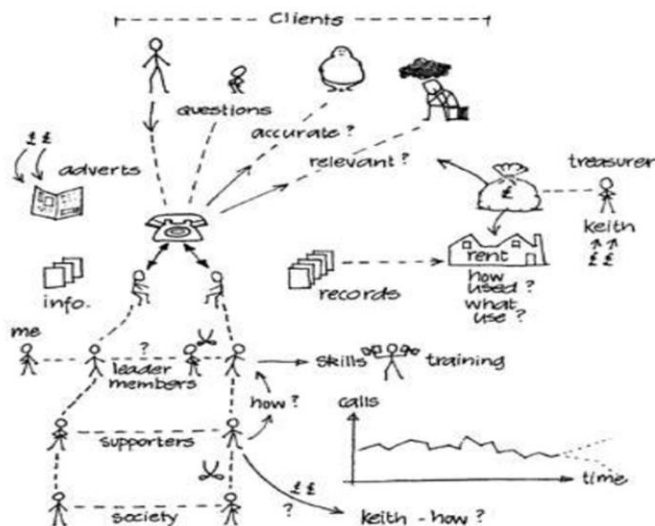
Exemplo I

# RichPicture



Exemplo II

## RichPicture










Nem todo RichPicture é bem feito. Muitas vezes, parecem incompletos, pois não são plenamente compreendidos.

Exemplo V – Ruim - Incompleto

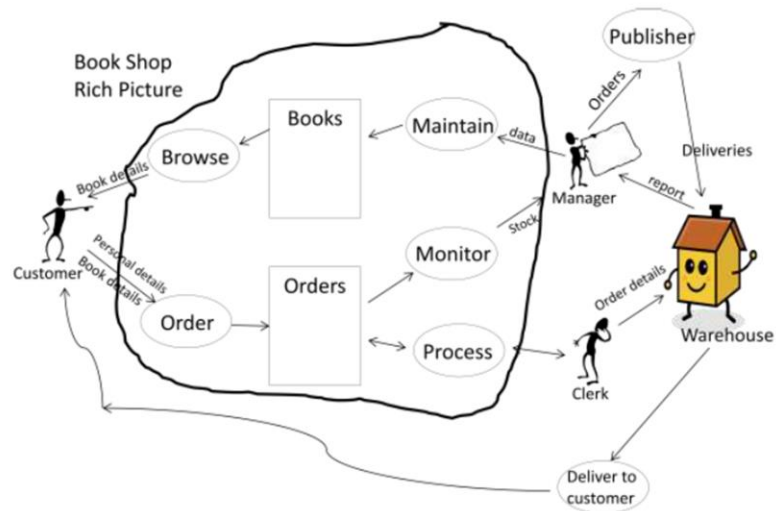
# RichPicture



Alguns componentes, para começar...

Rich Picture Components	Comments
<b>Actors (with descriptive labels)</b>  Manager  Clerk	Actors are the users of your system. An actor may also represent a group of users; e.g., one manager plus five data clerks will still show two actors. An actor may carry out any number of operations. Represented graphically as matchstick people.
<b>Operations (also known as processes or functions)</b> 	Operations specify what the system does. Each operation is executed either by an actor or another operation. Represented graphically as circles or ovals, with a descriptive label inside.
<b>Data stores (also known as tables)</b> 	Data stores are essentially the tables in your database or files in the system. It is also necessary to show the type of data they contain. Only operations may read from or write to data stores. Represented graphically as rectangles.
<b>Arrows</b>  User details  Confirmation	Arrows show the direction of data (or information) flow amongst actors, data stores and operations. Arrows may cross the system boundary (see below). Represented graphically as single-headed arrows. Descriptive labels indicate the nature of the data or information flowing.
<b>System boundary</b> (usually a solid line But may also be dashed) 	The system boundary identifies those operations that you are responsible for (i.e., your area of responsibility), which means that your system must carry out everything that is inside the system boundary. You can ignore what is outside. Represented graphically as a circular line. Normally, this is the last thing you should add to your rich picture.

# RichPicture



Exemplo III

# RichPicture

Frequentemente utilizado em tempo de reunião, e juntamente com as técnicas:

- **Brainstorming**, desenhando RichPictures "on the fly"
- **Storyboarding**, desenhando os fluxos de atividades dos usuários para que sejam revistos, avaliados e refinados.
- **Paper-based Prototyping**, para representar early design no desenvolvimento de produtos de software.





# RichPicture



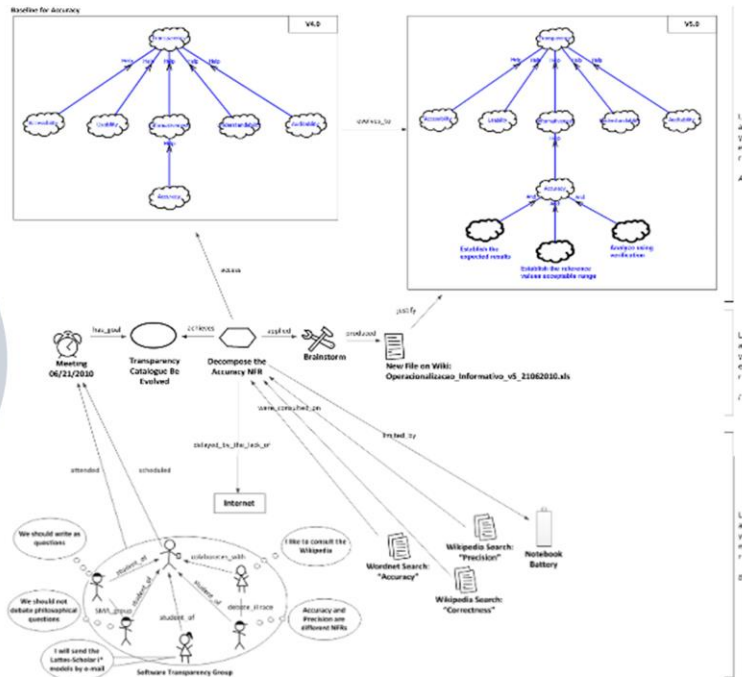
## Elementos para um RichPicture efetivo

Table 1. Elements of an Effective Rich Picture	
Element	Comment
1. Include <i>structure</i>	Include only enough structure to allow you to record the process and concerns. The latter requires that all the people who will use or could conceivably be affected by the introduction of the new system be included.
2. Include <i>process</i>	Do not attempt to record all the intricacies of process; a broad brush approach is usually all that is needed
3. Include <i>concerns</i>	Caricature the concern in a thought bubble (see Figures 1-3 for examples). A fuller explanation may be provided in a supplementary document
4. Use the language of the people depicted in it	This will make the rich picture comprehensible to your informants
5. Use any pictorial or textual device that suits your purpose	There is no correct way of drawing a rich picture. There are as many styles as analysts and the same analyst will find different styles useful in different situations

Apesar do slide conferir algumas diretrizes ou sugestões, é interessante não engessar a forma de construção de um RichPicture. Trata-se de um desenho a mão livre.

# RichPicture

It trace rastreando  
o Requisito Não  
Funcional  
**Accuracy**,  
decomposto em  
outros Requisitos  
Não Funcionais



ITrace

## Similares ao RichPicture

5W2H. Consultem:

<https://sites.google.com/site/planejajaweb/5w2h>

Mapa Mental. Consultem:

<https://cepein.femane.com.br/BDigital/arqPIBIC/1011321081B451.pdf>

<http://periodicos.unesc.net/sulcomp/article/download/1035/979>



# Pré-Rastreabilidade

## Argumentação

# Argumentação



A argumentação é uma forma vital de cognição humana. Diariamente, pessoas são confrontadas com informações conflitantes e forçadas a lidar com situações inconsistentes (BESNARD; HUNTER, 2008).

O uso da argumentação representa uma atividade de comunicação essencial para a sociedade.

Existem diversas tecnologias que oferecem suporte a essa prática, tais como listas de e-mails, sistemas de suporte à tomada de decisões e sistemas de suporte à negociação (MOOR; AAKHUS, 2006).

Apesar da diversidade, as boas práticas de uma argumentação nem sempre estão presentes nessas tecnologias.

Geralmente estes mecanismos desencorajam o debate e facilitam a argumentação de baixa qualidade e o pensamento devoluto (BEX et al., 2013).

Fonte: Cavalcante, A. C. A. ACE-CAST: Uma Ferramenta de Apoio à Argumentação Colaborativa. TCC, UnB - Campus Gama, 99 páginas, 2014.

# Argumentação



Segundo Rahwan (2005), a argumentação pode ser vista como uma interação social baseada em princípios.

Ela é composta de argumentos incompatíveis e almeja chegar a uma conclusão consistente e racional.

Uma das principais metas da argumentação é a resolução de pontos de vistas controversos.

Estes pontos de vistas devem ser justificáveis ou refutáveis, dependendo da informação disponibilizada.

Isto distingue a argumentação do raciocínio dedutivo clássico, em que o acréscimo de novas premissas não influencia uma prova previamente definida.

Fonte: Cavalcante, A. C. A. ACE-CAST: Uma Ferramenta de Apoio à Argumentação Colaborativa. TCC, UnB - Campus Gama, 99 páginas, 2014.

# Argumentação



Já Besnard e Hunter (2008) definem a argumentação como sendo um processo em que argumentos e contra-argumentos são construídos e manipulados.

A manipulação dos argumentos envolve comparação, avaliação e julgamento da aceitabilidade dos mesmos com base em critérios definidos.

Fonte: Cavalcante, A. C. A. ACE-CAST: Uma Ferramenta de Apoio à Argumentação Colaborativa. TCC, UnB - Campus Gama, 99 páginas, 2014.

# Argumentação

Em (EEMEREN; GROOTENDORST; HENKEMANS, 1996), argumentação é uma atividade verbal, social e racional.

É verbal devido à necessidade de representação dos argumentos em uma linguagem ordinária, seja ela oral ou escrita. Meios não verbais como gestos e expressões faciais podem ser considerados, porém não substituem as expressões verbais.

É social devido à participação de diversos interlocutores expressando suas ideias, considerando os pontos de vistas alheios e tomando decisões com base nessas informações.

É racional, pois o argumento apresentado por um interlocutor possui uma percepção racional a partir da posição que o mesmo defende.



Fonte: Cavalcante, A. C. A. ACE-CAST: Uma Ferramenta de Apoio à Argumentação Colaborativa. TCC, UnB - Campus Gama, 99 páginas, 2014.



# Argumentação

Atualmente, os modelos de argumentação têm sido utilizados em diversas áreas tais como gestão do conhecimento, elaboração de provas e teoremas, inteligência artificial, lógica de programação, sistemas jurídicos, sistemas de tomada de decisões e negociações (BENTAHAR; MOULIN; BÉLANGER, 2010).

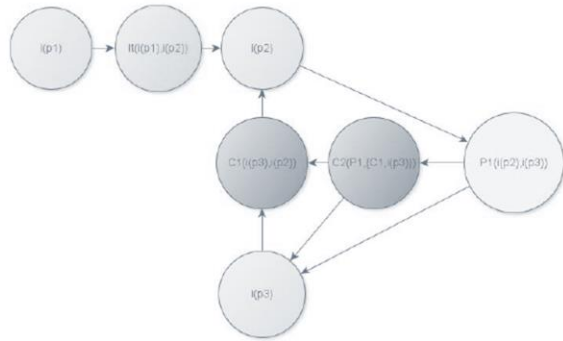
Devido ao caráter multidisciplinar da argumentação, diversas abordagens foram propostas nos últimos 60 anos. Iremos focar em uma dessas abordagens, no caso:

- ACE (JURETA; MYLOPOULOS; FAULKNER, 2009)

ACE - Acceptability Evaluation Framework ou Framework de Avaliação da Aceitabilidade

# Argumentação

ACE



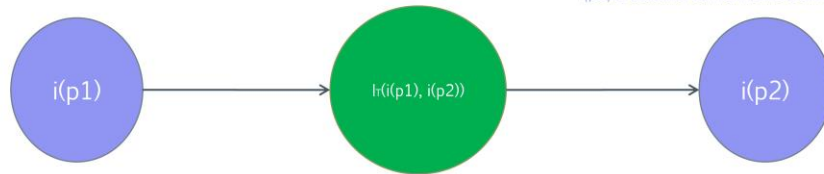
- framework de argumentação utilizado na engenharia de requisitos. O objetivo deste modelo é verificar a **validação relativa** dos artefatos discutidos em uma reunião envolvendo os interessados e os engenheiros de requisitos. O ACE consiste de uma linguagem para representar informações obtidas a partir de uma discussão; uma condição de aceitabilidade que denota uma decisão comum dos participantes acerca de um artefato, e algoritmos para checar automaticamente a condição de aceitabilidade nas discussões.

Fonte: Cavalcante, A. C. A. ACE-CAST: Uma Ferramenta de Apoio à Argumentação Colaborativa. TCC, UnB - Campus Gama, 99 páginas, 2014.

**ACE** - *Acceptability Evaluation Framework* ou Framework de Avaliação da Aceitabilidade

# Argumentação

A proposição  $i(p2)$  foi inferida a partir de uma regra indutiva tendo como insumo  $i(p1)$ .  
Com base nas ideias expressas em  $i(p1)$ , pode-se inferir que o "bom custo-benefício representa um produto com qualidade e acessível no contexto da empresa que está enfrentando problemas financeiros.  
A relação de inferência entre as proposições  $i(p1)$  e  $i(p2)$  é descrita formalmente como  $li(i(p1), i(p2))$ .



O exemplo modela uma discussão envolvendo uma provável aquisição de notebooks por parte de uma empresa.

O grafo da discussão contém duas proposições  $i(p1)$  e  $i(p2)$ :

- $i(p1)$ : Comprar um notebook da marca X oferece um bom custo benefício. Isto é essencial para empresa que esta enfrentando dificuldades financeiras.
- $i(p2)$ : O notebook da marca nacional X possui preço acessível e boa qualidade em suas peças.

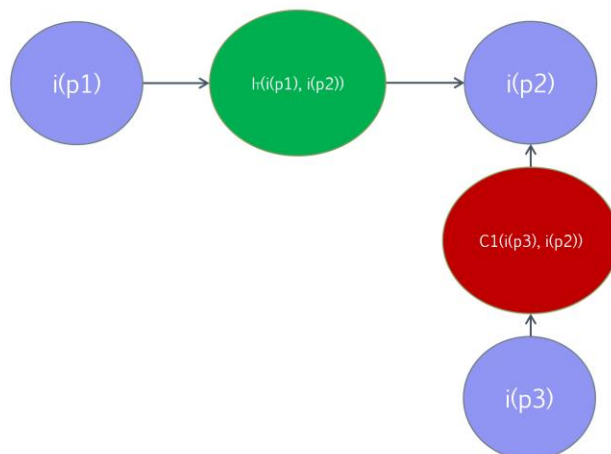
Fonte (adaptado): Cavalcante, A. C. A. ACE-CAST: Uma Ferramenta de Apoio à Argumentação Colaborativa. TCC, UnB - Campus Gama, 99 páginas, 2014.

# Argumentação

Fonte (adaptado): Cavalcante, A. C. A. ACE-CAST: Uma Ferramenta de Apoio à Argumentação Colaborativa. TCC, UnB - Campus Gama, 99 páginas, 2014.

No grafo da discussão, temos mais uma proposição,  $i(p3)$ :

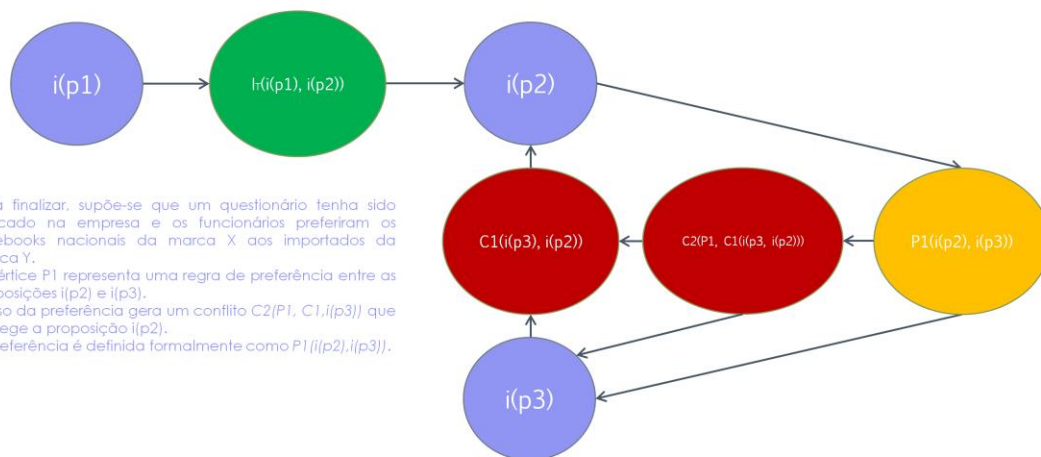
- $i(p3)$ : Alguns notebooks nacionais da marca X apresentam defeitos após um ano de uso. É interessante adquirir notebooks da marca internacional Y. Apesar de mais caros, a qualidade do produto compensa a diferença de preços.



A proposição  $i(p3)$  contrapõe as ideias apresentadas em  $i(p1)$ , exibindo um argumento contra a aquisição de notebooks de marcas nacionais. A relação de conflito envolvendo  $i(p3)$  e  $i(p2)$  é definida formalmente como  $C1(i(p3), i(p2))$ .

# Argumentação

Fonte (adaptado): Cavalcante, A. C. A. ACE-CAST: Uma Ferramenta de Apoio à Argumentação Colaborativa. TCC, UnB - Campus Gama, 99 páginas, 2014.



Para finalizar, supõe-se que um questionário tenha sido aplicado na empresa e os funcionários preferiram os notebooks nacionais da marca X aos importados da marca Y.  
O vértice  $P1$  representa uma regra de preferência entre as proposições  $i(p2)$  e  $i(p3)$ .  
O uso da preferência gera um conflito  $C2(P1, C1(i(p3), i(p2)))$  que protege a proposição  $i(p2)$ .  
A preferência é definida formalmente como  $P1(i(p2), i(p3))$ .

# Argumentação

Os modelos de argumentação, gerados com base na linguagem contida no ACE, são grafos direcionados com rótulos. Os vértices são classificados com base em quatro rótulos: i,  $I_i$ , P e C.

- Os vértices com rótulo (i) representam vértices de informação que servem de entrada ou saída para inferências ( $I_i$ ).
  - Os vértices com o rótulo ( $I_i$ ) representam a aplicação de inferências a fim de sustentar argumentos. O argumento inferido é uma conclusão lógica do primeiro.
  - Os vértices com rótulo (C) representam regras de conflito envolvendo dois ou mais vértices em um grafo.
  - Os vértices com rótulo (P) representam regras de preferência envolvendo a predileção de dois ou mais vértices do grafo.
- \* As arestas do grafo possuem apenas um rótulo ( $T_o$ ) <<omitido no grafo>>.

Fonte (adaptado): Cavalcante, A. C. A. ACE-CAST: Uma Ferramenta de Apoio à Argumentação Colaborativa. TCC, UnB - Campus Gama, 99 páginas, 2014.



## Considerações Finais

## Considerações Finais

- › Nessa aula, foi apresentado o tópico Rastreabilidade de Requisitos, mais especificamente, Pré-Rastreabilidade de Requisitos.
- › Adicionalmente, foram acordados dois recursos que auxiliam as demanda de Pré-Rastreabilidade de Requisitos. São eles:
  - RichPicture, e
  - Argumentação.
- › 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)



Dúvidas?

Orientações?

Sugestões?

FIM

[mileneserrano@unb.br](mailto:mileneserrano@unb.br) ou [mileneserrano@gmail.com](mailto:mileneserrano@gmail.com)  
[serrano@unb.br](mailto:serrano@unb.br) ou [serr.mau@gmail.com](mailto:serr.mau@gmail.com)