ICMC USP SÃO CARLOS

Entendimento do problema: comunicação com *stakeholders* e Engenharia de Requisitos

Disciplina: Engenharia de Software- SSC5764
Prof. José Carlos Maldonado
Palestrante: Cristiane Aparecida Lana

Agenda

- Introdução
- Modelos de Processos de Software Tradicional
- Engenharia de Requisitos para Software Tradicional
- Sistema-de-Sistemas (SoS)
- Processos de Desenvolvimento de SoS
- Engenharia de Requisitos para SoS
- Considerações Finais

Introdução

- Evolução tecnológica dos sistemas computacionais (hardware e software) aumenta as funcionalidades implementadas por software, criando sistemas de software mais complexo.

Introdução

Estudos ressaltam:

Falta de compreensão do negócio ou domínio pelo analista de sistema

+

Mal entendimento da finalidade do sistema

+

Falhas de comunicação entre os envolvidos

=

Requisitos incorretos, incompletos ou conflitantes

Fonte: Capers (1996), Charette (2005), Jun et al., (2011) e Sommerville (2011)

Introdução

"Nenhuma outra parte causa tantos danos, nenhuma outra é mais difícil de corrigir."

F. Brooks, *No Silver Bullet: Essence and Accidents of Software Engineering*, IEEE Computer, vol 20(4):10-19, april,1987

Modelos de Processos Tradicionais

- Desenvolver software é geralmente uma tarefa complexa e sujeita a erros.
- Sucesso ou fracasso dependem de inúmeros fatores que ocorrem durante todo o processo
- Necessidade de estabelecer processos sistemáticos para desenvolvimento

↓

Modelos de Processo de Software.

Modelos de Processos Tradicionais

- **Representação simplificada do processo de software**

www.cti.gov.br

Modelos de Processos Tradicionais

- **Processos de desenvolvimento de software visam assegurar :**

1. prazos e necessidade de recursos definidos
2. elevada produtividade (de forma econômica)
3. qualidade

Modelos de Processos Tradicionais

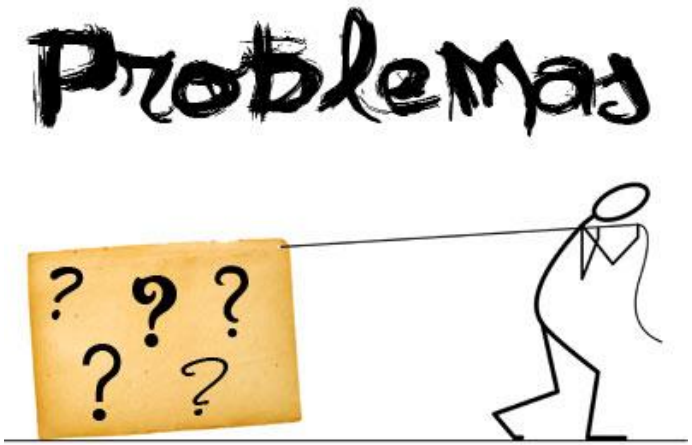
- Quais problemas podem ocorrer se não utilizar Processo de Desenvolvimento?

1. Procedimentos não documentados.
2. Erros são cometidos repetidamente.
3. Dificuldade de prever cronogramas e orçamentos.
4. Alto índice de defeitos, retrabalho e desperdício.
5. Dificuldade de implementar boas práticas e lições aprendidas.
6. Dificuldade de realizar ações para prevenções de defeitos.

Modelos de Processos Tradicionais

- **Exemplos de Modelos de Processos de Software**
  - Modelo em Cascata
  - Modelo de Prototipagem
  - Iterativo e Incremental
  - Modelo de Métodos Formais
  - Processo Unificado
  - Metodologias ágeis

Modelos de Processos Tradicionais



Análise e definição de requisitos.

Manutenção

- Ocorre um distanciamento do MUNDO REAL e um mergulho na TECNOLOGIA e SOLUÇÕES POSSÍVEIS.

Engenharia de Requisitos de Software

O que é Requisito?

- Uma descrição do que o sistema deve fazer, de como ele deve se comportar, bem como das suas restrições de operação.

- É importante ressaltar que os requisitos descrevem "o que o sistema deve fazer“ - e também "o que ele não deve fazer“.

Classificação de Requisitos

- Funcionais:
  - Descrevem o comportamento do sistema, ou seja o que o sistema deve fazer.

B - REQUISITOS FUNCIONAIS

B1 - Lançamentos diversos

1. O sistema deve permitir a inclusão, alteração e remoção de *leitores* da biblioteca, com os seguintes atributos: nome, endereço, cidade, estado, telefone, email, documento de identificação, categoria de leitor e data de nascimento.

2. O sistema deve permitir a inclusão, alteração e remoção das diversas *categorias de leitores*, com os seguintes atributos: código da categoria, descrição da categoria e número máximo de dias que essa categoria de leitor pode emprestar uma obra. Exemplos de categorias de leitores são: aluno de graduação, aluno de pós-graduação, professor, funcionário e usuário externo.

Classificação de Requisitos

- Não-Funcionais:
  - Descrevem o como deve ser feito. Em geral se relacionam com atributos de qualidades como: confiabilidade, usabilidade, etc.

C - REQUISITOS NÃO FUNCIONAIS

C1. Confiabilidade

14. O sistema deve ter capacidade para recuperar os dados perdidos da última operação que realizou em caso de falha.

15. O sistema deve fornecer facilidades para a realização de *backups* dos arquivos do sistema.

16. O sistema deve possuir senhas de acesso e identificação para diferentes tipos de usuários: administrador do sistema, funcionários da biblioteca e leitores que têm acesso ao sistema na biblioteca (em quiosques especiais).

Prioridade de Requisitos

- Requisitos podem ser vistos em três classes distintas
  - Essenciais/Permanentes
  - Importantes
  - Desejáveis

- Em princípio, deve ser abordado os requisitos de essenciais para desejáveis

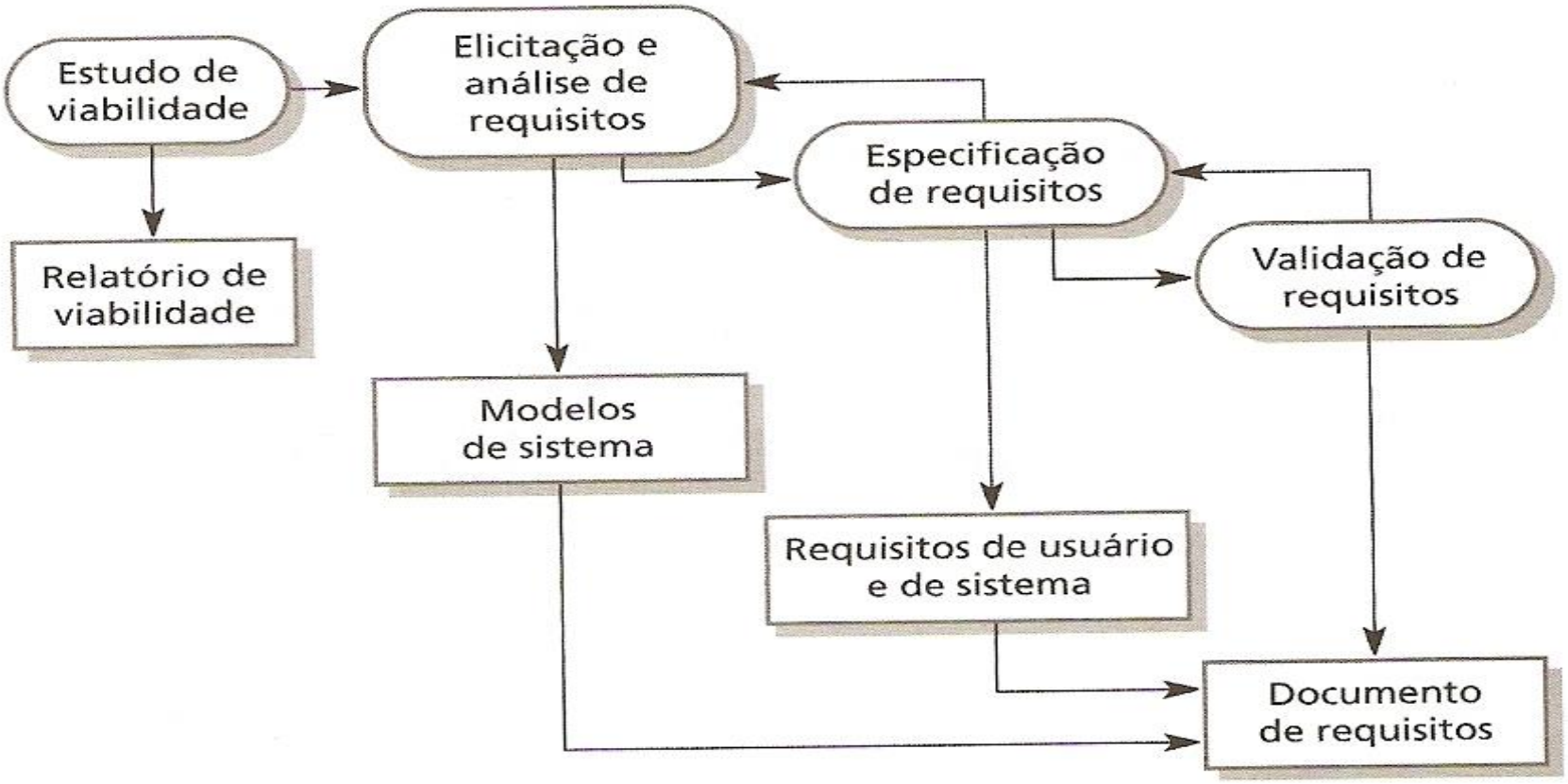
Prioridade de Requisitos

F1 - Módulo visualizador de conteúdo	Oculto( )			
Descrição: O sistema deve possuir um módulo leitor que permite a leitura dos arquivos manipulados pelo sistema.				
Requisitos não funcionais				
Nome	**Restrição**	**Categoria**	**Desejável**	**Permanente**
NF1.1 - Controle de acesso	O visualizador só deve permitir a visualização para o usuário se ele estiver autorizado.	Segurança	( )	(X)
NF1.2 - Sincronizar leitura	O visualizador deve iniciar a leitura da obra a partir da ultima página acessada.	Usabilidade	(X)	(X)
NF1.3 - Alterar tamanho de fonte	O visualizador deve permitir que o usuário altere tamanho da fonte dos caracteres.	Usabilidade	( )	(X)
NF1.4 - Aplicar zoom	O sistema deve permitir o uso de zoom nos textos e imagens visualizados.	Usabilidade	( )	(X)
NF1.5 - Alterar tema do visualizador	O sistema deve permitir que o usuário escolha temas de leitura das obras, como diurno, noturno e sépia.	Configurabilidade	(X)	(X)
NF1.6 - Decodificar arquivos	O visualizador deve efetuar a decodificação das obras baixadas e abri-las de forma correta.	Usabilidade	( )	(X)

Importância da Engenharia de Requisitos

- A Engenharia de Requisitos é a *espinha dorsal* de todo o processo de desenvolvimento de software.
  - Base de todo o desenvolvimento
- Ela contém atividades para determinar os *objetivos* do software e as *restrições* atreladas a elas. *(Requisitos)*
- Ela requer fundamentação e *processos próprios*, e que devem ser planejados e gerenciados ao longo de todo o ciclo de vida.
  - Estudo de viabilidade
  - Elicitação e Análise de Requisitos
  - Especificação de Requisitos
  - Validação de requisitos

Tarefas do Processos de Requisitos



Sommerville, 2011

Estudo de Viabilidade

- Atividade breve para responder
  - Em que o sistema contribui?
  - Pode ser implementado na tecnologia atual?
  - Restrições de prazo e custos
  - Pode ser integrado com outros sistemas?

– Atividade da fase de concepção
  - Produz PESw: Proposta de Especificação de Software

Elicitação e Análise de Requisitos

- É formada por obtenção de requisitos.

  - Interação com os *stakeholders* para coletar requisitos
    - Entrevistas
    - Observação *in loco*
    - Encontros *(reuniões com stakeholders)*
    - Análise de documentos

  - Classificação e organização de requisitos
    - Agrupa requisitos relacionados

Elicitação e Análise de Requisitos

- Priorização e negociação de requisitos.
  - Vários *stakeholders* => requisitos conflitantes
- Requisitos são modelados
  - UML (caso de Uso, DSS (diagrama de sequência)).
- Documentação de Requisitos
  - Documentação dos requisitos levantados, analisados, especificados, priorizados e negociados.

Elicitação de Requisitos

Adaptado de http://www.inf.ufes.br/~falbo/files/Notas_Aula_Engenharia_Requisitos.pdf

Especificação de Requisitos

- A especificação de requisitos é a *descrição sistemática e abstrata* do que o software deve fazer, a partir daquilo que foi analisado.

- Ela apresenta a especificação e a *caracterização do problema* a ser resolvido pelo software:
  - O que o software vai fazer?
  - Como ele irá fazer?
  - Determinar os critérios de verificação e validação. ***(o que deve ser testado?)***

- Durante a especificação de requisitos é gerado o *Documento de Requisitos* ou *Documento de Especificação de Requisitos*.
  - Oferece a concordância entre *stakeholders* sobre o que deve ser desenvolvido

Especificação de Requisitos

- Cada desenvolvedor utiliza um modelo específico para elaborar o *Documento de requisitos*.
  - Existem diversas modelos disponíveis na literatura (PRESSMAN, 2006; WAZLAWICK, 2004; WAZLAWICK, 2010)
- O modelo proposto por Wazlawick (2004), sugere que o documento de requisitos tenha um *índice* contendo:
  - Requisitos funcionais numerados
  - Requisitos não-funcionais numerados
  - Requisitos suplementares numerados
- O corpo do documento deve conter o *detalhamento de cada requisito* do software.

Documento de Requisitos

- O documento de requisitos deve seguir alguns padrões para serem expressos em linguagens natural.

1. Iniciar com “O sistema deve...”.
2. Usar frases curtas.

C3. Portabilidade

19. O sistema deve ser executado em computadores Pentium 200mHz ou superior, com sistema operacional Windows 98 ou acima.

20. O sistema deve ser capaz de armazenar os dados em base de dados Oracle ou Sybase.

Documento de Requisitos

- O documento de requisitos deve seguir alguns padrões para serem expressos em linguagens natural.

3. Os requisitos devem estar organizados logicamente
   - Entrada -> processamento -> saída

4. Cada Requisito deve ter um identificar único
   - RF01, RNF01, 17,

5. Os requisitos devem estar classificados em RF e RNF.

Documento de Requisitos

- O documento de requisitos deve seguir alguns padrões para serem expressos em linguagens natural.

6. Decisões de projetos não devem ser tomadas durante a fase de desenvolvimento do documento de requisitos

7. Os requisitos não devem conter detalhes de implementação.
   - É importante não usar termos relacionados à implementação como: “arquivo”, “cadastro” e “menu”.

Documento de Requisitos

8. A explicação dos termos do domínio da aplicação não deve estar presente nos requisitos, devendo aparecer em um glossário do domínio de aplicação.

Glossário

Termo	Descrição	Sinônimos
Backup	Cópia de segurança ou cópia de salvaguarda	
Categoria de leitor	Categorias nas quais os leitores são divididos, para facilitar o tempo máximo em que podem tomar obras emprestadas. Por exemplo, professores, alunos, funcionários, etc.	
Categoria de obra literária	Categorias nas quais as obras são divididas, para facilitar o tempo máximo em que podem ficar emprestadas e o valor diferenciado de multa por dia de atraso. Por exemplo, livros, revistas, teses de doutorado, etc.	Categoria de obra
Empréstimo de obra literária	Ato de emprestar uma obra a um *leitor* por um determinado período de tempo, desde o empréstimo até a *devolução da obra*. Significa que o *leitor* levou consigo a *obra* e que a mesma não se encontra mais em posse da biblioteca.	Empréstimo de obra, Empréstimo
Devolução da obra literária	Fim do *empréstimo da obra*. Significa que a obra foi devolvida à biblioteca pelo *leitor*.	Devolução de obra, Devolução
Funcionário	Pessoa que trabalha na biblioteca e faz a verificação do *empréstimo* e *devolução* de *obras* pelo *leitor*.	
Leitor	Pessoa que leva emprestada uma obra de uma biblioteca por um determinado período de tempo	
Multa	Valor a ser pago pelo *leitor* porque a *obra* foi devolvida após o prazo previsto para *devolução da obra*.	

Documento de Requisitos

- O documento de requisitos deve seguir alguns padrões para serem expressos em linguagens natural.

9. Manter consistência no uso dos termos do domínio de aplicação.

10. Problemas que podem ocorrer: falta de clareza; confusão entre requisitos, Fusão de requisitos, ambiguidade, omissão de requisitos, e requisitos oculto (i*).

Documento de Requisitos

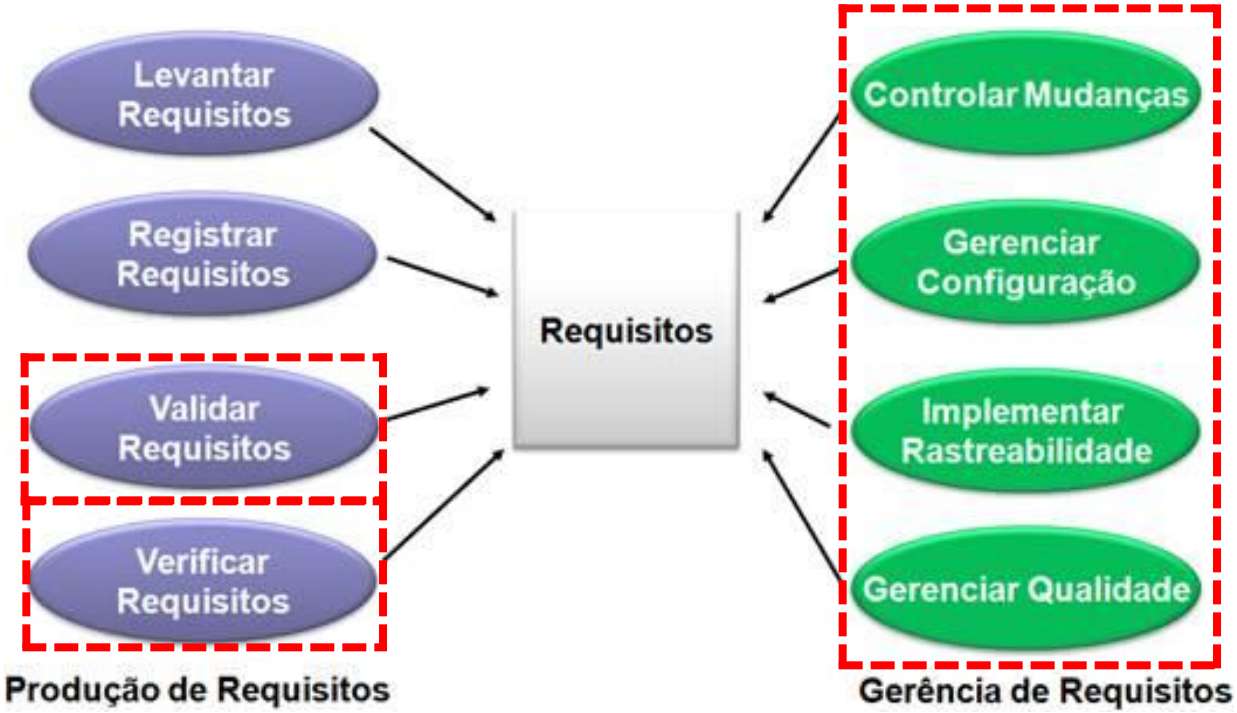
- O documento de requisitos deve seguir alguns padrões para serem expressos em linguagens natural.

11. Usar formato padronizado e padrões (ISO/IEC/IEEE 29148:2011)

12. Usar destaques no texto (cor, itálico, negrito)

13. Evitar o uso de jargão (termos, palavras) de informática
    - Deletar, debugar, printar.

Engenharia de Requisitos



Validação de Requisitos

- Tem por objetivo mostrar que o sistema realmente define o sistema que o usuário deseja.
  - Será que realmente entendemos o que o cliente deseja?
  - Devemos certificar de que não houve falha na interação. (comunicação)

- Há diversas técnicas de validação

Validação de Requisitos

- O que deve ser feito?
  - Verificação da validade
  - Verificação da Consistência
    - Sem conflitos
  - Verificação de completude
    - Requisitos definem o sistema como um todos

Validação de Requisitos

- O que deve ser feito?

  - Verificação de realismo
    - É possível fazer considerando prazo e custo? O requisito pode ser implementado?
  - Facilidade de verificação
    - Linguagem padronizada de escrita de requisitos
    - Caso de teste

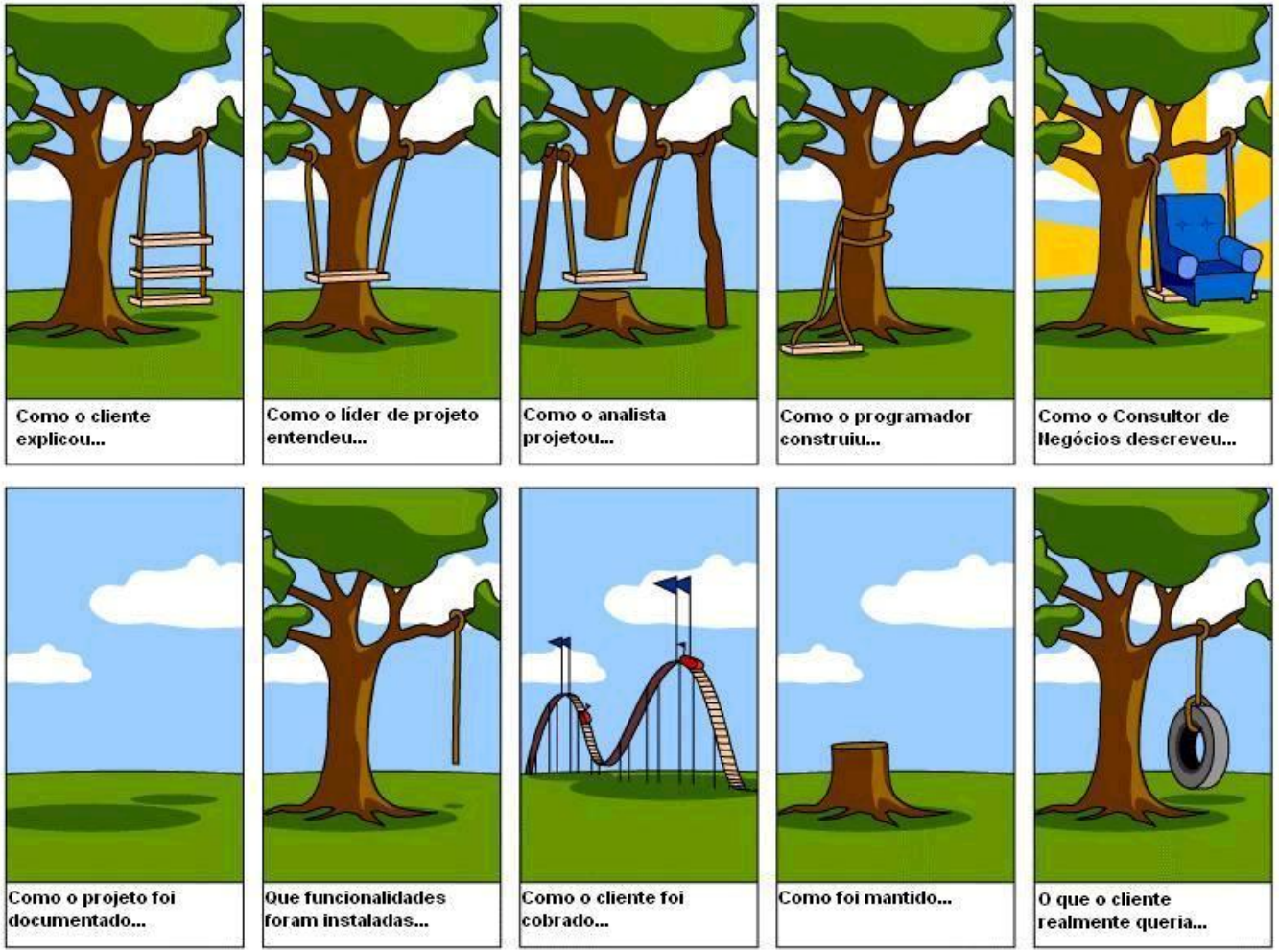
Técnicas de Validação de Requisitos

- Revisões de Requisitos
  - Análise manual sistemática dos requisitos
- Prototipação
  - Uso de modelo executável do sistema para avaliar requisitos
- Geração de Casos de Teste
  - Desenvolver testes específicos para avaliar os requisitos
- Análise de Consistência Automática
  - Avaliar a especificação dos requisitos

Importância da Especificação

- Não importa qução bem projetado ou codificado está o programa, se ele for mal analisado e especificado desapontará o usuário e trará aborrecimentos ao desenvolvedor

Engenharia de Requisitos para Software Tradicional



Desafios dos Requisitos

- Um dos grandes desafios da Engenharia de Requisitos são os Requisitos Instáveis.
- Geralmente clientes e usuários não entendem o problema real que desejam solucionar com o software a ser desenvolvido.
- Deve-se trazer o cliente para dentro do processo de desenvolvimento:
  - XP
  - SCRUM
- Requisitos não podem ser engessados e também não podem mudar descontroladamente.

Desafios dos Requisitos

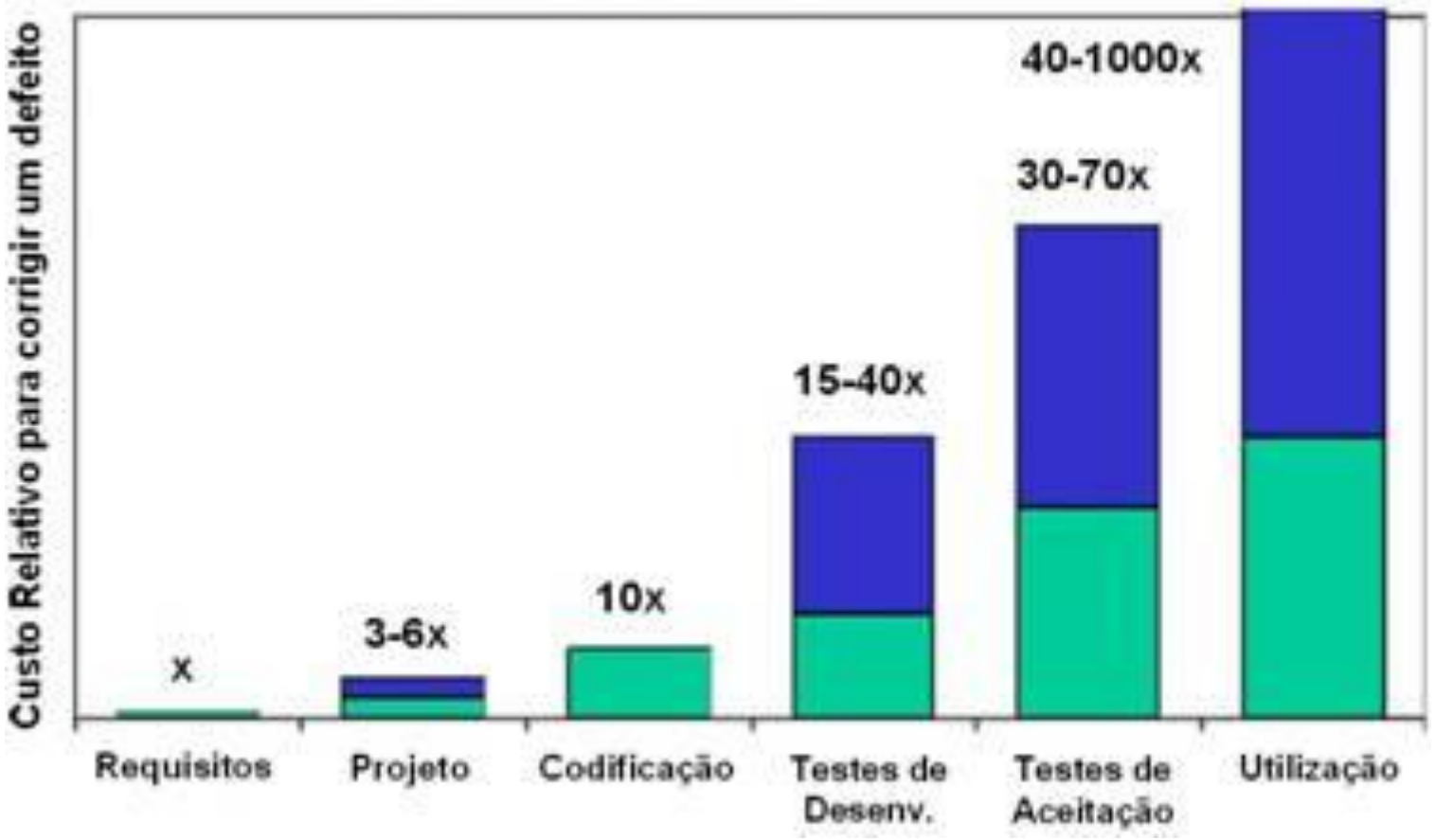
- Desenvolvimento de software *baseado em processos* ajuda no controle de requisitos instáveis.

- Processo de *requisição de mudanças* bem definido deve ser utilizado:
  - Viabilidade da mudança
  - Impactos da mudança

- Outro grande desafio da Engenharia de Requisitos é a *identificação rápida de erros* de requisitos.
  - Quanto mais cedo, menos custoso.

- Processos de desenvolvimento de software que seguem *metodologias iterativas* auxiliam na identificação rápida de erros de requisitos.

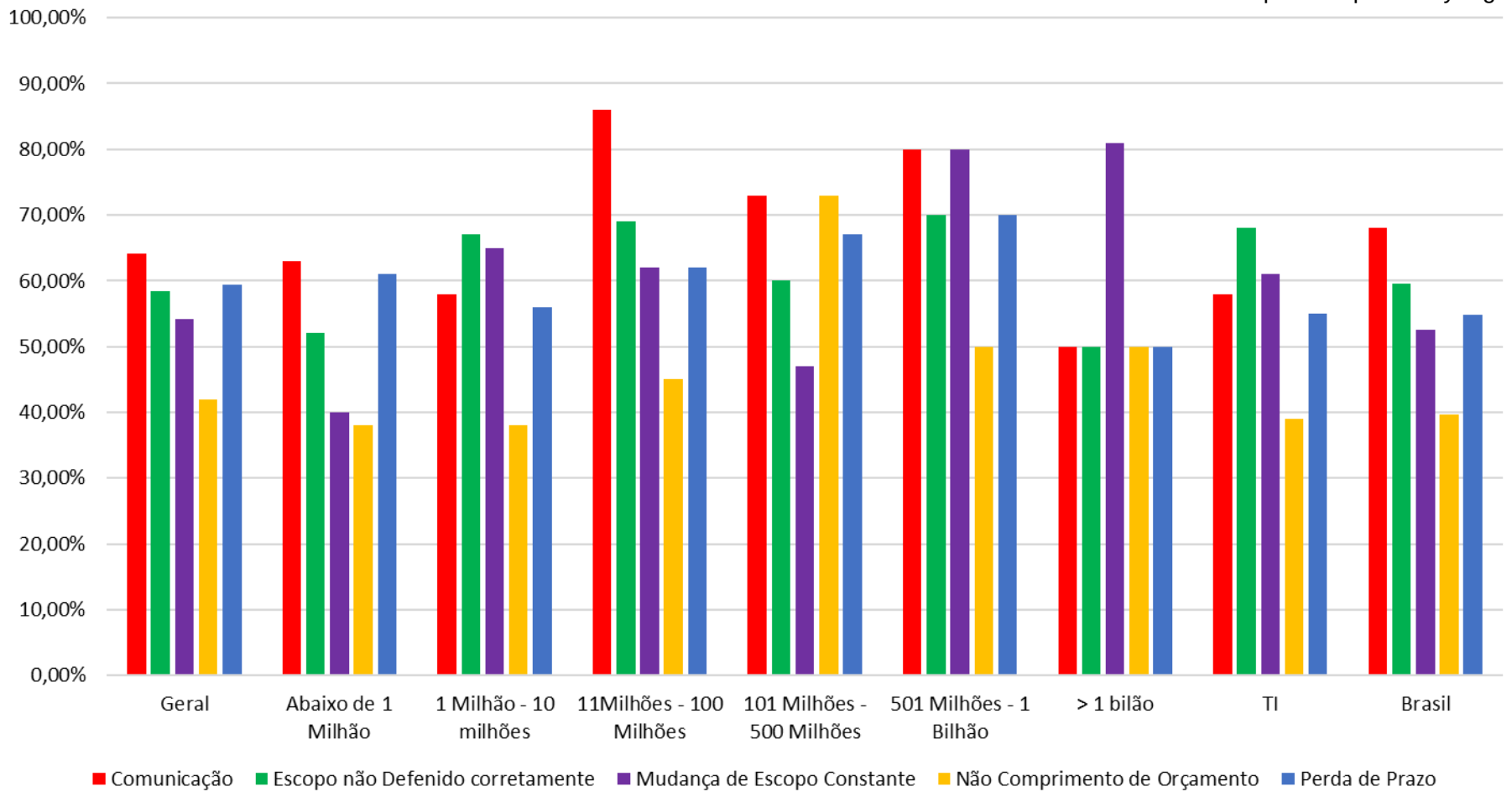
Desafios dos Requisitos



www.devmedia.com.br

Maiores Problemas em GP - 2014

http://www.pmsurvey.org/



Problemas da Engenharia de Requisitos

- **Problemas com a comunicação na fase de requisitos:**

Problemas da Engenharia de Requisitos

- **Quando não é feito corretamente:**

Sistema-de-Sistemas - SoS

Sistema-de-Sistemas

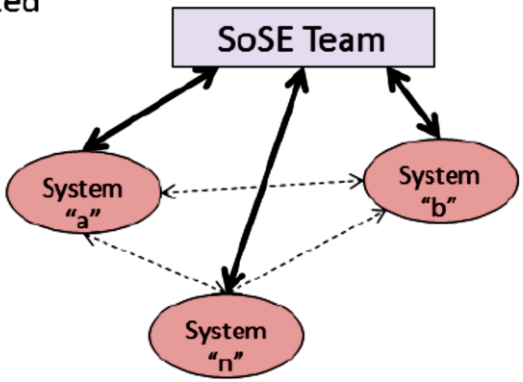
- **O que é:**
  - Systems of Systems ou simplesmente SoS.
  - São sistemas compostos por outros sistemas.
  - A grande maioria são sistemas críticos.
  - Requerem alto nível de qualidade.
  - Podem utilizar diferentes tecnologias e plataformas.
  - Apresentam desenvolvimento evolucionário (em tempo de execução).

Características de Sistemas-de-Sistemas

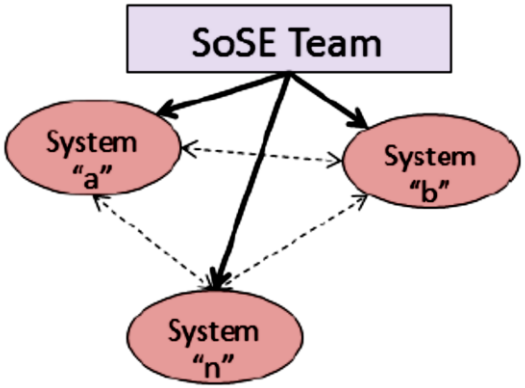
- ✓ Comportamento
  - Emergente;
  - Dinâmico;
- ✓ Interdependência
  - Operacional
  - Gerencial; e
- ✓ Interoperabilidade
- ✓ Desenvolvimento evolucionário
- ✓ Autônomos
- ✓ Distribuído Geograficamente

Topologia de SoS

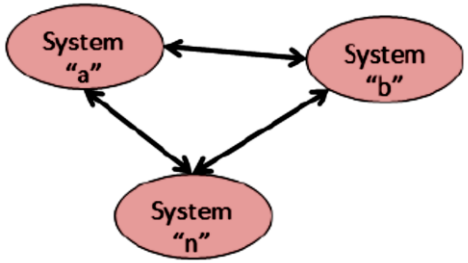
Directed



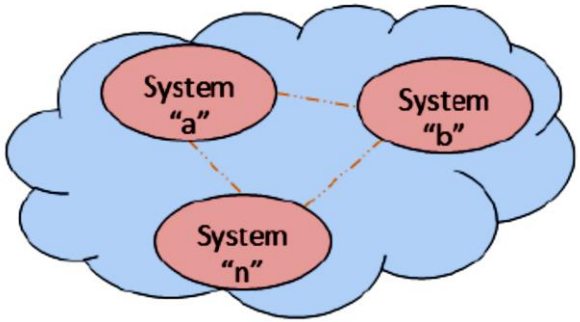
Acknowledged



Collaborative



Virtual



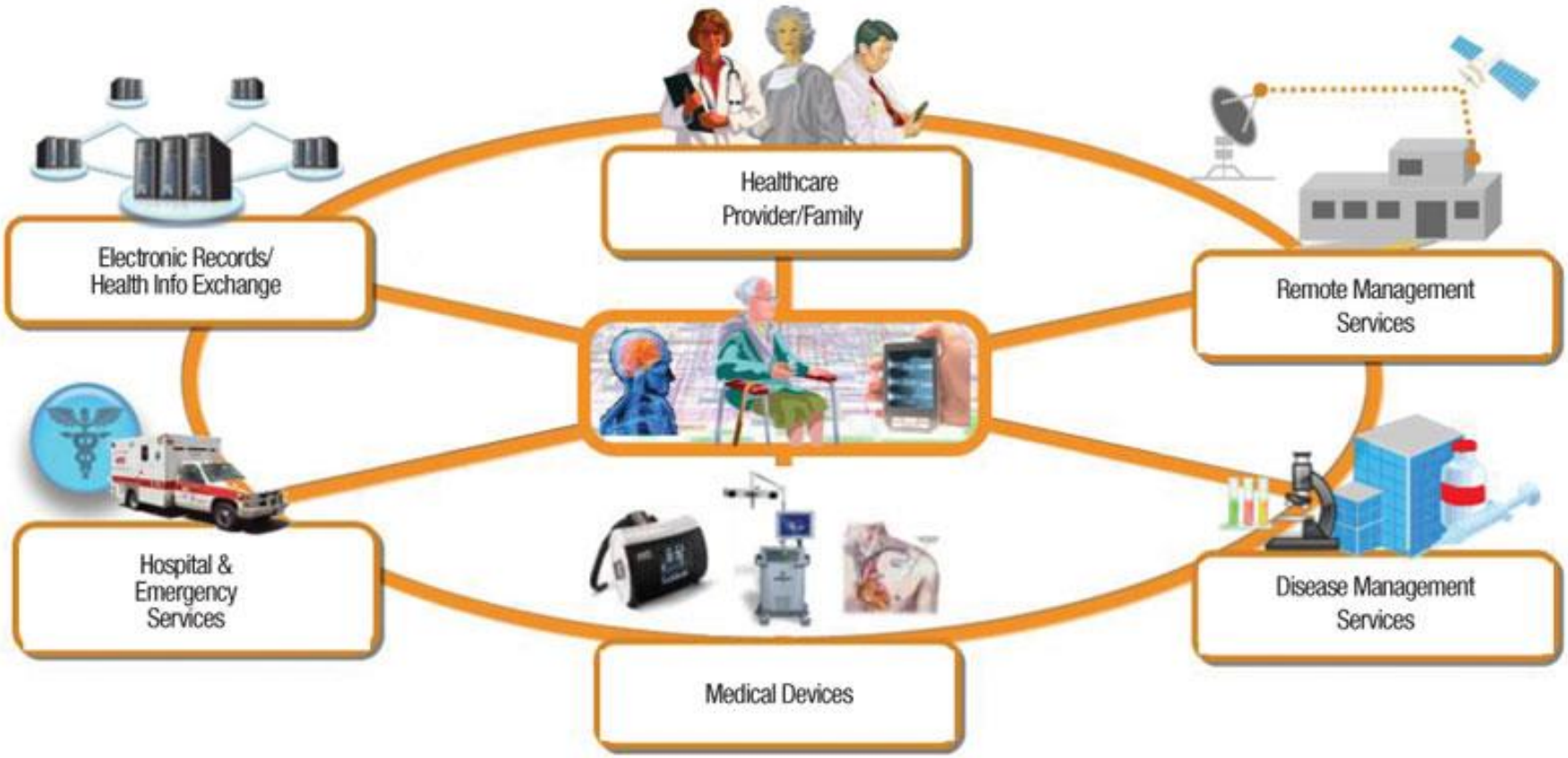
[Lane, 2013]

Desafios em SoS

- *Stakeholders* com interesses e prioridades concorrentes,
- Nenhuma autoridade centralizada sobre todos os sistemas,
- Aumento da complexidade devido a vários ciclos de vida do sistema,
- Características de comportamento, e
- Necessidades de desempenho entre os sistemas constituintes e do SOS

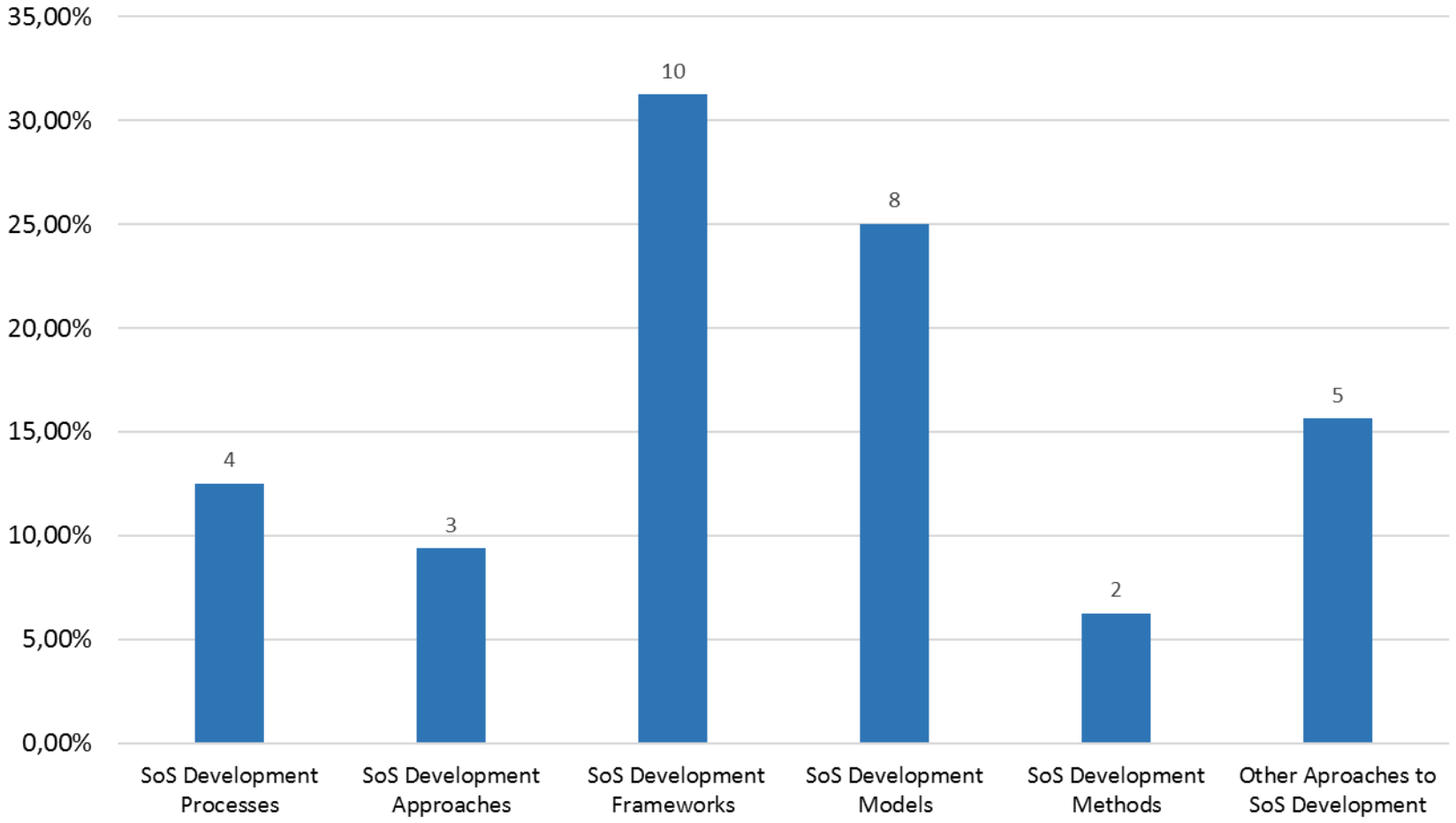
[Nielsen et al., 2015]

Exemplos de Sistemas-de-Sistemas

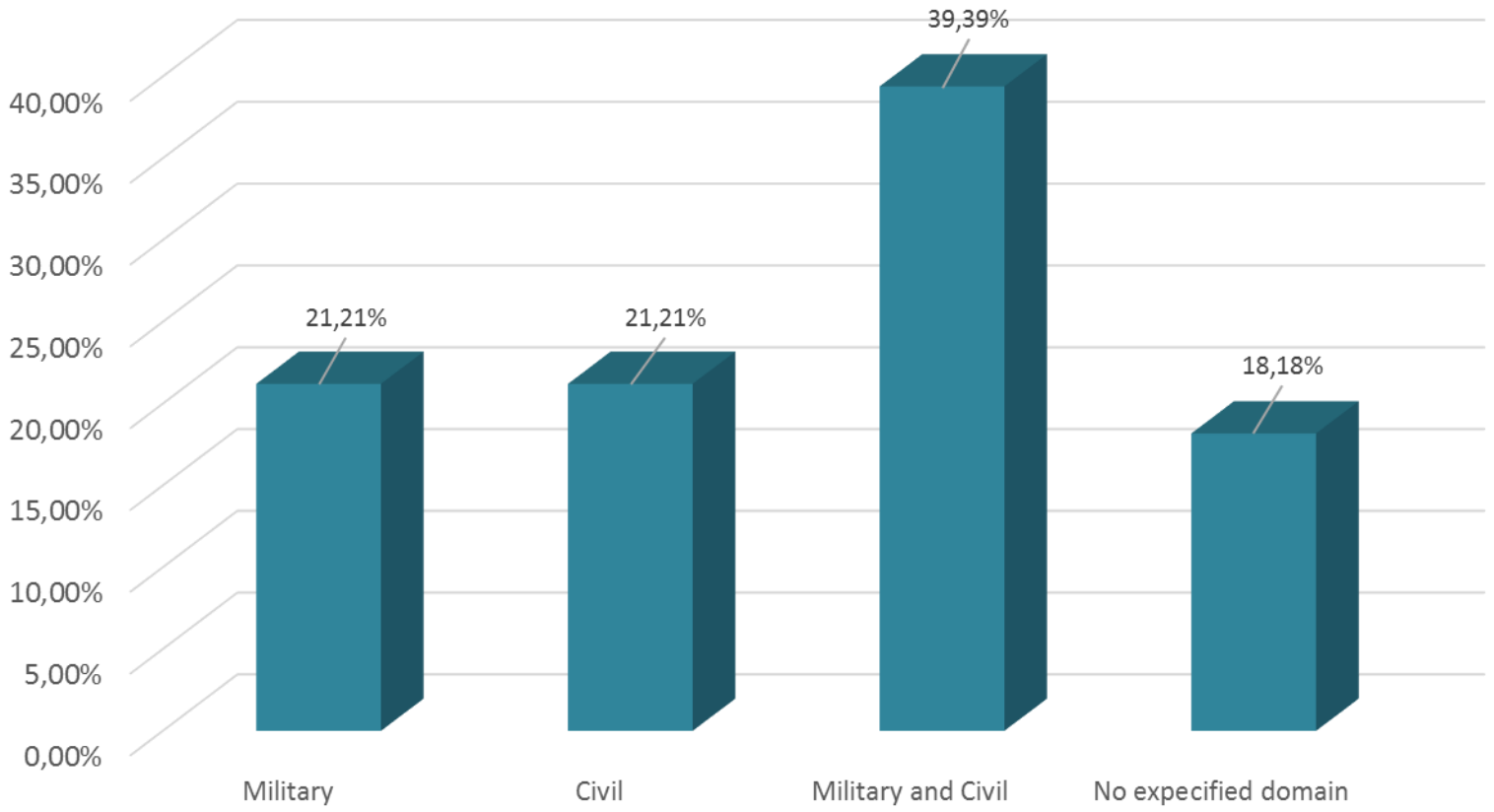


Exemplos de Sistemas-de-Sistemas

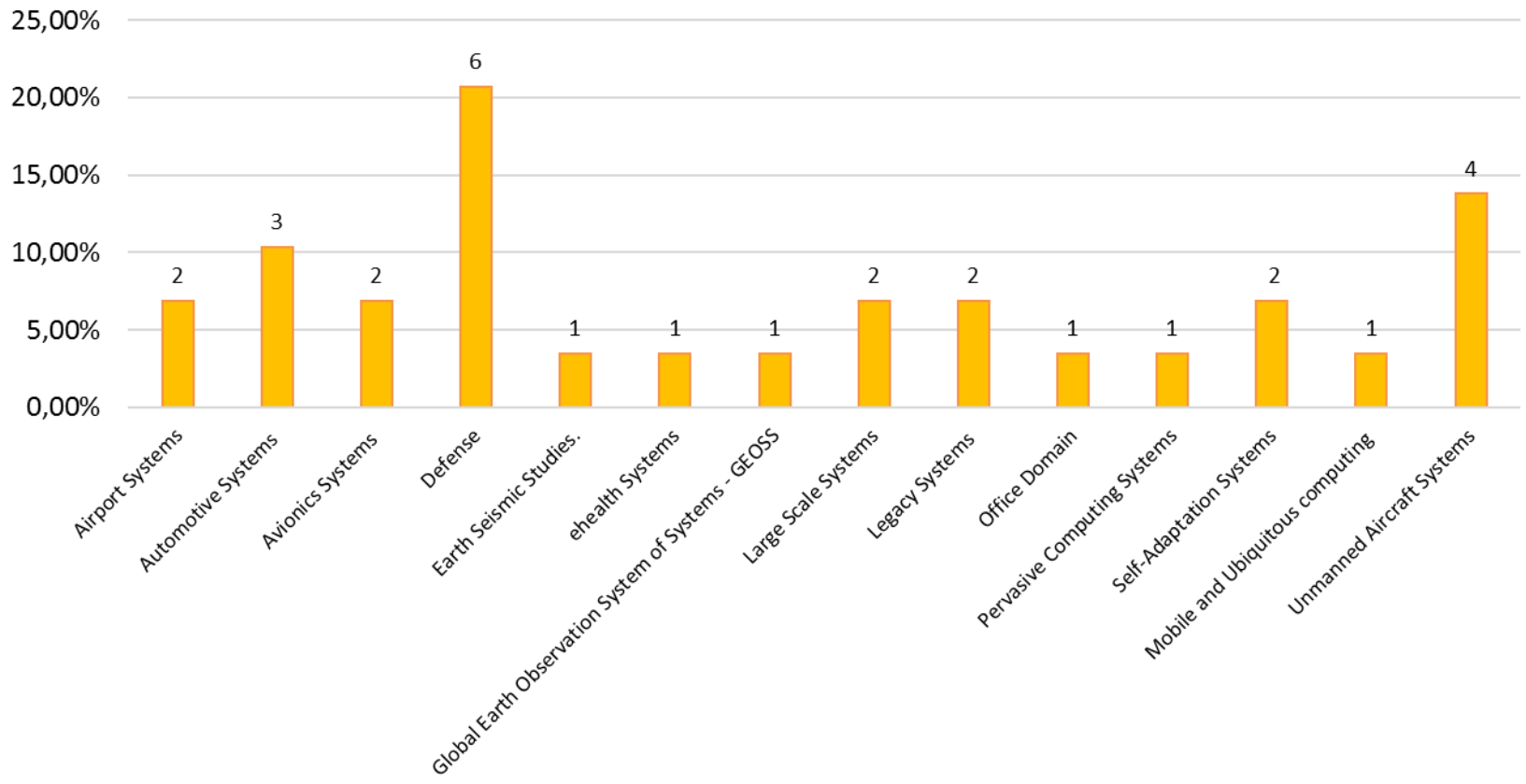
Processo de Desenvolvimento de SoS



Domínio dos Processos de Desenvolvimento de SoS



Processo de Desenvolvimento de SoS



Abordagens de ER para SoS (Lewis et al., 2009)

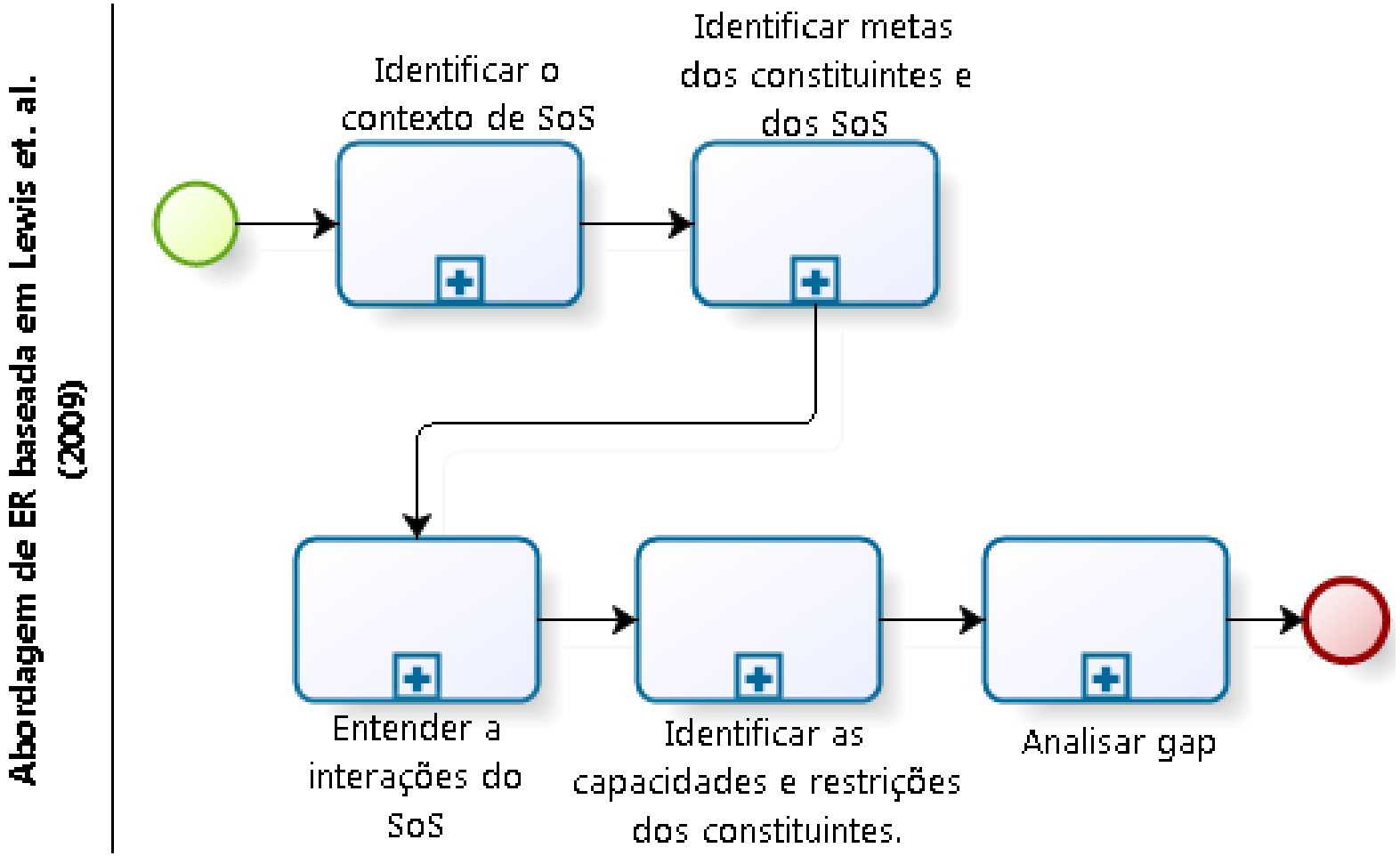
- Para ser bem sucedida deve usar duas abordagens:
  - **Top-down:** utilizada sozinha NÃO considera adequadamente as necessidades dos constituintes.
  - Incluem:
    - compreensão do tipo de SoS;
    - ambientes associados ao SoS;
    - sistemas constituintes;
    - identificação de objetivos; e
    - identificação da SoS interações.

Abordagens de ER para SoS (Lewis et al., 2009)

- **Bottom-up:** utilizada sozinha NÃO capturar as expectativas essenciais para o SOS como um todo.

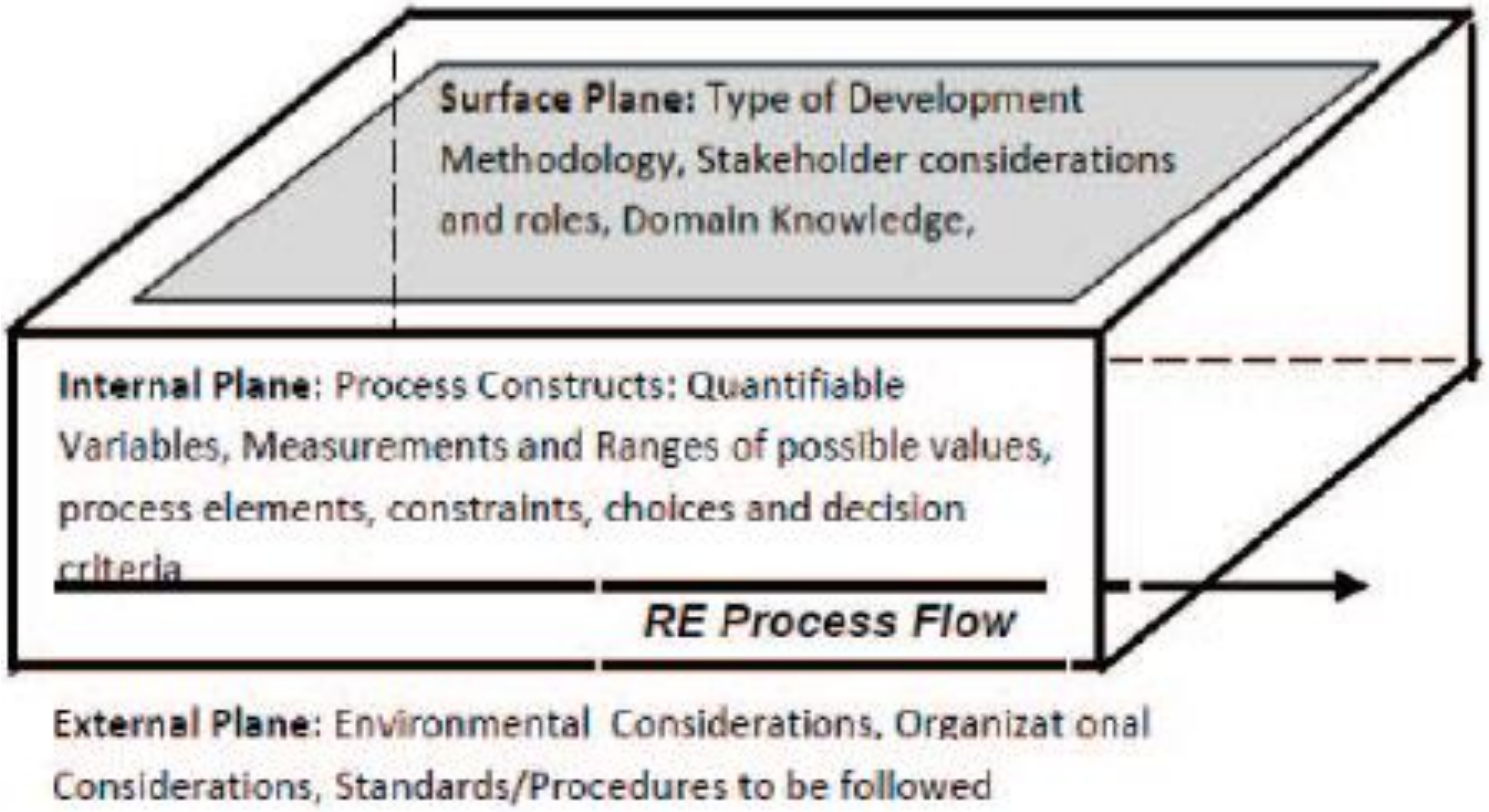
- Incluir:
  - compreensão das capacidades fornecidas pelos sistemas constituintes
  - restrições internas dos constituintes; e
  - externas impostas dos constituintes.

Abordagens de ER para SoS (Lewis et al., 2009)



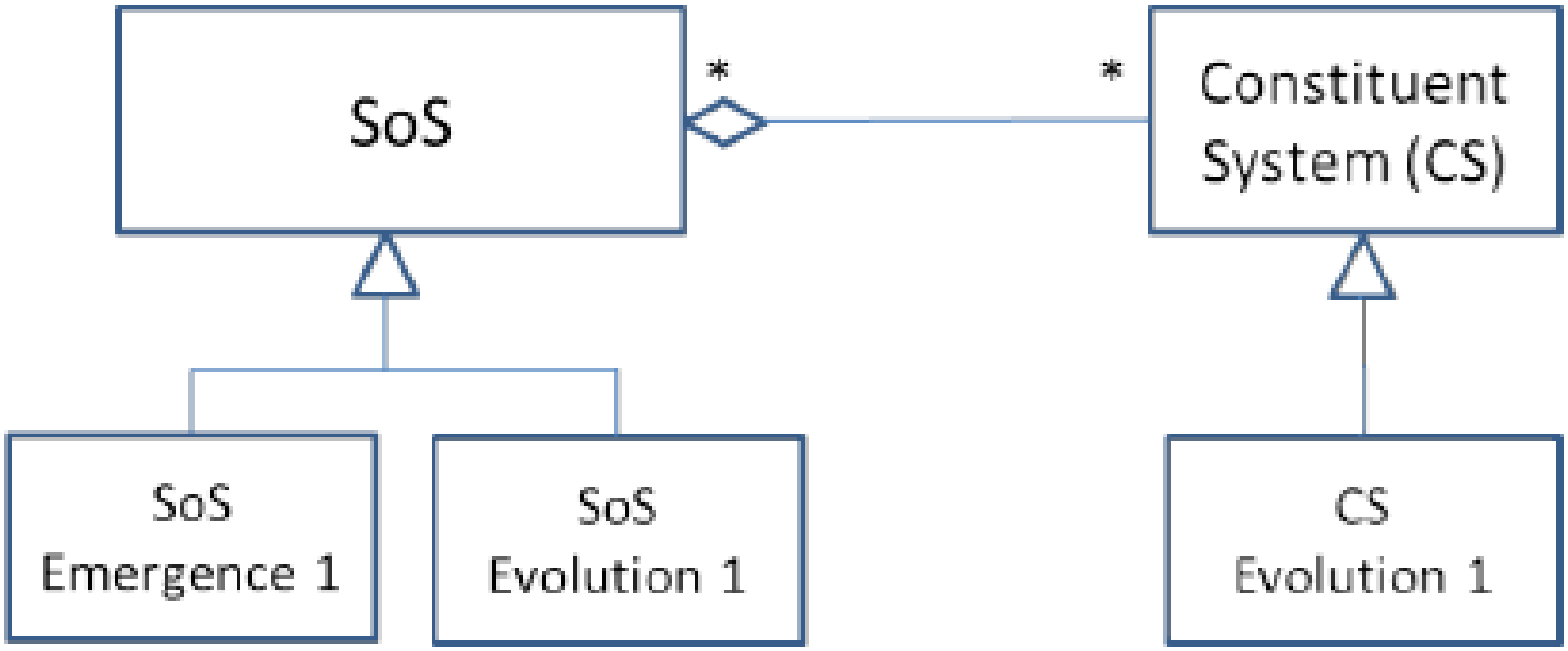
Abordagens de ER para SoS (Savio et al., 2011)

Uma visão multiplano do Modelo de processo genérico.



Abordagens de ER para SoS (Hallerstede et al., 2012)

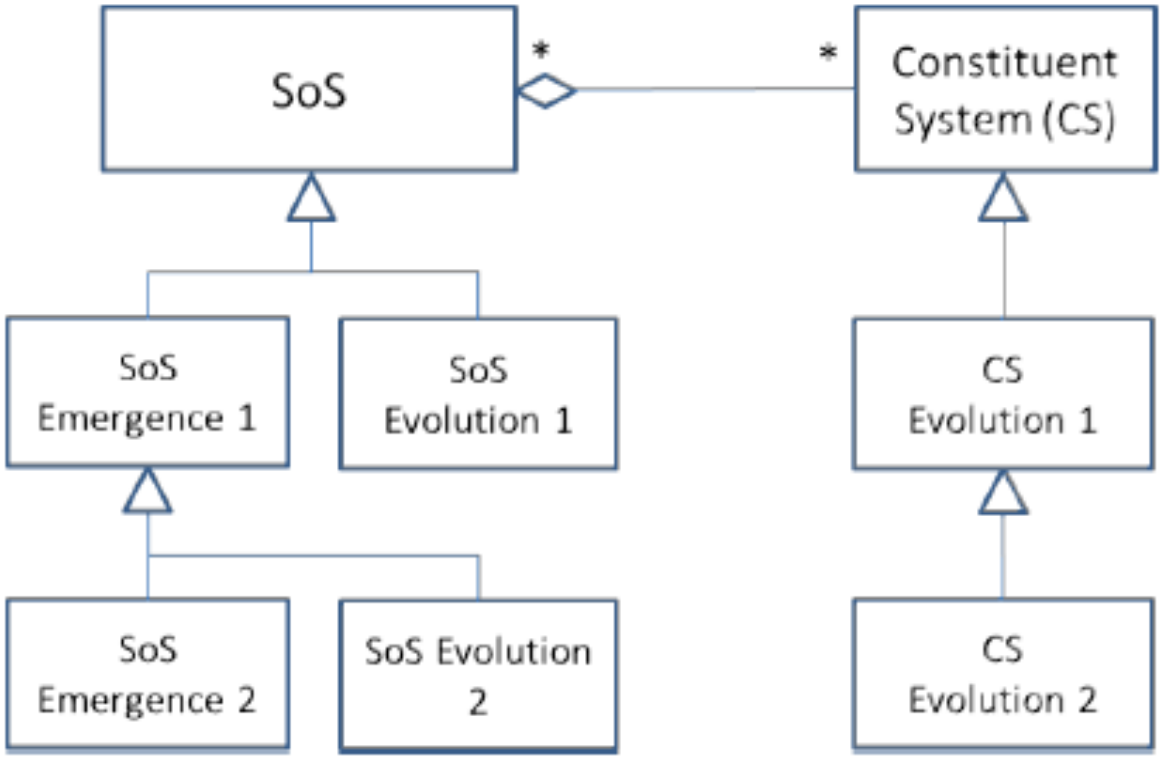
Corrobora com a abordagem de Lewis et. al. (2009), mas mostra



Os primeiros estágios do ciclo de vida de um Modelo de Sistema de SoS.

Abordagens de ER para SoS (Hallerstede et al., 2012)

Corrobora com a abordagem de Lewis et. al. (2009), mas mostra



Os últimos estágios do ciclo de vida de um Modelo de Sistema de SoS.

Desafios de ER para SoS (Lewis et al., 2009)

- **Escala**
  - Número de contexto;
  - Múltiplas interações entre os constituintes;
  - Grande número de *stakeholders.*

- **Multi-domínio**
  - Mistura domínios e os Constituintes podem pertencer a mais de um SoS.

- **Variado Contexto Operacional**
  - O mesmo sistema pode ter necessidades diferentes dependendo de onde for ser utilizado. Exemplo: EMR System.

Desafios de ER para SoS (Lewis et al., 2009)

- **Controle Descentralizado**
  - Controlado por diferentes organizações;
  - Geograficamente distribuído;
  - ER feita por colaboração de múltiplos *stakeholders* durante a longa e contínuo fase de desenvolvimento e evolução do SoS.

- **Rápida Evolução de Ambientes**
  - Modificação, inserção e exclusão de requisitos.

Desafios de ER para SoS (Lewis et al., 2009)

- **Execução contínua e muitas vezes desconectada das várias fases do ciclo de vida**
  - Os constituintes podem evoluir de forma não sincronizada.
  - **Tarefa mais difícil:** realizar análise de requisitos do SoS e entender quais capacidades estarão disponíveis (presente e futuro) e quais deixaram de existir.

- **Necessidades oportunistas para colaborar e integrar**
  - Surgem do processo de automação de novos processos ou de inserção de novos constituintes;

Iniciativas em SoS (Necub, 2011)

- **SoSEN: Rede de engenharia de SoS.**
  - Criado em 2010;
  - Estudar uma estratégia de pesquisa para SoS, focando ER, interoperabilidade, gerenciamento emergente, processo de desenvolvimento e educação.
  - Apoio de 33 acadêmicos de 24 universidade do Reino Unido
  - 13 líderes acadêmicos internacionais de 11 universidades, e
  - 12 grandes organizações (MoD do Reino Unido, DoD, DoD Austrália, IBM, entre outras.

Iniciativas em SoS (Necub, 2011)

- **IEEE: IEEE Comitê Técnico de SoS.**
  - Criar Frameworks para apoiar novas pesquisas, e
  - Facilitar interação entre academia e indústria.

- **INCOSE: Rede de conhecimento em SoS.**
  - Possui vinculo com a IEEE
  - Aumentar o número de especialistas que podem contribuir com o domínio, principalmente quando ligado a indústria.

Iniciativas em SoS (Necub, 2011)

- **ISoSE: Consorcio Internacional em SoSE.**
  - Criado em 2007;
  - Estudam uma forma de amenizar o problema de interoperabilidade dos SoS.
- **US-SoSEN: Sistemas de rede de SoSE dos Estados Unidos.**
  - Criada em 2010;
  - Examinam problemas associados a SoSE em diversos domínios na busca de soluções estratégicas;
  - Possui suporte do IEEE e INCOSE

Considerações Finais

- Apesar de existir estudos e iniciativas que abordem a Engenharia de Requisitos de SoS. Foi possível perceber que muito ainda precisa ser feito para ter um processo de ERSoS "eficiente".

- Os maiores desafios estão relacionados as características dos SoS, independência operacional e gerencial, distribuído geograficamente e comportamento emergente.

- Existem lacunas na área que podem impactar negativamente em todo o processo de construção dos SoS.

Referências

- HALLERSTEDE, Stefan et al. Technical Challenges of SoS Requirements Engineering. In: **International Conference on System of Systems Engineering,**. 2012.
- HOFMANN, H. F.; LEHNER, F. Requirements engineering as a success factor in software projects. **IEEE software,** v. 18, n. 4, p. 58-66, 2001.
- JAMSHIDI, Mo. System of systems engineering-new challenges for the 21st century. **Aerospace and Electronic Systems Magazine, IEEE** v. 23, n. 5, p. 4–19 , 2008.
- LANE, Jo Ann; BOHN, Tim. Using SysML modeling to understand and evolve systems of systems. **Systems Engineering**, v. 16, n. 1, p. 87-98, 2013.
- LEWIS, Grace A. et al. Requirements engineering for systems of systems. In: **Systems Conference, 2009 3rd Annual IEEE**. IEEE, 2009. p. 247-252.
- Masiero, P. **Documento de requisitos**: notas de aulas engenharia de software – SCE 5764. Prof. Paulo C. Masiero. Universidade de São Paulo, 2012.

Referências

- Nielsen, C. B. and Larsen, P. G. and Fitzgerrald, J. and Wooddock, J. and Peleska, J.. Systems of Systems Engineering: based conceptts, model-based techniques, and research directions. **ACM Computing Surveys**. v. 48, n. 2, p. 18:1 – 18:41, 2015.
- NCUBE, Cornelius. On the Engineering of Systems of Systems: key challenges for the requirements engineering community. In: **Requirements Engineering for Systems, Services and Systems-of-Systems (RESS), 2011 Workshop on**. IEEE, 2011. p. 70-73.
- SAVIO, D. and ANITHA, P. C. and IYER, Parameshwar P. Considerations for a requirements engineering process model for the development of systems of systems. In: **Requirements Engineering for Systems, Services and Systems-of-Systems (RESS), 2011 Workshop on**. IEEE, 2011. p. 74-76.
- SOMMERVILLE, Ian. **Engenharia de software**. São Paulo: Addison Wesley, 2003.
- WAZLAWICK, Raul Sidnei. **Análise e projeto de sistemas de informação orientados a objetos**, 2. ed. São Paulo, SP: Campus-Elsevier, 2011, 352p.
- SOMMERVILLE, Ian. **Engenharia de Software**. 9 ed. São Paulo: Pearson Prentice Hall, 2011.