

Requisitos de Software

Introdução a Engenharia de Requisitos (ER)

 georgemarsicano@unb.br

 George Marsicano

Prof. Dr. George Marsicano



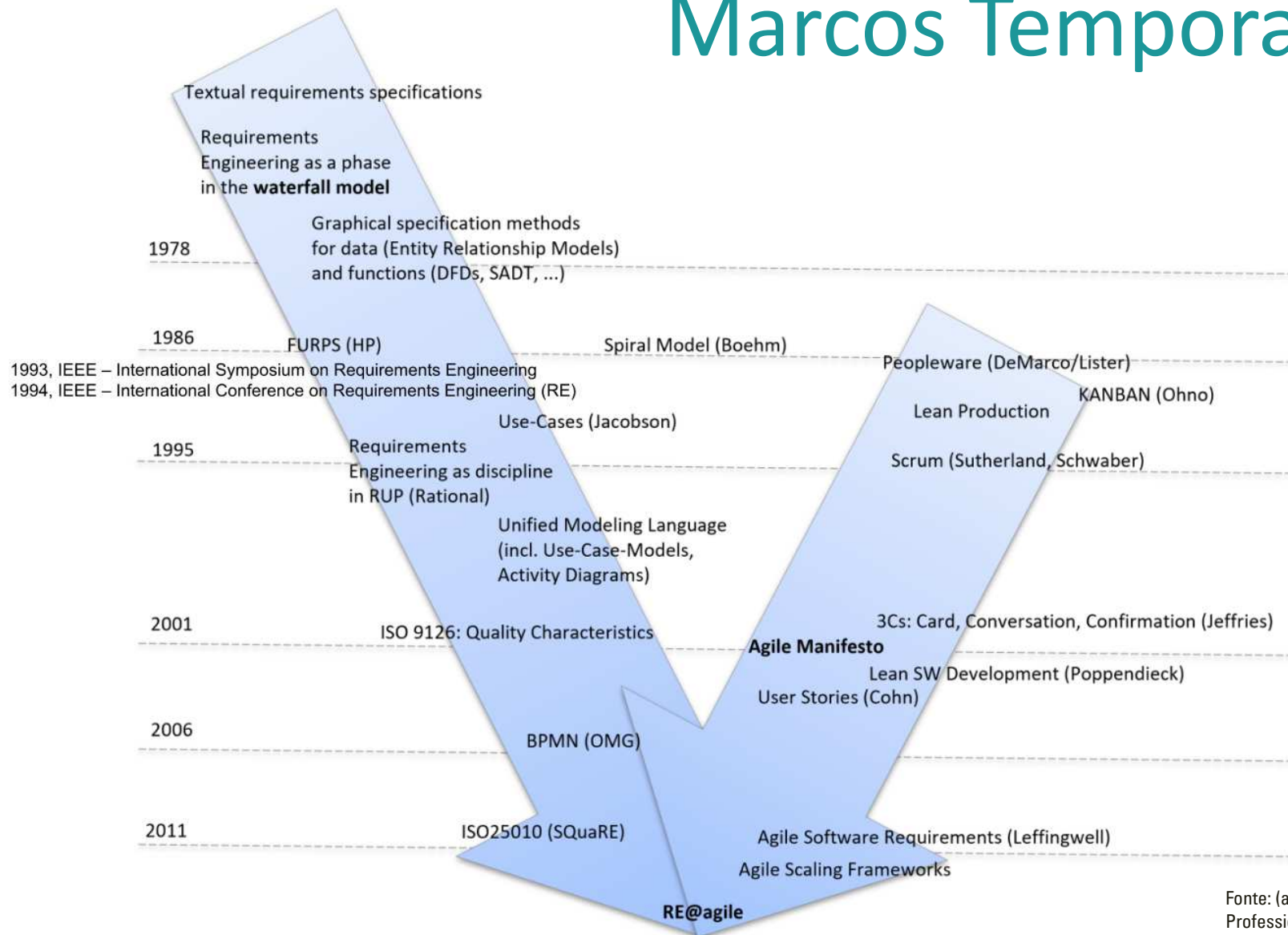
Introdução

- Dentro da *Engenharia de Software* vários modelos de processos e ciclos de vida de desenvolvimento de software, definem as etapas (disciplinas) necessárias para se construir software.
- Todos possuem algo em comum: uma disciplina dedicada a *compreensão dos problemas* a serem solucionados e a *definição de “o quê”* será feito.
- Essa disciplina, recebe o nome de *Engenharia de Requisitos (ER)*.

Quando surgiu a ER?

- Em *1993*, a partir da realização da primeira conferência *IEEE – International Symposium on Requirements Engineering*, foi criada uma nova área de pesquisa denominada *Engenharia de Requisitos (ER)*
- Em *1994*, foi realizada a primeira *IEEE – International Conference on Requirements Engineering (RE)* focada, exclusivamente, nos processos, práticas, ferramentas, conceitos, etc., vinculados a ER.

Marcos Temporais da ER



Fonte: (adaptado) Handbook RE@Agile, IREB Certified Professional for Requirements Engineering Advanced Level RE@Agile, Practitioner | Specialist, Version 2.0.0, July 2022.

O que é ER?

Engenharia de requisitos é uma área *interdisciplinar* da Engenharia de Software que visa realizar a *mediação* (ponte) entre os domínios do *adquirente* e do *fornecedor* ou *desenvolvedor*, para *estabelecer* e *manter os requisitos* a serem atendidos pelo sistema, produto de software ou serviço de interesse.

Fonte: ISO-IEC-IEEE 29148 - Systems and software engineering - Life Cycle Process - Requirements Engineering

Outras conceituações complementares

“Uma área *multidisciplinar, centrada no ser humano* e em seus *problemas*. Deve, portanto, investigar como as pessoas *percebem* e *entendem* o mundo ao seu redor, como elas interagem e como a *sociologia dos locais* de trabalho afeta suas ações. A ER deve se utilizar, portanto, das *ciências sociais* e *cognitivas*, tais como a filosofia, a psicologia cognitiva, a sociologia e a linguística, dentre outras, para fornecer fundamentos teóricos e técnicas para elicitar e modelar requisitos”. (NUSEIBEH, 2000)

Outras conceituações complementares

(cont.)

“Um processo *iterativo, incremental, cognitivo, social, comunicativo* e *criativo*, cujos objetivos são conhecer, entender, estruturar, representar, comunicar e transcrever as informações relevantes de um sistema, extraídas a partir de diferentes segmentos de informação: ambiente da organização; a organização; a gerência; e o desenvolvimento”. (CARVALHO, 2003)

Outras conceituações complementares

(cont.)

“Um conjunto de atividades utilizadas para *identificar* e *comunicar* a finalidade de um sistema de software, e o contexto no qual será usado. Assim, a ER atua como a ponte entre as *necessidades* reais dos usuários, clientes, e outros grupos afetados por um sistema de software, e as *potencialidades* e *oportunidades* oferecidas pela tecnologia”. (EASTERBROOK, 2004)

Outras conceituações complementares

(cont.)

*Área interdisciplinar da
Engenharia de Software*

*Ponte entre as necessidades
reais dos usuários*

*Área multidisciplinar, **centrada no
ser humano** e em seus problemas*

*Utilizar das **ciências sociais e cognitivas**
(filosofia, psicologia cognitiva, sociologia
e linguística)*

*Processo **iterativo, incremental, cognitivo,
social, comunicativo e criativo***



Qual o Foco da ER?

- A Engenharia de Requisitos está focada em:
 - Descobrir
 - Desenvolver
 - Rastrear
 - Analisar
 - Qualificar
 - Verificar e validar
 - Comunicar
 - Documentar e
 - Gerenciar os requisitos de um software.

Fonte: ISO-IEC-IEEE 29148 - Systems and software engineering - Life Cycle Process - Requirements Engineering
DICK, Jeremy; HULL, Elizabeth; JACKSON, Ken. Management Aspects of Requirements Engineering. In: Requirements Engineering. Springer, Cham, 2017. p. 207-230.



Quais são os Resultados da ER?

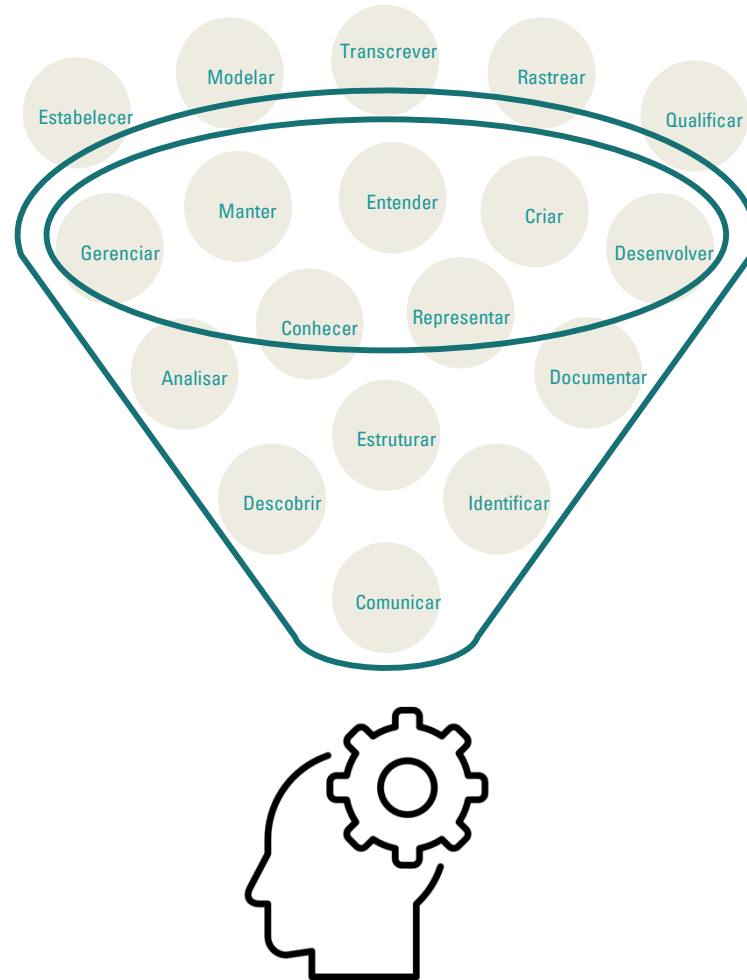
- O *principal resultado* da ER são *conjuntos de requisitos*, onde cada conjunto deve:
 - Ser *referência* para um sistema, software ou serviço definido;
 - Possibilitar um *entendimento acordado* entre as partes interessadas (por exemplo, adquirentes, usuários, clientes, operadoras, fornecedores);
 - Ser *validado* em relação às *necessidades* do mundo real;
 - Ser *implementável*, e
 - Fornecer uma *referência* para a *verificação* de projetos e soluções.

Fonte: ISO-IEC-IEEE 29148 - Systems and software engineering - Life Cycle Process - Requirements Engineering

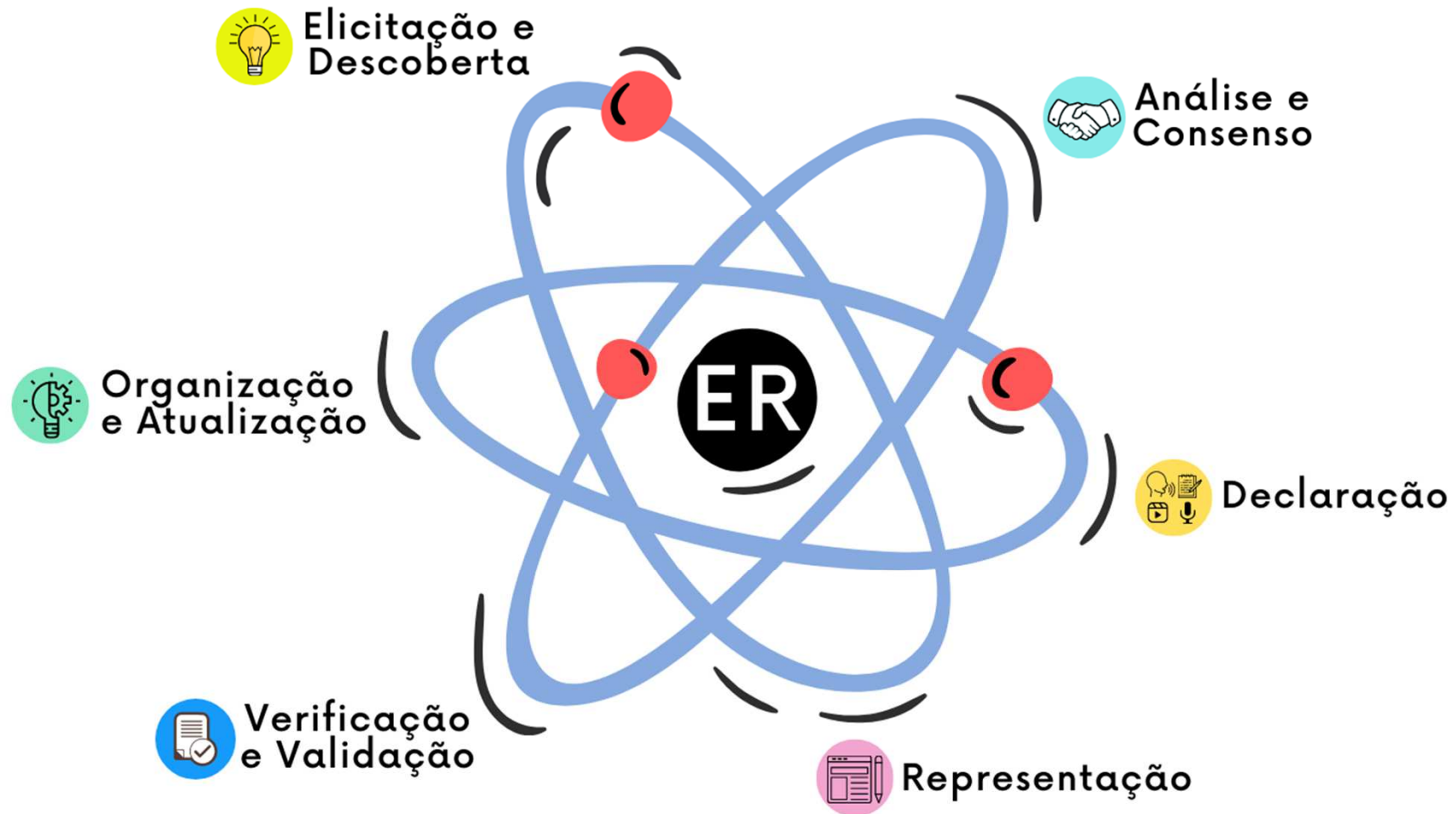
O que faz ?

- De uma maneira geral a Engenharia de Requisitos executa as *mesmas atividades* em processos *Dirigidos à Plano e Ágeis*.
- Contudo, *a filosofia* (Dirigidos à Plano e Ágeis) que sustenta essa execução, faz com que tais atividades sejam *realizadas de maneira singular*.

Quais são as Atividades de ER?



Atividades da Engenharia de Requisitos (ER)



© George Marsicano, 2023.



Elicitação e Descoberta

- **Elicitação**
 - *Extrair, obter* ou *provocar* uma *resposta, reação* ou *informação* de alguém ou de algo.
- **Descoberta**
 - *Encontrar* algo que antes *não* era *conhecido* ou *não* estava *disponível*.
- A elicitación e descoberta podem ser realizadas, por meio de várias fontes de informação como, por exemplo, *stakeholders, documentos, ferramentas (IA), softwares e sistemas existentes*.

© George Marsicano, 2023.



Fonte: baseado em Requirements Engineering for Software and Systems, Third Edition, Phillip A. Laplante, 2018.



Elicitação e Descoberta

- Deve-se estar atendo tanto aos requisitos *funcionais* (RFs), quanto aos requisitos *não funcionais* (RNFs).
- Também envolve descobrir quem são os *interessados*, seus *problemas*, *necessidades*, *desejos* e *expectativas*, além de outros fatores que podem representar *possibilidades* ou *restrições* ao produto de software.

© George Marsicano, 2023.



Fonte: baseado em Requirements Engineering for Software and Systems, Third Edition, Phillip A. Laplante, 2018.



- **Análise**

- Envolve *analisar os requisitos* em sua *forma “bruta”*.
- Requisitos brutos:
 - Nem sempre fazem sentido;
 - Muitas vezes eles se contradizem (e nem sempre de forma óbvia);
 - Eles podem ser inconsistentes;
 - Podem estar incompletos;
 - Podem ser vagos ou simplesmente errados.
 - Podem interagir e ser dependentes uns dos outros.

© George Marsicano, 2023.



- **Consenso**

- Os requisitos de um produto de software podem possuir *diversas fontes de informação* (stakeholders, documentos, ferramentas (IA), softwares e sistemas existentes, etc.), as quais podem ser divergentes.
- É preciso *conciliar* essas fontes em direção a um *entendimento comum* sobre o conjunto de requisitos.

© George Marsicano, 2023.



Declaração

- Refere-se a *comunicação* dos requisitos entre os envolvidos, por meio de *linguagem natural*, estruturada ou não, de maneira *escrita* e/ou *oral* (face a face, áudio, vídeo) em diferentes níveis de *granularidade* (mais detalhes, menos detalhes, agrupados ou divididos).
- Devem ser declarados todos os *requisitos funcionais* (RFs) e *não funcionais* (RNFs).

© George Marsicano, 2023.



Fonte: baseado em Requirements Engineering for Software and Systems, Third Edition, Phillip A. Laplante, 2018.



Representação

- Envolve a apresentação dos requisitos em *modelos e/ou visualizações do produto*.
- Representação de requisitos:
 - *Informal*: representações informais ainda que possam seguir algum padrão visual;
 - *Semiformal*: representações que seguem uma notação ou linguagem específicas.
 - *Formal*: representações baseadas em uma sintaxe e semântica matemática.

© George Marsicano, 2023.



Fonte: baseado em Requirements Engineering for Software and Systems, Third Edition, Phillip A. Laplante, 2018.



Verificação e Validação

- Dizem respeito a *qualidade dos requisitos*.
- **Validação** (qualidade externa)
 - É a confirmação (ou não) de que os requisitos (individualmente e/ou em conjunto) *definem a solução correta*, conforme combinado com as partes interessadas (patrocinadores, áreas de negócio, usuários finais, etc.).
 - A validação pode precisar retomar as *necessidades* do cliente, *problemas/oportunidades* e *objetivos* de negócio.

“Estou realizando o requisito correto?”

© George Marsicano, 2023.



Verificação e Validação

- **Verificação** (qualidade interna)
 - É a confirmação (ou não) de que os requisitos (individualmente e em conjunto) foram *realizados da maneira correta*.

“Estou realizando de maneira correta, o requisito?”

- Também faz parte da verificação e validação:
 - Estabelecer a *definição de preparado* (Definition of Ready - DoR);
 - Participar da *definição de pronto* (Definition of Done - DoD).

© George Marsicano, 2023.



Organização e Atualização

- Referem-se a *manter* o conjunto de requisitos do produto, adequadamente, organizados e atualizados *ao longo do tempo*.
- **Organização:**
 - Estabelecer a maneira como os requisitos do produto serão *estruturados* (lista, mapa), *rastreados*, *refinados* e *priorizados* com base no framework de requisitos adequado ao produto e/ou projeto.
- **Atualização:**
 - Manter a organização dos requisitos sempre em seu *estado mais atual*, diante das possíveis *mudanças* e *consenso* entre as fontes de requisitos.

© George Marsicano, 2023.

Para quê ER ?

- Para capturar os requisitos necessários a construção de software de qualidade
- Se não são capturados bons requisitos
 - Pode ser construído um excelente software para resolver o problema errado
 - Perda de tempo e dinheiro
 - Frustração pessoal
 - Usuários infelizes
 - Fracasso do projeto e do produto
 - Consequências ambientais negativas
 - ...

PROCESSO DE ER



Os processos e atividades da ER podem variar amplamente dependendo do(a)...

Contexto de desenvolvimento

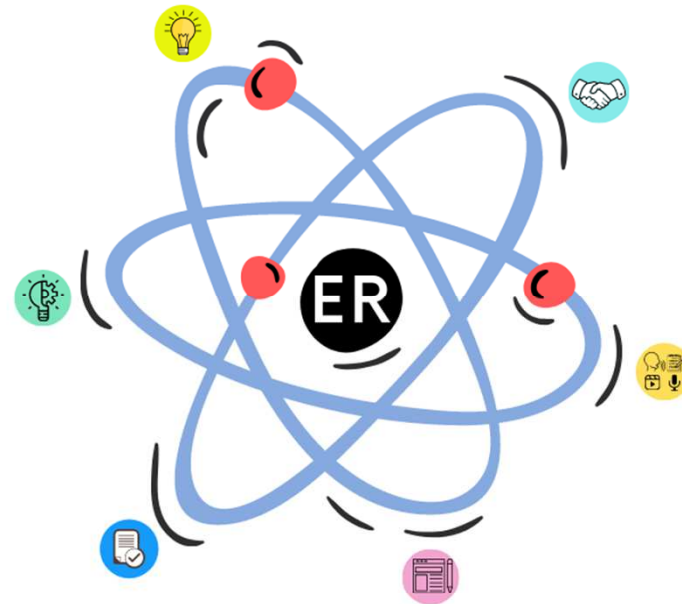
- Relação cliente-fornecedor-usuário
- Tipo de desenvolvimento
- Contrato
- Confiança

Características da Equipe

- Tamanho da equipe
- Experiência com requisitos
- Distribuição
- Competências

Restrições

- Regulatórias
- Clientes
- Tempo
- Orçamento
- Cultura



Entendimento compartilhado

- Problemas
- Necessidades
- Requisitos de produto de software

Requisitos de Software

- Variabilidade
- Riscos
- Viabilidade
- Dependência

Produto de Software

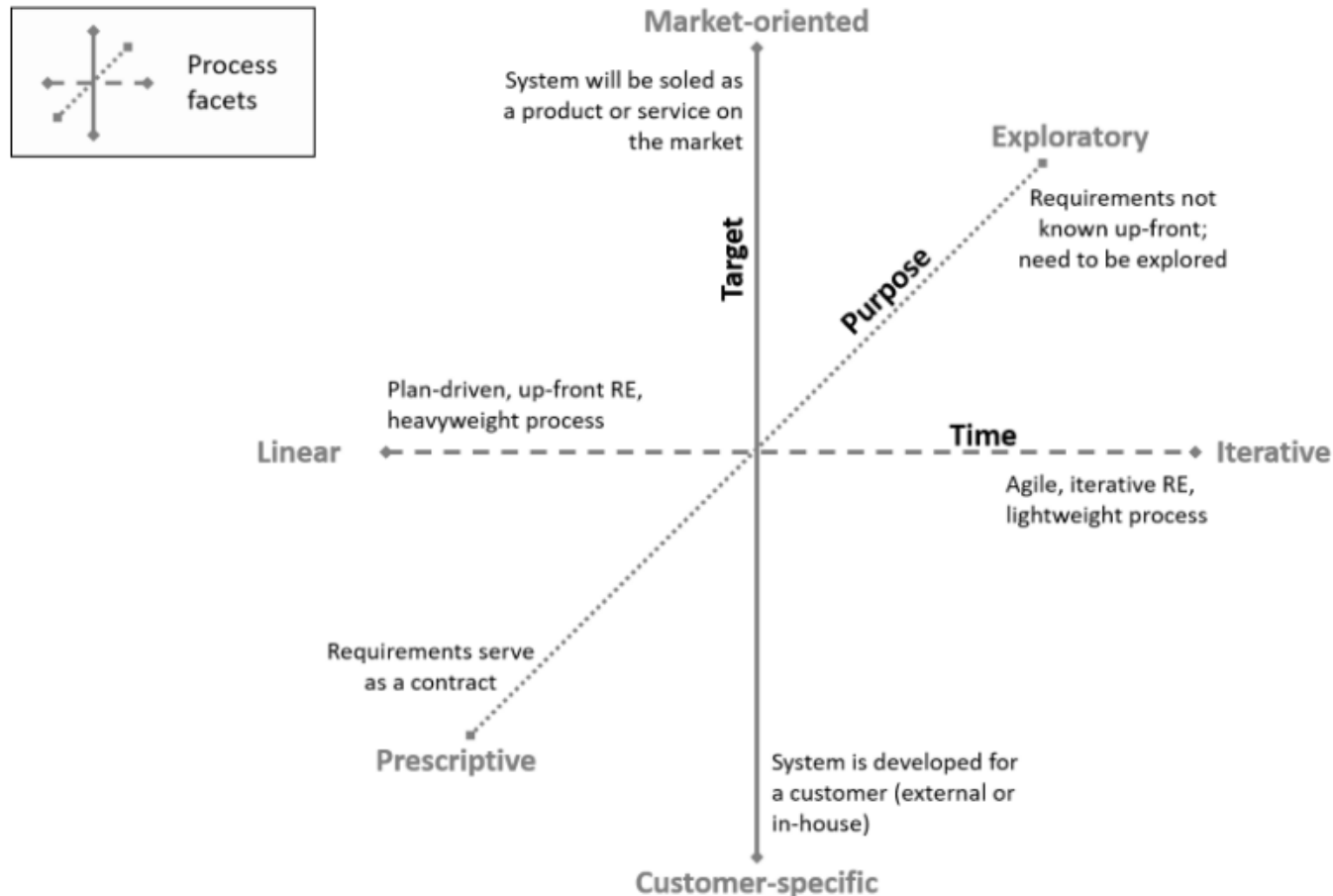
- Complexidade
- Criticidade

Stakeholders

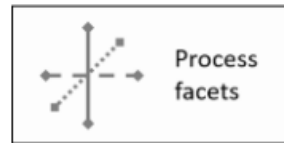
- Capacidade
- Disponibilidade

© George Marsicano, 2023.

Facetas do Processo de ER



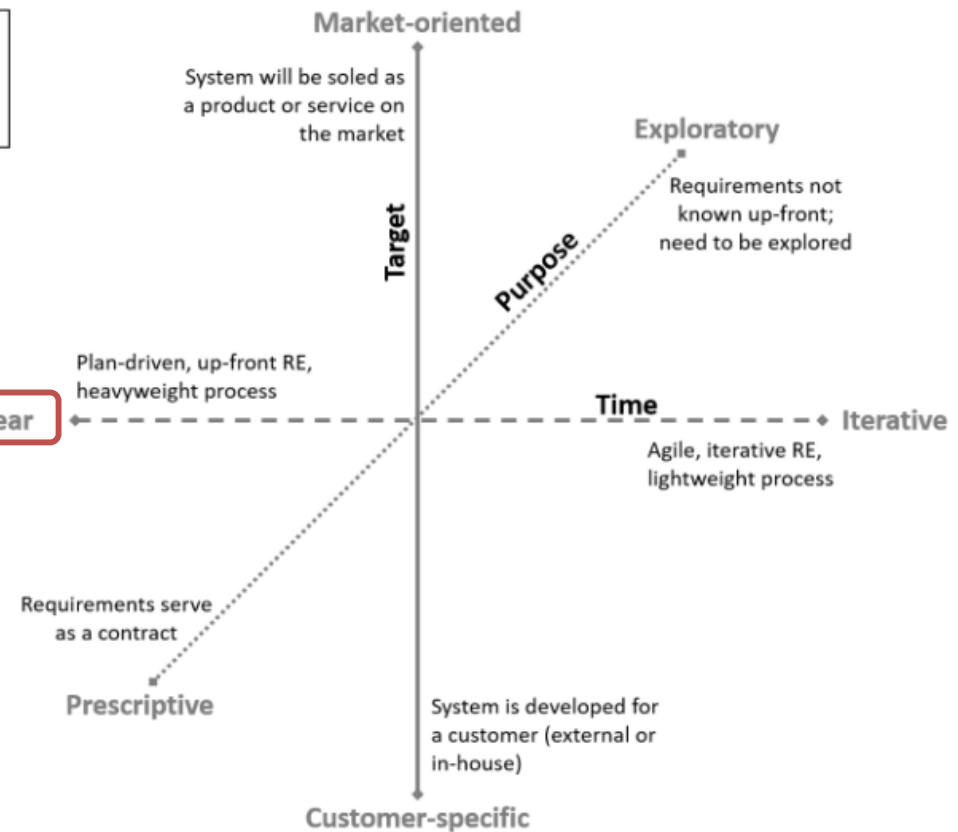
Facetas do Processo de ER



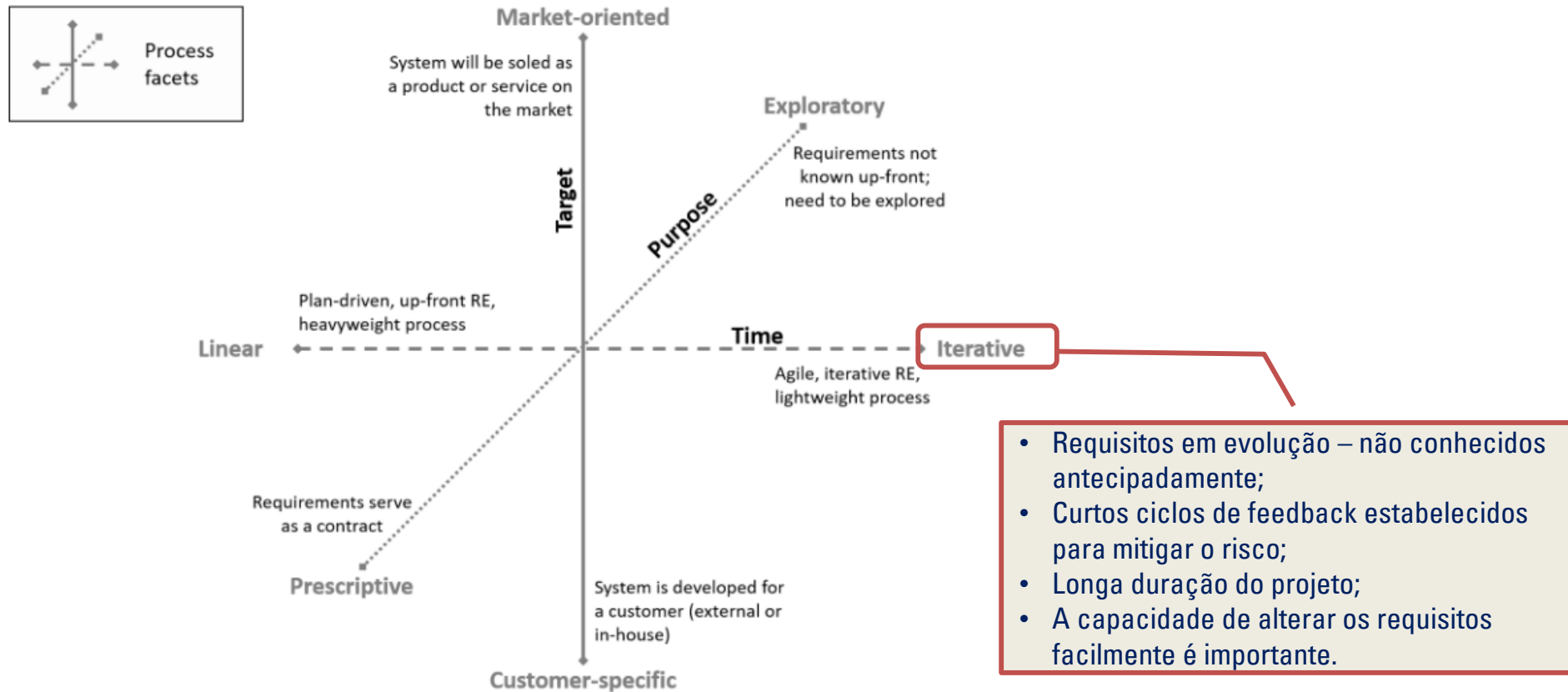
Linear

O processo de desenvolvimento é orientado a planos e, principalmente, linear:

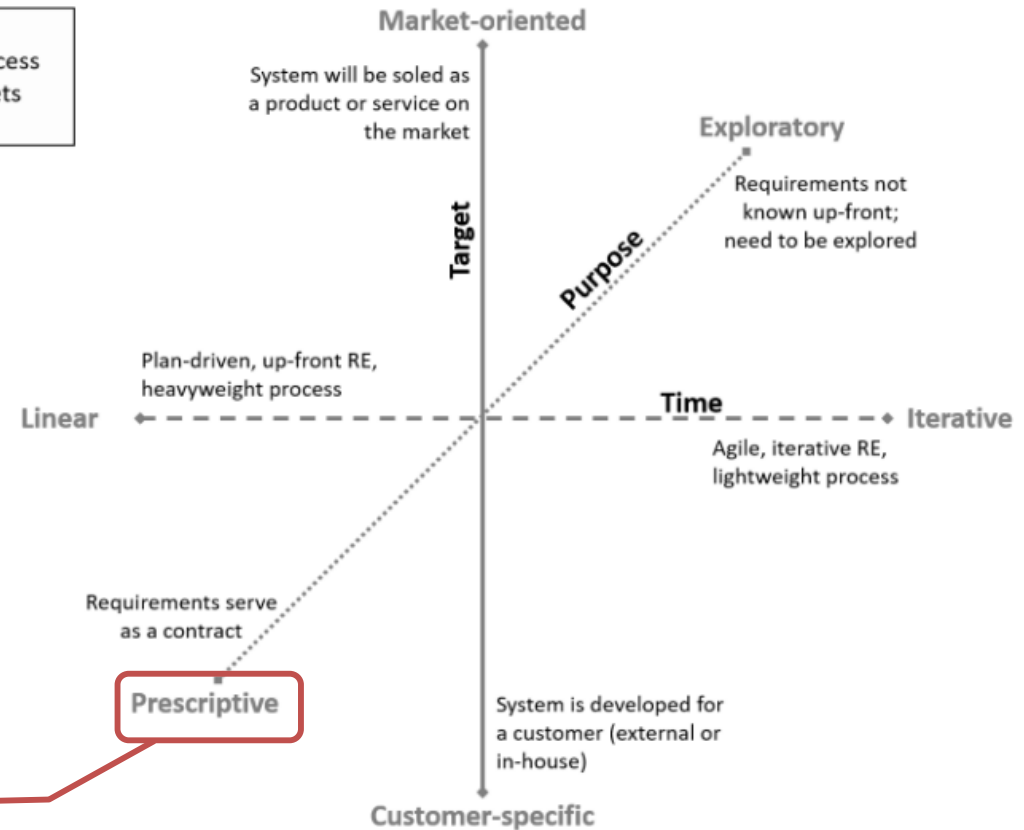
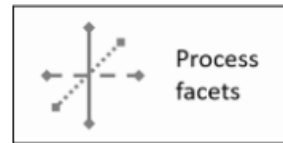
- Requisitos claros, estáveis e conhecidos à priori;
- Especificação abrangente de requisitos necessária como um base contratual para design e implementação de terceirização;
- As autoridades reguladoras exigem um congelamento de requisitos.



Facetas do Processo de ER



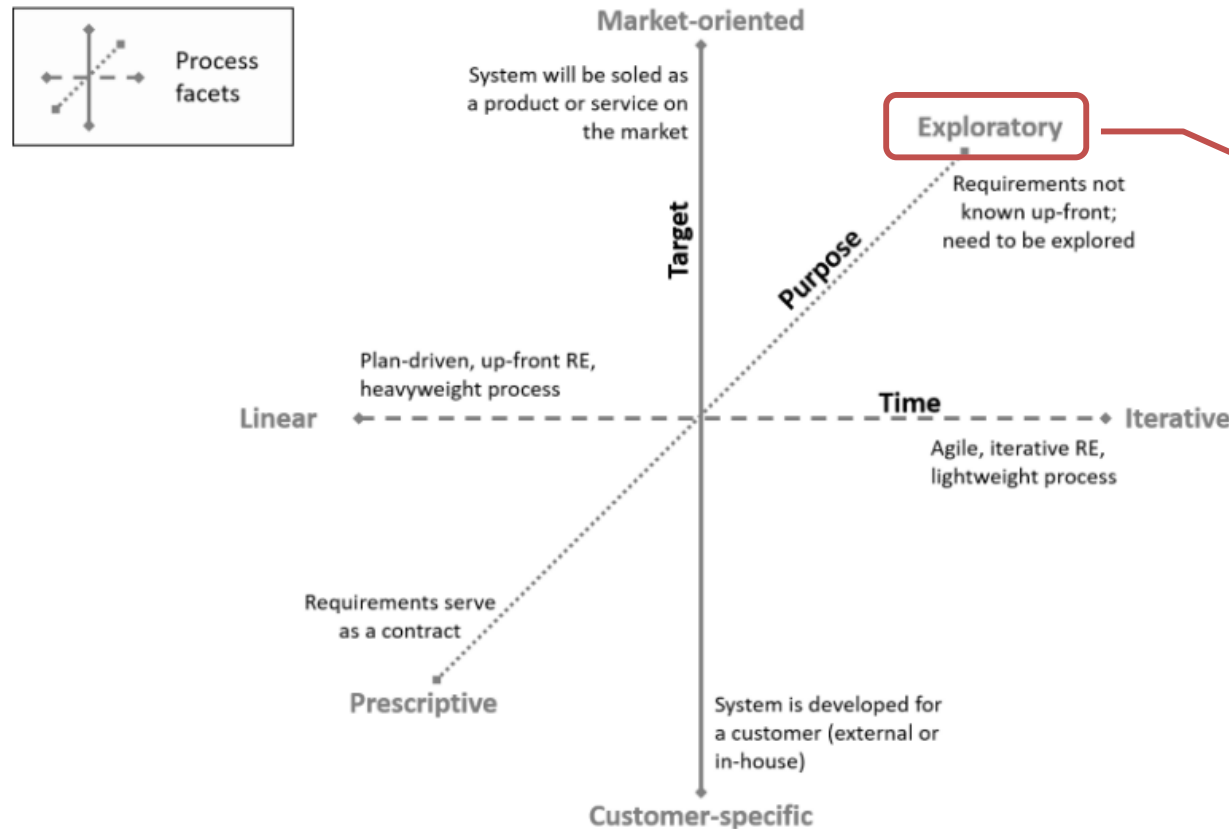
Facetas do Processo de ER



A especificação de requisitos é um contrato. Todos os requisitos são obrigatórios e devem ser implementados Critérios de seleção:

- Os requisitos determinam custos e prazos;
- O cliente exige um contrato de preço fixo;
- Projeto e implementação, geralmente, licitados ou terceirizados.

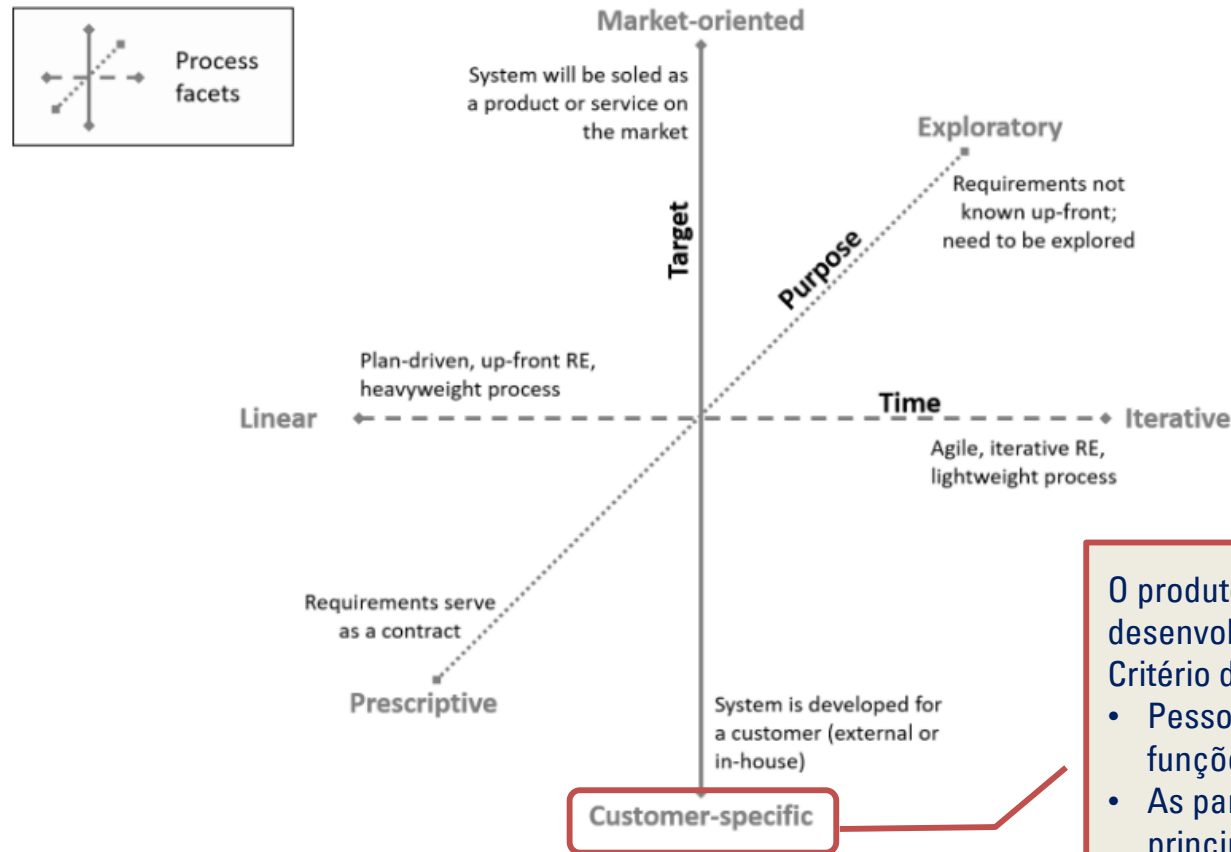
Facetas do Processo de ER



Apenas objetivos conhecidos, os requisitos concretos devem ser explorados. Critérios de seleção:

- Partes interessadas fortemente envolvidas, feedback contínuo;
- Prazos e custos restringem os requisitos;
- Priorizar e negociar os requisitos a serem implementados.

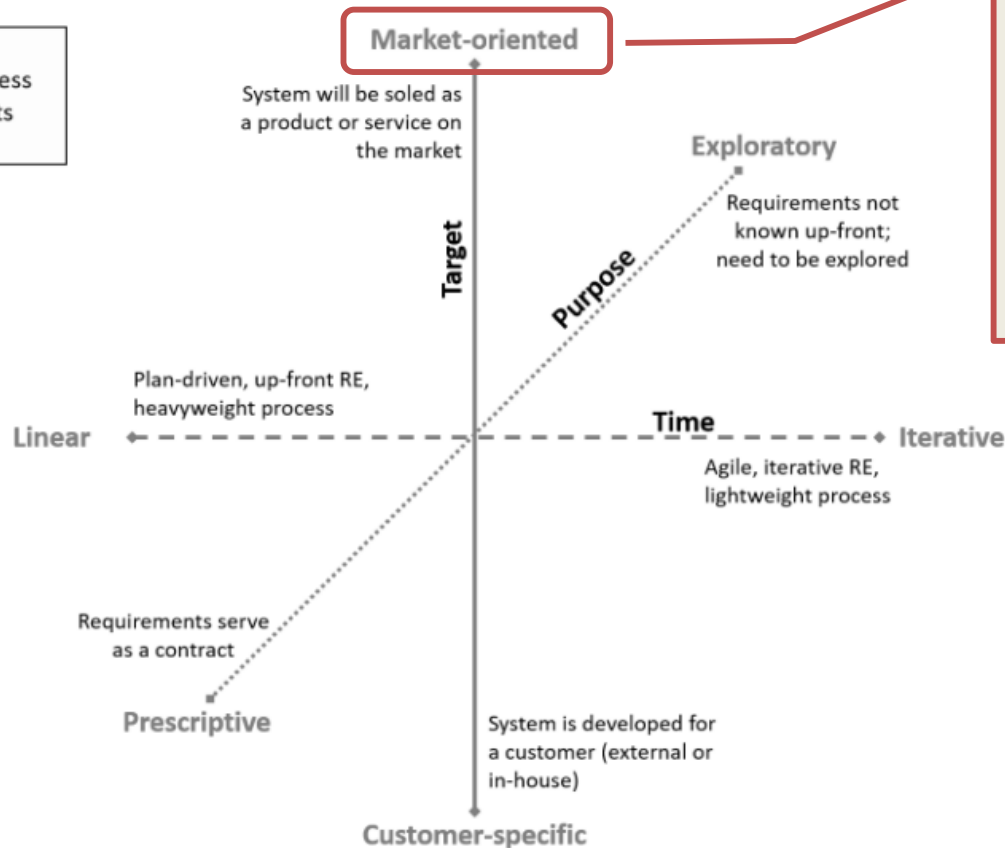
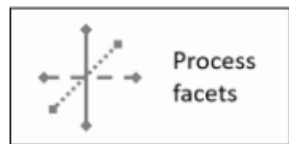
Facetas do Processo de ER



O produto é encomendado por um cliente e desenvolvido por um fornecedor para este cliente.
Critério de seleção:

- Pessoas individuais identificáveis para todas as funções das partes interessadas;
- As partes interessadas do lado do cliente são a principal fonte de requisitos;
- O relacionamento contratual do fornecedor do cliente afeta o processo.

Facetas do Processo de ER



O software é desenvolvido como um produto ou serviço para um mercado. Critério de seleção:

- Usuários em potencial não identificáveis individualmente;
- Requisitos especificados pelo fornecedor;
- O fornecedor tem que adivinhar/estimar/eliminar as necessidades dos clientes/usuários previstos;
- O pessoal de marketing, designers digitais e arquitetos de sistema são os principais partes interessadas.

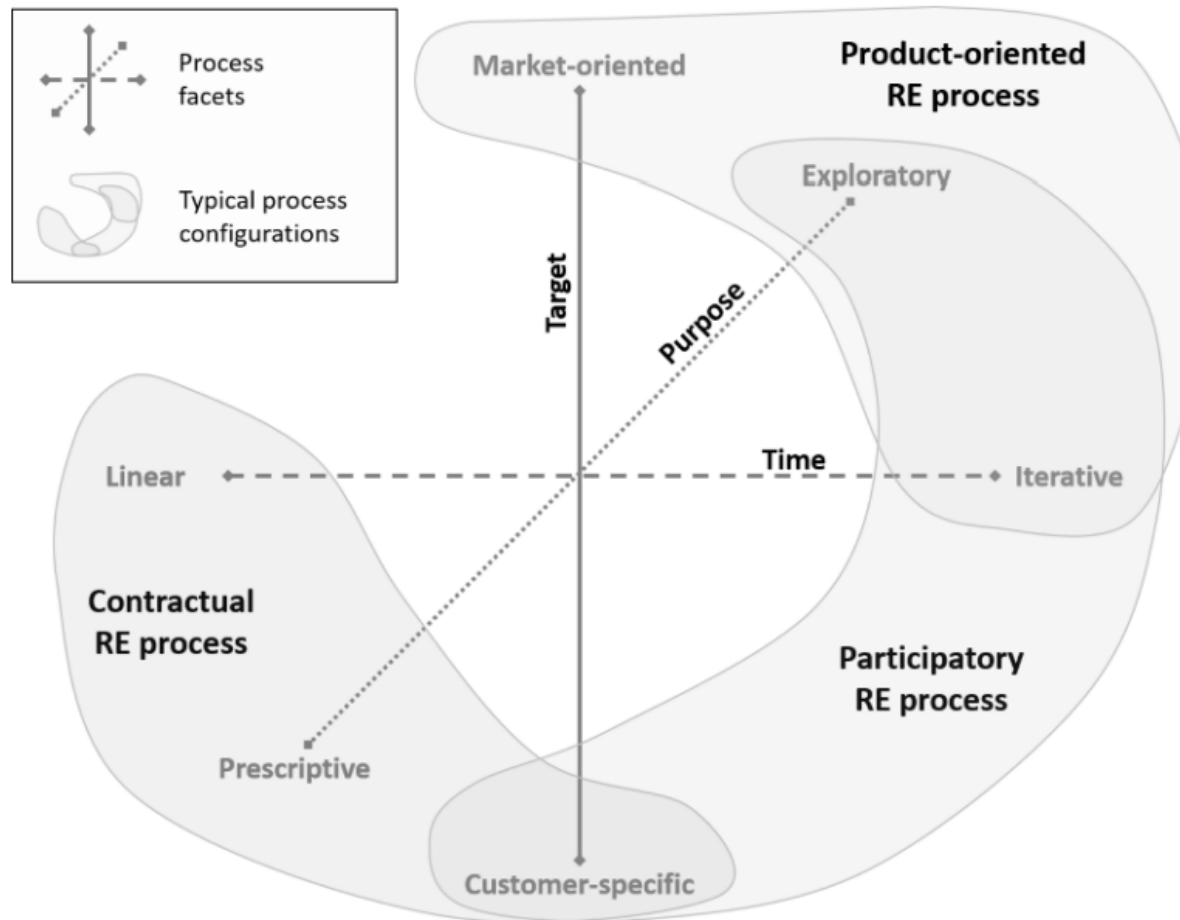
Dicas e Comentários

- Os processos de ER lineares funcionam apenas *se* houver um processo maduro de *alteração de requisitos* em vigor
- Processos de ER lineares implicam *longos ciclos de feedback*. validação intensiva de requisitos deve ser realizada
- Os processos de ER orientados para o mercado dependem crucialmente do *feedback rápido* dos usuários-piloto para validar se o produto, realmente, irá satisfazer as necessidades do segmento de usuários-alvo

Dicas e Comentários (cont.)

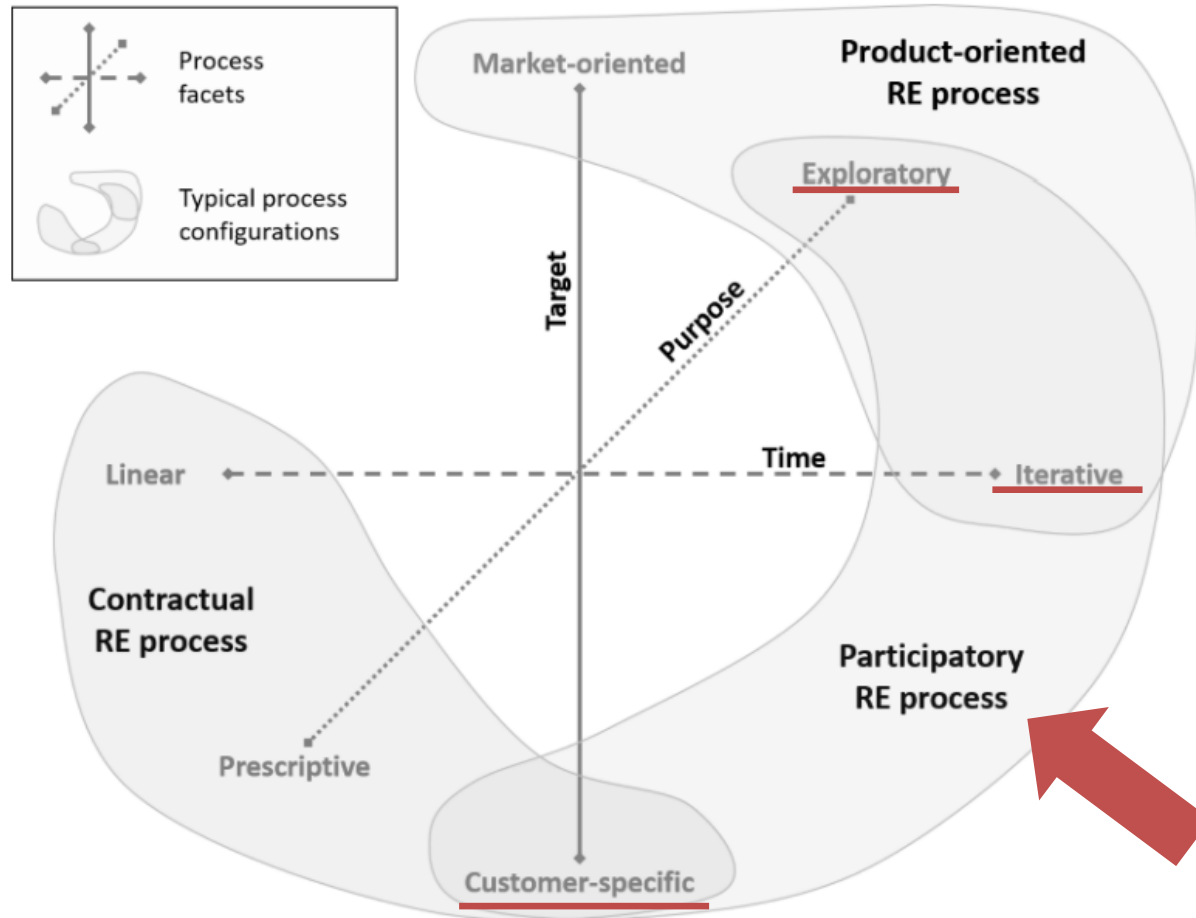
- Orientado para o mercado *não combina* bem com Linear e Prescritivo
- Combinações *frequentes*:
 - Linear e Prescritivo
 - Exploratório e Iterativo

Configurações 'Típicas' do Processo de ER



Configurações 'Típicas' do Processo de ER

(cont.)



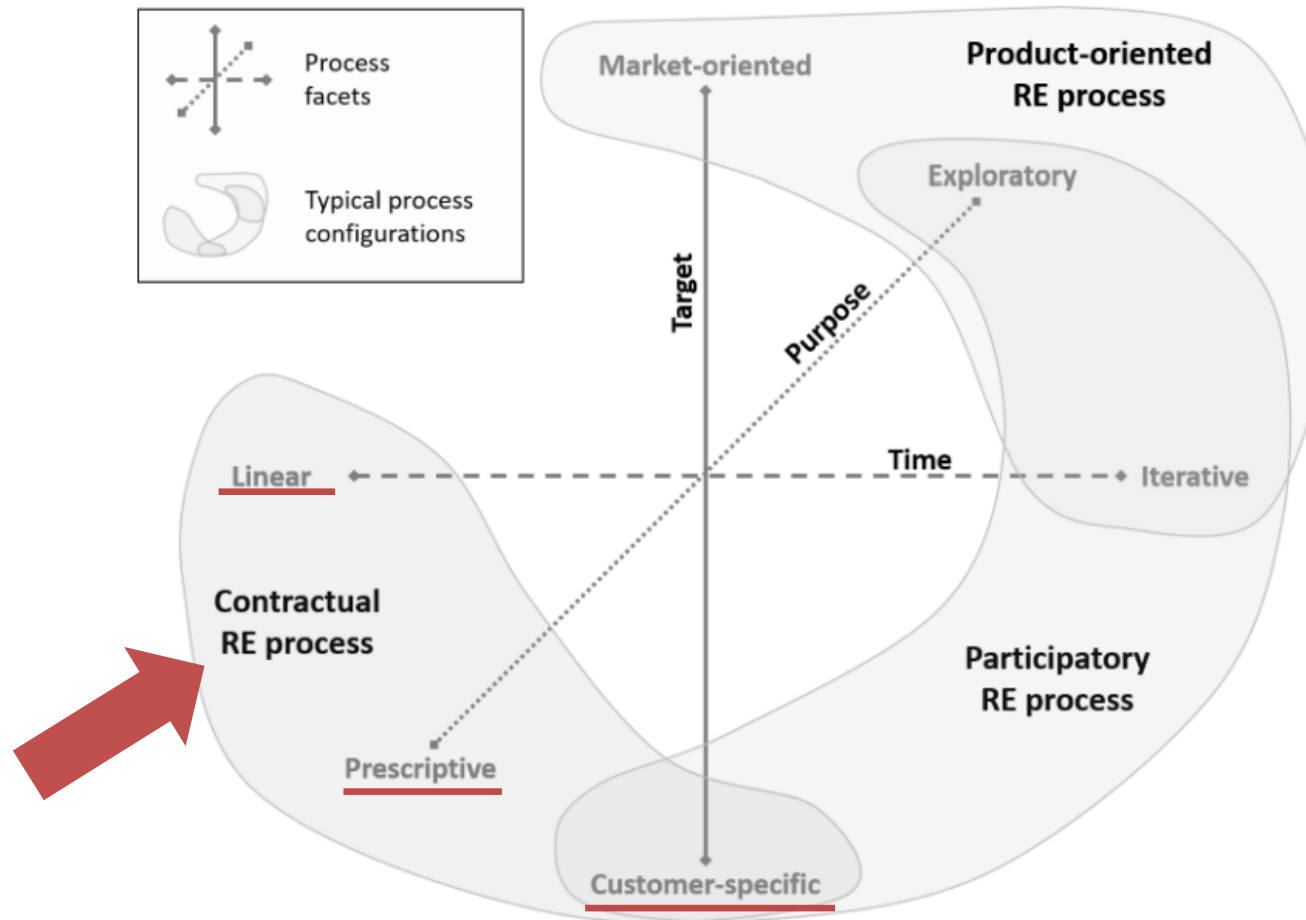
Configurações 'Típicas' do Processo de ER

(cont.)

- **Participativo:** iterativo e exploratório e cliente específico
 - Caso de aplicação principal
 - Fornecedor e cliente colaboram estreitamente; partes interessadas do cliente permanecem fortemente envolvidas nos processos de especificação e desenvolvimento.
 - Produtos de trabalho típicos
 - Backlog do produto, com histórias de usuários e/ou descrições de tarefas, protótipos.
 - Fluxo de informações típico
 - Interação contínua entre as partes interessadas, dono do produto, engenheiros de requisitos e desenvolvedores.

Configurações 'Típicas' do Processo de ER

(cont.)



Configurações 'Típicas' do Processo de ER

(cont.)

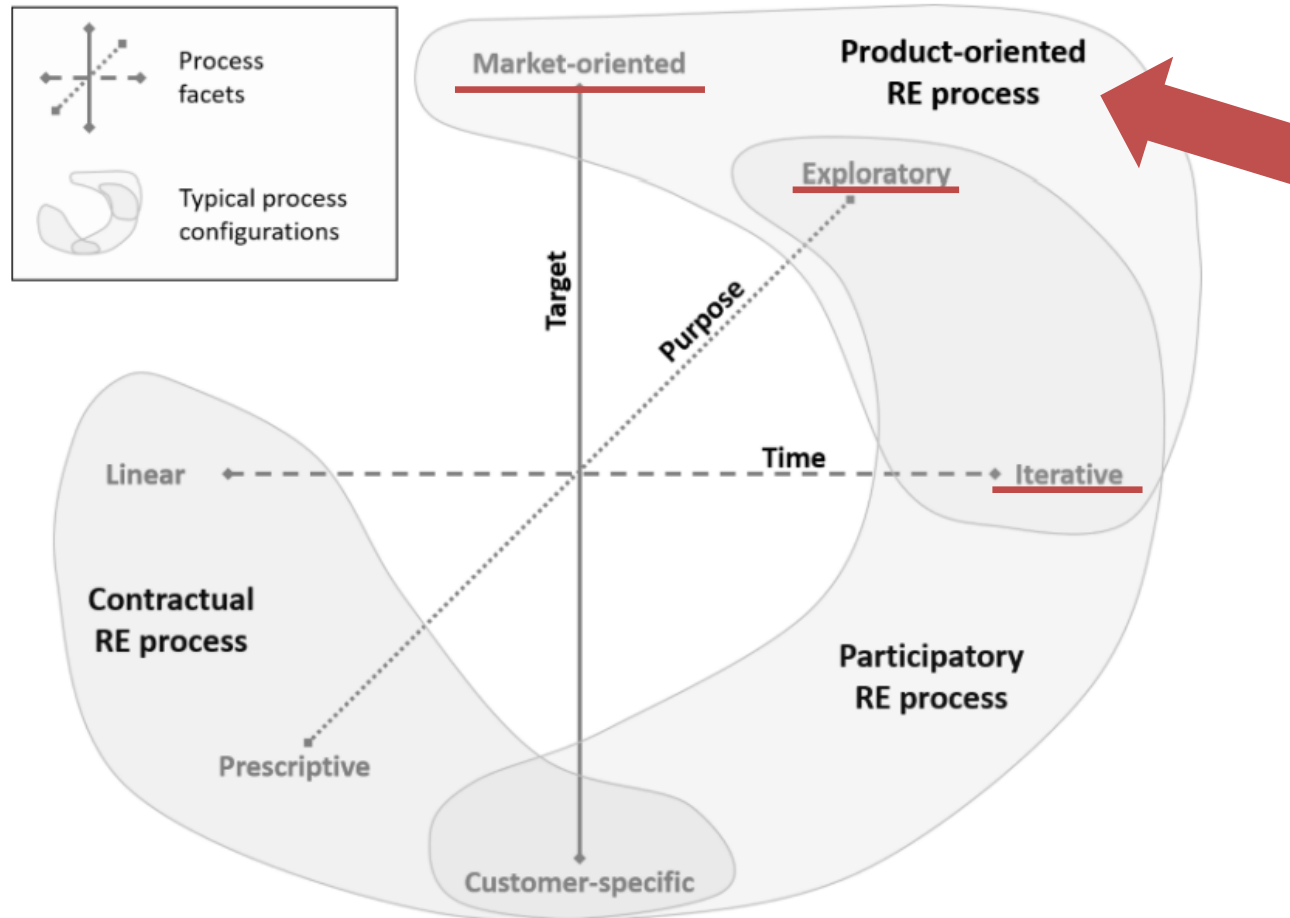
- **Contratual:** Normalmente linear (às vezes iterativo) e prescritivo e cliente específico
 - Caso de aplicação principal
 - A especificação constitui base contratual para o desenvolvimento de um sistema por pessoas não envolvidas na especificação e com pouca interação das partes interessadas após a fase de requisitos
 - Produtos de trabalho típicos
 - Especificação clássica de requisitos de sistema, consistindo em requisitos textuais e modelos.
 - Fluxo de informações típico
 - Principalmente das partes interessadas aos engenheiros de requisitos



Fonte:
Baseado em Requirements Engineering I – Part II: RE
Practices © 2019 Martin Glinz

Configurações 'Típicas' do Processo de ER

(cont.)



Configurações 'Típicas' do Processo de ER

(cont.)

- **Orientado a Produto:** iterativo e exploratório e orientado ao mercado
 - Caso de aplicação principal
 - Uma organização específica e desenvolve o software visando para vendê-lo / distribuí-lo como um produto, ou serviço.
 - Produtos de trabalho típicos
 - Backlog do produto, protótipos.
 - Fluxo de informações típico
 - Interação entre dono do produto, marketing, engenheiros de requisitos, designers digitais, desenvolvedores e (talvez) feedback rápido clientes / usuários.

Processos Ágeis de ER

- Leva a *incrementalidade* e a *exploração* ao extremo
 - Iterações de duração fixa de 1 a 6 semanas
 - O dono do produto ou representante do cliente está sempre disponível e tem poder para tomar decisões imediatas
 - Apenas metas e visão são estabelecidas antecipadamente
 - Os requisitos são declarados como histórias (poucos detalhes e critérios de aceitação)
 - No início de cada iteração:
 - O cliente / dono do produto prioriza os requisitos
 - Os desenvolvedores selecionam o que implementar nessa iteração
 - Ciclo curto de feedback dos requisitos para o produto implementado

Processo de ER “Ideal”

- Fortemente *interativo* e *incremental*
- *Colaboração* estreita e intensa entre
 - Partes interessadas (conhecem o domínio e o problema)
 - Engenheiros de software / requisitos (saber como especificar)
- Ciclos de *feedback* muito *curtos*
- Ciente do *risco* e ciente da *viabilidade*
 - Riscos técnicos/viabilidade
 - Riscos de prazo/viabilidade
- *Consenso* / resolução cuidadosa de *requisitos conflitantes*
- Concentre-se em estabelecer um *entendimento compartilhado*
- Esforça-se pela *inovação*

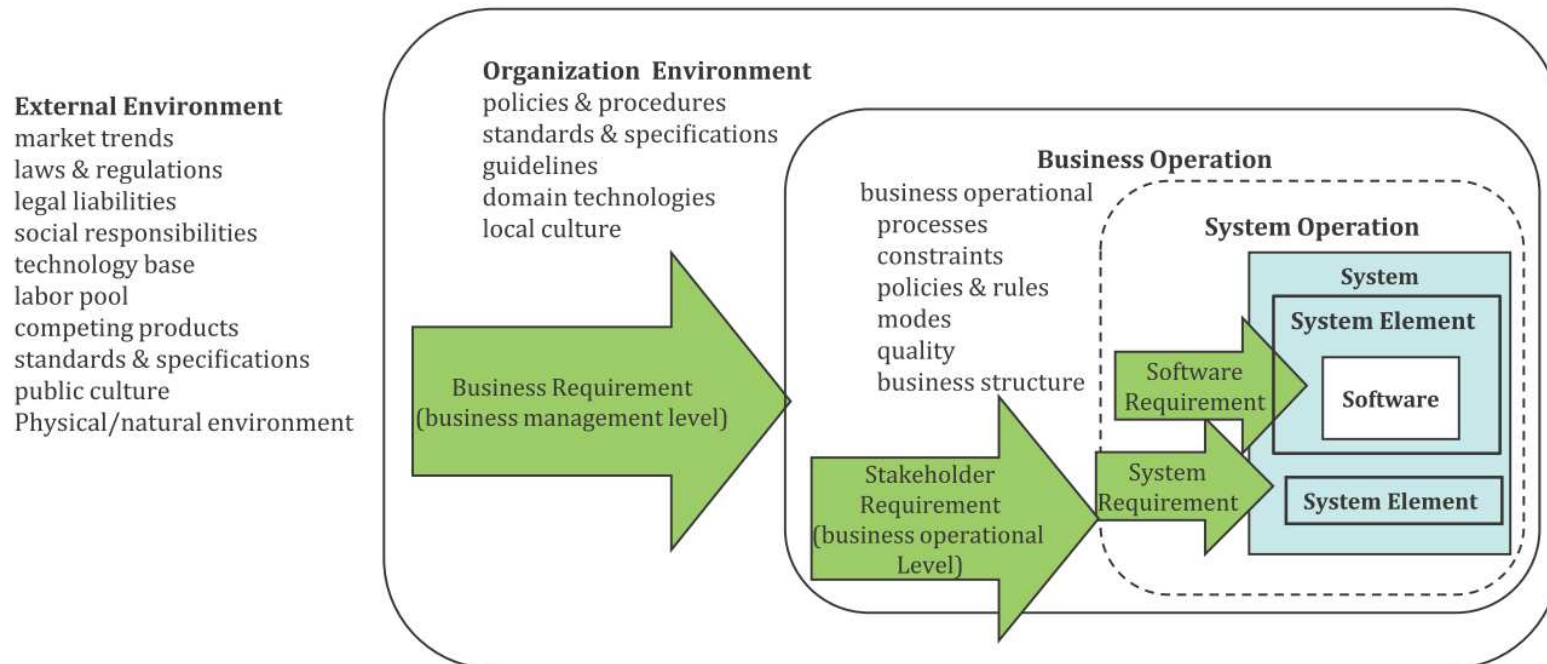
O que é um artefato da ER ?

- Local que onde os requisitos são registrados, em maior ou menor nível de detalhe, podendo ser RFs ou RNFs
- Um artefato auxilia no registro e organização dos requisitos.
- Os requisitos de software (RFs e RNFs) podem ser registrados, por exemplo, em:

Backlog de requisitos	Cenários
User story mapping	Documentos de especificação de requisitos
Histórias de usuários	Especificações Suplementares
Casos de uso	Código fonte

Em que contexto a ER está inserida?

- A ER, assim como a Engenharia de Software podem ser realizadas para gerar resultados a diversos contextos, por exemplo, contexto de negócio



Fonte: ISO-IEC-IEEE 29148 - Systems and software engineering - Life Cycle Process - Requirements Engineering

A INFLUÊNCIA DE REQUISITOS EM PROJETOS



A Influência de Requisitos em Projetos

Table 1.1 Reasons for project failure (Standish Group (1995))

**	Incomplete requirements	13.1%
**	Lack of user involvement	12.4%
	Lack of resources	10.6%
*	Unrealistic expectations	9.9%
	Lack of executive support	9.3%
**	Changing requirements/specifications	8.7%
	Lack of planning	8.1%
**	Didn't need it any longer	7.5%

Table 1.2 Project success factors (Standish Group (1995))

**	User involvement	15.9%
	Management support	13.9%
**	Clear statement of requirements	13.0%
	Proper planning	9.6%
*	Realistic expectations	8.2%
	Smaller milestones	7.7%
	Competent staff	7.2%
*	Ownership	5.3%

A Influência de Requisitos em Projetos

(cont.)

Table 5.1 A History of Project Failures Due to Poor Requirements Practices

<i>System Name</i>	<i>Year</i>	<i>Requirements Process Failure</i>
HMS Titanic	1912	Poor requirements design
Apollo-13	1970	Insufficient requirements verification
IBM PCjr	1983	Poor requirements design
Space Shuttle Challenger	1986	Insufficient requirements verification
Mars Climate Orbiter	1999	Poor requirements design
Space Shuttle Columbia	2002	Insufficient requirements verification

Source: Adapted from Bahill, T.A. & Henderson, S.J., *Syst. Eng.*, 8, 1–14, 2005.

Fonte: Phillip A. Laplante. Requirements Engineering for Software and Systems. Third Edition. CRC Press. 2018.

A Influência de Requisitos em Projetos

(cont.)

Desafios na relação entre
equipe de desenvolvimento
de software e seus clientes.

Fonte: HOFFMANN, Marco et al. The human side of Software Engineering Teams: an investigation of contemporary challenges. *IEEE Transactions on Software Engineering*, 2022.

TABLE 3: Client Challenges

No.	Client Challenge
C1	Lack of communication
C2	Client does not know what they want
C3	Lack of interest in the project by the client
C4	Missing IT project experience at client side
C5	Missing technical knowledge at client side
C6	Exaggerated quality expectation of the client
C7	Conflicts of interests at client side
C8	Client unable to specify functional requirements
C9	Client unable to specify non-functional reqs
C10	Unclear roles and responsibilities at client side
C11	Lack of prioritization by client
C12	Weak management at client side
C13	Insufficient collaboration
C14	Insufficient analysis at the beginning of the project
C15	No direct communication with client
C16	Communication plan is neglected
C17	Subjective interpretations of tasks
C18	Work is not solution-oriented
C19	Language barriers
C20	Misinterpretations
C21	Missing respect towards the client from the team
C22	Missing respect towards the team from the client
C23	Delays due to dependencies to client's third parties
C24	People not reporting problems in time
C25	Over-Confidence
C26	Communication of problems to client restricted
C27	Exaggerated seeking of project problems

A Influência de Requisitos em Projetos

(cont.)

Table 1.3 Project success factors (Standish Group (2015).)

*	Executive management support	20%
**	User involvement	15%
	Optimization	15%
	Skilled resources	13%
	Project management expertise	12%
*	Agile process	10%
**	Clear business objectives	6%
*	Emotional maturity	5%
	Execution	3%
	Tools and infrastructure	1%

Fonte: Raja Gupta. Fundamentals of Software Engineering. Engineering Handbook. 2019.

Requisitos de Software

Introdução a Engenharia de Requisitos (ER)

