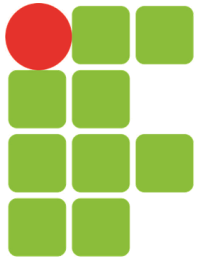


Nos Episódios Anteriores...



**INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
RIO GRANDE DO SUL
Campus Feliz

Engenharia de Software

Breve Revisão

Prof^a. Dra. [Ana Paula Lemke](#)

Contextualização

Desenvolver software (seguindo os preceitos da Engenharia de Software) é uma atividade que não deve ser confundida com escrever programas para o computador.*

**Adaptado do livro “Engenharia de Software e Sistemas de Informação”, escrito por Denis Alcides Rezende, 2005.*

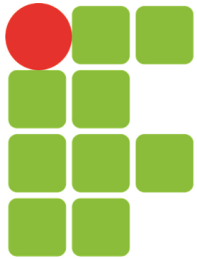


Contextualização

- O que é software?
 - Programa de computador + documentação associada
 - Podem ser desenvolvidos para um cliente específico ou para o público em geral.
- O que é a Engenharia de Software (ES)?
 - Disciplina de engenharia que se preocupa com todos os aspectos da produção de software.
- Quais são as principais atividades da ES?
 - Especificação, desenvolvimento, validação e evolução de software.

Fonte: Sommerville, I. “Engenharia de Software”, 9ª ed, 2011 (pág. 4).





**INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
RIO GRANDE DO SUL
Campus Feliz

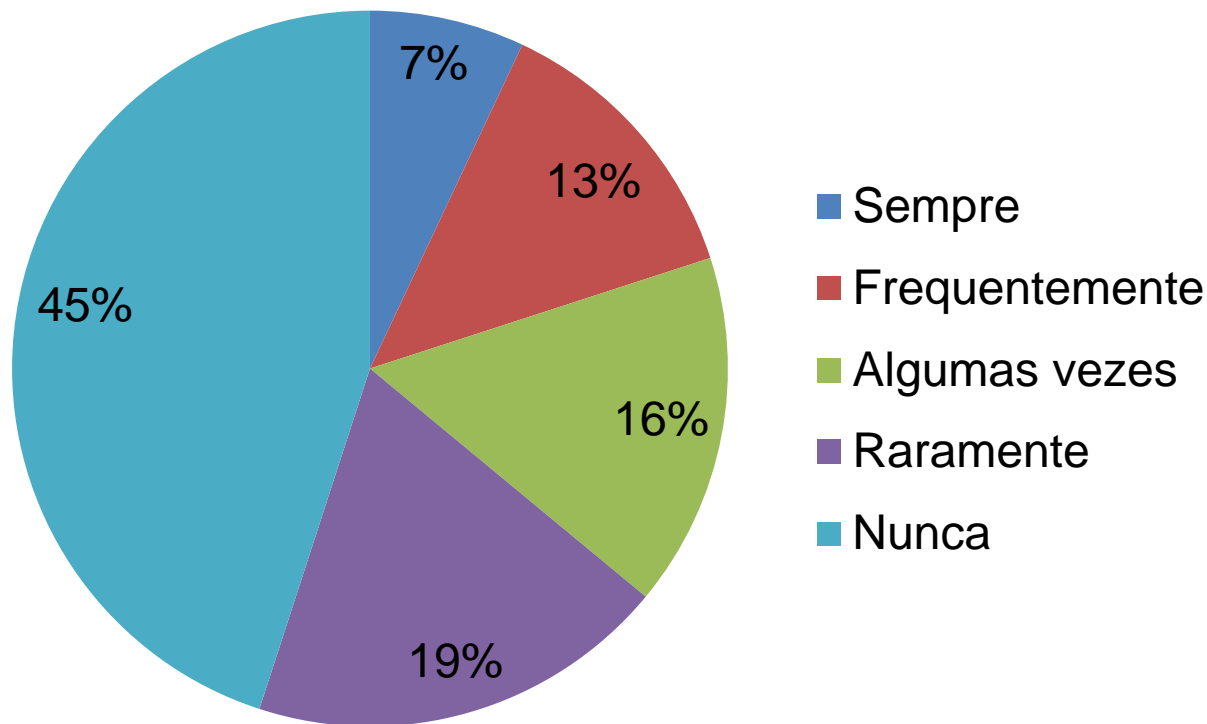
Esforços no desenvolvimento de Software

Desenvolvimento e Manutenção de Software

- Desenvolvimento (20%):
 - Início: quando a necessidade do produto é identificada;
 - Fim: quando o teste do produto implantado é concluído e o produto é entregue para a operação/produção.
- Manutenção (80%):
 - Todas as atividades após a entrega:
 - Aumento da capacidade do produto (60%)
 - Adaptação do produto a novos ambientes (20%)
 - Correção de erros (20%)



Uso das Funcionalidades do Software

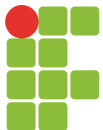


Baseado em <http://www.martinfowler.com/articles/xp2002.html> e <http://www.featuredrivendevelopment.com/node/614>.



O que é Engenharia de Software?

- O conceito de Engenharia de Software foi inicialmente proposto em 1968 (Sommerville, 2007).
- Definição amplamente utilizada:
*“Abrange um conjunto de três elementos fundamentais - **métodos, ferramentas e procedimentos** – que possibilita ao gerente controlar o processo de desenvolvimento de software e oferece ao profissional uma base para a construção produtiva de software de alta qualidade”* (Pressman, 1995).



O que é Engenharia de Software?

- Métodos
 - Proporcionam os detalhes de “como fazer” para construir o software. Envolvem um amplo conjunto de tarefas que incluem: planejamento, estimativas de projeto, análise de requisitos, etc.
- Ferramentas
 - Proporcionam apoio automatizado ou semi-automatizado aos métodos.
- Procedimentos
 - Definem a sequência em que os métodos serão aplicados, os produtos que serão disponibilizados, como será feito o controle de qualidade, entre outros.



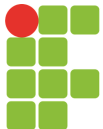
Objetivos da Engenharia de Software

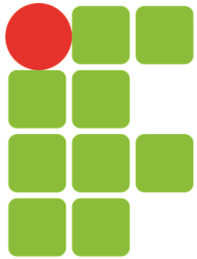
- Controle sobre o desenvolvimento de software dentro de custos, prazos e níveis de **qualidade** desejados.
- **Produtividade** no desenvolvimento, operação e manutenção de software.
 - Geralmente é mais barato, a longo prazo, usar métodos e técnicas da ES para sistemas de software, em vez de escrever os programas como se fossem algum projeto pessoal. **Para a maioria dos sistemas, a maior parte do custo é mudar o software depois que ele começa a ser usado.** (Sommerville, 2011).



Objetivos da Engenharia de Software

- O que é um software de qualidade?
 - O software que satisfaz os requisitos solicitados pelo usuário. Deve ser fácil de manter, ter boa *performance*, ser confiável e fácil de usar.
- O que é produtividade?
 - Custo de desenvolvimento reduzido:
 - A empresa consumidora quer investir pouco em software.
 - A empresa produtora tem que oferecer “software barato”.
 - Tempo de desenvolvimento reduzido:
 - Suporte rápido às necessidades do mercado.





**INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
RIO GRANDE DO SUL
Campus Feliz

Processos de Desenvolvimento de Software

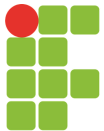
Definições

- Processo de *software*
 - “Conjunto de atividades relacionadas que levam à produção de um produto de *software*”. Sommerville, I. “Engenharia de Software”, 9º ed, 2011 (pág. 18).
 - A meta do processo pode ser o desenvolvimento de um novo produto ou a evolução de um produto existente (extensão e modificação de um produto existente).
 - “Conjunto de atividades, ações e tarefas realizadas na criação de algum produto de trabalho”. PRESSMAN, R. S. Engenharia de software: uma abordagem profissional. 7. ed. 2011. (pág. 37).



Definições

- Modelos de ciclo de vida de *software* (ou modelos de processo de software)
 - Descrição simplificada de um processo de software, que é apresentada a partir de uma perspectiva específica.



Atividades fundamentais para ES

Especificação de software: planejamento, definição dos requisitos, construção de protótipos, ...



Projeto e Implementação de software: implementação e teste unitário para atender a especificação



Validação de software: garantir que o software satisfaz o que o cliente deseja

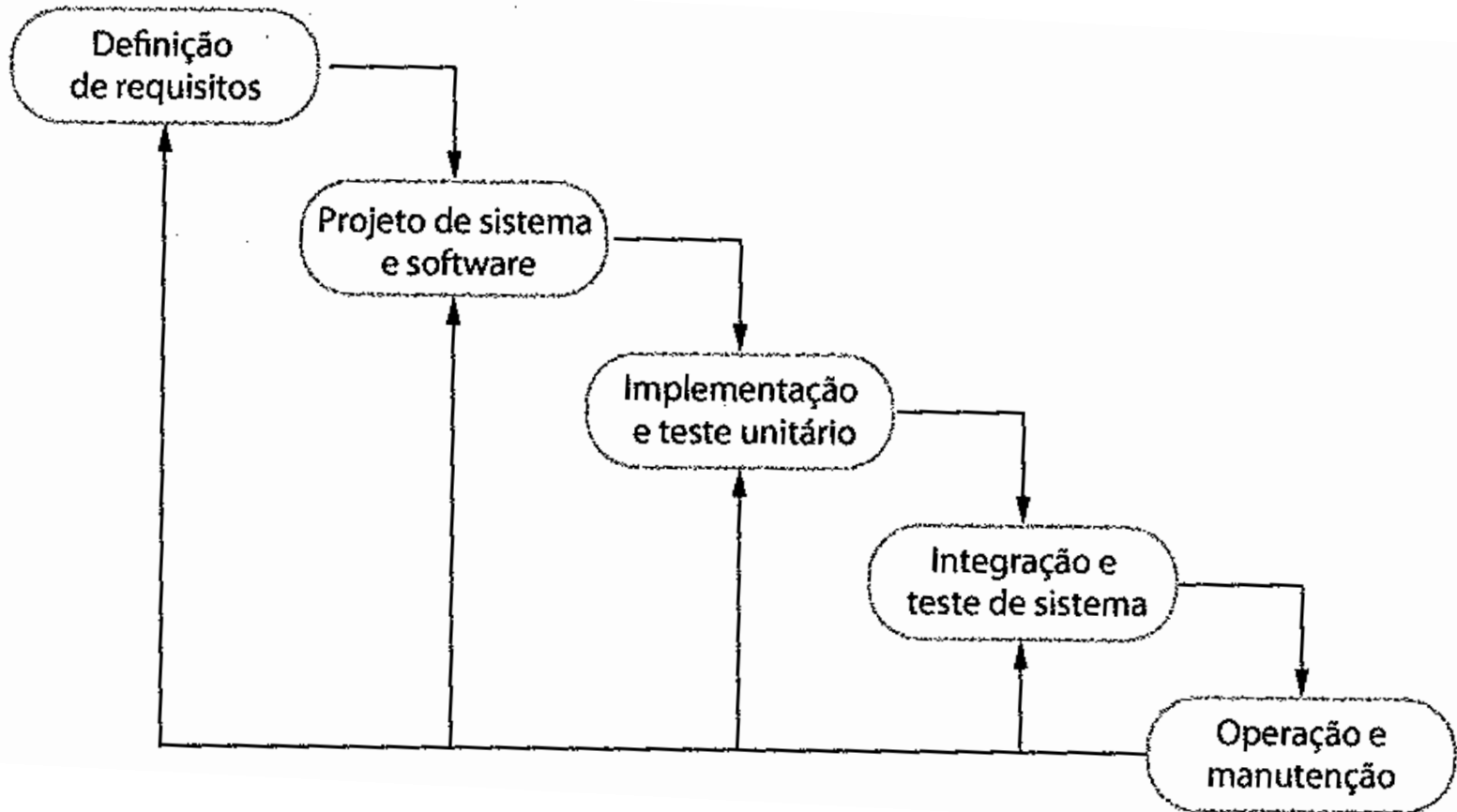


Evolução de software: para atender as necessidades mutáveis dos clientes

Baseado em Sommerville, I. “Engenharia de Software”, 9º ed, 2011 (pág. 18).



Modelo Cascata



Fonte: Sommerville, I. "Engenharia de Software", 9º ed, 2011 (pág. 20).



Modelos de processo Incremental

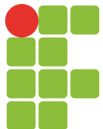
Entrega 1



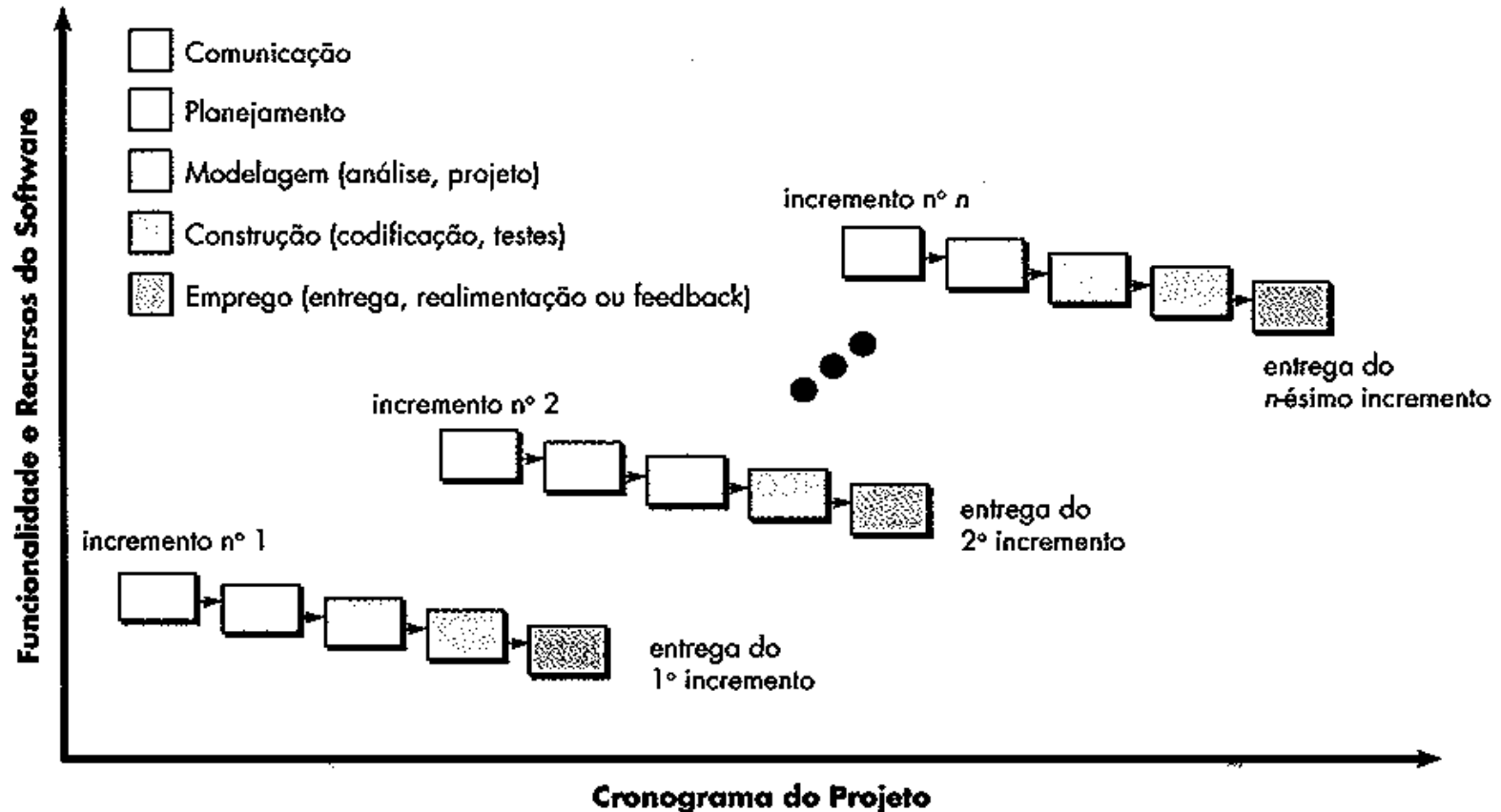
Entrega 2



Entrega 3



Modelos de processo Incremental



Fonte: PRESSMAN, R. S. Engenharia de software: uma abordagem profissional. 7. ed. 2011. (pp. 61).

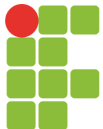


Modelos de processo Evolucionário

- São iterativos e possibilitam desenvolver versões cada vez mais completas do software.

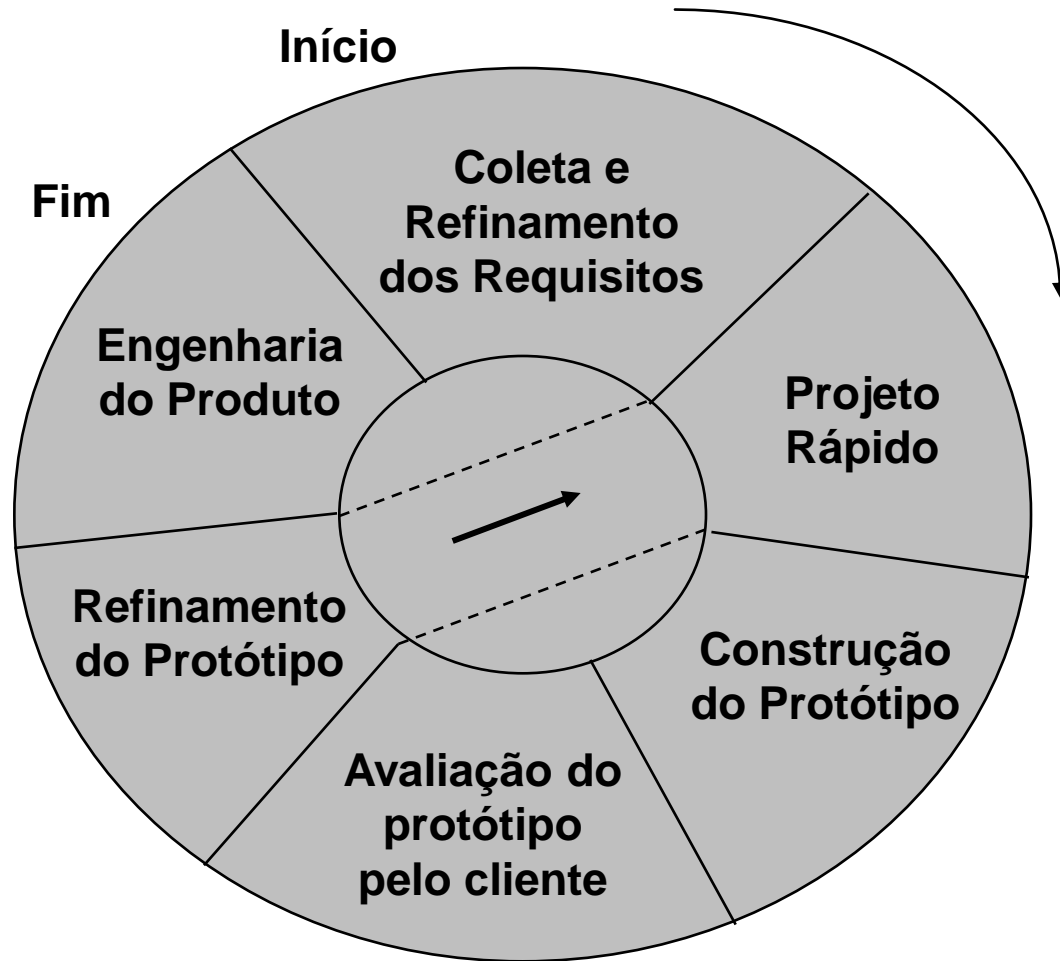


- Exemplos: **Prototipação** e **Espiral**



Modelos de processo Evolucionário

Prototipação

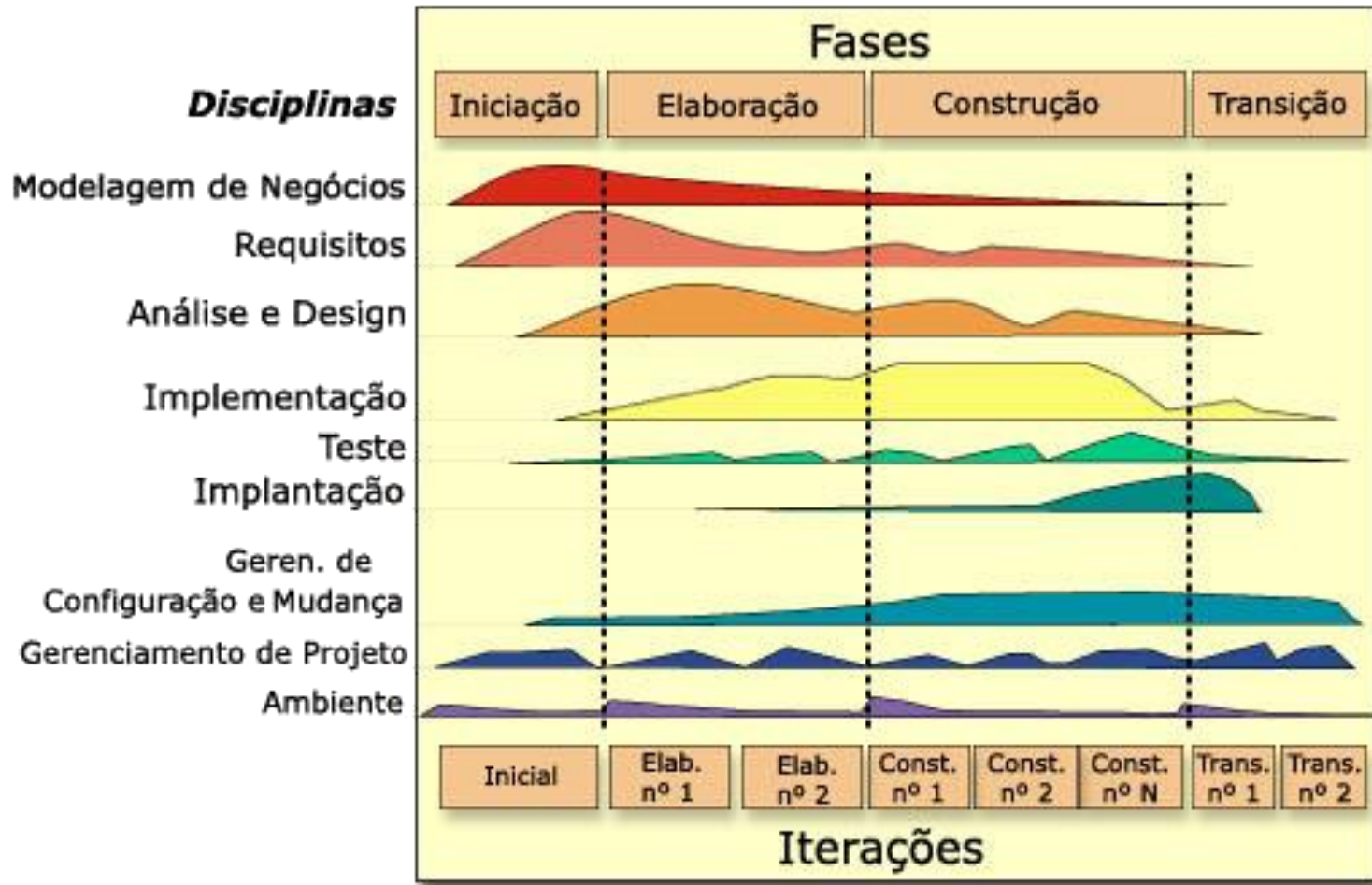


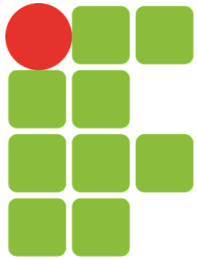
[2..5]

1. **Introduction**
 2. **Background**
 3. **Methods**
 4. **Results**
 5. **Conclusion**
 6. **References**
 7. **Appendix**
 8. **Figure**
 9. **Table**
 10. **Figure**
 11. **Table**
 12. **Figure**
 13. **Table**
 14. **Figure**
 15. **Table**
 16. **Figure**
 17. **Table**
 18. **Figure**
 19. **Table**
 20. **Figure**
 21. **Table**
 22. **Figure**
 23. **Table**
 24. **Figure**
 25. **Table**
 26. **Figure**
 27. **Table**
 28. **Figure**
 29. **Table**
 30. **Figure**
 31. **Table**
 32. **Figure**
 33. **Table**
 34. **Figure**
 35. **Table**
 36. **Figure**
 37. **Table**
 38. **Figure**
 39. **Table**
 40. **Figure**
 41. **Table**
 42. **Figure**
 43. **Table**
 44. **Figure**
 45. **Table**
 46. **Figure**
 47. **Table**
 48. **Figure**
 49. **Table**
 50. **Figure**
 51. **Table**
 52. **Figure**
 53. **Table**
 54. **Figure**
 55. **Table**
 56. **Figure**
 57. **Table**
 58. **Figure**
 59. **Table**
 60. **Figure**
 61. **Table**
 62. **Figure**
 63. **Table**
 64. **Figure**
 65. **Table**
 66. **Figure**
 67. **Table**
 68. **Figure**
 69. **Table**
 70. **Figure**
 71. **Table**
 72. **Figure**
 73. **Table**
 74. **Figure**
 75. **Table**
 76. **Figure**
 77. **Table**
 78. **Figure**
 79. **Table**
 80. **Figure**
 81. **Table**
 82. **Figure**
 83. **Table**
 84. **Figure**
 85. **Table**
 86. **Figure**
 87. **Table**
 88. **Figure**
 89. **Table**
 90. **Figure**
 91. **Table**
 92. **Figure**
 93. **Table**
 94. **Figure**
 95. **Table**
 96. **Figure**
 97. **Table**
 98. **Figure**
 99. **Table**
 100. **Figure**
 101. **Table**
 102. **Figure**
 103. **Table**
 104. **Figure**
 105. **Table**
 106. **Figure**
 107. **Table**
 108. **Figure**
 109. **Table**
 110. **Figure**
 111. **Table**
 112. **Figure**
 113. **Table**
 114. **Figure**
 115. **Table**
 116. **Figure**
 117. **Table**
 118. **Figure**
 119. **Table**
 120. **Figure**
 121. **Table**
 122. **Figure**
 123. **Table**
 124. **Figure**
 125. **Table**
 126. **Figure**
 127. **Table**
 128. **Figure**
 129. **Table**
 130. **Figure**
 131. **Table**
 132. **Figure**
 133. **Table**
 134. **Figure**
 135. **Table**
 136. **Figure**
 137. **Table**
 138. **Figure**
 139. **Table**
 140. **Figure**
 141. **Table**
 142. **Figure**
 143. **Table**
 144. **Figure**
 145. **Table**
 146. **Figure**
 147. **Table**
 148. **Figure**
 149. **Table**
 150. **Figure**
 151. **Table**
 152. **Figure**
 153. **Table**
 154. **Figure**
 155. **Table**
 156. **Figure**
 157. **Table**
 158. **Figure**
 159. **Table**
 160. **Figure**
 161. **Table**
 162. **Figure**
 163. **Table**
 164. **Figure**
 165. **Table**
 166. **Figure**
 167. **Table**
 168. **Figure**
 169. **Table**
 170. **Figure**
 171. **Table**
 172. **Figure**
 173. **Table**
 174. **Figure**
 175. **Table**
 176. **Figure**
 177. **Table**
 178. **Figure**
 179. **Table**
 180. **Figure**
 181. **Table**
 182. **Figure**
 183. **Table**
 184. **Figure**
 185. **Table**
 186. **Figure**
 187. **Table**
 188. **Figure**
 189. **Table**
 190. **Figure**
 191. **Table**
 192. **Figure**
 193. **Table**
 194. **Figure**
 195. **Table**
 196. **Figure**
 197. **Table**
 198. **Figure**
 199. **Table**
 200. **Figure**
 201. **Table**
 202. **Figure**
 203. **Table**
 204. **Figure**
 205. **Table**
 206. **Figure**
 207. **Table**
 208. **Figure**
 209. **Table**
 210. **Figure**
 211. **Table**
 212. **Figure**
 213. **Table**
 214. **Figure**
 215. **Table**
 216. **Figure**
 217. **Table**
 218. **Figure**
 219. **Table**
 220. **Figure**
 221. **Table**
 222. **Figure**
 223. **Table**
 224. **Figure**
 225. **Table**
 226. **Figure**
 227. **Table**
 228. **Figure**
 229. **Table**
 230. **Figure**
 231. **Table**
 232. **Figure**
 233. **Table**
 234. **Figure**
 235. **Table**
 236. **Figure**
 237. **Table**
 238. **Figure**
 239. **Table**
 240. **Figure**
 241. **Table**
 242. **Figure**
 243. **Table**
 244. **Figure**
 245. **Table**
 246. **Figure**
 247. **Table**
 248. **Figure**
 249. **Table**
 250. **Figure**
 251. **Table**
 252. **Figure**
 253. **Table**
 254. **Figure**
 255. **Table**
 256. **Figure**
 257. **Table**
 258. **Figure**
 259. **Table**
 260. **Figure**
 261. **Table**
 262. **Figure**
 263.



Processo Unificado - RUP





**INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
RIO GRANDE DO SUL
Campus Feliz

Desenvolvimento Ágil de Software

Desenvolvimento Ágil de Software

Manifesto Ágil

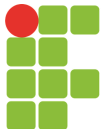
- Valores Fundamentais do Manifesto Ágil:

Indivíduos e interações ao invés de processos e ferramentas.

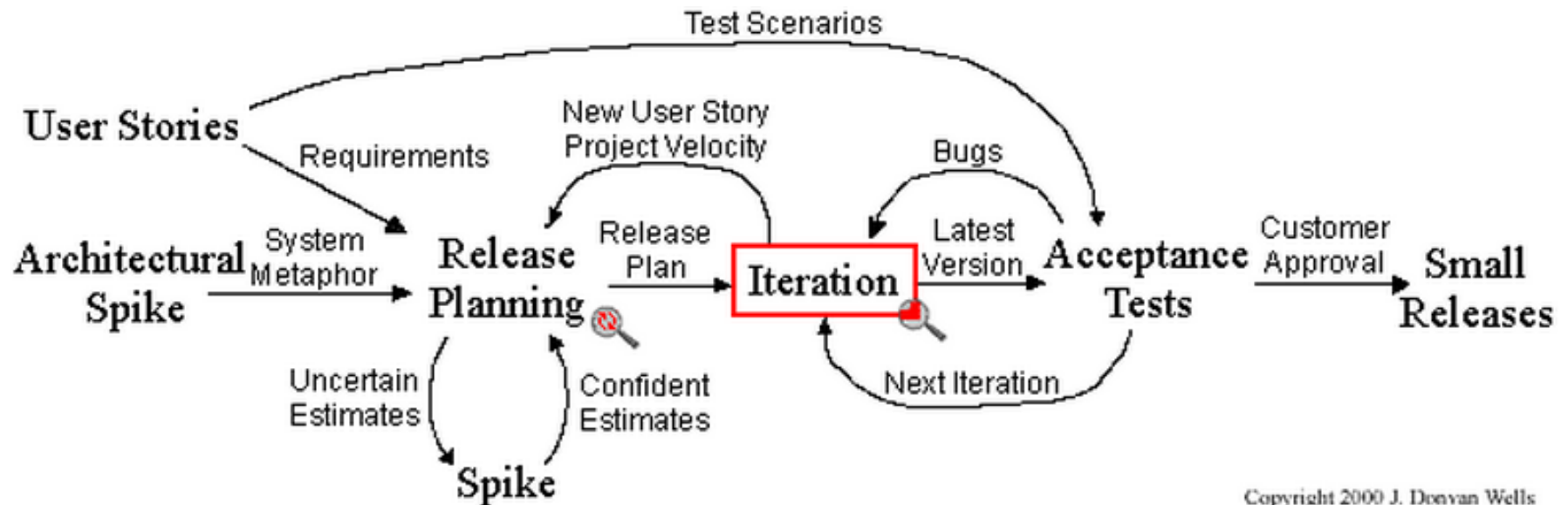
Software funcionando ao invés de documentação abrangente.

Colaboração com o cliente ao invés de negociação de contratos.

Resposta a mudanças ao invés de obediência a um plano.



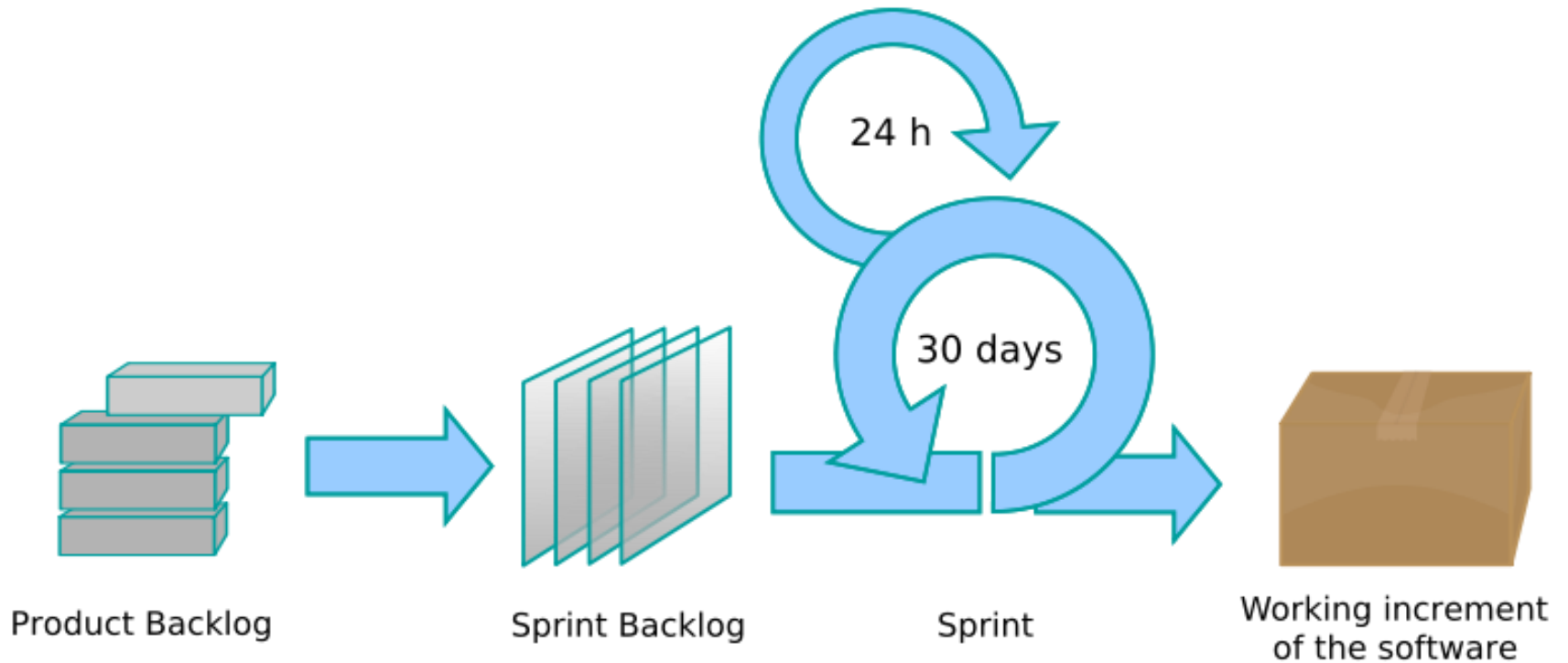
Extreme Programming – XP

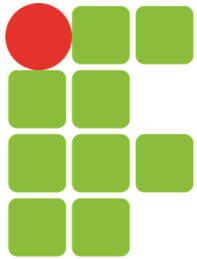


Fonte: <http://www.extremeprogramming.org/map/project.html>.



Scrum





**INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
RIO GRANDE DO SUL
Campus Feliz

Requisitos de Software

(visão tradicional)

Contextualização - Requisitos

- Visão histórica:
 - Requisitos são funções.
- Visão atual:
 - Requisitos são objetivos, funções, propriedades, restrições que o sistema deve possuir/obedecer para satisfazer contratos, padrões ou especificações de acordo com o(s) usuário(s);
 - Requisitos são as funcionalidades e condições que o sistema – e mais genérico, o projeto – deve estar em conformidade.

Os requisitos do sistema são influenciados pelas preferências, recusas e preconceitos dos usuários, além das questões políticas e organizacionais (Sommerville, 2007).



Requisitos – Níveis de descrição [1..3]

- **Requisitos de usuários:** requisitos abstratos de alto nível.
 - São declarações, em uma linguagem natural com diagramas, de quais serviços são esperados do sistema e as restrições as quais ele deve operar.
- **Requisitos de sistema:** descrição detalhada de o que o sistema deve fazer.
 - Definem, detalhadamente, as funções, serviços e as restrições operacionais do sistema. O documento de requisitos, chamado de **especificação funcional**, pode servir como um contrato entre cliente e desenvolvedor.

Baseado em Sommerville, I. “Engenharia de Software”, 9º ed, 2011 (pág. 58).



Requisitos - Classificação

- **Requisitos Funcionais**: são declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações. Em alguns casos, os requisitos funcionais podem também estabelecer explicitamente o que o sistema não deve fazer.
- **Requisitos não Funcionais**: são restrições aos serviços ou funções oferecidos pelo sistema. Eles incluem restrições de *timing*, restrições no processo de desenvolvimento e restrições impostas pelas normas. Frequentemente aplicam-se sobre o sistema com um todo.

Fonte: Sommerville, I. “Engenharia de Software”, 9º ed, 2011 (pág. 59).



Requisitos Não Funcionais

- Métricas para especificar requisitos não funcionais

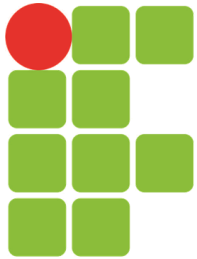
Propriedade	Medida
Velocidade	Transações processadas/segundo Tempo de resposta de usuário/evento Tempo de atualização de tela
Tamanho	Megabytes Número de chips de memória
Facilidade de uso	Tempo de treinamento Número de frames de ajuda
Confiabilidade	Tempo médio para falha Probabilidade de indisponibilidade Taxa de ocorrência de falhas Disponibilidade
Robustez	Tempo de reinício após falha Percentual de eventos que causam falhas Probabilidade de corrupção de dados em caso de falha
Portabilidade	Percentual de declarações dependentes do sistema-alvo Número de sistemas-alvo



O que **NÃO** faz parte de um Requisito?

- **Design:** como o software atenderá aos requisitos
- **Verificação:** como você sabe que o requisito foi atendido
- **Dados gerenciais do projeto:** cronogramas, controles, etc.

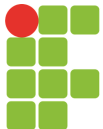
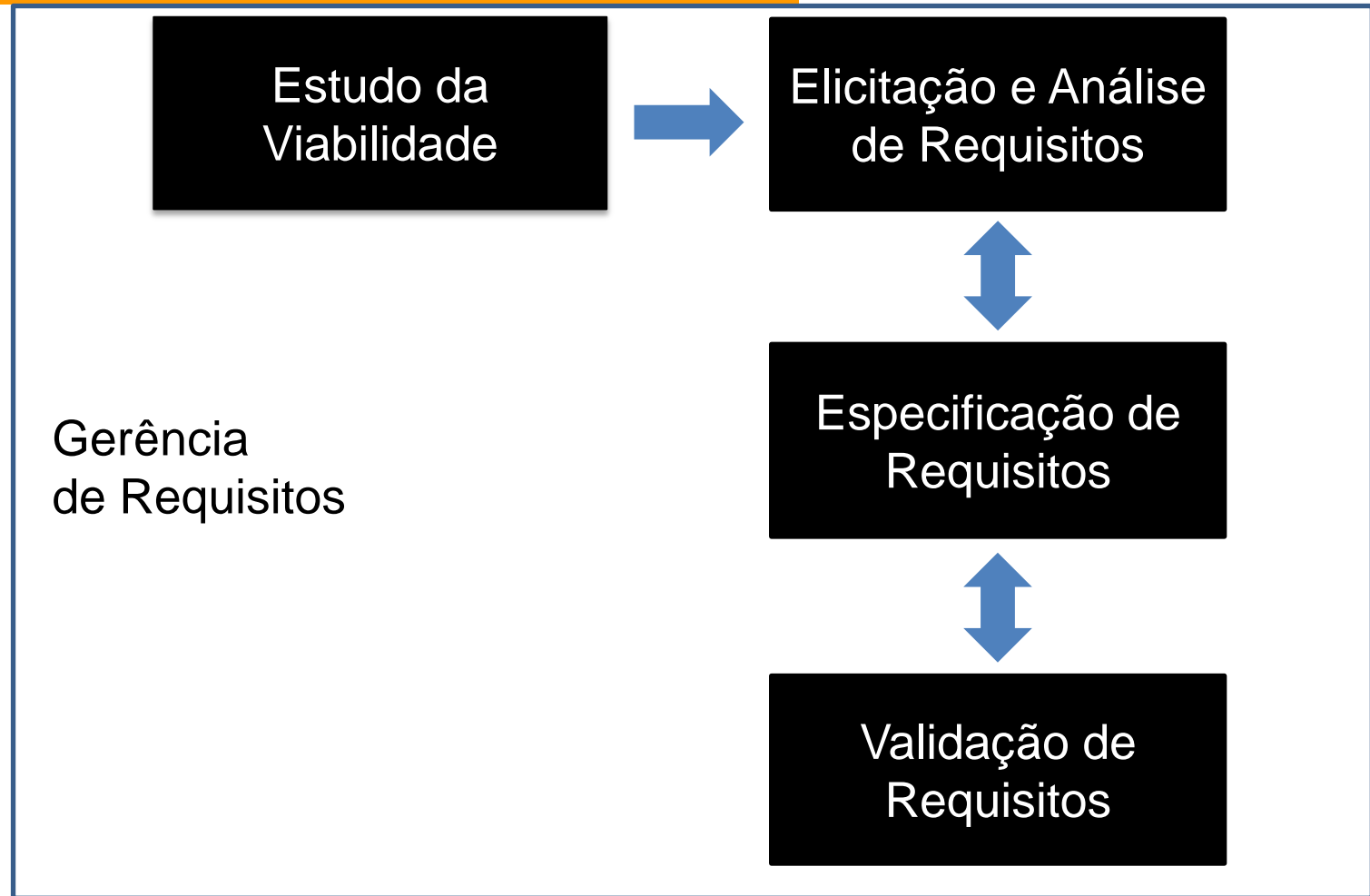




**INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
RIO GRANDE DO SUL
Campus Feliz

Engenharia de Requisitos

Engenharia de Requisitos



Dimensões da Engenharia de Requisitos



- Compreensão Vaga



- Compreensão mais completa e clara do sistema



- Pontos de vistas diferentes



- Consenso negociado



- Representação informal



- Representação Formal



Elicitação e Análise de Requisitos

Descoberta de Requisitos

- Técnicas/práticas para obtenção de requisitos:
 - *Brainstorming* e redução de ideias
 - *Storyboards*
 - **Entrevistas**
 - Questionários
 - *Workshop* de requisitos
 - **Etnografia** (observação)
 - Protótipos
 - Revisão de especificações de requisitos do cliente
 - **Leitura de documentos**

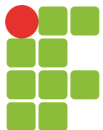


Especificação de Requisitos

Documento de requisitos de software ou
Especificação de requisitos de software ou
SRS – *Software Requirements Specification*

- É a declaração oficial do que os desenvolvedores do sistema devem implementar. Ele deve incluir os requisitos de usuário de um sistema e uma especificação detalhada dos requisitos do sistema.
 - Em alguns casos, os requisitos de usuário e de sistema podem ser integrados em uma única descrição.
 - Em outros, os requisitos de usuários são apresentados antes dos requisitos de sistema, introduzindo-os.

Fonte: Sommerville, 2007.



Qualidades de um SRS

- Correto
- Completo
- Consistente
- Unambíguo
- Verificável
- Priorizável
- Modificável
- Rastreável



Atividade

