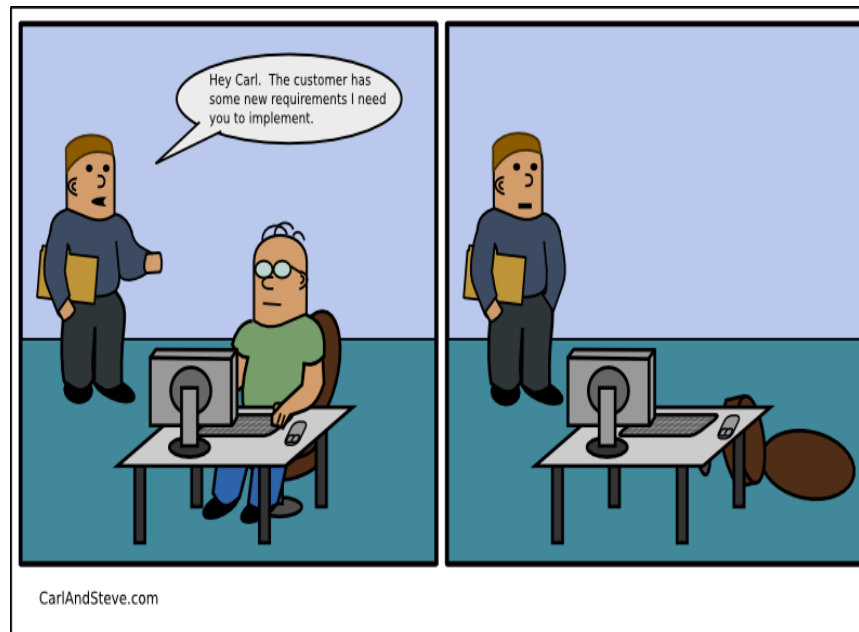


# Introdução aos métodos ageis

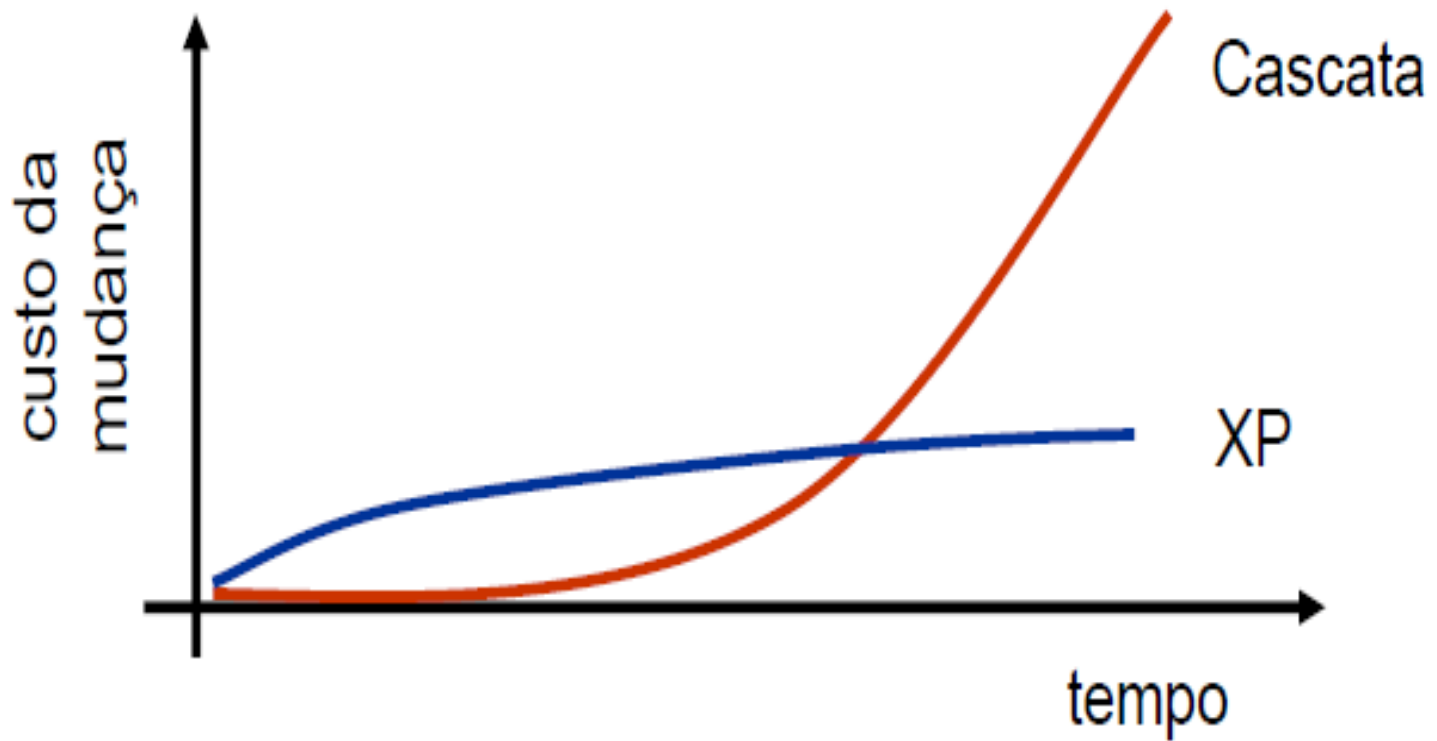
Prof. Marcelo Werneck

# Introdução

- ✓ Desenvolvimento iterativo e evolutivo.
- ✓ Entregas evolutivas.
- ✓ Valores e práticas incentivam agilidade.
- ✓ Rápida e flexível resposta a mudanças.



# Custo das Mudanças



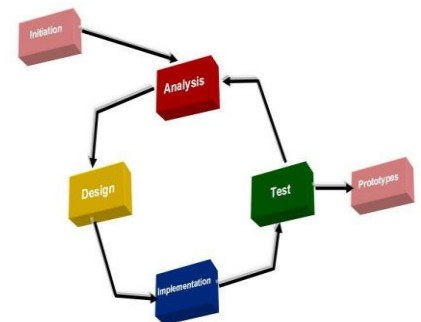
# Introdução

- ✓ Difícil definir métodos ágeis.
  - ✓ Práticas específicas variam.
- ✓ Características comuns:
  - ✓ Iterações curtas limitadas no tempo;
  - ✓ Refinamento evolutivo dos planos;



# Modelos Evolutivos

- ✓ Uma versão reduzida do software pode ser elaborada e incrementada.
- ✓ Um conjunto de requisitos básico é bem entendido mas detalhes ainda precisam ser esclarecidos.
- ✓ São iterativos. Permitem desenvolver versões cada vez mais completas do software.
- ✓ Adequado para acompanhar um produto que evolui com o tempo.



# Métodos Ágeis

- ✓ Modelagem Ágil é uma atitude, não um processo prescritivo de software.
- ✓ Compreende um conjunto de valores e princípios
- ✓ Quando utilizada em ambientes ágeis, resulta em software de melhor qualidade e de desenvolvimento mais rápido.
- ✓ Não é uma solução completa. Apresenta como suprimento aos métodos existentes.
- ✓ Visam tornar menos burocráticos os processos de desenvolvimento

# Motivações

- ✓ Princípio da Incerteza.
  - ✓ Incerteza é inerente e inevitável em projetos e processos de software
- ✓ Projetos de software geralmente possuem muitas solicitações de mudança.
- ✓ Projetos de tamanho médio possuem taxas de 25% e de tamanho grande 35% de mudança.

**CHANGE IS INEVITABLE.  
CHANGE IS CONSTANT.**

Benjamin Disraeli | (c) thomasingeekzumblo.com

# Motivações para Desenvolvimento Iterativo

- ✓ Possui baixo risco;
- ✓ Descoberta e mitigação dos riscos com antecedência;
- ✓ Produto parcial obtido mais cedo;
- ✓ Acompanhamento do progresso;
- ✓ Maior qualidade;
- ✓ Produto final atende melhor a necessidades do cliente;
- ✓ Acomoda mudanças

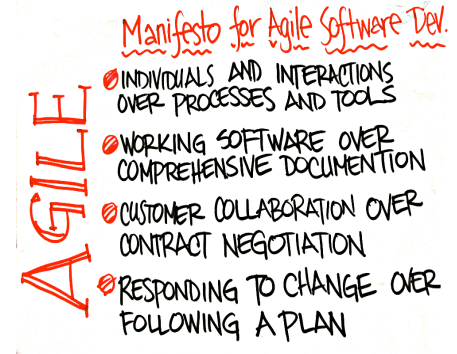


# Manifesto Ágil

- ✓ Aliança Ágil ([www.agilealliance.com](http://www.agilealliance.com))
- ✓ Formada em 2001
- ✓ Reunião de grupo de profissionais experientes com motivação de melhorar o desenvolvimento de software
- ✓ Criaram o Manifesto para o Desenvolvimento Ágil.



# Manifesto Ágil



- ✓ Passam a valorizar
  - ✓ Indivíduos e interação entre eles mais que processos e ferramentas;
  - ✓ Software em funcionamento mais que documentação abrangente;
  - ✓ Colaboração com o cliente mais que negociação de contratos;
  - ✓ Responder a mudanças mais que seguir um plano.

# Manifesto Ágil – 12 Princípios

- ✓ 1. A prioridade é satisfazer ao cliente através de entregas contínuas e freqüentes;
- ✓ 2. Receber bem as mudanças de requisitos, mesmo em uma fase avançada do projeto;
- ✓ 3. Entregas com freqüência, sempre na menor escala de tempo;
- ✓ 4. As equipes de negócio e de desenvolvimento devem trabalhar juntas diariamente;

# Manifesto Ágil – 12 Princípios

- ✓ 5. Manter uma equipe motivada fornecendo ambiente, apoio e confiança necessários;
- ✓ 6. A maneira mais eficiente da informação circular através de uma conversa cara-a-cara;
- ✓ 7. Ter o sistema funcionando é a melhor medida de progresso;
- ✓ 8. Processos ágeis promovem o desenvolvimento sustentável;

# Manifesto Ágil – 12 Princípios

- ✓ 9. Atenção contínua a excelência técnica e a um bom projeto aumentam a agilidade;
- ✓ 10. Simplicidade é essencial;
- ✓ 11. As melhores arquiteturas, requisitos e projetos provêm de equipes organizadas;
- ✓ 12. Em intervalos regulares, a equipe deve refletir sobre como se tornar mais eficaz.

# Quando não usar ágil?

- ✓ Quando você precisa formalmente documentar cada fase do ciclo de vida (requisitos, projeto, implementação, testes).
- ✓ Entregáveis devem passar por um conjunto de aprovações.
- ✓ Necessário seguir processo tradicional.
- ✓ Usuários não estão sempre acessíveis.
- ✓ Equipe não é qualificada.
- ✓ Abordagem deve ser sequencial (cascata).

# Modelos prescritivos x ágeis

- ✓ Modelos convencionais adotam a estratégia de previsibilidade.
- ✓ Tenta-se levantar todos os requisitos e compreender o domínio do problema antes de dar início ao processo de desenvolvimento.
- ✓ Após o levantamento dos requisitos, é feito um planejamento para que as mudanças possam ser controladas no decorrer do processo de desenvolvimento do software.

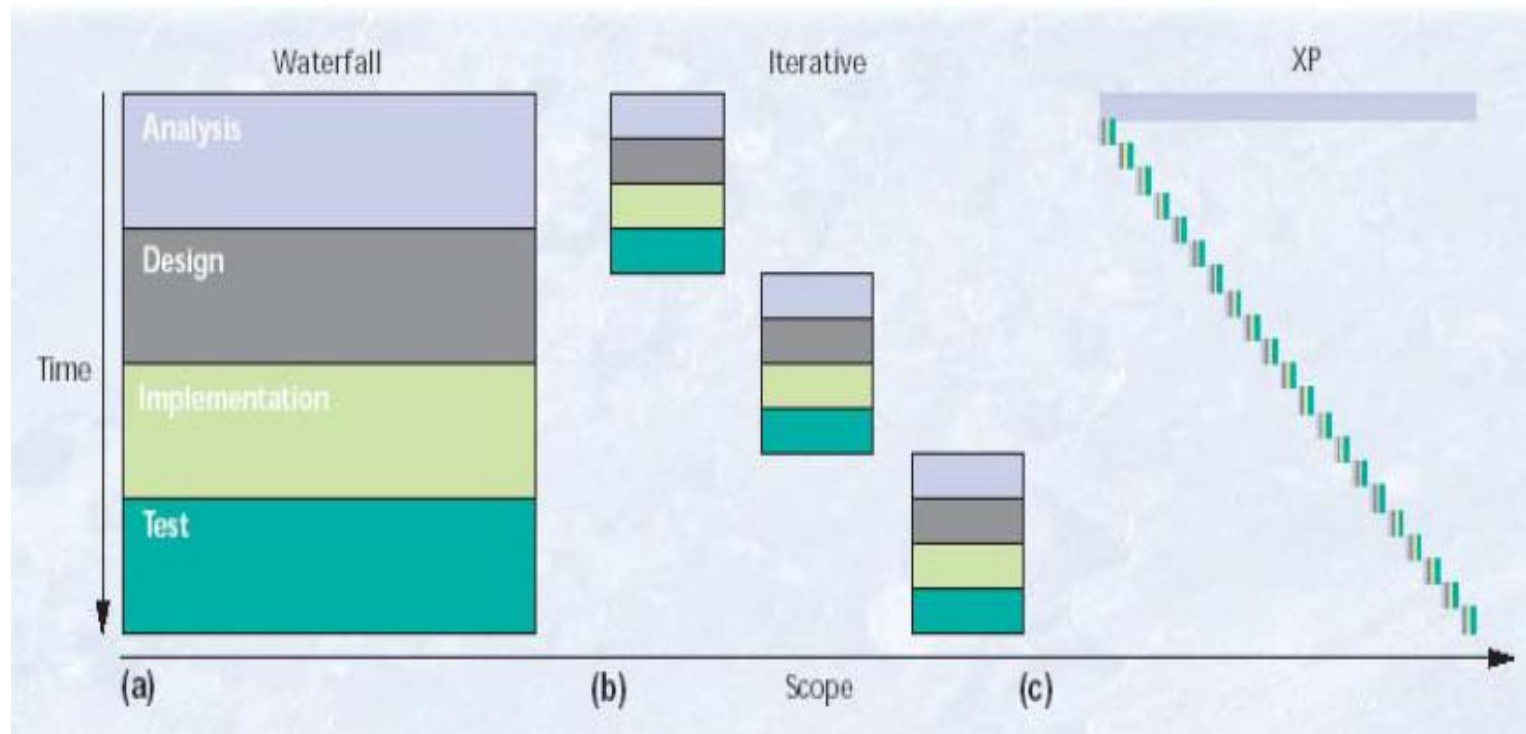
# Modelos prescritivos x ágeis

- ✓ Métodos ágeis optam pela adaptabilidade.
- ✓ Buscam levantar aos poucos os requisitos e o planejamento é contínuo, para que a adaptação às mudanças possa ocorrer em conformidade com o processo
- ✓ Enfatizam os aspectos humanos do desenvolvimento de software ao invés dos aspectos de Engenharia.
- ✓ Não são centrados nos artefatos.
- ✓ Optam por uma documentação apropriada para evitar redundâncias e excessos

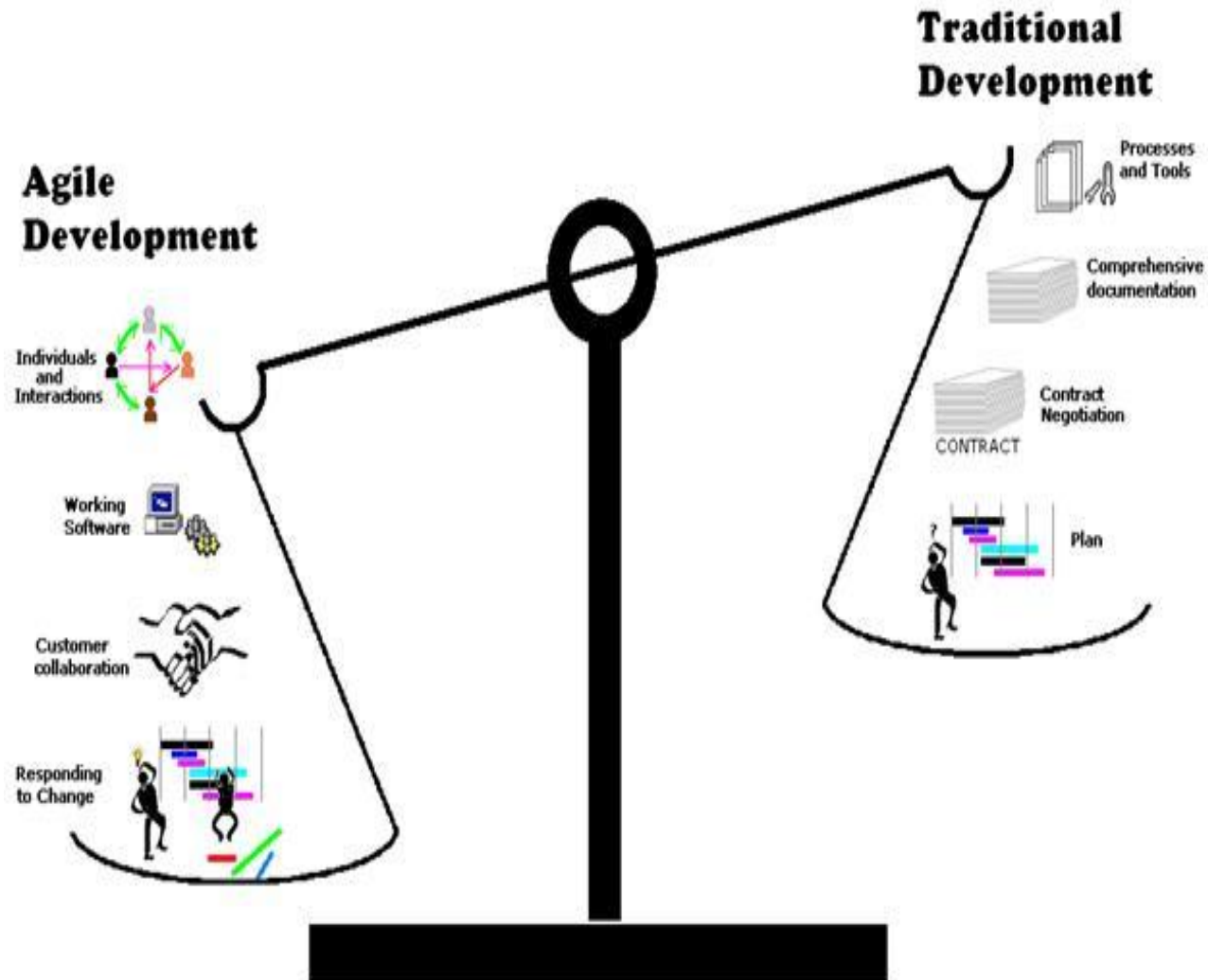


# Modelos prescritivos x ágeis

- (a) Cascata: Longo ciclo de desenvolvimento
- (b) Espiral: Iterações
- (c) XP: Iterações curtas

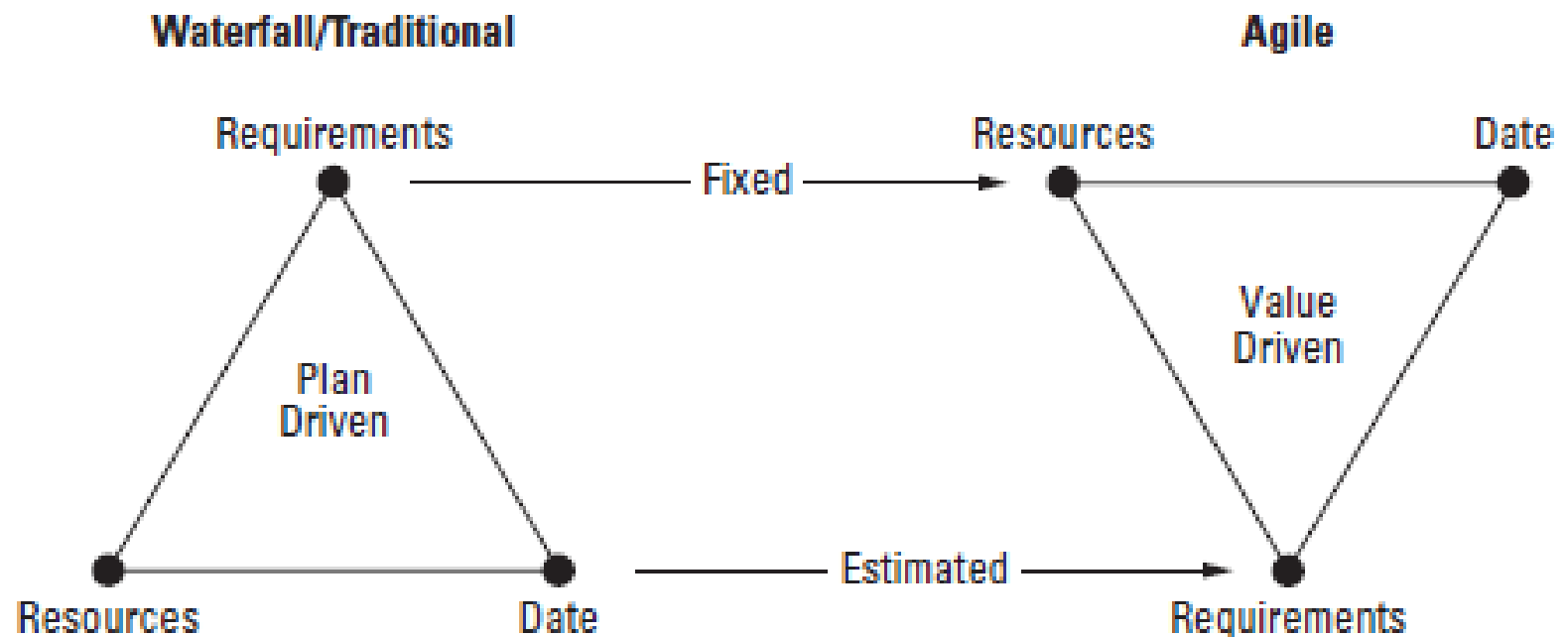


# Modelos prescritivos x ágeis



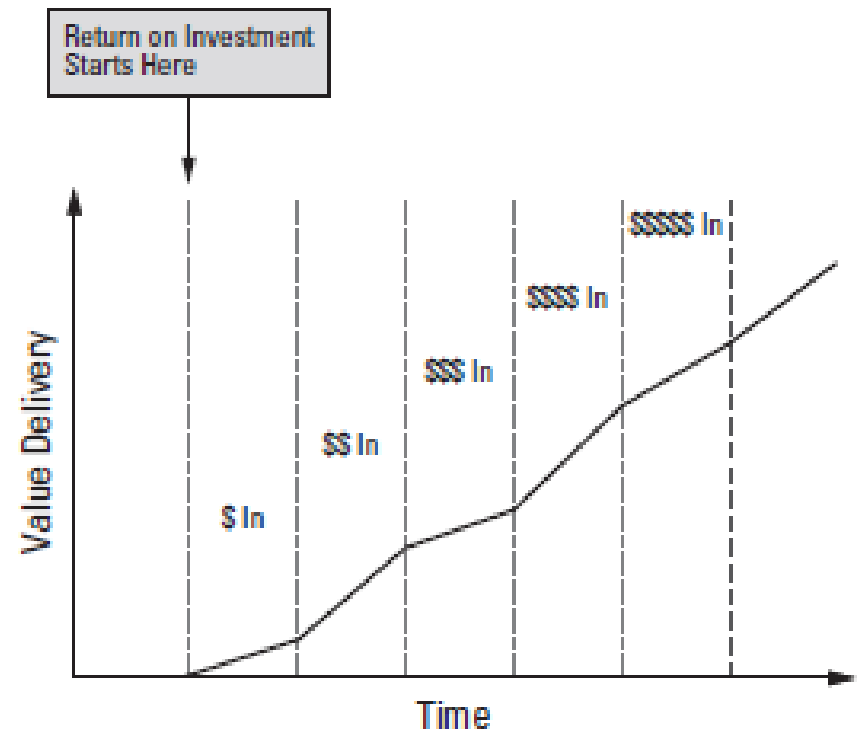
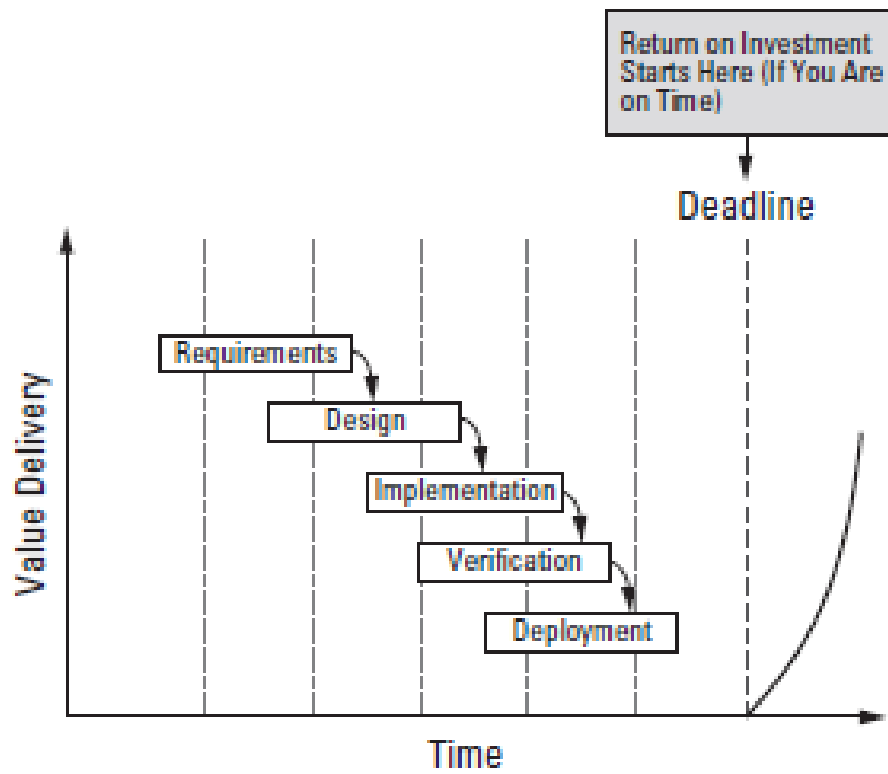
# Modelos prescritivos x ágeis

- Do ponto de vista dos requisitos



# Modelos prescritivos x ágeis

- Do ponto de vista das entregas e ROI



# Modelos convencionais x ágeis

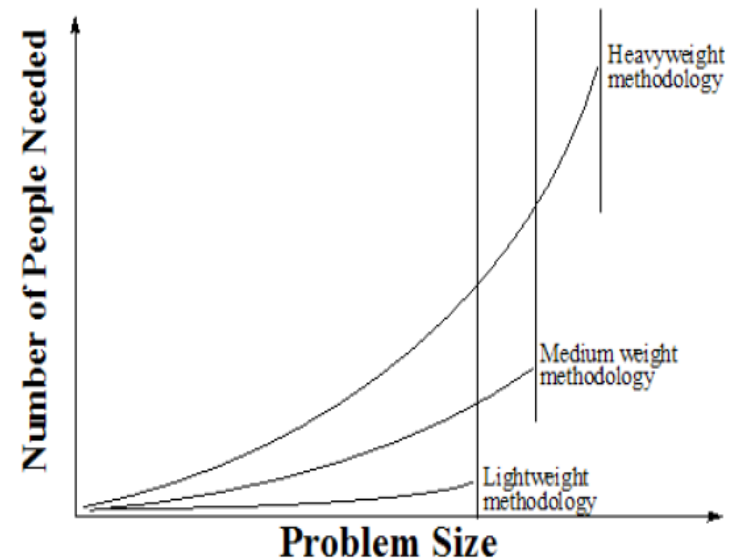
|  | Traditional   | Agile   |
|--|---|---|
| <b>Fundamental Assumptions</b>               | Systems are fully specifiable, predictable, and can be built through meticulous and extensive planning. | High-quality, adaptive software can be developed by small teams using the principles of continuous design improvement and testing based on rapid feedback and change. |
| <b>Control</b>                               | Process centric   | People centric  |
| <b>Management Style</b>                      | Command-and-control   | Leadership-and-collaboration  |
| <b>Knowledge Management</b>                  | Explicit  | Tacit   |
| <b>Role Assignment</b>                       | Individual—favors specialization  | Self-organizing teams—encourages role interchangeability  |
| <b>Communication</b>                         | Formal  | Informal  |
| <b>Customer's Role</b>                       | Important   | Critical  |
| <b>Project Cycle</b>                         | Guided by tasks or activities   | Guided by product features  |
| <b>Development Model</b>                     | Life cycle model (Waterfall, Spiral, or some variation)   | The evolutionary-delivery model   |
| <b>Desired Organizational Form/Structure</b> | Mechanistic (bureaucratic with high formalization)  | Organic (flexible and participative encouraging cooperative social action)  |
| <b>Technology</b>                            | No restriction  | Favors object-oriented technology   |

# Modelos convencionais x ágeis

|                              | <b>Agile Methods</b>      | <b>Heavy Methods</b>  |
|------------------------------|---------------------------|-----------------------|
| <b>Approach</b>              | Adaptive                  | Predictive            |
| <b>Success Measurement</b>   | Business Value            | Conformation to plan  |
| <b>Project size</b>          | Small                     | Large                 |
| <b>Management Style</b>      | Decentralized             | Autocratic            |
| <b>Perspective to Change</b> | Change Adaptability       | Change Sustainability |
| <b>Culture</b>               | Leadership-Collaboration  | Command-Control       |
| <b>Documentation</b>         | Low                       | Heavy                 |
| <b>Emphasis</b>              | People-Oriented           | Process-Oriented      |
| <b>Cycles</b>                | Numerous                  | Limited               |
| <b>Domain</b>                | Unpredictable/Exploratory | Predictable           |
| <b>Upfront Planning</b>      | Minimal                   | Comprehensive         |
| <b>Return on Investment</b>  | Early in Project          | End of Project        |
| <b>Team Size</b>             | Small/Creative            | Large                 |

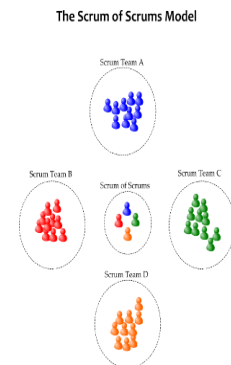
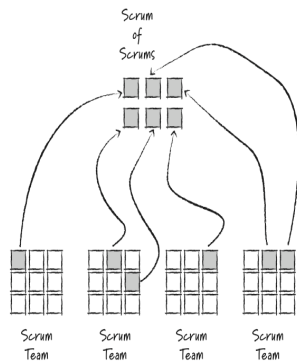
# Tamanho do Projeto

- ✓ Metodologia ágil – times pequenos.
- ✓ Metodologia tradicional – muitas pessoas são necessárias.
- ✓ “Há um limite para o tamanho do problema que pode ser resolvido com um dado número de pessoas”.



# Tamanho do Projeto

- ✓ O nível da metodologia aumenta na medida em que o número de pessoas cresce.
- ✓ Mais difícil usar métodos ágeis com equipes maiores que 40 pessoas.
- ✓ Metodologias tradicionais melhores para times grandes.
- ✓ Equipes grandes podem ser divididas em equipes menores (“scrums of scrums” - Ken Schwaber)





# Fatores Humanos

- ✓ Pessoas capacitadas e com experiência são fator chave para metodologias ágeis.
- ✓ Comprometimento e disponibilidade do cliente – cliente monitora progresso e solicita mudanças.
- ✓ Processo fica mais atraente.
- ✓ Cultura da organização é outro fator importante ao se escolher uma metodologia.

# Fatores de Risco

- ✓ Métodos ágeis são usados em aplicações que podem ser construídas rapidamente e não necessitam garantia extensiva da qualidade.
- ✓ Sistemas críticos que demandam alta confiabilidade e segurança são mais adequados para uma metodologia prescritiva.
  - ✓ Requisitos devem ser muito bem definidos antes do desenvolvimento do software.



# Resultados do uso do Ágil - Pesquisas

- ✓ 84.7% dos respondentes classificaram seu conhecimento em Ágil como mediano ou maior;
- ✓ 49% afirmaram que seus custos reduziram ou significativamente reduziram;
- ✓ 46% afirmaram que custos não foram modificados;
- ✓ Somente 5% afirmaram que houve um efeito negativo no custo;
- ✓ 93% afirmaram que a produtividade foi melhor ou significativamente melhor;
- ✓ 88% afirmaram que qualidade aumentou ou significativamente aumentou
- ✓ 83% afirmaram que satisfação do negócio foi melhor ou significativamente melhor
- ✓ Fonte: Shine Technologies

# Conclusão

- ✓ Quanto mais as características e condições do projeto diferem das recomendadas pelas metodologias, mais risco existirá ao se adotar tal abordagem.
- ✓ Não existe metodologia boa ou ruim.
- ✓ Importante conhecer a organização para melhor selecionar a metodologia mais adequada.