

AGILE DEVELOPMENT

The Art of Agile Development – James Shore e Shane Warden
Aula de Luiz Eduardo Guarino de Vasconcelos

Introdução

- Pratique
- Perfeição vem da experiência
- Siga as práticas e veja o que acontece
- Continue praticando até as práticas se tornarem fáceis
- Não quebre regras. Faça-as, repita-as. Só quebre as regras se tiver experiência e certeza.

Agenda

- Introdução
- O Manifesto Ágil
- 12 Princípios Aliança Ágil
 - Valores e Práticas Ágeis
 - Por que Ser Ágil?
 - Como Ser Ágil?
- Compreendendo Melhor
- Adotando a Metodologia

Introdução

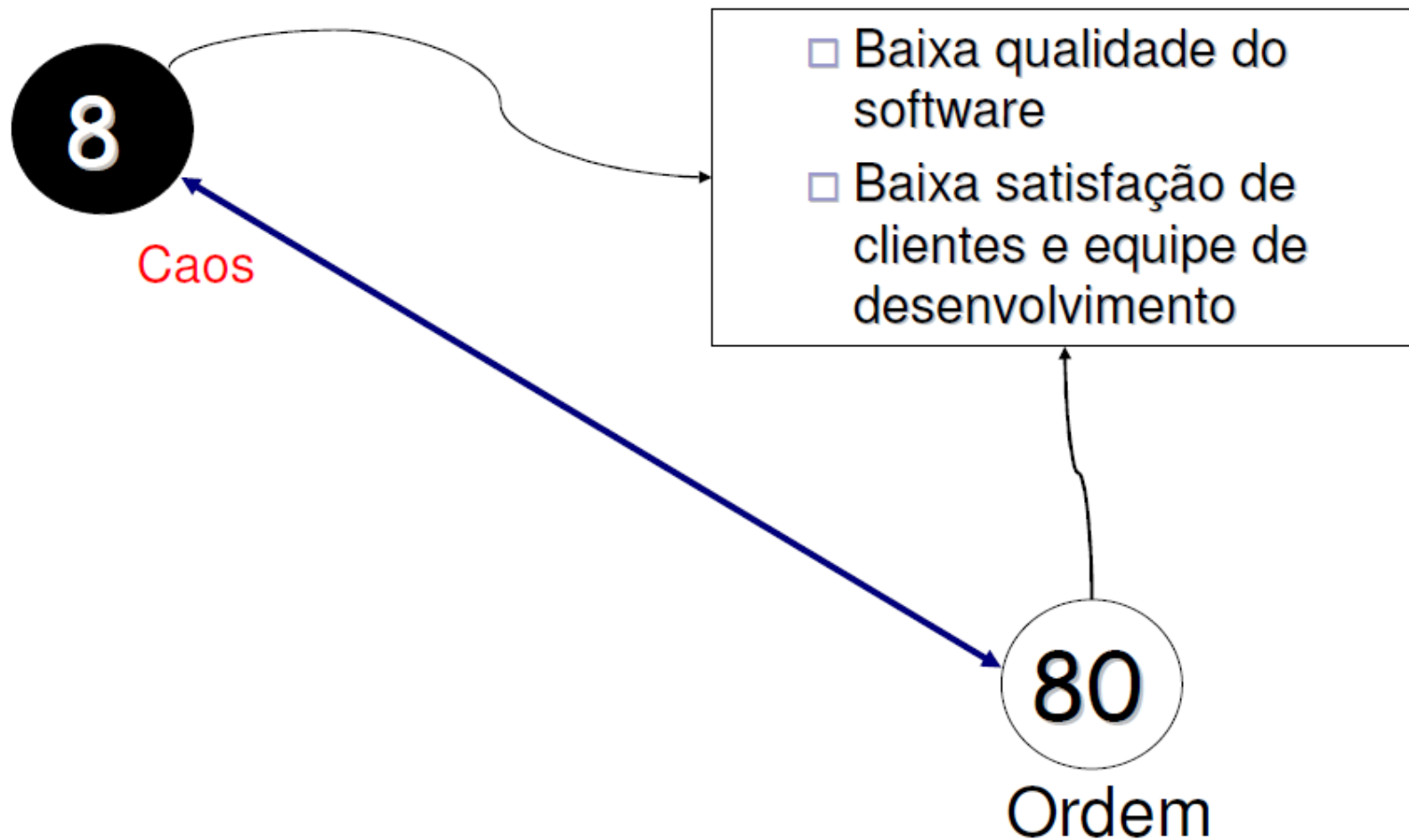
- Início do desenvolvimento de software: CAOS!
- Cada um fazia do seu jeito!
- Resultados
 - ▣ Software de baixa qualidade
 - ▣ Softwares atrasados, cancelados
 - ▣ Softwares com orçamento estourado
 - ▣ Mal estar entre desenvolvedores e clientes

Introdução

- Necessidade de Organização: ORDEM
- A busca pela Engenharia de Software
- Processos Descritivos de Software com deficiência
 - ▣ RUP, Cascata, etc...
 - ▣ Eles esquecem as fragilidades das pessoas que constroem softwares de computador
- Resultados
 - ▣ Software de baixa qualidade
 - ▣ Mal estar entre desenvolvedores e clientes

Introdução

- Qual a solução?



Motivação

- É difícil prever quais requisitos de software vão persistir e quais serão modificados.
 - ▣ Condições de mercado mudam
 - ▣ Necessidades dos usuários finais evoluem
 - ▣ Novas ameaças de competições emergem sem alerta
- Análise, projeto, construção e testes não são tão previsíveis como gostaríamos.
- Engenheiros de Software devem ser ágeis o suficiente para responderem a um ambiente de negócios mutante

Como ser ágil?

- Manifesto Ágil (Beck, 2001)
 - ▣ Assinado por 17 desenvolvedores em Utah em fevereiro/2001 (Aliança Ágil).
 - ▣ **Kent Beck**, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, **Ken Schwaber**, **Jeff Sutherland**, Dave Thomas

O Manifesto Ágil

□ **Objetivo**

- Satisfazer o cliente entregando, rapidamente e com frequência, sistemas com algum valor agregado.
 - Entregar versões funcionais em prazos curtos.
 - Estar preparado para requisitos mutantes.
 - Pessoal de negócios e desenvolvedores juntos.
 - Troca de informações através de conversas diretas.

O que é Agilidade?

- **Agilidade é a habilidade tanto para criar quanto para responder à mudança, de forma a obter lucro e agregar valor aos negócios.**
- Aspectos importantes da agilidade:
 - ▣ o foco nas pessoas e suas interações;
 - ▣ a capacitação de indivíduos para tomar decisões rapidamente;
 - ▣ a adaptação de seus próprios processos.
- Engloba a filosofia apresentada no manifesto

Manifesto Ágil

“Estamos descobrindo melhores maneiras/práticas de desenvolver software, fazendo-o e ajudando outros a fazê-lo.”

- ▣ Através deste trabalho passamos a valorizar:

Manifesto Ágil

Valores

- **Indivíduos e interações** são mais importantes que processos e ferramentas
- **Software funcionando** é mais importante que documentação completa e detalhada
- **Colaboração com o cliente** é mais importante que negociação de contratos
- **Adaptação a mudanças** é mais importante que seguir um plano

<http://www.agilemanifesto.org>

Manifesto Ágil

Valores

- **Indivíduos e interações** valem mais que processos e ferramentas
 - ▣ Quem desenvolve o software?
 - ▣ Melhores ferramentas não são segurança de trabalho bem feito
 - ▣ “Um idiota com uma ferramenta ainda é um idiota”

Manifesto Ágil

Valores

- Um **software funcionando** vale mais do que documentação extensa
 - ▣ O que um cliente gostaria de receber? O software ou um documento explicando o software??
 - ▣ Clientes não entendem documentos técnicos!

Manifesto Ágil

Valores

- A **colaboração do cliente** vale mais do que a negociação do contrato
 - ▣ Apenas os clientes podem dizer o que querem (não somos adivinhos!)
 - ▣ Eles nunca acertam da 1ª vez!
 - ▣ Contrato é importante → Direitos e deveres
 - ▣ Contrato não substitui a comunicação
 - ▣ Permite que o cliente tenha maior visibilidade do trabalho, gerando mais confiança
- Dificuldades em ter o cliente disponível, deve buscar alternativas para conciliar a presença do mesmo com os seus afazeres. Situações:
 - Ideal: cliente sempre disponível para a equipe
 - Pior caso: nunca disponível
 - Real: pelo menos o suficiente para atender as necessidades com rapidez

Manifesto Ágil

Valores

- **Responder a mudanças** vale mais do que seguir um plano
 - ▣ As pessoas mudam de prioridade!
 - ▣ O ambiente e as necessidades mudam
 - ▣ Plano de Projeto é essencial, mas deve ser maleável
 - ▣ Ciclo contínuo de feedback

12 Princípios da Aliança Ágil

1. Nossa maior prioridade é satisfazer o cliente desde o início por meio de entrega contínua de software valioso.
2. Modificações de requisitos são bem-vindas, mesmo que tardias no desenvolvimento.
3. Entrega de software funcionando freqüentemente no menor espaço de tempo.
4. O pessoal de negócio e os desenvolvedores devem trabalhar juntos diariamente durante todo o projeto.

12 Princípios da Aliança Ágil

5. Construção de projetos em torno de indivíduos motivados.
6. O método mais efetivo de levar informação para uma equipe de desenvolvimento é por meio da conversa face a face.
7. Software funcionando é a principal medida de progresso.
8. Processos ágeis promovem desenvolvimento sustentável.

12 Princípios da Aliança Ágil

9. Atenção contínua à excelência técnica e ao bom projeto facilitam a agilidade.
10. Simplicidade: a arte de maximizar a qualidade de trabalho efetuado.
11. As melhores arquiteturas, requisitos e projetos surgem de equipes auto organizadas.
12. Em intervalos regulares, a equipe reflete sobre como se tornar mais efetiva, então sintoniza e ajusta adequadamente seu comportamento

Manifesto Ágil

Práticas Ágeis

- São atividades empregadas para executar ou implementar os princípios e valores ágeis.

Práticas Ágeis

Um processo ágil, portanto, deve ser

- **Adaptável.**

- ▣ Mas a adaptação contínua sem progresso realiza pouco.

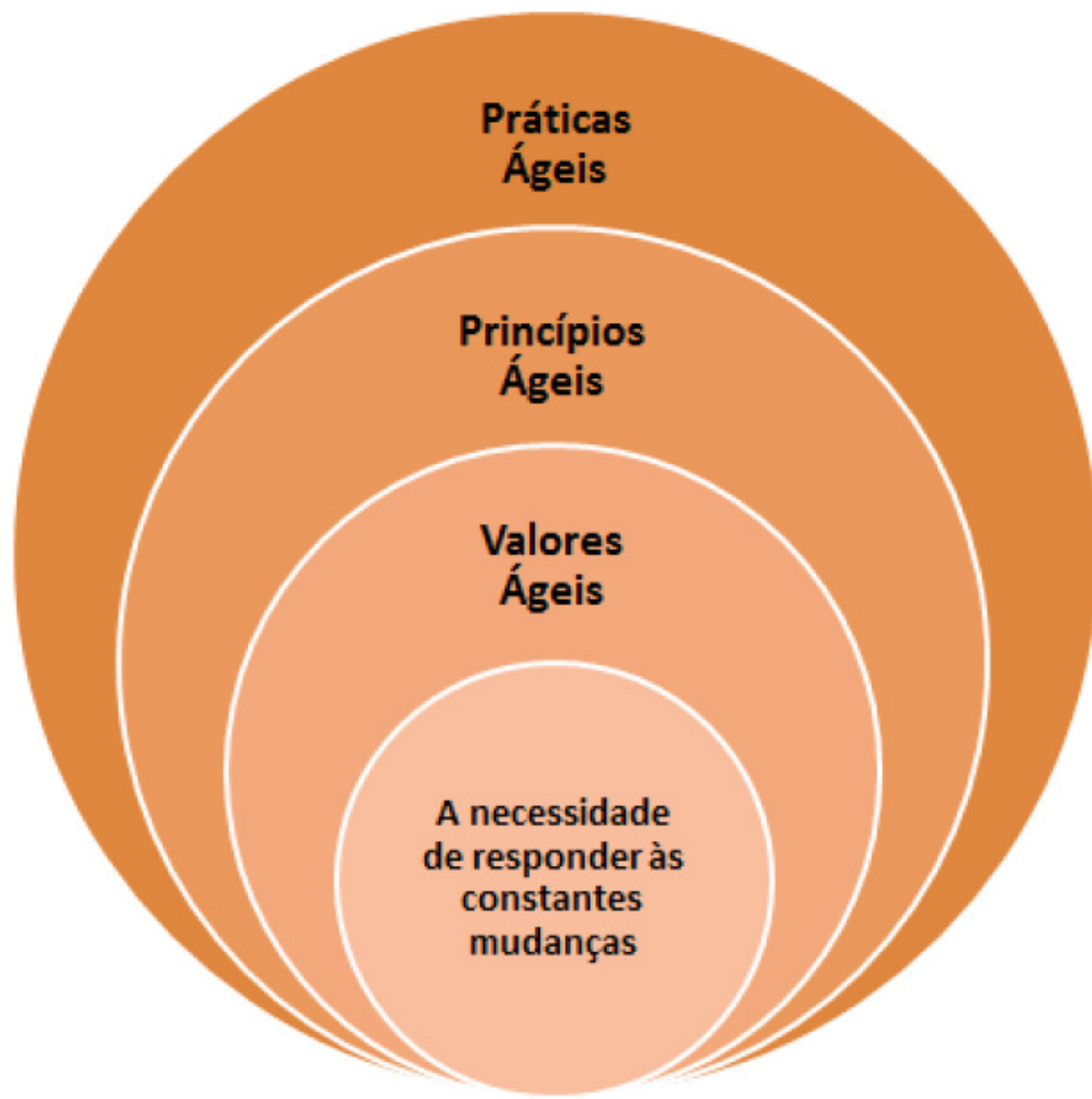
- **Deve ser adaptado incrementalmente.**

- ▣ Abordagem iterativa

Práticas Ágeis

Modelos Ágeis de Processo

- **Scrum**
- Dynamic Systems Development Method – DSDM
- Adaptive Software Development – ASD
- Crystal
- Feature-Driven Development – FDD
- **Extreme Programming – XP**
- Rational Unified Process – RUP (por alguns autores)
- **Agile Web Engineering - AWE**



**Práticas
Ágeis**

**Princípios
Ágeis**

**Valores
Ágeis**

**A necessidade
de responder às
constantes
mudanças**

Por que Ágil?

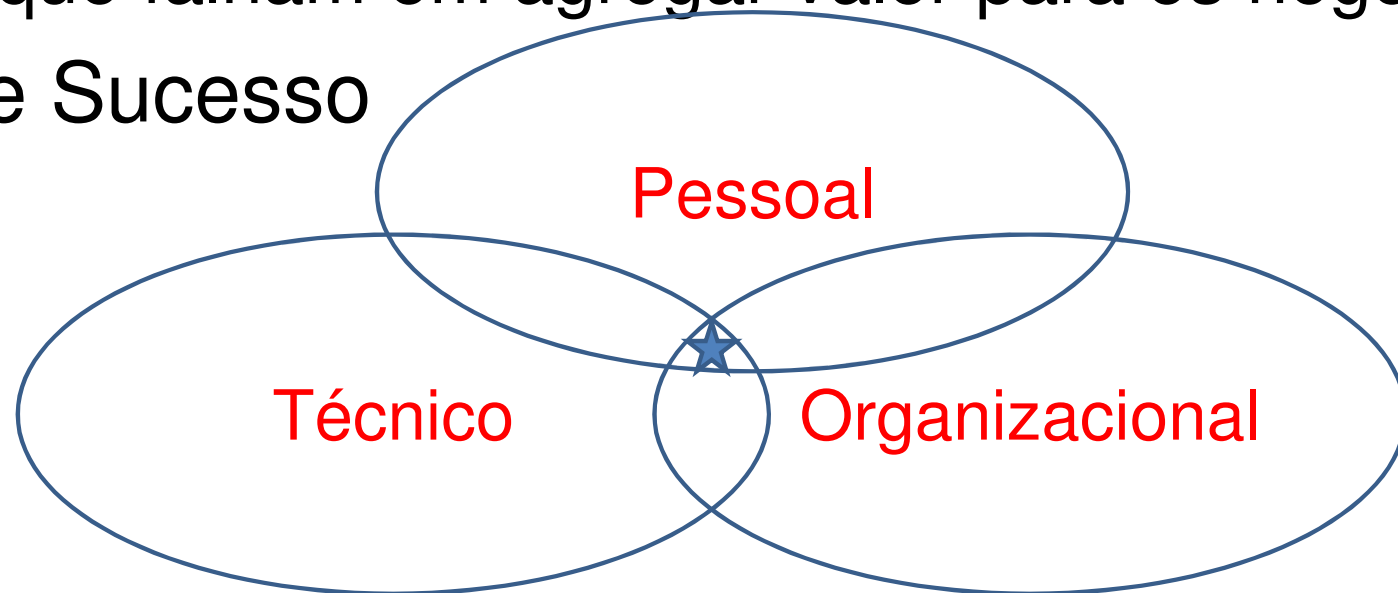
- Ágil é popular
- Google, Yahoo, Symantec, Microsoft...
- Em 1986 [Brooks] previu que não havia soluções que, por volta de 1996, nenhuma tecnologia ou técnica de gerenciamento ofereceria grande aumento de produtividade, confiança ou simplicidade. Foi o que aconteceu.
- **Ágil não é a solução.**

Por que Ágil?

- Equipes demoram de 3 a 6 meses – enquanto aprendem, são mais lentas, tem menos produtividade.
- O desenvolvimento ágil irá nos ajudar a ter mais sucesso?
 - R: Quando responder essa pergunta, você vai saber se ágil é apropriado para você.

Por que Ágil?

- Sucesso = entrega no prazo, dentro do orçamento e de acordo com as especificações.
- O que está errado?
 - A revista CIO coloca que alguns projetos ainda podem ser falhos no final por não atrair usuário finais ou porque falham em agregar valor para os negócios.
- Tipos de Sucesso



Por que Ágil?

- Não importa se seu software é requintado, passível de manutenção ou amado pelos usuários.
 - ▣ Para a gerência, o que importa é **RESULTADO = ROI**
- Quando a gerência está infeliz com os resultados, surgem os cortes.
- Prazos agressivos acabam aumentando os cronogramas ao invés de reduzi-los [McConnell 2006]
- Envio do trabalho para o exterior tem custos ocultos [Overby, 2003]

Por que Ágil?

- O que as empresas valorizam?
 - ▣ Lucros ****
 - ▣ Diferenciação competitiva
 - ▣ Projeção de marca
 - ▣ Lealdade aumentada do cliente
 - ▣ Satisfação das exigências regulatórias
 - ▣ Pesquisa original
 - ▣ Informações estratégicas

Por que Ágil?

- Sucesso Organizacional
 - ▣ Alcançar primeiro os mais valiosos recursos
 - ▣ Novas versões com frequência
 - ▣ Aumento do valor agregado
 - ▣ Equipes experientes tem
 - Quantidade menor de bugs por mês
 - Eliminação do desperdício
 - ▣ Revisão de processos regularmente
 - ▣ Comunicação melhor e intensiva (equipe, cliente)

Por que Ágil?

- Sucesso Técnico
 - ▣ Programadores trabalham juntos (e.g. XP)
 - ▣ Integração contínua de código
 - Lançamento de novas versões constantemente
 - ▣ Equipe concentrada em terminar as funcionalidades completamente, antes de iniciar a próxima.
 - Diminui atrasos
 - Permite mudança de direção à vontade
 - ▣ Desenvolvimento orientado a testes

Por que Ágil?

- Sucesso Pessoal
 - Executivos: apreciação da concentração da equipe pelo ROI sólido e longevidade do software
 - Desenvolvedores: apreciação por ter maior influência sobre estimativas e cronogramas, e autonomia da equipe
 - Testadores: apreciação pela integração com os demais da equipe
 - Usuários: apreciação pela capacidade de influenciar a direção do desenvolvimento e aumento da frequência de entrega.

Como ser Ágil?

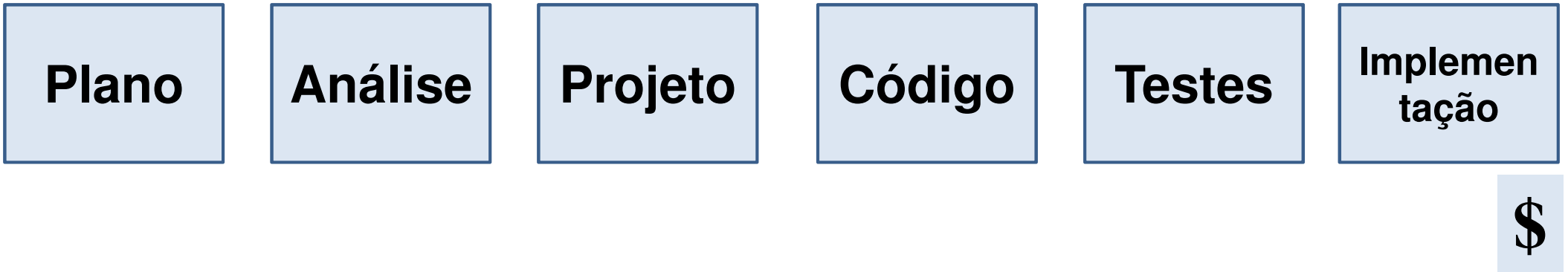
- Nenhuma equipe pratica o método Ágil.
- É uma filosofia. É uma maneira de pensar sobre o desenvolvimento de software.
- Para ser ágil tem que colocar em prática os princípios (12) e os valores (4).

Como ser Ágil?

- Não crie seu próprio método
 - ▣ É comum encontrar organizações que criam ou customizam as práticas existentes a fim de lidar com as necessidades específicas ao seu contexto.
- Caminho da Perfeição
 - ▣ Porque utilizar o desenvolvimento ágil: Vai ter mais sucesso? Como?
 - ▣ XP é certa? SCRUM é certa?
 - ▣ Adote o máximo de práticas que puder, de maneira rigorosa e consistente.
 - ▣ Após adquirir experiência, tente adotar pequenas mudanças. Observe o que acontece e faça melhorias.
- Encontre um mentor (com experiência).

Compreendendo Melhor

Ciclo de vida tradicional (cascata)



Ciclo de vida Iterativo



Cascata x Ágil

Processo	Cascata	Ágil
Medida de sucesso	Aderente ao planejado	Aberto a mudanças
Cultura do gerenciamento	Comando e Controle	Liderança e Colaboração
Requisitos e Projeto	Muita especificação no início do projeto	Especificação contínua e no momento em que ela é necessária
Codificação	Codificar todas as funcionalidades (em paralelo quando possível) e testar depois	Codificar e testar unidades em conjunto
Teste de garantia de qualidade	Grande/Planejado – testes em fases avançadas	Contínuo – testes em fases iniciais
Planejamento	Detalhado com escopo fixo	Com escopo aberto e baseado em ciclos de curta duração

Compreendendo Melhor

- Planejamento, Projeto e testes TODOS os dias.
 - ▣ Quantidade de teste é maior.
- A cada iteração a equipe passa por todas as fases várias vezes.
 - ▣ Não significa mais produtiva.
- Feedback contínuo permite refinamentos rápidos
- Trabalhar em cenários: recursos pequenos mas que trazem valor para o cliente.
 - ▣ Comprometimento semanal de entrega de cenários.
- Final da semana = revisão interna (ou pelo cliente)

Compreendendo Melhor

- Planejamento
 - ▣ Com clientes on-site
 - ▣ Jogo do Planejamento
 - ▣ Mais intenso nas primeiras semanas
 - ▣ Rápida Reunião todos os dias
 - ▣ Gerenciamento de Riscos
- Análise
 - ▣ Cliente deve estar disponível
 - ▣ Coleta de requisitos para o software
 - ▣ Testes unitários junto com programação

Compreendendo Melhor

- Projeto e Codificação
 - ▣ Guiado por TDD (desenvolvimento orientado a testes)
 - ▣ Arquitetura incremental
 - ▣ Programação em Pares
 - ▣ Refatoração
 - ▣ Controle de versão
 - ▣ Integração de código em poucas horas
 - ▣ Padrões de codificação
 - ▣ 40 horas semanais ou Trabalho Energizado
 - ▣ Trabalho em equipe

Compreendendo Melhor

□ Testes

- ▣ Foco na qualidade do software
- ▣ Uso de diferentes técnicas de testes (Unitário, End-to-End, etc)
- ▣ Testes automatizados
- ▣ Cliente ajuda a definir quais testes devem ser feitos (\$\$\$)
- ▣ Ao integrar código, rodam testes de regressão

□ Implementação

- ▣ Final de cada iteração
- ▣ Interna e no Cliente
- ▣ Treinamentos

Compreendendo Melhor

- Pessoas diferentes sabem:
 - ▣ Como projetar e programar o software (programador)
 - ▣ Porque o software é importante (gerente de produto)
 - ▣ As regras que o software deve seguir (especialista de domínio)
 - ▣ Como o software deve se comportar (projetista de interação)
 - ▣ Como a interface do usuário deve ser (projetista gráfico)
 - ▣ Onde os defeitos podem se esconder (testador)
 - ▣ Como interagir com o restante da empresa (gerente de projeto)
 - ▣ Onde melhorar os hábitos de trabalho (coach)

Compreendendo Melhor

- Toda a Equipe
 - ▣ Todos no mesmo espaço de trabalho (aberto)
 - ▣ Início de cada iteração:
 - Demonstração de iteração
 - Retrospectiva
 - Planejamento da iteração
 - ▣ 2 a 4 horas
 - ▣ Rápidas reuniões diárias (5 a 10 minutos), muitas vezes, em pé.
 - ▣ Cada um planeja o seu trabalho (o que não está explícito)

Compreendendo Melhor

- Clientes on-site ou apenas Clientes
 - ▣ Define o software que a equipe constrói.
 - ▣ Toda equipe pode contribuir, mas são clientes que definem.
 - ▣ Planejamento da versão
 - Visão do projeto
 - Cenários e recursos
 - Como agrupar recursos em versões pequenas e frequentes
 - Gerenciar riscos
 - Criar plano tangível (Jogo do Planejamento)
 - ▣ Sugestão inicial: 2 clientes para 3 programadores

Compreendendo Melhor

- Gerente do Produto (Dono do Produto)
 - ▣ Mantém e Promove a Visão do Produto
 - Documentar a visão
 - Compartilhá-la aos envolvidos
 - Incorporar feedback
 - Gerar recursos e cenários
 - Determinar prioridade para Planejamento da versão
 - Fornecer orientações para clientes on-site
 - Liderar demonstrações de iteração
 - Envolver clientes reais
 - Lidar com política organizacional
 - ▣ Propaganda, treinamento.
 - ▣ Em equipes pequenas são também especialistas do domínio.

Compreendendo Melhor

- **Projetista de Interação**
 - ▣ Interface é a face pública do produto.
 - ▣ Para muitos, a IU é o produto.
 - ▣ Compreender os usuários e como irão interagir com o produto.
 - ▣ Divide tempo com equipe e usuários.
 - ▣ Projetistas tem desafio em Ágil pois deverão ter refinamentos constantes.

Compreendendo Melhor

□ Programador

- ▣ 4 a 10 programadores
- ▣ 1 Projetista ou arquiteto
- ▣ Função do programador é minimizar os custos dos clientes.
- ▣ Encontrar maneiras eficazes de entregar cenários.
- ▣ Sugerem estimativas de esforço e alternativas.
- ▣ Ciência da dívida técnica.

□ Testadores

- ▣ O suficiente para que estejam um passo a frente dos programadores.
- ▣ Sugestão: 1 testador para 4 programadores.

Compreendendo Melhor

- 4 a 10 programadores
- 12 pessoas é aconselhável
- 20 membros é avançado = 10 programadores, 3 testadores, 6 clientes, 1 projetista
- Equipes grandes = comunicação extra, gerenciamento complexo, redução da produtividade individual.
- Tenha um coach (interno) e um mentor (externo)

Adotando a Metodologia

- Cultura Organizacional
 - A cultura da sua organização é um fator primário para o sucesso de qualquer iniciativa de mudança, inclusive a adoção de uma nova abordagem para desenvolver software.
 - Embora a cultura incorpore muitas facetas, algumas são importantes em metodologias ágeis:
 - Estrutura Hierárquica ou Cooperativa
 - Controle ou Reação à Mudanças

Adotando a Metodologia

□ Cultura Organizacional

▣ Estrutura Hierárquica

- Existem divisões claras de responsabilidade e autoridade, com papéis específicos responsáveis por atividades distintas.
- Os canais de comunicação tendem a ser claramente definidos e bem restritivos.
- Qualquer comunicação que ocorra fora dos caminhos definidos é considerada como quebra de padrão e pode ser considerado como ato de insubordinação passível de respectiva punição.

▣ Estrutura Cooperativa

Adotando a Metodologia

□ Cultura Organizacional

▣ Estrutura Hierárquica

▣ Estrutura Cooperativa

- A comunicação e o fluxo de trabalho são mais flexíveis.
- Quando algo precisa ser feito, a equipe não se limita a descrições de cargo ou ao departamento que pertence.
- Impulsionada pelas necessidades do negócio, a equipe trabalha num modelo de auto-organização buscando aproveitar da melhor maneira possível as habilidades disponíveis no grupo.

Adotando a Metodologia

- Cultura Organizacional
 - ▣ Controle (Receio) à Mudanças
 - Organização tende a valorizar a continuidade e conformidade aos planos e requisitos.
 - Como mudanças são inevitáveis, e o controle total é a regra de sucesso, a organização tenta gerenciar qualquer mudança para minimizar o efeito nos projetos
 - ▣ Reação à Mudanças

Adotando a Metodologia

- Cultura Organizacional
 - ▣ Controle à Mudanças
 - ▣ Reação à Mudanças
 - Organizações que reagem à mudança tendem a valorizar a satisfação do cliente.
 - Esse tipo de organização não prioriza a abordagem “evitar mudança a qualquer custo”.
 - Ao invés disso, desenvolve mecanismos para continuamente detectar o feedback do cliente e adaptar-se o mais rápido possível a qualquer problema ou opinião divergente que foram comunicados pelo mesmo.

Adotando a Metodologia

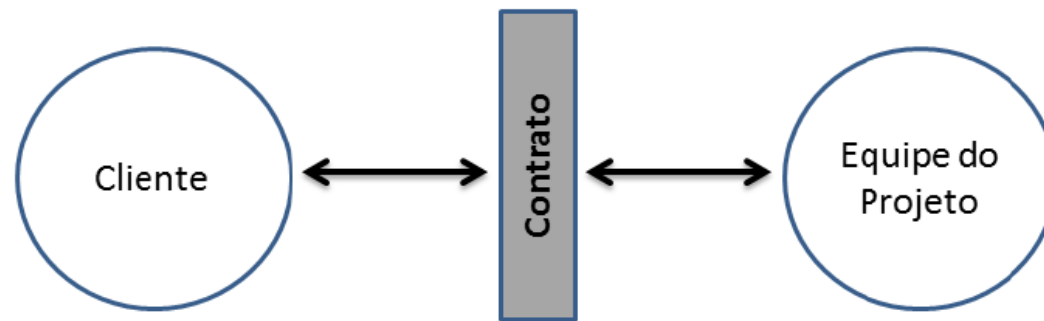
- Considerando os Clientes
 - ▣ O manifesto ágil declara que valoriza mais a colaboração do cliente do que a negociação por contratos.

Adotando a Metodologia

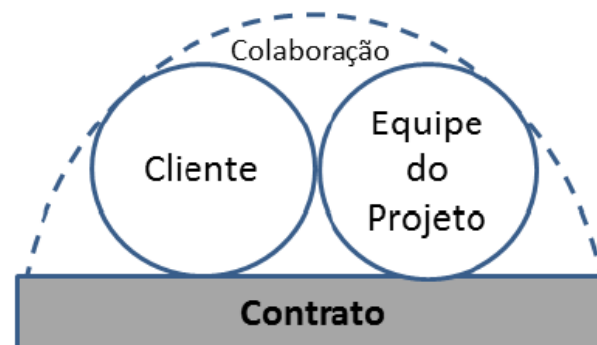
- Considerando os Clientes
- **Contratos:**
 - Contratos são exigidos para preencher a falta de confiança e as incertezas entre organizações diferentes, assim possibilitando que elas trabalhem juntas.
 - O contrato de projeto de prazo fixo tradicional não lida com as incertezas do ambiente de negócio e do ciclo de vida de desenvolvimento de software, colocando cliente e fornecedores uns contra os outros, onde, ao invés de colaboração, tempo e energia são gastos decidindo quem deve pagar por uma mudança.

Adotando a Metodologia

□ Considerando os Clientes



Tradicional: Contrato age como "obstáculo"



Ágil: Contrato age como "apoio"

Adotando a Metodologia

- Considerando os Clientes
- **Contratos:**
 - A adoção de métodos ágeis terá um impacto no relacionamento com os clientes, alterando a natureza dos acordos contratuais, a forma como os requisitos são definidos e gerenciados, e o grau de envolvimento do cliente nos projetos.

Adotando a Metodologia

□ Considerando Equipes

▣ Tamanho das Equipes

- Devem ser pequenas embora não se imponha limitação
- Priorização da comunicação face-to-face ao invés da documental

▣ Localização geográfica

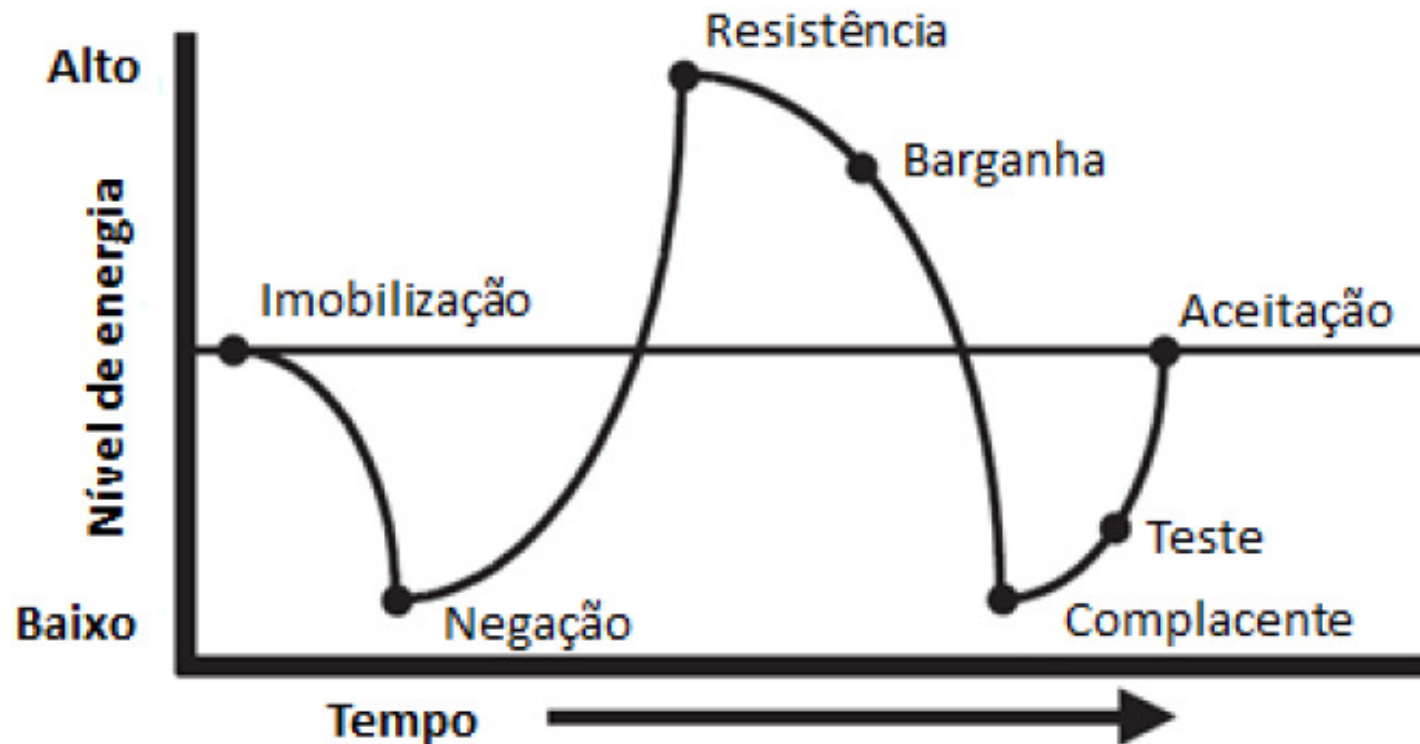
- Podem ser utilizadas equipes dispersas geograficamente
- Surgem desafios em relação a comunicação contínua e livre de ruídos

▣ Criticidade dos projetos

- Projetos críticos tendem a ser desenvolvidos com metodologias tradicionais, porém, avalie risco envolvido para definir qual a melhor estratégia (metodologia).

Adotando a Metodologia

- Considerando Equipes
 - ▣ Estágios em Processo de Mudança



Adotando a Metodologia

□ Considerando Equipes

▣ Fazendo a mudança “pegar”

- Se você decidir por adotar métodos ágeis baseado em “nada” (pouca informação e análise), o esforço certamente será em vão.
- Para a mudança “pegar”, é necessário mergulhar nos detalhes e “segredos” da metodologia até obter conhecimento suficiente sobre os pontos críticos ao seu contexto.
- Se possível, contrate um consultor ou profissional com experiência relevante no assunto e no seu modelo de negócio.

Adotando a Metodologia

□ Considerando Equipes

▣ Fazendo a mudança “pegar”

- Se você espera que as pessoas “comprem a ideia” a respeito da adoção, é importante considerar o planejamento que inclua, na medida do possível, as seguintes reflexões:

- Envolve as pessoas durante o processo de decisão?
- Reforço constantemente as razões para a mudança?
- Discuto sobre os planos de implementação da mudança?
- Comunico regularmente o status da iniciativa de mudança?
- Compartilho as informações sobre os problemas encontrados e como os mesmos foram “atacados”?
- Celebro cada conquista?

Adotando a Metodologia

□ Considerando Equipes

▣ Alterando o sistema de recompensas

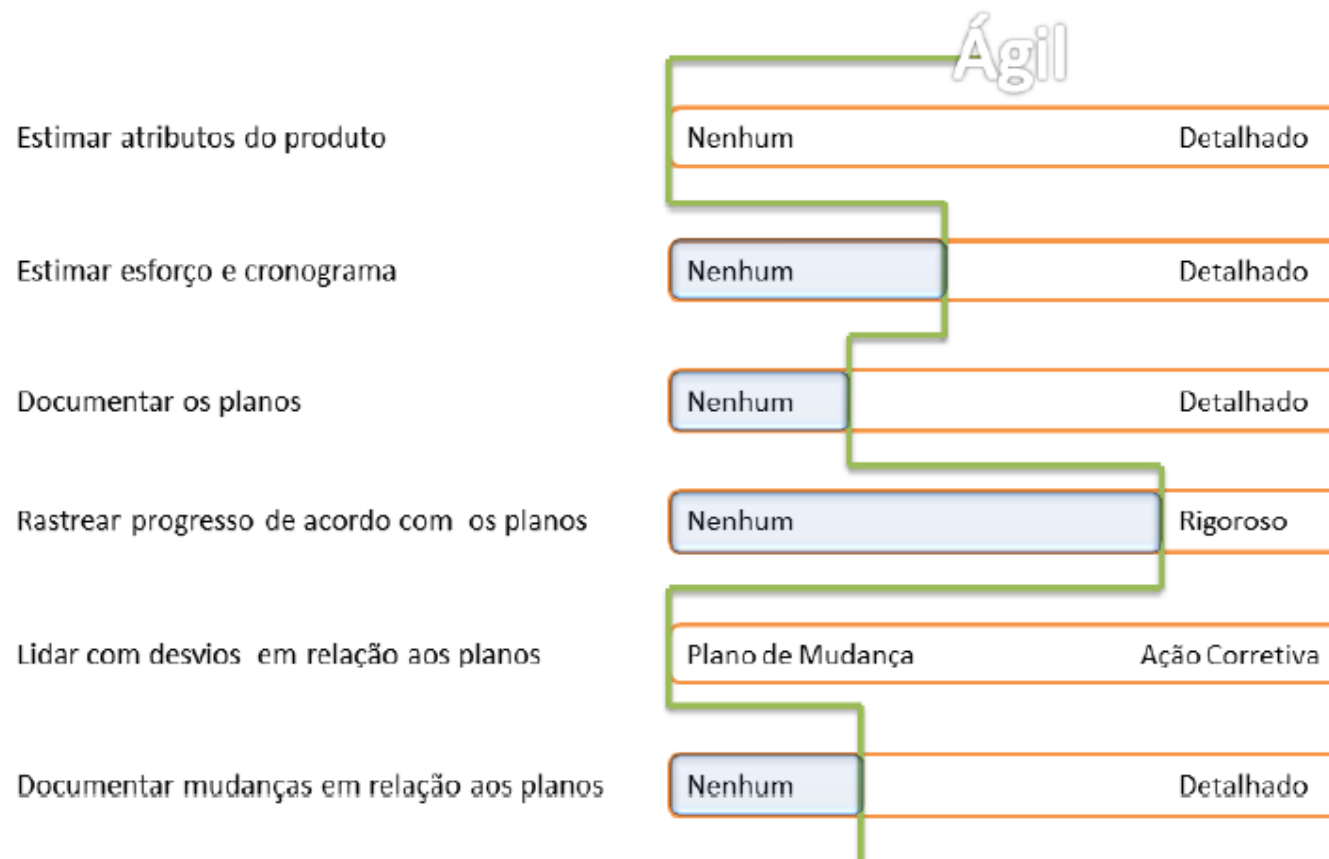
- Toda organização possui um sistema de recompensas. Seja formal ou não, todo funcionário entende muito bem quais comportamentos são recompensados.
- Sem sombra de dúvida, a adoção de um novo método vai acrescentar novas variáveis ao “sistema” e um novo padrão de comportamento vai emergir exigindo cuidados adicionais.

Adotando a Metodologia

- Considerando Ferramentas e Processos
 - ▣ Gerenciamento de Requisitos e Projeto
 - Se a organização já adota uma metodologia ou ferramenta que exige certo nível de documentação e gestão de requisitos, será necessário examinar cuidadosamente as restrições impostas pela abordagem atual para determinar se um método ágil se encaixará nos moldes existentes.
 - Se, por outro lado, a organização não possui métodos definidos à gestão de requisitos, é importante avaliar o impacto das mudanças necessárias para padronizar uma metodologia utilizando uma abordagem ágil (Por exemplo, seremos obrigados a utilizar ferramentas de requisitos? Será necessário um maior rigor ou mais burocracia? Nossa equipe está preparada para as novas competências?);

Adotando a Metodologia

- Considerando Ferramentas e Processos
 - ▣ Gerenciamento de Requisitos e Projeto
 - Gestão Ágil de Projetos



Adotando a Metodologia

- Ter como condição-alvo, a **melhoria contínua** dos processos de negócio que constituem a empresa.
- **Caminhar através dos obstáculos** que surgirão enquanto se avança em direção à condição-alvo.
- **A realidade, infelizmente, não é linear e nem previsível** para alcançar nossas condições-alvo;
- **Não adianta “gastar” muito tempo com planos muito detalhados** porque sempre que um passo é dado, o cenário de adoção, como consequência, pode ter mudado consideravelmente;
- O caminho em direção à condição-alvo é trilhado em **passos curtos e rápidos**, com **a aprendizagem** e os **ajustes** ocorrendo ao longo do caminho.

Adotando a Metodologia

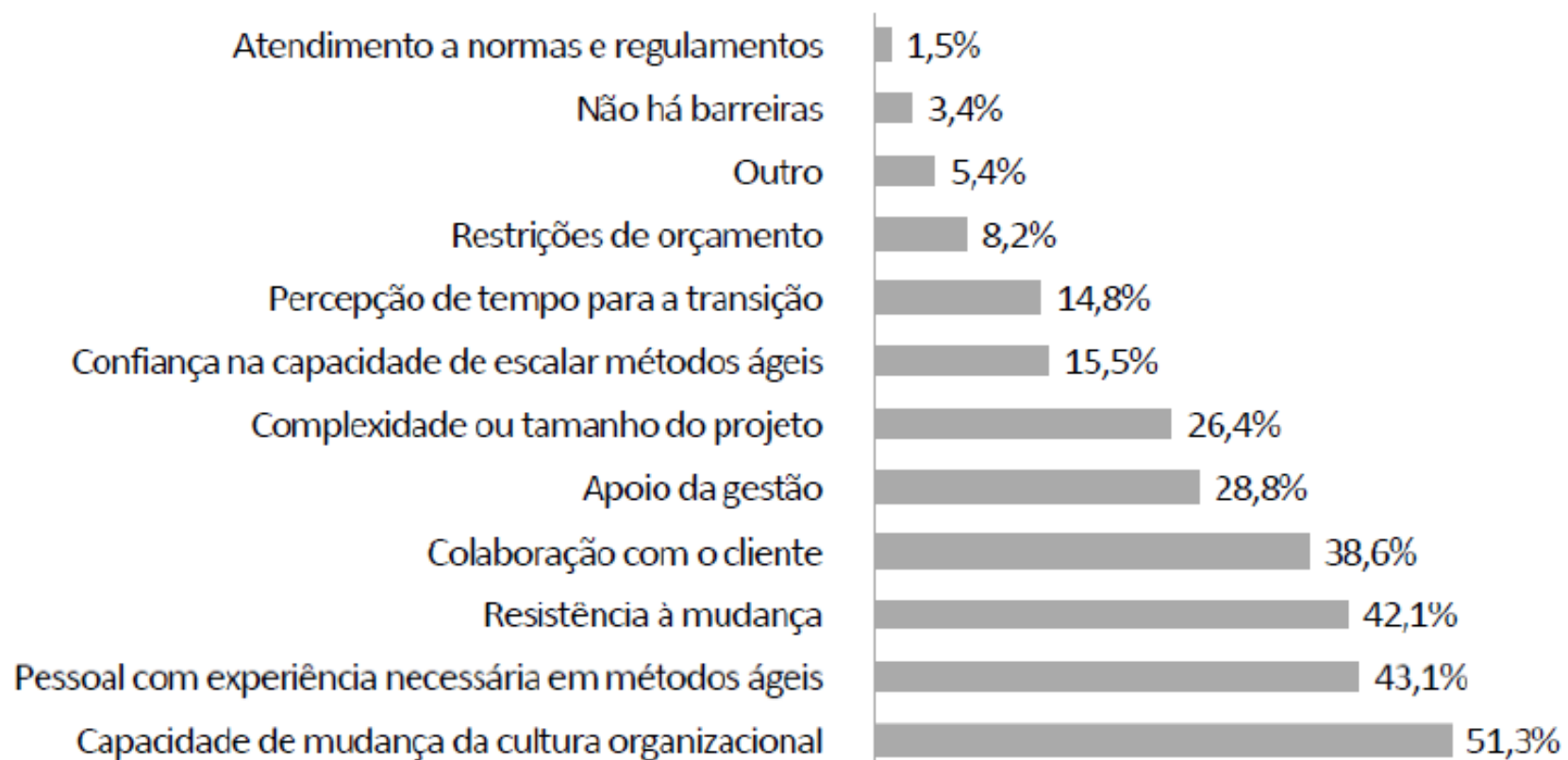
- State of Agile Development Survey 2011 (Global)
 - ▣ Obstáculos para o Processo de Adoção



Na Pesquisa de 2008 os números eram para os 3 primeiros: 45%, 42%, 44%.

Adotando a Metodologia

- Pesquisa pela USP em comemoração aos 10 Anos do Manifesto Ágil (Brasil)
 - ▣ Obstáculos para o Processo de Adoção



Adotando a Metodologia

- ThoughtWorks – referência mundial na utilização de práticas ágeis em seus projetos – e pesquisadores da UFRGS
 - ▣ Recomendações para adoção
 - Processo de decisão coletivo e participativo (ao invés de um baseado em autoritarismo e paternalismo);
 - Valorização do pragmatismo (em oposição a uma autoridade moralista);
 - Crença de que os seres humanos são basicamente bons (em oposição a basicamente maus);
 - Uma atitude proativa (em oposição a reativa e fatalista);
 - “Grupocentrismo” (em oposição ao individualismo).

Adotando a Metodologia

- ThoughtWorks – referência mundial na utilização de práticas ágeis em seus projetos – e pesquisadores da UFRGS
 - Quais razões para adotar metodologias ágeis
 - 77% buscavam acelerar o “time-to-market” (tempo de resposta a uma demanda do mercado);
 - 83% buscavam melhorar a capacidade de gerenciar a mudança de prioridades;
 - 80% buscavam melhorar a produtividade;
 - 65% buscavam melhorar o alinhamento entre TI e Negócio;
 - 71% buscavam aumentar a qualidade do software.

Adotando a Metodologia

- Recomendações para a implantação:
 - Apoio da Gerência - obtenha o suporte do “alto escalão executivo” e da liderança;
 - Use os princípios ágeis para tratar de questões relevantes ao negócio;
 - Defina o que significa “ser ágil” na sua organização;
 - O engajamento de toda a organização deve fazer parte da missão;
 - Invista na capacitação da equipe de transição;
 - Apresente resultados em termos de negócio (evite jargões desconectados do contexto);

Adotando a Metodologia

- Recomendações para a implantação:
 - ▣ Não argumente só com princípios. Use métricas objetivas;
 - ▣ Reconheça o esforço para mudar paradigmas, modelos mentais e comportamentos;
 - ▣ Prepare-se para superar os obstáculos durante a transição;
 - ▣ Tenha um plano. Tenha bom senso e seja ágil no planejamento.
 - ▣ Cliente On-Site é recomendável
 - ▣ Tamanho correto da Equipe
 - ▣ Utilize Todas as Práticas

Adotando a Metodologia

- Recomendações
 - ▣ Ótimos conhecimentos em Projeto
 - ▣ Uma Linguagem que seja fácil de Refatorar
 - ▣ Um Coach-programador experiente
 - ▣ Uma Equipe amigável e coesa
 - ▣ Extermine a dívida técnica
 - ▣ Organize seu backlog
 - ▣ Conserte bugs importantes

Adotando a Metodologia

- Encontre projeto adequado para começar a trabalhar
 - Evite pegar projeto de “baixo valor”
 - Aplique num Projeto NOVO (é mais fácil)
- Monte sua equipe
- Considere a contratação de um coach.

Adotando a Metodologia

- Compras
 - ▣ Equipamento para desenvolvimento em pares
 - ▣ Equipamento dedicado para Integração Contínua
 - ▣ Divisória anti-ruídos
 - ▣ Quadros brancos
 - ▣ Grande calendário
 - ▣ Software de testes unitários
 - ▣ Ferramenta de desenvolvimento automatizado (desenvolvimento em 10 minutos)
 - ▣ Fichas e Lápis

Considerações Finais

- Busque repostas para as questões:
 - Estamos prontos para adotar alguma metodologia ágil?
 - Dentre os princípios, valores e práticas conhecidos, existe um conjunto mais coerente com o nosso contexto?
 - Será necessário customizar a metodologia para atender as restrições e necessidades da nossa organização?
 - Devemos definir nossa própria metodologia a partir da escolha de algumas práticas que já testamos?

Considerações Finais

- Busque repostas para as questões:
 - Na atual conjuntura, faz sentido utilizar qualquer metodologia ágil?
 - Você está preparado para assumir os custos de um processo de adoção que é executado sem planejamento prévio a respeito dos passos necessários para uma análise inicial com o objetivo de responder sobre a viabilidade do projeto?

Considerações Finais

- Busque repostas para as questões:
 - Se os critérios de viabilidade não lhe parecem evidentes (estrutura da organização, ciclo de vida dos projetos, gestão do capital humano, modelo de negócio, relacionamento com clientes, etc.), qual será sua estratégia para escapar da tentação de definir como condição-alvo a “mera” adoção de uma metodologia enquanto fecha os olhos para o que realmente importa e precisa ser mudado?
 - Até que ponto o fato de conhecermos o que está por trás das falhas de diversas iniciativas de adoção pode representar algum benefício real? Defendemos a tese segundo a qual “não aprendemos com os erros, de fato, o aprendizado é confirmado pela correção dos erros”. Qual sua opinião a respeito disso? O que você e sua organização podem ganhar a partir dos erros (próprios e dos outros)?

Referências

- James Shore e Shane Warden. The Art of Agile development. Alta Books. 2008
- Brooks, Fredrick P. 1995. The Mythical Man-Month: Essays on Software engineering, 20th Anniversary edition. Boston: Addison-Wesley Professional.
- McConnell, Steve. 2006. Software Estimation: Demystifying the Black Art. Redmond, WA: Microsoft Press.
- Overby, Stephanie. 2003. The Hidden Costs of Offshore Outsourcing the Money. CIO Magazine.
- Beck, Kent et al. 2001. Manifesto for Agile Software Development. <http://agilemanifesto.org>.

Referências

- “Iterative and Incremental Development: A Brief History”, Craig Larman (Valtech), Victor R. Basili (University of Maryland), IEEE Computer Society, 2003.
- “CMMI® or Agile: Why Not Embrace Both!”, Hillel Glazer (Entinex, Inc.), Jeff Dalton (Broadsword Solutions Corporation), David Anderson (David J. Anderson & Associates, Inc.), Mike Konrad (SEI), Sandy Shrum (SEI), Technical Note, CMU/SEI-2008-TN-003, 2008.
- “Managing the Dynamics of Mutual Adaptation of Technology and Organisation in Information Systems Development Projects”, Bendik Bygstad, Software Process Improvement and Practice, Wiley InterScience, 2005.
- “Agile Software Development: Evaluating the methods for your organization”, Alan S. Koch, Artech House computing library, 2005.
- Doing Agile Isn't the Same as Being Agile
<http://martinfowler.com/articles/newMethodology.html>
- Keys to a Successful Agile Transformation
<http://www.pillartechnology.com/filestore/webinars/pdf/7%20Keys%20to%20Successful%20Agile%20Transition.pdf>