

Gerenciamento ágil

Leonardo Gresta Paulino Murta
leomurta@ic.uff.br

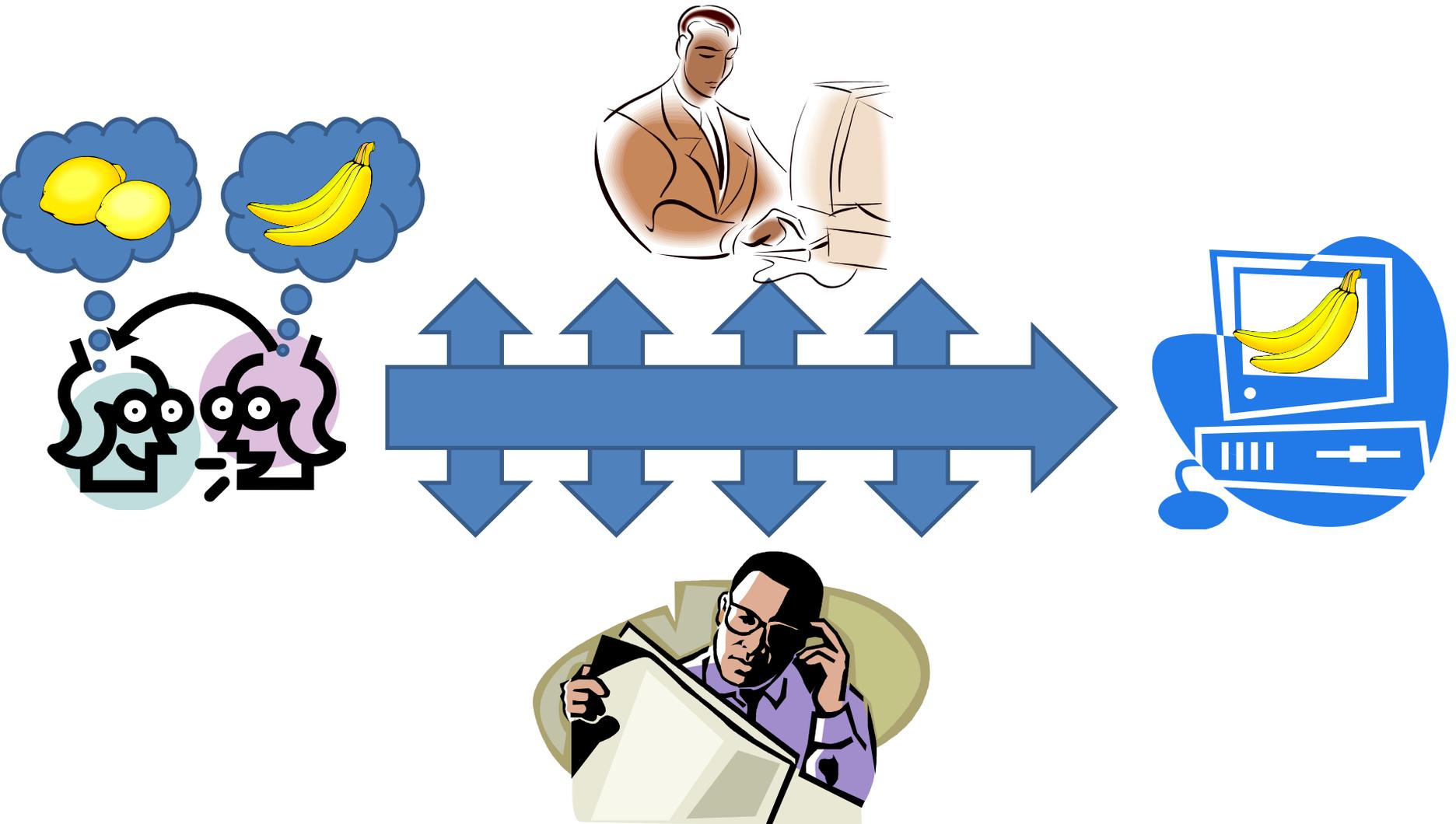
Como lidar com mudanças?

- Mudanças sempre acontecem em projetos de software
 - Requisitos mudam
 - O ambiente em que o software está inserido muda
 - As pessoas que operam o software mudam
- Estratégias para lidar com mudanças
 - Evitar as mudanças (corretivas) fazendo uso de boas técnicas de engenharia de software
 - Acolher mudanças (evolutivas) por meio de um processo iterativo

Desenvolvimento Cascata

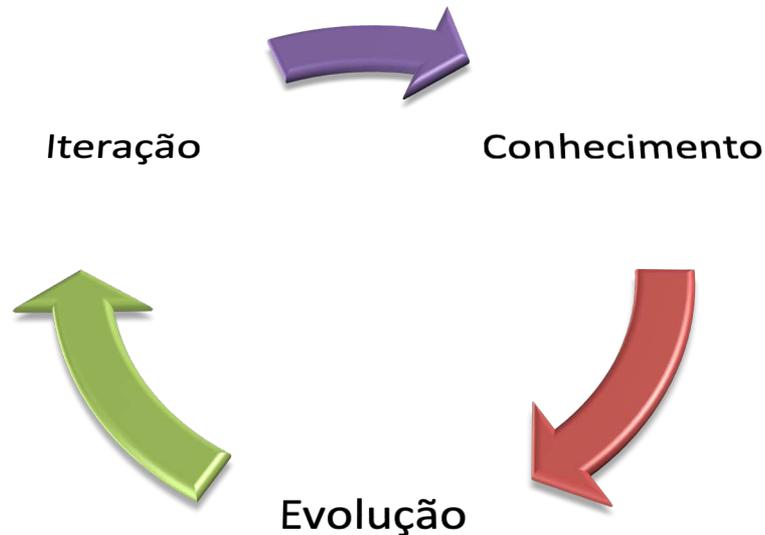


Desenvolvimento iterativo



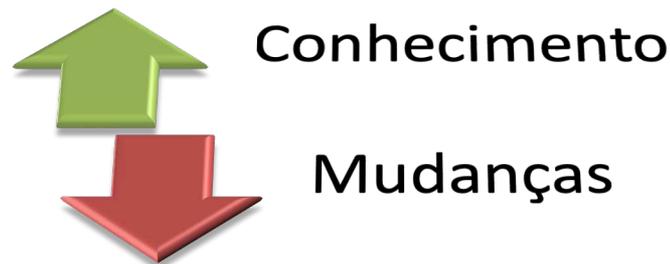
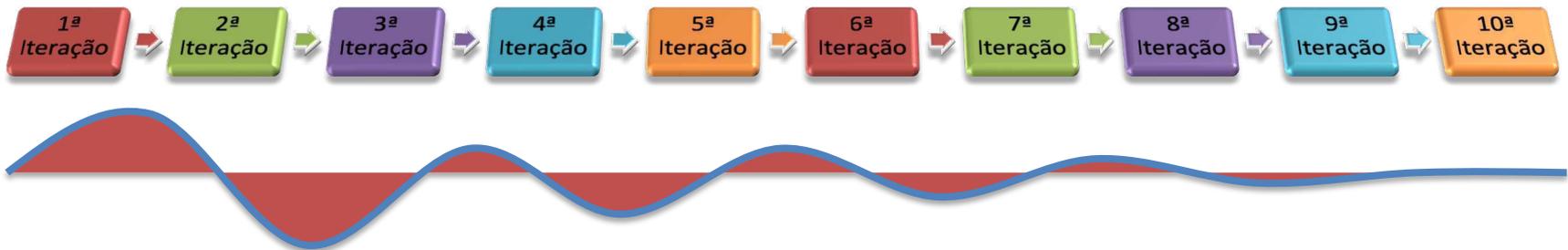
Desenvolvimento iterativo

- As especificações evoluem a cada iteração
 - A cada iteração, uma parte do software fica pronta
 - O conhecimento sobre o software aumenta pelo feedback do usuário
 - As especificações são evoluídas para retratar esse aumento de conhecimento sobre o que é o software



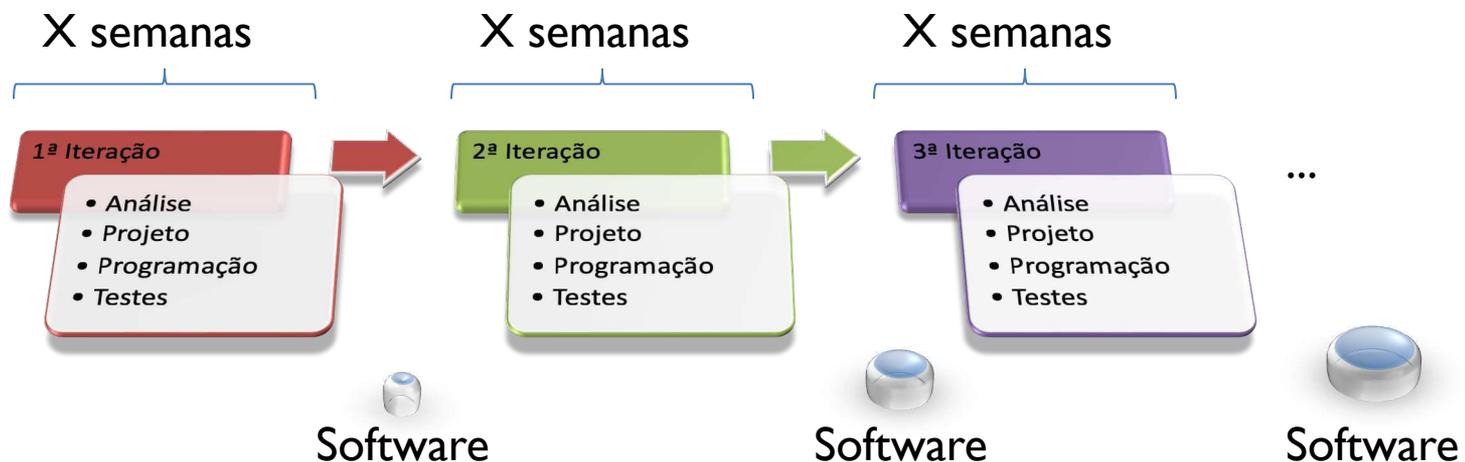
Mudanças x iterações

- Iterações curtas privilegiam a propagação de conhecimento
 - Aumento do conhecimento sobre o software
 - Diminuição das incertezas, que levam às mudanças



Iteração

- O desenvolvimento é organizado em “mini-projetos”
 - Cada “mini-projeto” é uma iteração
 - Cada iteração tem duração curta e fixa (de 2 a 6 semanas)
 - Cada iteração tem atividades de análise, projeto, programação e testes
 - O produto de uma iteração é um software parcial



Iteração

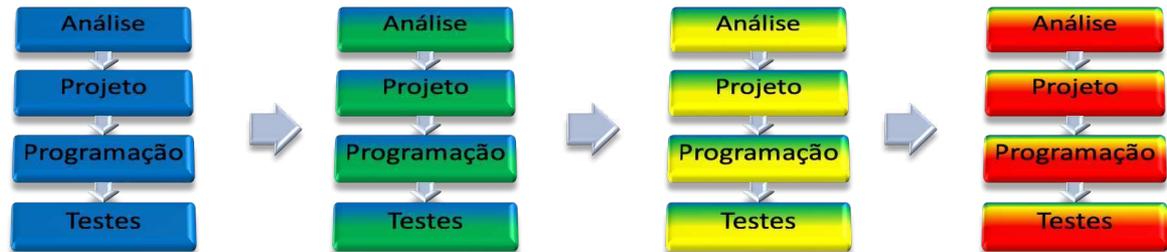
- A iteração deve ser fixa
 - Tarefas podem ser removidas ou incluídas
 - A iteração nunca deve passar da duração previamente estipulada
- O resultado de cada iteração é um software...
 - Incompleto
 - Em desenvolvimento (não pode ser colocado em produção)
 - Mas não é um protótipo!!!
- Esse software pode ser verificado e validado parcialmente
 - Testes
 - Usuários
- Podem ser necessárias diversas iterações (e.g. 10 a 15) para ter uma versão do sistema pronta para entrar em produção

Com x Sem iteração

Sem interação



Com interação



Desenvolvimento Ágil

- São dadas respostas rápidas e flexíveis a mudanças
 - O projeto é replanejado continuamente
 - São feitas entregas incrementais e constantes do software, refletindo as mudanças solicitadas



Desenvolvimento Ágil

12 Princípios Ágeis

Satisfazer o cliente com software que agregue valor

Acolher modificações nos requisitos

Entregar o software funcional com frequência

Trabalhar junto ao cliente

Manter as pessoas motivadas e confiar nelas

Promover conversas face a face

Medir o progresso com software funcionando

Manter um ritmo constante de trabalho

Prezar por excelência técnica

Buscar por simplicidade

Trabalhar com equipes auto-organizadas

Ajustar o comportamento da equipe buscando mais efetividade

Foco no método Scrum...



Fonte:
<http://www.whatisrugby.com>

Papéis no Scrum

- **Scrum Master** é responsável por facilitar e assegurar que o processo é entendido e seguido
- **Product Owner** é responsável por representar os interesses do cliente
- **Team** representa os desenvolvedores (5 a 9 membros), atuando em diferentes aspectos do desenvolvimento (análise, projeto, código, testes, etc.) de forma multidisciplinar

Galinhas x Porcos

Ei Porco, acho que deveríamos abrir um restaurante.

Não sei não. Qual seria o nome?



Que tal "Pernil com Ovos".

Não obrigado, eu estaria comprometido, enquanto você estaria apenas envolvido!

By Clark & Vizdos

© 2006 implementingscrum.com

Porcos: *Product Owner, Scrum Master e Team*
Galinhas: *Demais stakeholders*

Iterações

- ***Sprint*** é uma iteração de desenvolvimento que dura de 1 a 4 semanas (tempo fixo)

Artefatos

- ***História do usuário*** é uma especificação de uma ou mais sentenças em narrativa livre que diz o que o usuário necessita

Pagamento - Boleto

Como um Cliente, quero utilizar a forma de pagamento Boleto Bancário Para pagar meus pedidos.

Pagamento - Débito em Conta - Banco 1

Como um Cliente, quero utilizar a forma de pagamento Débito em Conta do Banco 1 Para pagar meus pedidos.

Pagamento - Débito em Conta - Banco 2

Como um Cliente, quero utilizar a forma de pagamento Débito em Conta do Banco 2 Para pagar meus pedidos.

Pagamento - Cartão de Crédito - Bandeira A

Como um Cliente, quero utilizar a forma de pagamento Cartão de Crédito da Bandeira A Para pagar meus pedidos.

Pagamento - Cartão de Crédito - Bandeira B

Como um Cliente, quero utilizar a forma de pagamento Cartão de Crédito da Bandeira B Para pagar meus pedidos.

Pagamento - Cartão de Crédito - Bandeira C

Como um Cliente, quero utilizar a forma de pagamento Cartão de Crédito da Bandeira C Para pagar meus pedidos.

Fonte: <http://www.metodoti.com>

Artefatos

- ***Product backlog*** consiste na lista ordenada de pacotes de trabalho que ainda não foram incorporados ao produto
- ***Sprint backlog*** consiste nos pacotes de trabalho selecionados do topo do *Product Backlog* para serem desenvolvidos no *Sprint*

Cerimônias

- ***Sprint Planning***

- Reunião de no máximo 8 horas (para *Sprints* de 4 semanas) no início do *Sprint*
- Visa definir o *Sprint Backlog*

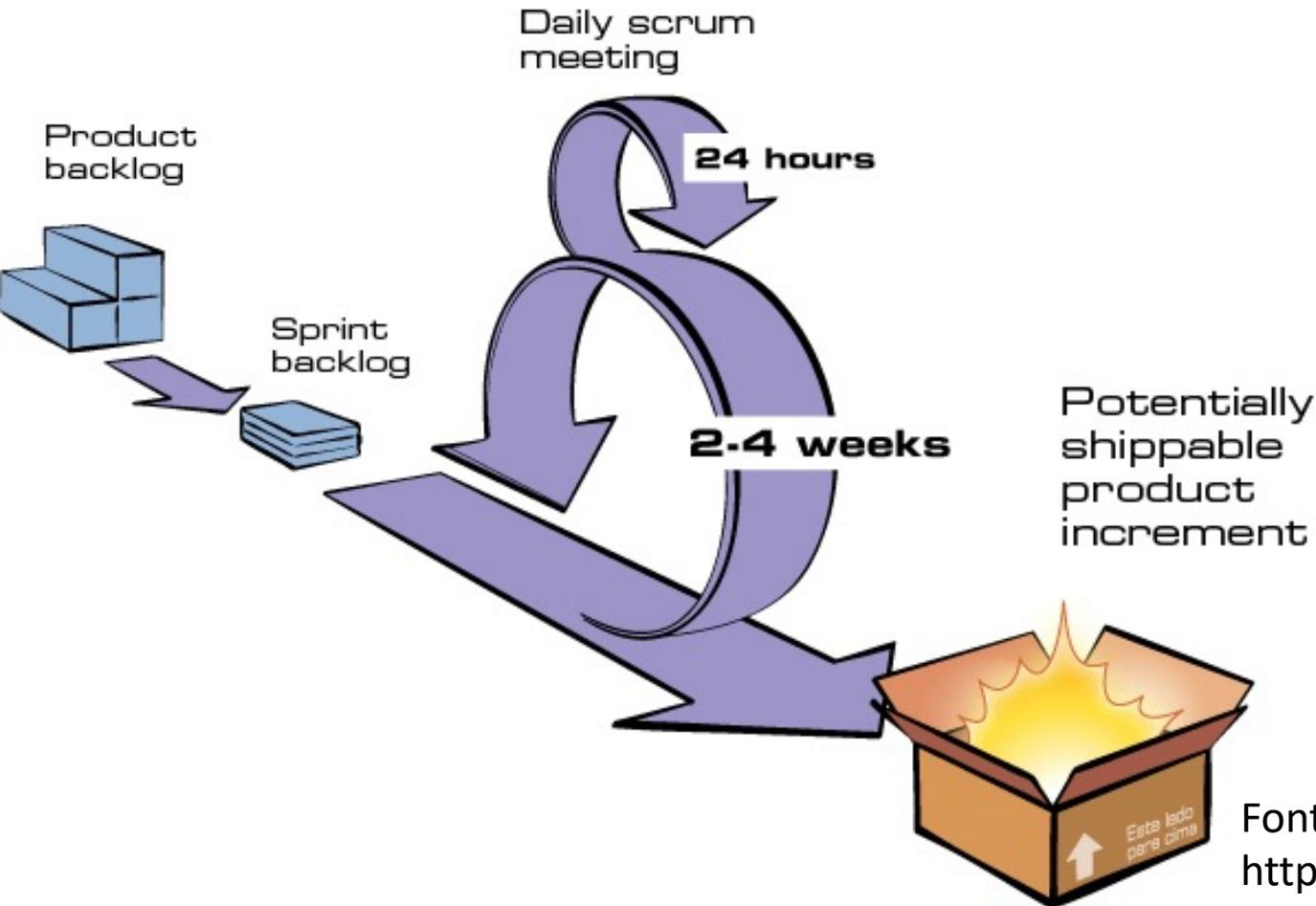
- ***Daily Meeting***

- Reunião de 15 minutos no início do dia
- Visa dar visibilidade do progresso
- Cada desenvolvedor responde: O que foi feito no dia anterior? O que será feito no dia? Há algo o impedindo de progredir?

Cerimônias

- ***Sprint Review***
 - Reunião de no máximo 4 horas ao final do *Sprint*
 - Visa demonstrar para o cliente o que foi feito no *Sprint*
- ***Sprint Retrospective***
 - Reunião de no máximo 3 horas ao final do *Sprint*
 - Visa apoiar uma reflexão para melhoria do processo
 - Cada desenvolvedor responde: O que deu certo? O que pode melhorar?

Visão geral de Scrum



Fonte:
<http://improveit.com.br/scrum>

Estimativas

- Estimativas em métodos ágeis: Planning Poker!



Fonte: <https://devsquad.com>

Vantagens

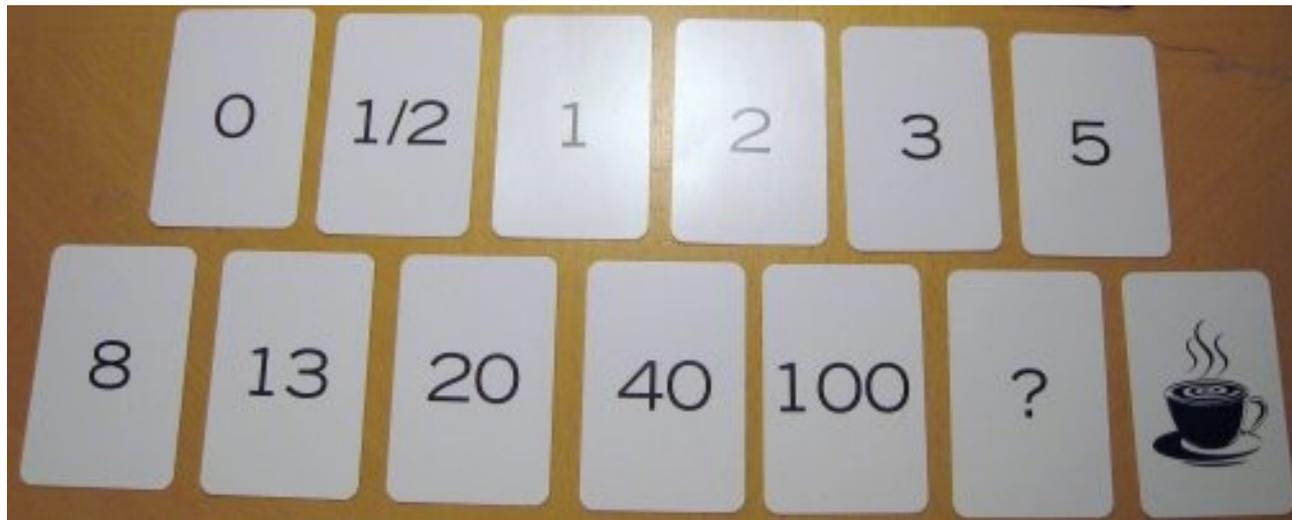
- Técnica que visa o **comprometimento dos membros da equipe**
 - Todos participam do processo de estimativa
 - Todos são responsáveis pela sua concretização
- Permite **rapidamente** chegar a uma estimativa
- Normalmente cativa os envolvidos por ter uma dimensão **lúdica**
- É baseada em **consenso!**

Artefatos necessários no *Planning Poker*

- Elementos a serem estimados:
 - Histórias
- Mas é possível estimar outros artefatos:
 - Casos de Uso
 - Pacotes de trabalho
 - Atividades
 - Etc.

Artefatos necessários no *Planning Poker*

- Um deque, usualmente de 13 cartas, para cada membro da equipe
 - As cartas representam esforço (pontos, homens-dia, homem-hora, etc.)
 - Ex.: 3 = 3 pessoas em 1 dia ou 1 pessoa em 3 dias



Processo do *Planning Poker*

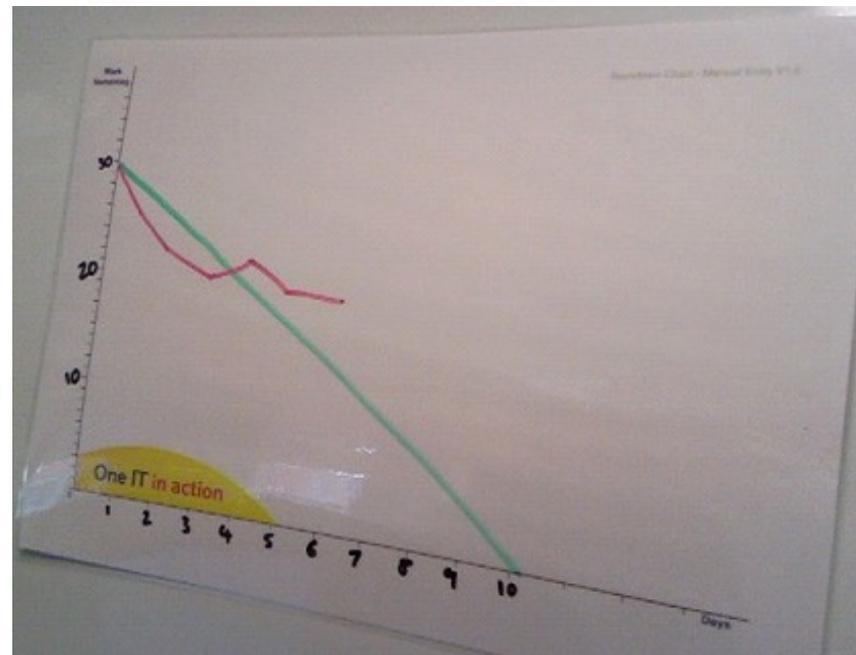
1. Coloque o elemento a ser estimado no centro da mesa
2. Cada membro coloca a sua carta de estimativa na mesa, virada para baixo
 - A estimativa não é só codificação, mas inclui também modelagem, testes, integração, etc.
 - Nenhum membro deve argumentar a razão da sua escolha
3. As cartas são virada para cima ao mesmo tempo
 - Raramente cartas iguais aparecem. Isso é normal!!!
4. Calcula-se a média das estimativas

Processo do *Planning Poker*

5. As estimativas são analisadas
 - Os membros com estimativas distantes da média **explicam seus raciocínios** (eles podem ser os certos!!!)
 - Se a média está muito alta, pode ser necessário **decompor o elemento** sendo estimado e estimar as partes
 - Se as estimativas estiverem baseadas em hipóteses não fundamentadas, essas **hipóteses devem ser discutidas com o usuário**
6. O processo se repete até que o consenso seja obtido

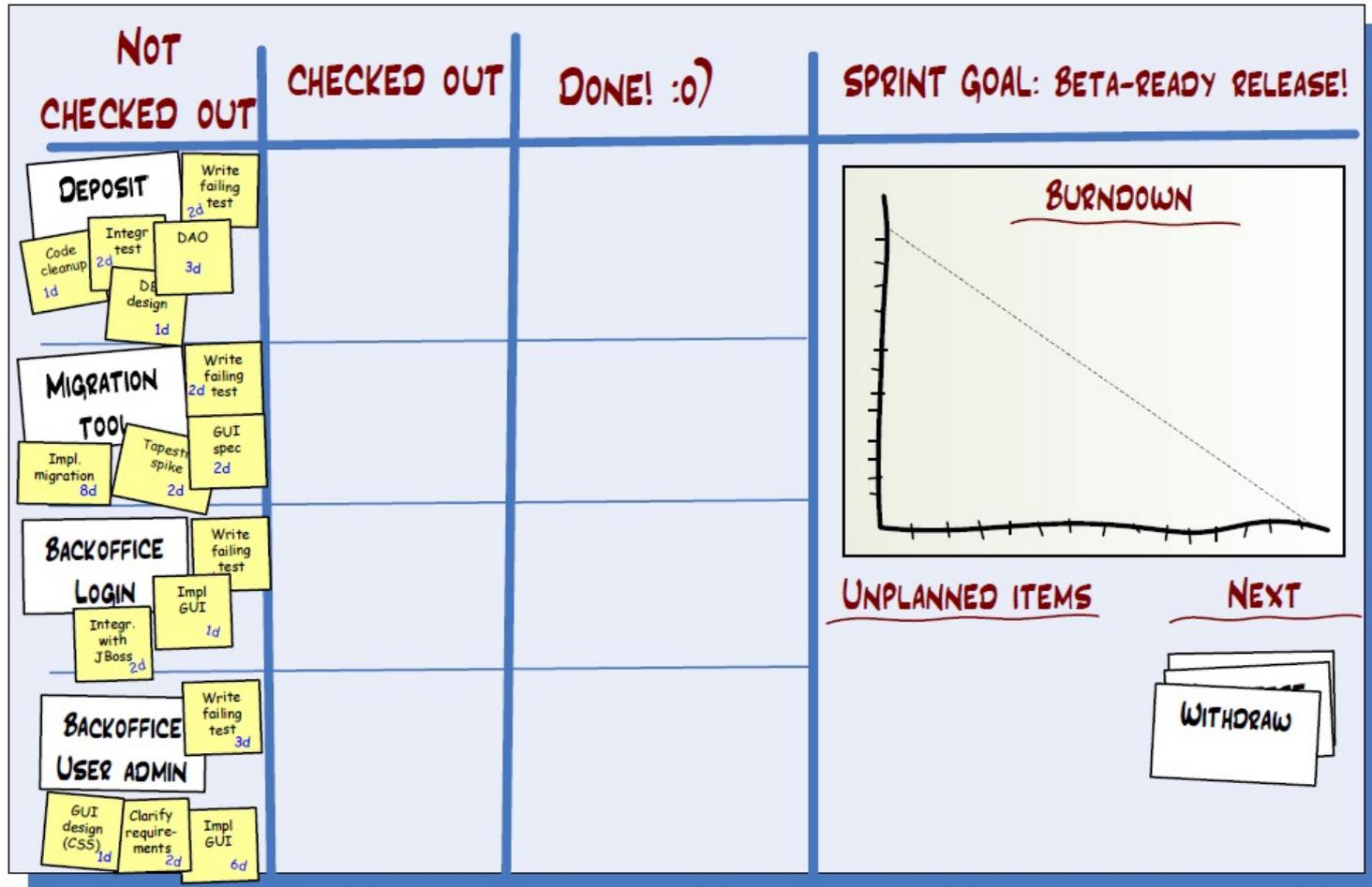
Monitoramento e controle

- Monitoramento e controle em métodos ágeis: quadro de *burndown*!

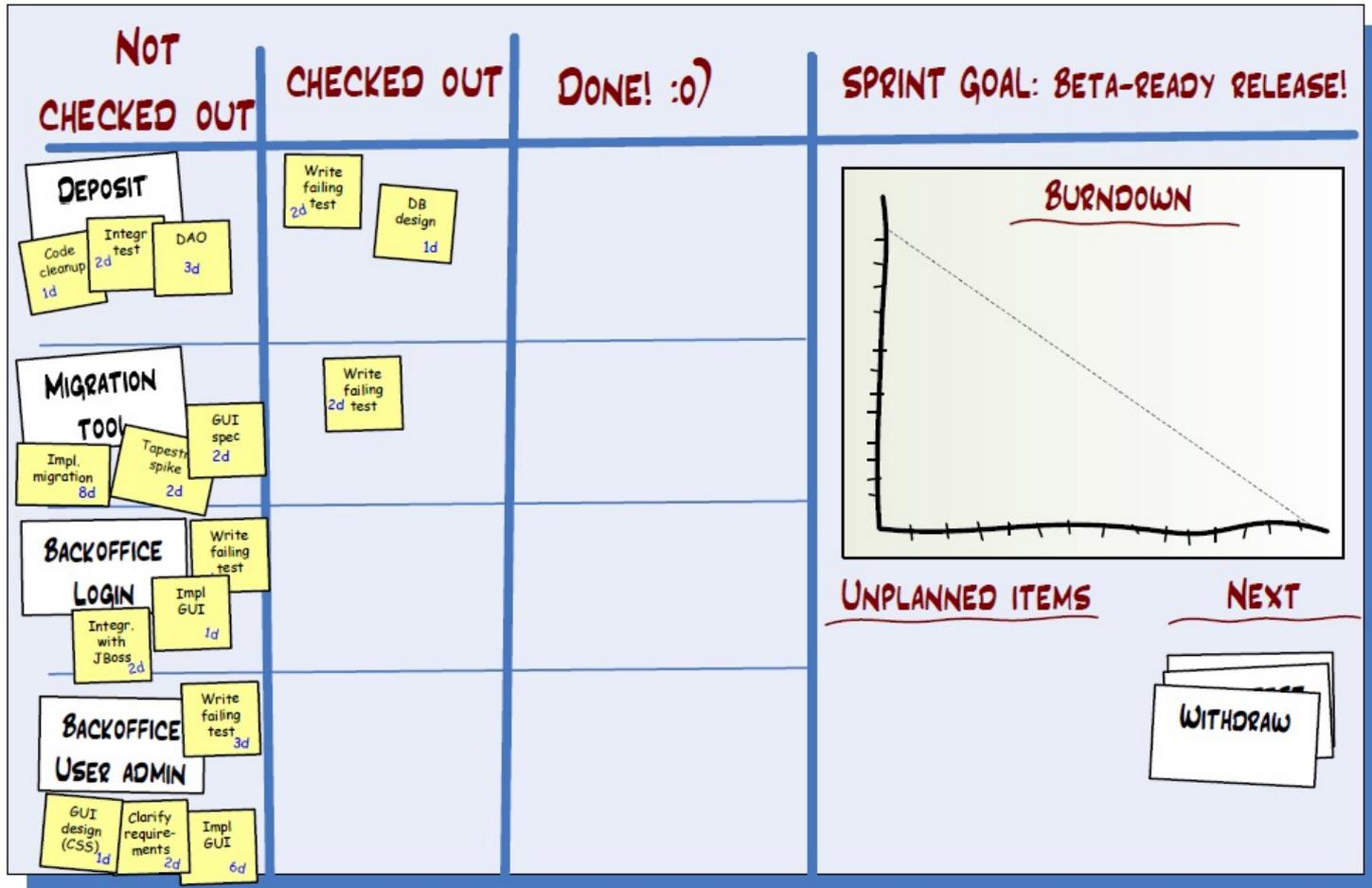


Fonte: <http://www.scrumalliance.org>

Quadro de *burndown*



Quadro de *burndown*



Quadro de *burndown*

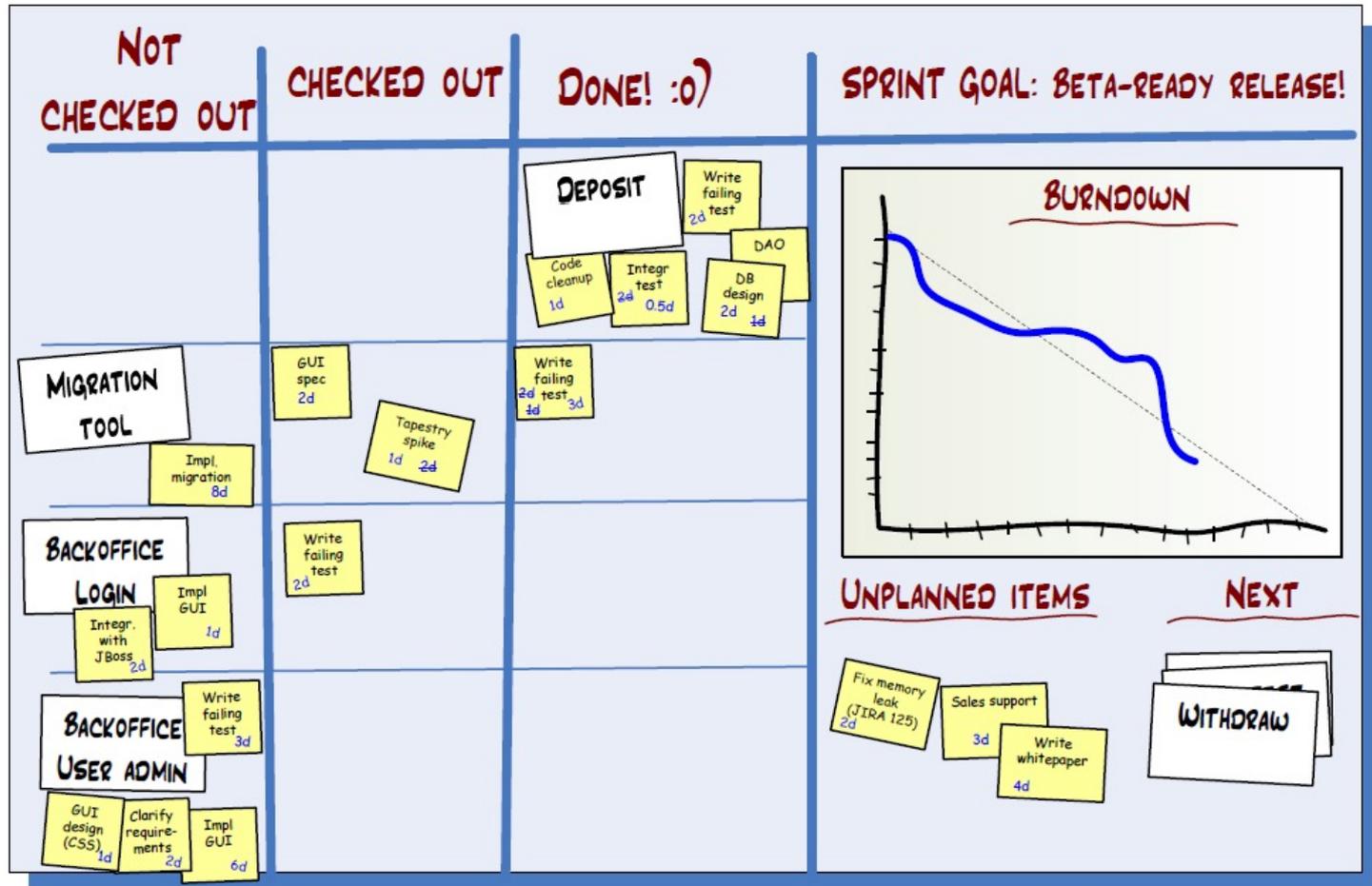
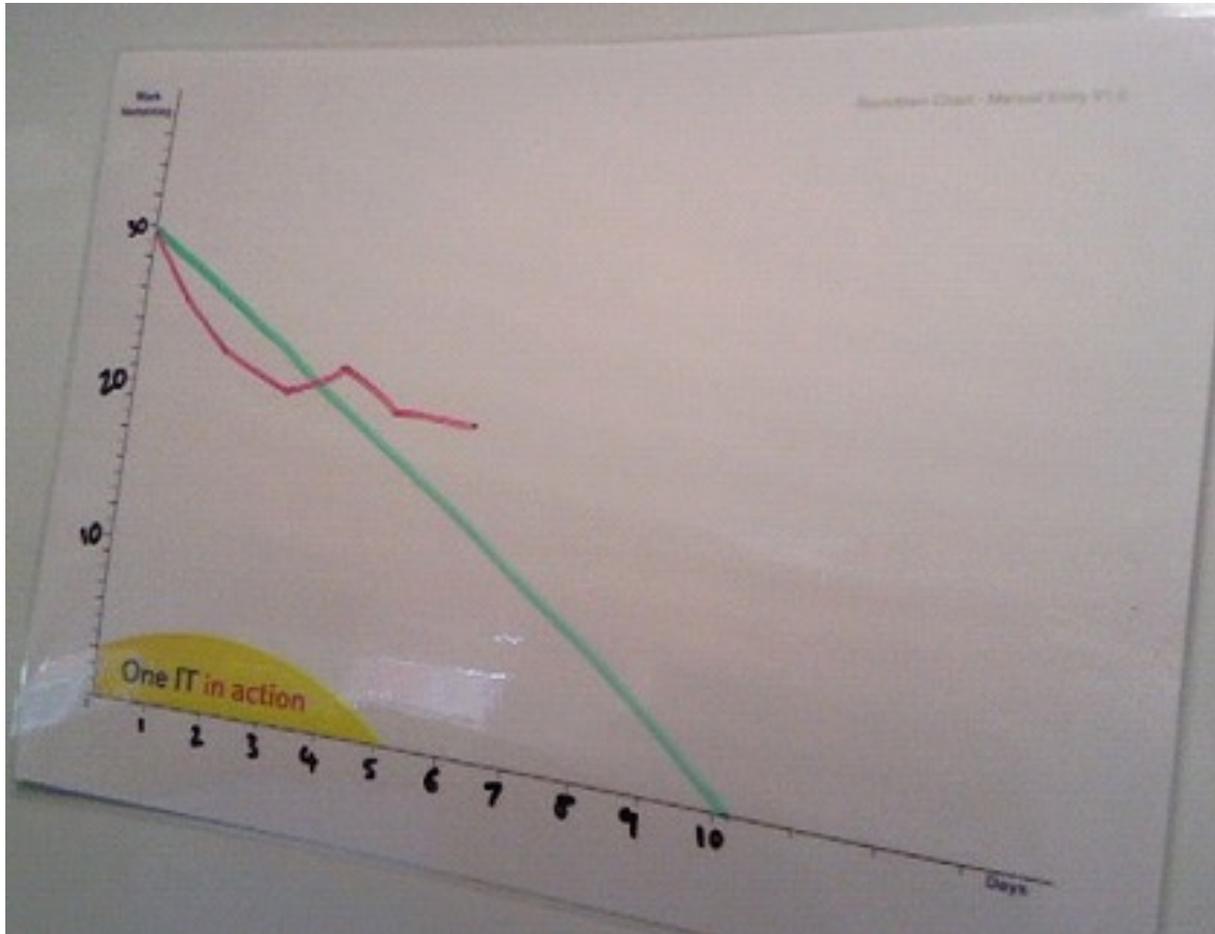


Gráfico de *burndown*

- Instrumento principal de monitoramento em métodos ágeis
- Tem como foco a iteração corrente
- Mostra esforço x tempo
 - Eixo X: tempo estimado para a iteração
 - Eixo Y: esforço restante para a iteração
 - Linha diagonal: decaimento de esforço assumindo linearidade

Gráfico de *burndown*



Fonte:
<http://www.scrumalliance.org>

Preenchimento do gráfico de *burndown*

- A cada dia, marcar o ponto no gráfico equivalente ao momento atual
 - Decaimento representa o quanto de trabalho previsto que já foi feito
- Conectar o ponto atual com o ponto do dia anterior com uma linha
- Analisar a situação atual do desenvolvimento e tomar as medidas necessárias

Análise do Gráfico de *burndown*

- O que significa quando o gráfico está mais alto que a linha de decaimento linear?
 - Atraso no cronograma
- O que significa quando o gráfico está mais baixo que a linha de decaimento linear?
 - Adiantamento no cronograma
- O que fazer em cada um desses casos?
 - Mitigar riscos
 - Refazer estimativas
 - Possibilidade de adicionar ou remover tarefas

Exemplos de Desenvolvimento Ágil

- Spotify
 - <https://labs.spotify.com/2014/03/27/spotify-engineering-culture-part-1/>
 - <https://labs.spotify.com/2014/09/20/spotify-engineering-culture-part-2/>

Exercício

- Faça um gráfico de *burndown* de uma das iterações do trabalho do curso
 - Listar as tarefas da iteração
 - Desenhar o gráfico de *burndown*

Referências

- Kniberg, H.; 2007. Scrum and XP from the Trenches: how we do Scrum; InfoQ.
- Pilone, D.; Miles, R.; 2008. Head First Software Development. O'Reilly Media.

Gerenciamento ágil

Leonardo Gresta Paulino Murta
leomurta@ic.uff.br