Artefatos Ágil

por Adriana Silveira de Souza



Sumário

- Revisão da aula anterior
- RH no Scrum
- História de Usuários
- Critérios de Aceitação de Histórias
- Backlog do produto
- Planning Poker
- Kanban
- Burndown

Scrum framework

Papéis

- Dono do produto
- ScrumMaster
- Equipe

Processo

- Planejamento
- Revisão e Retrospectiva
- Reunião diária

Artefatos

- Product backlog
- Sprint backlog
- Gráfico de Burndown

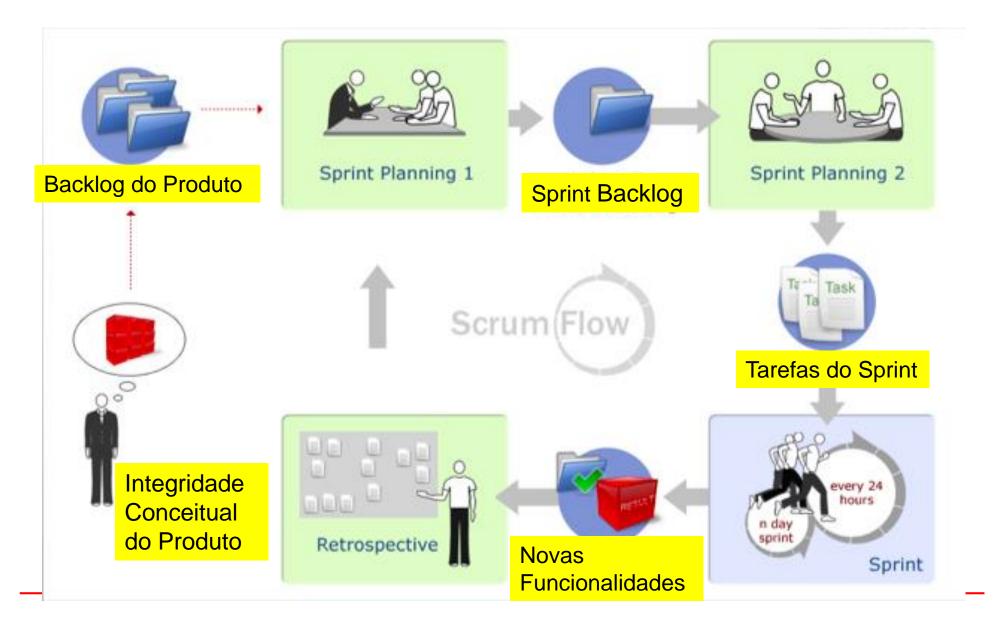
Filosofia do Scrum

- Scrum é orientado a comprometimento
 - → Mantenha os porcos e livre-se das galinhas
 - → Porcos são aqueles que perdem o emprego em caso de insucesso!
- Scrum é orientado a resultados
 - → Todo sprint deve gerar software pronto para uso
 - → Não necessariamente o software completo
- Scrum é disciplinado
 - → Caixa de tempo deve ser respeitada
 - → Há cerimônias executadas segundo um cronograma

Bases do Scrum

- Trabalho em equipe unida, integrada e autogerenciada
 - → O cliente é um contribuidor chave para o sucesso do projeto
- Mudanças são aceitas como vantagem competitiva
 - → Mas não durante o sprint
- Comunicar bem é melhor que programar bem
 - → Transparência sem burocracia; documentação deve ser útil
- Alta qualidade é a chave para a alta velocidade
 - → Pensar de forma simples, eficiente, e incremental
- Uma boa equipe em um bom ambiente
 - → Responsabilidade, honestidade, confiança e comprometimento

Padrão de Processo Scrum



Gestão de Pessoas no Scrum

- Relação entre Papéis no Scrum
 - → autoridade e responsabilidades
- Dono do Produto
- Scrum Master
- Time Multifuncional
 - → planejamento do trabalho no Sprint
- Envolvimento e Comprometimento
 - → o conceito de "Pronto" (ou "Terminado")

Equipes auto-organizadas

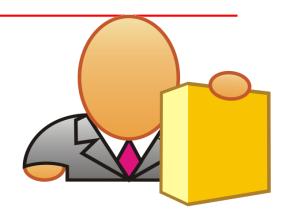
- Embora essas equipes de projeto estejam em grande parte por conta própria, elas não estão descontroladas
- A gerência estabelece pontos de controle suficientes para evitar a instabilidade, a ambiguidade e a tensão que pode se transformar em caos
- Ao mesmo tempo, a gerência evita o tipo de controle rígido que prejudica a criatividade e a espontaneidade
 - Takeuchi &Nonaka, "The New Product Development Game", Harvard Business Review, Jan/1986

Dono do produto

- Define as funcionalidades do produto
 - → Deve ter autoridade para a decisão

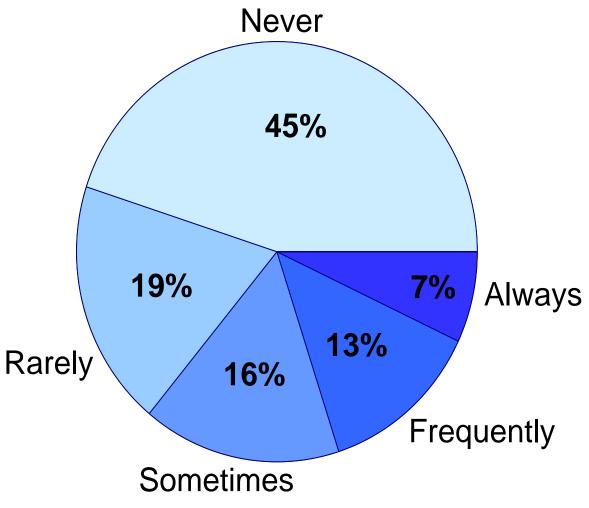


- → Prioriza funcionalidades de acordo com o valor de mercado
- Responsável pela rentabilidade (foco no ROI)
 - → Ajusta funcionalidades e prioridades
- Aceita ou rejeita o resultado dos trabalhos
 - → Negocia mudanças sugeridas pela equipe
- Está sempre disponível para a equipe
 - → Comprometimento com a equipe e com a organização
- Entende muito do negócio !!!!



Frequência de Uso do Software Entregue

No Scrum, o Dono de Produto é o responsável por inverter este cenário



Fonte: The Standish Group (2002).

ScrumMaster

- Representa
 - → a gerência para o projeto
 - → a equipe para a gerência da organização
- Responsável pela aplicação dos valores e práticas do Scrum
 - → Remove obstáculos
 - → Garante a colaboração entre os diversos papéis e funções
- Garante a plena funcionalidade e produtividade da equipe
 - → Compromentimento da equipe com a organização
- Escudo para interferências externas
 - → Comprometimento da organização com a equipe



Perfil do Scrum Master

- Inspira confiança no time
- Promove abertura, transparência e colaboração
- Sabe reconhecer um bom backlog para sua equipe
- Tem experiência em projetos ágeis
- Faz as perguntas certas para ajudar o projeto
- Compreende os desafios técnicos da equipe
- Entende os desafios do negócio na perspectiva do DP

Desafios para o Scrum Master

- Problemas típicos em projetos de software
- Cliente ausente
- Elo fraco na equipe
- Pouca familiaridade da equipe com a tecnologia usada
- Estimativas otimistas
- Influência de fatores externos ao projeto
- Mudanças no backlog do sprint por pressão do cliente
- Falta de conhecimento para teste ou validação efetiva
- Problemas de comunicação e colaboração
- Múltiplos donos de um produto
- Sugestão de aborto ao primeiro sinal de dificuldade

Equipe

- Entre 5 e 9 pessoas
- Tecnicamente Multi-funcional
 - Programadores, testadores, desenvolvedores de interfaces, e tudo o que a empresa define como padrão
- Tempo integral
 - → Raras exceções (Ex.: DBA)
- Auto-organizável
 - → Idealmente, sem títulos
 - Todos da equipe devem ser capazes de desenvolver código
- Trocas só na mudança de Sprints

Princípios para times ágeis

- Composição da Equipe
 - → Todos são equipe, e a equipe é formada por todos
 - Se ajuda o time, faz parte do time
 - → Não há hierarquia ou relação de comando e controle
- Mais Generalistas do que Especialistas
 - → Adaptação é essencial para o time
 - → Se o time precisa de você para um papel, você deve desempenhar esse papel
 - · Um time campeão, em vez de um time de campeões
- Reuniões diárias + quadros visíveis
 - → Um por todos, e todos por um!

Histórias de Usuários (User Stories)

- São artefatos de desenvolvimento utilizados em sistemas geridos segundo metodologias ágeis.
- Histórias de Usuário são iguais Casos de Uso?
- Como descrevo minhas Histórias de Usuário?
- Que tipo de informações podemos inserir nas Histórias de Usuário?
- De quem e pra quem são feitas?

- Casos de Uso e Histórias de Usuário são similares
 - → Ambos são utilizados para organizar requisitos
 - → Casos de Uso descrevem ações de interação segundo uma <u>narrativa impessoal</u> entre o <u>usuário e o</u> <u>sistema</u>
 - → Histórias de Usuário focam nos <u>objetivos do us</u>uário e como o <u>sistema alcança esses objetivos</u>.
- Histórias de Usuário <u>fracionam</u> os requisitos para que seja possível (e mais fácil) <u>estima</u>r o esforço para <u>realizar aquele objetivo</u>

- Resumindo, Histórias de Usuário são descrições simples que descrevem uma <u>funcionalidade</u> e é recomendável que sejam escritas segundo o <u>ponto de vista do usuário</u>
- Histórias de Usuário devem ser <u>curtas</u>, <u>simples e</u> <u>claras</u>. Devemos conseguir escrevê-las em um simples e pequeno cartão (conhecidos como *User Index Cards*)
 - → Se não há espaço para escrevê-la em um cartão é porquê devemos refiná-la mais, e as dividir em outras Histórias de Usuário.

Como um [ator]
 eu quero/preciso
 de/devo/gostaria
 de [ação] para
 [funcionalidade]

| Pescrici Pomo ve | ndedor en postaria de |
|---------------------|--|
| erificar | ndedor en gostaria de se um livro está disponível ue para venda. |
| o estoq | ue para venda. |
| | |
| | |
| | |
| MIT HOST | |
| | |

- Como um [ator] eu quero/preciso de/devo/gostaria de [ação] para [funcionalidade]
 - → Exemplo:
 - Como um vendedor responsável pelo setor de livros eu quero procurar por livros filtrando por nome para que seja possível verificar se o livro X está disponível para pronta entrega.

Estratégia TI 2.

Como vendedor en gostaria de

ficar se um livro está disponível

- ✓ Ator O proprietário da História de Usuário. De forma simplista é o usuário, o interessado naquela funcionalidade. Mas é recomendado descrever de forma específica quem é o ator para ser mais fácil identificar o contexto da história dentro do sistema.
- Ação É o que o ator quer fazer. Utilizando aquela ação ele espera alcançar seu objetivo dentro do sistema.
- ✓ Funcionalidade É o que o <u>ator espera que aconteça</u> <u>ao realizar a ação</u>. Ou seja, é o <u>resultado</u> de executar a ação segundo a ótica do ator. Também pode ser visto como justificativa.

Histórias de Usuários – Aspectos importantes

- Validar se a funcionalidade é realmente necessária antes de incluí-la
- Análise das necessidades reais e práticas do usuário (foco)
- Ajuda a priorizar o que deve ser feito
- É mais fácil estimar o esforço que será necessário para implementar a funcionalidade
 - → O interessante no uso de Histórias de Usuário é o foco nas necessidades reais e práticas do usuário

Imagine um sistema de aluguel de filmes pela internet, onde uma das funcionalidades do sistema é visualizar os filmes disponíveis para locação.

Imagine um sistema de aluguel de filmes pela internet, onde uma das funcionalidades do sistema é visualizar os filmes disponíveis para locação

Então a História de Usuário para esse caso seria:

Exemplo:

→ Como um cliente eu quero ver os filmes disponíveis para locação que eu possa aluga-lo

Agora, imagine o seguinte: além de visualizar os filmes disponíveis para locação, seria interessante que o cliente pudesse reservar também o filme desejado

Agora, imagine o seguinte: além de visualizar os filmes disponíveis para locação, seria interessante que o cliente pudesse reservar também o filme desejado.

Nesse caso a História de Usuário seria:

Como um cliente eu quero ver os filmes disponíveis para locação para que eu possa agendar uma reserva na data X.

- A pergunta é: será que essa História de Usuário é realmente útil agora de forma a atender ao que os usuários desejam mais nesse momento?
- Perguntas como essa surgem constantemente no processo de planejamento das sprints e cabe ao PO pensar sobre a necessidade de implementar ou não tal funcionalidade nesse momento.

Exercício

- Definir histórias de usuário para confeccionar um bolo de chocolate com recheio de coco e cobertura de calda de chocolate com bis.
- Deve ser particionado em pelo menos 3 histórias
- Deve ser definido os papéis na equipe (PO, SM e Time)

Histórias - sugestão

- Como Confeiteira, eu preciso separar os ingredientes para produzir o bolo.
- Como confeiteira, eu preciso misturar os ingredientes para produzir a massa do bolo.
- Como confeiteira, eu preciso misturar os ingredientes para produzir a cobertura do bolo.
- Como confeiteira, eu preciso misturar os ingredientes para produzir o recheio do bolo.
- Como confeiteira, eu preciso montar a massa, o recheio e a cobertura para que o bolo seja entregue.

- São representados por uma lista de itens de negócio que expressam formas de usar a funcionalidade implementada em uma História.
- Objetivo dessa lista é <u>validar</u> se a História foi implementada de acordo com o que o PO queria
- Na reunião de Revisão, a equipe apresentará História a História para o PO e com base nos Critérios de Aceitação
- As histórias são validadas quanto ao funcionamento da mesma

- Esses Critérios (itens) surgem de <u>perguntas</u> que a equipe faz ao <u>PO</u>, no momento em que a História está sendo descrita, na <u>busca</u> por obter mais <u>detalhes</u> do que deve ser implementado
- Algumas perguntas podem ser feitas ao PO, quanto a usabilidade dessa funcionalidade, como por exemplo:
 - → Como o usuário fará essa busca no sistema?
 - → Como o usuário deseja que seja apresentada a informação solicitada?
 - → O que deve ser feito se o livro não for encontrado?

- História
- Como vendedor eu gostaria de verificar se um livro está disponível no estoque para venda

- História ("Como vendedor eu gostaria de verificar se um livro está disponível no estoque para venda")
- Critérios de aceitação:
 - → O vendedor não pode solicitar a busca se não informar o nome do livro
 - → O sistema, encontrando o livro, deve apresentar nome completo, autores, editora, ano de edição e a quantidade em estoque
 - → O sistema, não encontrando o livro, deve informar que o livro não foi encontrado

- Como base nos Critérios de Aceitação a equipe deve escrever os <u>Testes de aceitação</u>
- Ao final do desenvolvimento são esses testes que dirão se a funcionalidade foi implementada de forma a ser <u>aceita ou não pelo PO</u>.
- Se os testes forem criados com uso de ferramentas específicas para automatização da execução dos mesmos, isso garantirá maior agilidade e qualidade do produto

- Incluir os critérios de aceitação como parte da história só vem a agregar vantagens, como:
 - Prover material para a <u>equipe pensar</u> em como uma <u>funcionalidade será executada</u> pelo ponto de vista do usuário
 - → Eliminar ambiguidades quanto aos requisitos.
 - → Confirmar que a história está completa e funcionando
 - → Garantir maior <u>satisfação do usuário</u>

Resumindo...

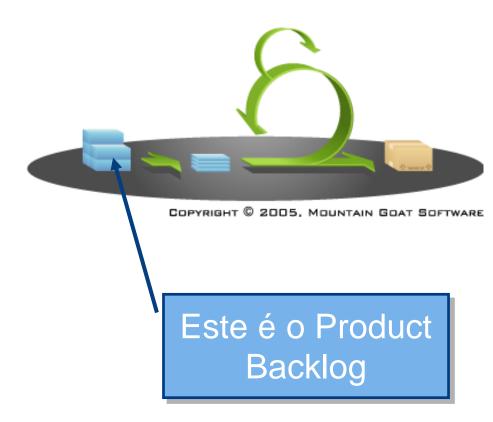


"Escrever Histórias, é muito mais do que apenas descrevê-las. É também especificar a usabilidade da mesma, com base nos Critérios de Aceitação"

Exercício

- Definir os critérios de aceitação para cada História de Usuário do nosso Bolo de Chocolate.
- Lembrem-se que:
 - → Os critérios surgem de perguntas que a equipe fez para o PO
 - No momento em que a História está sendo descrita ou apresentada na busca por obter mais detalhes do que deve ser implementado

Product Backlog



- Os requisitos
- Uma <u>lista de todo o trabalho</u> desejado no produto
- Idealmente, na forma em que cada item tenha seu peso de acordo com a vontade dos clientes ou usuários
- Priorizado pelo dono do produto
- Repriorizado no início de cada Sprint
- Aprimorado nas reuniões de revisão

Exemplo de Product Backlog

| Item do Backlog | Importância |
|--|-------------|
| Permitir que o usuário faça uma reserva | 3000 |
| Permitir cancelamento de reserva | 500 |
| Permitir a troca de datas da reserva | 300 |
| Permitir que empregado do hotel gere relatórios de lucratividade | 80 |
| Melhorar manipulação de erros | 50 |
| | 30 |
| ••• | 20 |

Outro Exemplo de Backlog de Produto

| | Item # | Description | Est | Ву |
|-----------|--------------|--|-----|------|
| Very High | Kom :: | Coonbaon | 200 | |
| very mgn | T 1 | Finish database versioning | 16 | KH |
| | | Get rid of unneeded shared Java in database | 8 | KH |
| | | Add licensing | ⊢° | КП |
| | 3 | | 16 | TG |
| | 4 | • | 16 | TG |
| | 1 " | | 10 | 10 |
| | | Analysis Manager | | |
| | 5 | | 160 | TG |
| | 6 | Round-trip Analyses | 250 | MC |
| High | | | | |
| | - | Enforce unique names | - | - |
| | 7 | | 24 | KH |
| | 8 | In import | 24 | AM |
| | - | Admin Program | - | - |
| | 9 | Delete users | 4 | JM |
| | - | Analysis Manager | - | - |
| | | When items are removed from an analysis, they should show | | |
| | 10 | , , | 8 | TG |
| | - | Query | - | - |
| | 11 | | 16 | T&A |
| | 12 | | 16 | T&./ |
| | 13 | | 12 | T&A |
| | - | Population Genetics | - | - |
| | 14 | | 400 | T&N |
| | 15 | , | 400 | T&N |
| | 16 | | 240 | T&N |
| | 17 | , | 240 | T&N |
| | 18 | | 320 | T&N |
| | 19 | Add icons for v1.1 or 2.0 | - | - |
| | - | Pedigree Manager | - | - |
| | 20 | Validate Derived kindred | 4 | KH |
| Medium | | | | |
| | - | Explorer | - | - |
| | | Launch tab synchronization (only show queries/analyses for | | |
| | 21 | logged in users) | 8 | T&A |
| | 22 | Delete settings (?) | 4 | T&A |

Planning Poker

- É uma técnica baseada no consenso para estimar, é um jogo e ao mesmo tempo um exercício
- Baseado no método Delphi Especialistas
- Podemos estimar o esforço necessário para determinada quantidade de trabalho
 - → tendo como base informações recolhidas do cliente por meio das histórias de usuário.
- Criado por James Grenning em 2002
- Popularizado por Mike Cohn, no livro Agile Estimating and Planning.

Planning Poker

- Cada membro da equipe recebe um conjunto de cartas, com os valores de uma determinada sequência.
- Com base na história de usuário analisada, cada membro da equipe joga uma carta com a face para baixo sobre a mesa, nela estará contido o valor numérico de pontos que o mesmo considera justo para que a estória seja concluída.

Planning Poker

- Caso haja grandes diferenças entre as cartas jogadas, os membros que jogaram as cartas de maior e menor valor explicarão suas razões
- Com base em suas explicações, as cartas são jogadas novamente até que um consenso seja encontrado e uma estimativa seja definida.

$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55...$$

Planning Poker - Baralho

| ? | 0.5 | 0 | 1 |
|----|-----|-----|----------|
| 2 | 3 | 5 | 8 |
| 13 | 21 | ••• | <u> </u> |

- ? (interrogação): Significa que o membro não se sente confiante para atribuir um valor a tarefa;
- 0 (zero): Significa que a tarefa é absolutamente desnecessária e deveria ser descartada;
- 0.5 (meio): Significa que a tarefa necessita de uma pequeno esforço para ser concluída;
- ... (infinito): Significa que a tarefa é extremamente importante;
- Xícara de café: Significa uma pausa para refletir antes de tomar a decisão. Esta pausa é importante e deve ser respeitada quando solicitada, é muito provável que os membros não abusem dela.

Exercício

- Definir a pontuação utilizando o planning poker para cada História de Usuário do nosso Bolo de Chocolate.
- Lembrem-se que:
 - → A time deve definir o que venha ser um ponto;
 - → Continuar a incentivar a discussão até que um consenso geral entre os membros seja obtido;
 - → Fazer uma média dos valores, levando em consideração a proximidade entre eles;
 - → Como os valores estão próximos, assumir o maior valor.

Planejamento do Sprint

- A equipe aceita itens prioritários do Product Backlog e se compromete a atendê-los
- O Sprint Backlog é criado
 - Tarefas identificadas e estimadas (1 a 16 horas)
 - De forma colaborativa pela equipe, com apoio do ScrumMaster
- Planejamento de alto nível é considerado

Quero que os usuários do portal possam planejar suas férias, escolhendo itinerários online Modelagem (8 horas)

Codificar interface (4)

Escrever textos (4)

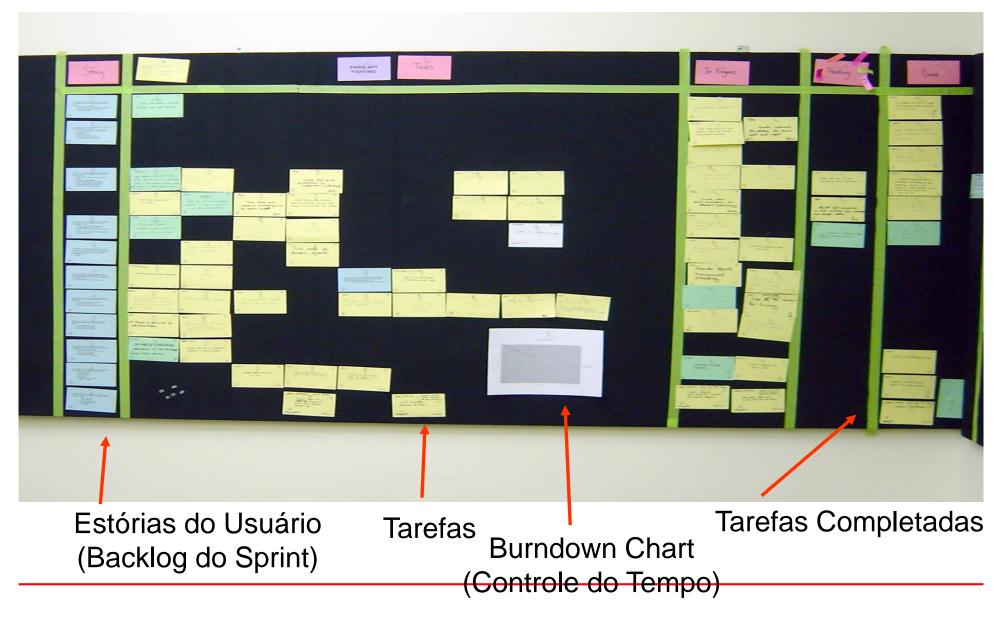
Codificar a classe Férias (6)

Atualizar testes de regressão (4)

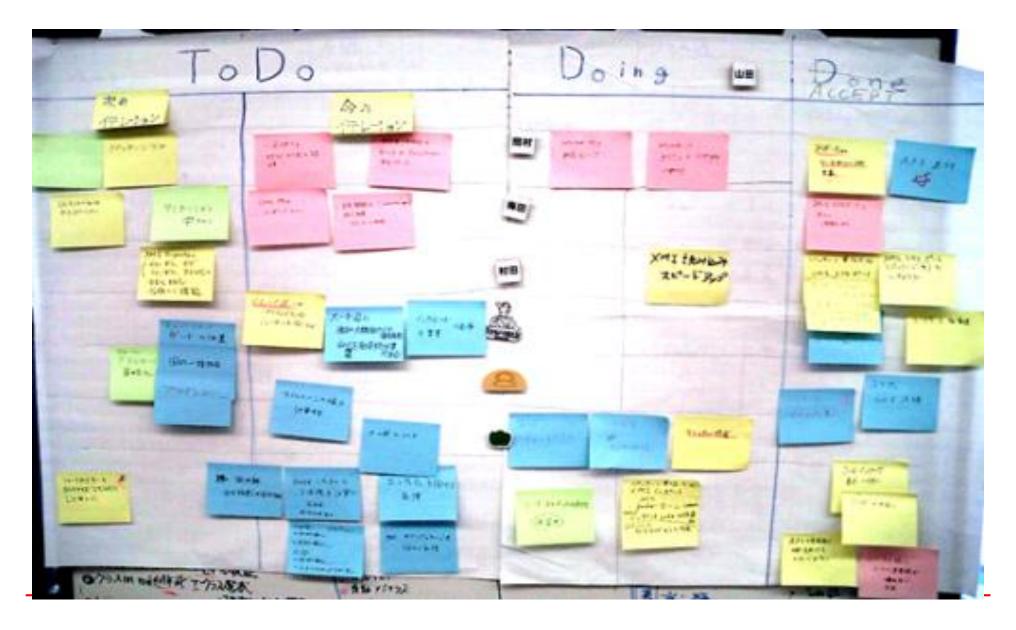
Estratégia TI

48

Kanban - Painel de Controle



Painel de Controle Real

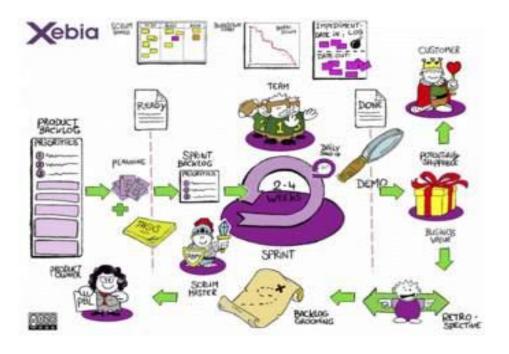


Exercício

- Fazer uma reunião de Sprint Planning 2
- Desmembrar a história de usuário em tarefas e anotar no post-it
- Colar na folha Kanban



É o gráfico utilizado pelas equipes Scrum para representar diariamente o progresso do trabalho em desenvolvimento.



Controle de Tempo

- O Burndown mostra o trabalho sendo "queimado"
 - Redução do número de horas estimadas (esforço restante) para completar o objetivo do Sprint
 - Exemplo: um trabalho de 600 horas = 6 pessoas x 20 dias x 5 horas/dia
 - A cada dia deve ser desenvolvido um trabalho de 30 horas
- O gráfico deve ser impresso de forma completa
 - Número de dias de duração do sprint
 - A velocidade e carga de trabalho ideais são mostradas como uma baseline



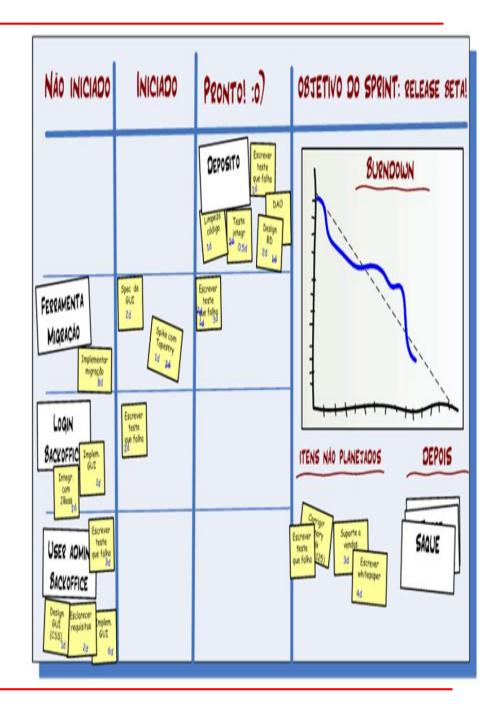






Exercício

- Construir o Burndown do bolo de Chocolate
- Para cada tarefa, jogar os dados
- O somatório dos dados será o resultado da produção em pontos pela equipe
- Marcar no gráfico de Burndown



Atividades para a próxima aula

- Definir o processo MPS.br Scrum, com as etapas, atividades e artefatos gerados
 - → Alimentar do backlog do produto
 - → Criar histórias
 - → Pontuar as histórias
 - → Planejar sp1
 - → Planejar sp2
 - → Executar
 - → Revisar
 - → Fazer retrospectiva
 - → Definir papéis

Atividade deve ser realizada em grupo

Deve ser postada no Moodle e entregue em papel na última aula

Atividades para a próxima aula

- Responder as questões a partir do texto: 2017
 Scrum Guide
- Atividade deve ser realizada individualmente
- Consiste na como revisão de tudo o que foi visto em sala de aula
- Deve ser postada no moodle