

Engenharia de Software

Prof. Vinicius Cardoso Garcia

vcg@cin.ufpe.br :: [@vinicius3w](https://twitter.com/vinicius3w) :: viniciusgarcia.me

[IF977] Engenharia de Software

<http://bit.ly/vcg-es>

Licença do material

Este Trabalho foi licenciado com uma Licença

Creative Commons - Atribuição-NãoComercial-
Compartilhagual 3.0 Não Adaptada



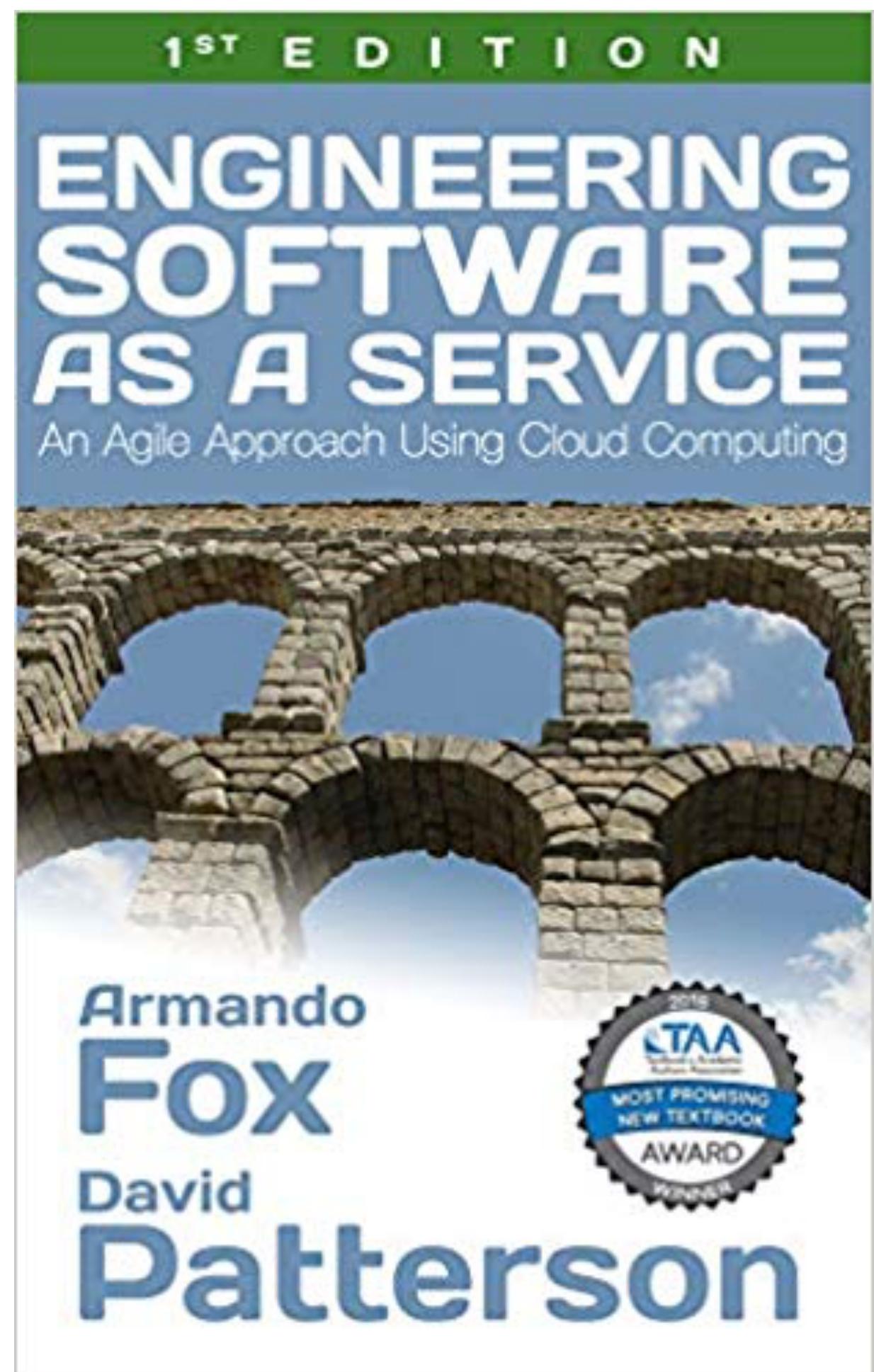
Mais informações visite

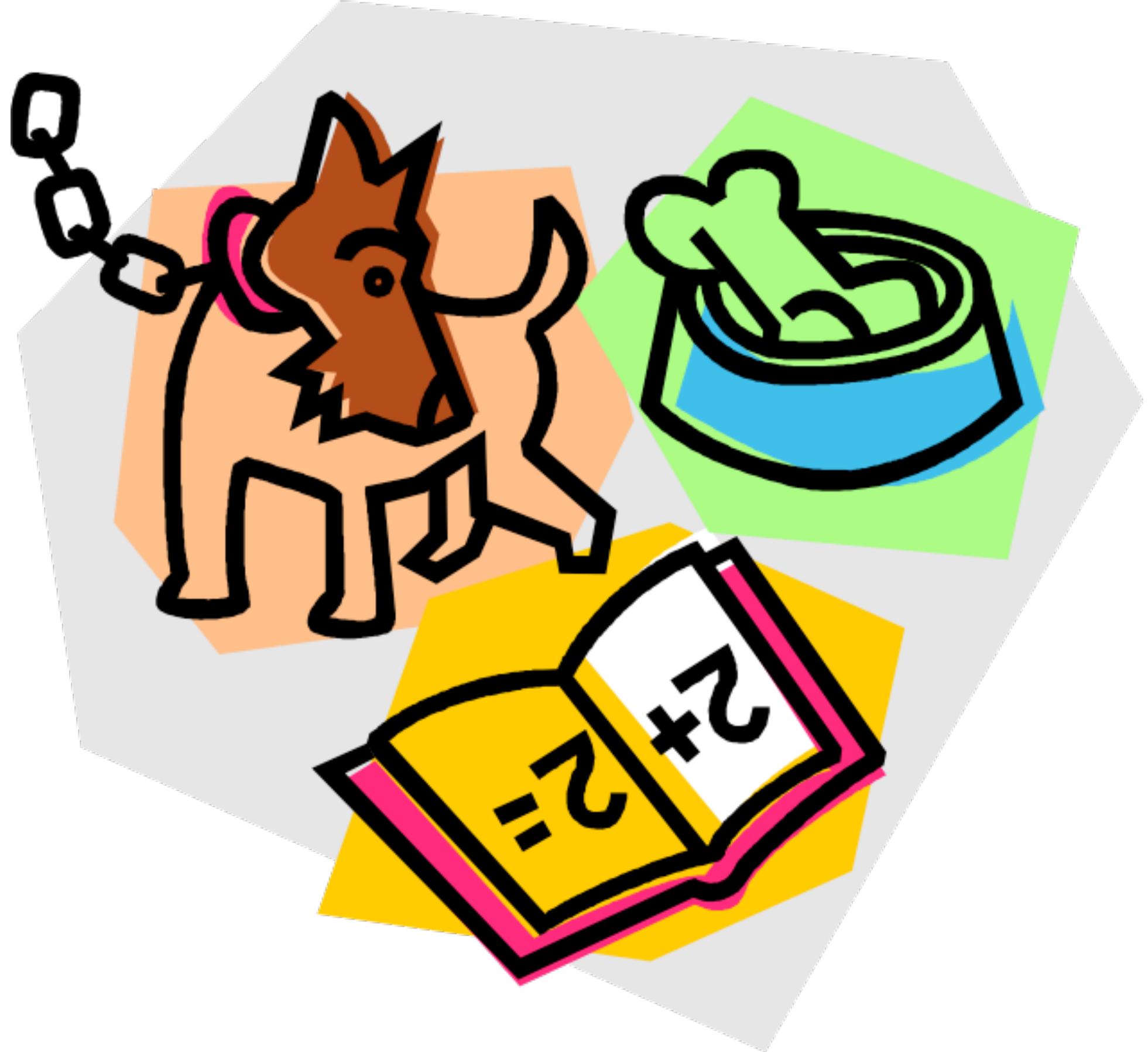
[http://creativecommons.org/licenses/by-nc-sa/
3.0/deed.pt](http://creativecommons.org/licenses/by-nc-sa/3.0/deed.pt)



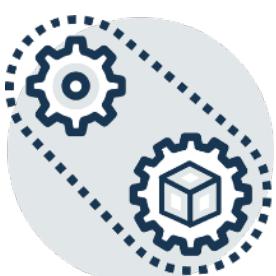
Referências

- A biblioteca do Desenvolvedor de Software dos dias de hoje
 - <http://bit.ly/31WYK5f>
- SWEBOK: Guide to the Software Engineering Body of Knowledge (SWEBOK)
 - <http://www.computer.org/web/swebok>
- Engineering Software as a Service: An Agile Approach Using Cloud Computing
 - <http://www.saasbook.info/>
- Marco Tulio Valente. Engenharia de Software Moderna
 - <https://engsoftmoderna.info/>





SMART User Stories

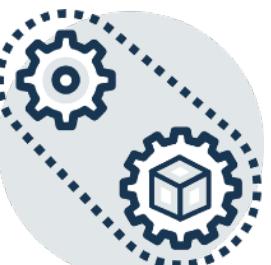


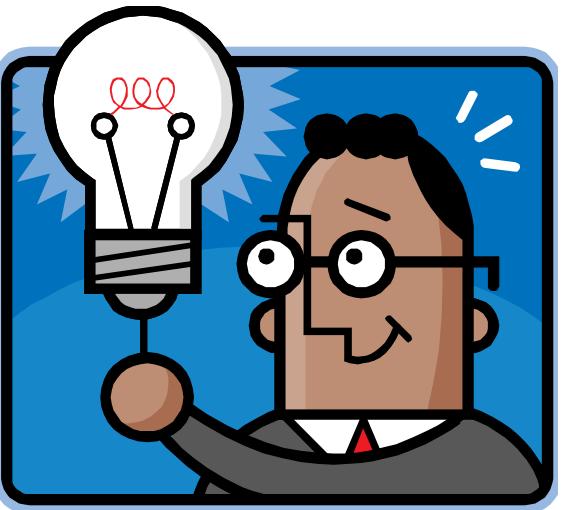
4



Criando Histórias do Usuário

- Como você sabe se você tem uma história do usuário boa ou ruim?
 - Tamanho certo?
 - Não é muito difícil?
 - Vale a pena?





SMART HU

- **S**pecific
- **M**easurable
- **A**chievable (idealmente, implementado em 1 iteração)
- **R**elevant (“os 5 porque's”)
- **T**imeboxed (saber quando desistir)



Specific & Measurable

- Cada cenário deve ser testável
 - Implica em conhecer boas entradas e existir os resultados esperados
- Anti-exemplo: "UI deve ser user-friendly"
- Exemplo: Given/When/Then.
 - 1.Dada alguma condição de partida específica(s),
 - 2.Quando eu faço X,
 - 3.Em seguida, um ou mais coisas deve acontecer



Specific, continuando

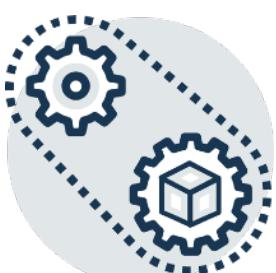
- "Dado que os ingressos para um filme estão disponíveis, eu devo comprar um ingresso"
- Qual é o foco principal da história?
 - disponibilidade de ingressos?
 - estado da conta do usuário?
 - fluxo de vendas?
- Criar cenários com base na história nos obriga a ser específicos, como veremos.



Achievable

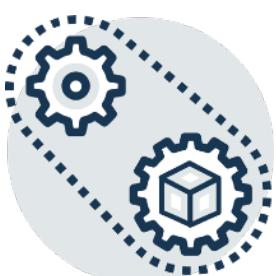


- Completar em 1 iteração
- Se não puder entregar recurso na 1 iteração, entregar subconjunto de histórias
- Sempre apontar para o código de trabalho @ final de iteração
- Se <1 história por iteração, precisamos melhorar estimativa pontual por história



Relevant: “business value”

- Descubra o valor de negócio, ou mate a HU:
 - Proteger as receitas
 - Aumentar a receita
 - Gerenciar o custo
 - Aumentar o valor da marca
 - Tornar o produto notável/reconhecido
 - Fornecer mais valor agregado aos seus clientes
- Um bom exemplo: <http://wiki.github.com/aslakhellesoy/cucumber>



5 Porquê's para encontrar relevância

- Show patron's Facebook friends
- As a box office manager
- So that I can induce a patron to buy a ticket
- I want to show her which Facebook friends are going to a given show
 1. Why?
 2. Why?
 3. Why?
 4. Why?
 5. Why?



5 Porquê's para encontrar relevância

1. Why add the Facebook feature? As box office manager, I think more people will go with friends and enjoy the show more.
2. Why does it matter if they enjoy the show more? I think we will sell more tickets.
3. Why do you want to sell more tickets? Because then the theater makes more money.
4. Why does theatre want to make more money? We want to make more money so that we don't go out of business.
5. Why does it matter that theater is in business next year? If not, I have no job.



Timeboxed

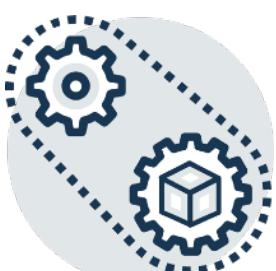


- Pare a HU quando ultrapassar orçamento ou o tempo
 - Desista, divida em histórias menores ou reagende o que for deixado de lado
- Para evitar subestimar o tamanho do projeto
- Pivotal Tracker rastreia a velocidade, o que ajuda a evitar esta subestimação



Pergunta

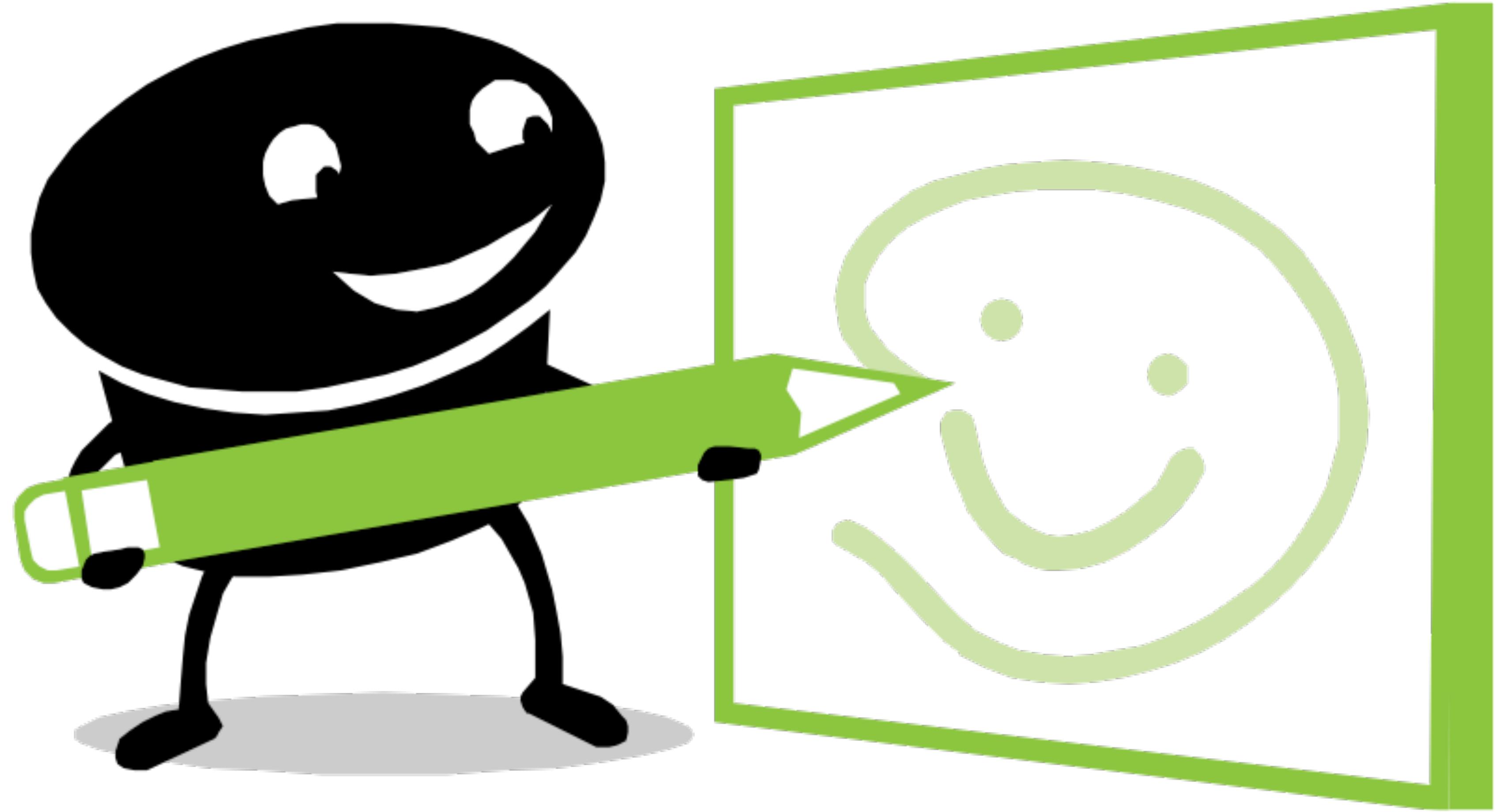
- Qual feature abaixo é a **MENOS SMART?**
 - A. Usuário pode buscar por um imóvel pelo título
 - B. Easylanding deve ter um bom tempo de resposta
 - C. Quando adicionar um imóvel, 99% das páginas "Add Property" devem ser exibidas em 3 segundos
 - D. Como cliente, eu quero ver os 10 imóveis mais locados, listados por preço, para que eu possa reservar o mais barato primeiro



Pergunta

- Qual feature abaixo é a **MENOS SMART?**
 - A. Usuário pode buscar por um imóvel pelo título
 - B. Easylanding deve ter um bom tempo de resposta
 - C. Quando adicionar um imóvel, 99% das páginas "Add Property" devem ser exibidas em 3 segundos
 - D. Como cliente, eu quero ver os 10 imóveis mais locados, listados por preço, para que eu possa reservar o mais barato primeiro

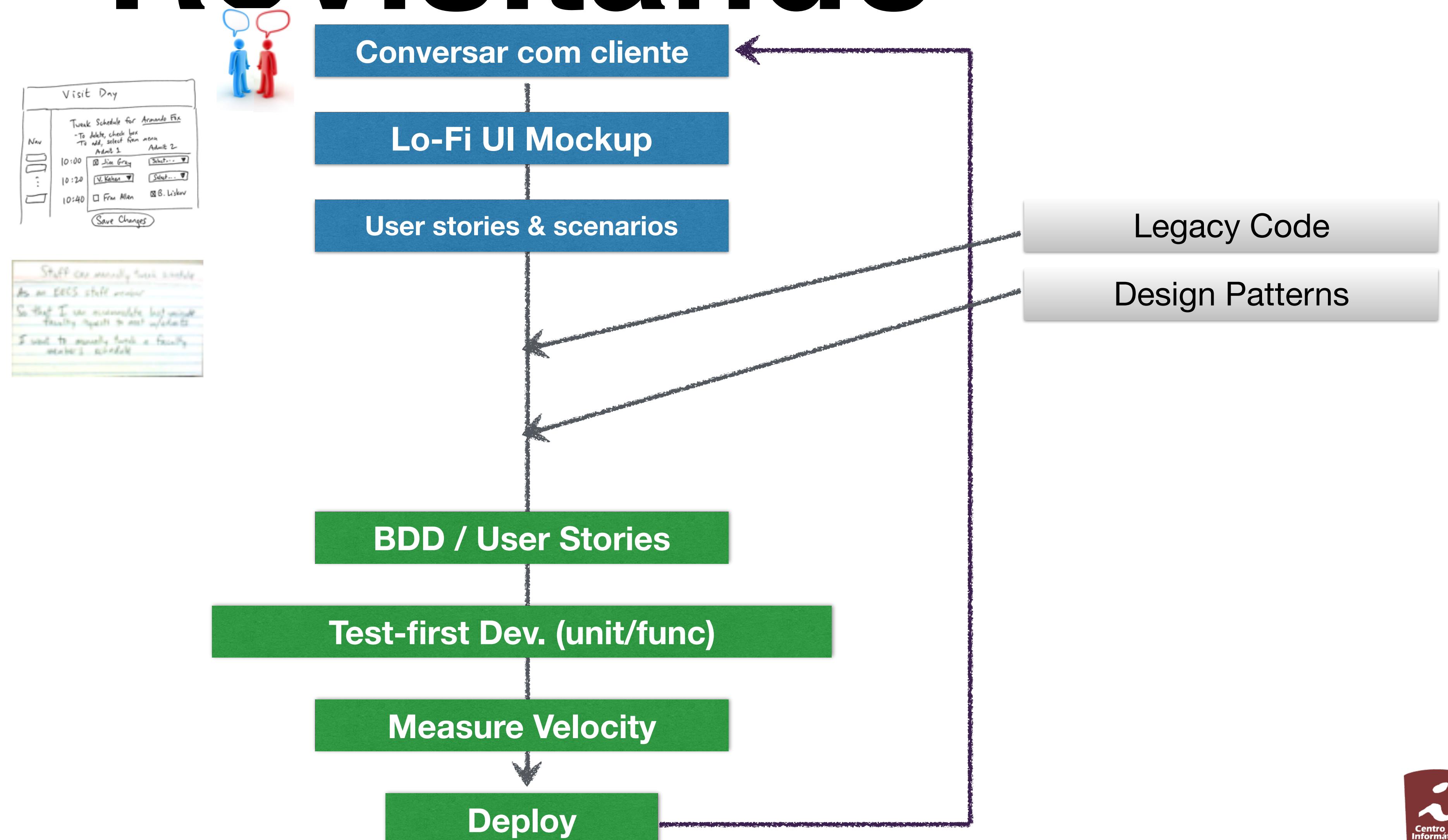




Lo-Fi UI Sketches and Storyboards



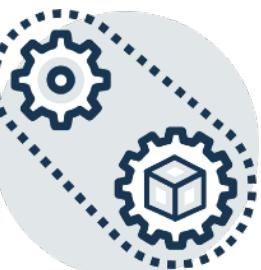
Revisitando



Construindo UI com sucesso

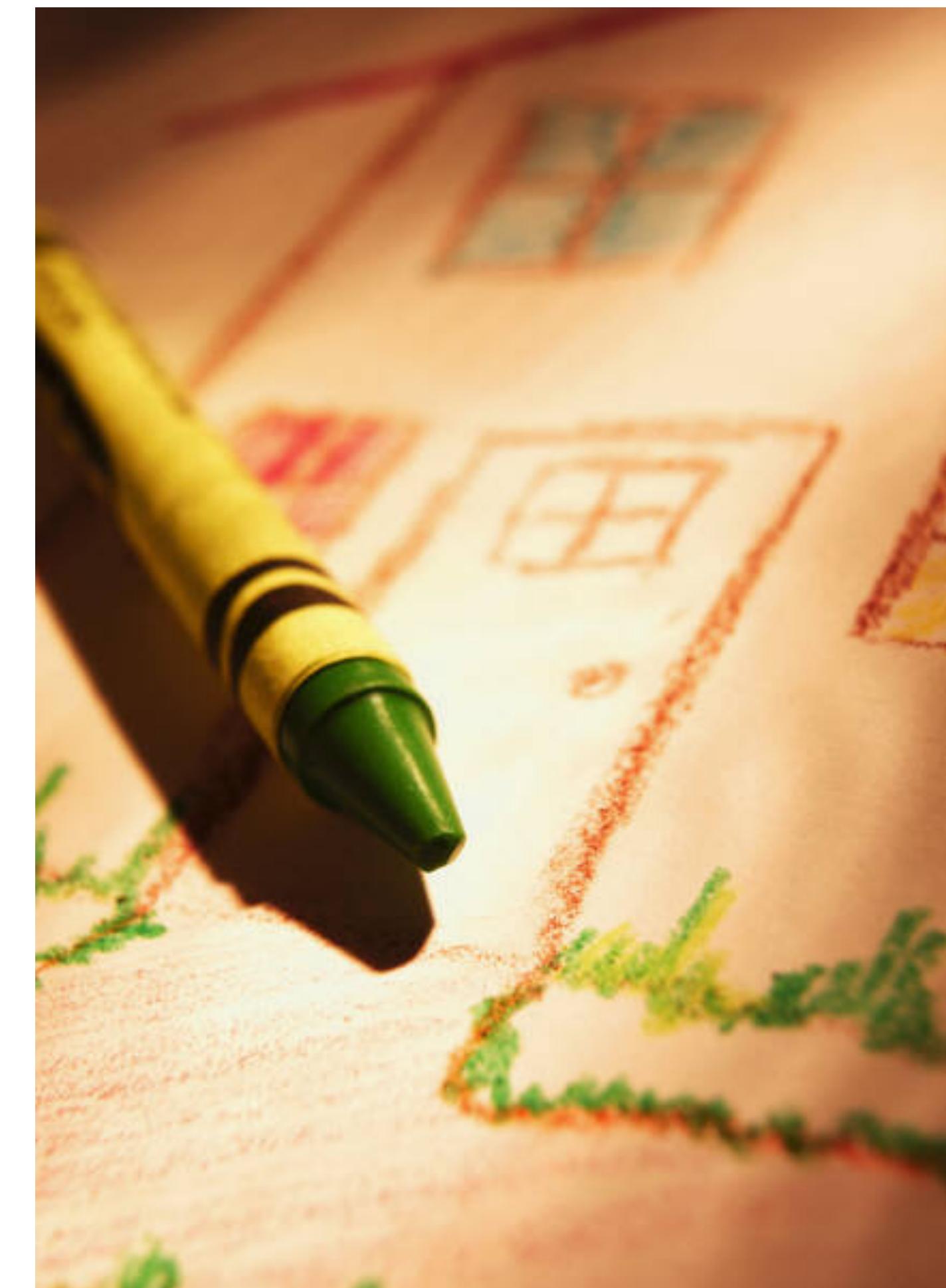
- Apps SaaS às vezes atrapalham usuários
- As histórias de usuário precisam de Interfaces de Usuário (UI - User Interface)
- Como conseguir clientes para participar de projetos de UI que se sintam felizes quando finalizado o projeto?
 - Evite WISBNWIW* em UI?
 - Versão UI de cartões 3x5?
- Como mostrar a interatividade sem a construção de protótipo?

* *What-I-Said-But-Not-What-I-Want*

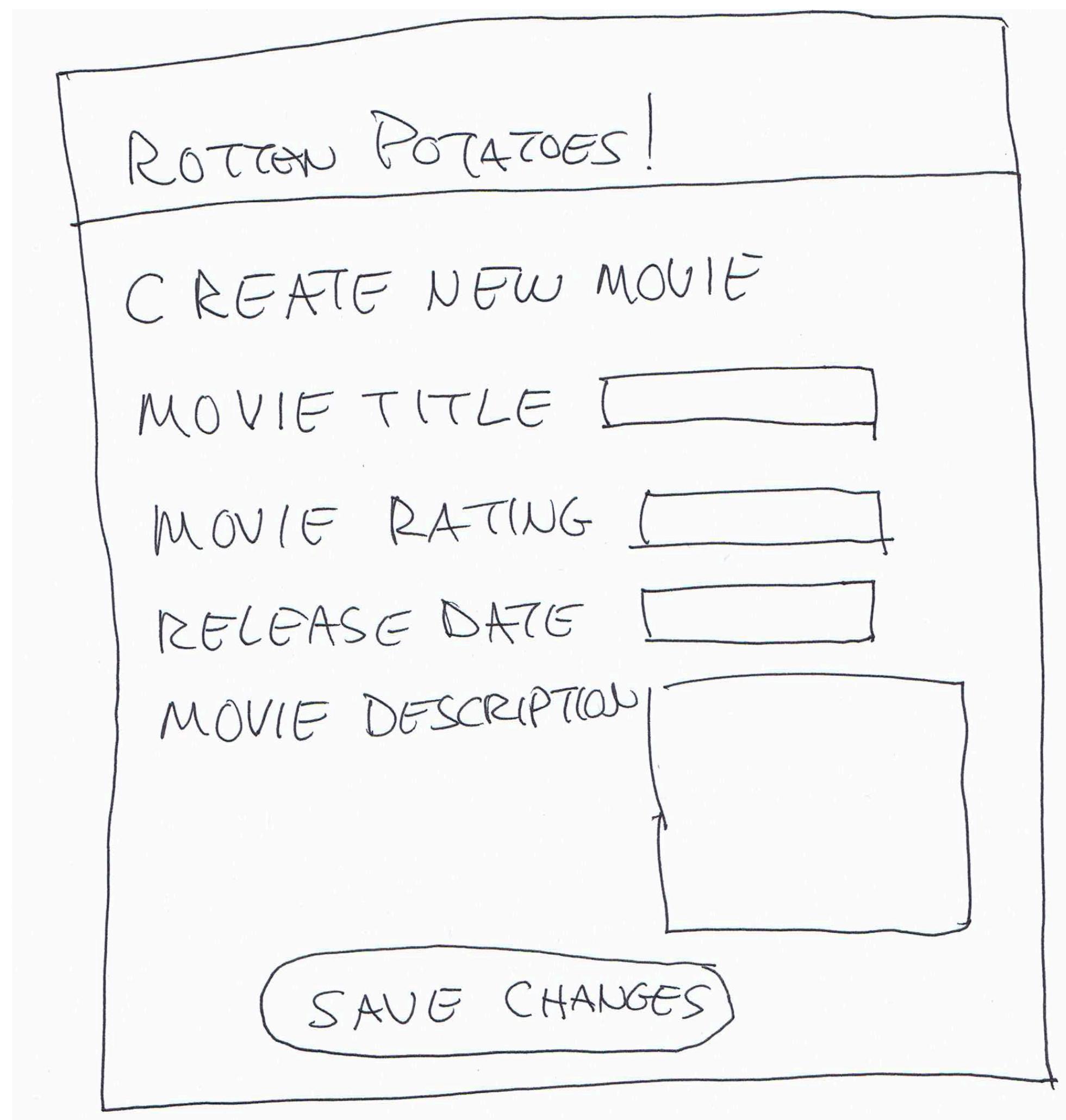


Projeto de UI em SaaS

- **Sketches UI**
 - lápis e papel para desenhar ou “**Lo-Fi UI**”



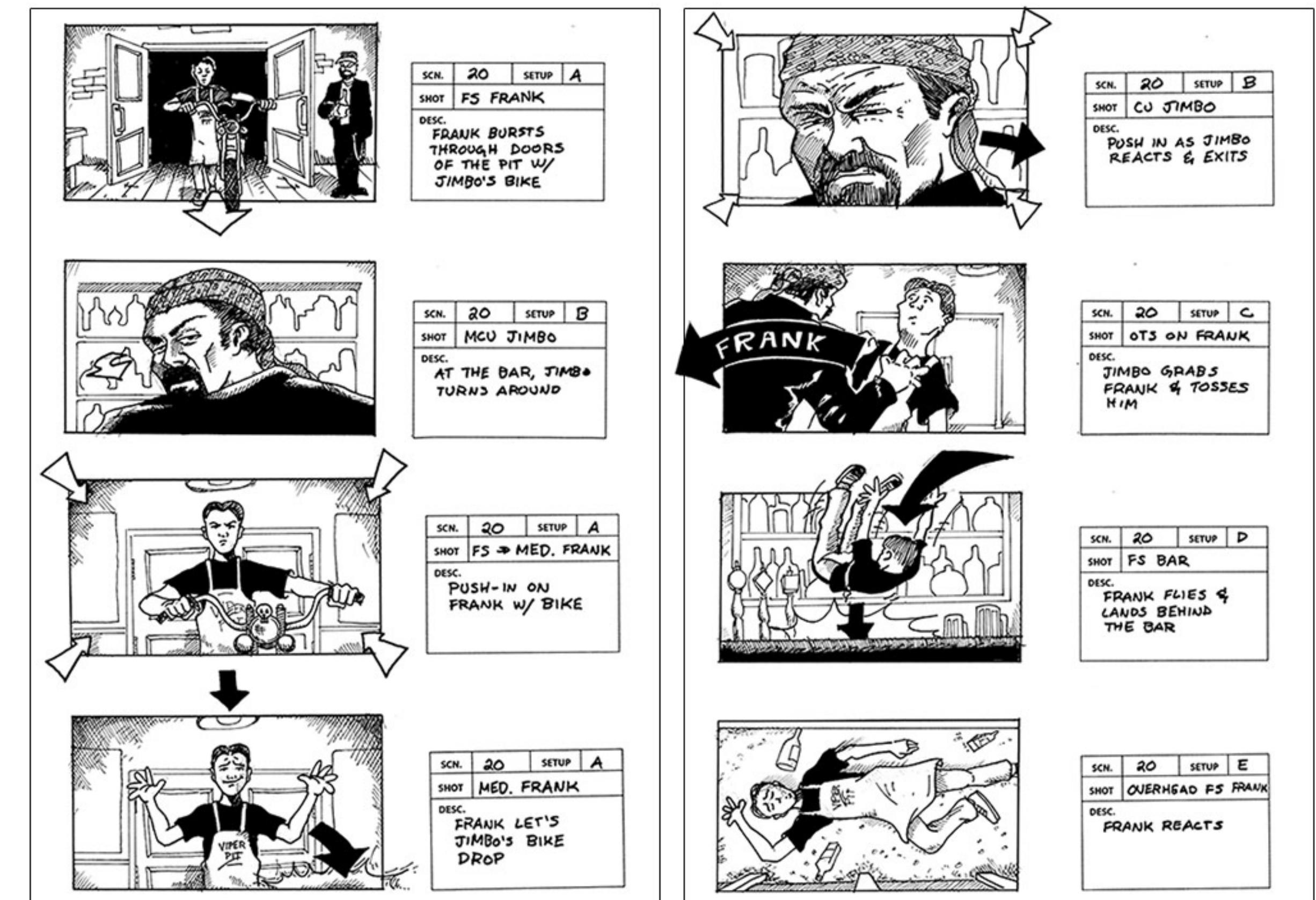
Exemplo de Lo-Fi UI



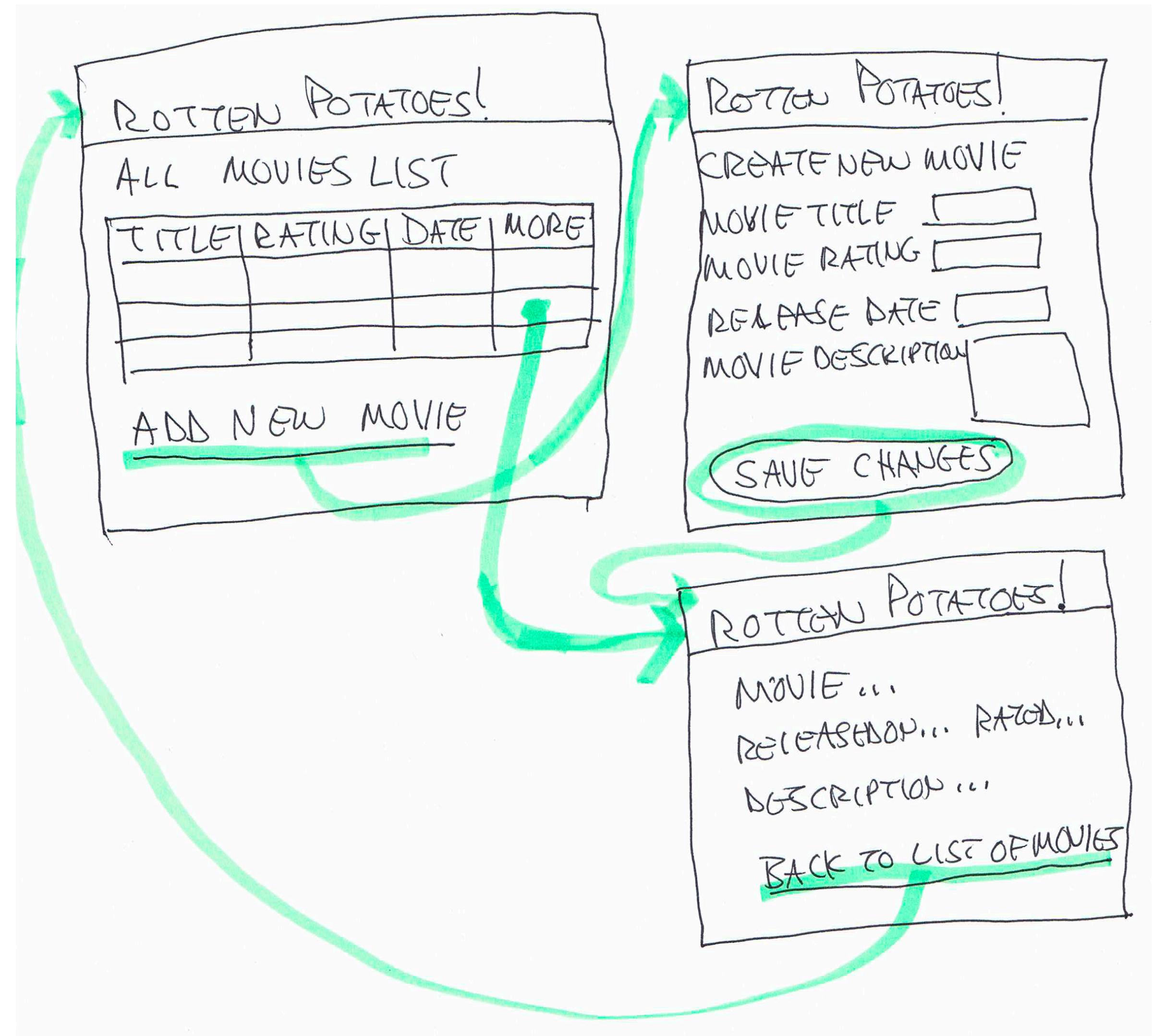
(Figure 4.3, *Engineering Long Lasting Software* by Armando Fox and David Patterson, Alpha edition, 2012.)

Storyboards

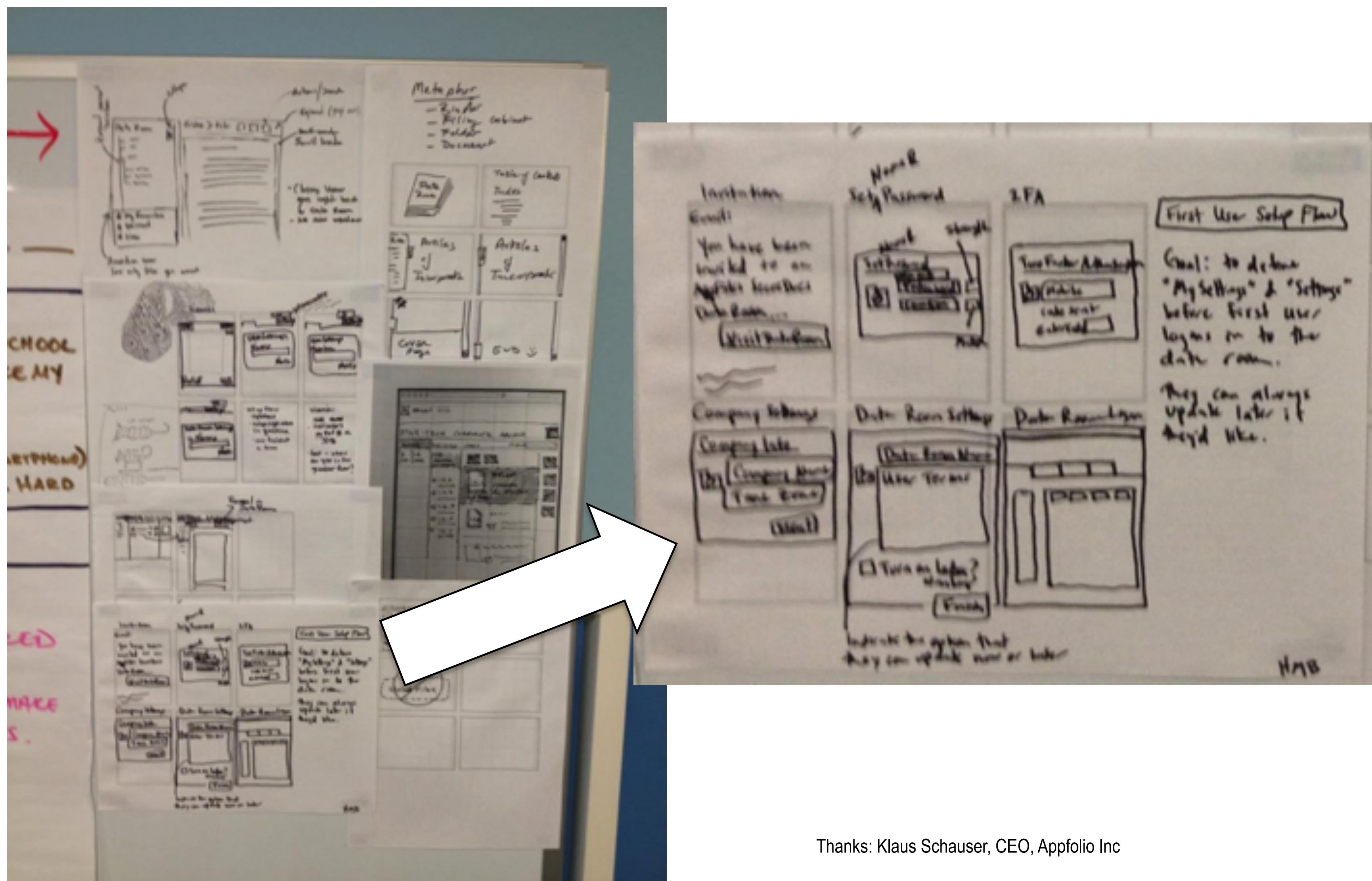
- Necessitam mostrar como a UI muda a partir de ações dos usuários
- IHC (HCI) => “storyboards”
- Como cenas em um filme
- Mas sem ser “linear”



Exemplo de Storyboard para Software



(Figure 4.4, *Engineering Long Lasting Software* by Armando Fox and David Patterson, Alpha edition, 2012.)



Thanks: Klaus Schauser, CEO, Appolio Inc

Lo-Fi para HTML

- É entediante construir sketches & storyboards, porém mais fácil do que construir HTML!
 - Além disso, **menos intimidador** para stakeholders **não-técnicos => incentivador** à sugestão de modificações de UI se não houver código por trás
 - Uma maior probabilidade de sucesso com a UI final
 - Concentração na interação e não nas cores
- Próximos passos: CSS (Cascading Style Sheets)
 - Melhorar a aparência depois de funcionando
 - Bootstrap



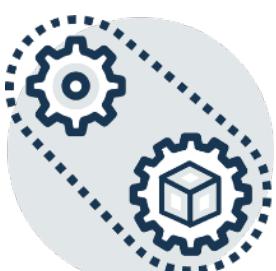
Trabalhando com o cliente: Prototipagem BDD e Lo-Fi

- “Lo-fi e storyboards são realmente úteis no trabalho com o cliente”
- “O feedback frequente do cliente é essencial”
- "O que pensávamos que seria legal não é o que o cliente se importa"
- "Fizemos protótipos de alta fidelidade e investimos muito tempo apenas para perceber que o cliente não gostava"
- “Nunca percebi o quanto difícil é passar da descrição do cliente ao plano técnico”



Pergunta

- O que é **FALSO** sobre Lo-Fi UI?
 - A. Assim como cartões 3x5, sketches & storyboards são mais propensos a envolver todos os stakeholders vs. código
 - B. O propósito da abordagem Lo-Fi UI é depurar a UI antes de você programá-la
 - C. Apps SaaS geralmente têm UIs associadas às suas histórias do usuário
 - D. Enquanto leva mais tempo do que a construção de um protótipo de UI em CSS e JS, a abordagem Lo-Fi é mais suscetível de conduzir a uma interface de usuário que os clientes gostam



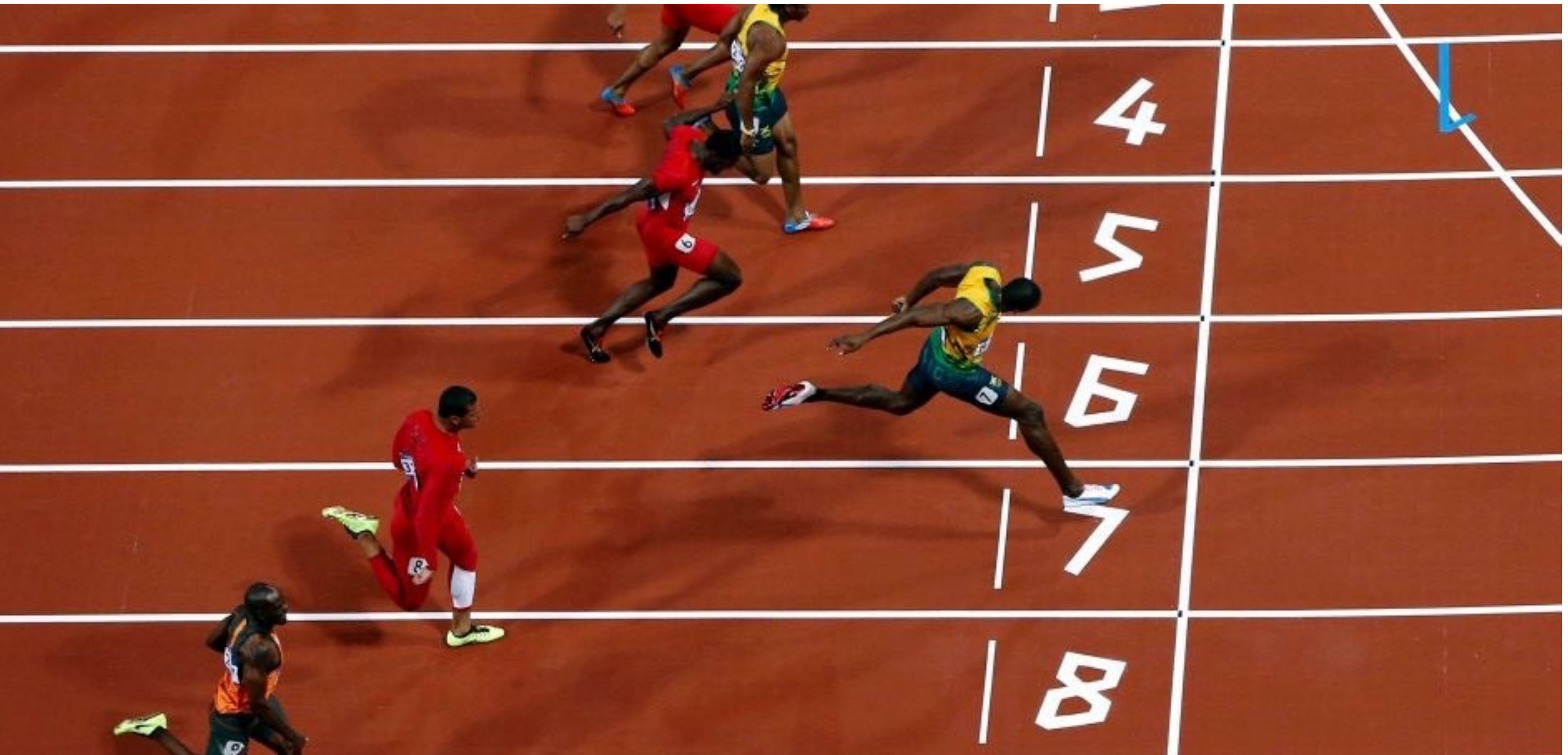
Pergunta

- O que é **FALSO** sobre Lo-Fi UI?
 - A. Assim como cartões 3x5, sketches & storyboards são mais propensos a envolver todos os stakeholders vs. código
 - B. O propósito da abordagem Lo-Fi UI é depurar a UI antes de você programá-la
 - C. Apps SaaS geralmente têm UIs associadas às suas histórias do usuário
 - D. Enquanto leva mais tempo do que a construção de um protótipo de UI em CSS e JS, a abordagem Lo-Fi é mais suscetível de conduzir a uma interface de usuário que os clientes gostam

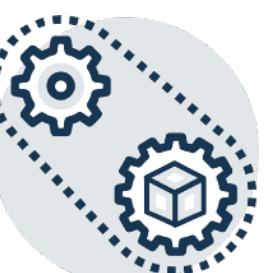
Histórias vs Camadas

- "Dividir o trabalho por histórias ajuda todos os membros da equipe a entender a aplicação e a ter mais confiança ao alterá-la"
- "O Tracker nos ajudou a priorizar recursos e estimar a dificuldade"
- "Dividimos por camadas [front-end x back-end x JavaScript, etc.] e era difícil coordenar a obtenção de recursos para o trabalho"
- "Era difícil estimar se o trabalho foi dividido de maneira justa ... não tenho certeza se nossa capacidade de estimar dificuldades melhorou com o tempo ou não"





Points and Velocity



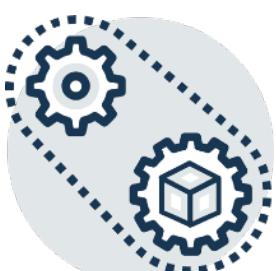
Produtividade e Estimativa de custos

- Queremos evitar grande esforço de planejamento nas Metodologias Ágeis? Se sim, como estimar o tempo sem um plano?
- Histórias do Usuário podem ser usadas para medir o progresso no projeto?
- O que uma ferramenta deve fazer para ajudar a medição de progresso para Metodologias Ágeis?



Estudo de caso: Como a Pivotal estima os custos para seus clientes?

- **Não:** comprometa-se a entregar recursos **no dia D**
- **Sim:** trabalharemos com você da maneira mais eficiente possível, **até o dia D**
 - 1 hora de telefonema para explicar o método
 - Reunião de escopo de 1,5 horas com o cliente e 2 engenheiros fundamentais
 - Papel técnico essencial: identificar incertezas
 - Engenheiros levam meia hora para estimar semanas
 - Pouco versus grande incerteza: 20-22 vs. 18-26 semanas
 - Custo da oferta como tempo e material para o cliente





Medindo Produtividade

- Uma medida de produtividade da equipe: calcular a média de histórias/semana?
 - Mas algumas histórias são muito mais difíceis do que outras
- Avalie cada história de usuário com antecedência em uma escala numérica simples
 - 1 para histórias simples, 2 para histórias médias, 3 para histórias muito complexas
- **Velocidade:** número médio de pontos/semana
- "Quanto custa 1 ponto?"
 - importante: **todos concordam na resposta**



Mais sobre Pontos

- Depois de ganhar a experiência, a escala Fibonnaci é uma boa sugestão: 1, 2, 3, 5, 8
 - (Cada novo número é a soma dos dois anteriores)
- Equipes atribuem valor: voto, mantendo-se os dedos simultaneamente, tirar média
 - Se uma grande discordância aparece (2 e 5), tem que discutir mais sobre a história

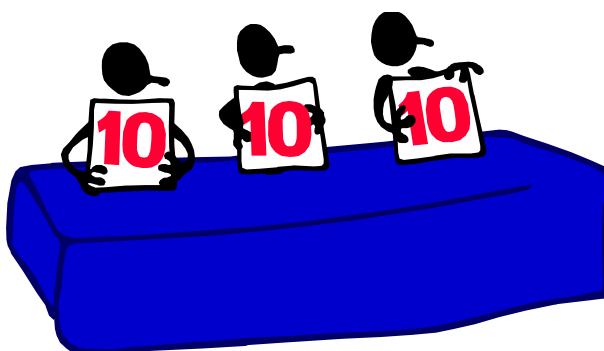


Reunião de Estimativa de Histórias

- Stakeholders envolvidos; sem laptops (exceto para ver histórias no Tracker)
- O PO prioriza histórias, aborda uma história por vez
- Todos votam **de forma independente** na dificuldade (*Planning Poker*)
 - Os votos mais alto e mais baixo explicam seu raciocínio
 - Repita até convergir ou faça um voto alto (**não médio**)
- Ambiente inclusivo: **todos fazem parte da mesma equipe**
 - alguns membros da equipe podem ter mais experiência com certos recursos
 - mas todos devem sempre se sentir seguros discordando da maioria, especialmente se errarem do lado conservador



Mais sobre Pontos



- $\geq 5 \Rightarrow$ divide a história em histórias mais simples
 - backlog não muito exigente
- Não importa se a velocidade é 5 ou 10 pontos por iteração
 - Desde que seja consistente
 - A ideia é melhorar a auto-avaliação e sugerir um número de iterações para um dado conjunto de features



Problemas Relacionados (Related Issue)

- **Spike**
 - Investigação corriqueira em uma técnica ou problema
 - e.g. spike on recommendation algorithms
 - Após a spike concluída, o código deve ser **descartado**
 - "Agora que sei o que utilizar, vou fazer certo!"



Rastreabilidade



- Priorizar histórias de usuários pela localização: Current, Backlog, Icebox
- Quando concluído, move para o painel Done
- Pode adicionar pontos de release lógicos, assim podemos descobrir quando um release realmente vai acontecer
 - pontos pendentes / velocidade
- **Epic** (com o próprio painel)
 - Combine as histórias de usuário relacionadas
 - Ordenadas independente das histórias do usuário no Backlog



Rastreie papéis

- Desenvolvedores **não decidem** quando a história foi completada
 - "Clique" no botão **Finish**, que a envia ao "Product Owner" (como em uma equipe Scrum)
- PO **avalia** a história do usuário e então decide
 - **Aceitar**, que marca história de usuário como feita, ou
 - **Rejeitar**, que marca história com a necessidade de ser reiniciada pelo desenvolvedor



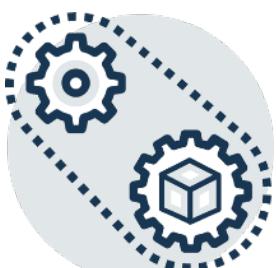
Pivotal Tracker: Features vs. Chores

- Features
 - Histórias de usuários que fornecem valor de negócio verificável para o cliente
 - "Adicionar checkbox de concordo à página"
 - Pontos valiosos e, portanto, deve ser estimado
- Chores
 - HU que são necessárias, mas não fornecem nenhum valor óbvio diretamente ao cliente
 - "Descubra porque a suíte de testes é tão lenta"
 - Não há pontos



Pergunta

- Qual afirmação em relação a Pontos, Velocidade, e Tracker é **VERDADEIRA**?
 - A. Quando comparar dois times, o com maior velocidade é mais produtivo
 - B. Quando você não sabe como abordar uma HU, basta dar 3 pontos
 - C. Com rastreabilidade, desenvolvedores pegam as HUs e marcam como Aceitas quando concluídas
 - D. Rastreabilidade ajuda a priorizar e acompanhar as HUs e seu status, calcula a velocidade e prevê o tempo de desenvolvimento de software



Pergunta

- Qual afirmação em relação a Pontos, Velocidade, e Tracker é **VERDADEIRA**?
 - A. Quando comparar dois times, o com maior velocidade é mais produtivo
 - B. Quando você não sabe como abordar uma HU, basta dar 3 pontos
 - C. Com rastreabilidade, desenvolvedores pegam as HUs e marcam como Aceitas quando concluídas
 - D. Rastreabilidade ajuda a priorizar e acompanhar as HUs e seu status, calcula a velocidade e prevê o tempo de desenvolvimento de software
- 