

Jogos para Simulação em Engenharia de Software

Eduardo Figueiredo

<http://www.dcc.ufmg.br/~figueiredo>
reuso.software@gmail.com

16 Abril 2012

Motivação

- Ensino tradicional de Engenharia de Software é composto de:
 - Aulas teóricas com exposição de conceitos
 - Pequenos trabalhos práticos para os alunos aplicarem os conceitos
- Problemas
 - Aulas têm geralmente uma única direção
 - Trabalhos não permitem explorar os problemas e ocorrem em projetos grandes

O Uso de Jogos

- Alguns jogos educacionais foram propostos para simular o processo de desenvolvimento de software
 - O objetivo principal é complementar o ensino tradicional de Eng. de Software
- Exemplos
 - *Problems and Programmers (PnP)*
 - SimulES

Problems & Programmers (PnP)

<http://www.problemsandprogrammers.com/>
(site não está mais no ar)

Visão Geral do PnP

- PnP simula um processo de software desde a fase de requisitos até a entrega do produto
 - O jogo tira o foco do produto que é geralmente explorado em TPs
 - O jogo prioriza conceitos gerenciais
- Eventos que ocorrem no jogo tem correspondência no mundo real

Exemplo

- Conceito no mundo real
 - Para descobrir se o código tem *bug*, este código deve ser inspecionado
- No jogo PnP
 - Uma carta de código produzido é mantida com a face para baixo (oculta)
 - A atividade de inspeção vira a carta com a face para cima (revela se tem ou não *bug*)

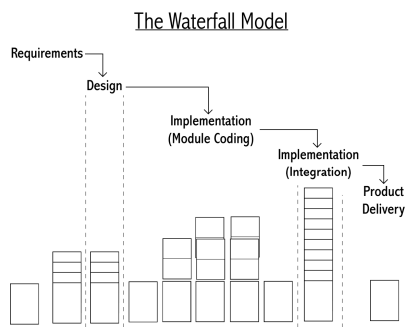
Ensino Causa-Efeito

- O jogo deixa claro as causas de sucesso e fracasso do jogador
 - Boas práticas levam ao sucesso
- Cartas de Problemas
 - Feedback negativo: indicam algo errado com o desenvolvimento
- Cartas de Conceitos
 - Feedback positivo: indicam boas práticas empregadas

Regras Elementares

- Jogadores assumem o papel de gerentes de projetos
- Todos têm que implementar o mesmo projeto no menor tempo possível
 - Quem completar o projeto primeiro ganha o jogo
- Jogadores devem seguir o Modelo Cascata

Disposição das Cartas



Fases do jogo PnP

- O jogo é dividido em seis fases
 1. Definir configuração inicial
 2. Fase de Requisitos
 3. Fase de Projeto
 4. Fase de Implementação
 5. Fase de Integração
 6. Entrega do Produto

Fases do jogo PnP

- O jogo é dividido em seis fases
 1. **Definir configuração inicial**
 2. Fase de Requisitos
 3. Fase de Projeto
 4. Fase de Implementação
 5. Fase de Integração
 6. Entrega do Produto

Fase 1: Configuração Inicial

- Uma carta de projeto é sorteada
 - Existem vários projetos com atributos diferentes
- Um projeto define quatro atributos
 - Tamanho (*length*)
 - Complexidade (*complexity*)
 - Qualidade (*quality*)
 - Orçamento (*budget*)

Exemplo de Carta de Projeto

Payroll Controller	
	\$\$\$\$\$\$
	\$\$\$\$\$\$
	\$\$\$\$\$\$
	\$\$\$\$\$\$
	\$\$\$\$\$\$

Complexity **4** Length **8**
Budget 220k Quality **8**

Atributos de um projeto

- **Tamanho** define quanto de código é necessário para completar o projeto
- **Complexidade** determina quanto de habilidade os programadores precisam para gerar um unidade de código
- **Qualidade** diz quanto do código precisa ser verificado ao final do projeto
- **Orçamento** limita a quantidade de programadores e cartas de conceitos

Início do Jogo

- Após o projeto ser sorteado, cada jogador pega 5 cartas no monte principal
- Existem três montes de cartas
 - **Monte principal** possui cartas de problemas, conceitos e programadores
 - **Monte de documentação** possui cartas de requisitos e de projeto
 - **Monte de implementação** possui cartas de código

Tipos de Cartas

- **Cartas de Conceito** representam boas decisões de desenvolvimento
- **Cartas de Programadores** representam os membros da equipe
- **Cartas de Problemas** definem uma penalidade quando boas práticas não são adotadas de jogo

Exemplos de Cartas

<p>Concept - Walkthrough</p> <div style="border: 1px solid black; height: 100px; width: 100%;"></div> <p>You may discard this card to replace an unclear requirements and/or an unclear design with new cards from the documentation deck.</p>	<p>Programmer - Arnold</p> <div style="text-align: center;"> </div> <p>Good experience, but is aloof and could be difficult to work with.</p> <p>Salary: 90k</p> <p>Skill 5 Personality 2</p>
--	--

Cartas de Problemas

- Cartas de problemas são jogadas para os adversários
- Se o jogador receber uma carta e satisfizer sua condição, ele será penalizado conforme a consequência

Misinformed Design (UR2)
Player with unclear requirements > 1
Discard 2 of this players code cards and one of their design cards.

Estrutura das Rodadas

- Na sua vez em cada rodada, o jogador deve seguir os seguintes passos
 1. Decidir se vai passar para a próxima fase do ciclo de desenvolvimento
 2. Comprar cartas
 3. **Fazer ações que são permitidas para a fase que ele se encontra**
 4. Empregar conceitos e/ou programadores
 5. Descartar cartas desnecessárias

Fases do jogo PnP

- O jogo é dividido em seis fases
 1. Definir configuração inicial
 2. **Fase de Requisitos**
 3. Fase de Projeto
 4. Fase de Implementação
 5. Fase de Integração
 6. Entrega do Produto



Fase de Requisitos

- Em sua vez, o jogador na fase de requisitos pode comprar duas cartas do monte de documentação
- Jogadores são encorajados a especificar os requisitos antes de moverem para as fases seguintes
 - Nenhum jogador é obrigado a trabalhar os requisitos

Requisitos Claros e Não Claros

- A maioria das cartas de documentação são brancas
 - Representam requisitos claros
- Podem haver cartas de documentação "Unclear"
 - Representam requisitos não claros
- Para transformar um requisito não claro em requisito claro, é necessário gastar uma das duas opções de compra

Fases do jogo PnP

- O jogo é dividido em seis fases
 1. Definir configuração inicial
 2. Fase de Requisitos
 3. **Fase de Projeto**
 4. Fase de Implementação
 5. Fase de Integração
 6. Entrega do Produto

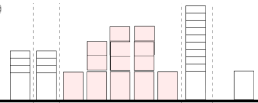


Fase de Projeto

- É muito semelhante a fase de requisitos
 - Compra-se duas cartas no mesmo monte de documentação
 - Pode haver documentação de projeto não claro (os mesmo procedimentos são adotados)
- As cartas são colocadas na coluna de projeto

[Fases do jogo PnP]

- O jogo é dividido em seis fases
 1. Definir configuração inicial
 2. Fase de Requisitos
 3. Fase de Projeto
 4. **Fase de Implementação**
 5. Fase de Integração
 6. Entrega do Produto



[Fase de Implementação]

- Nesta fase, os programadores fazem ações referentes aos seus respectivos valores de habilidades
- As ações possíveis são
 1. Produzir código bem estruturado
 2. Produzir código "espaguete"
 3. Inspecionar código
 4. Resolver bugs

[Ação de Codificar]

- Para produzir uma carta de **código estruturado**, o programador gasta tempo equivalente a complexidade do projeto
- Para produzir uma carta de **código espaguete**, o programador gasta metade da complexidade do projeto

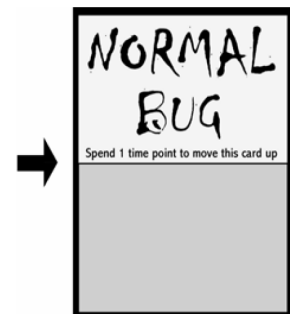
[Código por Programador]

- Cada carta de código produzido é colocada em uma coluna acima do programador que a produziu
 - Inicialmente, a carta é colocada com a face para baixo
- A face da carta para baixo significa que o código não foi inspecionado
 - Possíveis bugs no código estão ocultos

[Código Não Inspecionado]



[Código Inspecionado]



Código Estruturado x Espaguete

- A mesma carta é usada para código estruturado ou espaguete
- A carta possui duas metades
 - Metade escura (azul) para cima representa código estruturado
 - Metade clara (vermelho) para cima representa código espaguete
- Código espaguete revela bugs mais frequentemente que código estruturado

Ação de Inspeccionar

- Inspeccionar uma carta de código requer uma unidade de tempo
 - Unidade de tempo equivale a uma unidade de habilidade do programador
- Após inspeccionar uma carta de código, o programador pode virar a face da carta para cima
 - Revela a existência ou não de *bug*

Ação de Resolver *Bug*

- Os programadores podem usar um ponto de tempo tentar remover um *bug*
 - Entretanto, o *bug* pode não ser removido completamente da primeira vez
- Para remover um *bug* pode ser necessário várias unidades de tempo
 - Apenas *bugs* simples (*Simple*) podem ser trocados diretamente por uma nova carta de código usando um ponto de tempo

Normal e Nasty *Bug*

- Cada unidade de tempo gasto pelo programador move o bug uma posição para cima na coluna
 - Troca a carta com *bug* pela carta imediatamente acima na coluna do programador
- Quando a carta com *bug* atingir o topo
 - Com uma unidade de tempo, a carta com *bug* pode ser trocada por uma nova carta

Fases do jogo PnP

- O jogo é dividido em seis fases
 1. Definir configuração inicial
 2. Fase de Requisitos
 3. Fase de Projeto
 4. Fase de Implementação
 5. **Fase de Integração**
 6. Entrega do Produto



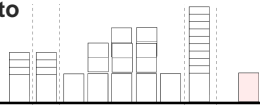
Fase de Integração

- Em cada rodada de integração, o código de um programador pode ser movido para a parte de código integrado
- O projeto termina quando for integrada quantidade de código equivalente ao tamanho do projeto
 - O jogador pode integrar código que não foi inspecionado (ou mesmo com *bug*)

[Fases do jogo PnP]

- O jogo é dividido em seis fases

1. Definir configuração inicial
2. Fase de Requisitos
3. Fase de Projeto
4. Fase de Implementação
5. Fase de Integração
6. **Entrega do Produto**



[Fase de Entrega do Produto]

- Durante a entrega do produto, é feita a verificação e validação com o cliente
- O cliente verifica quantidade de cartas de código equivalente a qualidade do projeto
- Se o cliente descobrir algum *bug* nessa fase, o jogador será penalizado
 - Podendo até mesmo ser desclassificado

[Penalidades por Bug na Entrega]

- Se o cliente encontrar apenas *bugs* simples (*Simple*) ou normais (*Normal*)
 - O jogador terá que criar e integrar novamente quantidade equivalente de cartas para substituir o código com *bug*
- Se o cliente encontrar algum *bug* grave (*Nasty*)
 - O jogador será desqualificado e o jogo continua com os demais jogadores

[Término do Jogo]

- Se nenhuma das cartas reveladas pelo cliente possuírem *bugs*, o jogador ganha!

SimuLES

<http://www.dcc.ufmg.br/~figueiredo/simules/>

[O Jogo SimuLES]

- Simulação de Engenharia de Software
- O jogo foi fortemente inspirado no PnP
- Para construção de SimuLES
 - O jogo PnP foi traduzido para o português
 - Alguns problemas em PnP foram identificados e corrigidos
 - Novas cartas foram criadas

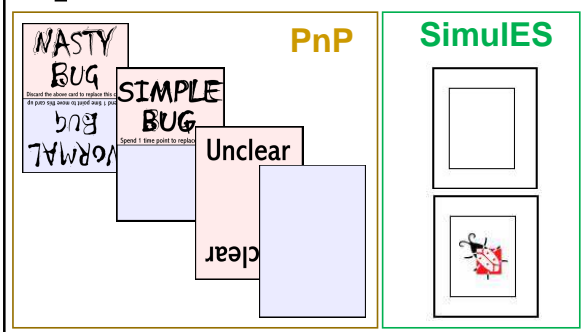
Problemas com o PnP

- Só considera o Modelo Cascata
- Pouca fonte de informação
- Sem entusiasmo nas primeiras rodadas
- Não limita o número de jogadores
- Sem organização das cartas na mesa
- Não é claro os conceitos de Engenharia de Software explorado

Cascata e Fonte de Informação

- PnP é muito amarrado ao Modelo Cascata
 - Em SimulES, as regras foram flexibilizadas para permitir a escolha de outros modelos
 - O Engenheiro de Software pode produzir tanto requisitos quanto código na mesma rodada
- Cartas abstratas e com pouca informação
 - Mais informações e identificadores foram adicionadas
 - Exemplos: carta de projeto

Artefatos: PnP x SimulES



Projeto PnP X SimulES

Payroll Controller <table border="1"> <tr><td>...</td><td>\$\$\$500</td></tr> <tr><td>...</td><td>\$\$\$500</td></tr> <tr><td>...</td><td>\$\$\$500</td></tr> <tr><td>...</td><td>\$\$\$500</td></tr> <tr><td>...</td><td>\$\$\$500</td></tr> </table> <p>Complexity 4 Length 8 Quality 7 Budget 220k</p>	...	\$\$\$500	...	\$\$\$500	...	\$\$\$500	...	\$\$\$500	...	\$\$\$500	Projeto PR 1 Expert Committee Expert Committee é um sistema multi-agente aberto para suporte ao gerenciamento de submissões e revisões de artigos submetidos a uma conferência ou workshop. O sistema oferece suporte a diferentes atividades, tais como, envio de trabalhos, atribuição de um artigo a um revisor, seleção de revisores, notificação da aceitação e recusa de artigos. [Garcia et al, 2004] <table border="1"> <tr> <td>Complexidade: 4</td> <td>Módulos</td> </tr> <tr> <td>Tamanho: 5</td> <td>1 2 RQ + 1 DS + 1 CD</td> </tr> <tr> <td>Qualidade: 2</td> <td>2 1 RQ + 1 RT + 1 CD</td> </tr> <tr> <td>Orçamento: 220 K</td> <td>3 1 DS + 1 RT + 1 CD</td> </tr> <tr> <td></td> <td>4 2 DS + 1 RT + 1 CD</td> </tr> <tr> <td></td> <td>5 1 DS + 2 AJ + 1 CD</td> </tr> <tr> <td></td> <td>6</td> </tr> </table>	Complexidade: 4	Módulos	Tamanho: 5	1 2 RQ + 1 DS + 1 CD	Qualidade: 2	2 1 RQ + 1 RT + 1 CD	Orçamento: 220 K	3 1 DS + 1 RT + 1 CD		4 2 DS + 1 RT + 1 CD		5 1 DS + 2 AJ + 1 CD		6
...	\$\$\$500																								
...	\$\$\$500																								
...	\$\$\$500																								
...	\$\$\$500																								
...	\$\$\$500																								
Complexidade: 4	Módulos																								
Tamanho: 5	1 2 RQ + 1 DS + 1 CD																								
Qualidade: 2	2 1 RQ + 1 RT + 1 CD																								
Orçamento: 220 K	3 1 DS + 1 RT + 1 CD																								
	4 2 DS + 1 RT + 1 CD																								
	5 1 DS + 2 AJ + 1 CD																								
	6																								

Projeto PnP X SimulES

Payroll Controller <table border="1"> <tr><td>...</td><td>\$\$\$500</td></tr> <tr><td>...</td><td>\$\$\$500</td></tr> <tr><td>...</td><td>\$\$\$500</td></tr> <tr><td>...</td><td>\$\$\$500</td></tr> <tr><td>...</td><td>\$\$\$500</td></tr> </table> <p>Complexity 4 Length 8 Quality 7 Budget 220k</p>	...	\$\$\$500	...	\$\$\$500	...	\$\$\$500	...	\$\$\$500	...	\$\$\$500	Projeto PR 1 Expert Committee Expert Committee é um sistema multi-agente aberto para suporte ao gerenciamento de submissões e revisões de artigos submetidos a uma conferência ou workshop. O sistema oferece suporte a diferentes atividades, tais como, envio de trabalhos, atribuição de um artigo a um revisor, seleção de revisores, notificação da aceitação e recusa de artigos. [Garcia et al, 2004] <table border="1"> <tr> <td>Complexidade: 4</td> <td>Módulos</td> </tr> <tr> <td>Tamanho: 5</td> <td>1 2 RQ + 1 DS + 1 CD</td> </tr> <tr> <td>Qualidade: 2</td> <td>2 1 RQ + 1 RT + 1 CD</td> </tr> <tr> <td>Orçamento: 220 K</td> <td>3 1 DS + 1 RT + 1 CD</td> </tr> <tr> <td></td> <td>4 2 DS + 1 RT + 1 CD</td> </tr> <tr> <td></td> <td>5 1 DS + 2 AJ + 1 CD</td> </tr> <tr> <td></td> <td>6</td> </tr> </table>	Complexidade: 4	Módulos	Tamanho: 5	1 2 RQ + 1 DS + 1 CD	Qualidade: 2	2 1 RQ + 1 RT + 1 CD	Orçamento: 220 K	3 1 DS + 1 RT + 1 CD		4 2 DS + 1 RT + 1 CD		5 1 DS + 2 AJ + 1 CD		6
...	\$\$\$500																								
...	\$\$\$500																								
...	\$\$\$500																								
...	\$\$\$500																								
...	\$\$\$500																								
Complexidade: 4	Módulos																								
Tamanho: 5	1 2 RQ + 1 DS + 1 CD																								
Qualidade: 2	2 1 RQ + 1 RT + 1 CD																								
Orçamento: 220 K	3 1 DS + 1 RT + 1 CD																								
	4 2 DS + 1 RT + 1 CD																								
	5 1 DS + 2 AJ + 1 CD																								
	6																								

Identificador da carta

Projeto PnP X SimulES

Payroll Controller <table border="1"> <tr><td>...</td><td>\$\$\$500</td></tr> <tr><td>...</td><td>\$\$\$500</td></tr> <tr><td>...</td><td>\$\$\$500</td></tr> <tr><td>...</td><td>\$\$\$500</td></tr> <tr><td>...</td><td>\$\$\$500</td></tr> </table> <p>Complexity 4 Length 8 Quality 7 Budget 220k</p>	...	\$\$\$500	...	\$\$\$500	...	\$\$\$500	...	\$\$\$500	...	\$\$\$500	Projeto PR 1 Expert Committee Expert Committee é um sistema multi-agente aberto para suporte ao gerenciamento de submissões e revisões de artigos submetidos a uma conferência ou workshop. O sistema oferece suporte a diferentes atividades, tais como, envio de trabalhos, atribuição de um artigo a um revisor, seleção de revisores, notificação da aceitação e recusa de artigos. [Garcia et al, 2004] <table border="1"> <tr> <td>Complexidade: 4</td> <td>Módulos</td> </tr> <tr> <td>Tamanho: 5</td> <td>1 2 RQ + 1 DS + 1 CD</td> </tr> <tr> <td>Qualidade: 2</td> <td>2 1 RQ + 1 RT + 1 CD</td> </tr> <tr> <td>Orçamento: 220 K</td> <td>3 1 DS + 1 RT + 1 CD</td> </tr> <tr> <td></td> <td>4 2 DS + 1 RT + 1 CD</td> </tr> <tr> <td></td> <td>5 1 DS + 2 AJ + 1 CD</td> </tr> <tr> <td></td> <td>6</td> </tr> </table>	Complexidade: 4	Módulos	Tamanho: 5	1 2 RQ + 1 DS + 1 CD	Qualidade: 2	2 1 RQ + 1 RT + 1 CD	Orçamento: 220 K	3 1 DS + 1 RT + 1 CD		4 2 DS + 1 RT + 1 CD		5 1 DS + 2 AJ + 1 CD		6
...	\$\$\$500																								
...	\$\$\$500																								
...	\$\$\$500																								
...	\$\$\$500																								
...	\$\$\$500																								
Complexidade: 4	Módulos																								
Tamanho: 5	1 2 RQ + 1 DS + 1 CD																								
Qualidade: 2	2 1 RQ + 1 RT + 1 CD																								
Orçamento: 220 K	3 1 DS + 1 RT + 1 CD																								
	4 2 DS + 1 RT + 1 CD																								
	5 1 DS + 2 AJ + 1 CD																								
	6																								

Descrição do projeto

Projeto PnP X Simules

Payroll Controller

Nome	SSSSSS
Idade	SSSSSS
Sexo	SSSSSS
Profissão	SSSSSS
Religião	SSSSSS
Estado	SSSSSS

Complexity **4**

Length **8**

Quality **7**

Budget **220k**

Projeto PR 1

Expert Committee

Expert Committee é um sistema multi-agente aberto para suporte ao gerenciamento de submissões e revisões de artigos submetidos a uma conferência ou workshop. O sistema oferece suporte a diferentes atividades, tais como, envio de trabalhos, atribuição de um artigo a um revisor, seleção de revisores, notificação de aceite e recusa de artigos.

[Garcia et al, 2004]

Complexidade: 4	Módulos
Tamanho: 5	1 2 RQ + 1 DS + 1 CD
Qualidade: 2	2 1 RQ + 1 RT + 1 CD
Orçamento: 220 K	3 1 DS + 1 RT + 1 CD
	4 2 DS + 1 RT + 1 CD
	5 1 DS + 2 AJ + 1 CD
	6

Referências adicionais (consultar catálogo)

Projeto PnP X Simules

Payroll Controller

Nome	SSSSSS
Idade	SSSSSS
Sexo	SSSSSS
Profissão	SSSSSS
Religião	SSSSSS
Estado	SSSSSS

Complexity **4**

Length **8**

Quality **7**

Budget **220k**

Projeto PR 1

Expert Committee

Expert Committee é um sistema multi-agente aberto para suporte ao gerenciamento de submissões e revisões de artigos submetidos a uma conferência ou workshop. O sistema oferece suporte a diferentes atividades, tais como, envio de trabalhos, atribuição de um artigo a um revisor, seleção de revisores, notificação de aceite e recusa de artigos.

[Garcia et al, 2004]

Complexidade: 4	Módulos
Tamanho: 5	1 2 RQ + 1 DS + 1 CD
Qualidade: 2	2 1 RQ + 1 RT + 1 CD
Orçamento: 220 K	3 1 DS + 1 RT + 1 CD
	4 2 DS + 1 RT + 1 CD
	5 1 DS + 2 AJ + 1 CD
	6

Um projeto é composto por diversos tipos de artefatos

Tabuleiro e Referências

- Ausência de tabuleiro para organização da área do jogador
 - O tabuleiro foi criado em Simules
- Ausência de mapeamento claro entre o jogo e conceitos de Eng. de Software
 - Simules inclui referências nas cartas para artigos ou livros que explicam o conceito
 - Uma lista de bibliografia foi anexado ao jogo

Tabuleiro do Simules

	Engenheiro ES1 Janaina	Engenheiro ES21 Carlos	Engenheiros de Software		
	Salário: 40 K	Salário: 70 K	Engenheiro 3	Engenheiro 4	Engenheiro 5
Requisitos	<input type="checkbox"/>	<input type="checkbox"/>			
Desenhos	<input type="checkbox"/>	<input type="checkbox"/>			
Códigos	<input type="checkbox"/>	<input type="checkbox"/>			
Rastros	<input type="checkbox"/>	<input type="checkbox"/>			
Ajudas	<input type="checkbox"/>	<input type="checkbox"/>			

Simulação do Mundo Real

- Eventos que ocorrem no jogo tem correspondência no mundo real
- Exemplo de conceito no mundo real
 - Um projeto de software complexo requer vários tipos de artefatos
- No jogo Simules
 - Cada projeto tem uma configuração de artefatos que deve ser entregue ao cliente

Fases do jogo Simules

- O jogo é dividido em quatro fases
 1. Definir configuração inicial
 2. Desenvolvimento
 - Requisitos, projeto, implementação, etc.
 3. Fase de Integração
 4. Entrega do Produto

1 - Fase Inicial

- Uma carta de projeto é sorteada
 - Existem vários projetos com atributos diferentes
- Como em PnP, um projeto define quatro atributos
 - Tamanho
 - Complexidade
 - Qualidade
 - Orçamento

Artefatos do Módulo

Projeto		PR 1
Expert Committee		
Expert Committee é um sistema multi-agente aberto para suporte ao gerenciamento de submissões e revisões de artigos submetidos a uma conferência ou workshop. O sistema oferece suporte a diferentes atividades, tais como, envio de trabalhos, atribuição de um artigo a um revisor, seleção de revisores, notificação da aceitação e recusa de artigos.		
[Garcia et al, 2004]		
Complexidade	4	Módulos
Tamanho	5	1 2 RQ + 1 DS + 1 CD
Qualidade	2	2 1 RQ + 1 RT + 1 CD
Orçamento	220 K	3 1 DS + 1 RT + 1 CD
		4 2 DS + 1 RT + 1 CD
		5 1 DS + 2 AJ + 1 CD
		6

O tamanho indica o número de módulos

Cada módulo é composto por diferentes tipos de artefatos

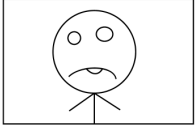

Início do Jogo

- Em SimulES, existem dois montes de cartas
 - Monte principal** possui cartas de problemas, conceitos e programadores
 - Monte de artefatos** possui cartas de artefatos
- Artefatos podem ser requisitos, projeto, implementação, ajuda e rastros

Programadores x Engenheiros

- Não existem programadores em SimulES, mas Engenheiros de Software
- Engenheiros de Software representam os membros da equipe
 - Eles podem criar requisitos, projeto, código, ajuda e rastros
 - Eles também podem inspecionar ou corrigir bugs (problemas) em qualquer tipo de artefato

Programador X Engenheiro

PnP	SimulES
<p>Programmer - Stevie</p>  <p>Your boss' son. Find a way to let him help out.</p> <p>Salary: Free</p> <p>Skill 1 Personality 1</p>	<p>Eng. de Software ES9 Estevão</p>  <p>O filho do chefe. Procure deixá-lo ajudar.</p> <p>Salário: 0 K</p> <p>Habilidade 1 Maturidade 1</p>

Estrutura das Rodadas

- Em cada rodada, o jogador deve seguir os seguintes passos
 - Jogar um dado
 - Comprar o número de cartas do monte principal equivalente ao valor do dado (limite máximo de 5 cartas na mão)
 - Fazer ações permitidas para a fase
 - Empregar conceitos e/ou programadores
 - Descartar cartas desnecessárias

Ações: Fase de Desenvolvimento

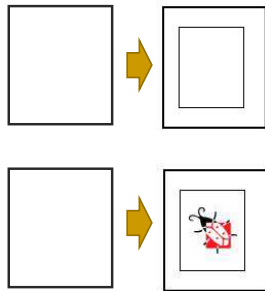
- Criar artefatos
 - Semelhante a criar código no PnP
 - Cada engenheiro cria artefatos de acordo com sua habilidade
- Inspeccionar código
 - Da mesma forma que em PnP
- Resolver bug
 - Da mesma forma que "Simple Bug" em PnP

Artefatos por Engenheiro

- Cada carta de artefato produzido é colocada na coluna abaixo do engenheiro que a produziu
 - Existem repartições específicas para cada tipo de artefato no tabuleiro
- Como em PnP, a face da carta para baixo significa que o artefato não foi inspecionado
 - Possíveis bugs no código estão ocultos

Inspeccionar Artefatos

- Artefatos inspecionados podem conter ou não bugs



Remover bugs

- Semelhante a um *Simple Bug* em PnP
 - Todo bug pode ser removido usando um ponto de habilidade do Engenheiro
 - Quando um bug é removido, a carta com bug é trocada por outra carta de artefato
- A nova carta de artefato é mantida com a face virada para baixo
 - Ou seja, depois de remover um bug, é preciso inspecionar novamente para garantir que não há outros bugs

Fase de Integração

- Em cada rodada, pode ser integrado um único módulo
 - Cada módulo é feito por um engenheiro
- Os módulos integrados devem seguir ao especificado no projeto
 - Não é necessário integrar os módulos na ordem especificada pelo projeto

Verificação e Validação

- O cliente verifica quantidade de módulos equivalente a qualidade do projeto
 - Cada módulo pode conter várias cartas
- Se o cliente descobrir algum bug em um módulo
 - O módulo com bug é descartado
 - O jogador deve criar e integrar novamente um módulo equivalente

[Término do Jogo]

- Se nenhuma das cartas reveladas pelo cliente possuírem *bugs*, o jogador ganha!

[Referências da Aula]

- Alex Baker, Emily Oh Navarro, and André van der Hoek. **An Experimental Card Game for Teaching Software Engineering Processes**. Journal of Systems of Software, 2004.
- Figueiredo, E.; Lobato, C.; Dias, K.; Leite, J. e Lucena, C. **"Um Jogo para o Ensino de Engenharia de Software Centrado na Perspectiva de Evolução"**. Workshop sobre Educação em Computação (WEI), Rio de Janeiro, pp. 37-46, 2007.

[Próxima Aula...]

- A turma será dividida em duas
 - Turma 1: aula de 9:00 as 10:30
 - Turma 2: aula de 10:00 as 11:30
 - **Sala ???**
- Iremos jogar os jogos PnP e/ou SimulES
 - Evitem atrasos para o início do jogo