

Padrões *USE Case*

Engenharia de Software II

Maria Clara Silveira

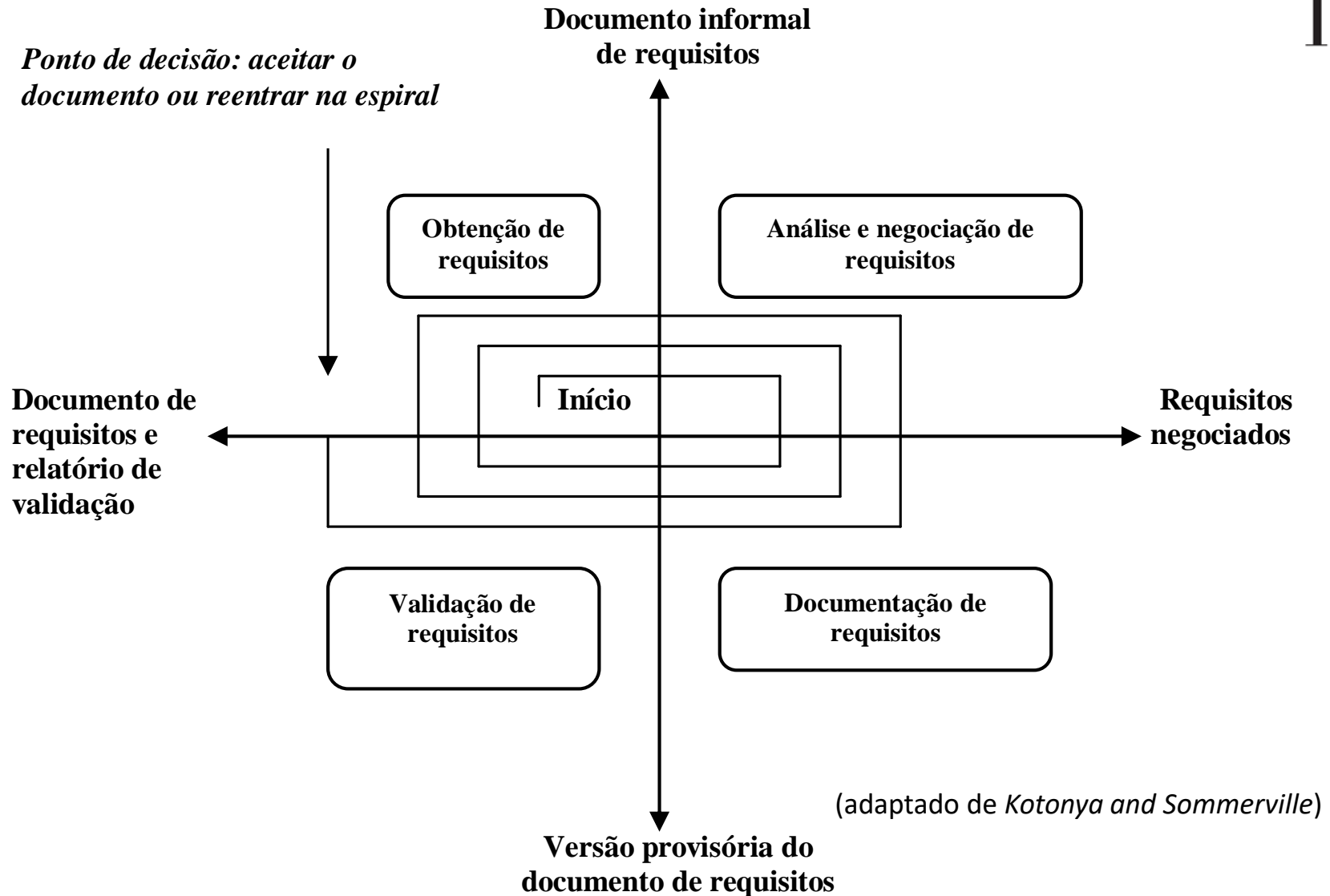
Disponível plataforma

- Livro em inglês: PatternsUseCaseBook.pdf
- Slides dos autores em inglês:
 - use_case_patterns slides autor livro.pdf
- Resumo em português dos padrões:
ResumoPadroesUseCase.pdf

Processo Engenharia Requisitos

- A engenharia de requisitos corresponde ao **processo** de descobrir, analisar ... requisitos
 - exige criatividade, interação entre pessoas, conhecimento e experiência
 - Conhecer variedade de técnicas: entrevista; análise de outras aplicações / sistemas; observação direta; *Workshop*; *Brainstorming* (e outras para explorar a potencialidade criativa); Ex-utilizadores; Estudo de caso; Filme

Processo Engenharia Requisitos

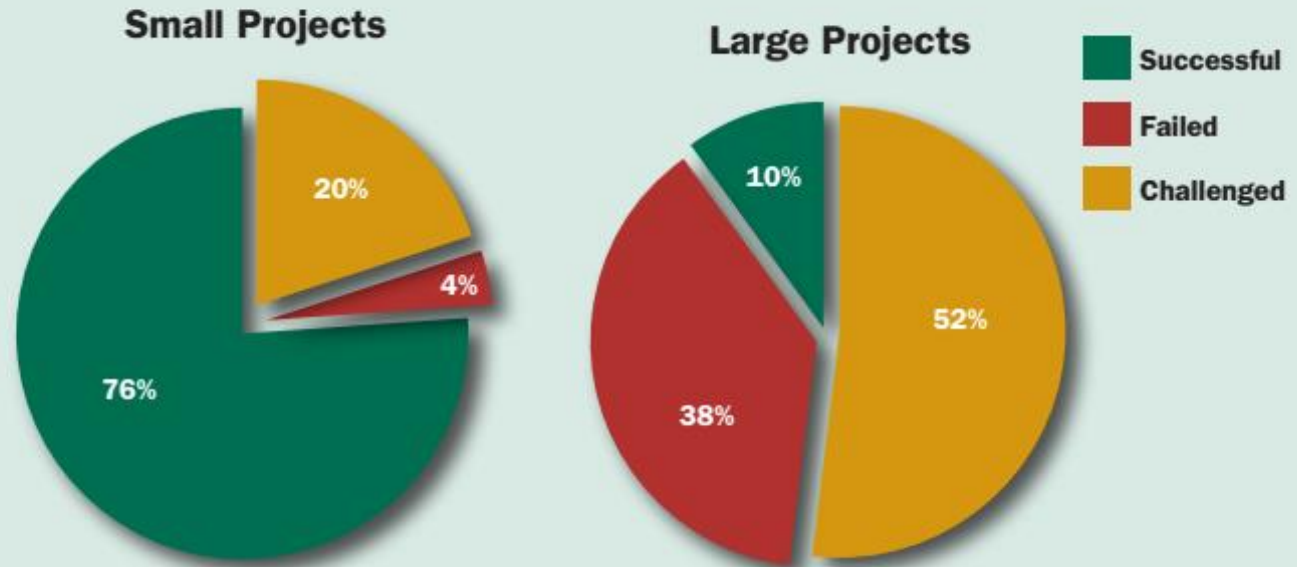


Relatório CHAOS - *Standish Group's*



CHAOS RESOLUTION BY LARGE AND SMALL PROJECTS

Project resolution for the calendar year 2012 in the new CHAOS database. Small projects are defined as projects with less than \$1 million in labor content and large projects are considered projects with more than \$10 million in labor content.



Algumas causas:

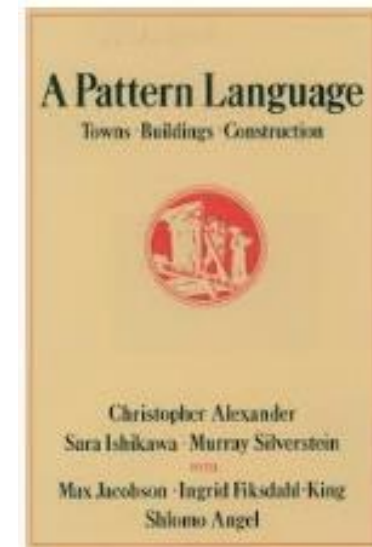
- má qualidade das entradas (inputs);
- a definição pouco clara dos objetivos;
- a má especificação e definição dos requisitos; complexidade; ...

Conceito de padrão (*pattern*)

- Movimento que teve origem no trabalho de *Cristopher Alexander*
 - “os padrões fornecem soluções genéricas para problemas recorrentes num determinado contexto”
- A ideia de padrão foi extrapolada para o software
- *Design patterns*: são soluções úteis para problemas de desenho
- ***Patterns for Effective Use Cases***: apresenta soluções para problemas relacionados com a criação e manutenção de casos de uso (use cases)

Conceito padrão (*pattern*) - Alexander

- *Cristopher Alexander* recolheu e documentou, durante 10 anos, soluções genéricas para problemas recorrentes no domínio da arquitetura
 - O livro “*A Pattern Language*” apresenta soluções para problemas recorrentes na criação de cidades, vilas, urbanizações e edifícios
 - 253 padrões que formam a linguagem
- Exemplos:
- 27 - Homens e Mulheres
 - 83 - Mestre e Aprendizes



Alexander – 27 Homens / Mulheres

- Cada edifício, espaço aberto, comunidade de trabalho, será construído com uma mistura de instintos masculinos e femininos
- Ter sempre presente este equilíbrio, entre masculino e feminino, em todos os projetos a qualquer escala

Alexander – Mestre e Aprendizizes

- Organizar o local de trabalho de forma que mestres e aprendizes possam trabalhar juntos e possam reunir
- Mestre ao centro com os aprendizes à sua volta

A situação básica de **aprendizagem** é aquela na qual uma pessoa aprende com a ajuda de alguém que realmente sabe o que está a fazer

Alexander – 139 Cozinha Rural

A cozinha isolada e separada da família é considerada como uma fábrica de alimentos eficiente mas desagradável ...

- Cozinha bastante grande, próxima das zonas comuns e que incluía *sala de estar*. Mesa grande ao centro com cadeiras de diferentes estilos,... zona luminosa e confortável.
- Esquema coloca a mesa ao centro, luz em dois lados, gabinetes próximos, ...

Alexander – 179 Gabinetes

Grupo de pessoas reunidas enquanto grupo, mas que também possam estar sozinhos!

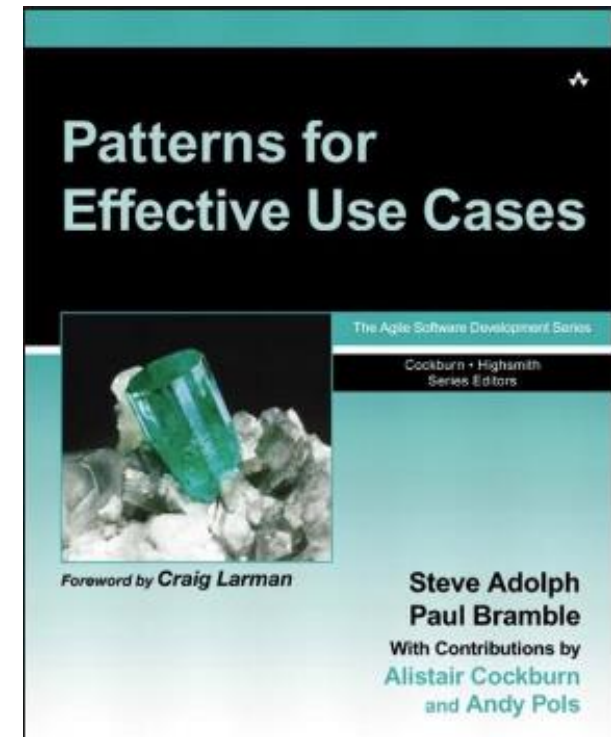
- Criar pequenos compartimentos (de 90cm a 1.80m), ligados à zona comum, onde se possam sentar duas pessoas para conversar ou jogar.
- Esquema coloca os gabinetes próximos e ligados à parte comum. Corredor de Informática ter um bar ao fundo por exemplo.

Patterns for Effective Use Cases

Patterns for Effective Use Cases: Steve Adolph e Paul Bramble

Apresenta 31 padrões em 2 categorias:

- Padrões de **desenvolvimento**
 - Equipa; Processo; Edição e revisão
- Padrões de **estrutura**
 - Conjunto; individuais;
 - Cenários e passos



Porquê padrões?

- Padrões descrevem propriedades de bons exemplos: **Sinais de qualidade** (*Patterns for Effective Use Cases* - Steve Adolph e Paul Bramble)
- Segundo *Alexander*, as agradáveis sensações que uma construção nos transmite: liberdade, conforto, harmonia e vida são tudo reflexos da presença **Qualidade sem nome**
- Os padrões e as linguagens de padrões ajudam a equipa de desenvolvimento a comunicar e reutilizar conhecimento

Porquê Casos de Uso?

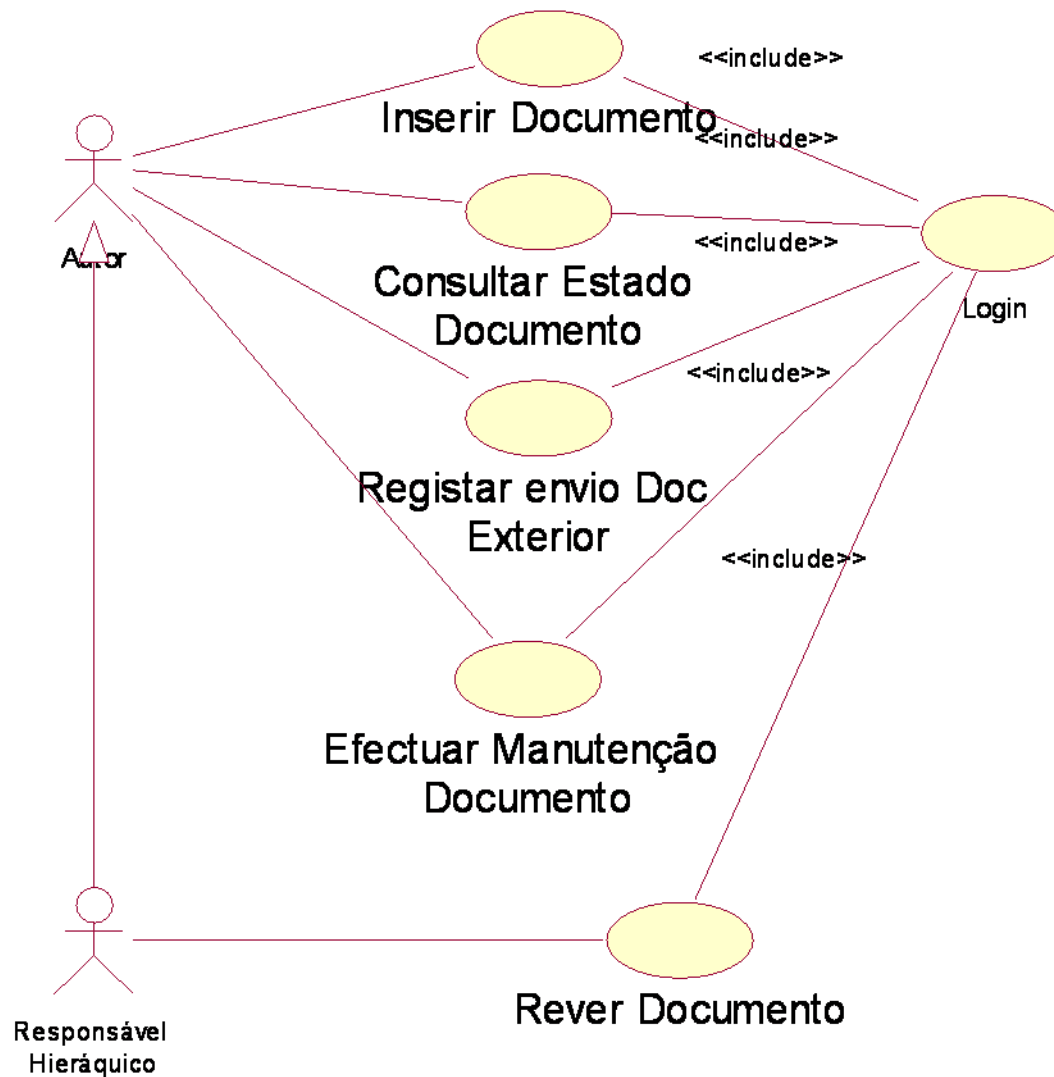
- São focados, ágeis e versáteis (como os gatos)



Use Case: agile, simple-minded, with a side view (P. Picasso)

(fonte: caminao.blog/2015/11/26/use-cats/)

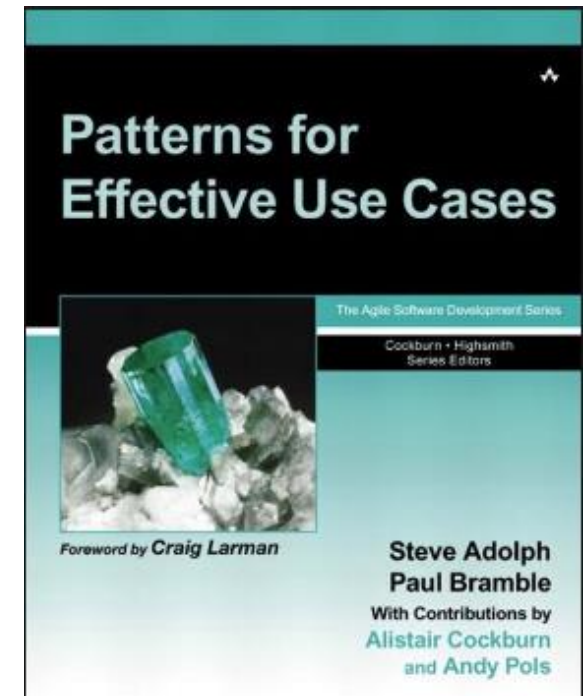
Diagrama de Casos de Uso



Patterns for Effective Use Cases

Documentar - formato Alexandrino

- Nome do padrão
- Fotografia (desenho / imagem)
- Contexto
- Problema (descrição)
- História (metafórica)
- Forças (que afectam o problema)
- Solução
- Exemplos



Equipa

Padrões de desenvolvimento: a equipa

Padrão	Problema	Solução
SmallWritingTeam	Usar muitas pessoas para escrever um caso de uso é ineficiente, e os compromissos feitos para ajustar os diferentes pontos de vista, podem resultar num sistema menos satisfatório.	Restringir o número de pessoas que trabalham num produto para duas ou três. Usar o processo do padrão TwoTierReview para incluir mais pessoas.
ParticipatingAudience	Não se pode satisfazer as necessidades dos <i>stakeholders</i> sem a sua participação e contribuição.	Sempre que possível envolver activamente clientes e <i>stakeholders</i> internos no processo de desenvolvimento de casos de uso.
BalancedTeam	Equipas com indivíduos que têm a mesma maneira de pensar na forma de desenvolver casos de uso, pode resultar num conjunto de casos de uso estritamente limitados que não satisfazem as necessidades.	Formar a equipa com pessoas de diferentes especialidades de forma a incentivar os interesses dos <i>stakeholders</i> no processo de desenvolvimento. A equipa deve ter programadores e utilizadores finais.

Equipa: *SmallWritingTeam*

- Problema
 - Muitas pessoas para escrever um caso de uso é ineficiente
- Solução
 - Restringir o número de pessoas que trabalham num produto para duas ou três. Usar o processo do padrão *TwoTierReview* para incluir mais pessoas



Equipa: *Participating Audience*

- Problema
 - Não se pode satisfazer as necessidades dos *stakeholders* sem a sua participação e contribuição
- Solução
 - Sempre que possível **envolver** ativamente clientes e *stakeholders* internos no processo de desenvolvimento de casos de uso

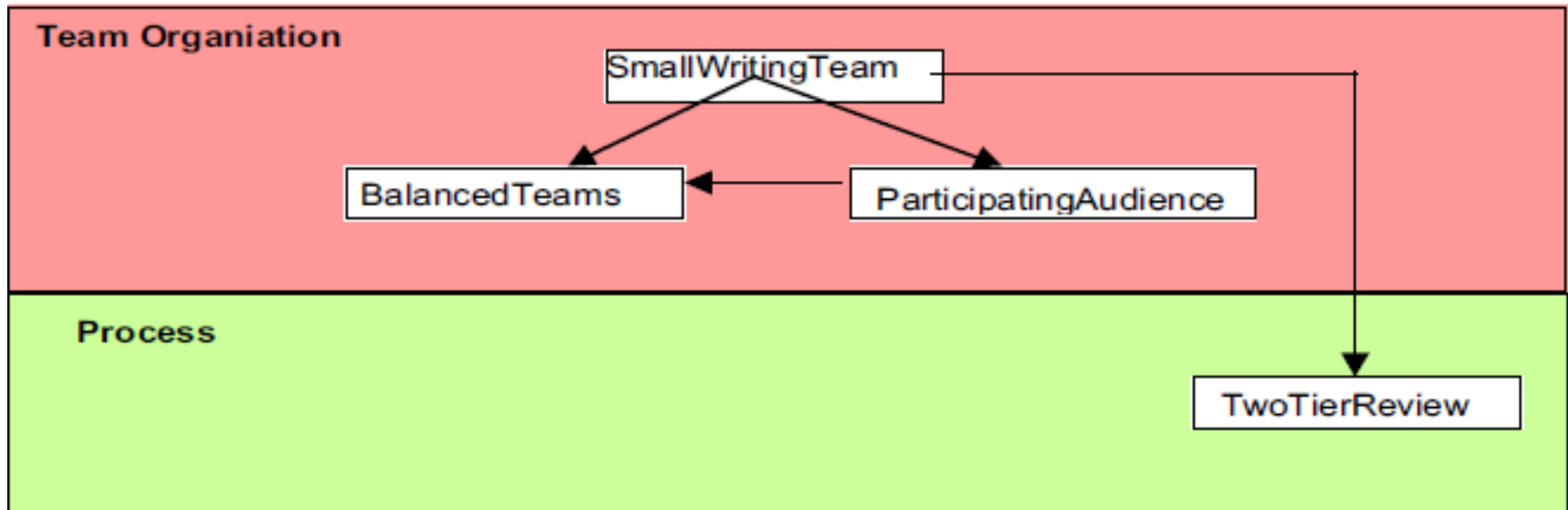


Equipa: *BalancedTeam*

- Problema
 - Equipas com indivíduos que têm a mesma maneira de pensar na forma de desenvolver casos de uso, pode resultar num conjunto de casos de uso estritamente limitados
- Solução
 - Formar a equipa com pessoas de **diferentes** especialidades de forma a **incentivar** os interesses dos *stakeholders* no processo de desenvolvimento. Exemplo: programadores e utilizadores finais



Padrões Equipa: relacionamentos



*Divida uma grande **ParticipatingAudience** em **SmallWritingTeam**; depois use o **TwoTierReview** para obter feedback.*

Padrões de Desenvolvimento: O Processo

Padrão	Problema	Solução
BreadthBeforeDepth	Não haverá progressos no tempo, nem serão criados de casos de uso coerentes se for gasta a energia a escrever casos de uso muito detalhados.	Direccionar as energias do desenvolvimento de casos de uso para uma visão global e adicionar progressivamente mais detalhes.
SpiralDevelopment	Desenvolver casos de uso num único passo é difícil, impede a incorporação de novos dados, podendo atrasar a descoberta de factores de risco.	Desenvolver os casos de uso num processo iterativo e em cada iteração aumentar progressivamente a precisão e a objectividade dos casos de uso.
MultipleForms	Projectos diferentes necessitam de diferentes formalidades durante a sua execução. Pessoas diferentes têm preferências diferentes para o desenvolvimento. Assim, é contra-productivo tentar que todas as pessoas usem o mesmo método de trabalho.	Seleccionar um formato de desenvolvimento baseado nos riscos do projecto e nas preferências das pessoas envolvidas.
TwoTierReview	Muitas pessoas podem necessitar de rever os casos de uso, tornando este processo caro e demorado.	Assegurar dois tipos de revisões: a primeira efectuada pela equipa a nível interno e repetida várias vezes. A segunda deverá ser realizada pelo grupo completo, podendo fazer só uma única revisão.
QuittingTime	Desenvolver o modelo de casos de uso para além das necessidades dos <i>stakeholders</i> , consome recursos e atrasa o projecto.	Desenvolver os casos de uso até ficarem completos e satisfazerem adequadamente as necessidades da audiência.
WritersLicence	Uma ênfase excessiva no estilo dificulta a escrita dos casos de uso.	Pequenas diferenças no estilo de escrita são inevitáveis, mas cada caso de uso deve passar nos seguintes testes: deve seguir o modelo e o estilo básico da organização; estar logicamente correcto; ser legível para os avaliadores finais; ser suficientemente preciso para ser implementado.

Processo: *BreadthBeforeDepth*

- Problema
 - Se for gasta a energia a escrever casos de uso muito detalhados, não haverá progressos
- Solução
 - Direcionar as energias do desenvolvimento de casos de uso para uma **visão global** e adicionar **progressivamente** mais **detalhes** (é um mecanismo para lidar com a complexidade)



Abstração

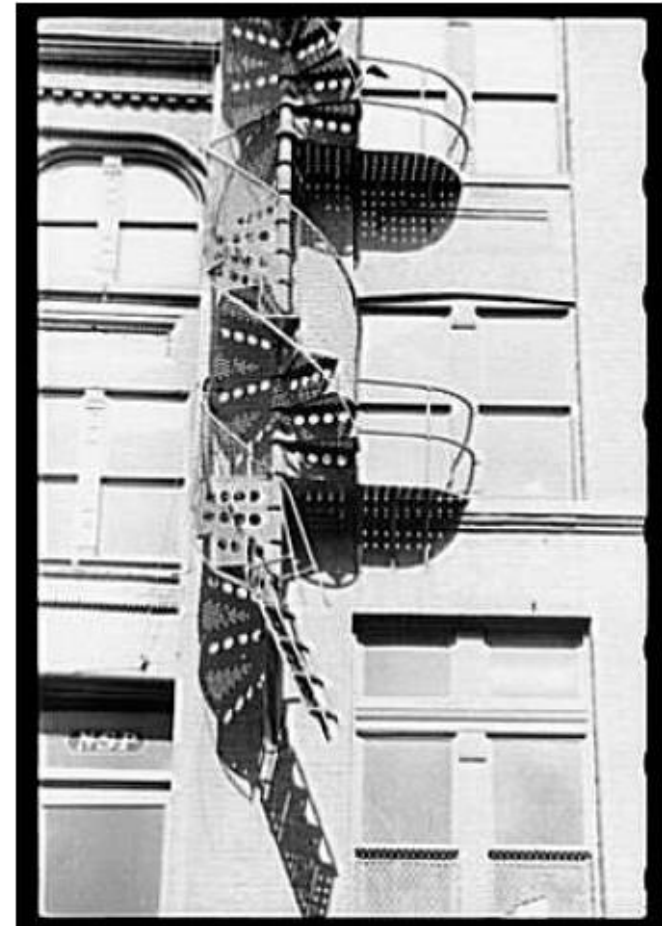
Muitas abordagens de resolução de problemas passam pela divisão em frações, processo que designamos por **abstração**

- Admitimos que o que estamos a considerar é complexo e em vez de tentarmos compreender o todo selecionamos parte dele. Sabemos que existem detalhes adicionais, simplesmente optamos por não os considerar naquele momento

Técnica para **gerir a complexidade**

Processo: *SpiralDevelopment*

- Problema
 - Desenvolver casos de uso num único passo é difícil, impede a incorporação de novos dados, podendo atrasar a descoberta de fatores de risco
- Solução
 - Desenvolver os casos de uso num **processo iterativo** e em cada iteração aumentar progressivamente a precisão e a objetividade dos casos de uso



Processo: *MultipleForms*

- Problema
 - Projetos diferentes necessitam de diferentes formalidades durante a sua execução. Pessoas diferentes têm preferências diferentes para o desenvolvimento
- Solução
 - Selecionar um formato de desenvolvimento baseado nos **riscos** do projeto e nas **preferências** das **pessoas** envolvidas



Processo: *MultipleForms*

- Prioridade, tamanho, pré e pós condições, ...

Use Case Narrative: Make Travel Plans

Basic Flow:

1. Traveller provides travel details (travel dates and destination)
2. System searches and displays hotels for the travel dates and destination.
3. Traveller selects a hotel and room type.
4. System computes and display.
5. System makes a tentative reservation for the traveler on the selected hotel.

Alternate Flows:

- A1. Travel plan with multiple destinations
- A2. Travel plan having a single destination but non-consecutive dates
- A3. Travel plan with non-consecutive dates and multiple destinations

Processo: *QuittingTime*

- Problema
 - Desenvolver o modelo de casos de uso para além das necessidades dos *stakeholders*, consome recursos e atrasa o projeto
- Solução
 - Desenvolver os casos de uso até ficarem completos e **satisfazerem** adequadamente as necessidades da audiência



CrITÉRIOS de paragem:

1. Identificados e documentados todos os atores e objetivos do sistema?
2. O cliente, reconheceu os casos de uso como completos, legíveis e corretos?
3. Os programadores conseguem implementar?

Processo: *TwoTierReview*

- Problema
 - Muitas pessoas podem necessitar de rever os casos de uso, tornando este processo caro e demorado
- Solução
 - Assegurar dois tipos de revisões:

1ª pela **equipa** a nível interno e repetida várias vezes

2ª realizada pelo **grupo** completo, podendo fazer só uma única revisão

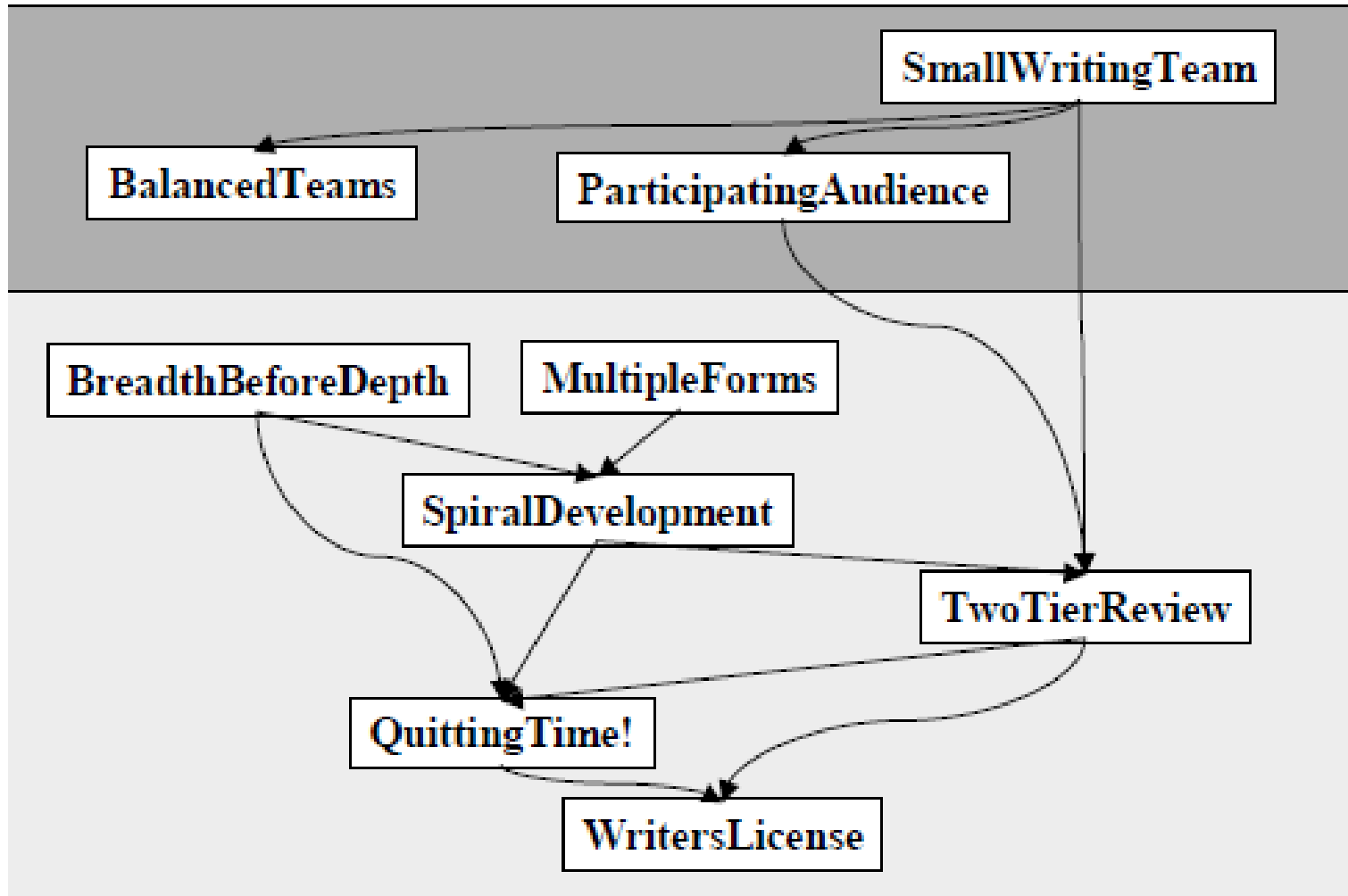


Processo: *WritersLicence*

- Problema
 - ênfase excessiva no estilo dificulta a escrita dos casos de uso
- Solução
 - permitir a **criatividade** individual
 - seguir o **estilo básico** da organização
 - estar **logicamente** correto
 - ser **legível** para os avaliadores finais
 - ser suficientemente **preciso** para ser implementado



Rede padrões: Equipa e Processo



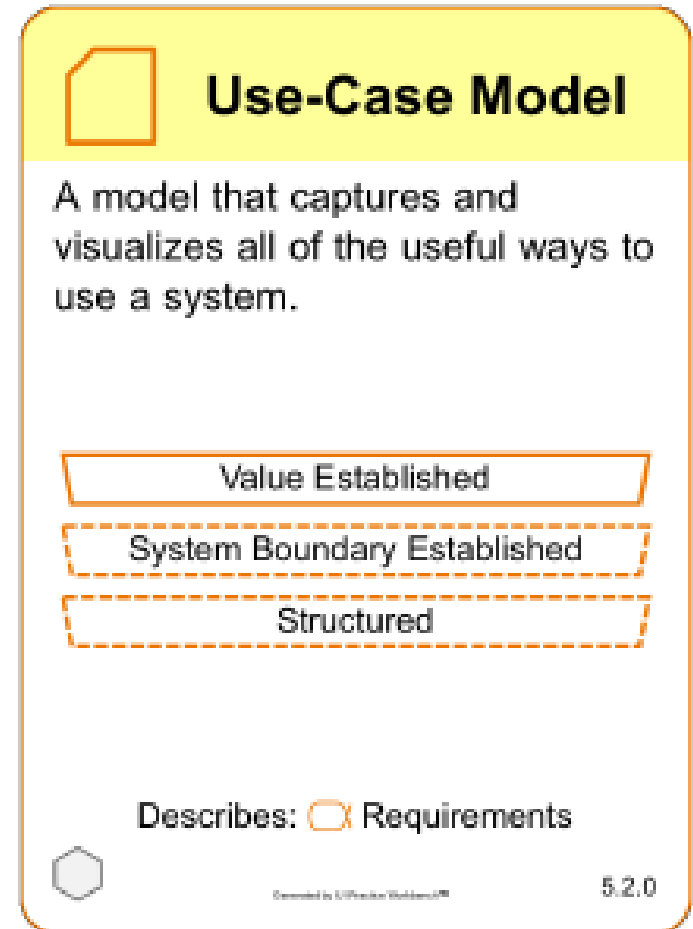
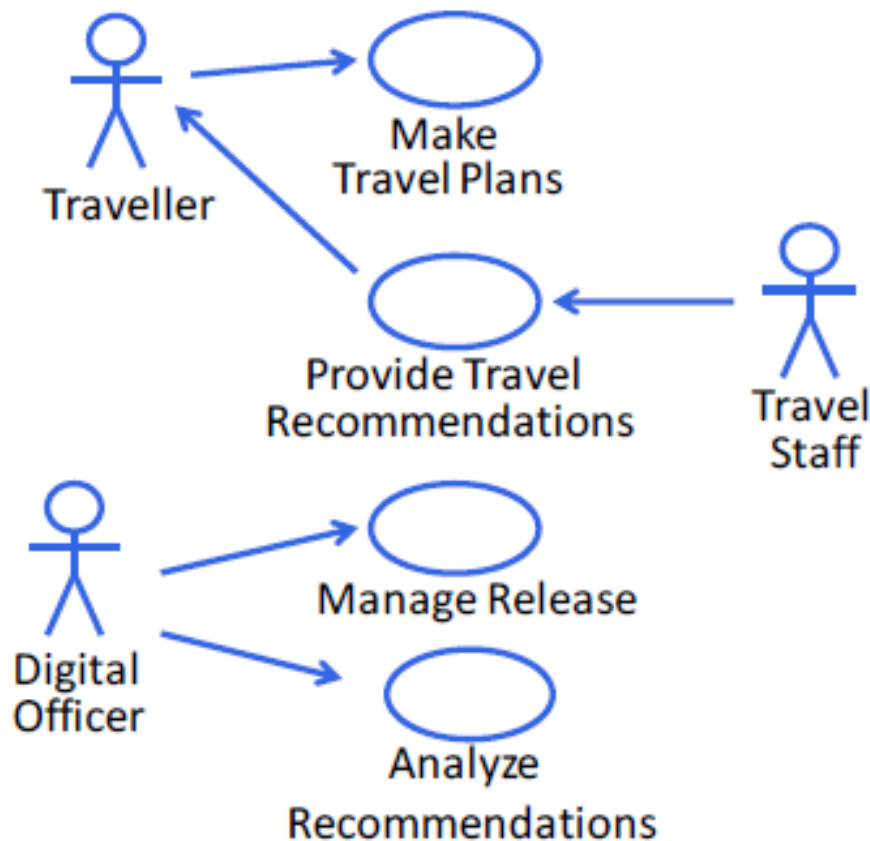
Padrões de Estrutura

Conjunto de casos de uso



- **SharedClearVision**: Identificar claramente os **objetivos** do sistema. Assegurar uma visão clara dos objetivos e partilhada por todos os *stakeholders*
- **VisibleBoundary**: Estabelecer uma **fronteira visível** entre o sistema e o seu ambiente, enumerando quem e o quê pode interagir com o sistema
- **ClearCastOfCharacters**: Identificar os atores que interagem com o sistema e o **papel que cada um deles representa**
- **UserValuedTransactions**: Identificar os **serviços essenciais** para satisfazer os objetivos do negócio
- **EverUnfoldingStory**: Organizar o conjunto de casos de uso como uma **história hierárquica** de maneira que possa ser desdobrada para adquirir mais detalhe ou compactada para adquirir mais contexto

Diagrama casos de uso



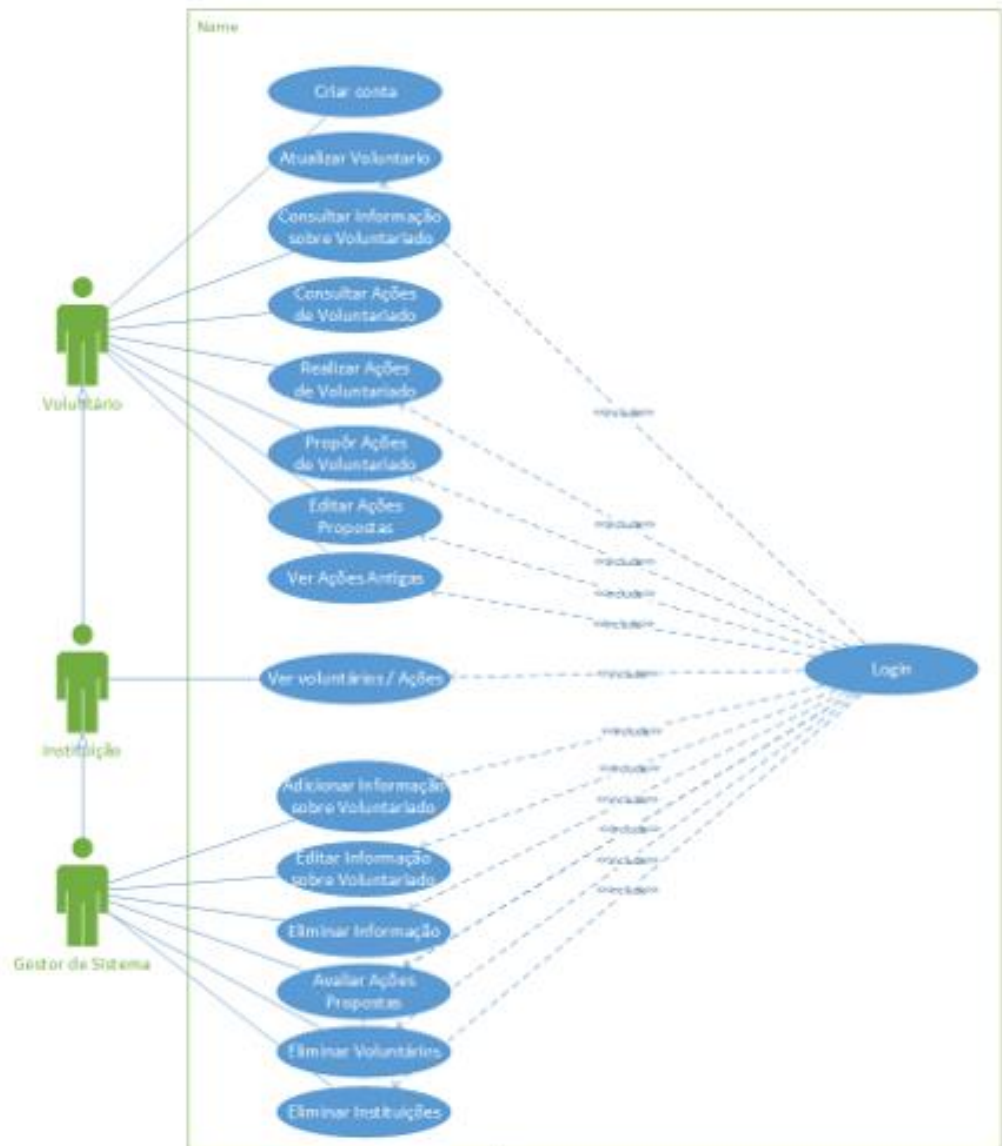
Software Engineering Essentialized; Jacobson et al, 2017

EverUnfoldingStory

- Organizar o conjunto de casos de uso como uma **história hierárquica** de maneira que possa ser desdobrada para adquirir mais



Organizar o conjunto de casos de uso como uma **história hierárquica** de maneira que possa ser desdobrada para adquirir mais detalhe ou compactada para adquirir mais contexto



Padrões de Estrutura Casos de Uso individuais



- **CompleteSingleGoal**: Escrever cada caso de uso com um **objetivo** completo e bem definido.
- **VerbPhraseName**: Nome do caso de uso deve ter **verbo forte** numa frase representando o objetivo do ator principal
- **ScenarioPlusFragments**: história como um **cenário simples** sem considerar possíveis falhas
- **ExhaustiveAlternatives**: Identificar todas as **alternativas** e falhas
- **Adornments**: criar **campos adicionais** para requisitos não funcionais
- **PreciseAndReadable**: caso de uso **acessível à audiência** menos técnica e **suficientemente preciso** para o programador

Exemplo descrição de um caso de uso

- Use-Case Narrative example:

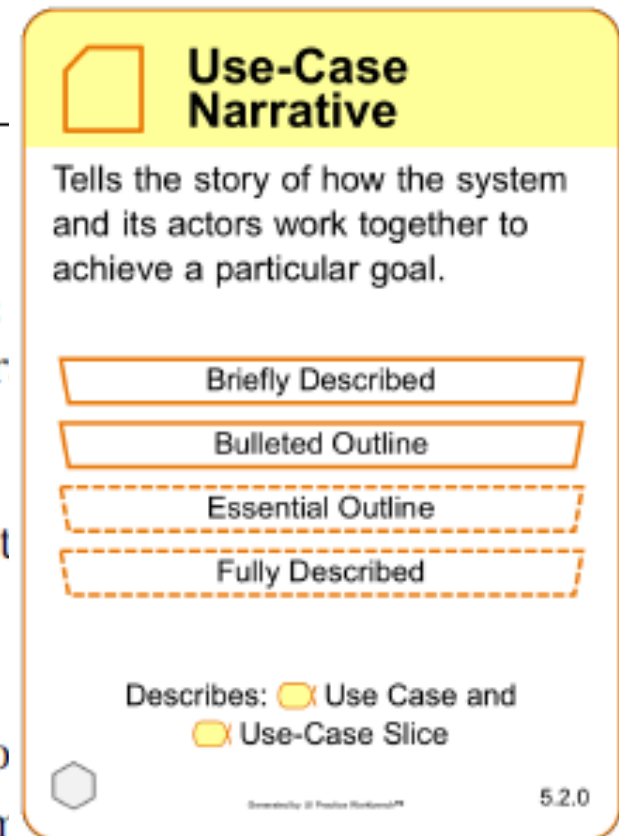
Use Case Narrative: Make Travel Plans

Basic Flow:

1. Traveller provides travel details (travel dates
2. System searches and displays hotels for the tr
3. Traveller selects a hotel and room type.
4. System computes and display.
5. System makes a tentative reservation for the t

Alternate Flows:

- A1. Travel plan with multiple destinations
- A2. Travel plan having a single destination but no
- A3. Travel plan with non-consecutive dates and n



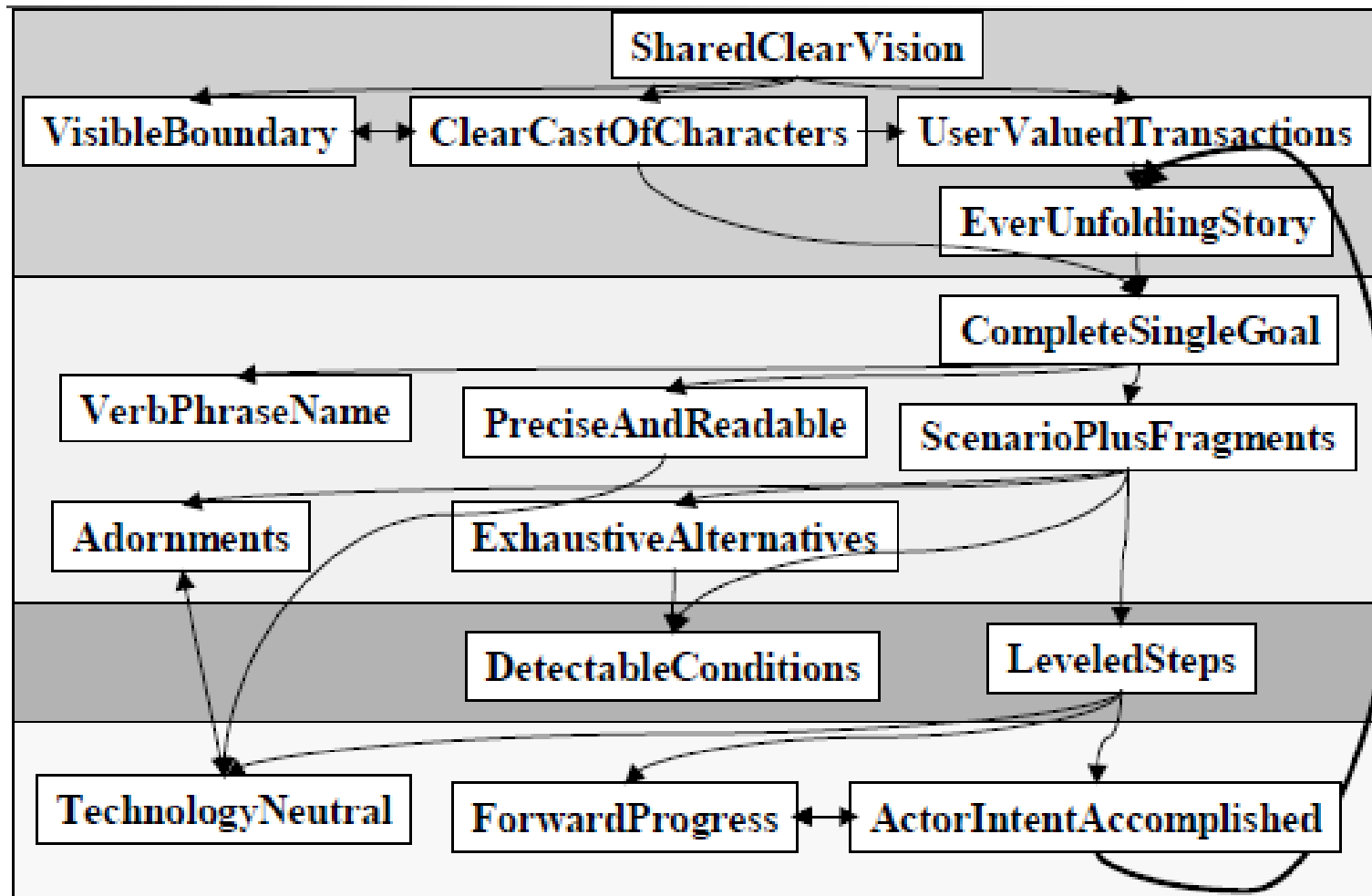
Software Engineering Essentialized; Jacobson et al, 2017

Padrões de Estrutura Cenários e passos



- **DetectableConditions:** soluções concretas
- **LeveledSteps:** 3 a 9 passos e nivelados (detalhe)
- **ActorIntentAccomplished:** cada passo mostrar claramente qual a acção que o actor vai desempenhar
- **ForwardProgress:** Eliminar ou juntar os passos que não deixam o ator avançar
- **TechnologyNeutral:** Escrever cada caso de uso de uma forma que seja independente da tecnologia

Rede padrões de Estrutura



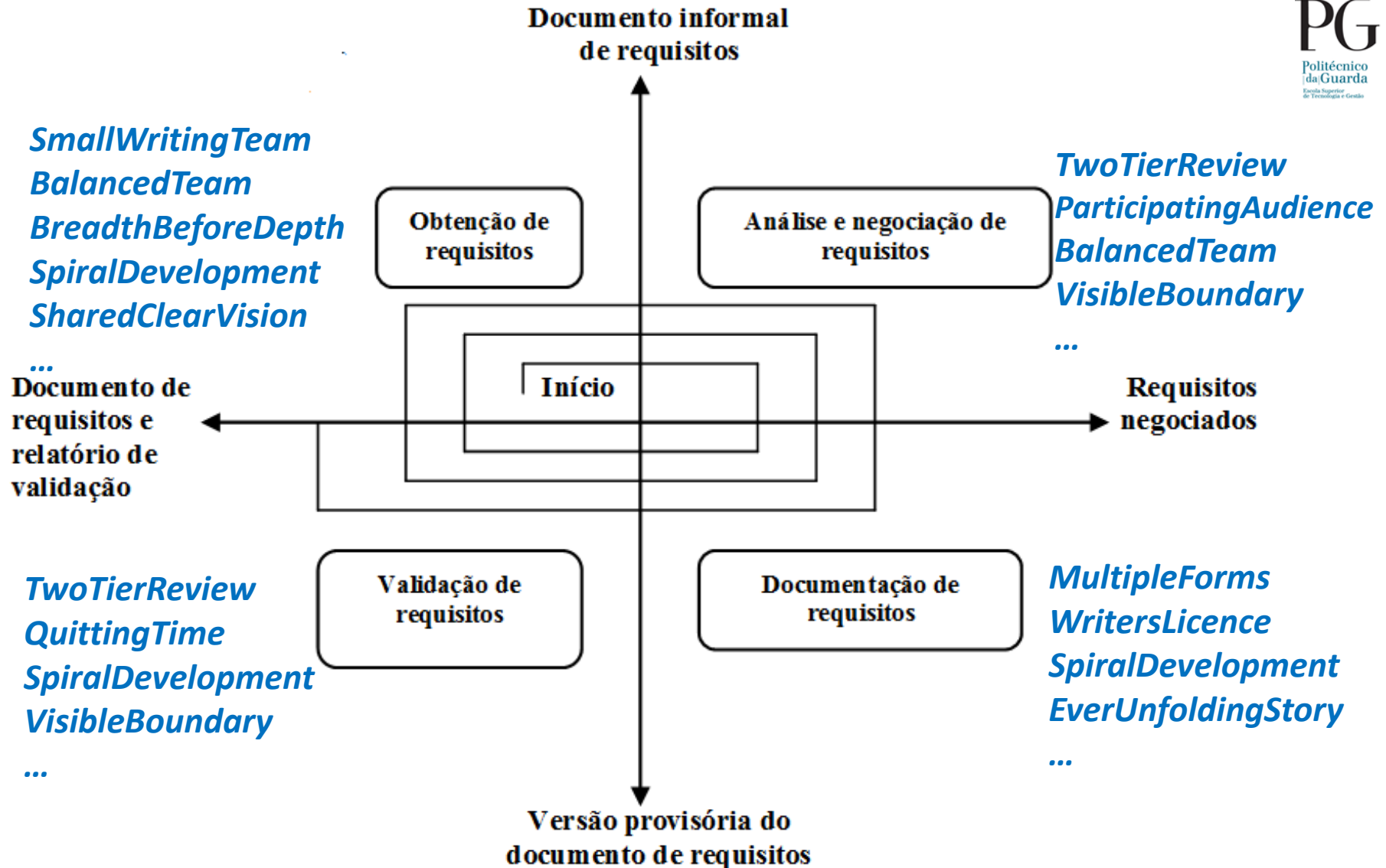
Use Case Relationships: Include

- **CommonSubBehavior**: diz-nos que só se deve usar a relação << *include* >> entre casos de uso quando dois ou mais casos de uso partilham um conjunto comum de acções
- **Objectivo**: consolidar o comportamento comum e simplificar o modelo eliminando a redundância
- Usados para reutilizar texto
- São abstractos – não existem sozinhos
- Evitar *include* como mecanismo de simplificação de casos de uso complexos

Padrões de Edição

- Têm por objetivo tratar três situações:
 - **Dividir** os casos de uso extremamente grandes e reduzi-los para um tamanho utilizável (RedistributeTheWealth)
 - **Combinar** fragmentos incompletos de casos de uso em novos casos de uso ou em casos de uso existentes (MergeDroplets)
 - **Eliminar** casos de uso desnecessários (CleanHouse)

Padrões no Processo ER



Resumindo

- Os **padrões** representam conhecimento e experiência que pode ser transmitida e ensinada
 - boas práticas
 - lições apreendidas
 - partilha de experiência
 - reduzem a complexidade
 - são independentes dos paradigmas ou estilos utilizados

Resumindo

- Os padrões contribuem para a existência de **sinais de qualidade** na Engenharia de Requisitos
- Incluir as técnicas de **criatividade** na Engenharia de Requisitos para promover a **inovação** nos produtos de software