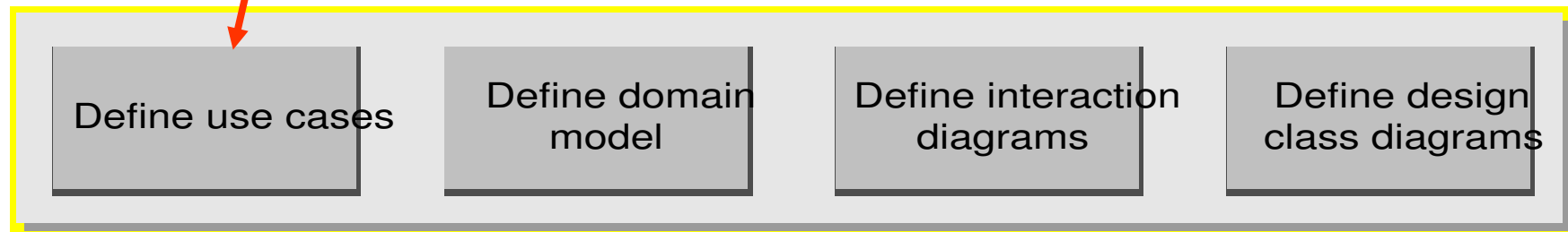

Use-Case Model: Escrevendo Requisitos

Objetivos

Modelo FURPS+

Identificar e escrever *use cases*



Estudo de Caso: O Sistema NextGen POS

Point Of Sale [POS]
system

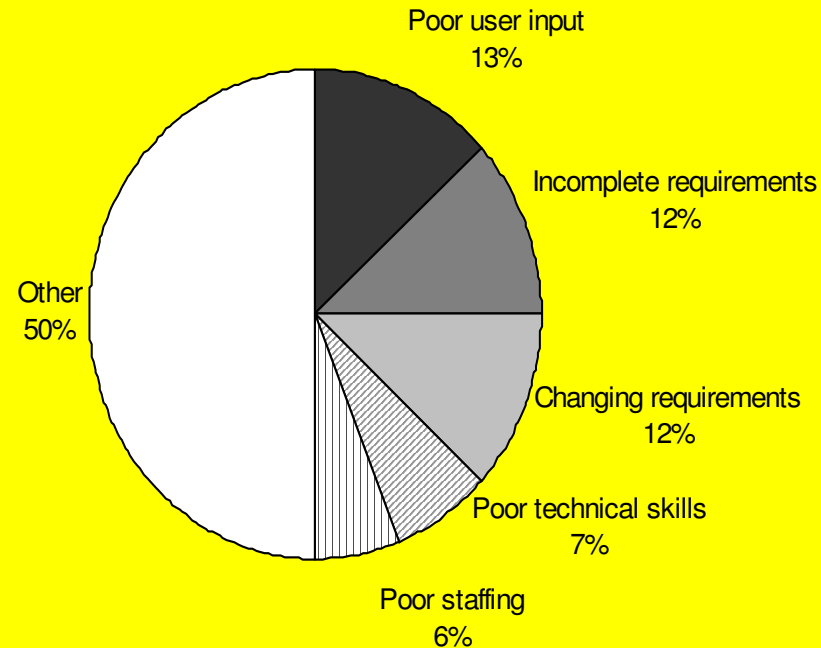


O Sistema NextGen POS

- *A POS system is a computerized application used (in part) to record sales and handle payments; it is typically used in a retail store. It includes hardware components such as a computer and a bar code scanner, and a sftw to run the system. It interfaces to various service applications, such as a third-party tax calculator and inventory control. These systems must be relatively fault-tolerant; that is, even if remote services are temporarily unavailable (such as the inventory system), they must still be capable of capturing sales and handling at least cash payments (so that the busines is not crippled)*

Desafios no projeto de sftw

37% de fatores relacionados a problemas com requisitos



Mudança e feedback são fundamentais na descoberta de requisitos

O Modelo *FURPS+*

- No UP requisitos são classificados em:
 - **Functional** – *features*, funcionalidades
 - **Usability** – fatores humanos, help, doc.
 - **Reliability** – freqüência de erros, recuperação, previsibilidade
 - **Performance** – tempos de resposta, precisão
 - **Supportability** - adaptabilidade, manutenibilidade, internacionalização

- +

- Implementação, Interface, Operações, Empacotamento, Aspectos Legais
- Categorias FURPS+ são importantes como checklist para requisitos
- Requisitos funcionais são explorados e registrados no *Use-Case Model*
- Discussão sobre requisitos:
 - www.swebok.org [Sfte Eng. Body of Knowledge]

Use-Case Model

- Um modelo da funcionalidade e ambiente do sistema
- *Use-cases*: estórias de uso do sistema para atingir metas

Processa Venda: *A customer arrives at a checkout with items to purchase. The cashier uses the POS system to record each purchased item. The system presents a running total and line-item details. The customer enters payment information, which the system validates and records. The system updates inventory. The customer receives a receipt from the system and then leaves with the items.*

- Um UC é uma coleção de cenários relacionados, de sucesso e fracasso, que descrevem atores usando o sistema para atingir uma meta.

Lida com Devoluções:

Cenário Principal de Sucesso: *a customer arrives at a checkout with items to return. The cashier uses the POS system to record each returned item...*

Cenários Alternativos:

if they paid by credit, and the reimbursement transaction to their credit account is rejected, inform the customer and pay them with cash

if the system detects failure to communicate with the external accounting system, ...

Algumas Definições

- **Ator:** algo que tem comportamento; ex. uma pessoa [papel], sistema computacional, organização.
 - ex. A caixa do supermercado
- **Cenário:** uma seqüência específica de ações e interações entre atores e o sistema sob discussão
 - Tb chamado instância do UC
- use-cases definem uma promessa ou contrato de **como o sistema irá se comportar**

- O quê o sistema deve fazer (os requisitos funcionais),
-
- sem decidir como ele irá fazê-lo (design)

Estilo Caixa Preta :

The system records the sale

Não

The system generates a SQL Insert statement for the sale

C1

Slide 11

C1

Cecilia; 19/4/2006

Use cases são escritos com níveis variados de formalidade

- **Tipos:**
 - **breve**: sumário de 1 parágrafo, usualmente o cenário principal de sucesso [o *processa venda* ex.]
 - **casual**: múltiplos parágrafos que cobrem vários cenários [o *lida com devoluções* ex.]
 - **fully dressed**: todos os passos e variações escritas em detalhe
- ver template em www.usecases.org

As Seções de um fully dressed use case

- **Ator Primário**: ator principal que chama os serviços do sistema para realizar uma meta
- **Stakeholders e Interessados**: *“O quê deve estar em um use case?”* *“Aquilo que satisfaz a todos os stakeholders”*
- **Pré-condições**: estado que deve sempre ser verdadeiro antes de iniciar um cenário
- **Pós-condições (garantias de sucesso)**: estado que **deve** ser verdadeiro na finalização bem sucedida do cenário

Cenário principal de sucesso: Caminho de sucesso típico que satisfaz interesses dos stakeholders

Não inclui condições ou ramificações

- O cenário registra passos de 3 tipos:
 - Uma interação entre atores
 - Uma validação [usualmente pelo sistema]
 - Uma mudança de estado pelo sistema

1. Customer arrives at a POS checkout with items to purchase.

2. Cashier starts a new sale.

...

Extensões ou Fluxos Alternativos:

- Ramos do cenário principal de sucesso
- Uma extensão tem 2 partes: a condição e a ação (*handling*)

3-6a: Customer asks Cashier to remove an item from the purchase:

- 1. Cashier enters the item identifier for removal from the sale.*
- 2. System displays updated running total*

Requisitos Especiais: registro de requisitos não-funcionais, atributo de qualidade ou restrição relacionada ao UC

- *credit authorisation response within 30 seconds.*

Tecnologia e Lista de Variação de Dados:

- *3a Item identifier entered by laser scanner or keyboard*

Metas e Escopo do UC

- Em que nível e escopo devem os UC ser expressos?
- Quais destes são use case válidos?
 - *Negotiate a Supplier Contract*
 - *Handle Returns*
 - *Log In*
- Uma questão mais relevante:
 - Que nível seria útil para expressar UC para análise de requisitos de aplicação?*

Guideline: o EBP Use Case

- Para análise de requisitos focar em casos de uso no nível de *Elementary Business Processes* (EBPs)
 - EBP: uma tarefa realizada por uma pessoa em um lugar por vez, em resposta a um evento de negócio, que adiciona valor mensurável ao negócio e deixa os dados em um estado consistente
 - Ex. *approve Credit or Price Order*

Use Cases e metas do usuário

- Atores têm metas [ou necessidades] e usam aplicações para ajudar a realizá-las.
 - UC de nível EBP é chamado *user goal-level* para enfatizar que ele serve [ou deveria servir] para satisfazer uma meta do usuário do sistema [o ator primário]
 - *1. Find the user goals*
 - *2. Define a use case for each*
- Em vez de “quais são os uc” “quais são as metas”

Encontrando Atores Primários, Metas e Casos de Uso

1. Escolha a fronteira do Sistema
 - É apenas uma aplicação de sftw, o hrdw e a aplicação como uma unidade, aquilo + uma pessoa usando, ou a organização inteira?
2. Identifique os atores primários
 - Aqueles que têm as metas de usuário realizadas ao utilizar serviços do sistema
3. Para cada um
 - identificar suas metas de usuário [EBP guideline]
4. Definir use cases que satisfaçam metas do usuário
 - Nomeá-los de acordo com suas metas

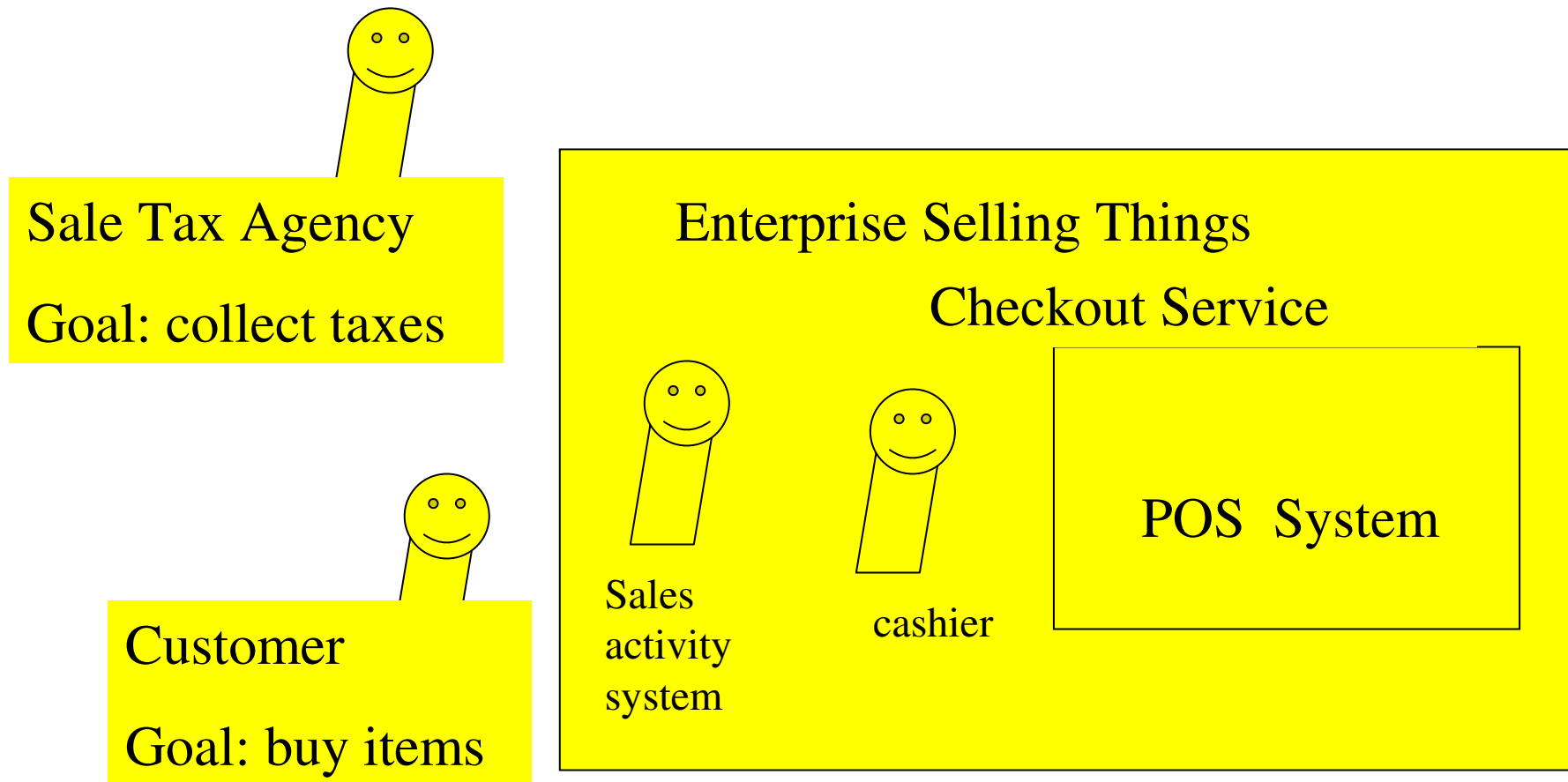
Encontrando as Fronteiras do Sistema

- Para o estudo de caso, o *POS system* é o sistema em design. Tudo fora dele está fora da fronteira do sistema, incluindo a caixa
- Definir o que está fora. Ex.
 - *Is the complete responsibility for payment authorisation within the system boundary?*
 - *No. there is an external payment authorisation service actor*

Questões para encontrar Atores e Metas

- Quem inicia e pára o sistema?
- Quem gerencia usuário e segurança?
- Há processo de monitoramento que restarta o sistema se ele falha?
- Como updates do sftw são tratados? [push or pull]?
- Quem avalia atividade e performance do sistema?
- Quem avalia logs?
- Quem administra o sistema?

Atores Primários e Metas em diferentes fronteiras do sistema



Use Case no estilo essencial

- A narrativa é expressa no nível de **intenções** do usuário e **responsabilidades** do sistema em vez de suas ações concretas
- Permanecem livres de tecnologia e detalhes de mecanismos Ex.
 - *1. Administrator identifies self*
 - *2. System authenticates identity*

Use Case no Estilo Concreto

- Decisões de Interface de Usuário são embutidas no texto do UC.
 - O texto pode mostrar janelas de interface
- Ex.
 - 1. Administrator enters ID and password in dialog box [see picture x]*
 - 2. System authenticates administrator*
 - 3. System displays the “edit users” windows [see picture y]*

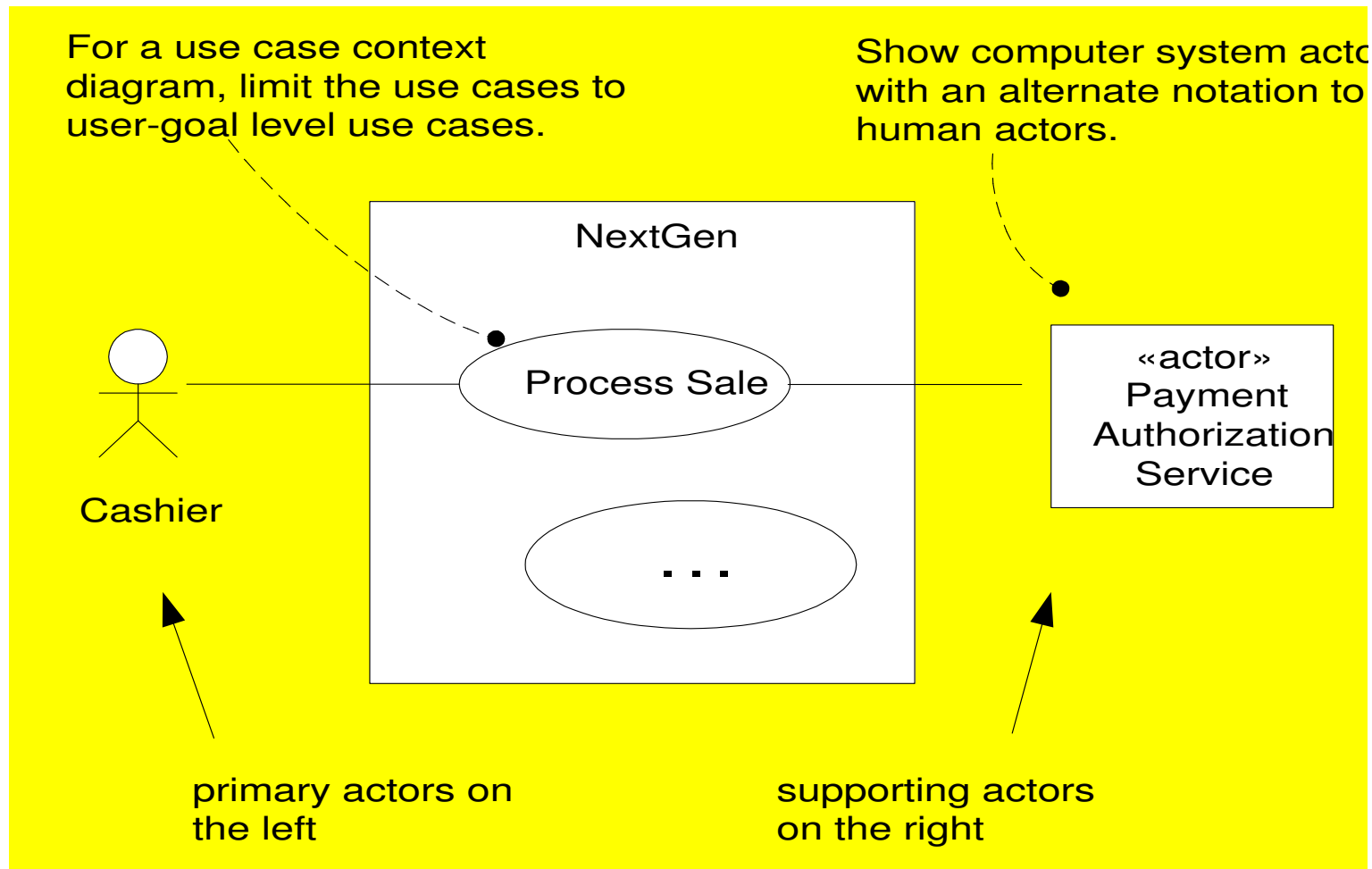
Atores

- Qualquer elemento com comportamento, incluindo o *system under discussion* (SuD) qdo chama serviços de outros sistemas
- 3 tipos de atores em relação ao SuD
 - **Ator Primário**: tem as metas realizadas pelo uso de serviços do sistema SuD [caixa]
 - **Ator de Suporte**: provê um serviço [informação] ao SuD [o serviço de pagamento automático]
 - **Ator fora do palco**: tem interesse no comportamento do UC [agência governamental de cobrança de imposto]

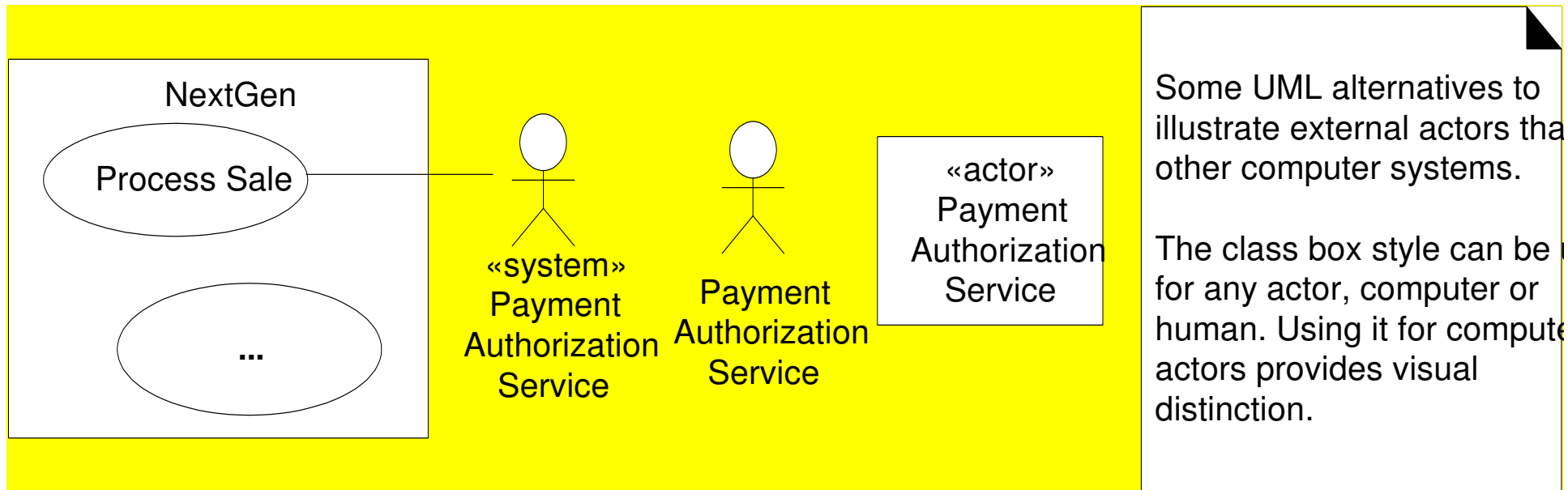
Diagrama de Use Case

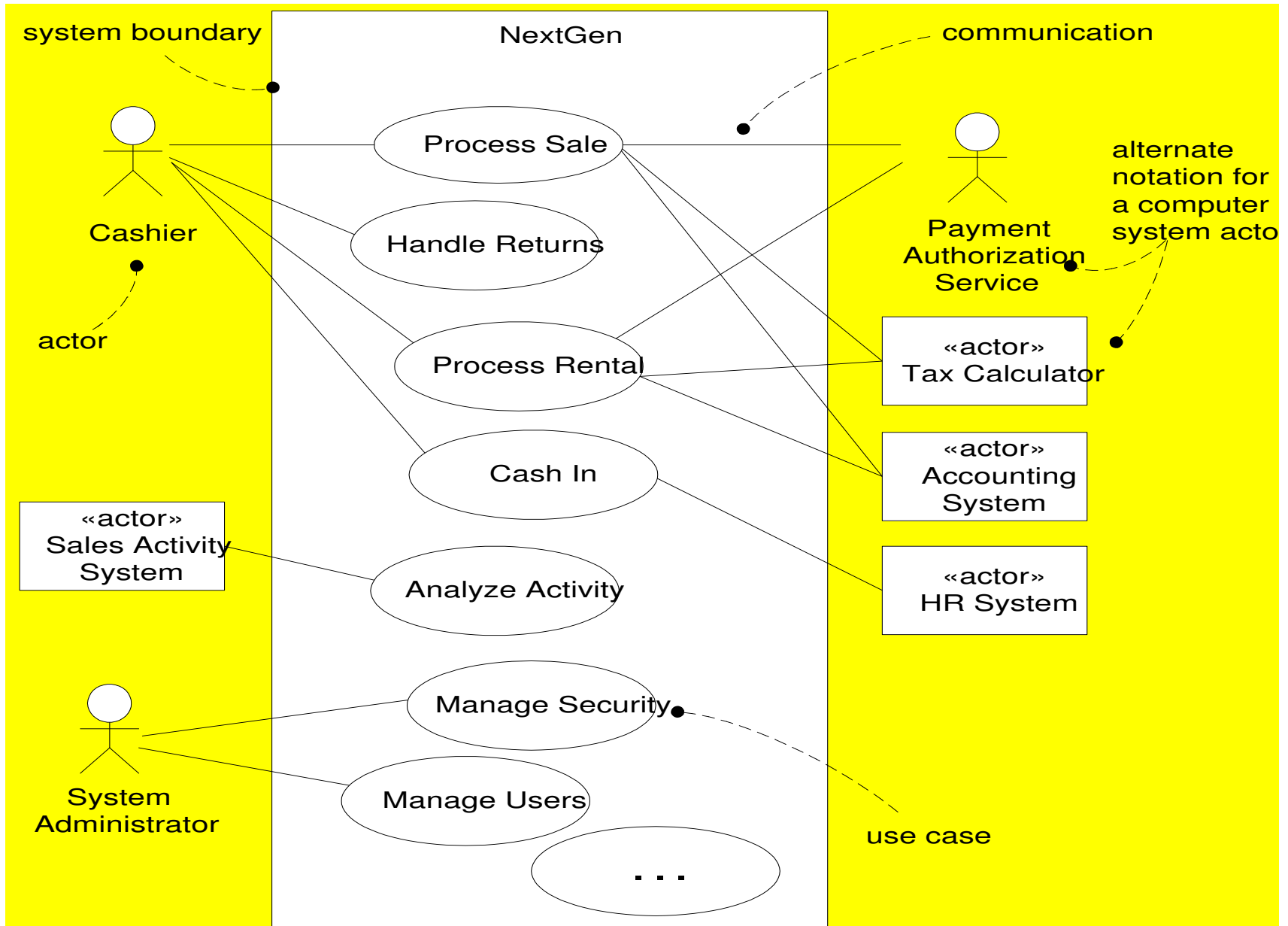
- **UML** provê notação para diagrama de use case para ilustrar os nomes dos UC, atores e relações entre eles
 - É um diagrama sucinto que ilustra visualmente o sistema mostrando: atores externos e como eles usam o sistema
- **O trabalho importante no Use Case é escrever texto primeiro.**

Diagrama de Use Cases



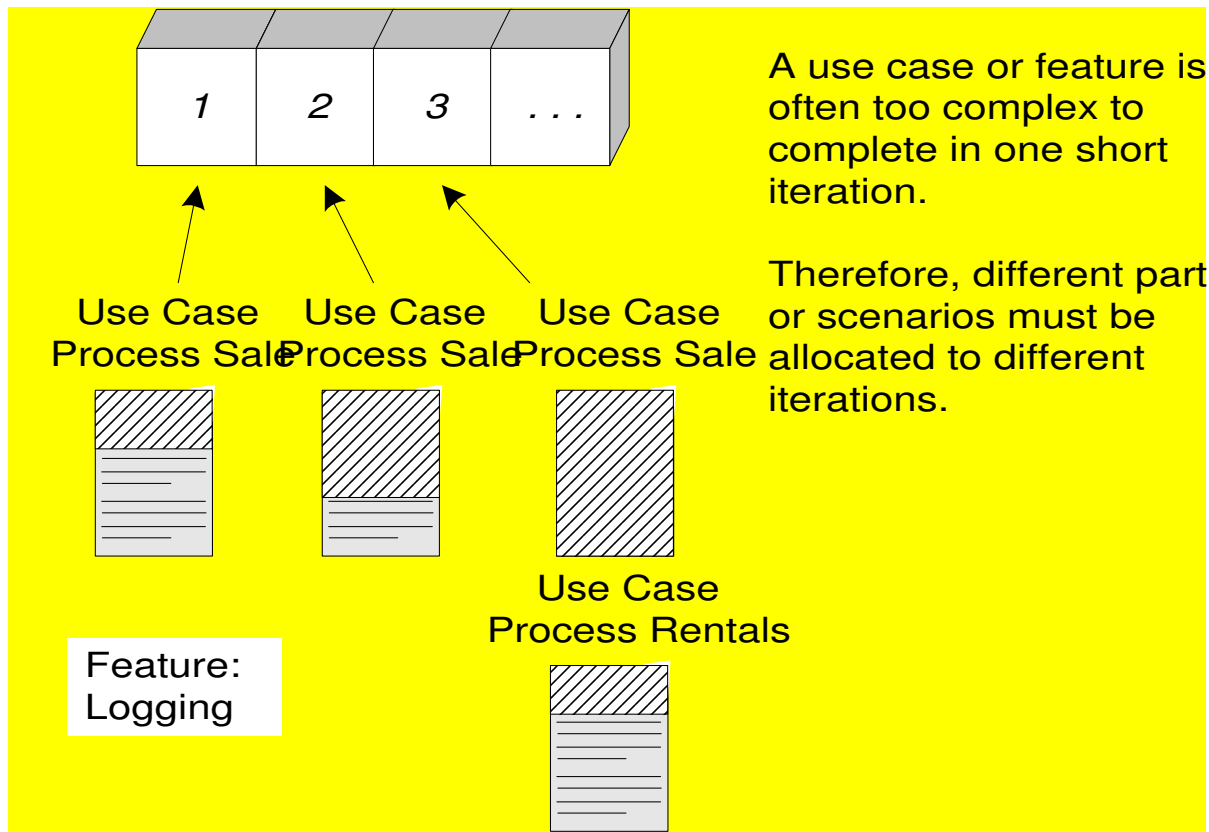
Notação alternativa para Atores





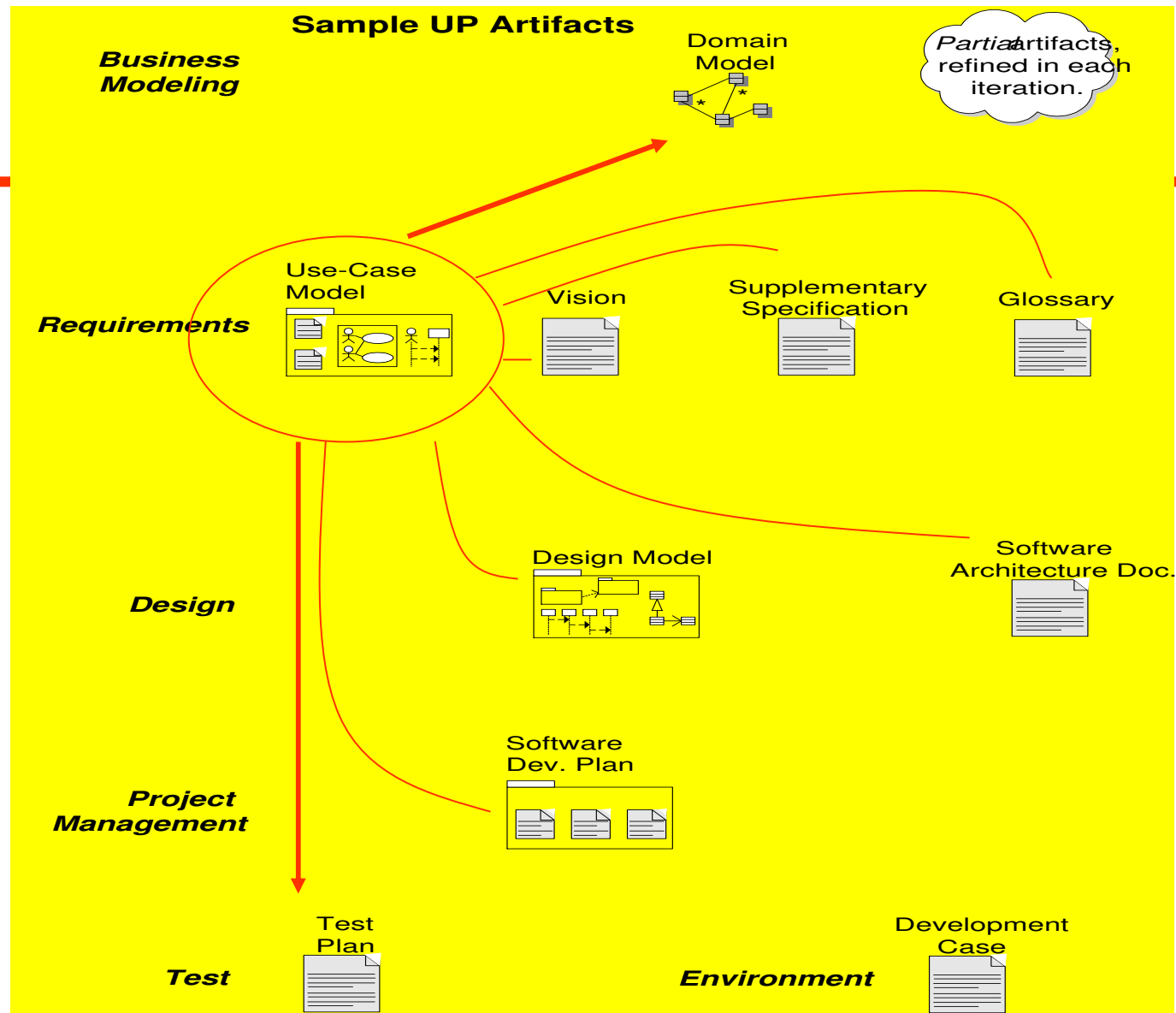
Use Cases na Fase de Inception do PU

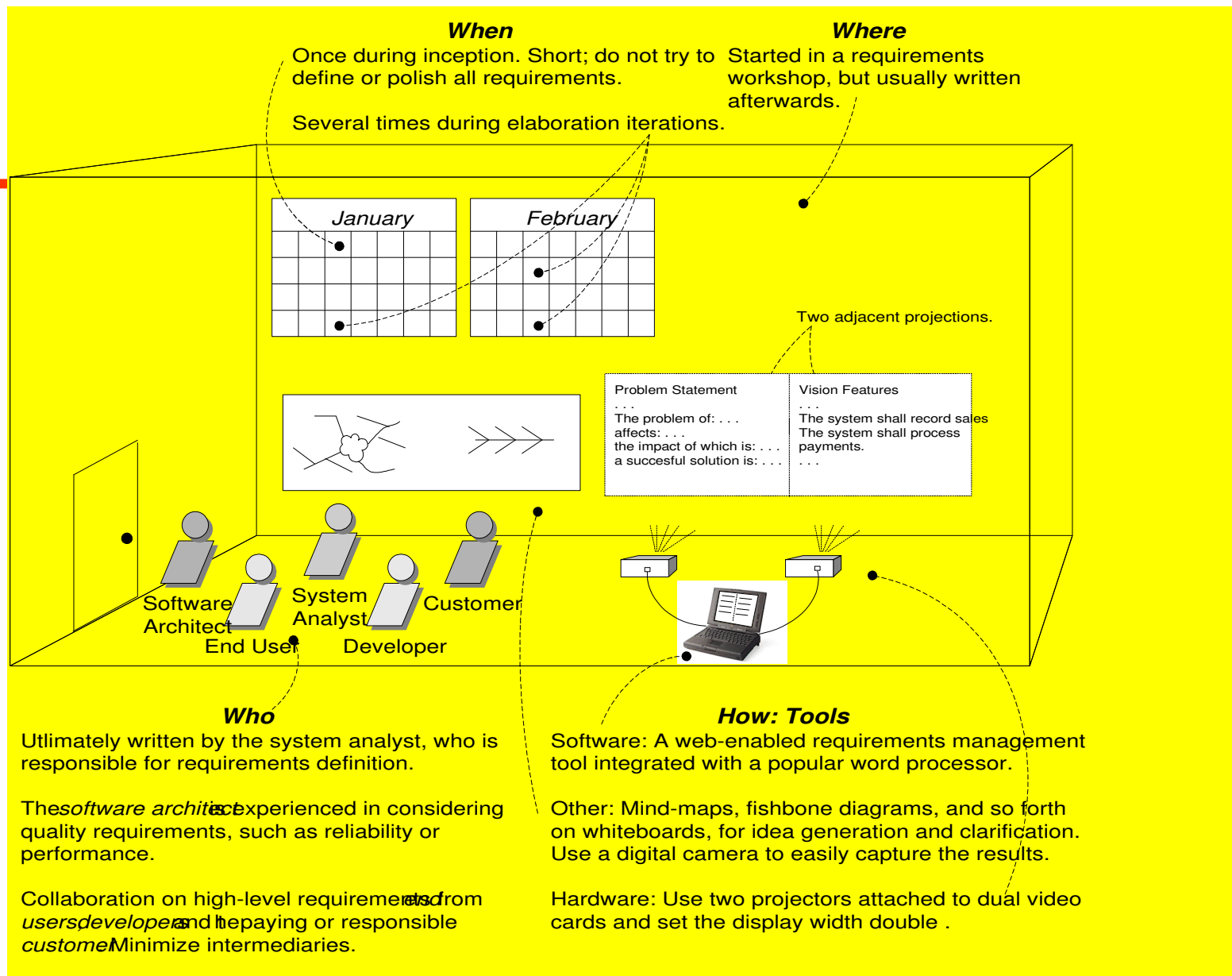
- Nem todos os UC são escritos na forma *fully dressed* nesta fase
- Para o Estudo:
 - ***Fully dressed:***
 - *Process Sale, Handle Returns*
 - **Casual:**
 - *Process Rental, Analyse Sales Activity, Manage Security, etc.*
 - **Breve:**
 - *Cash in, Cash out, Start up, Shut down, etc.*

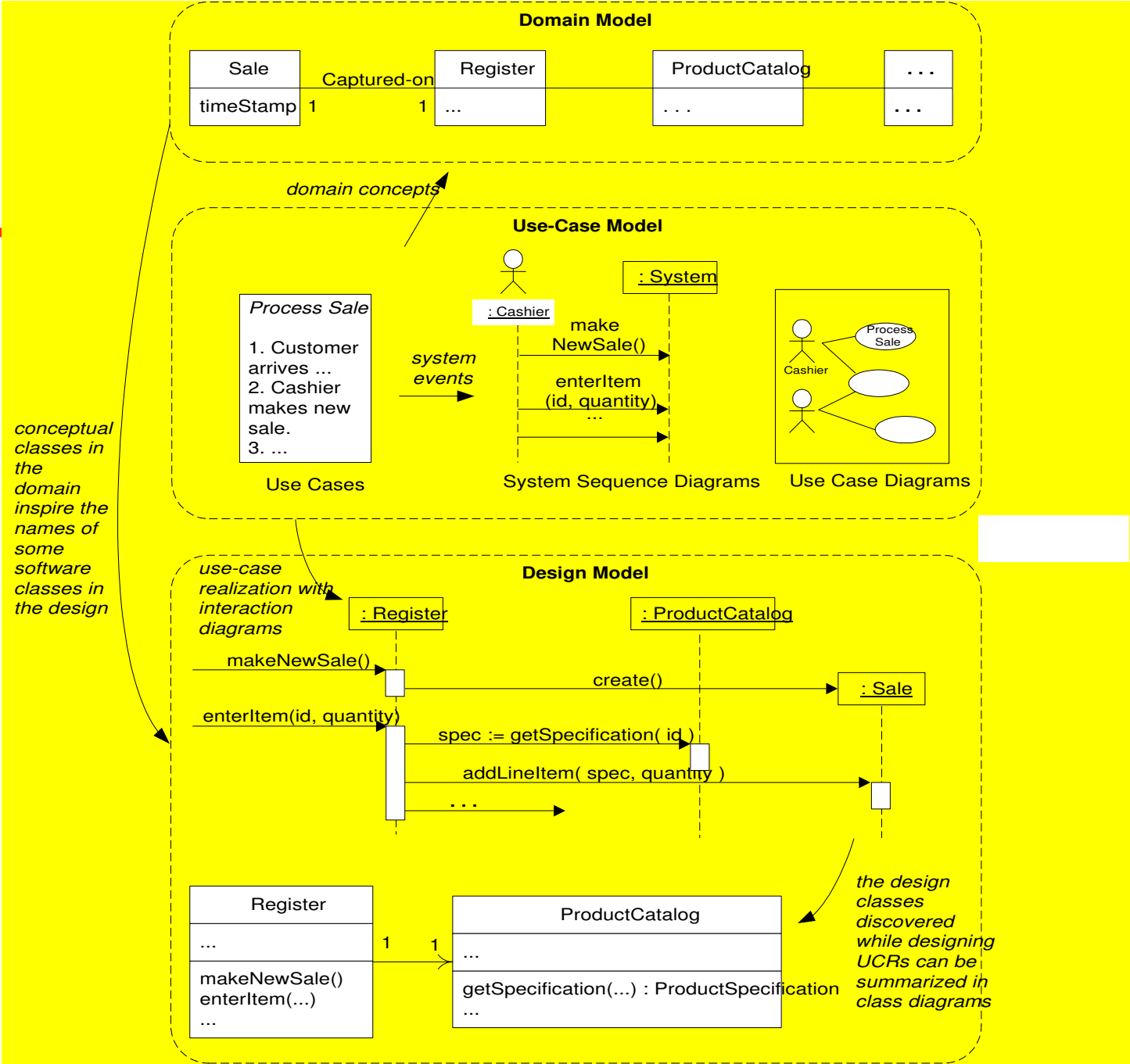


A use case or feature is often too complex to complete in one short iteration.

Therefore, different parts or scenarios must be allocated to different iterations.







Referências

- Larman, C. (2002) *Applying UML and Patterns – An Introduction to Object Oriented Analysis and Design and the Unified Process*, Prentice-Hall Inc.
- ©Ian Sommerville 2007 Software Engineering, 8th edition. Chapter 6