Applying UML and Patterns

An Introduction to Object-oriented Analysis and Design and Iterative Development

Part II Inception

Chapters

Chap 6 Use Cases

★★

Actors(参与者)

- Actor: **external entity** interacts (behavior) with system, such as a person (identified by role), computer system, or organization; for example, a cashier.
- Three kind of Actors
  - **Primary actor** has user goals fulfilled through using services. (e.g., the cashier). Find user goals to drive the use cases.
  - **Supporting actor** provides a service (e.g., the automated payment authorization service is an example). Often a computer system, but could be an organization or person. The purpose is to clarify external interfaces and protocols.
  - **Offstage actor** has an interest in the behavior of the use case, but is not primary or supporting (e.g., a government tax agency).
- 最特殊的参与者
  - 系统时钟,例如：schedule

识别Actor练习

- “牧师与魔鬼”小游戏
  - 玩家？ 时钟？ 开发人员？
- “神庙逃亡”游戏
  - 玩家？ 时钟？ Facebook？
- 小明打算制作一款“中大课程表”的手机应用。它从教务系统获取学生的选课信息和课程安排。教师用它可以给学生和班委推送通知，学生查看课程表，了解上课时间和教室安排；设置课前提醒等。
  - ？

★

Use Case(用例) 1

- Use case
  - is a collection of related **success and failure scenarios** that describe an actor using a system to support a **goal**.
  - be **text** documents, not diagrams
  - Use case modeling is primarily an act of writing text, not drawing diagrams.
  - There **is nothing object-oriented about use cases**;
  - Use cases are a key **requirements** input to classic OOA/D.
  - be functional or behavioral requirements that indicate what the system will do. In terms of the FURPS+ requirements types, they **emphasize the "F"**, but can **also** be used for other types.

Use Case 2

- The usage of use case
  - Decide and describe the **functional** requirements of the system
  - Bring **agreement** between the customer and software developer
  - Give **a clear and consistent description** of what the system should do.
  - Provide a basis for performing **tests that verify** the system delivers the functionality stated.
  - **Trace** the functional requirements into actual classes and operations in the system.

Scenarios(场景) 1

- Scenario
  - be a specific sequence of actions and interactions（会话） between actors and the system; it is also called **a use case instance**.
  - It is one particular story of using a system, or one path through the use case.
  - for example, the scenario of successfully purchasing items with cash, or the scenario of failing to purchase items because of a credit payment denial.

I would like a book of stamps, please.
OK. Will that be all?
Yes.
That will be $7.80.
Here is $10.
Thanks. Here are your stamps and your change.

Use Case & Scenarios 2

- Scenario
  - A use case represents a collection of scenarios: primary, plus zero or more alternates.
  - The primary scenario（主场景 / 基本流） corresponds to the main system interactions, usually the ‘success’ scenario. 最常用，直接地实现用户目标的故事
  - Alternate scenarios（可选场景 / 备选流） correspond to less frequent interactions and exceptions. 在实现用户目标过程中较少适用与意外故事

Use Case 案例：打电话

- System under Design(SuD): 电话系统
- Goal：与被叫方通话
- Actor:
  - 主叫方(primary)，被叫方
  - 计费系统(supporting)
  - 运营商
- Primary scenario:
  - 拨号，系统建立连接，回呼叫音
  - 系统连接完成，取消呼叫音
  - 与被叫方通话
  - 挂机，系统拆线
- Alternate scenario: 占线
  - 拨号，系统建立连接，回忙音
  - 挂机，系统拆线
- Alternate scenario: 号码不存在
  - 。。。

简单用例练习

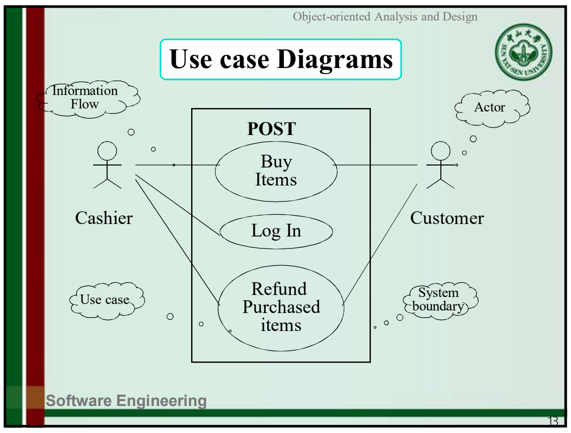
小明打算制作一款“中大课程表”的手机应用。它从教务系统获取学生的选课信息和课程安排。教师用它可以给学生和班委推送通知，学生查看课程表，了解上课时间和教室安排；设置课前提醒等。

- 请描述学生“设置课前提醒”的用例（Use Case）
  - Goal:
  - Actor:
  - Primary scenario:
  - List of alternate scenarios

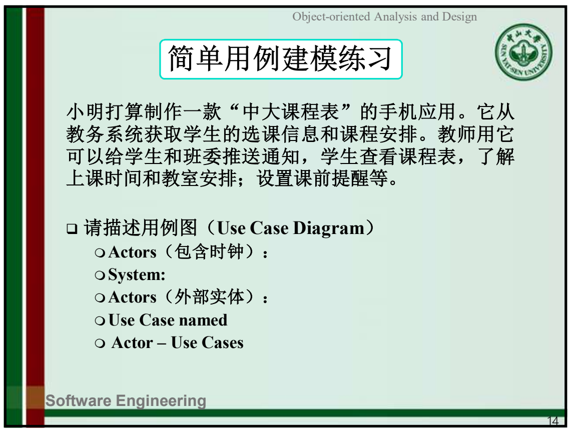
★

Use Case Modeling

- Use case model
  - Be the set of all written use cases; it is a model of the **black-box** system's functionality and environment.
  - be **not the only requirement** artifact in the UP. There are also the Supplementary Specification, Glossary, Vision, and Business Rules.
  - may optionally include a UML use case diagram to show the names of use cases and actors, and their relationships. This gives **a nice context diagram of a system and its environment.**



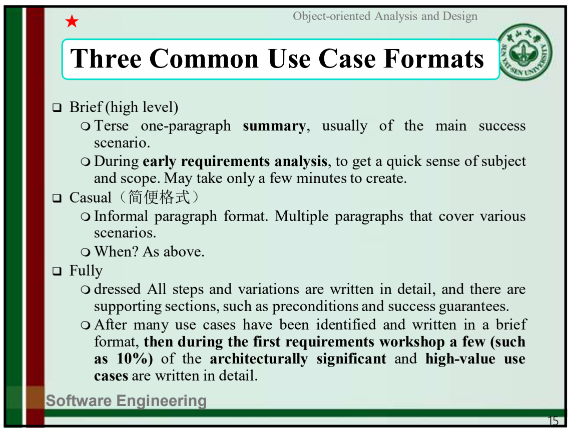
Use case Diagrams



简单用例建模练习

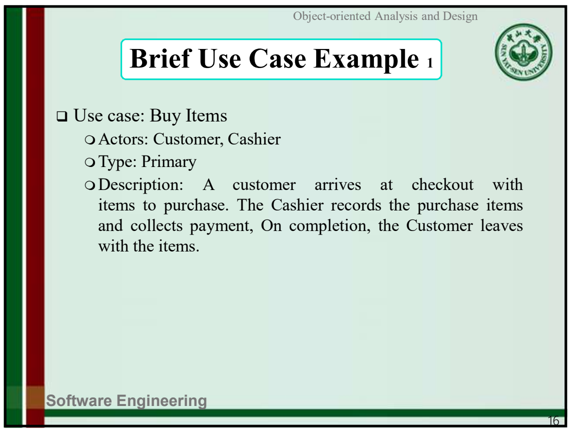
小明打算制作一款“中大课程表”的手机应用。它从教务系统获取学生的选课信息和课程安排。教师用它可以给学生和班委推送通知，学生查看课程表，了解上课时间和教室安排；设置课前提醒等。

- 请描述用例图（Use Case Diagram）
  - **Actors（包含时钟）：**
  - **System:**
  - **Actors（外部实体）：**
  - **Use Case named**
  - **Actor – Use Cases**


★

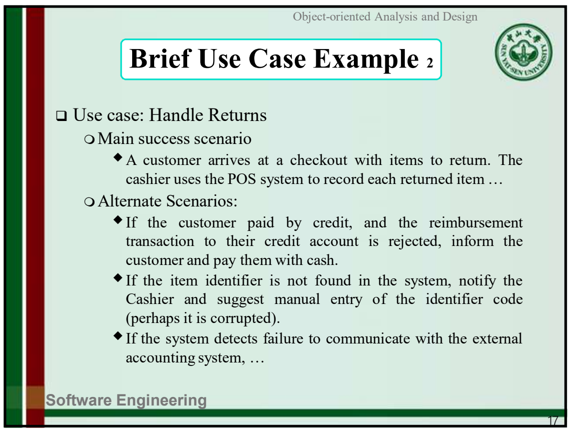
Three Common Use Case Formats

- Brief (high level)
  - Terse one-paragraph **summary**, usually of the main success scenario.
  - During **early requirements analysis**, to get a quick sense of subject and scope. May take only a few minutes to create.
- Casual（简便格式）
  - Informal paragraph format. Multiple paragraphs that cover various scenarios.
  - When? As above.
- Fully
  - dressed All steps and variations are written in detail, and there are supporting sections, such as preconditions and success guarantees.
  - After many use cases have been identified and written in a brief format, **then during the first requirements workshop a few (such as 10%)** of the **architecturally significant** and **high-value use cases** are written in detail.



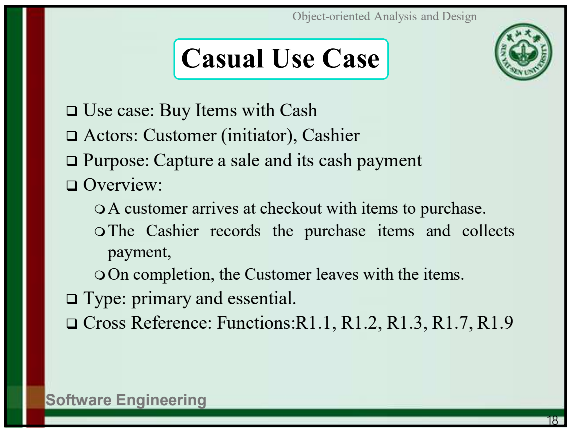
Brief Use Case Example 1

- Use case: Buy Items
  - Actors: Customer, Cashier
  - Type: Primary
  - Description: A customer arrives at checkout with items to purchase. The Cashier records the purchase items and collects payment, On completion, the Customer leaves with the items.



Brief Use Case Example 2

- Use case: Handle Returns
  - Main success scenario
    - A customer arrives at a checkout with items to return. The cashier uses the POS system to record each returned item …
  - Alternate Scenarios:
    - If the customer paid by credit, and the reimbursement transaction to their credit account is rejected, inform the customer and pay them with cash.
    - If the item identifier is not found in the system, notify the Cashier and suggest manual entry of the identifier code (perhaps it is corrupted).
    - If the system detects failure to communicate with the external accounting system, …



Casual Use Case

- Use case: Buy Items with Cash
- Actors: Customer (initiator), Cashier
- Purpose: Capture a sale and its cash payment
- Overview:
  - A customer arrives at checkout with items to purchase.
  - The Cashier records the purchase items and collects payment,
  - On completion, the Customer leaves with the items.
- Type: primary and essential.
- Cross Reference: Functions:R1.1, R1.2, R1.3, R1.7, R1.9

Use Case Scenario: Buy Items 1

1. When a Customer arrives at the POS Terminal checkout with items to purchase.
2. The Cashier records each items. If there is more than one of an item, the Cashier can enter the quantity as well.
3. The system determines the item price and adds the item information to running sales transaction. The description and price of the current item are presented.
4. On completion of item entry, the Cashier indicates to the POS Terminal that item entry is complete.
5. The system calculates and presents the sale total.
6. The Cashier tells the Customer the total.

Use Case Scenario: Buy Items 2

7. Customer choose payment type: If cash payment, see section ***Pay by Cash***. If credit payment, see section ***Pay by Credit***.
8. The system logs the complete sale.
9. The system updates inventory.
10. The system generates a receipt.
11. The Cashier gives the receipt to the Customer.
12. The Customer leaves with the items purchases.

Variation

2.1. If invalid item identifier entered, indicate error.

7.1. If Customer could not pay, cancel sales transaction.

Use Case Scenario: Pay by Cash

1. The Customer gives a cash payment – the cash tendered possibly greater than the sale total.
2. The Cashier records the tendered.
3. The system presents the balance due back to the Customer.
4. The Cashier deposits the cash received and extracts the balance owing.
5. The Cashier gives the balance owing to the Customer.

Variation

1.1. If customer does not have sufficient cash, Cashier may cancel sale or initiate another payment method.

4.1. If cash drawer does not contain sufficient cash to pay balance, Cashier requests additional cash from supervisor or asks Customer for different payment amount or method.

Use Case Scenario: Pay by Credit

1. The Customer communicates their credit information for the credit payment.
2. The system generates a credit payment request and send it to an external Credit Authorization Service (CAS).
3. Credit Authorization Service authorizes the payment.
4. The system receives a credit approval reply from the CAS.
5. The system posts (records) the credit payment and approval reply information to the Account Receivable system.

Variation

3.1. If credit request denied by CAS, Cashier suggests different payment method.

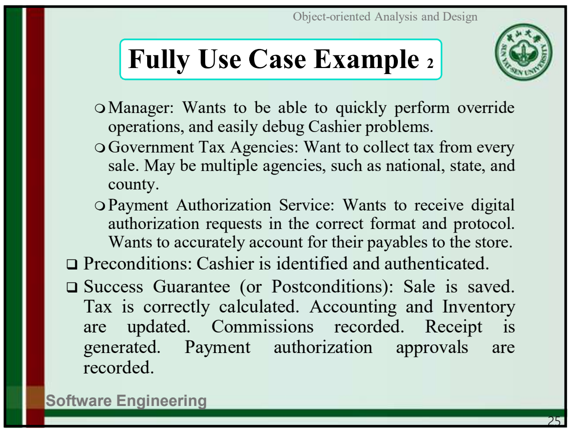
★

Fully Use Case Template

Use Case Section	Comment
Use Case Name	Start with a verb.
Scope	The system under design.
Level	"user-goal" or "subfunction"
Primary Actor	Calls on the system to deliver its services.
Stakeholders and Interests	Who cares about this use case, and what do they want?
Preconditions	What must be true on start, and worth telling the reader?
Success Guarantee	What must be true on successful completion, and worth telling the reader.
Main Success Scenario	A typical, unconditional happy path scenario of success.
Extensions	Alternate scenarios of success or failure.
Special Requirements	Related non-functional requirements.
Technology and Data Variations List	Varying I/O methods and data formats.
Frequency of Occurrence	Influences investigation, testing, and timing of implementation.
Miscellaneous	Such as open issues.

Fully Use Case Example 1

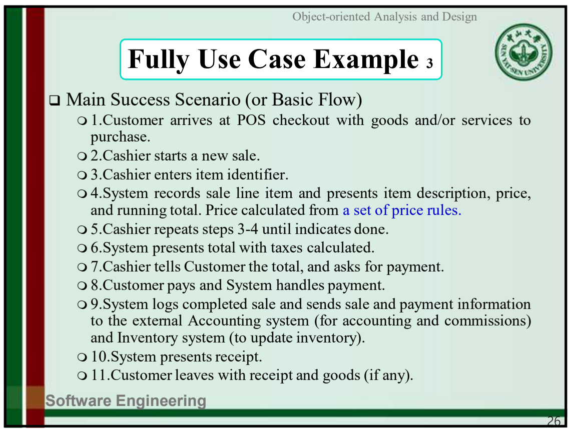
- Scope: NextGen POS application
- Level: user goal
- Primary Actor: Cashier
- Stakeholders and Interests:
  - Cashier: Wants accurate, fast entry, and no payment errors, as cash drawer shortages are deducted from his/her salary.
  - Salesperson: Wants sales commissions updated.
  - Customer: Wants purchase and fast service with minimal effort. Wants easily visible display of entered items and prices. Wants proof of purchase to support returns.
  - Company: Wants to accurately record transactions and satisfy customer interests. Wants to ensure that Payment Authorization Service payment receivables are recorded. Wants some fault tolerance to allow sales capture even if server components (e.g., remote credit validation) are unavailable. Wants automatic and fast update of accounting and inventory.



Fully Use Case Example 2

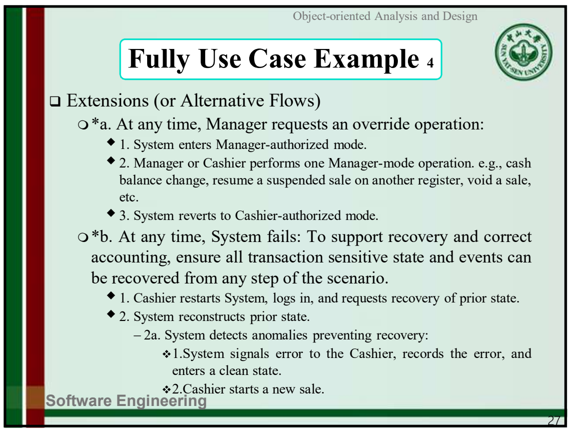
- Manager: Wants to be able to quickly perform override operations, and easily debug Cashier problems.
- Government Tax Agencies: Want to collect tax from every sale. May be multiple agencies, such as national, state, and county.
- Payment Authorization Service: Wants to receive digital authorization requests in the correct format and protocol. Wants to accurately account for their payables to the store.

- Preconditions: Cashier is identified and authenticated.
- Success Guarantee (or Postconditions): Sale is saved. Tax is correctly calculated. Accounting and Inventory are updated. Commissions recorded. Receipt is generated. Payment authorization approvals are recorded.



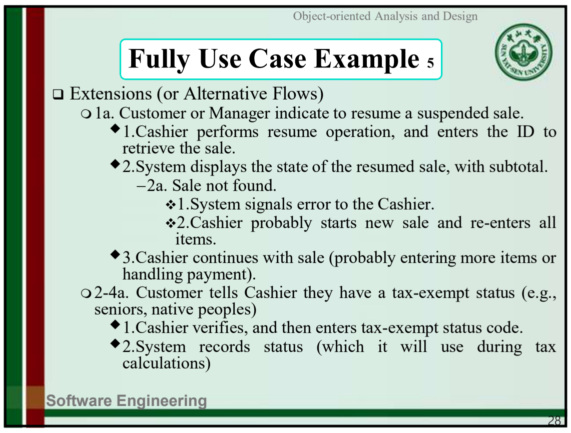
Fully Use Case Example 3

- Main Success Scenario (or Basic Flow)
  - 1.Customer arrives at POS checkout with goods and/or services to purchase.
  - 2.Cashier starts a new sale.
  - 3.Cashier enters item identifier.
  - 4.System records sale line item and presents item description, price, and running total. Price calculated from a set of price rules.
  - 5.Cashier repeats steps 3-4 until indicates done.
  - 6.System presents total with taxes calculated.
  - 7.Cashier tells Customer the total, and asks for payment.
  - 8.Customer pays and System handles payment.
  - 9.System logs completed sale and sends sale and payment information to the external Accounting system (for accounting and commissions) and Inventory system (to update inventory).
  - 10.System presents receipt.
  - 11.Customer leaves with receipt and goods (if any).



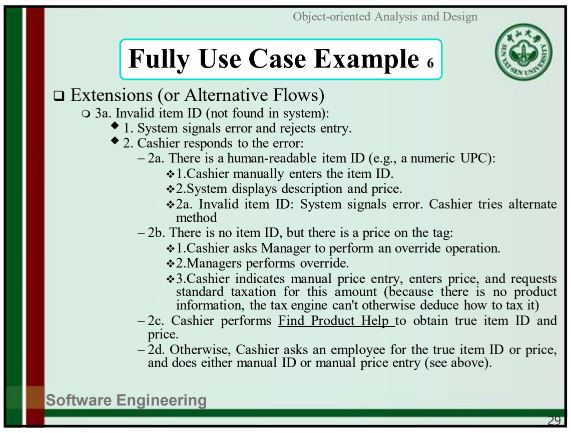
Fully Use Case Example 4

- Extensions (or Alternative Flows)
  - *a. At any time, Manager requests an override operation:
    - 1. System enters Manager-authorized mode.
    - 2. Manager or Cashier performs one Manager-mode operation. e.g., cash balance change, resume a suspended sale on another register, void a sale, etc.
    - 3. System reverts to Cashier-authorized mode.
  - *b. At any time, System fails: To support recovery and correct accounting, ensure all transaction sensitive state and events can be recovered from any step of the scenario.
    - 1. Cashier restarts System, logs in, and requests recovery of prior state.
    - 2. System reconstructs prior state.
      - 2a. System detects anomalies preventing recovery:
        - 1.System signals error to the Cashier, records the error, and enters a clean state.
        - 2.Cashier starts a new sale.



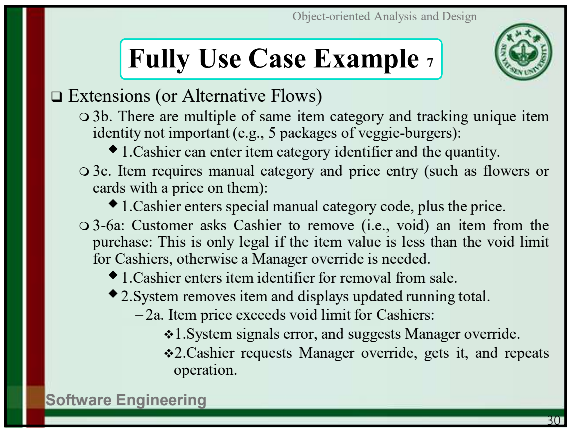
Fully Use Case Example 5

- Extensions (or Alternative Flows)
  - 1a. Customer or Manager indicate to resume a suspended sale.
    - 1.Cashier performs resume operation, and enters the ID to retrieve the sale.
    - 2.System displays the state of the resumed sale, with subtotal.
      - 2a. Sale not found.
        - 1.System signals error to the Cashier.
        - 2.Cashier probably starts new sale and re-enters all items.
    - 3.Cashier continues with sale (probably entering more items or handling payment).
  - 2-4a. Customer tells Cashier they have a tax-exempt status (e.g., seniors, native peoples)
    - 1.Cashier verifies, and then enters tax-exempt status code.
    - 2.System records status (which it will use during tax calculations)



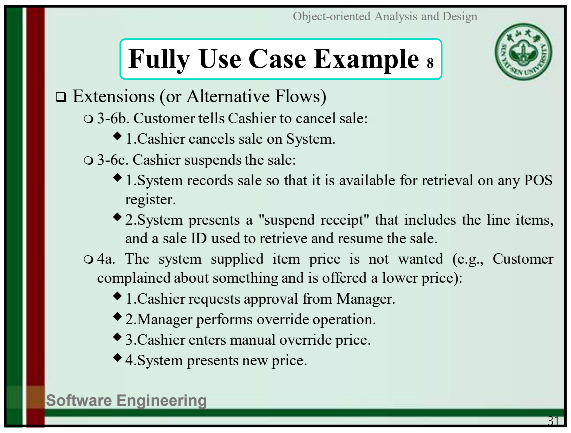
Fully Use Case Example 6

- Extensions (or Alternative Flows)
  - 3a. Invalid item ID (not found in system):
    - 1. System signals error and rejects entry.
    - 2. Cashier responds to the error:
      - 2a. There is a human-readable item ID (e.g., a numeric UPC):
        - 1.Cashier manually enters the item ID.
        - 2.System displays description and price.
        - 2a. Invalid item ID: System signals error. Cashier tries alternate method
      - 2b. There is no item ID, but there is a price on the tag:
        - 1.Cashier asks Manager to perform an override operation.
        - 2.Managers performs override.
        - 3.Cashier indicates manual price entry, enters price, and requests standard taxation for this amount (because there is no product information, the tax engine can't otherwise deduce how to tax it)
      - 2c. Cashier performs Find Product Help to obtain true item ID and price.
      - 2d. Otherwise, Cashier asks an employee for the true item ID or price, and does either manual ID or manual price entry (see above).



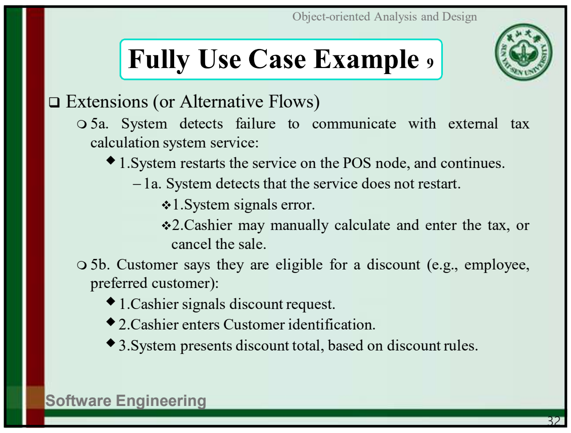
Fully Use Case Example 7

- Extensions (or Alternative Flows)
  - 3b. There are multiple of same item category and tracking unique item identity not important (e.g., 5 packages of veggie-burgers):
    - 1.Cashier can enter item category identifier and the quantity.
  - 3c. Item requires manual category and price entry (such as flowers or cards with a price on them):
    - 1.Cashier enters special manual category code, plus the price.
  - 3-6a: Customer asks Cashier to remove (i.e., void) an item from the purchase: This is only legal if the item value is less than the void limit for Cashiers, otherwise a Manager override is needed.
    - 1.Cashier enters item identifier for removal from sale.
    - 2.System removes item and displays updated running total.
      - 2a. Item price exceeds void limit for Cashiers:
        - 1.System signals error, and suggests Manager override.
        - 2.Cashier requests Manager override, gets it, and repeats operation.



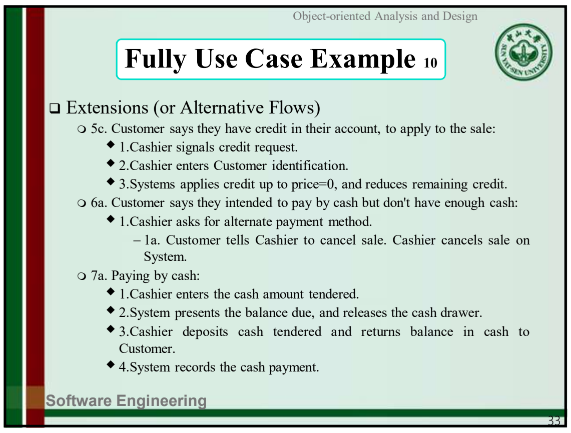
Fully Use Case Example 8

- Extensions (or Alternative Flows)
  - 3-6b. Customer tells Cashier to cancel sale:
    - 1.Cashier cancels sale on System.
  - 3-6c. Cashier suspends the sale:
    - 1.System records sale so that it is available for retrieval on any POS register.
    - 2.System presents a "suspend receipt" that includes the line items, and a sale ID used to retrieve and resume the sale.
  - 4a. The system supplied item price is not wanted (e.g., Customer complained about something and is offered a lower price):
    - 1.Cashier requests approval from Manager.
    - 2.Manager performs override operation.
    - 3.Cashier enters manual override price.
    - 4.System presents new price.



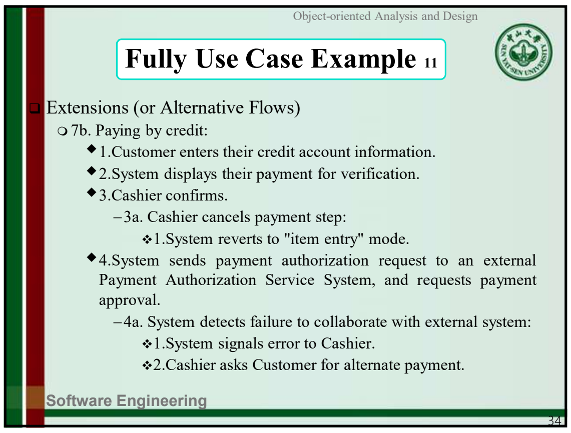
Fully Use Case Example 9

- Extensions (or Alternative Flows)
  - 5a. System detects failure to communicate with external tax calculation system service:
    - 1.System restarts the service on the POS node, and continues.
      - 1a. System detects that the service does not restart.
        - 1.System signals error.
        - 2.Cashier may manually calculate and enter the tax, or cancel the sale.
  - 5b. Customer says they are eligible for a discount (e.g., employee, preferred customer):
    - 1.Cashier signals discount request.
    - 2.Cashier enters Customer identification.
    - 3.System presents discount total, based on discount rules.



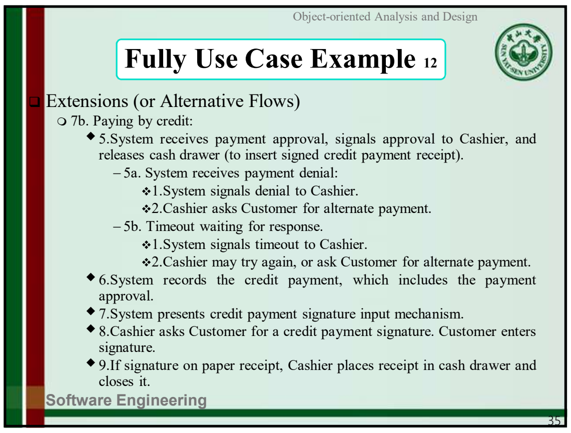
Fully Use Case Example 10

- Extensions (or Alternative Flows)
  - 5c. Customer says they have credit in their account, to apply to the sale:
    - 1.Cashier signals credit request.
    - 2.Cashier enters Customer identification.
    - 3.Systems applies credit up to price=0, and reduces remaining credit.
  - 6a. Customer says they intended to pay by cash but don't have enough cash:
    - 1.Cashier asks for alternate payment method.
      - 1a. Customer tells Cashier to cancel sale. Cashier cancels sale on System.
  - 7a. Paying by cash:
    - 1.Cashier enters the cash amount tendered.
    - 2.System presents the balance due, and releases the cash drawer.
    - 3.Cashier deposits cash tendered and returns balance in cash to Customer.
    - 4.System records the cash payment.



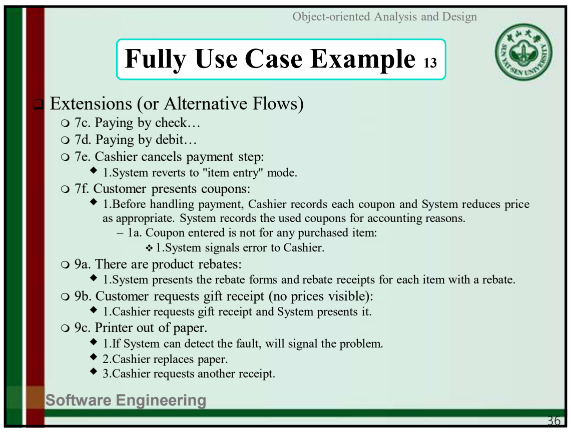
Fully Use Case Example 11

- Extensions (or Alternative Flows)
  - 7b. Paying by credit:
    - 1.Customer enters their credit account information.
    - 2.System displays their payment for verification.
    - 3.Cashier confirms.
      - 3a. Cashier cancels payment step:
        - 1.System reverts to "item entry" mode.
    - 4.System sends payment authorization request to an external Payment Authorization Service System, and requests payment approval.
      - 4a. System detects failure to collaborate with external system:
        - 1.System signals error to Cashier.
        - 2.Cashier asks Customer for alternate payment.



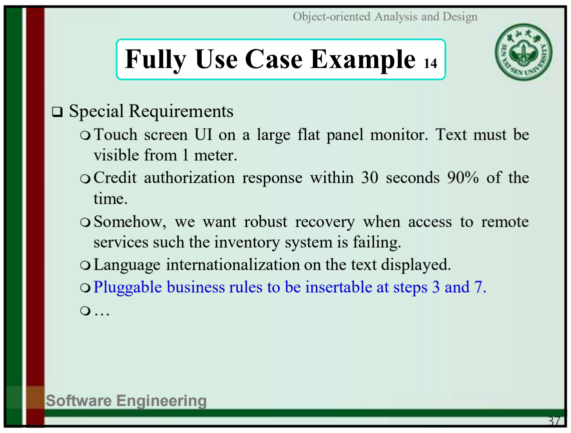
Fully Use Case Example 12

- Extensions (or Alternative Flows)
  - 7b. Paying by credit:
    - 5.System receives payment approval, signals approval to Cashier, and releases cash drawer (to insert signed credit payment receipt).
      - 5a. System receives payment denial:
        - 1.System signals denial to Cashier.
        - 2.Cashier asks Customer for alternate payment.
      - 5b. Timeout waiting for response.
        - 1.System signals timeout to Cashier.
        - 2.Cashier may try again, or ask Customer for alternate payment.
    - 6.System records the credit payment, which includes the payment approval.
    - 7.System presents credit payment signature input mechanism.
    - 8.Cashier asks Customer for a credit payment signature. Customer enters signature.
    - 9.If signature on paper receipt, Cashier places receipt in cash drawer and closes it.



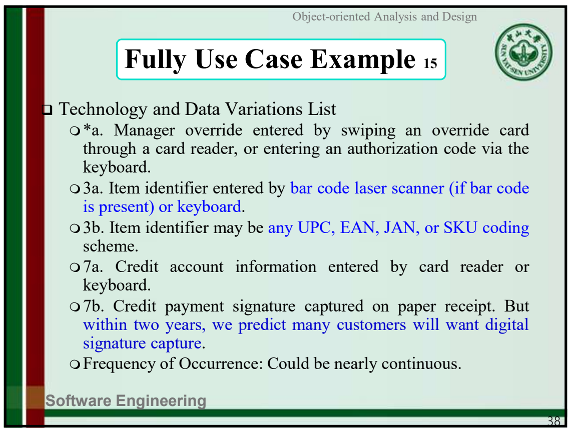
Fully Use Case Example 13

- Extensions (or Alternative Flows)
  - 7c. Paying by check…
  - 7d. Paying by debit…
  - 7e. Cashier cancels payment step:
    - 1.System reverts to "item entry" mode.
  - 7f. Customer presents coupons:
    - 1.Before handling payment, Cashier records each coupon and System reduces price as appropriate. System records the used coupons for accounting reasons.
      - 1a. Coupon entered is not for any purchased item:
        - 1.System signals error to Cashier.
  - 9a. There are product rebates:
    - 1.System presents the rebate forms and rebate receipts for each item with a rebate.
  - 9b. Customer requests gift receipt (no prices visible):
    - 1.Cashier requests gift receipt and System presents it.
  - 9c. Printer out of paper.
    - 1.If System can detect the fault, will signal the problem.
    - 2.Cashier replaces paper.
    - 3.Cashier requests another receipt.



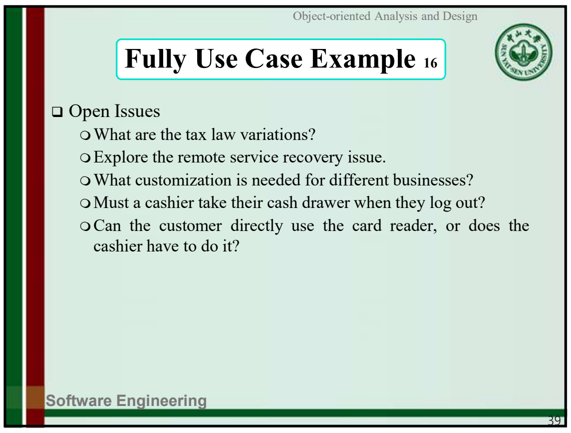
Fully Use Case Example 14

- Special Requirements
  - Touch screen UI on a large flat panel monitor. Text must be visible from 1 meter.
  - Credit authorization response within 30 seconds 90% of the time.
  - Somehow, we want robust recovery when access to remote services such the inventory system is failing.
  - Language internationalization on the text displayed.
  - Pluggable business rules to be insertable at steps 3 and 7.
  - …



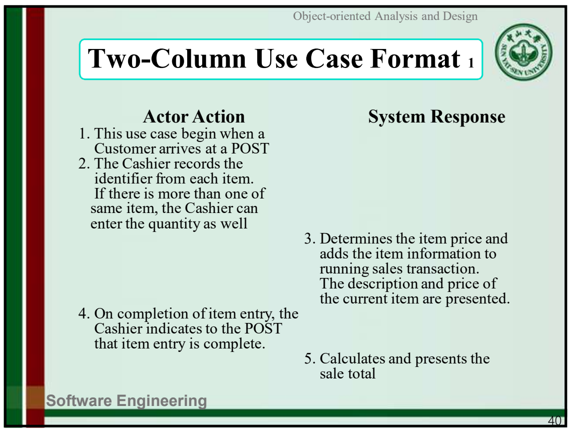
Fully Use Case Example 15

- Technology and Data Variations List
  - *a. Manager override entered by swiping an override card through a card reader, or entering an authorization code via the keyboard.
  - 3a. Item identifier entered by bar code laser scanner (if bar code is present) or keyboard.
  - 3b. Item identifier may be any UPC, EAN, JAN, or SKU coding scheme.
  - 7a. Credit account information entered by card reader or keyboard.
  - 7b. Credit payment signature captured on paper receipt. But within two years, we predict many customers will want digital signature capture.
  - Frequency of Occurrence: Could be nearly continuous.



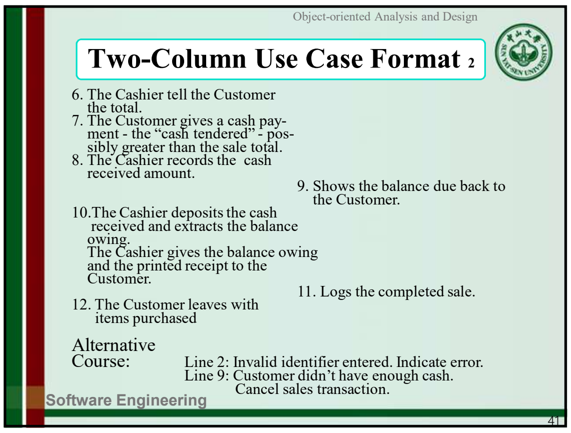
Fully Use Case Example 16

- Open Issues
  - What are the tax law variations?
  - Explore the remote service recovery issue.
  - What customization is needed for different businesses?
  - Must a cashier take their cash drawer when they log out?
  - Can the customer directly use the card reader, or does the cashier have to do it?



Two-Column Use Case Format 1

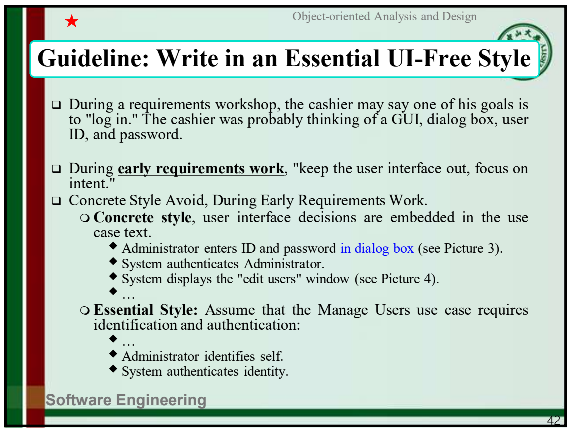
Actor Action	System Response
1. This use case begin when a Customer arrives at a POST	
2. The Cashier records the identifier from each item. If there is more than one of same item, the Cashier can enter the quantity as well	
	3. Determines the item price and adds the item information to running sales transaction. The description and price of the current item are presented.
4. On completion of item entry, the Cashier indicates to the POST that item entry is complete.	
	5. Calculates and presents the sale total



Two-Column Use Case Format 2

6. The Cashier tell the Customer the total.	
7. The Customer gives a cash payment - the "cash tendered" - possibly greater than the sale total.	
8. The Cashier records the cash received amount.	
	9. Shows the balance due back to the Customer.
10.The Cashier deposits the cash received and extracts the balance owing. The Cashier gives the balance owing and the printed receipt to the Customer.	
	11. Logs the completed sale.
12. The Customer leaves with items purchased	

Alternative Course:
Line 2: Invalid identifier entered. Indicate error.
Line 9: Customer didn't have enough cash. Cancel sales transaction.



Guideline: Write in an Essential UI-Free Style

- During a requirements workshop, the cashier may say one of his goals is to "log in." The cashier was probably thinking of a GUI, dialog box, user ID, and password.
- During **early requirements work**, "keep the user interface out, focus on intent."
- Concrete Style Avoid, During Early Requirements Work.
  - **Concrete style**, user interface decisions are embedded in the use case text.
    - Administrator enters ID and password in dialog box (see Picture 3).
    - System authenticates Administrator.
    - System displays the "edit users" window (see Picture 4).
    - …
  - **Essential Style:** Assume that the Manage Users use case requires identification and authentication:
    - …
    - Administrator identifies self.
    - System authenticates identity.

★

Guideline: Write Terse Use Cases

- Write terse use cases. Delete "noise" words.
  - Even small changes add up, such as "System authenticates…" rather than "**The** System authenticates…"

★

Guideline: Write Black-Box Use Cases

- Black-box use cases
  - do not describe the internal workings of the system, its components, or design.
- By defining system responsibilities with black-box use cases, one can specify what the system must do (the behavior or functional requirements) without deciding how it will do it (the design).
  - the definition of "analysis" versus "design" is "what" versus "how."

Black-box style	**Not**
The system records the sale.	The system writes the sale to a database. …or (even worse): The system generates a SQL INSERT statement for the sale…

★

Guideline: Take an Actor and Actor-Goal Perspective

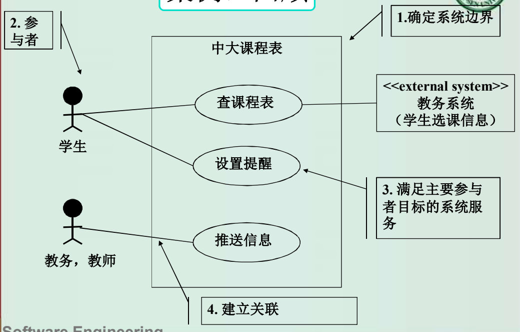
- To stresses two attitudes during requirements analysis:
  - Write requirements focusing on the users or actors of a system, asking about their goals and typical situations.
  - Focus on understanding what the actor considers a valuable result.

★★★

Guideline: How to Find Use Cases 1

- Use cases are defined to satisfy the goals of the primary actors. The basic procedure is:
  - **1.** Choose the **system boundary**.
    - Is it just a software application, the hardware and application as a unit, that plus a person using it, or an entire organization?
  - **2.** Identify the primary **actors**
    - that have goals fulfilled through using services of the system.
  - **3.** Identify **the goals** for each primary actor.
  - **4.** Define use cases that satisfy user goals; name them according to their goal.
    - user-goal level use cases will be **one-to-one** with user goals, but there is at least one exception, as will be examined.

案例：回顾

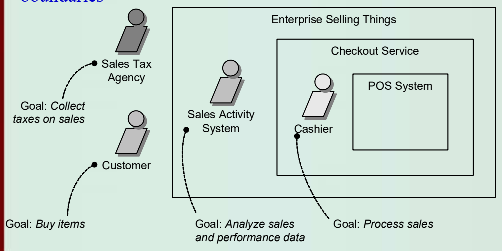


Guideline: How to Find Use Cases 2

- Questions to Help Find Actors and Goals?
  - Who starts and stops the system?
  - Who does system administration?
  - Who does user and security management?
  - Is "time" an actor because the system does something in response to a time event?
  - Is there a monitoring process that restarts the system if it fails?
  - Who evaluates system activity or performance?
  - How are software updates handled? Push or pull update?
  - Who evaluates logs? Are they remotely retrieved?
  - In addition to human primary actors, are there any external software or robotic systems that call upon services of the system?
  - Who gets notified when there are errors or failures?

Guideline: How to Find Use Cases 4

- Case study: Primary actors and goals at different system boundaries



Guideline: How to Find Use Cases 3

- Organize the Actors and Goals
  - Case study

Cashier	process sales process rentals handle returns cash in cash out	System Administrator	add users modify users delete users manage security manage system tables …
Manager	start up shut down …	Sales Activity System	analyze sales and performance data

Find Actors and Goals: Event Analysis

- **Event Based**
  - to identify external events that a system must respond to.
  - . What are they, where from, and why? Often, a group of events belong to the same use case.
  - Relate the events to actors and use cases.

External Event	From Actor	Goal/Use Case
enter sale line item	Cashier	process a sale
enter payment	Cashier or Customer	process a sale

Guideline: What Tests Can Help Find Useful Use Cases (1)

- "What is a useful level to express use cases for application requirements analysis?" rules of thumb, including:
  - The Boss Test
    - Your boss asks, "What have you been doing all day?" You reply: "Logging in!" Is your boss happy?
    - If not, the use case fails the Boss Test, which implies it is not strongly related to achieving results of measurable value.
  - The EBP Test
  - The Size Test

Guideline: What Tests Can Help Find Useful Use Cases (2)

- The EBP Test
  - Elementary Business Process (EBP) is a term from the business process engineering field
  - EBP is similar to the term **user task** in usability engineering
  - A task performed by one person in one place at one time, in response to a business event, which adds **measurable business value** and leaves the data in a consistent state, e.g., Approve Credit or Price Order
- The Size Test
  - A common mistake in use case modeling is to define **just a single step** within a series of related steps as a use case by itself, such as defining a use case called **Enter an Item ID**.

Guideline: What Tests Can Help Find Useful Use Cases (3)

- **Example: Applying the Tests**
  - Negotiate a Supplier Contract
    - Much broader and longer than an EBP. Could be modeled as a business use case, rather than a system use case.
  - Handle Returns
    - OK with the boss. Seems like an EBP. Size is good.
  - Log In
    - Boss not happy if this is all you do all day!
  - Move Piece on Game Board
    - Single step, fails the size test.

Motivation: Other Benefits of Use Cases

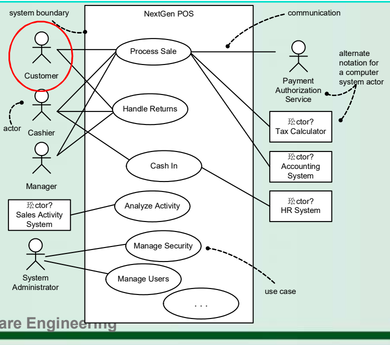
- To replace detailed, low-level function lists with use cases
- High-Level System Feature Lists Are Acceptable
  - a terse, high-level feature list, called system features, added to a Vision document can usefully summarize system functionality.
  - Summary of System Features, for **POS.**
    - sales capture
    - payment authorization (credit, debit, check)
    - system administration for users, security, code and constants tables, and so on
    - …

Applying UML: Use Case Diagrams 1

- Draw a simple use case diagram in conjunction with an actor-goal list.
- Use case diagram is an excellent picture of the system context;
  - it makes a good context diagram,
  - showing the boundary of a system, what lies outside of it, and how it gets used.
  - It serves as a communication tool that summarizes the behavior of a system and its actors.
- Guideline: Downplay Diagramming, **Keep it Short and Simple**.
- If an organization is spending many hours/days working on a use case diagram and discussing use case relationships, rather than focusing on writing text, effort has been misplaced.

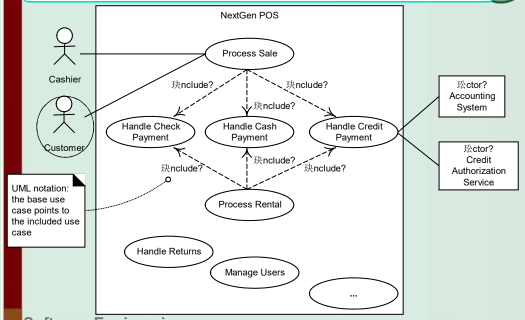
★★★

Applying UML: Use Case Diagrams 2

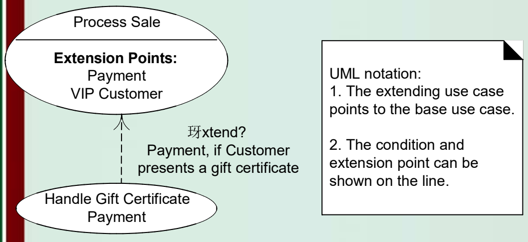


★★★

Applying UML: Use Case Diagrams 3



Applying UML: Use Case Diagrams 4


Process Sale
Extension Points:
Payment
VIP Customer
玬xtend?
Payment, if Customer
presents a gift certificate
Handle Gift Certificate
Payment
UML notation:
1. The extending use case points to the base use case.
2. The condition and extension point can be shown on the line.

案例研究：自助存包机1

Automated bar-code based lockers system for supermarket

案例研究：自助存包机2

- 系统边界
  - 存包机？柜子？锁？条码扫描机？联网控制软件？
- 主要参与者
  - 用户？管理员？联网控制软件？开门条码？
- 主要参与者的目标
  - 存包？取包？开门？关门？检查箱柜？
- 用例
- 存取常见的意外场景？
- 用户存包用例的前置条件？后置条件？
- 存包过程中用户关注哪些状态或事物？
- 用例图？

案例研究：淘宝 vs. 京东

- 某团队计划“山寨”购物网站。他们列出了客户使用淘宝和京东的部分行为（事件）：
  - 购买商品、查找商品、查看商品明细、咨询销售人员、加入购物车、管理购物车、立即购买、生成订单、管理收件人列表、在线付款、……
- 请使用前面的判断用例的方法，回答：
  - 上述哪些是淘宝／京东客户的user-goal级别的用例？
  - 上述哪些是子用例级别的用例？
  - 如何从用例或用例描述中体现b2c与c2c的差异？请举例说明

Work With Use Cases in Iterative Methods

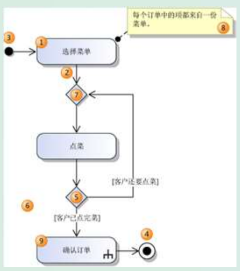
- UP encourages use-case driven development.
  - Functional requirements are primarily recorded in use cases
  - Use cases are an important part of iterative planning. The work of an iteration is in part defined by choosing some use case scenarios, or entire use cases. And use cases are a key input to estimation.
  - Use-case realizations drive the design. The team designs collaborating objects and subsystems in order to perform or realize the use cases.
  - Use cases often influence the organization of user manuals.
  - Functional or system testing corresponds to the scenarios of use cases.
  - UI "wizards" or shortcuts may be created for the most common scenarios of important use cases to ease common tasks.

Chap 6X1 用例可视化－UML活动图

使用UML活动图

- UML活动图的应用场合
  - 描述某一用例中执行的步骤，使复杂的多场景用例以及与Include或extend用例的关系可视化。
  - 描述用户和系统之间的业务流程协作。
  - 描述软件中的方法、函数或操作。（描述算法）
- UML活动图的必要性
  - 需求调查现场，与用户高效沟通
  - 描述用例实现过程。项目早期，编写完整Use Case文本太耗费时间，你可以使用活动图将重要的场景描述出来
  - 用业务流程图来识别用例。复杂业务活动包含很多用例，他们与过程关系用户难以理解。

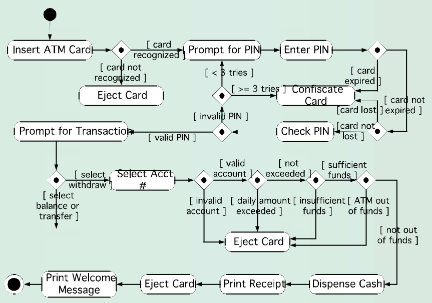
UML活动图基本符号



	符号解释
1	重要操作／子用例
2	控制流
3	起始节点：必须，唯一
4	中止节点：必须
5	决策节点
6	限定条件：［条件］
7	归并节点
8	注释：前置／后置条件；支持系统；用户感兴趣的各种需求；问题

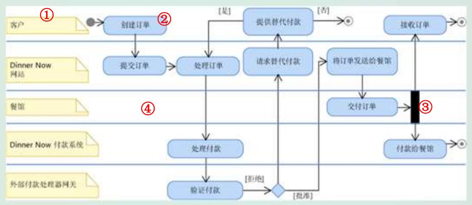
更多细节：
UML软件设计图绘制准则（微软）

ATM Withdraw Activities Diagram



UML活动图用于业务建模

① 参与者 / 设计的系统
② 责职 / 用例 / 子用例 注意：业务建模一般不要细到操作级别。



③ 并行，Fork操作
④ 泳道，表示该泳道上的活动都是其上参与者的责职

Evolve Use Cases Across the Iterations 1

Discipline	**Artifact**	Incep 1 week	Elab 1 4 weeks
Requirements	Use-Case Model	2-day requirements workshop. Most use cases identified by name, and summarized in a short paragraph. Pick 10% from the high-level list to analyze and write in detail. This 10% will be the most architecturally important, risky, and high-business value.	Near the end of this iteration, host a 2-day requirements workshop. Obtain insight and feedback from the implementation work, then complete 30% of the use cases in detail.

Evolve Use Cases Across the Iterations 2

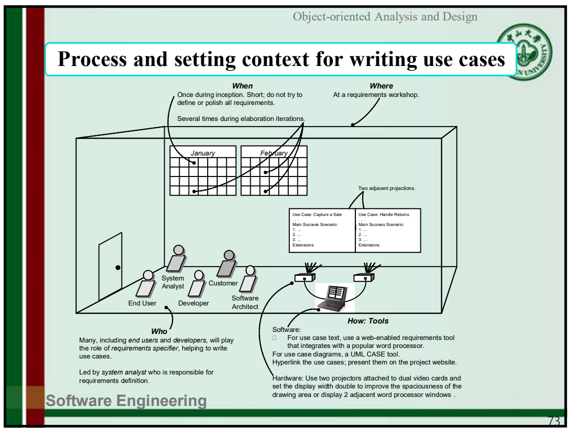
Elab 2 **4 weeks**	Elab 3 3 weeks	Elab 4 3 weeks
Near the end of this iteration, host a 2-day requirements workshop. Obtain insight and feedback from the implementation work, then complete 50% of the use cases in detail.	Repeat, complete 70% of all use cases in detail.	Repeat with the goal of 8090% of the use cases clarified and written in detail. Only a small portion of these have been built in elaboration; the remainder are done in construction.

Evolve Use Cases Across the Iterations 3

Discipline	**Artifact**	Incep 1 week	Elab 1 4 weeks	Elab 2 4 weeks	Elab 3 3 weeks	Elab 4 3 weeks
Design	Design Model	none	Design for a small set of high-risk architecturally significant requirements.	repeat	repeat	Repeat. The high risk and architecturally significant aspects should now be stabilized.
Implementation	Implementation Model (code, etc.)	none	Implement these.	Repeat. 5% of the final system is built.	Repeat. 10% of the final system is built.	Repeat. 15% of the final system is built.

Evolve Use Cases Across the Iterations 4

Discipline	**Artifact**	Incep 1 week	Elab 1 4 weeks	Elab 2 4 weeks	Elab 3 3 weeks	Elab 4 3 weeks
Project Management	SW Development Plan	Very vague estimate of total effort.	Estimate starts to take shape.	a little better …	a little better …	Overall project duration, major milestones, effort, and cost estimates can now be rationally committed to.



Process and setting context for writing use cases

When
Once during inception. Short; do not try to define or polish all requirements.

Several times during elaboration iterations.

Where
At a requirements workshop.

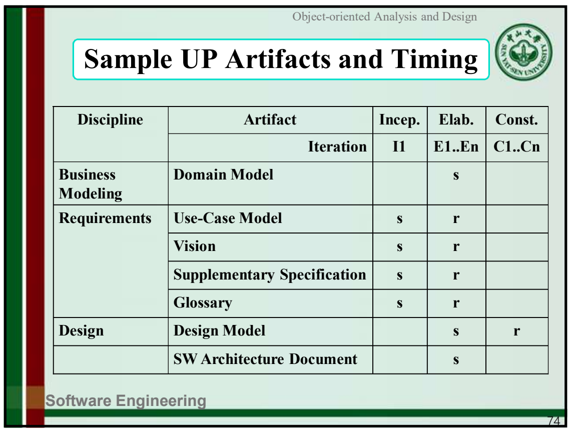
Who
Many, including *end users* and *developers,* will play the role of *requirements specifier*, helping to write use cases.

Led by *system analyst* who is responsible for requirements definition.

How: Tools
Software:
- For use case text, use a web-enabled requirements tool that integrates with a popular word processor.

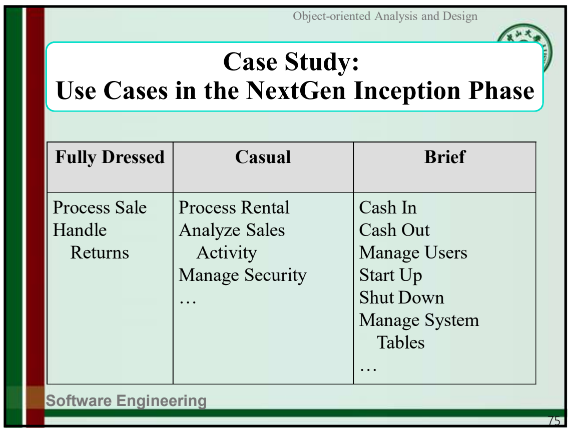
For use case diagrams, a UML CASE tool.
Hyperlink the use cases; present them on the project website.

Hardware: Use two projectors attached to dual video cards and set the display width double to improve the spaciousness of the drawing area or display 2 adjacent word processor windows .



Sample UP Artifacts and Timing

Discipline	Artifact	Incep.	Elab.	Const.
	Iteration	I1	E1..En	C1..Cn
Business Modeling	Domain Model		s	
Requirements	Use-Case Model	s	r	
	Vision	s	r	
	Supplementary Specification	s	r	
	Glossary	s	r	
Design	Design Model		s	r
	SW Architecture Document		s	



Case Study: Use Cases in the NextGen Inception Phase

Fully Dressed	Casual	Brief
Process Sale Handle Returns	Process Rental Analyze Sales Activity Manage Security …	Cash In Cash Out Manage Users Start Up Shut Down Manage System Tables …