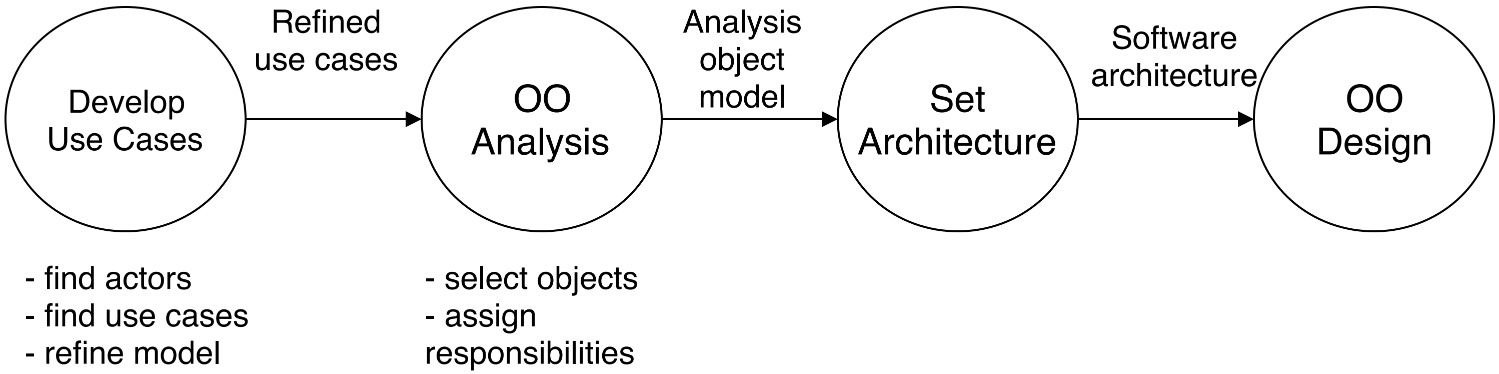
CIS 422/522

Problem Analysis with Use Cases

Why document?

Use Cases

- Often done as a prelude to OO modeling
- A form of *User Centered Analysis* – capturing requirements from the user's point of view
  - Goal of helping identify user needs
  - Solve the right problem
- Best fit with operational requirements (mission statement, ConOps, BRD, etc.)



Scenario Analysis Process

Applying scenario analysis in the requirements process

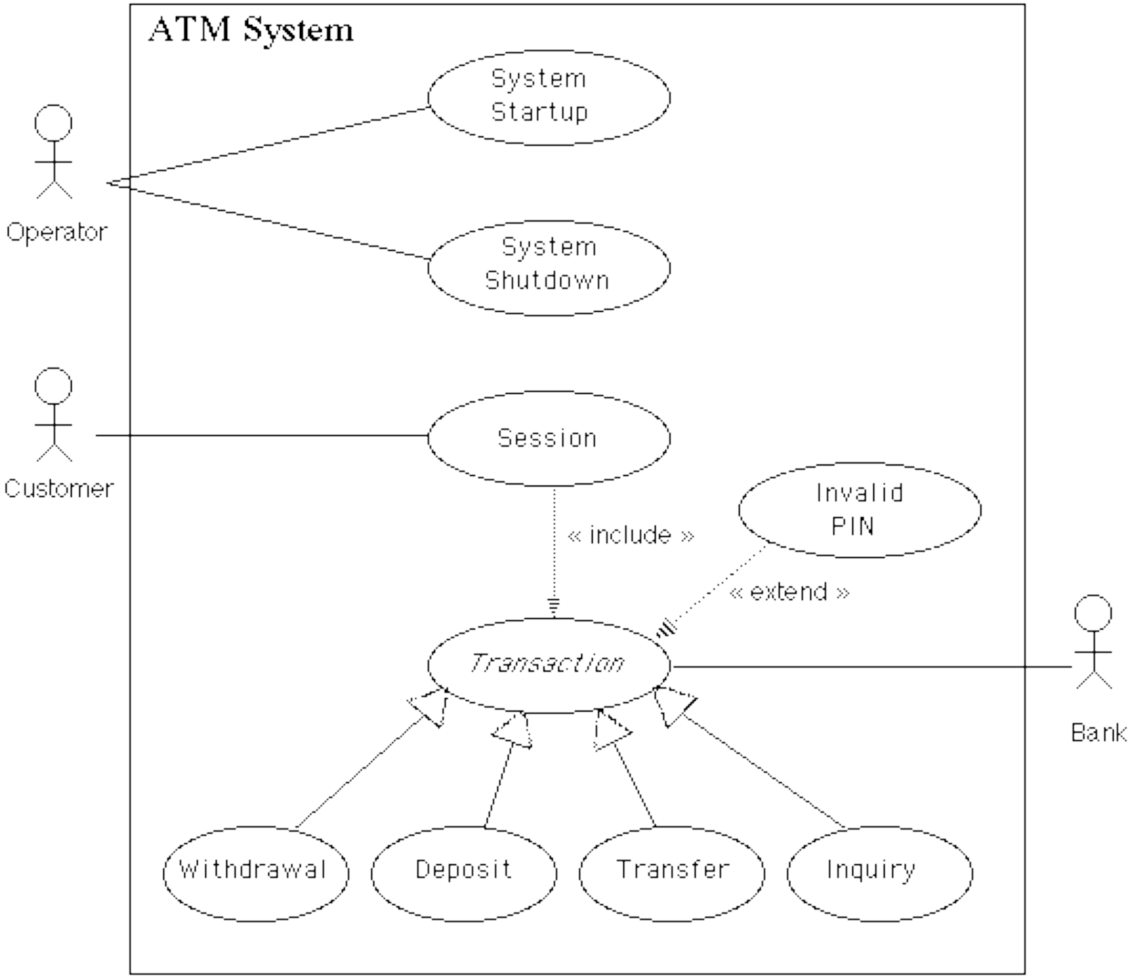
- Requirements Elicitation
  - Identify stakeholders who interact with the system
  - Collect "user stories" - how people would interact with the system to perform specific tasks
- Requirements Specification
  - Record as use-cases with standard format
  - Use templates to standardize, drive elicitation
- Requirements verification and validation
  - Review use-cases for consistency, completeness, user acceptance
  - Apply to support prototyping
  - Verify against code (e.g., use-case based testing)

Identifying Actors

- Actors – identifies a role different users plays with respect to the system
  - Roles represent different classes of users (use the system with different goals)
  - Actors carry out use cases
- Primarily useful in identifying different kinds of use cases
  - "How would depositors use the system?"
  - "How would a library patron use the system?"
- Important to keep in mind that there may be several diverse classes of users with very different goals and interfaces
  - E.g., for the Marathon software have a volunteer interface and an administrator interface. For SARA, readers vs. authors or instructors
  - Also, system administration, maintenance, testing, may be viable roles

UML Graphic Example

http://www.math-cs.gordon.edu/local/courses/cs211/ATMExample/



Scenario Elicitation

- Each class of actor is interviewed and/or observed
  - How do you do task T?
  - How will the user interact with the system to do X?
- Collect in the form of “user stories”
  - Documented as scenarios (informal or standardized)
  - Identify relative priorities of tasks
  - Resolve conflicts, tradeoffs

ATM Session Use Case (Informal)

A session is started when a customer inserts an ATM card into the card reader slot of the machine. The ATM pulls the card into the machine and reads it. (If the reader cannot read the card due to improper insertion or a damaged stripe, the card is ejected, an error screen is displayed, and the session is aborted.) The customer is asked to enter his/her PIN, and is then allowed to perform one or more transactions, choosing from a menu of possible types of transaction in each case. After each transaction, the customer is asked whether he/she would like to perform another. When the customer is through performing transactions, the card is ejected from the machine and the session ends. If a transaction is aborted due to too many invalid PIN entries, the session is also aborted, with the card being retained in the machine.

The customer may abort the session by pressing the Cancel key when entering a PIN or choosing a transaction type

http://www.math-cs.gordon.edu/local/courses/cs211/ATMExample/

Terminology

- Scenario – description of a sequence of interactions between a user and the system from the user's point of view
  - What does the user see or do
  - What does the system do in response
- Use Case – a set of scenarios related by a common user *goal*
  - Goal – an objective the user is employing the system to achieve
  - Scenarios represent different possible outcomes (nominal case, error case, etc.)

1 Use Case: Manage Reports

1.1 Description

This Use Case describes operation for Creating, Saving, Deleting, Printing, Exiting and Displaying reports.

1.2 Actors

User
Project database

1.3 Triggers

Program Manager selects operations from menu.

1.4 Flow of events

1.4.1 Basic Flow

1. **User chooses desired report by selecting "Report" -> "Open" from the menu bar**
2. **System displays report to screen**
3. **User selects desired report layout using Use Case Specify Report**
4. **Steps 2 and 3 are repeated until user is satisfied**
5. **User can Save or Print report using use case Save Report or Print Report**
6. **User Exits report by selecting "Exit" from the "File" menu**

1.4.2 Alternative Flows

1.4.2.1 Create New Report

1. **User selects "Create New Report" from file menu**
2. **…**

1.4.2.2 Delete Report

……………..

1.4.3 Preconditions

etc

A more semi-systematic approach

- **Uses a standard template**
- **Easier to make sure it is complete**

Use Case Contents (Generic)

- Use case identifier
- Summary – summary of use case
- Actors – roles enacting use case
- Scenarios
  - Basic scenario – the normal case
  - Alternative scenarios – other ways to reach goal
  - Exceptions – problem scenarios
- Trigger – what causes the use case to start
- Assumptions
- Preconditions – what must be true before the interaction can occur
- Post conditions – what must be true after the interaction occurs

Basic Scenario

- Sequence when the user proceeds to his goal as system designers intend

1.Customer puts card into ATM card slot and enters PIN number.
2.Card verified and main menu presented.
3.Customer selects the transaction services menu and the corresponding menu is displayed.
4.Customer selects ``automated payment service'' and is prompted for the recipient's account number.
5.Customer enters recipient's account number.
6.Account verified and a menu with payment schedules is presented.
7……..

Exceptions

- What is the scenario if the customer enters the wrong PIN?

1. Customer puts card into ATM card slot and enters PIN number.
2. Incorrect PIN identified and error message “Incorrect PIN” and error menu displayed.
3. Customer selects the “try again” option
4……..

For Your Project

- Since this project is driven by the user interface, use cases will work for many requirements
- Apply Use Cases to describe the system’s mission from the user’s point of view
  - Answers the questions, “What is the system for?” and “How will the user use it?”
  - Should have use cases for all major system interactions
- For the “Functional Requirements” be as rigorous as you can
  - Use tables, bullets, or case-by-case behavior description
  - Capture requirements outside the user interface
    - Maintainability, extendibility

Summary

- Use cases can be an effective tool for identifying:
  - The actors,
  - Major analysis objects,
  - Interactions between actors and analysis objects, and
  - Collaborations among analysis objects
- Use case analysis is a generally applicable technique that can be done independent of OO design and implementation
- Generally not adequate detailed technical requirements
  - State history, branching, difficult to capture
  - Natural language: imprecise, ambiguous, cannot establish consistency or completeness
  - Possible exception: applications doing simple user-centric tasks with little computation (e.g., your project)

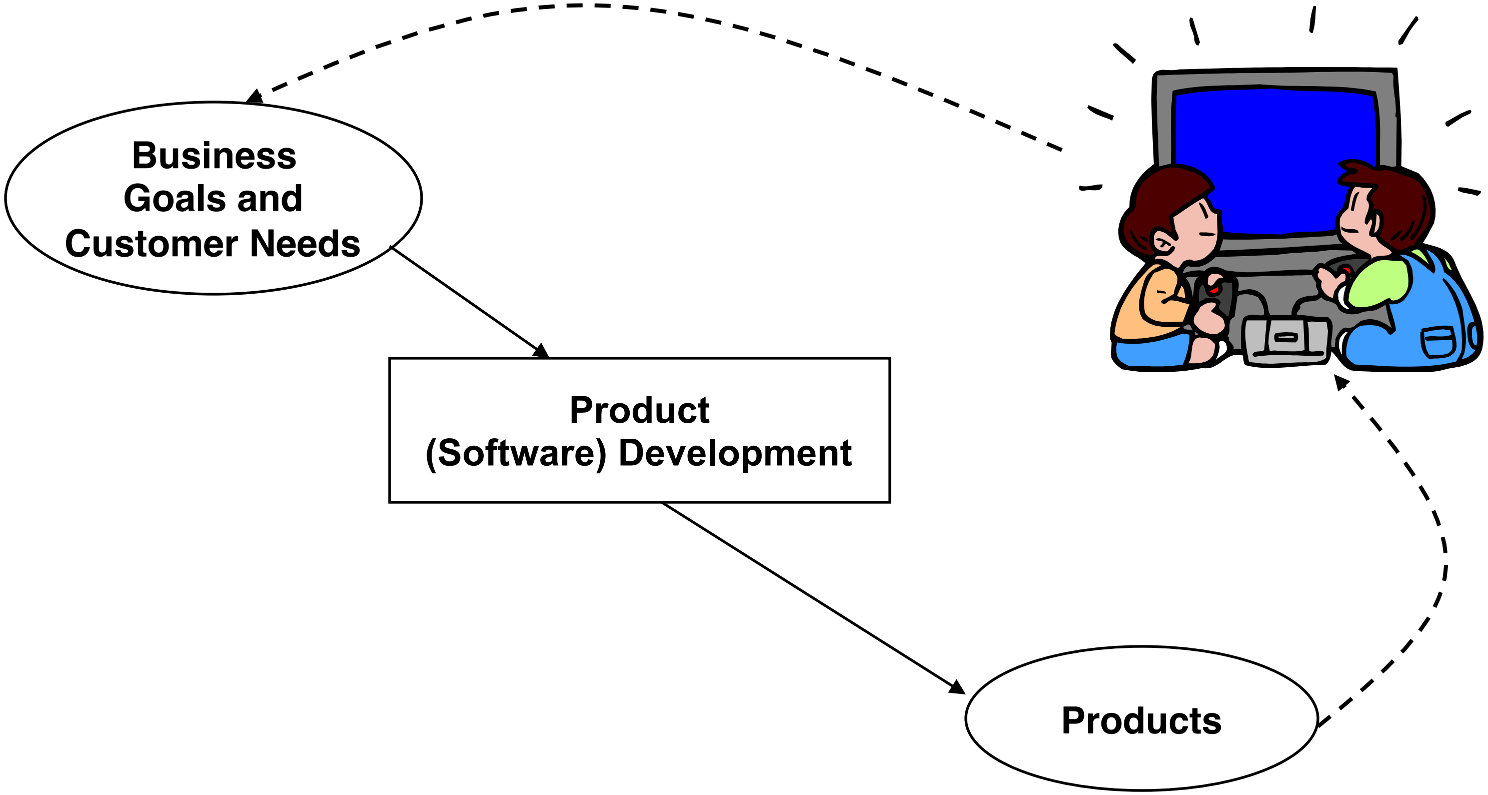
Documenting Development Decisions

Why document?

Project Documentation

- Many kinds identified
  - Project plan, schedule, meeting notes
  - Software Requirements
  - Software Architecture
  - Software documentation
- Why document (when agile methods don’t)?

10,000 ft. View



Business Goals and Customer Needs

Product (Software) Development

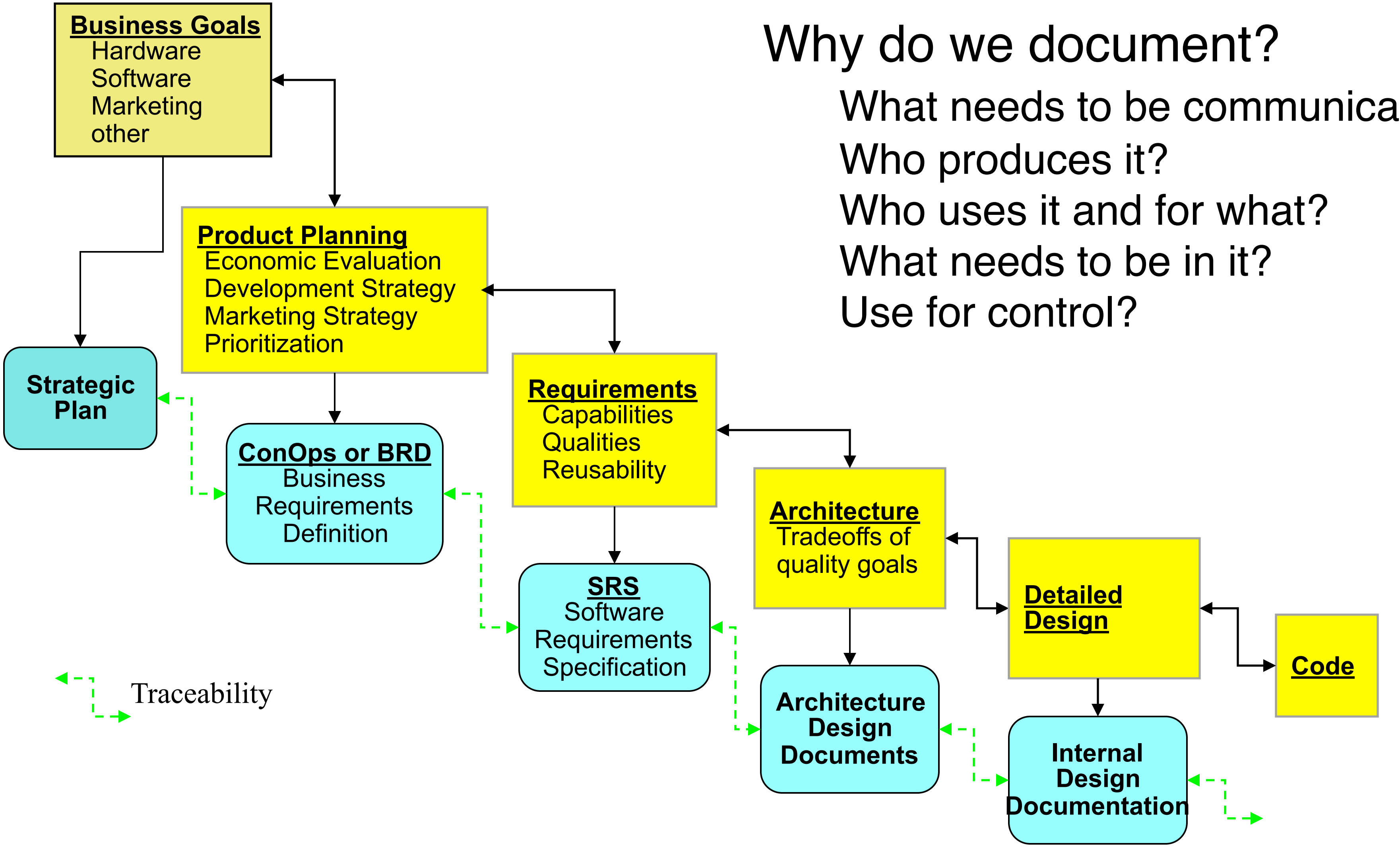
Products

What should the development process accomplish?

Role of Documentation

- To understand what kind of documentation is useful, helps to understand the "why"
- Consider
  1. Goal is to turn an idea into a product
  2. Software engineering is a decision making process
  3. We decompose a complex development process into distinct concerns (requirements, design, code, test, deployment, maintenance, etc.)
- Why document?
  - What purpose does it play?
  - What kinds of things should be documented?

Product Development Cycle and Documentation



Why do we document?

What needs to be communicated?

Who produces it?

Who uses it and for what?

What needs to be in it?

Use for control?

Document Types and Purposes

- Management documents
  - Basis for project management (managerial control of resources)
    - Calendar time, skilled man-hours budget
    - Other organizational resources
  - Project plan, WBS, Development schedule
  - Use: allows managers to track actual against expected consumption of resources
- Development documents
  - Basis for product management (intellectual control)
  - ConOps, Requirements (SRS), Architecture, Detail design, etc.
  - Uses:
    - Making and recording development decisions
    - Allows developers to track decisions from stakeholder needs to implementation

Meeting Developmental Goals Means…

- We have a clear understanding of customer needs and product goals
- External view: We develop products the customer’s wants, on time and within budget
- Internal view: We create process and product infrastructures supporting our business goals
- For most developments, these are “document” driven

Questions?

Project 1 Reports

- For Mon: prepare a 10 minute presentation with slides (practice)
- Status against project plan
  - What was planned for this date?
  - What was actually produced (status of work products and deliverables even if not complete)?
  - Demo
- Lessons learned and planned changes
  - How did the team coordinate activities and make decisions?
  - How effective was project planning?
  - What were the root causes of any schedule delays?
  - Was the risk management approach effective?
    - Were risks adequately identified and defined?
    - Did the risk mitigation strategies allow work to proceed in spite of problems?
  - Did the software decomposition into modules support distributed work in parallel?
    - Were interfaces well enough defined for developers to proceed independently?
  - Did the components work at integration time?
- What do you plan to change for Iteration 2?