System Analysis

Dr. Isaac Griffith Idaho State University

Outcomes

Idaho State University | Computer Science

After today's lecture you will be able to:

- Understand and describe the activities conducted in the analysis phase
- Understand and describe the difference between Functional and Non-Functional requirements
- Understand and be able to model the requirements of a system as a set of Use Cases
- Understand and be able to describe the concept of Business Rules

Analysis Phase Overview

Idaho State University | Computer Science

- **Typical CS Curriculum Projects**
  - Well written easy to follow program requirements
  - Requires limited design
  - Requires coding
  - Tend to be small
- **Real Life Projects**
  - Exceptionally larger in scope and size
  - Complex and ambiguous requirements
  - Large amount of money involved
  - Deadlines are typically much “harder”
- Yet in both there are still two parties: **Users** and **Developers**
- To achieve success in large projects we need a process to guide us.
- The first phase of this process is the **Analysis Phase** and it is made up of the following activities:
  1. Gather the requirements (interviews with users, documentation analysis, etc.)
  2. Precisely document the requirements
  3. Develop a conceptual model of the system

Gathering Requirements

CS 2263

Requirements

- Requirements (broadly speaking) come in two forms:
  - **Functional Requirements:** Describe the interaction between the system and its users, and between the system and any other systems, which may interact with the system by supplying or receiving data
  - **Non-functional Requirements:** Any requirement that does not fall in to the above category. For example:
    - Usability
    - Response time
    - Accuracy
    - Constraints on system development (hardware, software, budget, time)

This is the most important step in creating a software system

- **Goal:** to define what the new system should do.
- Errors made in this step
  - Compound into all following steps
  - Become more and more difficult **(and costly)** to undo as time goes on
  - **Lead to the creation of the wrong system**
- Errors occur for many reasons
  - Clients are not exactly sure of what they want
  - Clients lack the technical skill to understand what technology can/cannot deliver
  - Clients assume too much and analysts skip over “obvious details”

Requirements Analysis

- Requirements are produced by a team of analysts working with client representatives
  - Interviews
  - Surveys
  - Observations
  - Documentation analysis
- They then extract requirements into a specification

Requirements Analysis

Idaho State University | Computer Science

- Requirements are produced by a team of analysts working with client representatives
  - Interviews
  - Surveys
  - Observations
  - Documentation analysis
- They then extract requirements into a specification

But, I will save this discussion for CS 3321

Case Study

Idaho State University | Computer Science

- Throughout this lecture and the next, we will be considering the development of a basic Library Information Management system.
- This system will have the following basic features:
  - Register new Members
  - Add books to the collection
  - Issue a book to a member (or user)
  - Record the return of a book
  - Remove books from the collection
  - Print out a user's transactions
  - Place/remove a hold on a book
  - Renew books issued to a member
  - Notify member of book's availability

Requirements Specification

CS 2263

Use Case Analysis

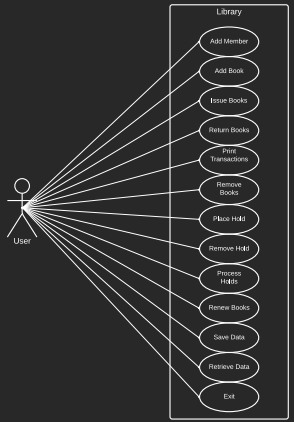
Idaho State University | Computer Science

- **Goal**: To define and document a system's requirements
  - Uses a case-based approach for describing how a system will be used
  - Each use case presents a narrative describing how a user completes a process
- Uses cases have the following components
  - Two or more parties interacting
    - **Actors**: the user who interacts with the system
    - **System**: the system itself
- The process is initiated by thinking through how the system will work with the user
  - We can normally assume that there will some form of interface (i.e., UI, CLI, Web, Mobile, etc)
  - We can then use this to consider what the user will need to do (i.e., processes) and how the system will respond

Use Cases

For the library system we could assume that some sort of menu will provide a list of choices

1. Add a member
2. Add books
3. Issue books
4. Return books
5. Remove books
6. Place a hold on a book
7. Remove a hold on a book
8. Process Holds
9. Renew books
10. Print out a member's transactions
11. Store data on disk
12. Retrieve data from disk
13. Exit



Use Cases

Idaho State University | Computer Science

- Two-column Format
  - **Left-column**: actions actors perform on the system
  - **Right-column**: responses of the system
- Aspects of use cases:
  1. Every use case is identified by a name
  2. Should represent a reasonably-sized activity in the organization
     - Not all actions and operations should be identified by a Use Case
  3. The first step specifies a “real-world” action which triggers the exchange described by the use case
     - Does not have any real bearing on the system being developed (mostly for completeness)
  4. **The use case does not specify how the functionality is to be implemented**
     - This is what design and implementation are for
  5. The use case does not cover all possible situations, only the most commonly-ocurring, or **main flow**, scenario is described
     - Errors and exceptions are not detailed

Use Case 1

Idaho State University | Computer Science

Register New Member

Actions performed by the actor	Responses from the system
1) The customer fills out an application form containing the customer's name, address, and phone number and gives this to the clerk	
2) The clerk issues a request to add a new member	
	3) The system asks for data about the new member
4) The clerk enters the data into the system	
	5) Reads in data, and if the member can be added, generates an identification number (which is not necessarily a number in the literal sense just as social security numbers and phone numbers are not actually numbers) for the member and remembers information about the member. Informs the clerk if the member was added and outputs the member's name, address, phone and id
6) The clerk gives the user his identification number	

Acceptance Tests

Idaho State University | Computer Science

- As we are writing our use cases, we should be thinking about how we can validate that the requirements encoded in the use case.
- To do this we need to create acceptance tests, which are essentially tests based on whether the user will accept that this requirement is met.
- These tests use example scenarios similar to a use case and are written using a GIVEN-WHEN-THEN format:
  - **GIVEN:** provides the staring condition of the scenario (conditions/data used to configure the test)
  - **WHEN:** an event that happens (i.e. the initial event of the use case)
  - **THEN:** the expected result as per the defined behavior of the system
- Though these three are typically all we will need, we can also add in additional lines to either the GIVEN, WHEN, or THEN using:
  - **AND**
  - **BUT**

Acceptance Test Example

Idaho State University | Computer Science

Acceptance Test: Register New Member

- **GIVEN:** A user fill in a form with the provided information

Field	Value
Name	Isaac Griffith
Address	921 S. 8th Ave, Pocatello, ID 83201
Phone	208-282-4876

- **AND:** The user has handed the form to the clerk
- **AND:** The clerk has requested to add a new user
- **WHEN:** Clerk enters this information into the system as a new user
- **THEN:** The clerk provides the user with ID Number 123

Use Case 2

Adding New Books

Actions performed by the actor	Responses from the system
1) Library receives a shipment of books from the publisher	
2) The clerk issues a request to add a new book	
	3) The system asks for the identifier, title, and author name of the book
4) The clerk generates the unique identifier, enters the identifier, title, and author name of a book	
	5) The system attempts to enter the information in the catalog and echoes to the clerk the title, author name, and id of the book. It then asks if the clerk wants to enter information about another book
6) The clerk answers in the affirmative or in the negative	
	7) If the answer is in the affirmative, the system goes to Step 3, Otherwise, it exits.

Use Case 3

Book Checkout

Actions performed by the actor	Responses from the system
1) The member arrives at the check-out counter with a set of books and supplies the clerk with his/her identification number	
2) The clerk issues a request to check out books	
	3) The system asks for the user ID
4) The clerk inputs the user ID to the system	
	5) The system asks for the ID of the book
6) The clerk inputs the ID of a book that the user wants to check out	
	7) The system records the book as having been issued to the member; it also records the member as having possession of the book. It generates a due-date. The system displays the book title and due-date and asks if there are any more books
8) The clerk stamps the due-date on the book and replies in the affirmative or negative	
	9) If there are more books, the systems moves to Step 5; otherwise it exits
10) The customer collects the books and leaves the counter	

- Issues with this use case
  - How are due dates calculated?
  - What do we do when things go wrong: (i) Person is not a member or (ii) Clerk entered invalid book ID

Business Rules

Idaho State University | Computer Science

- **Business Rules:** Often we need to have defined rules for describing how our business operates
- Including these in use cases tends to make them messy and overly complicated.
- Instead we should create a separate table, and reference these as needed.

Rule Number	Rule
Rule 1	Due-date for a book is one month from the date of issue
Rule 2	All books are issuable
Rule 3	A book is removable if it is not checked out and if it has no holds
Rule 4	A book is renewable if it has no holds on it
Rule 5	When a book with a hold is returned, the appropriate member will be notified
Rule 6	Holds can be place only on books that are currently checked out

Use Case 3 - Revised

Idaho State University | Computer Science

Book Checkout - Revised

Actions performed by the actor	Responses from the system
1) The member arrives at the check-out counter with a set of books and supplies the clerk with his/her identification number	
2) The clerk issues a request to check out books	
	3) The system asks for the user ID
4) The clerk inputs the user ID to the system	
	5) If the ID is valid, the system asks for the ID of the book; otherwise it prints an appropriate message and exits the use case
6) The clerk inputs the ID of a book that the user wants to check out	
	7) The system records the book as having been issued to the member; it also records the member as having possession of the book. It generates a due-date as in Rule 1. It then displays the book's title and due-date. If the book is not issuable as per Rule 2, the system displays a suitable error message. The system asks if there are more books
8) The clerk stamps the due-date, }}prints out}} the transaction (if needed) and replies positively or negatively	
	9) If there are more books for checking out, the system goes back to Step 5; otherwise it exits
10) The clerk stamps the due date and gives the user the books checked out. The customer leaves the counter	

Use Case 4

Idaho State University | Computer Science

Return Book

Actions performed by the actor	Responses from the system
1) The member arrives at the return counter with a set of books and leaves them on the clerk's desk	
2) The clerk issues a request to return books	
	3) The system asks for the identifier of the book
4) The clerk enters the book identifier	
	5) If the identifier is valid, the system marks that the book has been returned and informs the clerk if there is a hold placed on the book; otherwise it notifies the clerk that the identifier is not valid. It then asks if the clerk wants to process the return of another book
6) The clerk answers in the affirmative or in the negative and sets the book aside in case there is a hold on the book (see Rule 5)	
	7) If the answer is in the affirmative, the system goes to Step 3. Otherwise, it exits

Use Case 5

Removing Books

Actions performed by the actor	Responses from the system
1) Librarian identifies the books to be deleted	
2) The clerk issues a request to delete books	
	3) The system asks for the identifier of the book
4) The clerk enters the ID for the book	
	5) The system checks if the book can be removed using Rule 3. If the book can be removed, the system marks the book as no longer in the library's catalog. The system informs the clerk about the success of the deletion operation. It then asks if the clerk wants to delete another book
6) The clerk answers in the affirmative or in the negative	
	7) If the answer is in the affirmative, the system goes to Step 3. Otherwise, it exits

Use Case 6

Idaho State University | Computer Science

Member Transactions

Actions performed by the actor	Responses from the system
1) The clerk issues a request to get member transactions	
	2) The system asks for the user ID of the member and the date for which the transactions are needed
3) The clerk enters the identity of the user and the date	
	4) If the ID is valid, the system outputs information about all transactions completed by the user on the given date. For each transaction, it shows the type of transaction (book borrowed, book returned or hold placed) and the title of the book
5) Clerk prints out the transactions and hands them to the user	

Use Case 7

Idaho State University | Computer Science

Place a Hold

Actions performed by the actor	Responses from the system
1) The clerk issues a request to place a hold	
	2) The system asks for the book's ID, the ID of the member, and the duration of the hold
3) The clerk enters the identity of the user, the identity of the book and the duration	
	4) The system checks that the user and book identifiers are valid and that Rule 6 is satisfied. If yes, it records that the user has a hold on the book and displays that; otherwise, it outputs an appropriate error message

Use Case 8

Idaho State University | Computer Science

Remove a Hold

Actions performed by the actor	Responses from the system
1) The clerk issues a request to remove a hold	
	2) The system asks for the book's ID and the ID of the member
3) The clerk enters the identity of the user and the identity of the book	
	4) The system removes the hold that the user has on the book (if any such hold exists), prints a confirmation and exits

Use Case 9

Idaho State University | Computer Science

Process Holds

Actions performed by the actor	Responses from the system
1) The clerk issues a request to process holds (so that Rule 5 can be satisfied)	
	2) The system asks for the book's ID
3) The clerk enters the ID of the book	
	4) The system returns the name and phone number of the first member with an unexpired hold on the book. If all holds have expired, the system responds that there is no hold. The system then asks if there are any more books to be processed
5) If there is no hold, the book is then shelved back to its designated location in the library. Otherwise, the clerk prints out the information, places it in the book and replies in the affirmative or negative	
	6) If the answer is yes, the system goes to Step 2; otherwise, it exits

Use Case 10

Idaho State University | Computer Science

Renew Books

Actions performed by the actor	Responses from the system
1) Member makes a request to renew several of the books that he/she has currently checked out	
2) Clerk issues a request to renew books	
	3) System asks for the member's ID
4) The clerk enters the ID into the system	
	5) System checks the member's record to find out which books the member has checked out. If there are none, the system prints an appropriate message and exits; otherwise, it moves to Step 6
	6) The system displays the title of the next book checked out to the member and asks whether the book should be renewed.
7) The clerk replies yes or no	
	8) The system attempts to renew the book using Rule 4 and reports the result. If the system has displayed all checked-out books, it reports that exits; otherwise the system goes to Step 6

Business Rules & Use Cases

- Business rules are the details by which a business implements strategy.
- These rules are gathered and documented by **Business Analysts**
- There are four categories:

Business Rules & Use Cases

Idaho State University | Computer Science

- There are four categories:
  - **Definitional Rules:** Explain what is meant when a particular word is used in the context of business operations.
    - Ex: In our system "Book" means any book owned by the library.
  - **Factual Rules:** Explain things about the business's operations, and how terms connect to each other.
    - Ex: In our system: `Books are issued to Members` and `Members can place holds on Books`
  - **Constraints:** Specific conditions that govern the manner in which terms can be connected.
    - Ex: A bank may have the constraint: `balance in an account cannot be less than zero`
  - **Derivations** Knowledge that can be derived from the facts and constraints.
    - Ex: A bank may have the constraint: `balance in an account cannot be less than zero` from which we can derive that no withdrawal operation will succeed if it is for greater than the current account balance.
- Typically when working with use cases we are only concerned with `Constraints` and `Derivations`

Use Case Guidelines

Idaho State University | Computer Science

1. Uses cases must provide value to an actor or the business
   - Once the scenario has played out, the actor has accomplished a task
2. Use cases should be **functionally cohesive**
   - They encapsulate a single service of the system
3. Use cases should be **temporally cohesive**
   - There should not be significant delays due to the system
4. If there are multiple actors, they should each be involved in at least one use case
5. The model is the **set of use cases**, there are no relationships between use cases

Use Case Guidelines

6. Exceptional exit conditions are not handled by use cases
   - Assume a reasonable outcome will occur.
7. Use cases are from the point of view of the actor, written in active voice
8. A use case describe a scenario
   - Tells us the visible outcome and does not detail other requirements
9. Use cases change throughout system analysis

For Next Time

- Review Chapter 6
- Review this lecture
- Come to class

Are there any questions?