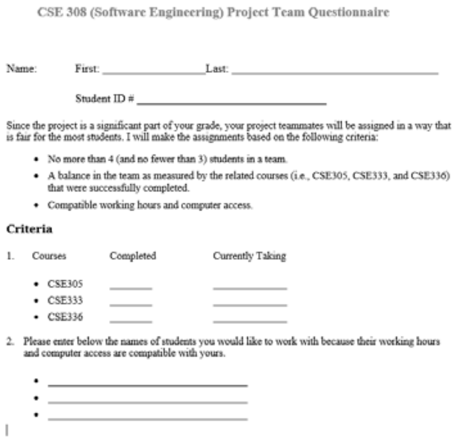
CSE 308

Semester Project Discussion
Electronic Book System

Teams

- Fill out the forms
- Forms collected today
- Teams assigned before class next week



CSE 308 (Software Engineering) Project Team Questionnaire

Name: First: __________ Last: ____________________

Student ID # ____________________

Since the project is a significant part of your grade, your project teammates will be assigned in a way that is fair for the most students. I will make the assignments based on the following criteria:

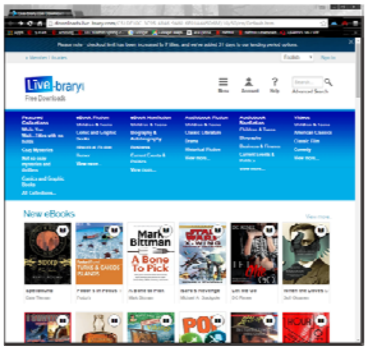
- No more than 4 (and no fewer than 3) students in a team.
- A balance in the team as measured by the related courses (i.e., CSE305, CSE333, and CSE336) that were successfully completed.
- Compatible working hours and computer access.

Criteria

1. Courses Completed Currently Taking
   - CSE305 _____ ________
   - CSE333 _____ ________
   - CSE336 _____ ________
2. Please enter below the names of students you would like to work with because their working hours and computer access are compatible with yours.
   - ____________________
   - ____________________
   - ____________________

eBook System

- The project is a re-engineering of a typical electronic book lending system (e.g., Live-Brary)
- Initial requirements are in the existing system, but you will expand and improve the system



Your grade on the project will be based on the "degree of difficulty" or number of completed use cases of the project

Project Components

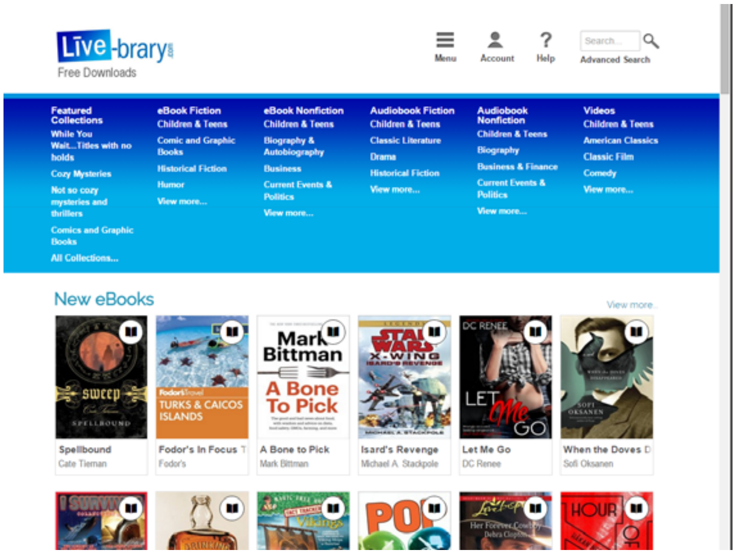
- Web GUI (multiple roles)
- DB
- OO structure
- Persistence layer
- Business Logic
- Reports
- E-mail alerts
- Other servers (e.g., Amazon)
- Multi-server coordination
- Test data

Your first step in the project is to define the requirements

Some requirements you can derive from Live-Brary, others will your analysis

Next week we will discuss how to define these requirements

Web GUI



5

© Robert Kelly, 2005-2016

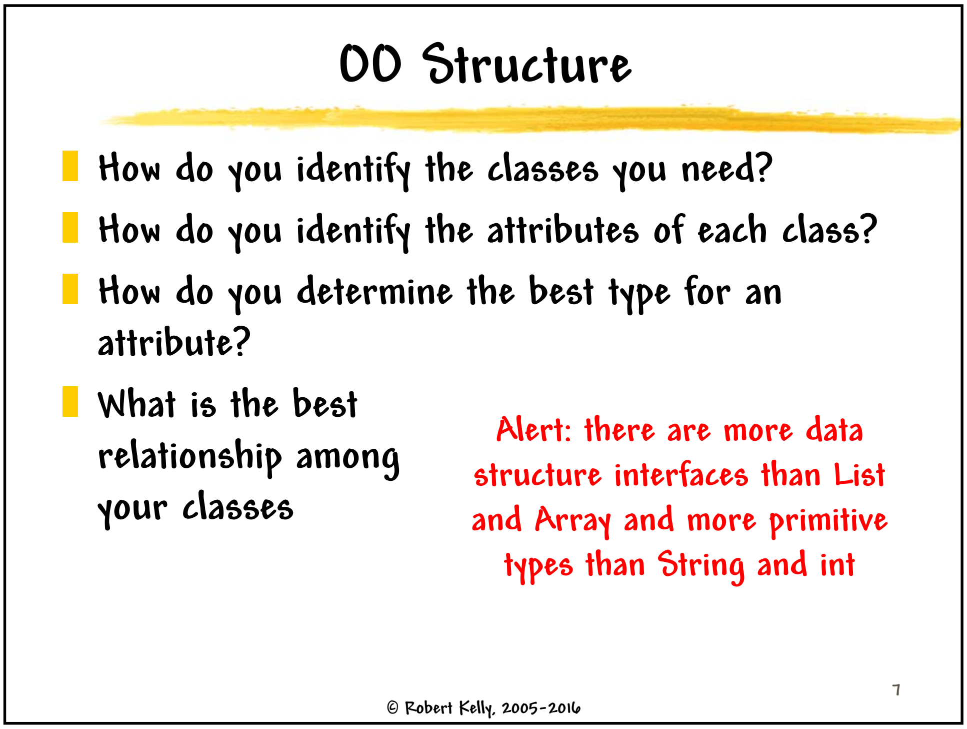
Basic Web GUI Requirements

- HTML 5
- Technologies (CSS, JavaScript, jQuery, etc.)
- Integrated text and graphics
- Some mouse-over actions
- Limited dynamic updating
- Various filters for movies displayed

Your GUI does not need to look exactly the same as Live-Brary

6

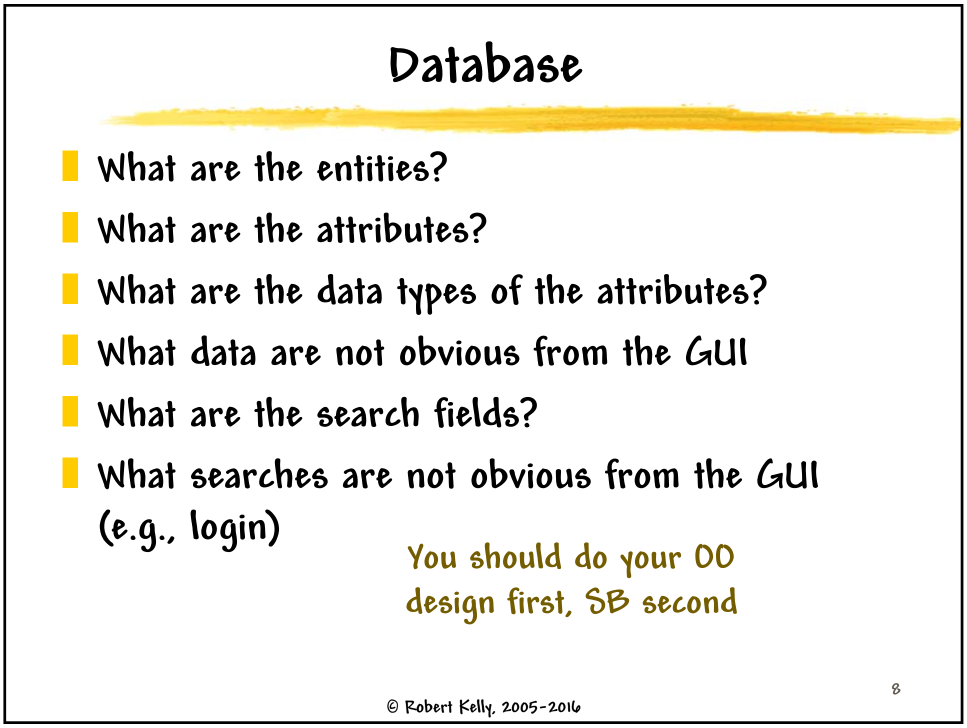
© Robert Kelly, 2005-2016



OO Structure

- How do you identify the classes you need?
- How do you identify the attributes of each class?
- How do you determine the best type for an attribute?
- What is the best relationship among your classes

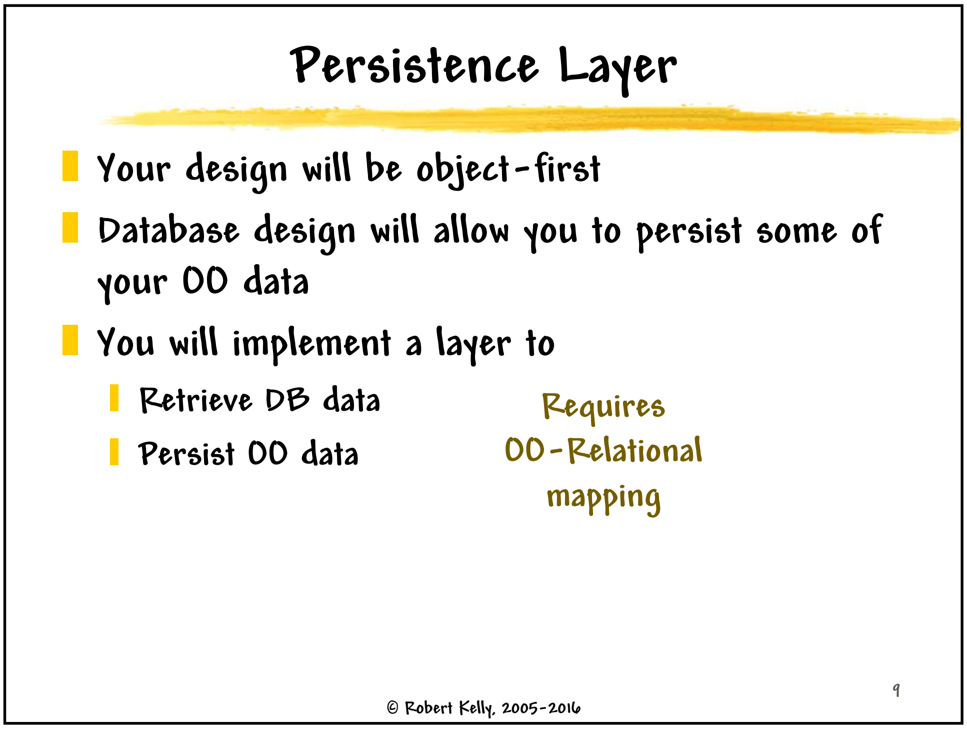
Alert: there are more data structure interfaces than List and Array and more primitive types than String and int



Database

- What are the entities?
- What are the attributes?
- What are the data types of the attributes?
- What data are not obvious from the GUI
- What are the search fields?
- What searches are not obvious from the GUI (e.g., login)

You should do your OO design first, SB second



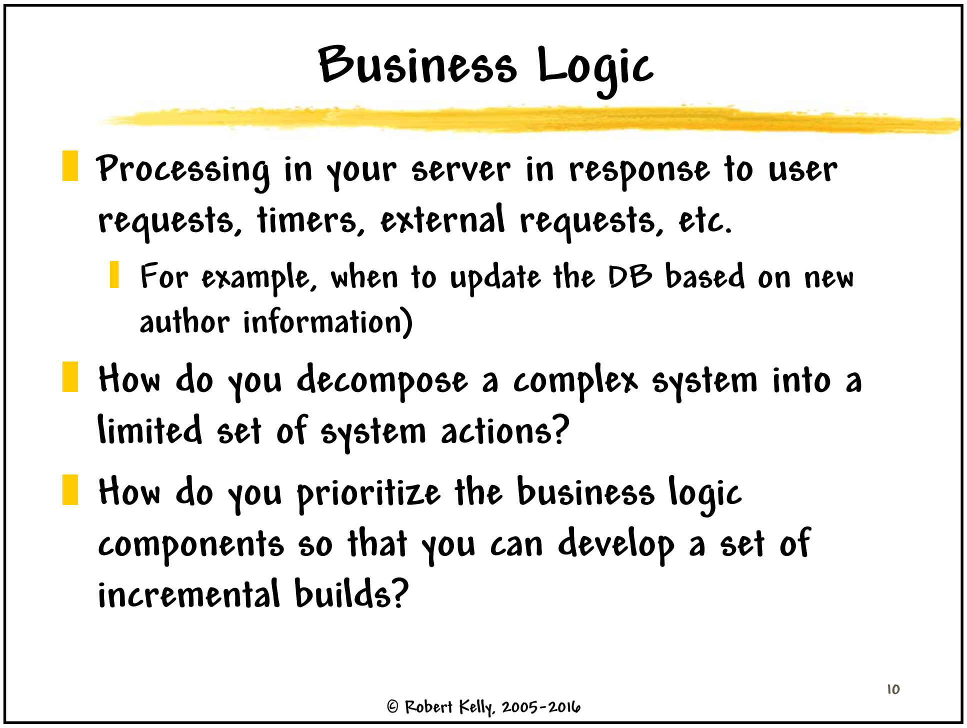
Persistence Layer

- Your design will be object-first
- Database design will allow you to persist some of your OO data
- You will implement a layer to
  - Retrieve DB data
  - Persist OO data

Requires OO-Relational mapping

© Robert Kelly, 2005-2016

9

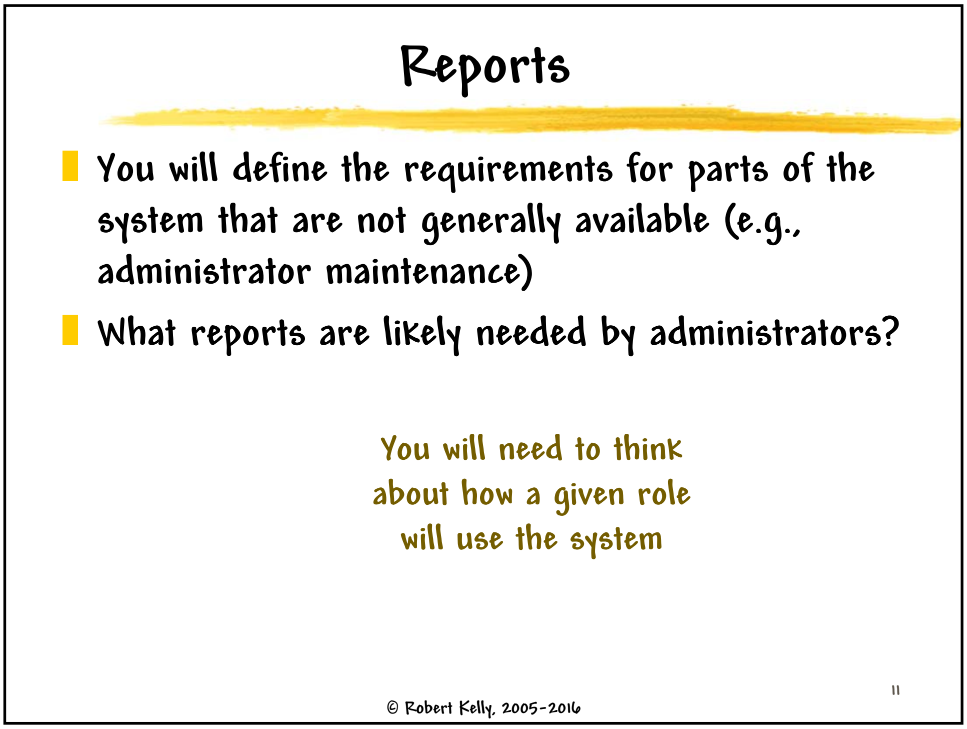


Business Logic

- Processing in your server in response to user requests, timers, external requests, etc.
  - For example, when to update the DB based on new author information)
- How do you decompose a complex system into a limited set of system actions?
- How do you prioritize the business logic components so that you can develop a set of incremental builds?

© Robert Kelly, 2005-2016

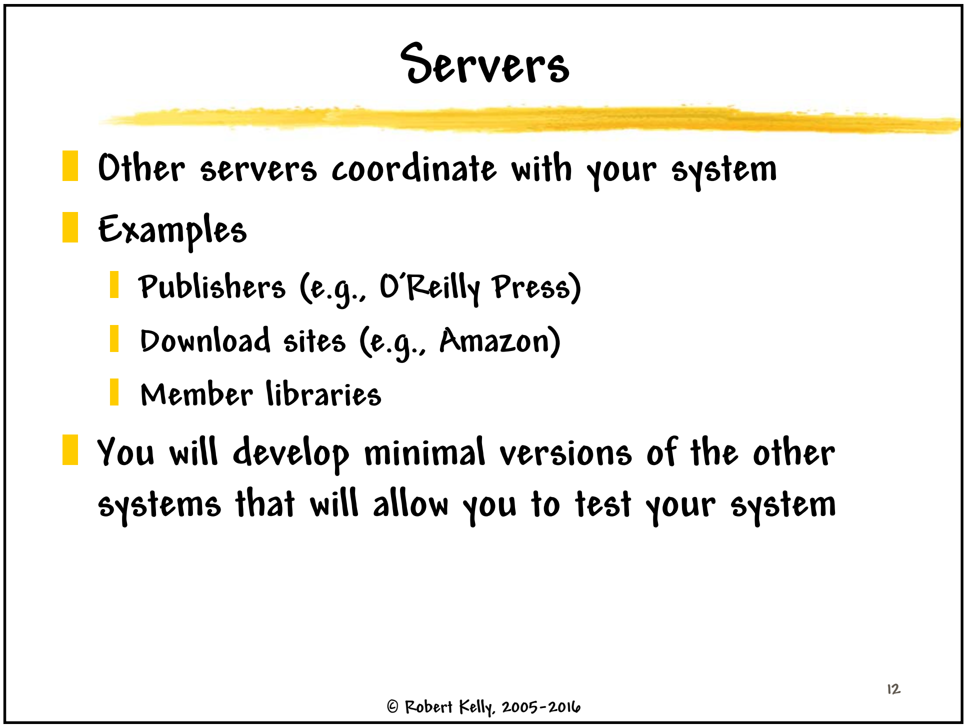
10



Reports

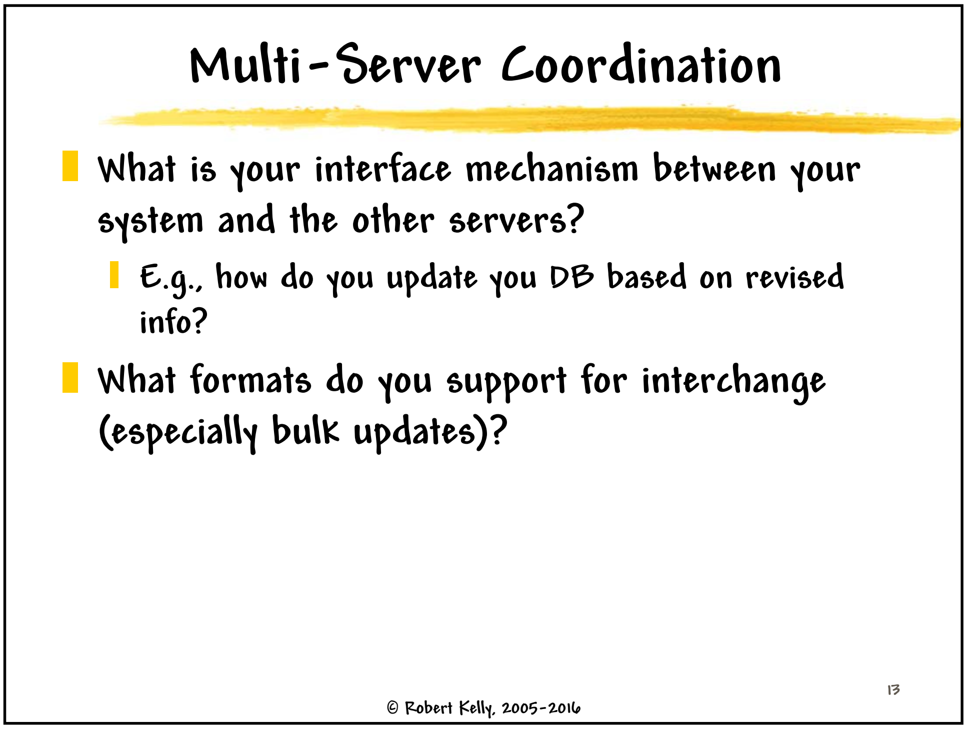
- You will define the requirements for parts of the system that are not generally available (e.g., administrator maintenance)
- What reports are likely needed by administrators?

You will need to think about how a given role will use the system



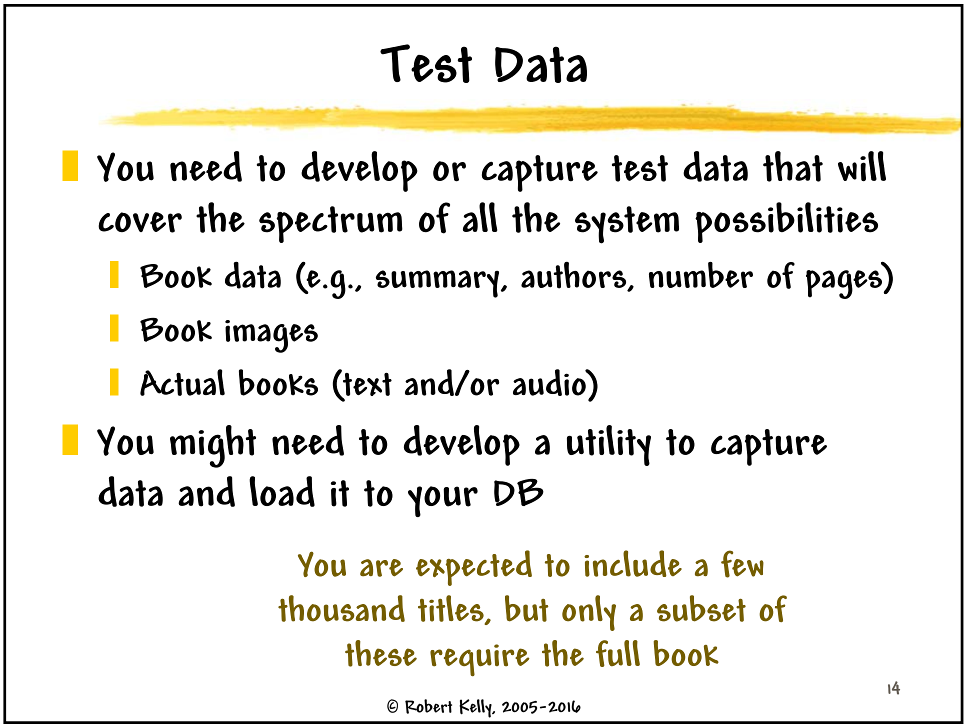
Servers

- Other servers coordinate with your system
- Examples
  - Publishers (e.g., O'Reilly Press)
  - Download sites (e.g., Amazon)
  - Member libraries
- You will develop minimal versions of the other systems that will allow you to test your system



Multi-Server Coordination

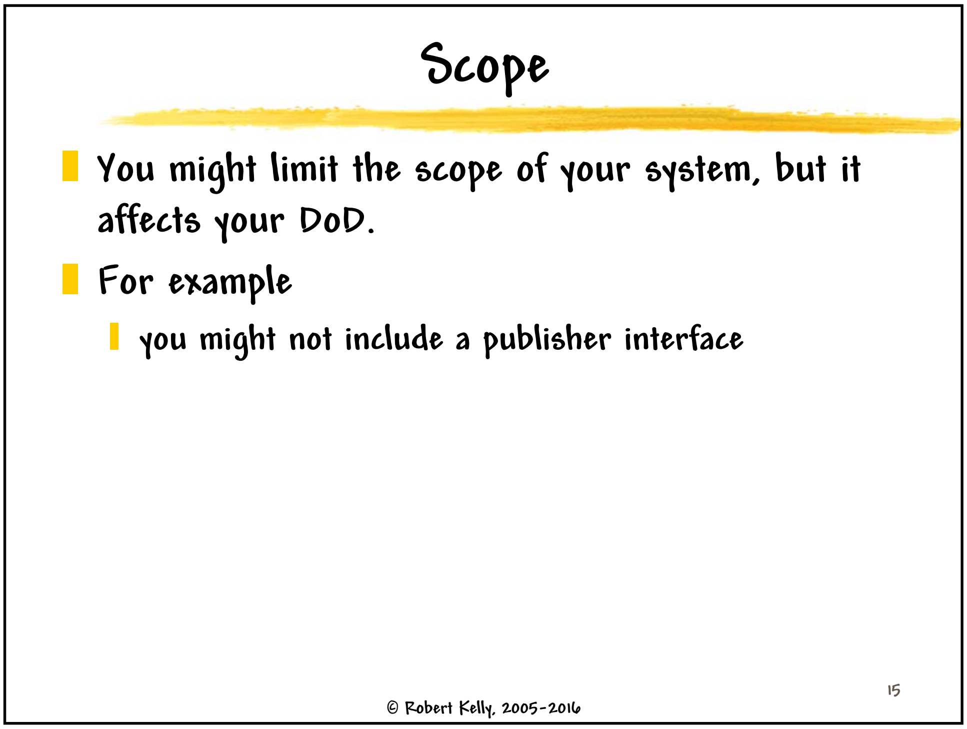
- What is your interface mechanism between your system and the other servers?
  - E.g., how do you update you DB based on revised info?
- What formats do you support for interchange (especially bulk updates)?



Test Data

- You need to develop or capture test data that will cover the spectrum of all the system possibilities
  - Book data (e.g., summary, authors, number of pages)
  - Book images
  - Actual books (text and/or audio)
- You might need to develop a utility to capture data and load it to your DB

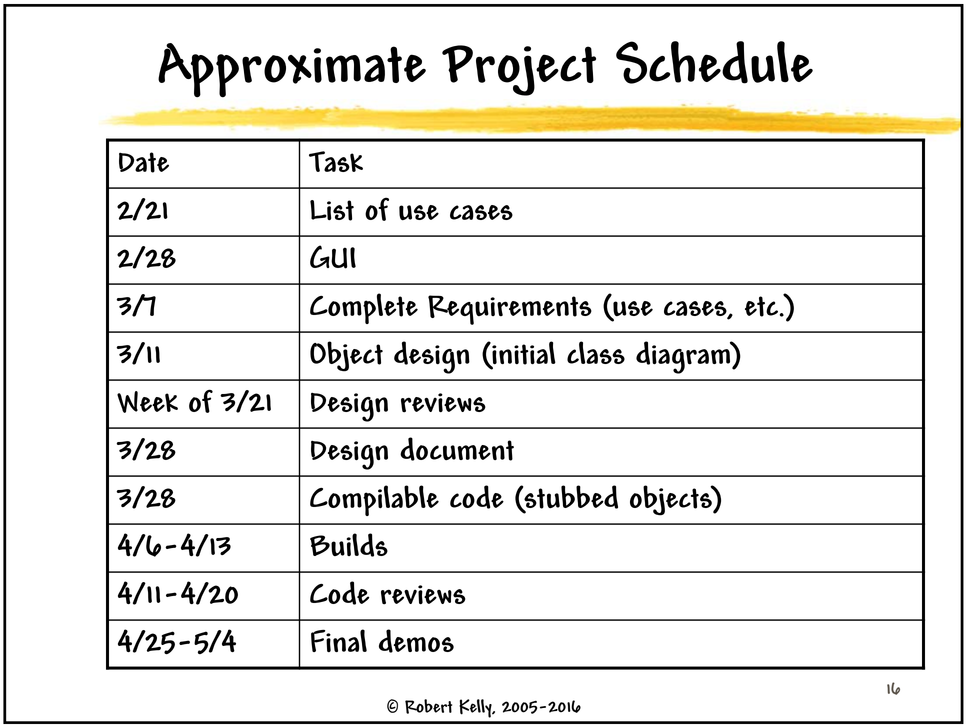
You are expected to include a few thousand titles, but only a subset of these require the full book



Scope

- You might limit the scope of your system, but it affects your DoD.
- For example
  - you might not include a publisher interface

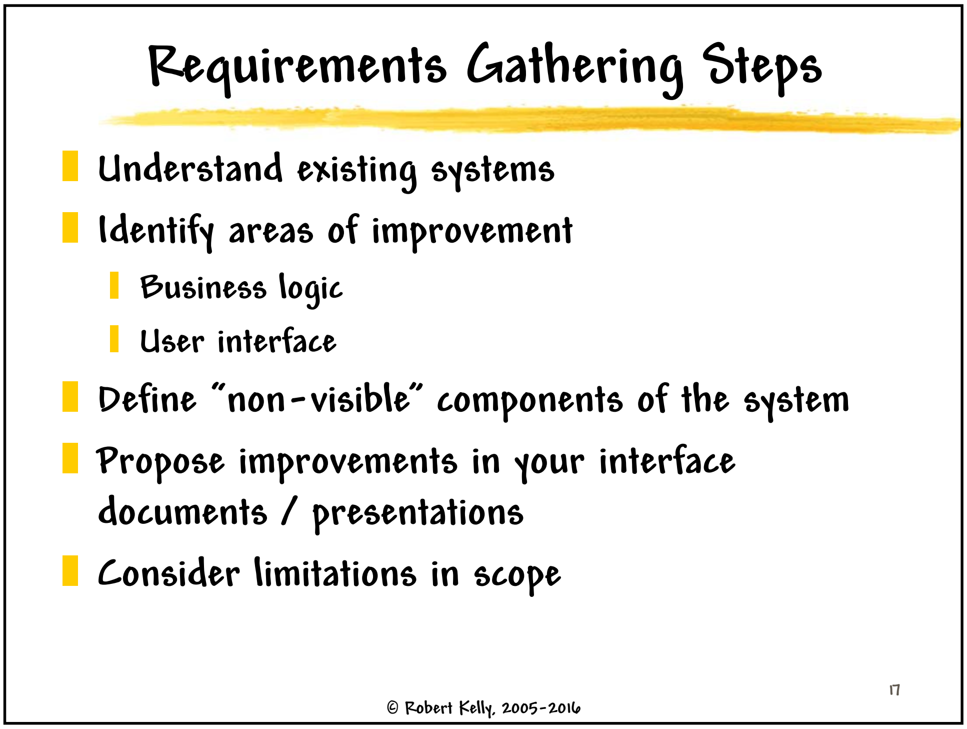
15



Approximate Project Schedule

Date	Task
2/21	List of use cases
2/28	GUI
3/7	Complete Requirements (use cases, etc.)
3/11	Object design (initial class diagram)
Week of 3/21	Design reviews
3/28	Design document
3/28	Compilable code (stubbed objects)
4/6-4/13	Builds
4/11-4/20	Code reviews
4/25-5/4	Final demos

16

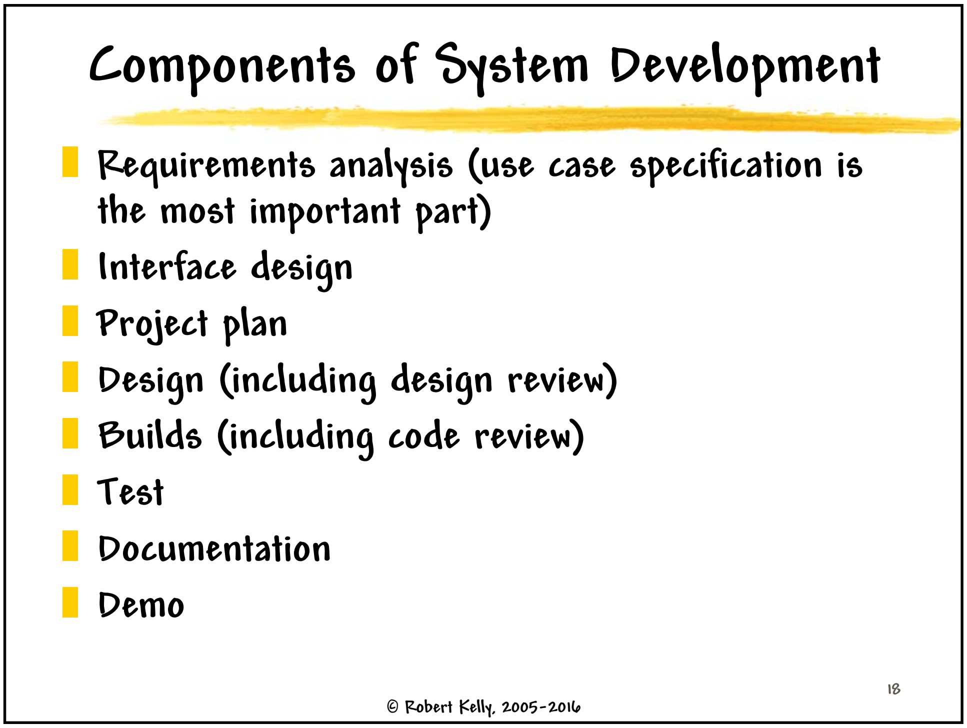


Requirements Gathering Steps

- Understand existing systems
- Identify areas of improvement
  - Business logic
  - User interface
- Define "non-visible" components of the system
- Propose improvements in your interface documents / presentations
- Consider limitations in scope

© Robert Kelly, 2005-2016

17

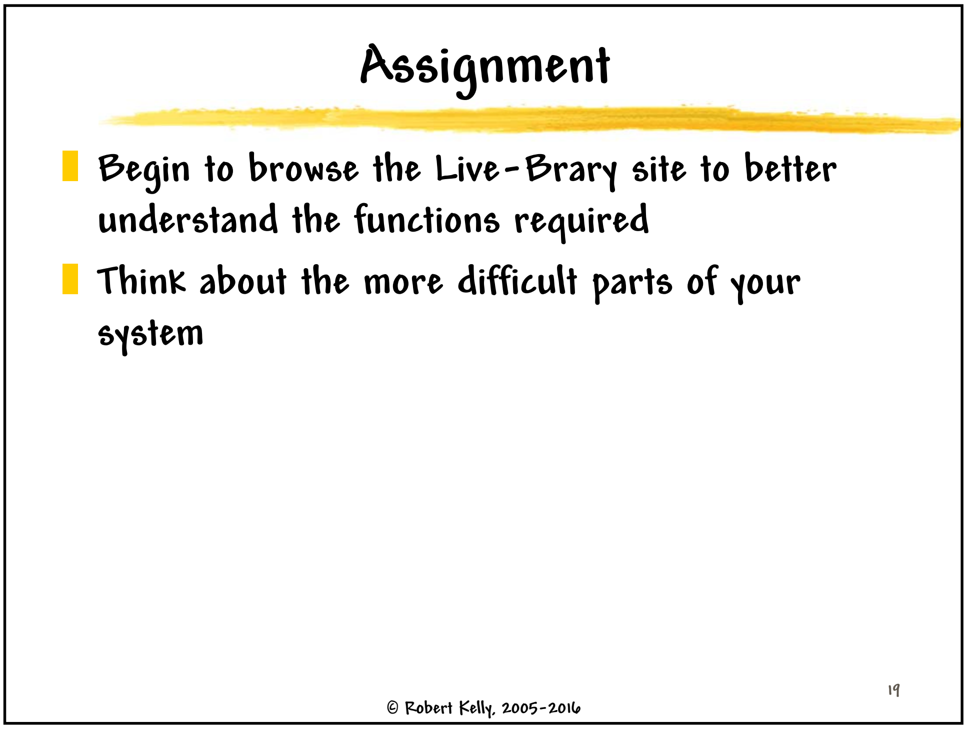


Components of System Development

- Requirements analysis (use case specification is the most important part)
- Interface design
- Project plan
- Design (including design review)
- Builds (including code review)
- Test
- Documentation
- Demo

© Robert Kelly, 2005-2016

18



Assignment

- Begin to browse the Live-Brary site to better understand the functions required
- Think about the more difficult parts of your system