

# **Software Requirements Elicitation**

Dr John H Robb, PMP, IEEE SEMC

UT Arlington

Computer Science and Engineering

# Key Takeaway Points

- Requirements are capabilities that the system must deliver.
- *The hardest single part of building a software system is deciding precisely what to build—i.e., the requirements.* (Frederick P. Brooks, Jr.)
- Requirements are the main challenge in developing software - this is the problem area!
- Software requirements elicitation is aimed to identify the real requirements for the system – this may not be the same as what the customer asked for!
- This module provides an overview of the requirements process and then focuses on the requirements elicitation - which is the emphasis of the textbook and requirements specification - which is the focus of the project

# Overview of the Requirements Process

# The Requirements Process

The Requirements Process consists of the following steps

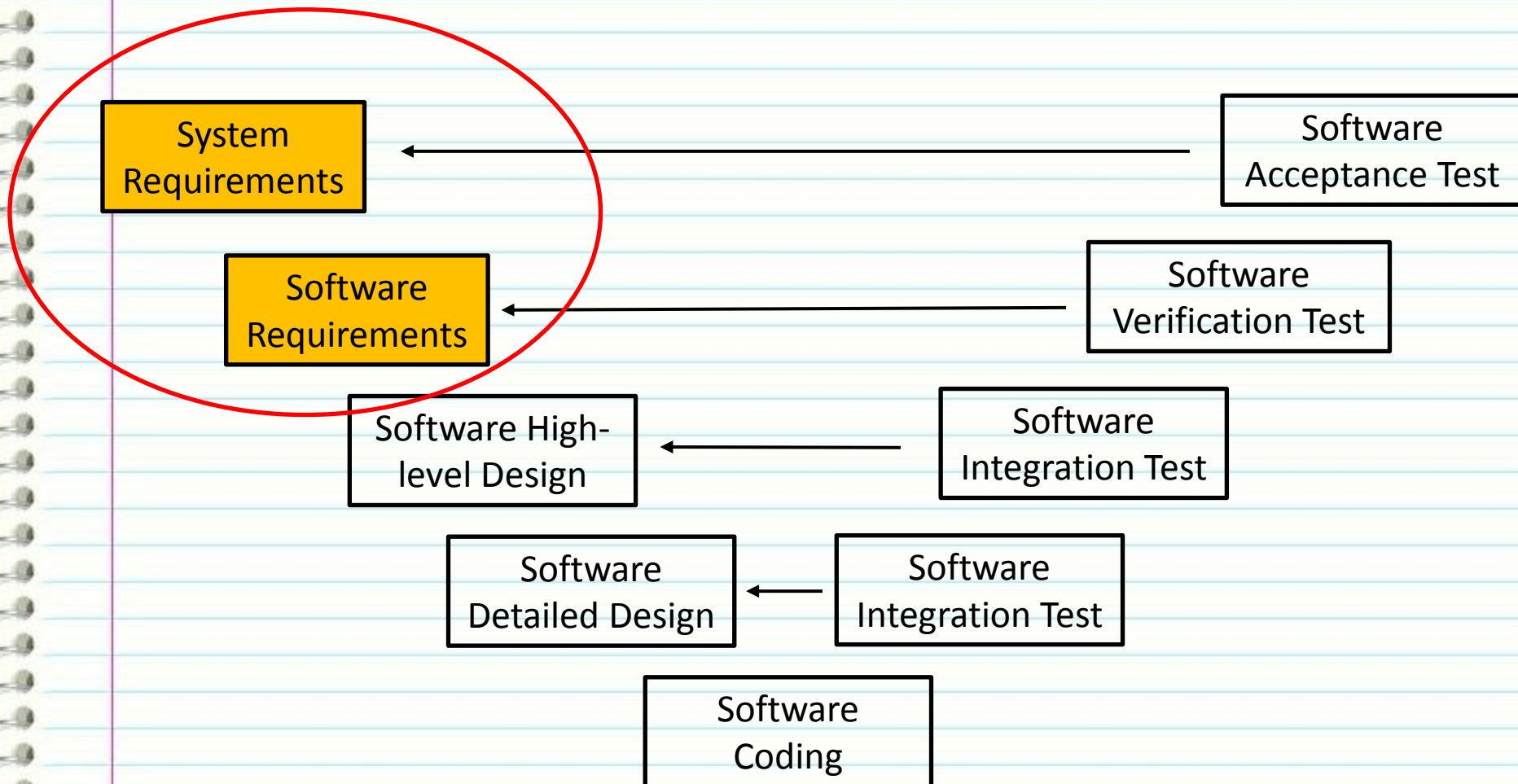
- A. Requirements Planning (estimating requirements work)
- B. Requirements Elicitation (draw-out the requirements)
- C. Requirements Analysis (do they work and work together?)
- D. Software Requirements Specification (capture requirements)
- E. Requirements Validation
- F. Requirements Management (requirements will change - they must be managed)
- G. Requirements status reporting

Most of the industry is particularly weak in all but D and E above and many are weak here as well.

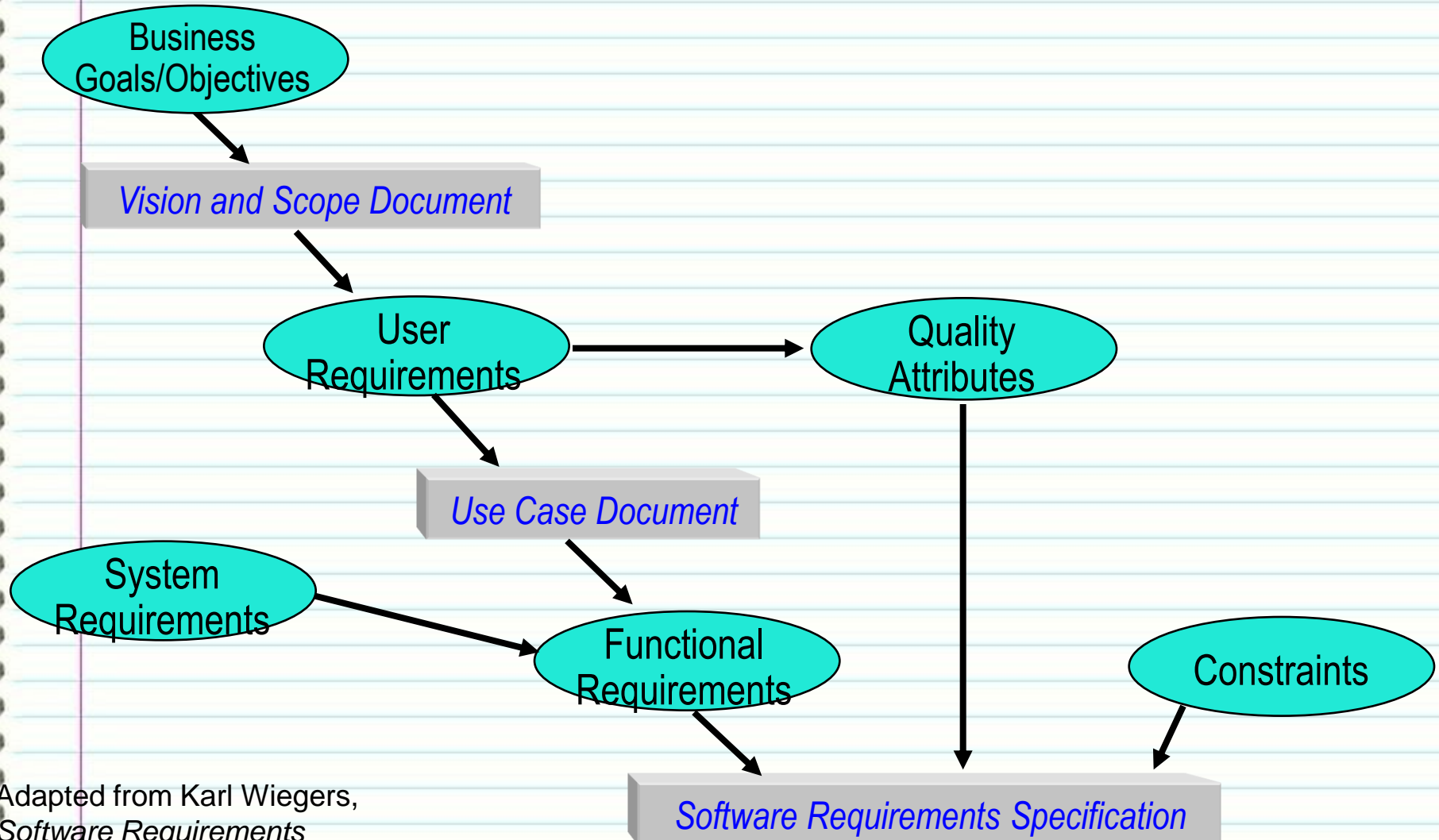
Questions to discuss:

1. What are typical software estimation measures and how do they apply to software requirements?

# Requirements in the Life Cycle



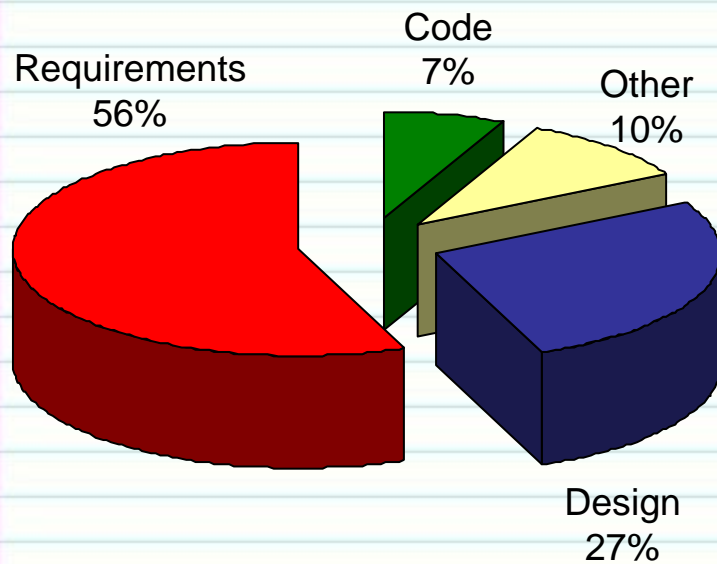
# Software Requirements Specification



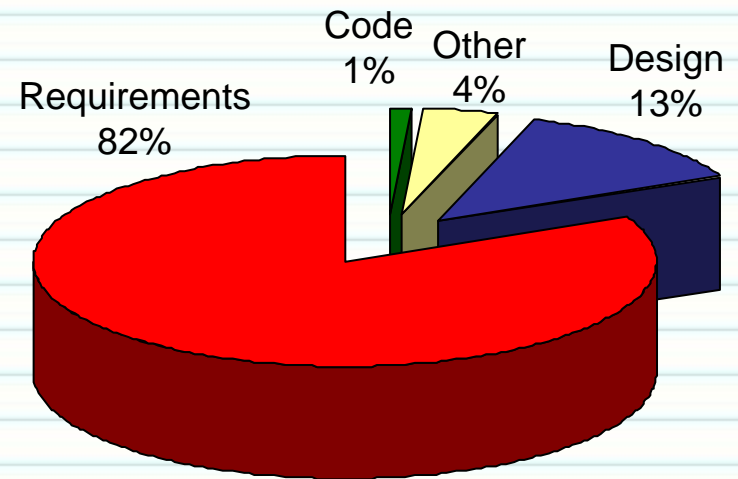
Adapted from Karl Wieggers,  
*Software Requirements*

# Why Do We Manage Requirements ?

*Distribution of Defects*



*Distribution of Effort to Fix Defects*



(Martin & Leffinwell)



# Requirements Elicitation

The book concentrates mostly on Requirements Elicitation

- Requirements are capabilities (stated as part of a contract) that the system must deliver.
- Requirements are documented in a requirements specification, which serves as part of the contract.
- Requirements elicitation is the process to identify and formulate the capabilities for the software system.
  - Identifying problems and needs
  - Constructing analysis models to help understanding
  - Formulating system/software requirements
  - Conducting feasibility study
  - Checking the requirements and models for desired properties such as correctness, and consistency
  - Specifying acceptance tests
  - Formulating an iterative development plan



# Typical Elicitation Mistakes

- **Noise**
  - the presence of text that carries no relevant information to any feature of the problem.
- **Silence**
  - a feature that is not covered by any text.
- **Over-specification**
  - text that describes a feature of the solution, rather than the problem.
- **Contradiction**
  - text that defines a single feature in a number of incompatible ways.
- **Ambiguity**
  - text that can be interpreted in at least two different ways.
- **Forward reference**
  - text that refers to a feature yet to be defined.
- **Wishful thinking**
  - text that defines a feature that cannot possibly be validated.
- **Jigsaw puzzles**
  - e.g. distributing requirements across a document and then cross-referencing
- **Inconsistent terminology**
  - Inventing and then changing terminology
- **Putting the onus on the development staff**
  - i.e. making the reader work hard to decipher the intent
- **Writing for the hostile reader**
  - There are fewer of these than friendly readers

*Source: Steve Easterbrook, U. of Toronto*

# Requirements Problems Can Be Disastrous!

- **Insufficient requirements specification and their ad hoc management**
- **Ambiguous and imprecise communication**
- Brittle architecture
- Overwhelming complexity
- **Undetected inconsistencies in requirements, design, and implementation**
- Poor and insufficient testing
- Subjective assessment of project status
- Failure to attack risk
- Uncontrolled change propagation
- Insufficient automation

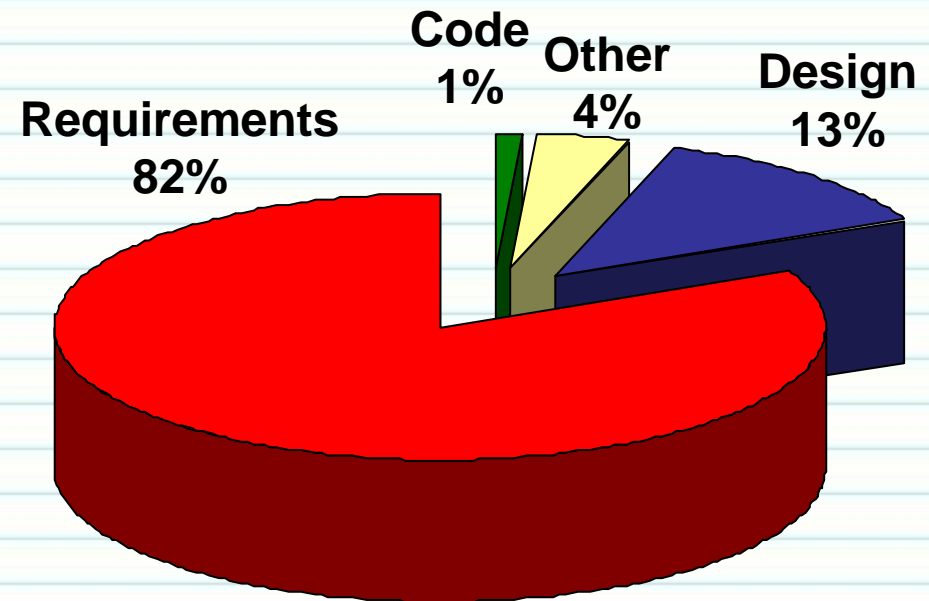
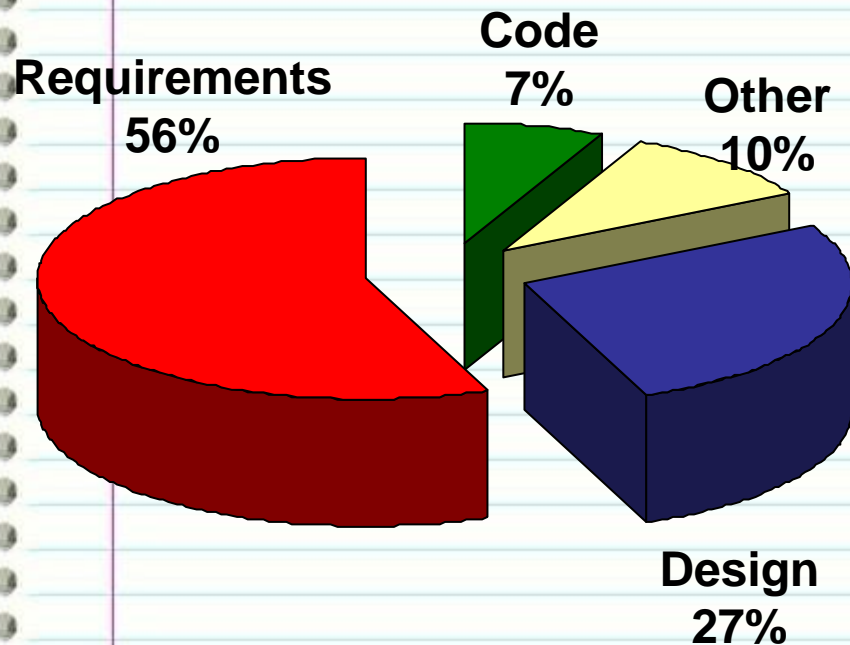
# Statistics from NIST Report

- NIST (**National Institute of Standards and Technology**) has published a comprehensive (309 pages) and very interesting report on project statistics and experiences based on data from a large number of software projects<sup>1</sup>
  - 70% of the defects are introduced in the **specification** phase
  - 30% are introduced **later** in the technical solution process
  - Only 5% of the specification inadequacies are corrected in the specification phase
  - 95% are **detected later** in the project or after delivery where the cost for correction on average is 22 times higher compared to a correction directly during the specification effort
  - The NIST report concludes that extensive testing is essential, however testing detects the dominating specification errors late in the process

[1] [http://www.nist.gov/public\\_affairs/releases/n02-10.htm](http://www.nist.gov/public_affairs/releases/n02-10.htm) (May 2002)

# Why Focus on Requirements ?

- Distribution of Defects
- Distribution of Effort to Fix Defects



## Back To Requirements Elicitation

# Challenges of Requirements Elicitation

"The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is as difficult as establishing that detailed technical requirements, including all the interfaces to people, to machines, and to other software systems. No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later."

Frederick Brooks 1987



# Challenges of Requirements Elicitation

As an analyst, I need to know what do you want?



I want you to design the software for me.



But what do you want to do with the software?



I don't know until you tell me what the software can do.



Well, I can design the software to do anything!



Can you design the software to tell you my requirements?!



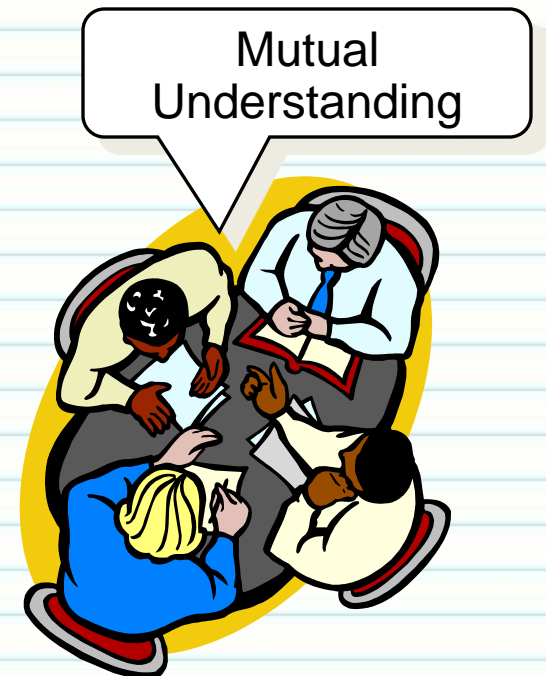


# Communication Barrier - Misunderstanding



# What is a Requirement?

- Statement of some THING you want or need  
*OR*  
A characteristic of some THING you want or need
- A requirement is also...
  - A *Contractually Binding* Statement
  - Documentation of *Problem Space*
  - The *Means* We Use to Communicate



# Types of Requirement

- Functional requirements – statements of information processing capabilities that the software system must possess.
- Nonfunctional requirements include
  - Performance requirements
  - Quality requirements
  - Safety requirements
  - Security requirements
  - Interface requirements

# Examples of Functional Requirements

- For a car rental system:
  - The system *shall* allow a potential customer to inquire information and availability of rental cars using various combinations of search criteria including make, model, from date, to date, price range, and class (small size, medium size, large size, and luxury cars).
- For a study abroad system:
  - The system *shall* provide interactive as well as batch-processing means for an OIE (Office of International Education) staff to enter the exchange programs into the database.

# Examples of Non-Functional Requirements

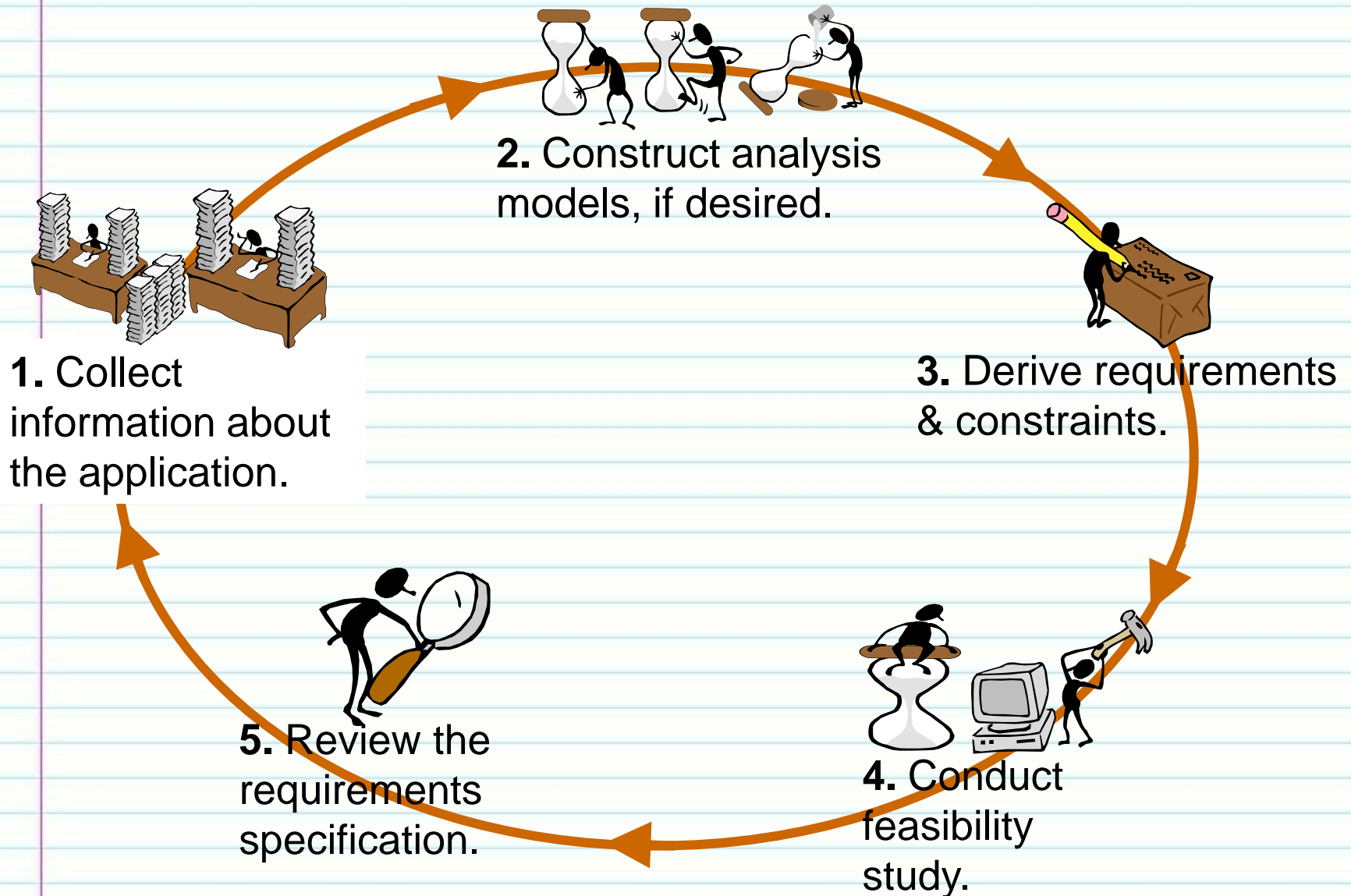
- Workload:
  - The system shall be capable of handling a typical workload of 10,000 (ten thousand) inquiries at the same time.
- Response time:
  - The system's response time shall not exceed 3 (three) seconds under the typical workload.

# Examples of Constraints

- A constraint is a limitation against the design or implementation (coding) space. These are not functional or non-functional requirements.
- Development Methodology (sometimes specified by the customer)
  - The System/software shall use the Object Oriented methodology
  - The Software shall be coded in Java
- What constraints will we have regarding the project?



# Requirements Elicitation Steps





# Information Collection Techniques



Customer presentation



business forms



operating procedures



regulations & standards

Literature survey



Stakeholder survey



User interviewing

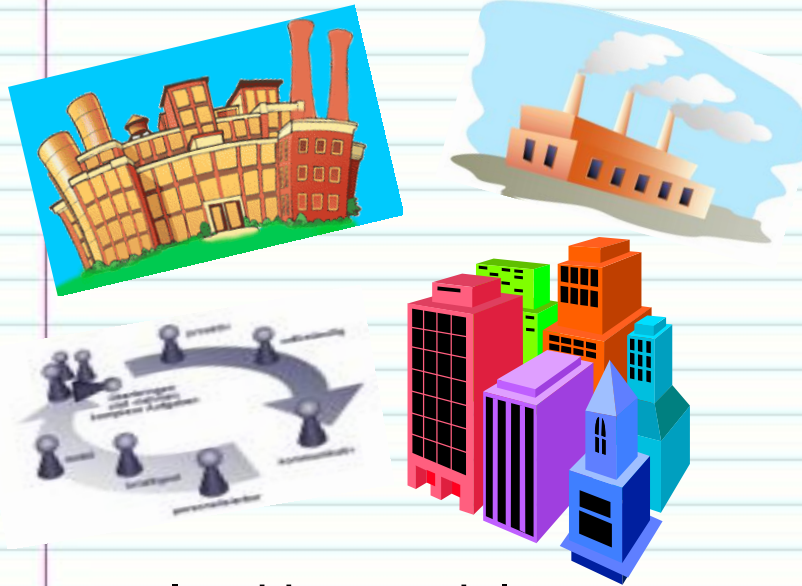


Writing user stories

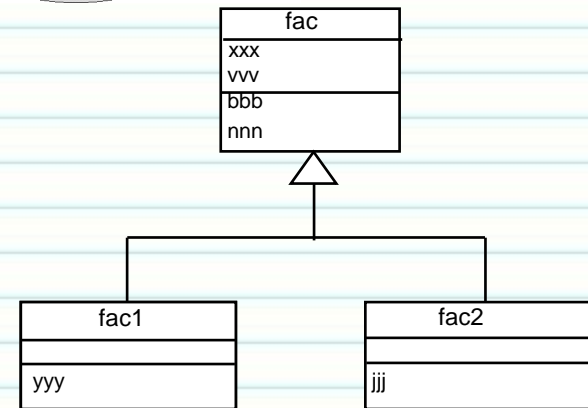
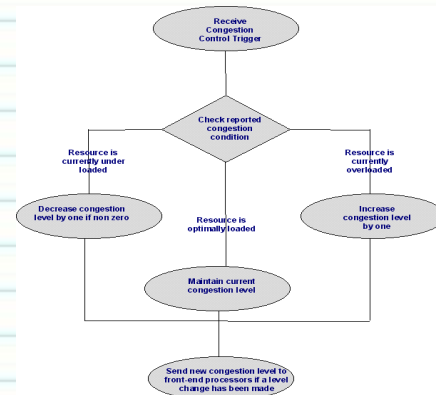
# Focuses of Information Collection Activities

- What is the business, the current business situation, and how does it operate?
- What is the system's environment or context?
- What are existing business processes, their input and output, and how do they relate to each other?
- What are the problems with the current system?
- What are the business or product goals?
- Who are the users of the current and future systems, respectively?
- What do the customer and users want, and what are their business priorities?
- What are the quality, performance, and security considerations?

# Constructing Analysis Models



Intuitive models



Purpose: to aid understanding of the application, requirements, Formal or informal models and constraints.



# Businesses in Different Domains



Finance



Insurance



Health Care



Transportation



Defense



Telecommunication



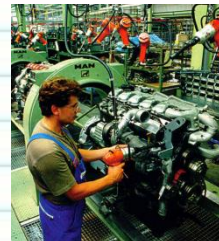
Retailing



Energy



government



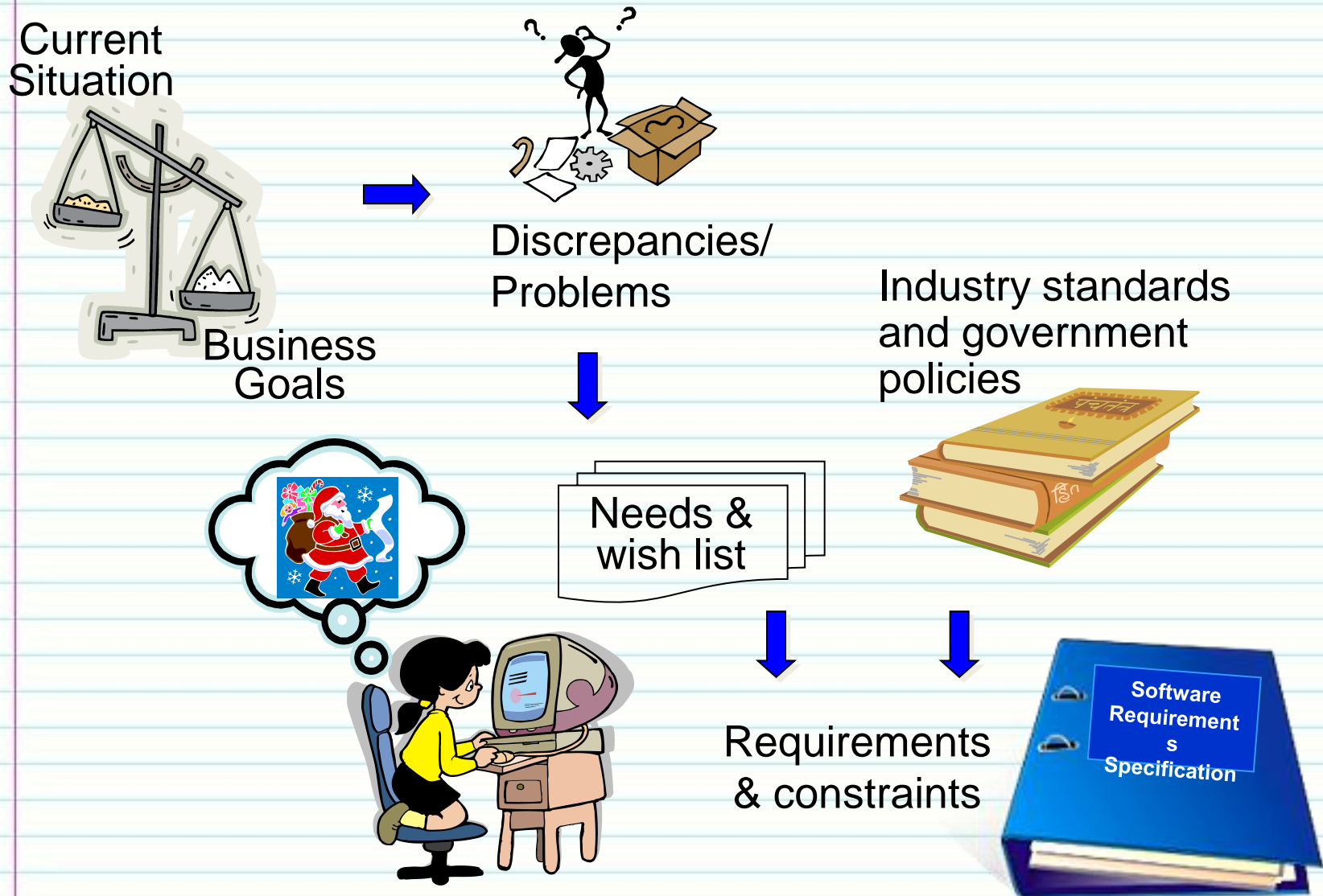
Manufacturing



Education

and more

# Deriving Requirements and Constraints



# Requirements Specification Details

# Requirements Specification

- 1. Introduction to Document
  - 1.1 Purpose of Product
  - 1.2 Scope of Product
  - 1.3 Acronyms, Abbreviations, Definitions
  - 1.4 References
  - 1.5 Outline of the Rest of the SRS
- 2. General Description of Product
  - 2.1 Context of Product
  - 2.2 Product Function
  - 2.3 User Characteristics
  - 2.4 Constraints
  - 2.5 Assumptions and Dependencies
- 3. Specific Requirements
  - 3.1 External Interface Requirements
    - 3.1.1 User Interfaces
    - 3.1.2 Hardware Interfaces
    - 3.1.3 Software Interfaces
    - 3.1.4 Communication Interfaces
  - 3.2 Functional Requirements
    - 3.2.1 Class 1
    - 3.2.2 Class 2
    - 3.2.3 ...
  - 3.3 Performance Requirements
  - 3.4 Design Constraints
  - 3.5 Quality Requirements
  - 3.6 Other Requirements
- 4. Appendices

IEEE SRS Standard by Objects, 1998



# Feasibility Study

- Not all projects are practically doable with technology, time, and resource constraints.
- Feasibility study aims at determining if the project is doable under the given constraints.
- Feasibility study in RE is concerned with
  - the feasibility of the functional, performance, nonfunctional, and quality constraints
  - adequacy of the technology
  - timing and cost constraints
  - constraints imposed by the customer, industry and government agencies

# Three Types of Requirements Review

- Technical review is an internal review performed by the technical team. Techniques include:
  - **peer review** - peers perform informal “desktop reviews” sometimes guided by a review questionnaire
  - **walkthrough** - the analyst explains each requirement while the reviewers examine it and raise doubts
  - **inspection** - inspector is guided by a checklist of commonly encountered problems in SRS (e.g., incompleteness, duplicate definition, inconsistency, etc.)
- In every case there is a requirement for period of time between which the review materials are available and the review is held – to allow for the pre-review (this is where most errors are found)

# Three Types of Requirements Review

- Additional reviews (these are validation activities)
  - Expert review means review of the requirements specification by domain experts.
  - Customer/user reviews are performed by involving the customer and/or users of the system.

# What is the right system to build ?



How the customer explained it



How the Project Leader understood it



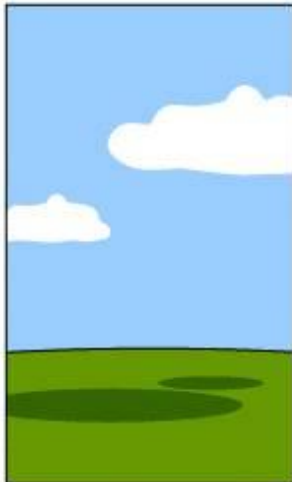
How the Analyst designed it



How the Programmer wrote it



How the Business Consultant described it



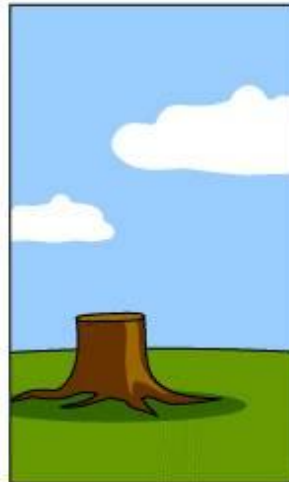
How the project was documented



What operations installed



How the customer was billed



How it was supported



What the customer really needed

# Quality Requirement Characteristics

- Correct
- Feasible
- Necessary
- Prioritized
- Unambiguous
- Verifiable



# Necessary Requirement

- Something the customer really needs
- Something required for conformance to external requirement, external interface, or standard
- Should be traceable back to its source
  - Source should be someone authorized to specify requirements
  - Untraceable requirements may not really be necessary
- Estimates are that over 30% of most software requirements are capabilities that the user never asked for
  - "Gold plating"

# Module Summary: Requirements The Basics

- Requirements define the problem to be solved and establish the terms by which mission success will be measured.
- Requirements problems are the single biggest problem on development projects so care in creating good requirements always pays off.
- The later a problem is discovered the more costly it is to recover from.
- Requirements are distributed within the system architecture via flow-down, allocation and derivation.
- Requirements traceability is a technique of tracking the source and connections between requirements. It is used to assess the consequences of potential requirements changes.
- When a system is decomposed into smaller segments, interfaces are created that must be defined and managed.



# Requirements and the Class Project

1. Start with your analysis functions
2. Use the requirements template to scope the requirements tables (UCID)

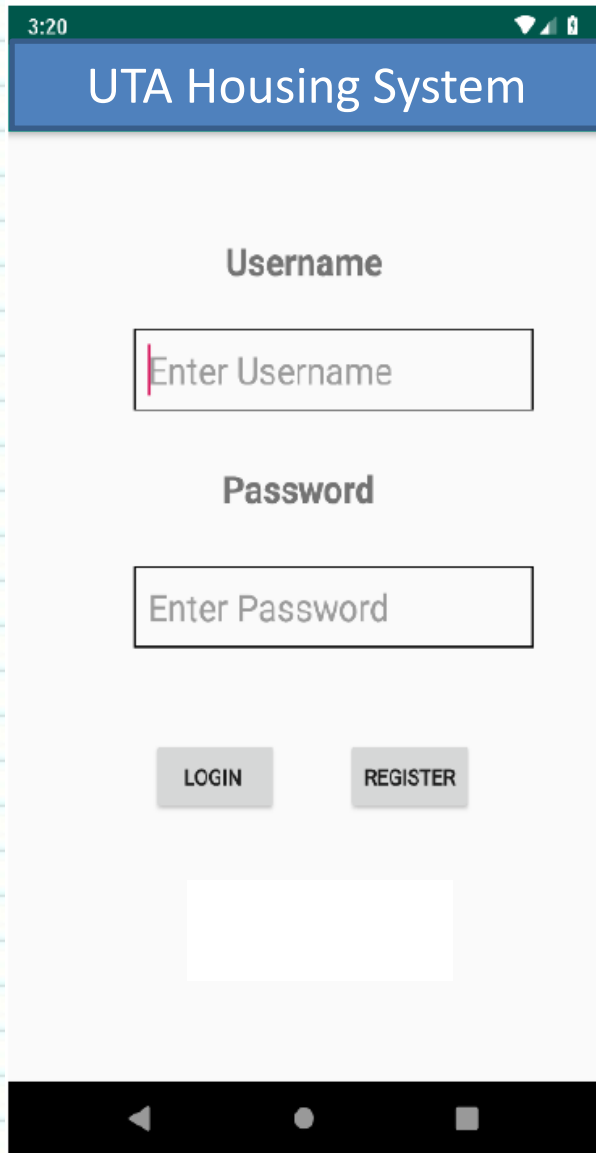
Req Id	Description	Actor
<b>R1</b>	<b>Main Application</b>	System User
R1.1	The Main Application shall provide the functions defined in UCID table 1.	
<b>R2</b>	<b>Student Homescreen</b>	Student
R2.1	The Student Homescreen shall provide the functions defined in UCID table 2.	
<b>R3</b>	<b>Housing Manager Homescreen</b>	Housing Manager
R3.1	The Housing Manager Homescreen shall provide the functions defined in UCID table 3.	
<b>R4</b>	<b>Admin Homescreen</b>	Admin
R4.1	The Admin Homescreen shall provide the functions defined in UCID table 4.	
<b>C1</b>	<b>The Student Housing application has the following constraints</b>	
C1.1	Constraint 1	
C1.2	Constraint 2	

# Requirements and the Class Project

3. Develop the flow of the application using the UCID tables - notice that we have added the ***Starts from*** and ***Ends On*** columns
  - a. Visualize the Homescreen/Main Application
  - b. Click “Select” functions from these
  - c. Input data when prompted and Submit
  - d. Provide some sort of confirmation (see checklist)
  - e. Where do you go after this?

Table ID	Function Number	Function Name	System User	Starts From	Input (input by system user)	Output (output to system user)	Ends On	Description
1	1	Registration	Student	Application mainscreen	username password last name first name role UTA Id (student only) phone email street address city state zip code	none	Application mainscreen (to login after registering)	System User fills in the registration form to sign up. The profile is saved after a confirmation message.
1	2	Login	Student, Residence Manager, Admin	Application mainscreen	username password	none	System user homescreen	The System User logs in to the system

# Requirements and the Class Project (cont.)



3:20

UTA Housing System

Username

Enter Username

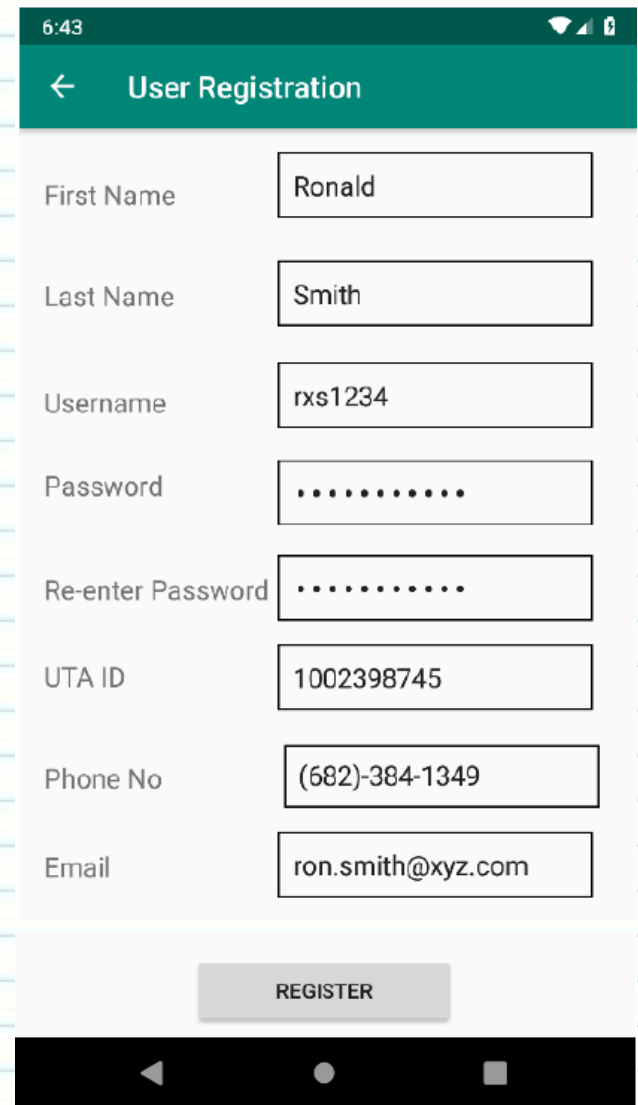
Password

Enter Password

LOGIN REGISTER

## Main Application screen

- 1) Select Register
- 2) Go to Registration screen
- 3) Fill in data
- 4) Click Register



6:43

User Registration

First Name Ronald

Last Name Smith

Username rxs1234

Password .....

Re-enter Password .....

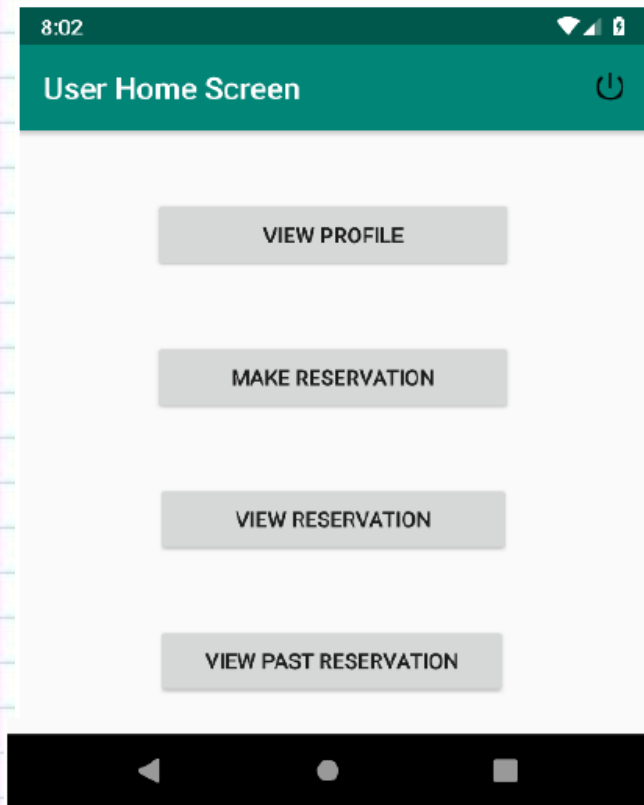
UTA ID 1002398745

Phone No (682)-384-1349

Email ron.smith@xyz.com

REGISTER

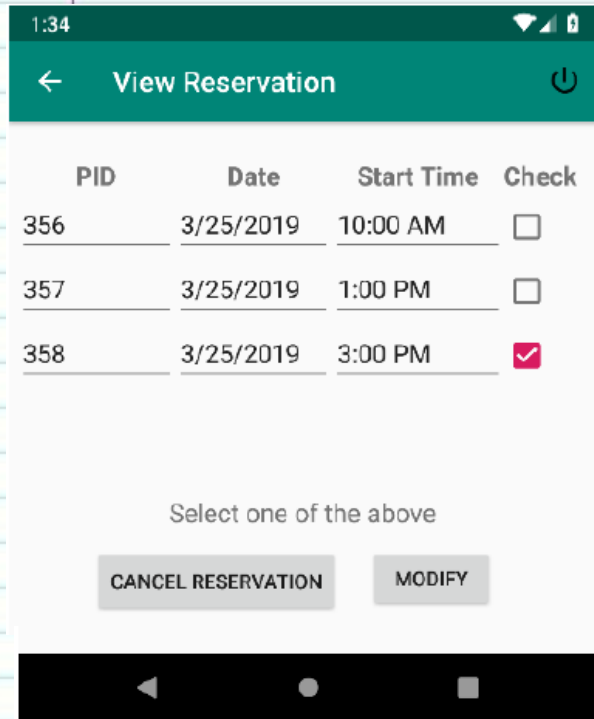
# Requirements and the Class Project (cont.)



## User Homescreen

- 1) Select a function
- 2) Enter data
- 3) Receive confirmation
- 4) Where do you go?

# Requirements and the Class Project (cont.)



1:34

← View Reservation

PID	Date	Start Time	Check
356	3/25/2019	10:00 AM	<input type="checkbox"/>
357	3/25/2019	1:00 PM	<input type="checkbox"/>
358	3/25/2019	3:00 PM	<input checked="" type="checkbox"/>

Select one of the above

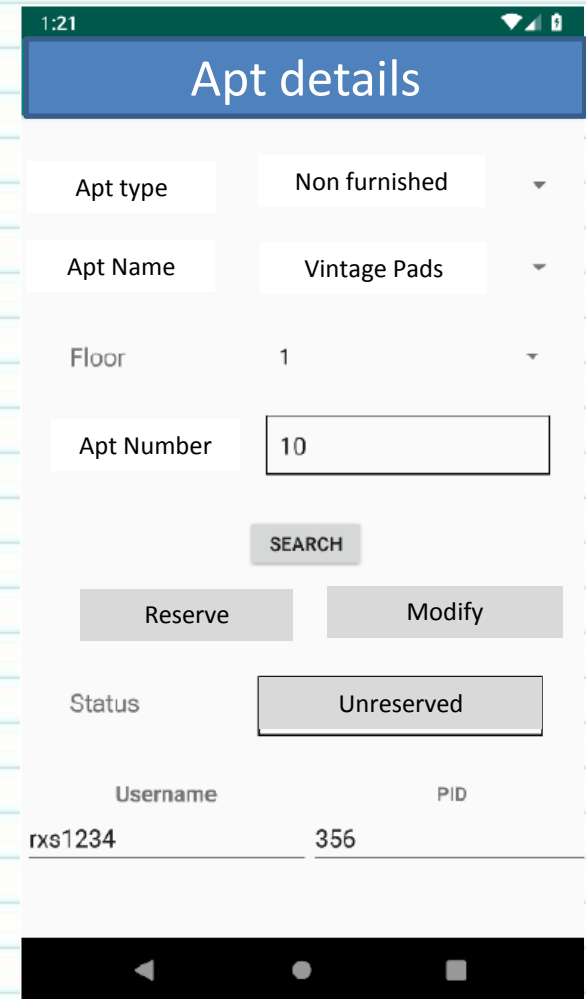
CANCEL RESERVATION MODIFY

Two different kinds of outputs: lists and details.

The left is a list - it provides a summary.

To the right is a specific item in the list.

The list has to provide the ability to select an individual item



1:21

Apt details

Apt type Non furnished

Apt Name Vintage Pads

Floor 1

Apt Number 10

SEARCH

Reserve Modify

Status Unreserved

Username PID

rxs1234 356

# Requirements and the Class Project (cont.)

- What you are doing for the requirements is laying out the
  1. Inputs
  2. Outputs
  3. Processing (sorting on what and how/confirmation & redirect)
  4. Application Flow (UCID tables)