

# SENG2130 – Week 10

## Testing

## User Interface

## Deployment

SENG2130 – Systems Analysis and Design  
University of Newcastle



THE UNIVERSITY OF  
**NEWCASTLE**  
AUSTRALIA

FACULTY OF  
ENGINEERING AND  
BUILT ENVIRONMENT



[www.newcastle.edu.au](http://www.newcastle.edu.au)

# Overview

- System Testing
- User Interface
- System Deployment approaches

# System Testing

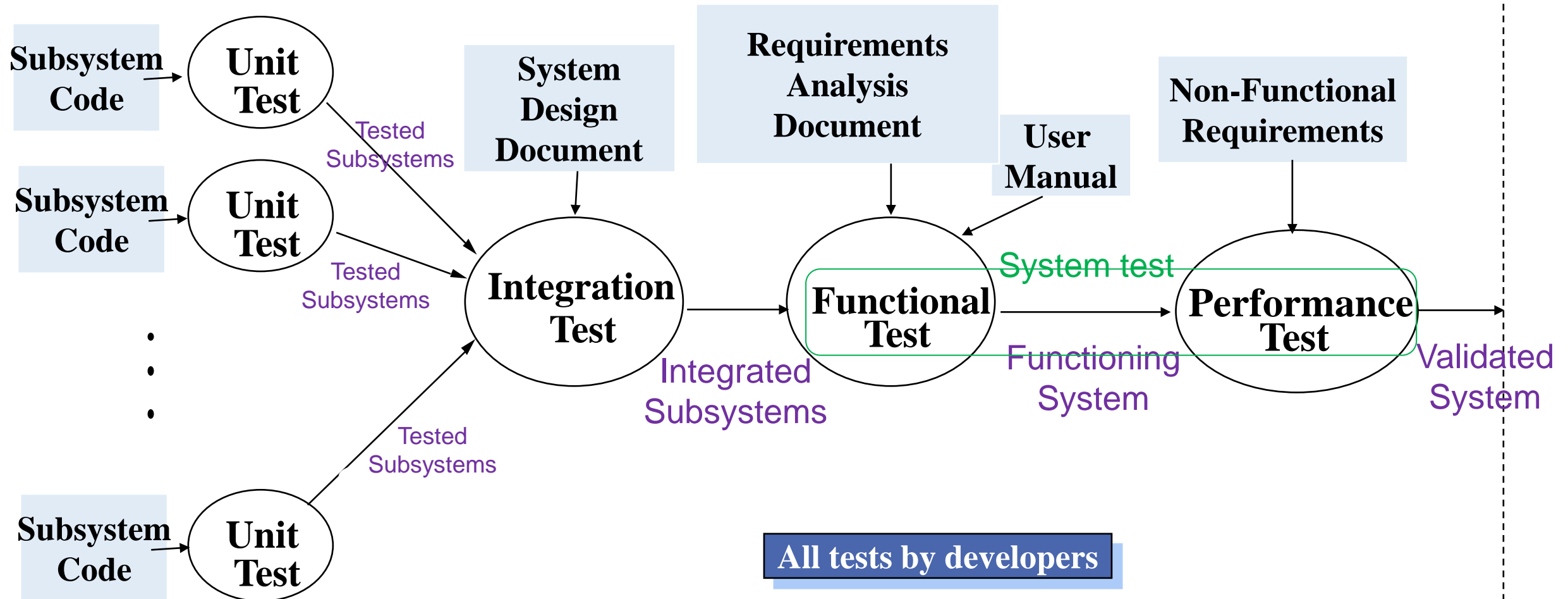
May 16, 2018

**SENG2130 Systems Analysis and Design**

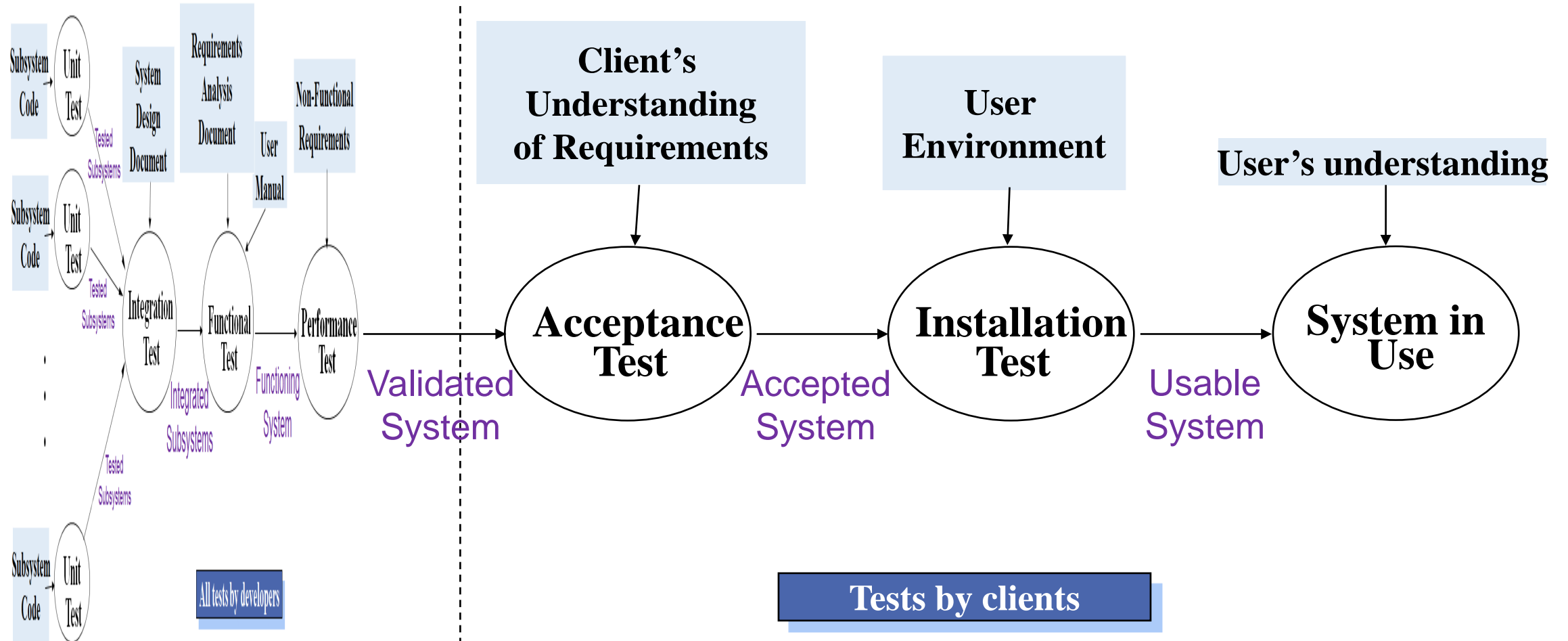
# Good Testing

- To develop an effective test, one must have:
  - Detailed understanding of the system
  - Knowledge of the testing techniques
  - Skill to apply these techniques in an effective and efficient manner
- Programmer often stick to the data set that makes the program work
  - Therefore, Testing is done best by **independent testers**
- A program often does not work when tried by somebody else.
  - Don't let this be the end-user.

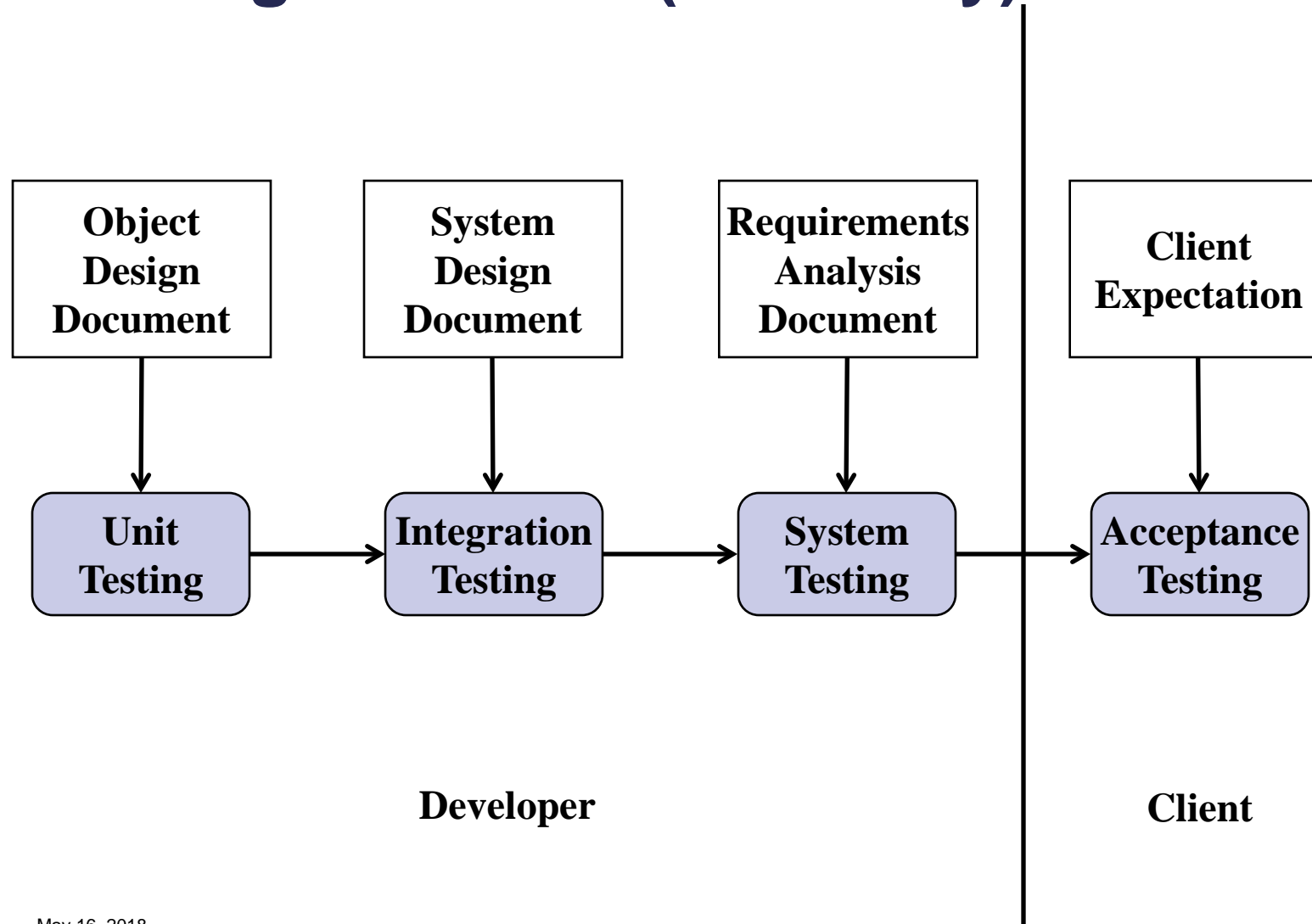
# Testing Activities

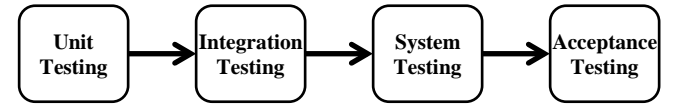


# Testing Activities



# Testing Activities (summary)





# Types of Testing

## Unit Testing

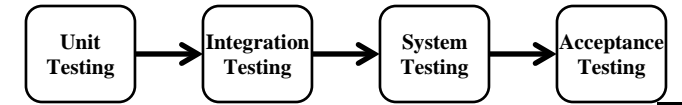
- Looks for errors in Individual subsystem or object
- Carried out by developers
- Goal: Confirm that subsystems is correctly coded and carries out the intended functionality

## Integration Testing

- Find errors with connecting subsystems together
- Carried out by developers
- Goal: Test the interface among the subsystem



# Types of Testing



9

## System Testing

- Test entire system behavior as a whole, with respect to scenarios and requirements
- Carried out by developers
- Goal: Determine if the system meets the requirements (functional and non-functional)

## Acceptance Testing

- Evaluates the system delivered by developers
- Carried out by the client. May involve executing typical transactions on site on a trial basis
- Goal: Demonstrate that the system meets customer requirements and is ready to use

# User Interface

May 16, 2018

**SENG2130 Systems Analysis and Design**

# User interface

- The user interfaces embody the data and functions of computer-based products
- User interfaces handle inputs and outputs that involve the system user *directly*
- Interactions with the user and computer (termed Human-Computer interactions or HCI) can be modeled with *dialog designs*
  - Use metaphors, standard guidelines, and UML diagrams to design user interfaces
- One of the important challenges to user interface design is how to help the novice user become quickly proficient and eventually become an expert user

# User vs. System interface

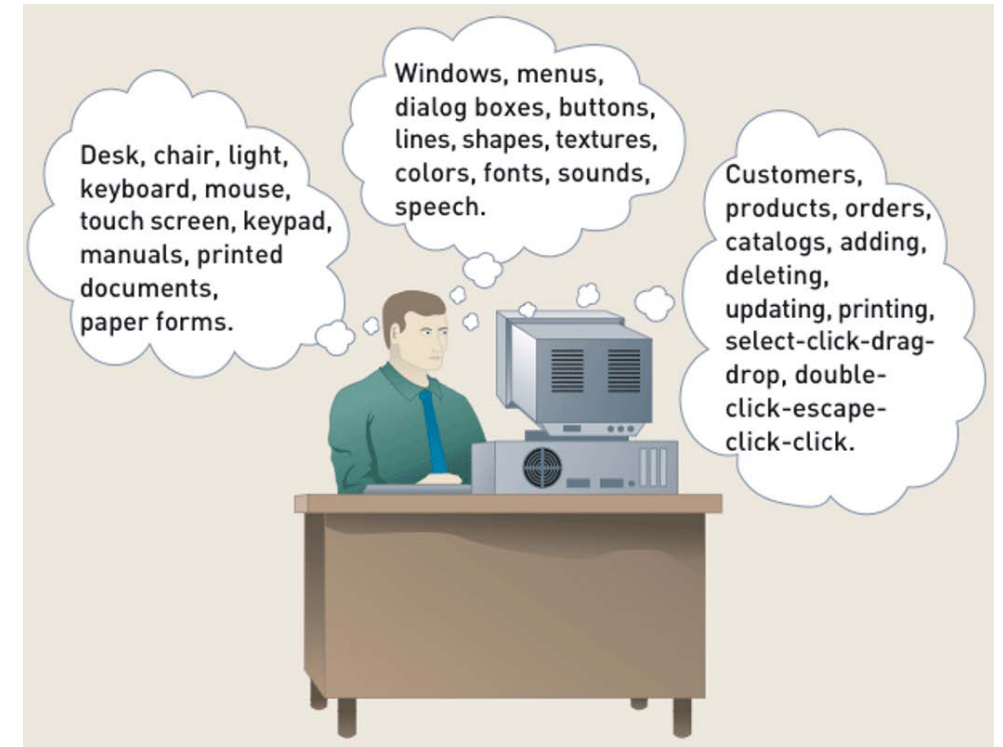
- System interfaces
  - Input / Output requiring minimal human interaction
- User interfaces
  - Requires user interaction to produce inputs and outputs
  - To the user, the interface is the system itself
    - Physical devices, parts, or documents
    - Perceptual aspects including seeing, hearing, and touching
    - Conceptual details about how to use the system
- Analyst designs system interfaces separate from user interfaces
  - Each requires different expertise and technologies

# Good vs. Bad design

- It is important to avoid bad design
  - It is often easy to detect a bad design – just try it with a few users
  - It can be fun to spot the flaws
    - UI Hall of Shame <http://hallofshame.gp.co.at/metaphor.htm>
    - UI's Greatest Bloopers
- It is much harder to teach / learn good design
  - Look at & appreciate good examples
  - Follow best practices
  - Be willing to redesign
  - Get lots of practice!

# Understanding the User Interface

- Physical aspects of the user interface
  - Devices touched by user, manuals, documentation, and forms
- Perceptual aspects of the user interface
  - Everything else user sees, hears, or touches such as screen objects, menus, and buttons
- Conceptual aspects of the user interface
  - What user knows about system and logical function of system



# User interface design principles

## User familiarity

- The interface should use terms and concepts which are drawn from the experience of the people who will make most use of the system.

## Consistency

- The interface should be consistent in that, wherever possible, comparable operations should be activated in the same way.

## Minimal surprise

- Users should never be surprised by the behaviour of a system.

## Recoverability

- The interface should include mechanisms to allow users to recover from errors.

## User guidance

- The interface should provide meaningful feedback when errors occur and provide context-sensitive user help facilities.

## User diversity

- The interface should provide appropriate interaction facilities for different types of system user.

# User-Centered Design

- Focus early on the users and their work by focusing on requirements
- **Usability** - system is easy to learn and use
- Iterative development keeps focus on user
  - Continually return to user requirements and evaluate system after each iteration
- **Human-computer interaction (HCI)**
  - Study of end users and interaction with computers
- **Human factors engineering (ergonomics)**



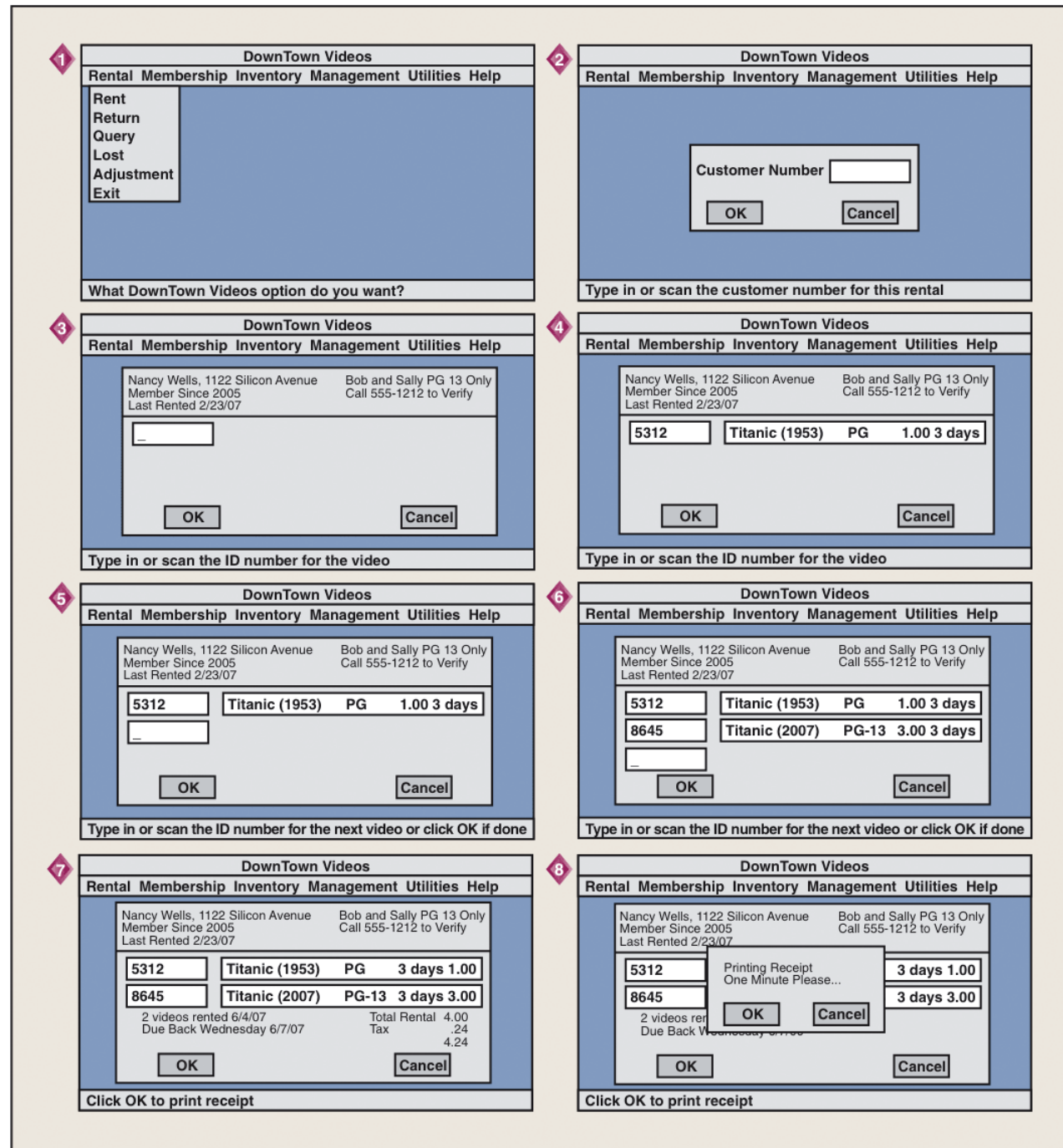
# Guidelines for Designing User Interfaces

- **Visibility**
  - All controls should be visible
  - Provide immediate feedback to indicate control is responding
- **Affordance**
  - Appearance of control should suggest its functionality – purpose for which it is used
- **Dialogs and Storyboards**
- **Metaphor:** like affordances, having a metaphor can tell you how you might use something

# Dialogs and Storyboards

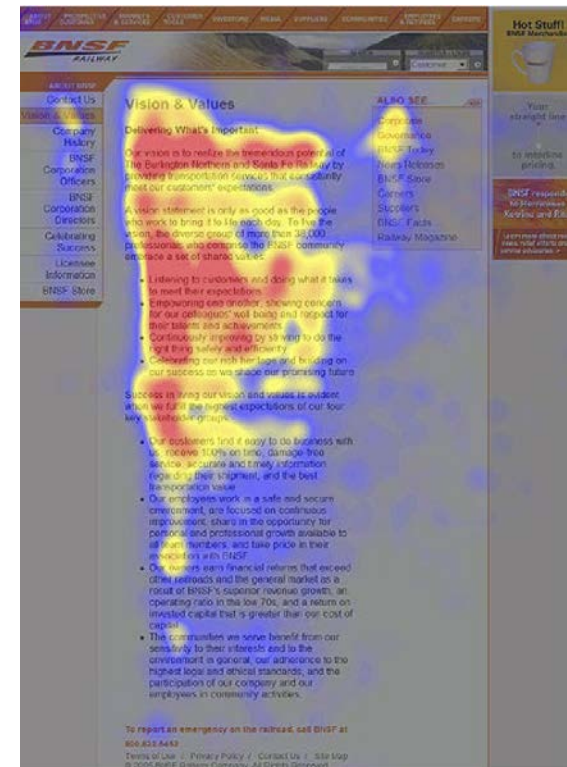
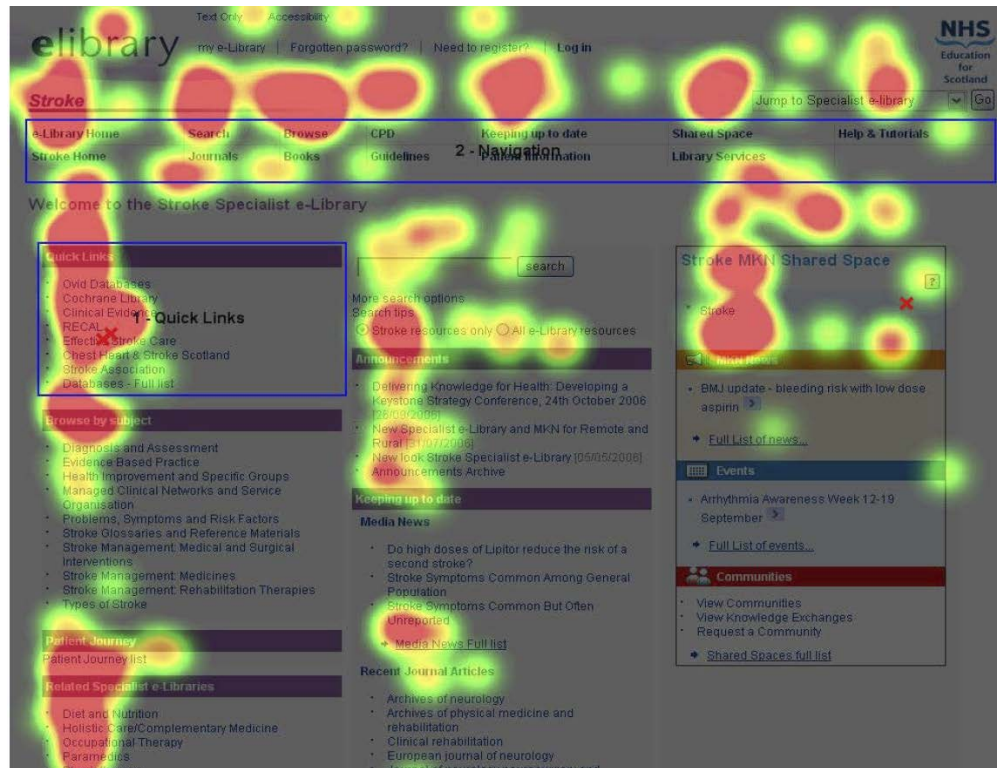
- Many methods exist for documenting dialogs
  - Written descriptions following flow of activities like in use case description
  - Narratives
  - Sketches of screens
  - Storyboarding – showing sequence of sketches of display screen during a dialog

- Storyboard for the Downtown Videos Rent Videos Dialog



# User behavior

## • Heat map evaluation



May 16, 2018

SENG2130 Systems Analysis and Design

# User behavior

- Find the price of a double room at the Holiday Inn in Bradley

Pennsylvania  
 Bedford Motel/Hotel: Crinaline Courts  
 (814) 623-9511 S: \$18 D: \$20  
 Bedford Motel/Hotel: Holiday Inn  
 (814) 623-9006 S: \$29 D: \$36  
 Bedford Motel/Hotel: Midway  
 (814) 623-8107 S: \$21 D: \$26  
 Bedford Motel/Hotel: Penn Manor  
 (814) 623-8177 S: \$19 D: \$25  
 Bedford Motel/Hotel: Quality Inn  
 (814) 623-5189 S: \$23 D: \$28  
 Bedford Motel/Hotel: Terrace  
 (814) 623-5111 S: \$22 D: \$24  
 Bradley Motel/Hotel: De Soto  
 (814) 362-3567 S: \$20 D: \$24  
 Bradley Motel/Hotel: Holiday House  
 (814) 362-4511 S: \$22 D: \$25  
 Bradley Motel/Hotel: Holiday Inn  
 (814) 362-4501 S: \$32 D: \$40  
 Breezewood Motel/Hotel: Best Western Plaza  
 (814) 735-4352 S: \$20 D: \$27  
 Breezewood Motel/Hotel: Motel 70  
 (814) 735-4385 S: \$16 D: \$18



# User behavior

- Find the price of a double room at the Quality Inn in Columbia

South Carolina					
City	Motel/Hotel	Area code	Phone	Rates	
				Single	Double
Charleston	Best Western	803	747-0961	\$26	\$30
Charleston	Days Inn	803	881-1000	\$18	\$24
Charleston	Holiday Inn N	803	744-1621	\$36	\$46
Charleston	Holiday Inn SW	803	556-7100	\$33	\$47
Charleston	Howard Johnsons	803	524-4148	\$31	\$36
Charleston	Ramada Inn	803	774-8281	\$33	\$40
Charleston	Sheraton Inn	803	744-2401	\$34	\$42
Columbia	Best Western	803	796-9400	\$29	\$34
Columbia	Carolina Inn	803	799-8200	\$42	\$48
Columbia	Days Inn	803	736-0000	\$23	\$27
Columbia	Holiday Inn NW	803	794-9440	\$32	\$39
Columbia	Howard Johnsons	803	772-7200	\$25	\$27
Columbia	Quality Inn	803	772-0270	\$34	\$41
Columbia	Ramada Inn	803	796-2700	\$36	\$44
Columbia	Vagabond Inn	803	796-6240	\$27	\$30

# Learn from User behavior

- User's attention span is short.
- They read the beginning and the end of the page
- People spend more time looking at the left side of your page
- People read your content in an F-shaped pattern



# Metaphor in User Interface Design

- Outside of the computer domain
  - Metaphors as being restricted to poetry and flowery writing
  - Shakespeare's As You Like It: "All the world's a stage ..."
- Within the computer domain
  - Desktop metaphor is well-known, and other kinds of graphical user interfaces are often consciously designed with a metaphor



# Metaphor in User Interface Design

- Metaphor
  - The transference of the relation between one set of objects to another set for the purpose of brief explanation
  - The fundamental concepts, terms, and images by which and through which information is easily recognized, understood, and remembered.
- A user feels comfortable when the mental model matches the real model

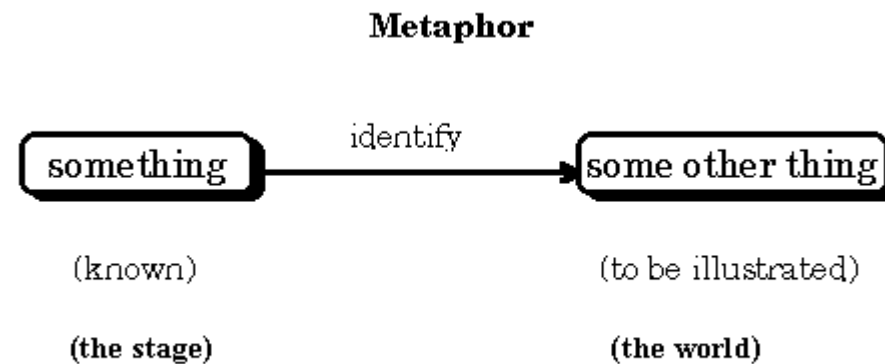


Figure 1.

# Metaphor: Guidelines for Design

- Interface Metaphors
  - A set user interface visuals, actions and procedures that exploit specific knowledge that users already have of other domains.
- The purpose of the interface metaphor
  - Is to give the user instantaneous knowledge about how to interact with the user interface.
- Provide a good conceptual model
  - allows users to predict consequences of actions
  - communicated through the image of the system

# More or less successful metaphors

- text editing as using a typewriter
- voice mail as **answering machine** or **mailbox**
- data as **files** (in folders or directories), represented as icons on desktop/in windows
- deleting a file as throwing it in the **trash**
- applications as **tools** (sometimes w/ icons)
- programming as building **objects**
- programming as directing actors on a **stage**
- applications as **agents**

# Metaphor in User Interface

- Metaphors for HCI
  1. Menus
  2. Desktop (Direct Manipulation)
    - Interaction with a display screen that includes objects commonly found on a desk (trash, folders, calculator ...)
  3. Browsing
  4. Forms
  5. Immersive environments

# 1. Menu

- Users are given predetermined choices
- Are often part of a **WIMP** interface (point & click)
  - Windows and window managers
  - Icons
  - Menus
  - Pointing devices

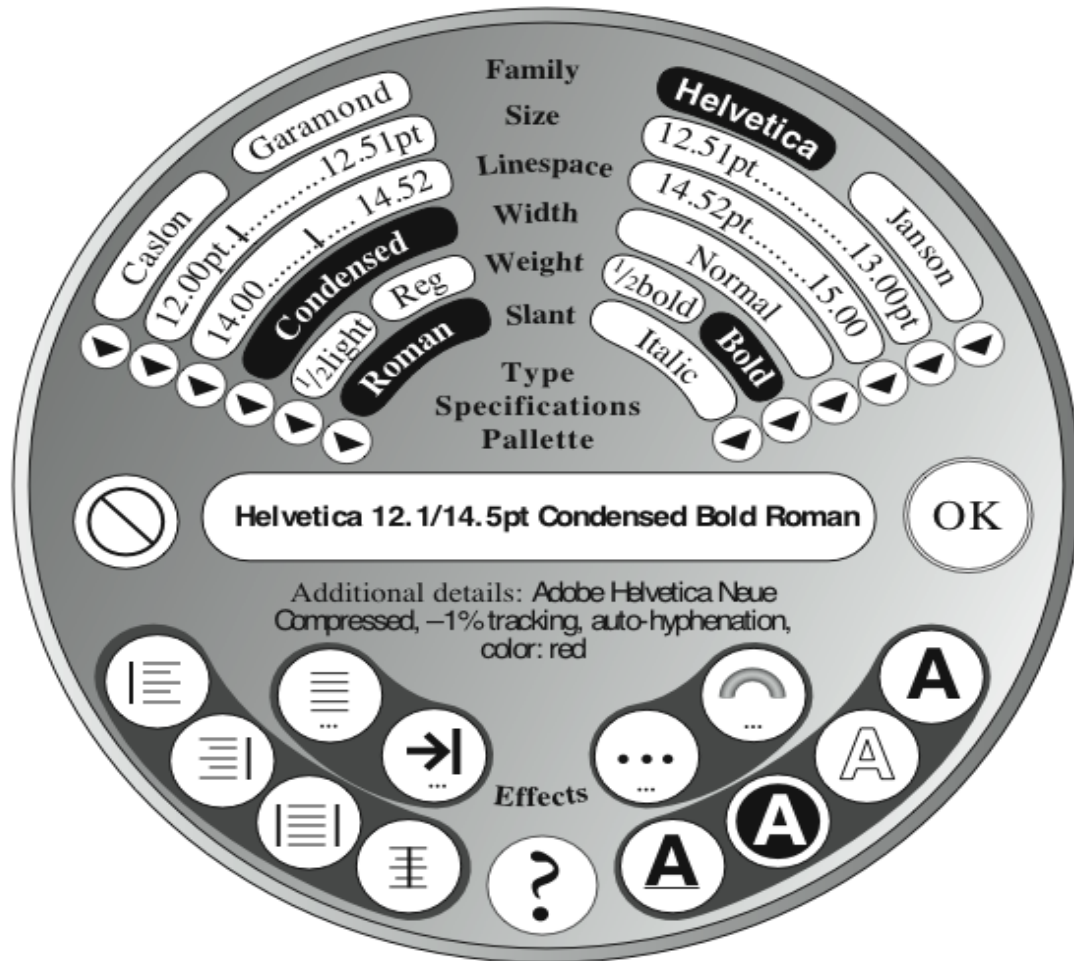
# Menus

- enforce a hierarchy on the user's goals
- You cannot filter and combine the world in novel ways
- Decisions in menu design
  - What is the task hierarchy?
  - How to order or group items?
  - How to trade off depth vs. breadth?
  - What labels to choose?
  - Whether to include shortcuts?

# Menu types

- Pull-down
- Fall-down
- Pop-up
- Multiple selection
- Radio buttons

# Which one do you prefer?



PLEASE SPECIFY TYPE

Family  
Helvetica ▶

Size  
12pt ▶

Linespace  
14pt ▶

Width  
Condensed ▶

Weight  
Bold ▶

Slant  
Roman ▶

Alignment  
☐ ☐ ☐ ☐

Effects  
Reverse ☐ ☐ Outline  
Shadow ☐ ☐ Underline

Block... Special... Tab...

Helvetica 12/14pt Condensed Bold Roman

Help OK Cancel



## 2. Direct Manipulation uses a Metaphor

- Metaphor
  - Computer objects as visible, moveable objects
- Consequences
  - Items represented as icons
  - Items can be “picked up” and “moved” on a surface
  - Items can be “thrown out”
  - Items can be “copied”

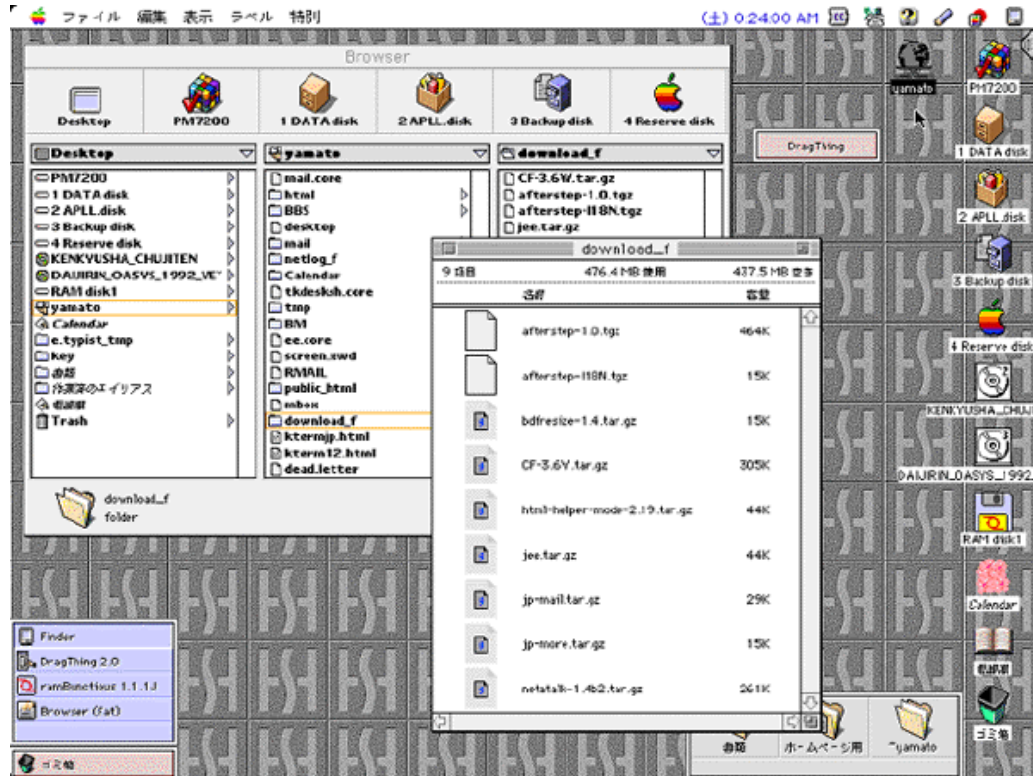
# Direct Manipulation uses a Metaphor

- Representations behave as if they were the objects they represent
- This reduces the distance between users and their goals
- **What You See Is What You Get (WYSIWYG)**
- **Physical actions, not complex syntax**
  - Eg, > copy file.txt to b:\file.txt

Versus

**Dragging a file icon to a diskette icon**

# Macintosh desktop



# Microsoft Bob's desktop



# Beyond the Desktop

36

- Robertson, George et al. "*The Task Gallery: A 3D Window Manager.*" In Proceedings of CHI 2000



# Beyond the Desktop

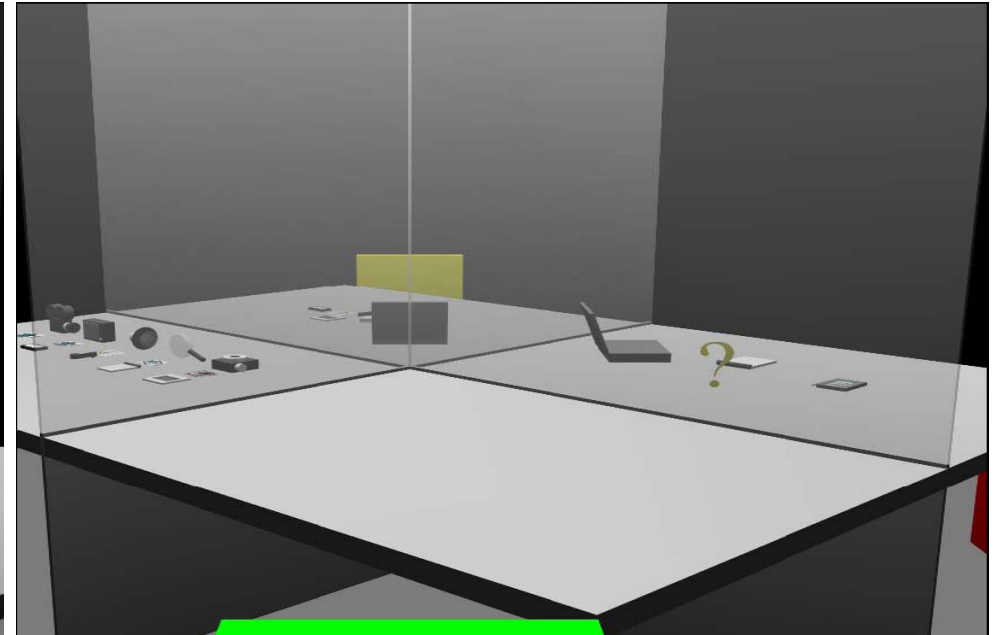
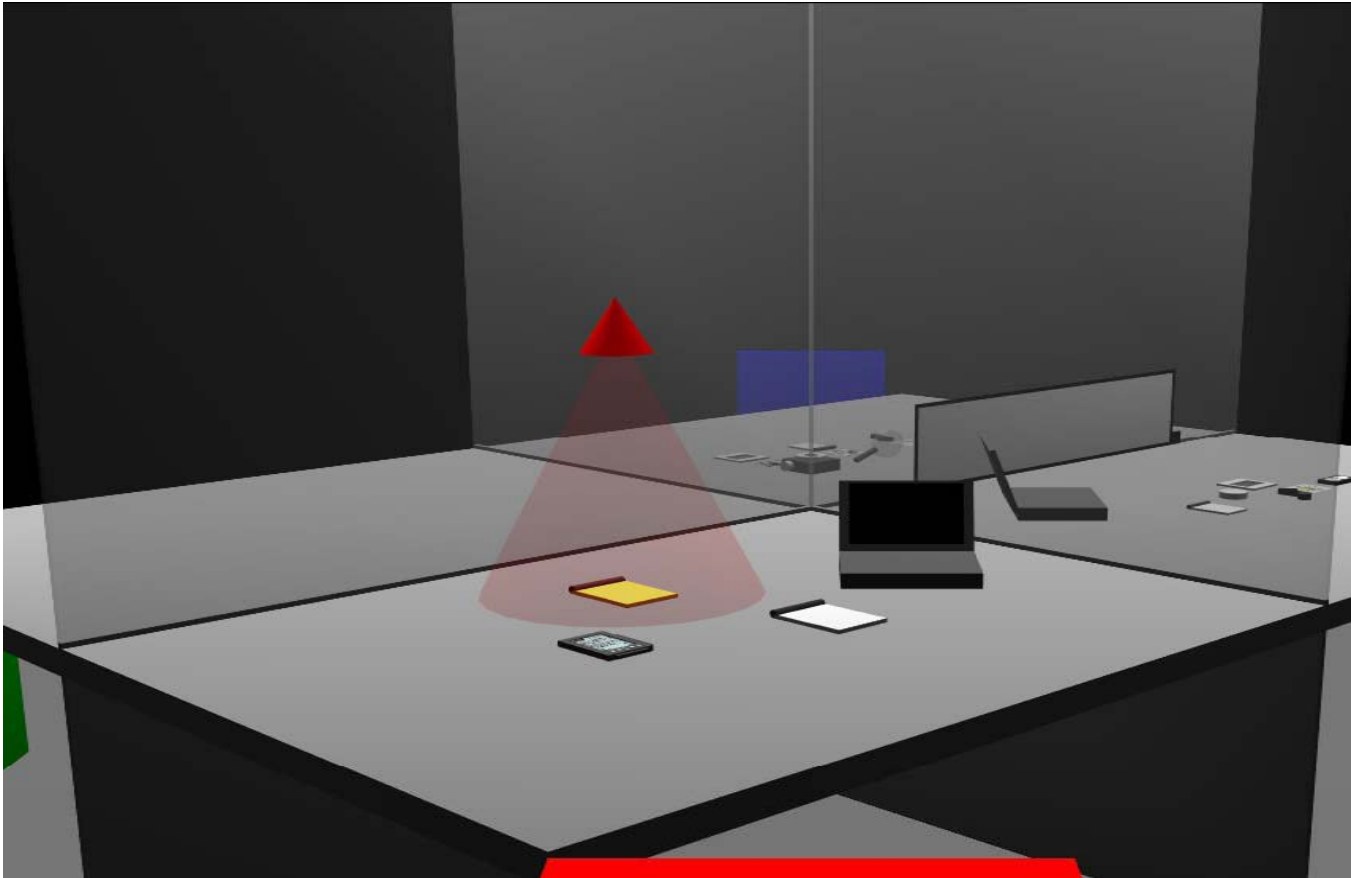
37



May 16, 2018

SENG2130 Systems Analysis and Design

# Example: Privacy Lamp



- Intuitive physical metaphor
- Several objects under one lamp
- Height of lamp determines size of light cone
- Inversely as publicity lamp



# Identify the mis-matched metaphors

- The classic (from the mac desktop)
  - To eject a disk you drag it to the trashcan
- VCR buttons to control a printer??



# Some problems?

- Sometimes: information overload or screen clutter
- Not all tasks can be represented by objects
  - it's hard to represent abstract things
  - DM must function in the “here” and “now”
  - icons can be just as cryptic as words (*the Vocabulary Problem: stay tuned*)



### 3. Browsing

- Similar to how people browse information with existing media (e.g. newspapers, magazines, libraries, pamphlets)
- Information is structured to allow flexibility in way user is able to search for information
  - e.g. multimedia, web
- Browsers, hypertext, links

According to a research study at Cambridge University, it doesn't matter in what order the letters in a word are, the only important thing is that the first and last letter be in the right place. The rest can be a total mess and you can still read it without problem. This is because the human mind does not read every letter by itself, but the word as a whole.

## 4. Forms

- Forms can vary from over-determined (enforcing a fixed order) to quite flexible (allowing user initiative)

# Form Design Issues

## 1. Keep it short.

Remove fields if the information can be derived or just omitted.

## 2. Visually group related labels and fields.

Labels should be close to their field and clear about which field is related.

## 3. Use logical sequencing.

## 4. Match fields to the type and size of the input.

## 5. Distinguish optional and required fields.

Cardholder name\*



Debit/Credit card number\*

Expiration date\*

Month ▼

Year ▼

Security code\*

# Form Design Issues

## 6. Explain any input or formatting requirements.

**NETGEAR®  
SUPPORT**

DOWNLOADS MyNETGEAR COMMUNITY CONTACT US

MyNETGEAR / Reset Password

### Change Your Password

You can reset the password for your account by completing this form

New Password:

.....

Password must contain atleast one uppercase letter [A-Z], atleast one lowercase letter [a-z], atleast one number [0-9] and have a minimum length of 6 characters. Allowed symbols are !@#%\*&\*()

Re-enter new password:

|

**SUBMIT**

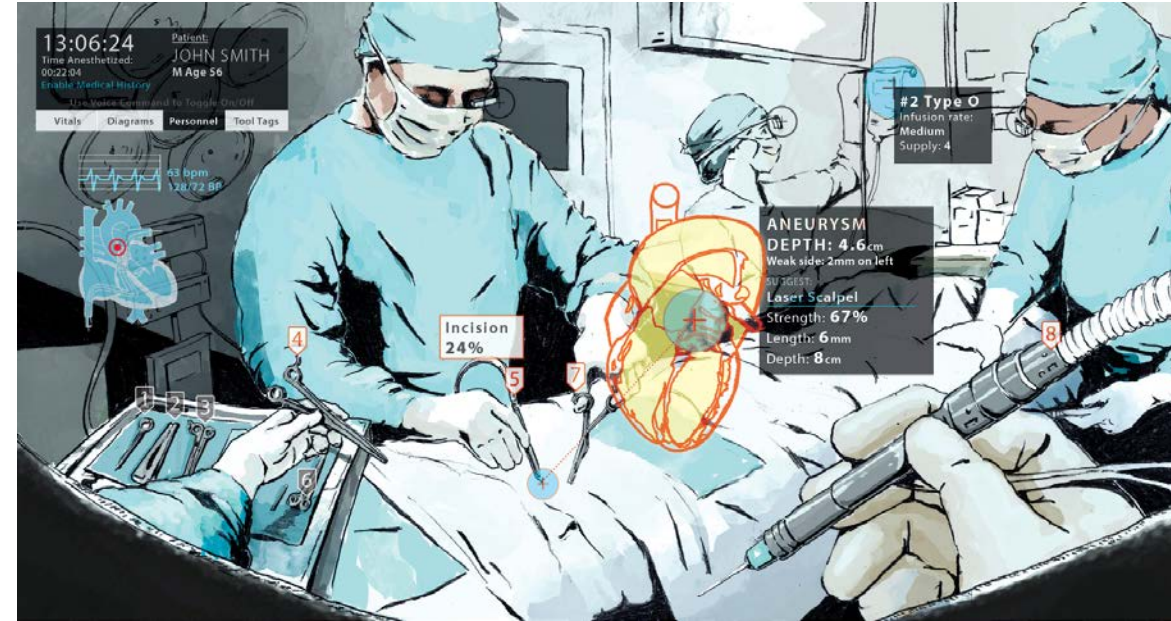
## 7. Keep *Reset/Clear* buttons well away from *Submit* button.

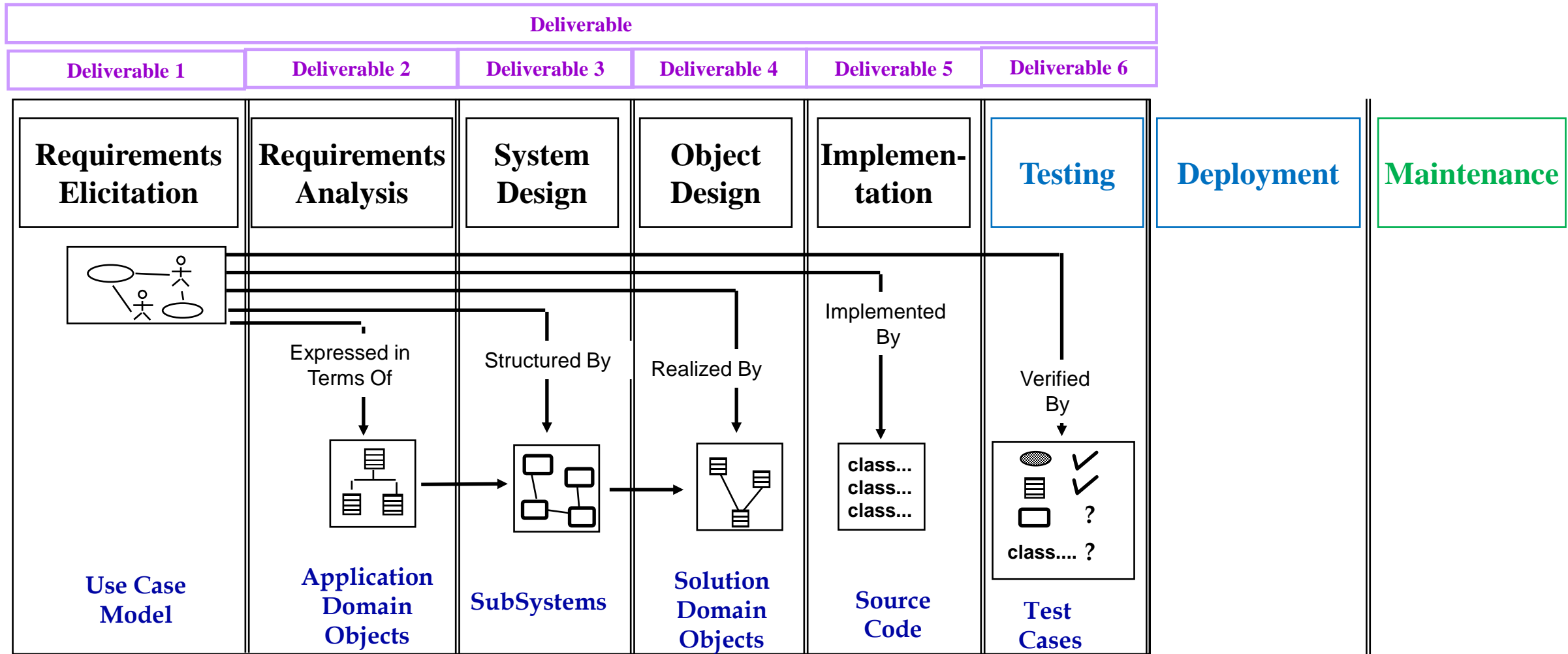
Maybe even use different colours.

## 8. Provide highly visible and specific error messages.

## 5. Immersive environments

- Virtual reality
- Often mimic real life (representations are highly literal; go well beyond metaphors)





# Deployment

May 16, 2018

**SENG2130 Systems Analysis and Design**

# Deployment

- After development and testing, system must be put into operation
- Deployment is the placing of software on the hardware where it is supposed to run
- Important planning considerations
  - Costs of operating both systems in parallel
  - Detecting and correcting errors in new system
  - Potentially disrupting the company and IS operations
  - Training personnel and customers with new procedures

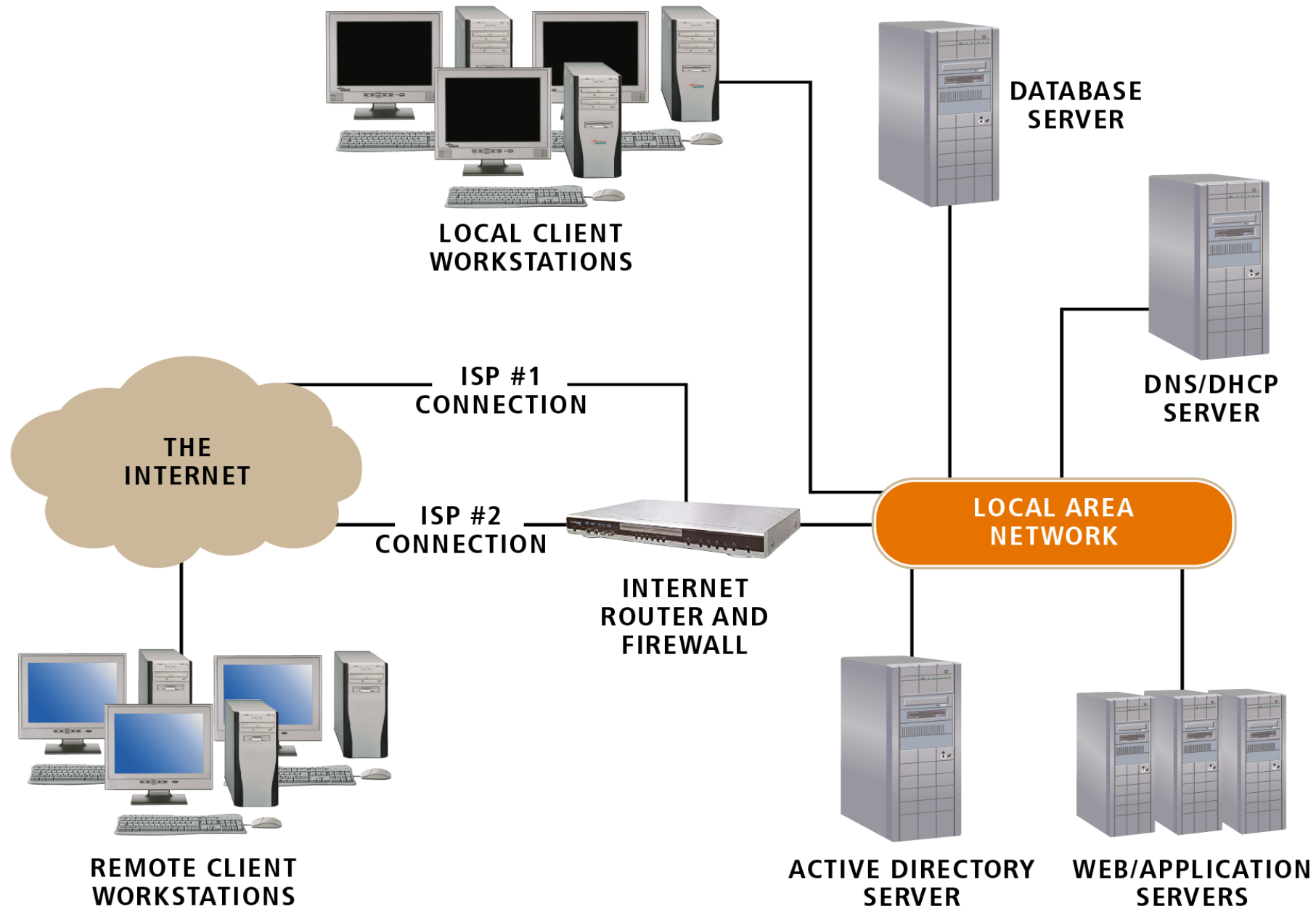


# Deployment Basic Concepts

- Software system is deployed on one or more hardware devices called hosts or sites.
- Each site provides some resources.
- E.g. –
  - Hardware (Memory, CPU)
  - Network (Protocols and IP port numbers)
  - Peripheral Devices (hard disks and keyboard)
  - System Software (OS, drivers)
  - Other application level software (GUI builders, DB)
  - Data Resources (data files, database)

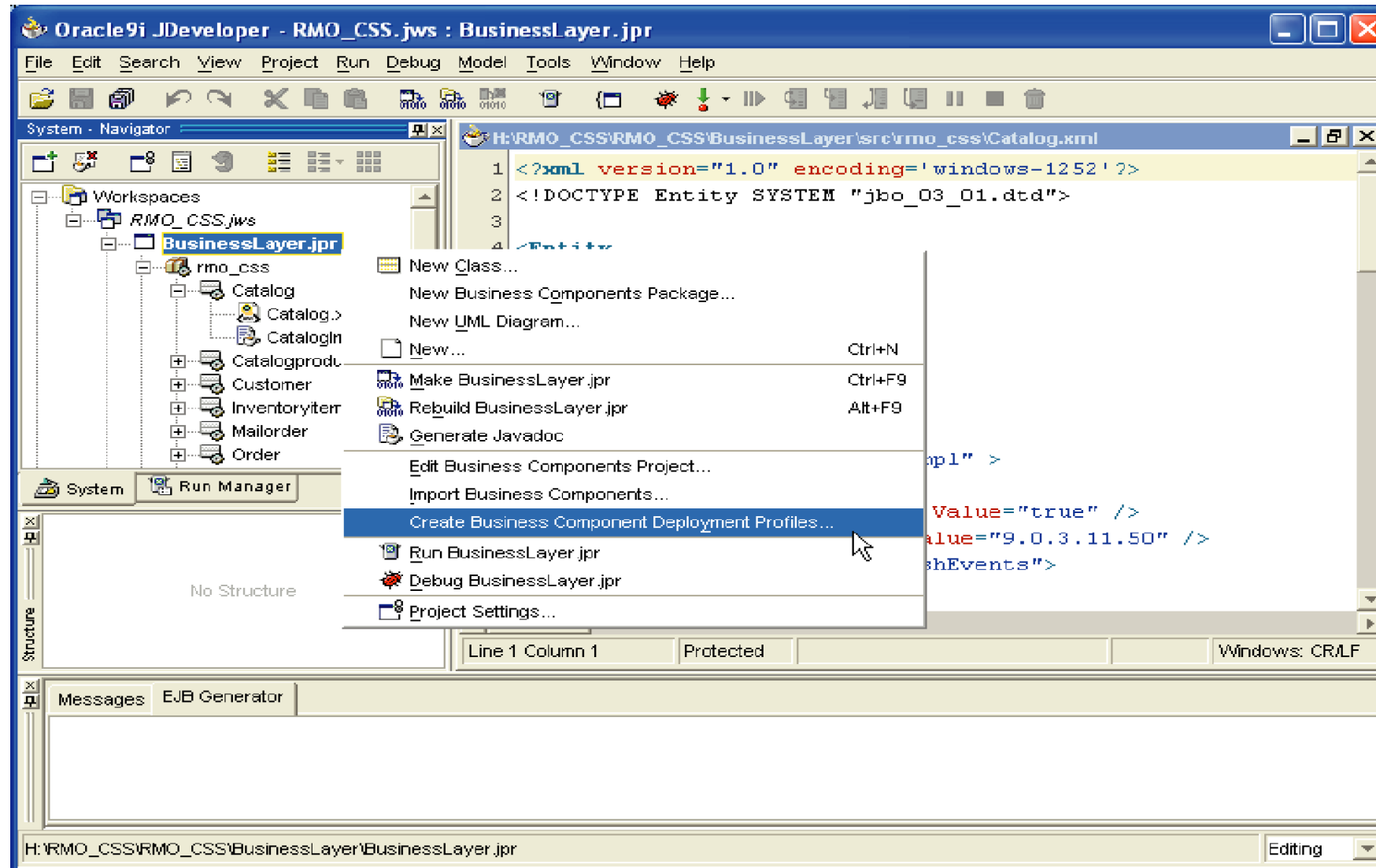
# Acquiring Hardware and System Hardware

- Application software must have a supporting infrastructure (which may already be in place)
- Acquisition of an entirely new infrastructure includes
  - Planning
  - Developing a request for proposal
  - Evaluating results
  - Choosing one or more vendors
  - Installation and configuration



# Packaging and Installing Components

- Components must be
  - Installed on a host server
  - Added to a component registry
  - Assigned one or more network addresses
- May include XML files to store registration and access information
- Developers can package and install components using development tools and utilities



## Automated component deployment with Oracle Developer

May 16, 2018

SENG2130 Systems Analysis and Design

# Training Users

- End users and system operators need training
- End user training
  - Hands-on training and tutorials
  - Group tutorials
- System operator training
  - Less formal
  - Self-study
- Training materials are developed as soon as the interfaces are reasonably stable

# Converting and Initializing Data

- Data needed at system startup can be obtained from
  - Files or databases of a system being replaced
  - Manual records
  - Files or databases of other systems in the organizations
  - User feedback during normal system operation
- Existing databases are commonly modified for reuse in new or upgraded systems

# Deployment plan

- Direct Deployment
- Parallel Deployment
- Phased Deployment
- Pilot Deployment
- Each deployment approach has strengths and weaknesses

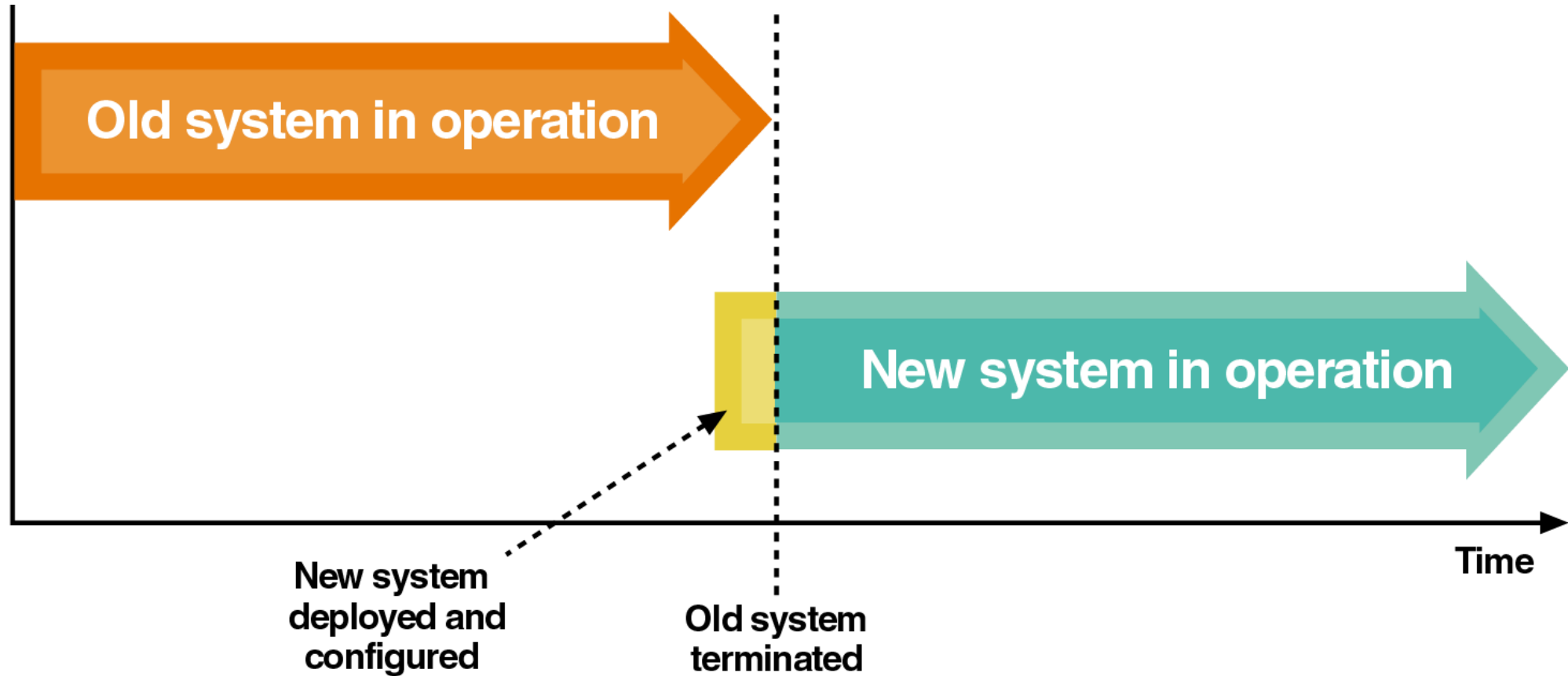


# Direct Deployment

- Installs a new system, quickly makes it operational, and immediately turns off any overlapping systems
- Advantages
  - Simplicity
- Disadvantages
  - Risk of system unavailability
- Used when a new system is not replacing an old system and/or downtime can be tolerated

# Direct Deployment and cutover

58

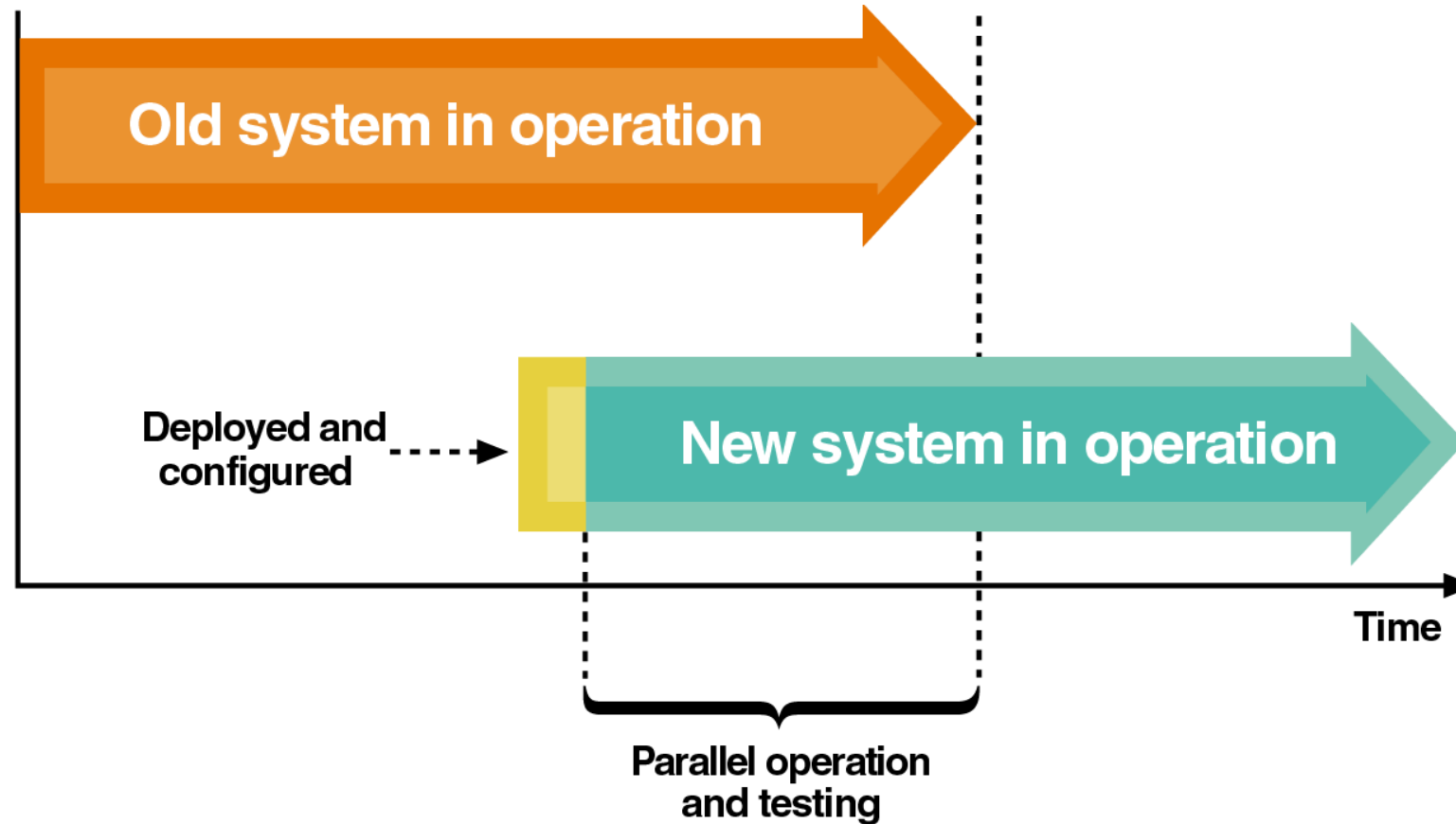


# Parallel Deployment

- Operates both old and new systems for an extended time period
- Advantages
  - Relatively low risk of system failure
- Disadvantage
  - Cost to operate both systems
- Used for mission-critical applications
- Partial parallel deployment can be implemented with increased risk of undetected errors

# Parallel Deployment and operation

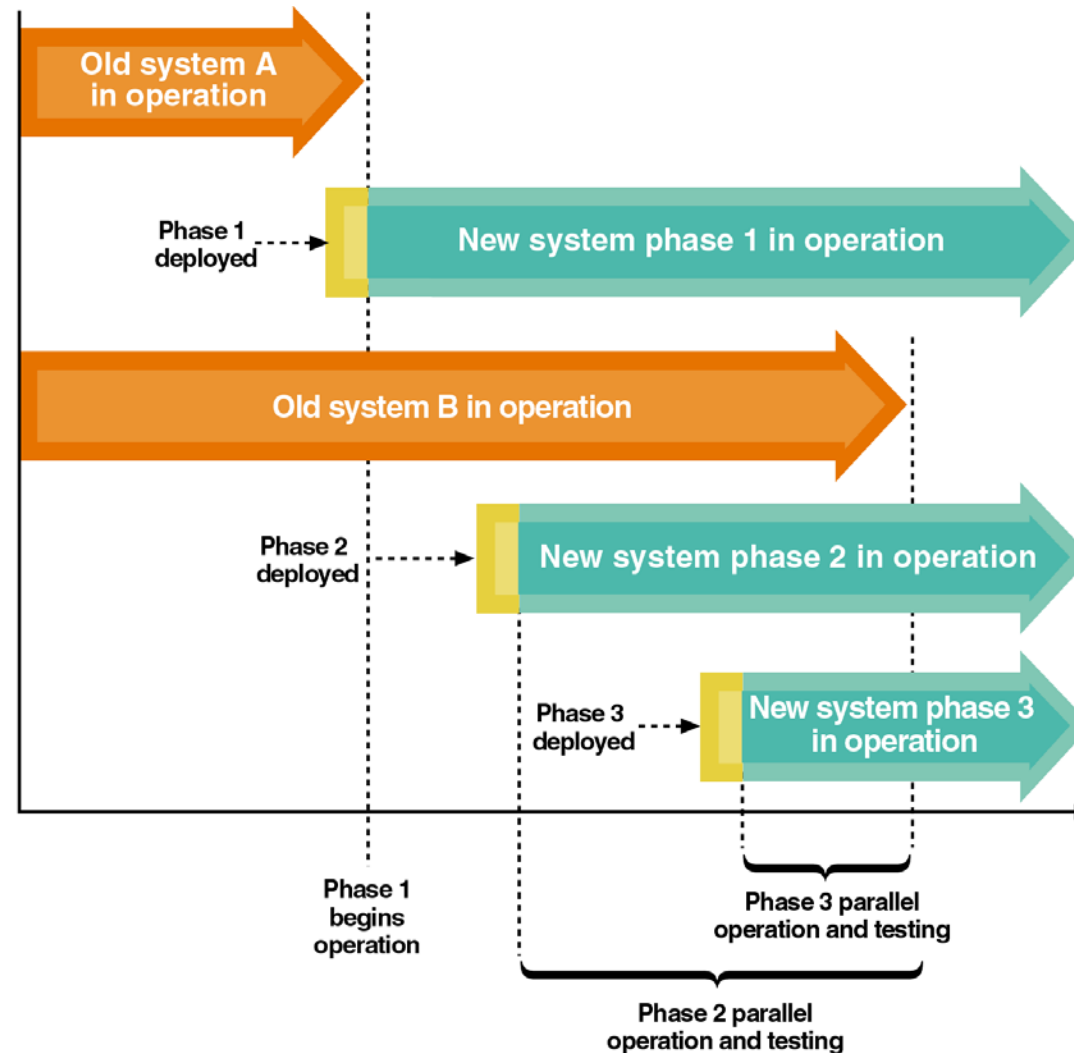
60



# Phased Deployment

- Installs a new system and makes it operational in a series of steps or phases
- Advantages
  - Reduced risk
- Disadvantages
  - Increased complexity
- Useful when a system is large, complex, and composed of relatively independent subsystems

# Phased deployment with direct cutover and parallel operation



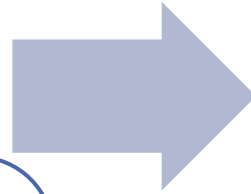
# Pilot Deployment

- Involves having a few offices, departments or other organization units "try" or "test" a prototype or the full system.
- Advantages
  - Reduced risk, feedback, gain deployment experience and validate the application
- Disadvantages
  - The cost to operate the pilot system (extra time, personnel and operations as data is input into the pilot systems)
- Once the pilot has been evaluated as a success, the implementation team would expand the pilot with phased adoption, direct changeover or parallel running.
- A pilot is generally an appropriate initial deployment strategy for large-scale, enterprise-wide applications.

# Pilot Deployment

## Pilot deployment

- July 2017
- Recipients
  - Approx. 50 users in 2 locations
- Accomplishments
  - Provide successful installation
  - Obtain user feedback for final release at full deployment
  - Show users a successful rollout



## Full deployment

- September ~ October 2017
- Recipients
  - Approx. 500 users in 20 locations
- Accomplishments
  - Rolled-out new system release incorporating pilot feedback
  - Deploy the full system to all end-users
  - Give benefit of pilot experience to deployers and end-users



# Summary

- System Testing: Unit test, Integration test, System test, Acceptance test
- User Interface Design
  - Metaphor
- Deployment activities Acquiring hardware and system software
  - Direct Deployment
  - Parallel Deployment
  - Phased Deployment
  - Pilot Deployment