ITM 5400
Systems Analysis, Design, and Implementation

Dan Brandon, Ph.D., PMP

Copyright Dan Brandon, PhP, PMP
Phased Approach to System Development

- Planning
- Analysis
- Design
  - User Interface
  - Database
  - System Architecture
- Implementation
- Systems Support
Session Objectives

System Architecture

- Provide issues to consider when selecting a system architecture
- Describe servers, server-based processing, clients, and client-based processing
- Explain client/server architecture
- Compare in-house development with packaged solutions
- Discuss the potential impact of cloud computing and Web 2.0
- Explain the difference between online and batch processing
- Define network topology, including hierarchical, bus, ring, and star models
- Explain network protocols and licensing issues
- Describe wireless networking, including wireless standards, topologies, and trends
- Describe the system design specification
System Architecture

• An effective system combines elements into an architecture that is:
  • Flexible (agile – easily modified)
  • Cost-effective
  • Technically sound (usable, maintainable)
  • Able to support both the current and future information needs of the business (scalable)

• System architecture translates the logical design of an information system into a physical structure
Rating an Architecture

![Radar Chart](chart.png)

- Functionality
- Expandability
- Ergonomics / Usability
- Technical Scalability
- Maintainability
- Agility
System Architecture Checklist

Before translating the logical design of an information system into a physical structure, the analyst should consider several issues:

- Enterprise resource planning (ERP)
  - Company-wide strategy for using IT resources
- Total cost of ownership
- Scalability
- Legacy systems and other interface requirements
- Processing options
- Data communication options
- Security issues
- Web and mobility integration
Planning the Architecture

- Architecture options:
  - Mainframe
  - File servers
  - Clients/server
    - Fat client
    - Thin client
    - 2 or 3 Tier
  - Web based
    - In-house server
    - Cloud based
As PC technology exploded in the mid-1980s and 1990s, powerful microcomputers quickly appeared on corporate desktops.

Users found that they could run their own word processing, spreadsheet, and database applications.

Companies linked the stand-alone computers into local area networks (LANs).

LAN’s were then connected to corporate resources via WANs.
LAN/WAN
When a client (PC) makes a request such as:

- Display customers in zip code 38104
  - SELECT * FROM CUSTOMER WHERE ZIP=38104

How is this processed in a LAN architecture (such as Microsoft Access file residing on a network server via file redirection), as opposed to a client server network architecture (such as Microsoft SQL Server)?
Don’t look ahead ...
File Server vs Client/Server

[Select * from Customer where zip = 38104]
The web environment is a form of client/server computing.
## Fat and Thin Clients

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Fat Client</th>
<th>Thin Client</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network traffic</td>
<td>Higher, because the fat client must communicate more often with the server to access data and update processing results</td>
<td>Lower, because most interaction between code and data takes place at the server</td>
</tr>
<tr>
<td>Performance</td>
<td>Slower, because more network traffic is required</td>
<td>Faster, because less network traffic is required</td>
</tr>
<tr>
<td>Initial cost</td>
<td>Higher, because more powerful hardware is required</td>
<td>Lower, because workstation hardware requirements are not as stringent</td>
</tr>
<tr>
<td>Maintenance cost</td>
<td>Higher, because more program code resides on the client</td>
<td>Lower, because most program code resides on the central server</td>
</tr>
<tr>
<td>Ease of development</td>
<td>Easier, because systems resemble traditional file-server designs where all processing was performed at the client</td>
<td>More difficult, because developers must optimize the division of processing logic</td>
</tr>
</tbody>
</table>
Client/Server Tiers

- Two-tier design
  - Server provides both application logic and database management software

- Three-tier design
  - There is a separate application server and database server
Cost-Benefit Issues

- Client/server systems enable the firm to scale the system in a rapidly changing environment.
- Client/server computing also allows companies to transfer applications from expensive mainframes to less expensive client platforms.
- Client/server systems reduce network load and improve response times.
- Client/server systems provide a richer user experience.
Client/Server Performance Issues

- In contrast to the centralized system, a client/server design separates applications and data.
- The system is scalable, so new data sites can be added without reworking the system design.
- The system is less likely to experience catastrophic failure.
Data Processing Modes

- Input transaction data can be processed (master files/tables updated) in several manner
- The processing involves updating the master files (tables) – changing master file data to reflect the impact of the transaction
- Three possible modes are:
  - Batch
  - Online
  - Hybrid
Batch Processing

- **Batch Processing** is when:
  - Transaction data is collected as it occurs.
  - The transactions are placed in groups or *batches*.
  - Then those batches are processed periodically (typically every night).

- Safer, more “controlled”

- But less powerful inquiry
  - Users can only see the data after it has already been processed into the master files (tables).
Traditional IT Batch TPS
Online Processing

- **Online Transaction Processing (OLTP)** is when business transactions are processed against the master files (tables) online as soon as they occur.

- Riskier, but provides better inquiry.

**ATM QUERY PROCESS**

1. **Step 1:** Customer enters his or her account number and requests an account balance.

2. **Step 2:** Retrieves current account balance.

3. **Step 3:** Verifies bank account number and displays balance on ATM screen.
Hybrid Online and Batch Processing
Computer Data Communications

Diagram showing the flow of data between a client and a server, passing through communications processors and channels and media.
TCP/IP

- **Internet Protocol (IP)**: the set of rules used to send and receive packets from one machine to another over the Internet.
- **TCP**: the control portion of the overall TCP/IP protocol.
- **Packet switching**: is a transmission technology that breaks up blocks of text into small, fixed bundles of data called packets.
Packet Switching

[packets of a single message may follow different paths]
Computer Networking Media

- **Wired Media**
  - Twisted Pair – 100 Mbps (million bits per second)
  - Coaxial Cable – 200 Mbps
  - Fiber Optic – 6 Tbps (trillion bits per second)

- **Broadcast (wireless) Media** (up to 200 Mbps)
  - Microwave
  - Satellite
  - Radio
  - Cellular Radio
  - Wi-fy, Wi-max
  - Infrared

Copyright – Dan Brandon
Wired Media

Pro’s and con’s of each?
Don’t look ahead …
# Advantages and Disadvantages of Wireline Communications Channels

<table>
<thead>
<tr>
<th>Channel</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Less susceptible to electromagnetic interference.</td>
<td></td>
</tr>
<tr>
<td>Fiber-optic cable</td>
<td>Very high bandwidth. Relatively inexpensive. Difficult to tap (good security).</td>
<td>Difficult to work with (difficult to splice).</td>
</tr>
</tbody>
</table>
LAN Wiring

- Within a LAN, the typical wiring used is a special type of twisted pair.
- **CAT 5** is the most commonly used today.
- This standard provides performance of up to 100 MHz and is suitable for **10BASE-T**, **100BASE-TX** (Fast Ethernet), and **1000BASE-T** (Gigabit Ethernet); Cat 5 is also used to carry other signals such as telephony and video.
- Most category 5 cables are unshielded, relying on the balanced line twisted pair design and differential signaling for noise rejection.
- Category 5 has been superseded by the **category 5e** (enhanced) specification and **category 6 cable**.

Copyright Dan Brandon, PhP, PMP
CAT 5 LAN Wiring

<table>
<thead>
<tr>
<th>Pin</th>
<th>Pair</th>
<th>Wire</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td>white/green</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td>green</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td>white/orange</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2</td>
<td>blue</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>white/blue</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>2</td>
<td>orange</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>1</td>
<td>white/brown</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>2</td>
<td>brown</td>
</tr>
</tbody>
</table>
Broadcast (wireless) Media (200 Mbps)

- Wireless
  - Microwave
  - Infrared
  - Satellite
  - Radio
  - Cellular
    - 3G
    - 4G - Wi-max & LTE
  - Wi-fy
  - Bluetooth
Wireless Transmission Media

- **Infrared** light is red light that is not commonly visible to human eyes
  - Low volume (short distance); common uses in remote control units for TVs, VCRs, DVDs, CD players - **short distance**

- **Microwave transmission systems** are widely used for high-volume, **medium-distance**, and point-to-point
  - *Point-to-point communication* has two characteristics:
    - first, the transmitter and receiver must be in view of each other (called **line-of-sight**)
    - and second, the transmission itself must be tightly directed from transmitter to receiver

- **Radio transmission** uses radio-wave frequencies to send data directly between transmitters and receivers (**longer distances**)
Radio waves travel very far, but what is the limitation in using radio waves for data communication?
Don’t look ahead …
Satellite Transmission

- **Satellite transmission** systems make use of communication satellites
  - **Geostationary earth orbit (GEO)** orbits 22,300 miles directly above the equator and maintains a fixed position; excellent for TV (one way) signals
  - **Medium earth orbit (MEO)** are located 6,000 miles above the earth’s surface and move; used for GPS and are less expensive
  - **Low earth orbit (LEO)** are 400 to 700 miles above the surface and move much quicker so many are needed to have adequate coverage; used mainly for telephone (two way)
Satellite Transmission (Con’t)

- **Footprint** is the area of earth’s surface reached by a satellite’s transmission – overcomes the limitations of microwave data relay stations.

- **Broadcast transmission** allows satellites to send signals to many receivers at one time.

- **Propagation delay** is a brief pause in transmissions from satellites which make two-way telephone conversations difficult with higher satellites.
<table>
<thead>
<tr>
<th>Channel</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microwave</td>
<td>High bandwidth. Relatively inexpensive</td>
<td>Must have unobstructed line of sight. Susceptible to environmental interference.</td>
</tr>
<tr>
<td>Satellite</td>
<td>High bandwidth. Large coverage area.</td>
<td>Expensive. Must have unobstructed line of sight. Signals experience propagation delay. Must use encryption for security.</td>
</tr>
<tr>
<td>Radio</td>
<td>High bandwidth. Signals pass through walls. Inexpensive and easy to install.</td>
<td>Creates electrical interference problems. Susceptible to snooping unless encrypted.</td>
</tr>
<tr>
<td>Infrared</td>
<td>Low to medium bandwidth.</td>
<td>Must have unobstructed line of sight. Used only for short distances.</td>
</tr>
</tbody>
</table>
Bluetooth

- **Bluetooth** is used to create small PANs:
  - can link up to 8 devices within a 10-meter area, and newer versions extend distance
  - uses low-power, radio-based communications
  - can transmit up to 1 Mbps

- **Personal area network (PAN)** is a computer network used for communication among computer devices (e.g., telephones, PDAs, smart phones) close to one person
  - Security issues even in the PAN
Zigbee targets applications that need low data transmission rates and low power consumption:

- Moves data only one-fourth as fast as Bluetooth
- Can handle hundreds of devices at once
- One promising application is meter reading

Current focus is to wirelessly link sensors that are embedded into industrial controls, medical devices, smoke and intruder alarms, and building and home automaton
Wireless Local Area Networks (Wi-Fi)

- **WLAN** (IEEE 802.11 or Wi-Fi) requires a transmitter with an antenna, called a **wireless access point**, that connects to a wired LAN (or to satellite dishes) that provide an Internet connection.

- **Wireless network interface card (NIC)** is needed to communicate wirelessly and has a built-in radio and antenna – typically USB or card (slot).
Wireless LAN

Wireless network card
With a wireless network card installed into your laptop you can effortlessly connect to your wireless network.

Wireless desktop card
The wireless desktop card is a great way to enable your desktop computer with wireless.

wireless USB adapter
Your notebook in the kitchen is fitted with a USB adapter. An easy and convenient way to get onto your network.

ADSL 2/2+ wireless modem/router
Your wireless router is situated in your office/study and is your main point of access for your wireless network.
Wi-Fi - Wireless Fidelity (802.11) Protocols

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Release Date</th>
<th>Op. Frequency</th>
<th>Throughput (Typ)</th>
<th>Data Rate (Max)</th>
<th>Modulation Technique</th>
<th>Range (Radius Indoor) Depends, # and type of walls</th>
<th>Range (Radius Outdoor) Loss includes one wall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legacy</td>
<td>1997</td>
<td>2.4 GHz</td>
<td>0.9 Mbit/s</td>
<td>2 Mbit/s</td>
<td></td>
<td>~20 Meters</td>
<td>~100 Meters</td>
</tr>
<tr>
<td>802.11a</td>
<td>1999</td>
<td>5 GHz</td>
<td>23 Mbit/s</td>
<td>54 Mbit/s</td>
<td>OFDM</td>
<td>~35 Meters</td>
<td>~120 Meters</td>
</tr>
<tr>
<td>802.11b</td>
<td>1999</td>
<td>2.4 GHz</td>
<td>4.3 Mbit/s</td>
<td>11 Mbit/s</td>
<td>DSSS</td>
<td>~38 Meters</td>
<td>~140 Meters</td>
</tr>
<tr>
<td>802.11g</td>
<td>2003</td>
<td>2.4 GHz</td>
<td>19 Mbit/s</td>
<td>54 Mbit/s</td>
<td>OFDM</td>
<td>~38 Meters</td>
<td>~140 Meters</td>
</tr>
<tr>
<td>802.11n</td>
<td>June 2009[3] (est.)</td>
<td>2.4 GHz</td>
<td>74 Mbit/s</td>
<td>248 Mbit/s</td>
<td></td>
<td>~70 Meters</td>
<td>~250 Meters</td>
</tr>
<tr>
<td>802.11y</td>
<td>June 2008[3] (est.)</td>
<td>3.7 GHz</td>
<td>23 Mbit/s</td>
<td>54 Mbit/s</td>
<td></td>
<td>~50 Meters</td>
<td>~5000 Meters</td>
</tr>
</tbody>
</table>

Copyright – Dan Brandon
What are some problems and issues with wireless networks?
Don’t look ahead …
Wireless Network Issues

- Distance limitations
- Speed limitations
- May pick up interference
- Security issues
- Can be “jammed”
Cellular Wireless Communication

- A **cellular network** is a **radio** network made up of a number of **radio cells** (or just **cells**) each served by a fixed transmitter, known as a **cell site** or **base station**

- These cells are used to cover different areas in order to provide radio coverage over a wider area than the area of one cell - cellular networks are inherently asymmetric with a set of fixed main **transceivers** each serving a cell and a set of distributed (generally, but not always, mobile) transceivers which provide services to the network's users

- Cellular networks offer a number of advantages over alternative solutions:
  - increased capacity
  - reduced power usage
  - better coverage

- A good (and simple) example of a cellular system is an old **taxi** driver's radio system where the taxi company will have several transmitters based around a city each operated by an individual operator

Copyright – Dan Brandon
The increased capacity in a cellular network, compared with a network with a single transmitter, comes from the fact that the same radio frequency can be reused in a different area for a completely different transmission.

As the phone user moves from one cell area to another, the switch automatically commands the handset and a cell site with a stronger signal (reported by the handset) to go to a new radio channel (frequency). When the handset responds through the new cell site, the exchange switches the connection to the new cell site.
3G

- **3G** is the **third generation** of mobile phone standards and technology, superseding **2.5G** and based on the **International Telecommunication Union** (ITU) family of standards under the **IMT-2000**
- 3G networks enable network operators to offer users a wider range of more advanced services while achieving greater network capacity through improved **spectral efficiency**
- Services include wide-area wireless voice **telephony**, **video calls**, and broadband wireless data, all in a mobile environment; additional features also include **HSPA** data transmission capabilities able to deliver **speeds up to 14.4Mbit/s on the downlink** and 5.8Mbit/s on the uplink
- **Unlike IEEE 802.11** (common names **Wi-Fi** or **WLAN**) networks, 3G networks are **wide area cellular telephone networks** which evolved to incorporate high-speed internet access and **video telephony**
What are the major problems with cellular service?
Don’t look ahead ...
Real Problems with Cellular Service

- Areas not covered via cell towers:
  - 70% of the earth’s surface – oceans
  - Rural areas
  - Deserts
  - Wilderness
  - Polar areas

- Cell towers need power
  - In a disaster (i.e. Hurricane Katrina) there is a loss of both power and some towers

- Overcrowding
Iridium - Mobile Phone via Satellite
[needs clear view of sky]
WiMax & Beyond

- WirelessMAN or Worldwide Interoperability for Microwave Access, popularly known as **WiMax**, is the name for IEEE standard 802.16.
  - Wireless access range of up to 31 miles
  - Data transfer rate of **75 Mbps**
  - Secure system that offers both voice and video
4G

- Long-Term Evolution (a GSM standard) and Wi-Max (a IEEE Standard) are all “4G” wireless data transfer technologies
- Some carriers will deliver 4G via WiMax and some via LTE
- Wi-Max and LTE will change the last networking mile in the same way that Wi-Fi changed the last 100 feet of networking: by complementing or possibly replacing the existing technologies
WAN Connections

- Connecting the business to the internet
- T1 and T3 are provided by telecons
- T1 is 1.5Mbps (about $300 to $500 per month)
- A T3 is not 3 T1s, but is actually the bandwidth of 28 T1s bundled together for an approximate speed of 45Mbps (about $3000 to $9000 per month)
- Cable is provided by the cable company and typically falls between 3 and 10 Mbps, and it is the cheapest, about $50/month
- DSL provided by AT&T ranges from 128 Kbps to 3 Mbps (3000 Kbps) and priced similarly to cable (upload and download speeds typically vary)
Extranets  
(EDI over the Net)

- **Extranets** link business partners to one another over the Internet by providing access to certain areas of each other’s corporate intranets
- The main goal of **extranets** is to foster collaboration between business partners
- An **extranet** is open to selected B2B suppliers, customers and other business partners
- **Common protocol**: XML/JSON via **VPN**
Internet Business Connections
Corporate Overall DataCom Needs
[wired or wireless: speed, accessibility, cost, security issues]

T1, T3, DSL?
Detail Datacom Design
[routers, firewall, switches, etc.]
Exercise

- Draw your Corporate Overall DataCom Needs diagram:
A typical system design specification uses a structure similar to the following:

1. Management summary
2. System components
3. System environment
4. Implementation requirements
5. Time and cost estimates
6. Additional material
Systems Design Approval

- **User Approval**
  - Users must review and approve the interface design, report and menu designs, data entry screens, source documents, and other areas of the system that affect them.
  - Other IT department members (datacom, security, operations, etc.) also need to review the system design specification.
  - When the system design specification is complete, you distribute the document to a target group of users, IT department personnel, and company management.
Systems Design Approval (con’t)

- Presentations
  - The first presentation is to the systems analysts, programmers, and technical support staff members
  - Your next presentation is to department managers and users from departments affected by the system
  - The final presentation is for company management
  - Management might reach one of three decisions:
    - Proceed with systems development
    - Perform additional work on the systems design
    - Terminate the project
SYSTEM IMPLEMENTATION
Phased Approach to System Development

- Planning
- Analysis
- Design
- Implementation
- Systems Support
Phase Description

- Systems Implementation is the fourth of five phases in the systems development life cycle and includes:
  - Application Development
  - Documentation
  - Testing
  - Training
  - Data Conversion
  - System Changeover

- The deliverable for this phase is a completely functioning information system.
Session Objectives
Implementation

- Explain software quality assurance and software engineering
- Describe the application development process for structured, object-oriented, and agile methods
- Draw a structure chart showing top-down design, modular design, cohesion, and coupling
- Explain the coding process
- Explain unit, integration, and system testing
- Program, system, operations, and user documentation
- List the main steps in system installation and evaluation
- Develop training plans for various user groups, compare in-house and outside training, and describe effective training techniques
- Describe data conversion and changeover methods
- Explain post-implementation evaluation and the final report to management
Introduction

• The system design specification serves as a blueprint for constructing the new system
• The initial task is application development
• Before a changeover can occur:
  • The system must be tested and documented
  • Users must be trained
  • Existing data must be converted
• A formal evaluation of the results takes place as part of a final report to management
Application Development Methods

- **Traditional methods**
  - Start by reviewing documentation from prior SDLC phases and creating a set of program designs
  - At this point, coding and testing tasks begin

- **Agile Methods**
  - Intense communication and collaboration will now begin between the IT team and the users or customers
  - Objective is to create the system through an iterative process
Application Development

- Translate user interface into code
  - Mainframe: COBOL, RPG, Natural
  - Client/server: Java, C++, VB, …
  - Web based: HTML, JavaScript, CSS [client side]
- Translate processes (flowcharts, pseudo code) into code
  - Mainframe: COBOL, RPG, Natural
  - Client/server: Java, C++, VB, …
  - Web based: PHP, Python, Ruby, … [server side]
- Database design (ER Diagram, …) into relational database tables
Expediting Development

- Prototyping
- CASE, 4GL, GUI and other “tools”
- Buy versus Build
  - Need to build the part of the system from which competitive advantage is realized
- Object Oriented Technology
- Lifecycle Revisions
- Agile Methods
Prototyping

- A live implementation of the design (or portion thereof)
- Can be readily modified to explore different behaviors
- Serves as a blueprint for portions of the final system
- Speed and capacity not major issues
- Not fully tested
- Contains known limitations and anomalies
Prototyping - Important Uses

- Feasibility
- Evaluate alternatives
- Clarify user requirements
- Evaluate user “friendliness” and other human factors
Prototyping - What Can Go Wrong

- Prototype develops life of its own
- Prototype becomes final product
- Prototype over-accelerates development schedule
- Important documentation bypassed
- No quality legacy re-usable modules left behind
CASE, 4GL, GUI, other tools

- Can perform certain steps in lifecycle faster
  - design
  - construction (code generation)
  - testing
  - documentation
  - building prototypes

- Should not replace lifecycle; if so you simply build the wrong system even faster

Contractor with “blueprints” analogy
Overlap Methods
Evolutionary Methods

Version 1
- Req's Subset
- Design
- Code & Test

Version 2
- Req's Subset
- Design
- Code & Test

Version N
Agile Development

- Is a distinctly different systems development method
- Development team is in constant communication with the customer
- Focuses on small teams, intense communication, and rapid development iterations
- Extreme Programming (XP) is one of the newest agile methods
Agile Manifesto

- We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:
  - **Individuals and interactions** over processes and tools
  - **Working software** over comprehensive documentation
  - **Customer collaboration** over contract negotiation
  - **Responding to change** over following a plan

- That is, while there is value in the items on the right, we value the items on the left more
SRUM (con’t)

Sprint Backlog

Product Backlog
As prioritized by Product Owner

Daily Scrum Meeting

Backlog tasks expanded by team

24 hours

30 days

Potentially Shippable Product Increment

Source: Adapted from Agile Software Development with Scrum by Ken Schwaber and Mike Beedle.
Reported Benefits of Using Agile

- Productivity
- Time to Market
- Quality
- Cost
- Stakeholder Satisfaction

- VersionOne
- Tech Target
- Dr. Dobbs
Agile vs WaterFall

AGILE

- Failed 12%
- Challenged 45%
- Successful 43%

WATERFALL

- Failed 15%
- Challenged 59%
- Successful 26%

Copyright – D. Brandon, Ph.D., PMP
The Future of Agile Development

- Critics claim it lacks discipline and produces systems of questionable quality
- Before implementing agile development, the proposed system and development methods should be examined carefully
- Agile is not appropriate for all projects
- A one-size-fits-all solution does not exist
When most people think of quality in the context of IT, Engineering and other technical systems, they think of “bugs”

Where did the term “bug” arise?
Do not look ahead!
The term “bug” was originally coined by Dr. Grace Hopper who developed the COBOL language. She had found that a computer crash was due to a moth that lodged inside of the hardware.

For large IT systems of 1000 function points or more, total defects will average five bugs per function point.

Even with high quality software development, for every 500 or so lines of procedural code there is one bug.

Your razor has dozens of bugs, your TV may have hundreds of bugs, and your car may have thousands of bugs. Now most of these bugs are not encountered regularly, only when a certain set of circumstances arise.
Software Bugs

- Software bugs cost about $100 billion annually as estimated by the National Institute of Standards and Technology.
- Microsoft Windows 2000 was reported by inside sources to be released with 63,000 potential defects.
- There are more software bugs all of the time in our modern world as our dependency on IT deepens and automation and embedded software grows.
What is Quality?

How would you define quality?
Do not look ahead!
Some Quality Definitions

- “Even though quality cannot be defined, you know what it is.” (Pirsig, 1974)
- “Quality is the degree to which a specific product conforms to a design or specification.” (Gilmore, 1974)
- “Quality is fitness for use.” (Juran, 1974)
- “Quality is the totality of features and characteristics of a product or service that bears on its ability to satisfy stated or implied needs.” (Johnson and Winchell, 1989)
- “Quality is defined by the customer; customers want products and services that, throughout their lives, meet customers’ needs and expectations at a cost that represents value.” (Ford, 1991)
Modern Quality “Definition”

- Conformance to requirements, specifications, standards and fitness for use!
IT Quality

IT quality involves more than just “bugs” – what other matters are involved with IT product quality?
Systems Quality “Big Three”

- A system is **effective** if it supports business requirements and meets user needs
- A system is **reliable** if it handles input errors, processing errors, hardware failures, or human mistakes
- A system is **maintainable** if it is flexible, scalable, and easily modified
More Than Just Bugs - Extended IT Quality Definition

- Conforms to requirements and specifications
- Meets “customer expectations”
- Is defect free
- Is highly usability
- Is consistent with adopted standards
- Is reliability (does it do it right all the time)
- Is robust (can handle invalid/unusual data and usage)
- Is testable
- Is auditable
- Is maintainable and readable
- Is secure
- Is recoverable
- Is appropriately documented (External and Internal)
- Is efficient (with respect to speed, storage, clicks, keystrokes, and other resources)
- Is platform independent (portability)
- Is flexible and adaptable
The term “V & V” (verification and validation) has emerged in recent years and it has been applied to software development projects.

IEEE 1012-1998 is the new standard for V & V.

Formally verification is proof of compliance with requirements, specifications, and standards.

Verification is primarily concerned with built-in quality and is a process related notion that asks the question “are we building the product in the correct manner.”

Answering that question requires both the internal testing of the product and the inspection of processes.

For IT projects inspection involves primarily design walkthroughs, code walkthroughs, and method and tool evaluations.
Verification And Validation (con’t)

- Validation is proof that the customer and end users are satisfied with the system.
- Validation is a **product related notion** that asks the question “**have we built the correct product**”
- Answering that question requires acceptance (external) testing of the product, which for IT systems involves not only reviewing the final product, but also **acceptance of preliminary artifacts**, including:
  - Requirement documents are drawings
  - Design documents and drawings
  - Prototypes
Software testing and defect correction typically takes between 20% to 60% of project development time and budget.

Complex application developments can spend more than half of their total program effort on testing.

When a time crunch arises in a project, often the full testing of a product is compromised.

One rule of thumb is that testing costs will be at least 25% of the total development costs.
When is testing not necessary?
Do not look ahead!
However there may be times that testing is not necessary; the Software Program Managers Network has identified these circumstances:

- When the responsibility for failure can be shifted to someone else
- When the impact or significance of any problem is insignificant
- When the software does not have to work
- When no one is or will be using the system
- When the project has already failed
Unit Testing

- Unit testing is typically done by the developers and is a process to make sure that each software module performs properly relatively independently from other modules.
- Input, output, and interface operations are typically simulated.
- **Unit testing is typically mostly “White-Box” (“structural”) testing** where knowledge of module internals is utilized to insure that each source code statement and “path” through the code is exercised, possibly as a function of different external and database conditions.
- Often “test harnesses” are created for each module which is a set of specific tests for that module.
- These harnesses are checked into the source code version control system, and they are usually automatically repeated for each change to that module.
- The unit testing of a module is typically included in the WBS (Work Breakdown Structure) task to build that module.
Integration and System Testing

- Integration testing involves sub-system testing
- System testing involves testing the entire system working together; this is often called an “end-to-end” test
- This type of testing should not be done by the developers, but by an independent testing group
- The project WBS should have separate WBS code(s) for integration testing
- This testing process is typically composed of three phases: test planning, test set construction, and test execution
Testing the System

- Unit Testing
- Integration Testing
- System Testing
- Destructive and Load Testing
- Intrusion Testing (security)
- Acceptance Testing
IEEE/ANSI STD 1008, 2002

- Test planning
  - Plan the general approach
  - Plan resources and schedule
  - Determine test environment(s)
  - Determine features to be tested
  - Refine the general plan
- Test set construction
  - Design the set of tests and associated procedures
  - Implement that design (build test scripts)
- Testing
  - Execute the test sets
  - Check for correct behavior and results
  - Evaluate the test results, effort, and other relevant metrics
  - Document results
Test Plan Document

- Identification of IT product and version
- Background of product/version including purpose
- Prior versions including revision history
- Purpose and objectives of this testing
- Reference to other product documents (i.e. requirements, design, etc.)
- Relevant standards
- Project team and stakeholder identification and contact info
- Test organization contact info
- Test resources (people, training, access to other people, hardware, space, etc.)
- Test environment(s), platforms, locations, and conditions
- Assumptions and dependencies
- Test scope and focus
- Relation to other validation activities
- Testing outline
- Testing data and other setup
- Test metrics
- Testing tools
- Test scripts
- Detail test procedures
- Test reporting, documentation, deliverables
A key part of this test process is the construction of the test sets, sometimes called “test scripts”.

The test scripts should be based upon the product requirements either directly or via the use case scenarios and storyboards (screen design or “paper prototypes”).

*If the test scripts are taken from the use case scenarios, then the likelihood is much greater that the testing will involve features and code paths that are more important to the customer*

*Testing is expensive, so one does not want to waste time testing features that no one will use or cares little about*

*However, if the test scripts are not taken directly from the requirements, there still needs to be a check that each requirement is included somewhere in the test scripts*

These test scripts indicate specifically how the product is to be tested and what the expected results are at each point in the test. For example in testing a form to add a new entity to a database, the script would indicate exactly what was to be typed into each field on the form including typing of invalid entries (for which the product should give meaningful error messages).

In conjunction with the generation of test scripts, there is a need to generate test data. The ideal situation is to have the test data created as part of running earlier test scripts.
Test Scripts (con’t)
Acceptance Testing

- Acceptance testing is often carried out after system testing and this type of testing involves the benefiting organization (customer).

- Acceptance testing may be called for in the contract for the project, and sometimes this testing is performed exclusively by the benefiting organization (customer or user).

- Normally acceptance testing is on a much smaller scale than integration testing and represents a “sampling” of features and conditions.

- After (or instead of) acceptance testing, the product might be placed into production for a limited set of users.
  - “Alpha Testing” involves a small set of “friendly” users. Then after alpha testing is successful, “Beta Testing” is used which involves a larger and not necessarily friendly group.
Automated Testing

- Much testing of IT systems is still done manually, however, automated testing can often produce better results at a cheaper price.
- Newer automated tools cover all phases of the testing process including planning.
- The Quality Assurance Institute conducted benchmarks comparing manual and automated testing methods and found an overall reduction in person time/cost of 75%.
- As well as saving time and cost (after an initial learning curve in time and dollars), automatic testing has these further advantages:
  - Test repeatability and consistency
  - Expanded and practical test reuse
  - Practical baseline suites
Automated Testing (con’t)

- Automatic testing tools really come in handy when one has to repeat the same tests for a number of different client and/or server platform environments as is the case with modern web and e-commerce applications.

- Automated testing is also very useful when load testing is necessary with a high volume of users, transactions, and/or data.

- However, there are situations where automatic testing is less useful and these involve situations where human judgment is necessary and it is difficult to define criteria in a quantitative manner.

- Usability testing is one such area (usability of documentation as well as the product), and another area is “localization” particularly of global e-commerce applications.
Root Causes Of Quality Problems

- Lack of an understanding of what quality means for software
- Inadequate defect prevention
- Inadequate use of reviews and inspections
- Insufficient or careless testing
- Lack of quality measurements
- Lack of understanding by project management that quality is critical to completion metrics and user satisfaction
- Excessive schedule pressure leading to reduced quality efforts
- Unstable and ambiguous user requirements
Fishbone Diagram for Root Causes
Defect Control

Cumulative Defects

X = Found
O = Corrected

Time →
Documentation

- Program (internal) Documentation
- System Documentation
- Operations Documentation
- User (external) Documentation
  - Users Manual
    - Online
    - Printed
  - Help Subsystem
Management Approval

- After system testing is complete, one presents the results to management.
- If system testing produced no technical, economical, or operational problems, management determines a schedule for system installation and evaluation.
Remaining Steps In Implementation

[are these in your WBS?]

- Prepare a separate operational and test environment
- Provide training for users, managers, and IT staff
- Perform data conversion and system changeover
- Carry out post-implementation evaluation of the system
- Present a final report to management
Operational and Test Environments

IT Staff Test Environment

Test Data
Procedures
Programs

Authorized Changes

Users Operational Environment

Test Data
Procedures
Programs
Training

- **Training Plan**
  - The three main groups for training are *users*, *managers*, and IT *staff*
  - You must determine how the company will provide training

- **Vendor Training**
  - If the system includes the purchase of software or hardware, then vendor-supplied training is one of the features you should investigate in the RFPs and RFQs (requests for quotation) that you send to potential vendors
Training Tips

- Train people in groups, with separate training programs for distinct groups
- Select the most effective place to conduct the training
- Provide for learning by hearing, seeing, and doing
- Prepare effective training materials, including interactive tutorials
- Many training consultants, institutes, and firms are available that provide either standardized or customized training packages
- Consider online training (webinars) and/or video tutorials
Data Conversion Strategies

- The old system might be capable of exporting data in an acceptable format for the new system or in a standard format such as ASCII or ODBC.
- If a standard format is not available, you must develop a program to extract the data and convert it.
- Often requires additional data items, which might require manual entry.
- You must ensure that all system control measures are in place and operational to protect data from unauthorized access and to help prevent erroneous input.
- It is essential that the new system be loaded with accurate, error-free data.
System Changeover Options

- **DIRECT CUTOVER**
- **PARALLEL OPERATION**
- **PILOT OPERATION**
- **PHASED OPERATION**
Direct Cutover

- Involves more risk than other changeover methods
- Companies often choose the direct cutover method for implementing commercial software packages
- Cyclical information systems usually are converted using the direct cutover method at the beginning of a quarter, calendar year, or fiscal year
Parallel Operation

- Easier to verify that the new system is working properly under parallel operation than under direct cutover.
- Running both systems might place a burden on the operating environment and personnel and cause processing delay.
- Is not practical if the old and new systems are incompatible technically.
- Also is inappropriate when the two systems perform different functions.
Pilot Operation

- The group that uses the new system first is called the pilot site.
- The old system continues to operate for the entire organization.
- After the system proves successful at the pilot site, it is implemented in the rest of the organization, usually using the direct cutover method.
- Is a combination of parallel operation and direct cutover methods.
Phased Operation

- You give a part of the system to all users.
- The risk of errors or failures is limited to the implemented module only.
- Is less expensive than full parallel operation.
- Is not possible, however, if the system cannot be separated easily into logical modules or segments.
System Changeover Methods
Post-Implementation Evaluation

- A post-implementation evaluation should examine all aspects of the development effort and the end product
  - Complete requirements traceability
  - Customer sign off
  - Close out contracts
  - Lessons Learned
Final Report to Management

- Final versions of all system documentation
- Planned modifications and enhancements to the system that have been identified
- Recap of all systems development costs and schedules
- Comparison of actual costs and schedules to the original estimates
- Post-implementation evaluation, if it has been performed
Modern Web Based Development

[Diagram of the development process including Use Case Scenarios, Preliminary User Manual, Help Screens, Test Plans, Integration Testing, Online User Manual, Deployment, Requirements, Screen Design (Storyboards), Content Development, Overall Site Design, Database Design, Program Development, Database Development (SQL Create Table Scripts), Database Testing (SQL Insert, Select, Update Scripts), and related processes.]
Software Engineering Institute (SEI)

- Capability Maturity Model (CMM)
- Capability Maturity Model Integration (CMMI)
- Best practices
- Process improvement
- CMMI tracks an organization's processes, using five maturity layers
References

- **Systems Architecture** by Stephen D. Burd
- **The Art of Systems Architecting, Third Edition (Systems Engineering)** by Mark W. Maier
- **IT Architecture For Dummies** by Kirk Hausman and Susan L. Cook
- **The Art of Enterprise Information Architecture: A Systems-Based Approach for Unlocking Business Insight** by Mario Godinez

Copyright Dan Brandon, PhP, PMP
Homework

- Textbook chapters ten and eleven
- Quiz on those chapters
- Textbook questions: Chapter 10: 1, 2, 7; Chapter 11: 1, 3, 6, 9
- ABC Deliverables:
  - System architecture (see next slide)
  - Application development methodology (SDLC, CASE, Agile, etc.)
  - Data communications needs and methods
  - Quality/Acceptance/Training/Support requirements
  - System changeover method
- Project:
  - Website overall design diagram and planning questions
  - Web page prototypes (major pages only)
  - Database design (ER diagram)
  - Optional: Microsoft Access Prototype
  - CRUD diagram
Explain how your architecture addresses these needs:

- Enterprise resource planning (ERP)
  - Company-wide strategy for using IT resources
- Total cost of ownership
- Scalability
- Legacy system and other interface requirements
- Processing options
- Data Communication options
- Security issues
- Web and mobility integration