

# Chapter 5

## The Database Life Cycle

# Database Design Life Cycle

- *Database Life Cycle*: all the events from the time the database is first proposed to the time it is retired from service
- Work is typically divided into project with each having a finite set of *deliverables*

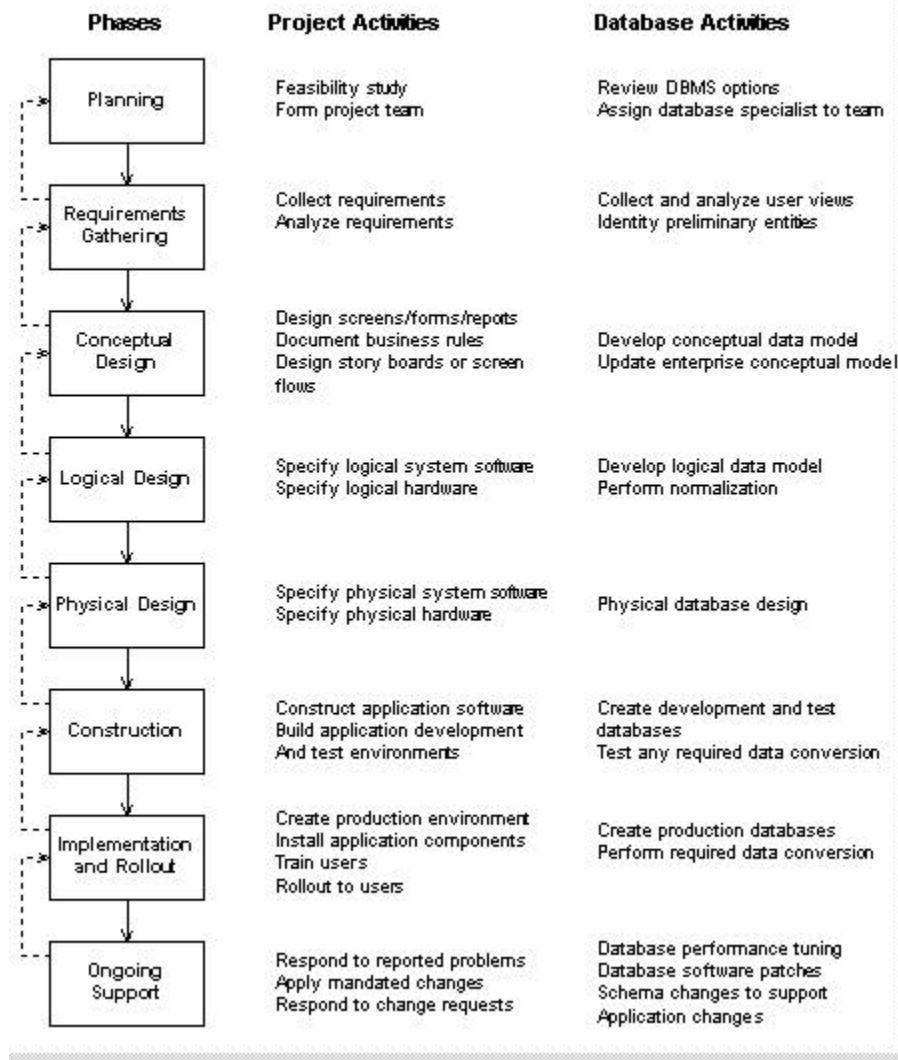
# Development Methods

- Traditional
- Prototyping
- Rapid Application Development (RAD)
- Agile Methods (more detail later)
- And of course, there are hybrids

# Traditional Method

- Follows process called *Systems Development Life Cycle (SDLC)*
- There are many variations (as many as there are authors and software vendors)
- Most methodologies contain the same basic components:
  - **Tasks** – the smallest unit of work, which typically has a completion time of a few hours to a few days, and which is assigned to one person or a group consisting of a few people.
  - **Work Breakdown Structure (WBS)**: A hierarchy used to organize tasks into manageable groups

**Figure 5-1: Traditional System Development Life Cycle (SDLC)**



# Planning Phase

- Understand where enterprise is and where it wants to be
- Often done over longer time span than any one individual project
- May include feasibility studies
- May or may not have a database specialist involved (depending on the organization)

# Requirements Gathering

- Gather and document high-level, yet precise, description of what project is to accomplish
- Focus on *what*, not *how*
- Consider all existing business processes, business rules, and entities
- Data requirements focus on *user views*

# Requirements Gathering Methods

- Conduct Interviews
- Conduct a Survey
- Observation
- Document Review



# Conduct Interviews

- Advantages
  - Receive answers to unasked questions
  - Includes non-verbal responses (body language)
- Disadvantages
  - Takes more time than other methods
  - Interviewer may “telegraph” desired answers

# Conduct a Survey

- Advantages
  - Covers a lot of ground in a short time
  - Questions presented uniformly to all
- Disadvantages
  - Typically has very poor response rates
  - Unbiased questions very difficult to compose
  - Does not include non-verbal clues

# Observation

- **Advantages**

- See normal processes in everyday use
- May observe events that would not otherwise be mentioned

- **Disadvantages**

- *Hawthorne Effect*: observation changes behavior
- May never see the exceptions
- Travel costs can be high

# Document Review

- Advantages
  - Typically the least time consuming method
  - Documents often provide better overviews
  - Pictures and diagrams worth 1,000 words
- Disadvantages
  - Documents may not reflect actual practices
  - Documentation is often out of date



How the customer explained it



How the project leader understood it



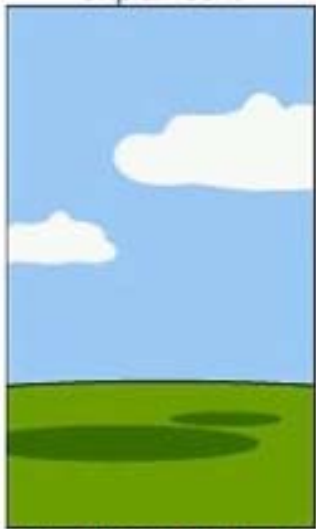
How the engineer designed it



How the programmer wrote it



How the sales executive described it



How the project was documented



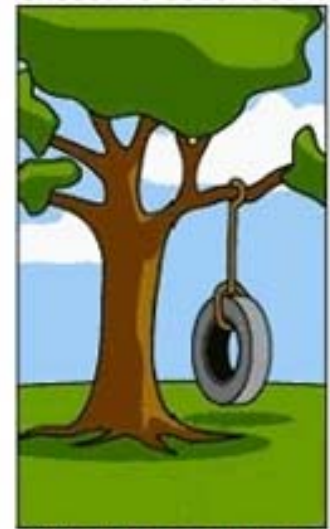
What operations installed



How the customer was billed



How the helpdesk supported it



What the customer really needed

# Conceptual Design

- Design the externals of the application(s) and database(s)
- Sometimes called *External Design*
- Finalizes the layout of reports, screens, forms, web pages and other data entry and presentation vehicles
- Flow of application captured using flow charts, storyboards, and/or screen flows
- DBA or Data Modeler updates Enterprise Data Model (usually in ERD form)

# Logical Design

- Technical design of the application(s) and database(s)
- Sometimes called *Internal Design*
- Applications specified as *modules*
- Database design using *normalization*

# Physical Design

- Logical design is mapped (converted) to the actual platform(s) (hardware and systems software) on which it will run
- Includes capacity estimates for processors, disk devices, and network bandwidth
- Database DDL is created or generated for each DBMS to be used

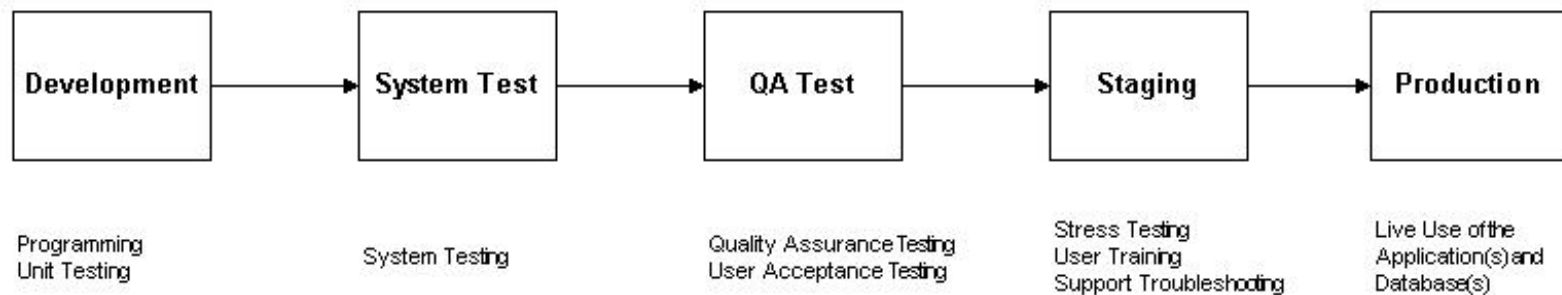


# Construction

- Developers code and test the application modules
- Tested modules are promoted through various environments, eventually being installed into production
- Database components move in concert with application modules (sometimes in advance)

# Hardware/Software Environments

## F05-02: Development Hardware/Software Environments



# Implementation and Rollout

- *Implementation*: the process of installing the new application system's components into the live production system
- *Rollout*: the process of placing groups of business users on the new application system

# Ongoing Support

- Turnover to production support team
- Respond to any issues that occur:
  - Performance problems
  - Abnormal or unexpected results
  - Complete failures
  - Requests for enhancements
  - Isolation of bugs vs. enhancements

# Ongoing Support

- Routine tasks:
  - Application of patches
  - Performance monitoring and tuning
  - Space monitoring and management
  - Bug fixes, including corresponding database changes

# Database Environments

- Development environment
- Test environment
- QA environment
- Staging environment
- Production environment

# Six Phases of a Project

- Enthusiasm
- Disillusionment
- Panic
- Search for the Guilty
- Punishment of the Innocent
- Praise and Honors for the Non-Participants



# Nontraditional Methods: Prototyping

- Rapid development of the application using iterative sets of design, development, and implementation steps as a method for determining requirements
- Some methods carry prototype all the way to production with each version adding more detailed functions (perpetual prototyping)
- Joint Application Design (JAD) is a popular requirements gathering method



# Nontraditional Methods: Rapid Application Development (RAD)

- Allows functioning application systems to be built in as little as 60-90 days
- Compromises often made using 80/20 rule
- System ultimately built out to meet 100% of requirements
- Not useful in controlling project schedules or budgets
- Requires a highly skilled project manager
- Most applicable when rapid deployment more important than product quality

# Agile Methods

- Based on Iterative and Incremental methods
- Requirements and solutions evolve
- Teams are cross-functional
- Development cycles are time-boxed
- Examples include Agile Unified Process (AUP), Rational Unified Process (RUP), Scrum and Dynamic Systems Development Method (DSDM)
- Collectively called Agile since publication of the Agile Manifesto in 2001