### Software Development Lifecycle (SDLC)

#### Development Methodologies

**Build and Fix**
- No key architecture design
- Problems fixed as they occur
- No formal feedback loop
- Reactive, not proactive

**Waterfall**
- Linear, sequential lifecycle
- Each phase completed before moving on
- No formal way to make changes during cycle
- Project ends before collecting feedback and re-starting

**V-shaped**
- Rapid prototyping - quick sample to test the current project
- Evolutionary prototyping - incremental improvements to design
- Operational prototypes - incremental improvements intended for production

**Incremental**
- Iterative
- Risk analysis during development
- Future information and considerations for risk analysis
- Allows for testing early in development

**Spiral**
- Rapid prototyping
- Design, analysis, and development
- Analysis and design are quickly demonstrated
- Testing and requirements are often revised

**Agile**
- Umbrella term - multiple methods
- Highlights efficiency and iterative development
- User stories describe what a user does and why
- Prototypes filtered down to individual features

**DevOps (Development & Operations)**

**Software Development**
- Quality Assurance
- IT Operations

### Software Development Methods

#### Database Systems

- **Database**
  Define storing and manipulating data

- **DBMS (database management system)**
  - Software program control access to data stored in a database.

- **DBMS Types**
  - Hierarchical - Network - Mesh
  - Object-oriented - Relational

- **DDL (Data Definition Language)**
  - Data definition language defines structure and schema SQL

- **DoD (Degree of Db)**
  - Number of attributes (columns) in table

- **Tuple (row)**

- **DDE (Dynamic Data Exchange)**

- **DCL (Data Control Language)**
  - Subset of SQL.

#### Semantic Integrity

- To ensure semantic rules are enforced between data types

#### Referential Integrity

- All foreign keys reference existing primary keys

#### Candidate Key

- Primary key
  - Unique data identification

#### Foreign Key

- Reference to another table which includes primary key
- Foreign and primary keys link data in a table

#### DBMS terms

- Incorrect Summaries - Dirty Reads
- Lost Updates
- Dynamic Lifetime Objects: Objects developed using software in an Object Oriented Programming environment.
- ODBC: Open Database Connectivity: Database feature where applications to communicate with databases with different types of databases without a program code.
- Database contamination - Mixing data with different classification levels
- Database partitioning - splitting a single database into multiple parts with unique contents
- Polymorphism: two or more rows in the same relational database table appear to have identical primary key and different data in the table.

### Programming Language Types

#### Machine Languages

- Assembly Language
  - Direct instructions to processor - binary representation

- High-Level Languages
  - Processor independent programming languages - use IF, THEN and ELSE statements as part of the code logic.

- Very high-level languages
  - Generation 4 languages further reduce amount of code required - programmers can focus on algorithms.

- Natural languages
  - Natural languages generate 5 languages enable system to learn and change on its own.

#### Database Architecture and Models

- **Relational Model**
  - Uses attributes (columns) and tuples (rows) to organize data

- **Hierarchical Model**
  - Parent child structure: An object can have one child, multiple children or no children.

- **Network Model**
  - Similar to hierarchical model but objects can have multiple parents.

- **Object Oriented Model**
  - Has the capability to handle a variety of data types and is more dynamic than a relational database.

- **Object Relational Model**
  - Combination of object oriented and relational models.

#### Programming Channels (Storage & Timing)

- **Executable Content**
  - Mobile code

- **Mobile Code**
  - Actively controls, Java applets, browser scripts

- **Web Pages**
  - Propagates with help from the host

- **Worms**
  - Propagates without any help from the host

- **Logic Bomb/Code Bomb**
  - Runs when a specific event happens

- **Buffer Overflow**
  - Memory buffer exhaustion

- **Backdoor**
  - Malicious code installed at back end with the help of a front end user

- **Covet Channel**
  - Unauthorized information gathering

- **Botnet**
  - Zombie code used to commence thousands of systems

- **Trojan**
  - Malicious code that outwardly looks or behaves as harmless or necessary code

### Data Warehousing and Data Mining

- **Data Warehousing**
  - Combines data from multiple sources.

- **Data Mining**
  - Arrange the data into a format easier to make business decisions based on the content.

### Data Threats

- **Aggregation**
  - The act of combining information from various sources

- **Inference**
  - Process of information pacing

- **Access Control**
  - Content Dependent Access Control: access is based on the sensitivity of the data

- **Access Control Mechanisms**
  - Database Views: set of data a user or group can see

- **Polys instantiation**
  - Prevents data interference violations in databases

### Traditional SDLC

- **Steps**
  - Analysis, High level design, Detail Design, Construction, testing, Implementation

- **Phases**
  - Initiation: Feasibility, cost analysis, risk analysis, Management approval, basic security controls

- **Software development: Coding. Unit testing Prototyping, Verification, Validation**

- **Acceptance testing and implementation: security testing, data validation**

### Software Assessment & Testing Terms

#### Knowledge Management

- **Expert Systems**
  - Two main components: Knowledge base and the Inference engine
    - Use human reasoning
    - Rule based knowledge base
    - If then statements

- **Neural Networks**
  - Accumulates knowledge by observing events, taking inferred rules to a class model

- **Neural Networks**
  - Forward chaining: Begins with known facts and applies inference rule to extract more data unit it reaches to the goal. A bottom up approach. Breadth-first search strategy

- **Backward chaining**
  - Begins with the goal, works backward from inference rules to deduce the required facts that support the goal. A top down approach. Depth-first search strategy

#### Security Assessment & Testing Terms

- **Cross-site request forgery**
  - Browser side trust is exploited by trying to submit authenticated requests forcefully to third party sites

- **Session Hijacking**
  - Uses inputs to pretend a user's browser to execute untrusted code from a trusted site

- **SQL Injection**
  - Directly attacks a database through a web app

- **Hijack / Update / Security fix**
  - Updates to operating systems and applications

- **Service Pack**
  - Collection of patches for a complete operating system

### Change Management Process

- **Process**
  - Development organizational framework where users can request modifications, conduct cost/benefit analysis by management, and task prioritization by developers

- **Release Control**
  - Change approval before release

### Configuration Management Process

- **Software Version Control (SCV)**
  - A methodology for storing and tracking changes to software

- **Configuration Control**
  - The labeling of software and hardware configurations with unique identifiers

- **Configuration Audit**
  - Ensure that the production environment is consistent with the accounting records

### capability Maturity Model

- **Reactive**
  - 1. Initiating – informal processes.
  - 2. Repeatable – project management processes

- **Proactive**
  - 3. Defined – engineering processes, project planning, quality assurance, configuration management practices

- 4. Managed – product and process improvement

- 5. Optimizing – continuous process improvement

### Project Management Tools

- **Gantt chart**
  - Type of bar chart that illustrates the relationship of tasks and schedules over time

- **Program Evaluation Review Technique (PERT)**
  - Project scheduling tool used to measure the capacity of a software product in development which uses to calculate risk.

### Phases of object-oriented design

- **ORQA (Requirements Analysis)**
  - Define classes of objects and interactions

- **OOA (Analysis)**
  - Identify classes and objects which are common to any applications in a domain - process of discovery

- **OOD (Design)**
  - Objects are instances of classes

- **OOP (Programming)**
  - Introduce objects and methods

- **ORBQ (Object Request Brokers)**
  - Work as middleware locators and distributors for the objects

- **CORBA (Common object request)**
  - Architecture and standards that use ORBS to allow different systems and software on a system to interface with each other

### Virus Types

- **Boot sector**
  - Infects executable system files, BIOS and system commands

- **System infector**
  - Infects a system's factory installed UEFI (firmware)

- **UFEI**
  - Infects systems a stored location other than in the main system folder. Example NOTEPAD.EXE

- **MultiPart**
  - Infects both boot sector and executable files

- **Self-garbling**
  - Attempts to hide from anti-virus by changing the encoding of its own code, a.k.a “garbling” monitored

- **Polymorphic**
  - The virus modifies the "garble" pattern as it spreads

- **Resident**
  - Loads as and when a program loads to the memory

### Anti-Virus Types

- **Signature based**
  - Not able to detect new malware a.k.a. Zero day attacks

- **Heuristic based**
  - Static analysis without relying on signatures

### Protection Rugs

- **Layer 0**
  - Operating system kernel

- **Layer 1**
  - Parts of the operating system other than the kernel

- **Layer 2**
  - I/O drivers and utilities

- **Layer 3**
  - Applications and programs