**Chapter 20: Software Development Security**

Security should be a consideration at every stage of a system’s development, including the software development process.

Developers often rely on shared software libraries that contain reusable code. Vulnerability in a widely used library may impact a huge number of businesses across the globe.

Cohesion describes the strength of the relationship between the purposes of the methods within the same class. High Cohesion indicates good software design principles.

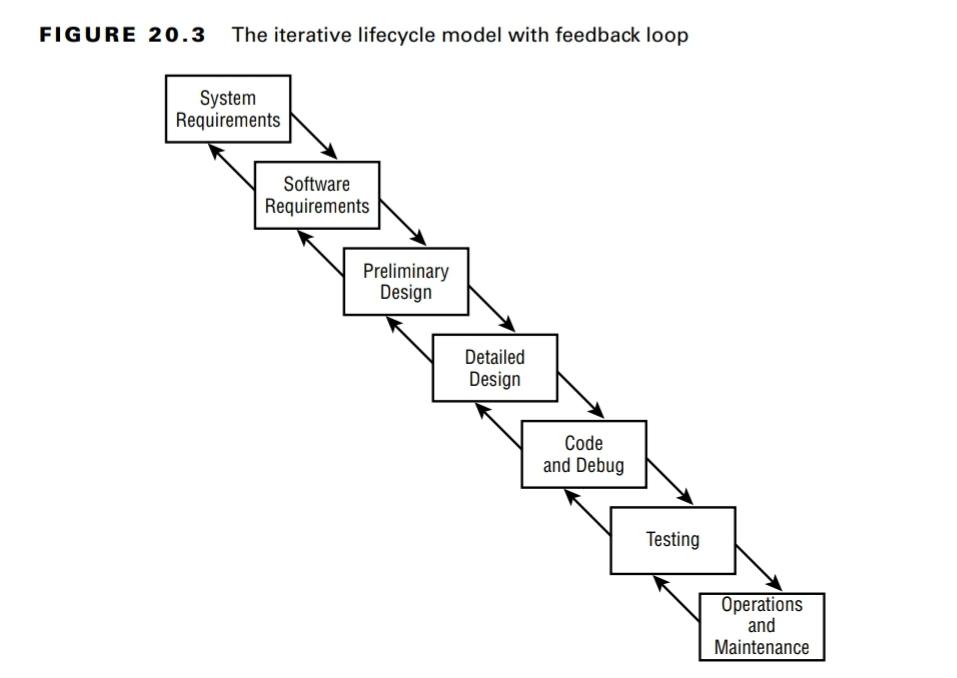
Coupling is the level of interaction between objects or classes. Lower coupling provides better software design because objects are more independent.

Core activities that are essential to the development of secure systems:

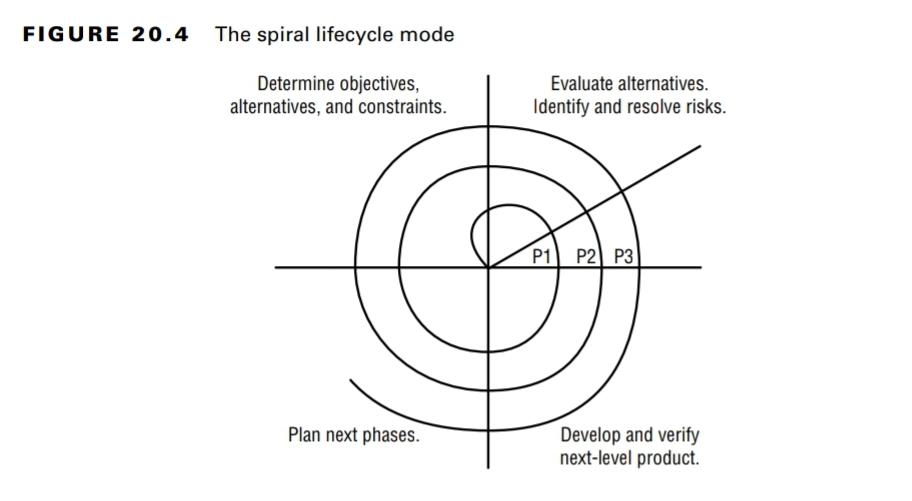
* Conceptual definition
* Functional requirements determination
* Control specifications
* Design review
* Coding
* Code review walk-through
* System testing
* Maintenance and change management

**Software Development Lifecycle Models**

1. **Waterfall Model -** The traditional waterfall model has seven stages of development. As each stage is completed, the project moves into the next phase but only in forward direction. In practical application, the waterfall model, of necessity, evolved to a more modern model. The iterative waterfall model does allow development to return to the previous phase to correct defects discovered during the subsequent phase. This is often known as the feedback loop characteristic of the waterfall model.



1. **Spiral Model -** The spiral model encapsulates a number of iterations of another model (the waterfall model), and is therefore also known as a metamodel, or a “model of models.” Each “loop” of the spiral results in the development of a new system prototype (represented by P1, P2, and P3).



1. **Agile Software Development -** Agile is not a methodology, it is a philosophy based on the 12 Agile principles defined in Agile Manifesto.

In an Agile approach, the team embraces the principles of the Agile Manifesto and meets regularly to review and plan their work. Agile majorly focuses on customer satisfaction through early and continuous delivery of valuable software. It also aims to accommodate changes even in the later stages of development.

Several specific methodologies have emerged that take Agile principles and define specific processes that implement them. These include Scrum, Kanban, Rapid Application Development (RAD), etc.

1. **Capability Maturity Model (CMM or SW-CMM) -** The idea behind the SW-CMM is that the quality of software depends on the quality of its development process. Organizations engaged in software development move through a variety of maturity phases in sequential fashion. The stages of the SW-CMM are as follows:

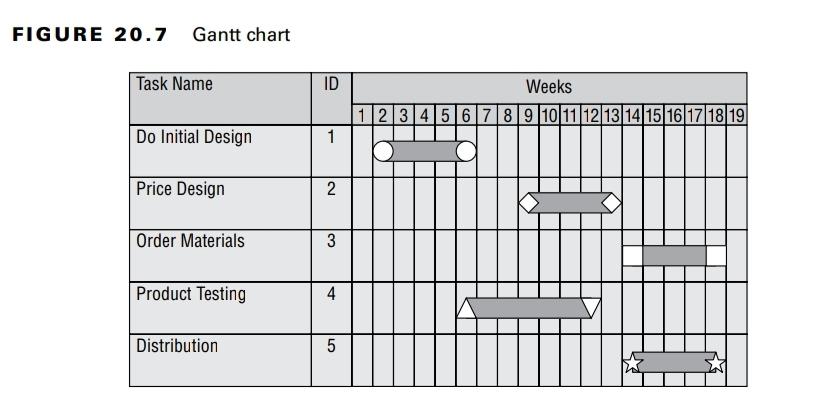
* **Initial** - There is usually little or no defined software development process.
* **Repeatable** - In this phase, basic lifecycle management processes are introduced.
* **Defined** - In this phase, software developers operate according to a set of formal, documented software development processes.
* **Managed** - In this phase, management of the software process proceeds to the next level. Quantitative measures are used to gain a detailed understanding of the development process.
* **Optimizing** - In the optimized organization, a process of continuous improvement occurs.

1. **IDEAL model -** Itimplements many of the SW-CMM attributes. The IDEAL model has five phases quite similar to the CMM model. The five phases are Initiating, Diagnosing, Establishing, Acting, Learning.

**Gantt Charts and PERT**

A Gantt chart is a type of bar chart that shows the interrelationships over time between projects and schedules. It provides a graphical illustration of a schedule that helps you plan, coordinate, and track specific tasks in a project.

Program Evaluation Review Technique (**PERT**) is a project-scheduling tool used to judge the size of a software product in development and calculate the standard deviation (SD) for risk assessment.



The change management process has three basic components:

**Request Control** - Provides an organized framework within which users can request modifications.

**Change Control** - Developers re-create the situation and analyze the appropriate changes to remedy the situation.

**Release Control** - Once the changes are finalized, they must be approved for release through the release control procedure.

The software configuration management (SCM) has four main components:

**Configuration Identification** - Document the configuration of software products.

**Configuration Control** - Ensure the changes to software versions are made in accordance with the change control and configuration management policies.

**Configuration Status Accounting** - Formalized procedures are used to keep track of all authorized changes that take place.

**Configuration Audit** - Periodic configuration audit to ensure that the actual production environment is consistent with the accounting records and no unauthorized configuration changes have taken place.

The word DevOps is a combination of Development and Operations, symbolizing that these functions must merge and cooperate to meet business requirements. The DevOps model is closely aligned with the Agile development approach and aims to dramatically decrease the time required to develop, test, and deploy software changes.

If code is being rapidly developed and moved into production, security must also move with that same agility. For this reason, many people prefer to use the term DevSecOps to refer to the integration of development, security, and operations.

APIs allow application developers to bypass traditional web pages and interact directly with the underlying service through function calls.

A hierarchical data model combines records and fields that are related in a logical tree structure. This results in a one-to-many data model, where each node may have zero, one, or many children but only one parent.

The distributed data model has data stored in more than one database, but those databases are logically connected. The user perceives the database as a single entity, even though it consists of numerous parts interconnected over a network. Distributed databases have many-to-many mapping.

A relational database consists of flat two-dimensional tables made up of rows and columns. The row and column structure provides for one-to-one data mapping relationships. The number of rows in the relation (table) is referred to as cardinality, and the number of columns is the degree.

A candidate key is a subset of attributes that can be used to uniquely identify any record in a table. Each table may have one or more candidate keys.

A primary key is selected from the set of candidate keys for a table to be used to uniquely identify the records in a table. Each table has only one primary key.

Any candidate key that is not selected as the primary key is referred to as an alternate key.

A foreign key is used to enforce relationships between two tables, also known as referential integrity.

Relational databases support the use of transactions to ensure data integrity. Each transaction is a discrete set of SQL instructions that should either succeed or fail as a group.

Relational database transactions have four required characteristics known as the ACID model.

1. **Atomicity** - Database transactions must be atomic. If any part of the transaction fails, the entire transaction must be rolled back as if it never occurred.
2. **Consistency** - Database must be consistent with all the database's rules (for example, all records have a unique primary key) before and after the transaction.
3. **Isolation** - The isolation principle requires that transactions operate separately from each other. If a database receives two SQL transactions that modify the same data, one transaction must be completed in its entirety before the other transaction is allowed to modify the same data.
4. **Durability** - Database transactions must be durable. That is, once they are committed to the database, they must be preserved.

Databases that fail to implement concurrency correctly may suffer from the following issues:

* **Lost Updates** - Occur when two different processes make updates to a database, unaware of each other’s activity
* **Dirty Reads** - Occur when a process reads a record from a transaction that did not successfully commit.

SQL provides a number of functions that combine records from one or more tables to produce potentially useful information. This process is called aggregation.

Aggregation attacks involve the use of specialized database functions to combine information from a large number of database records to reveal information that may be more sensitive than the information in individual records would reveal.

Inference attacks use deductive reasoning to reach conclusions from existing data.

Polyinstantiation allows the insertion of multiple records that appear to have the same primary key values into a database at different classification levels.

Contamination is the mixing of data from a higher classification level with data from a lower classification level in the database.

Expert systems use a knowledge base consisting of a series of “if/then” statements to form decisions based on the previous experience of human experts.

COTS - commercial-off-the-shelf software

ERP - enterprise resource planning