DCIT26 (1/1)

Software design is the process by which an agent creates a specification of a software artifact, intended to accomplish goals, using a set of primitive components and subject to constraints.
- involves problem solving and planning a software solution.

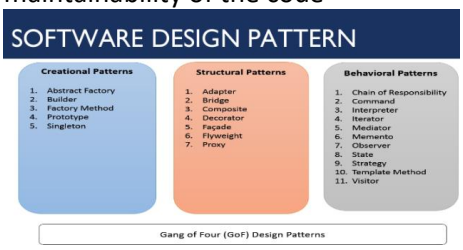
IMPORTANCE OF SOFTWARE DESIGN

1. Dynamic
2. Flexible
3. Reusability
4. Easy to Understand
5. Cost-efficiency is increased

PATTERN - a solution to a common software problem in a context capture design expertise and allow that expertise to be communicated patterns are designs; they are structure of codes must be instantiated/applied

IMPORTANCE OF PATTERN

1. Help reuse existing high quality solutions to commonly occurring problems
2. Improve individual and team learning
3. Shift level of thinking to a high perspective
4. Illustrate basic object-oriented principles
5. Improves modifiability and maintainability of the code



CREATIONAL PATTERN

◆ **Abstract Factory**: Allow creation of families of related objects independent of implementation
◆ **Builder**: Can be used to ease the construction of a complex object from simple objects.
◆ **Factory Method**: Used to create objects without specifying concrete classes.
◆ **Prototype**: allows cloning objects, even complex ones, without coupling to their specific classes.
◆ **Singleton**: Ensure a class has one only instance, and provide a global point of access of it.
◆ **Adapter**: Convert interface of a class into one that another class requires e.g. (Cables, Media Player)
◆ **Bridge**: decouples an abstraction from its implementation, so that the two can vary independently.
◆ **Composite**: Compose objects into tree structures
◆ **Decorator**: Attach additional responsibilities to an object keeping the same interface.
◆ **Façade**: Provide a simple unified interface to a set of interfaces in a subsystem.
◆ **Flyweight**: It is used to create a large number of objects of almost similar nature
◆ **Proxy**: Surrogate or substitute for another object.
e.g. (ATM, Cheque, Credit Cards)

BEHAVIORAL PATTERN

◆ **Chain of Responsibility**: Avoid coupling sender to receiver by passing request along until someone handles it e.g. (Kiosk System)
◆ **Command**: it's used to manage algorithms, relationships and responsibilities between objects.
◆ **Iterator**: Provide a way to access the elements of an objects sequentially without exposing its underlying representation
◆ **Interpreter**: Language interpreter for a small grammar
◆ **Mediator**: Coordinates interactions between its associates
- Used to restore state of object to a previous state e.g. (Archive, Undo method)
◆ **Observer**: Dependents update automatically when subject changes e.g. (Social Media)
◆ **State**: Object whose behavior depends on its state
◆ **Strategy**: Abstraction for selecting one of many algorithms
◆ **Template Method**: A skeleton of an algorithm in an operation, and defers some steps to subclasses
◆ **Visitor**: Represents an operation applied to elements of an object structure.

DESIGN PRINCIPLES

◆ **Modularization**: Is the process of continuous decomposition of the software system until fine – grained components are created.
◆ **Abstraction**: A view of an object that focuses on the information to a particular purpose
◆ **Encapsulation**: exposing only the information that is essential while hiding details of how the services are carried out.
◆ **Cohesion**: is defined as the degree to which all elements of a module, class, or component work together as a functional unit.
High cohesion is good, and low cohesion is bad.
◆ **Coupling**: is defined as the degree of interdependence between two or more classes, modules, or components.
Tight coupling is bad, and loose coupling is good.
◆ **Sufficiency**: Measures how well the designed units are at providing only the services that are sufficient for achieving the intent (no more).
◆ **Completeness**: Measures how well the designed units are at providing the required services for achieving the intent (no less).

INSY 55 (1/2)

FORM - a business document that contains some predefined data and often includes some areas where additional data are to be filled in
- has a stylized format and is typically not in a simple row and column format
- examples: order forms, employment applications, and class registration sheets

REPORT - a business document that includes only predefined data
- a passive document that is used only for reading and viewing
- can also be printed to a computer file
- often has rows and columns of data
- for reading only and includes data on multiple unrelated records

Forms and Reports Format
1. **Meaningful Information**
- Only needed information should be displayed
- Information should be provided in a manner that is usable without modification
2. **Balance the Layout**
- Balanced on screen or on page
- Adequate spacing and margins
- All data and entry fields should be clearly labeled

3. **Design an Easy Navigation System**
- Show how to move forward and backward
- Show where you are

Most commonly used methods:
- Blinking and audible tones
- Color differences
- Intensity differences
- Size differences
- Reverse video
- Boxing
- Underlining
- All capital letters

Useful when:
- notifying users of errors in data entry or processing
- providing warnings to users regarding possible problems, such as unusual data values or an unavailable device
- drawing attention to keywords, commands, high-priority messages, and data that have changed or gone outside normal operating ranges

Color versus No-Color
Benefits:
- Soothes or strikes the eye.
- Accents an uninteresting display.
- Facilitates subtle discriminations in complex displays.
- Emphasizes the logical organization of information.
- Draws attention to warnings. Evokes more emotional reactions.

Designing Tables and Lists
1. **Use meaningful labels**
2. **Formatting columns, rows, and text**
3. **Formatting numeric, textual, and alphanumeric data**

Table versus Graphs
Use table for
- Reading individual data values

Use graph for
- Providing a quick summary of data
- Detecting trends over time
- Comparing points and patterns of different variables
- Forecasting activities
- Reporting huge amounts of information when relatively simple impressions are to be drawn

Assessing Usability
Characteristics of usability
- Speed
- Accuracy
- Satisfaction

Usability
- an overall evaluation of how a system performs in supporting a certain user for a certain task

Consistency
- influences users' ability to increase proficiency when interacting with a system

Other factors of usability
- Efficiency
- Ease
- Format

Usability Factors	Guidelines for Achievement of Usability
Consistency	• Consistent use of terminology, abbreviations, formatting, titles, and navigation within and across outputs • Consistent response time each time a function is performed
Efficiency	• Formatting should be designed with an understanding of the task being performed and the intended user • Text and data should be aligned and sorted for efficient navigation and entry • Entry of data should be avoided where possible
Ease	• Outputs should be self-explanatory and not require users to remember information from prior outputs in order to complete tasks • Labels should be extensively used, and all scales and units of measure should be clearly indicated
Format	• Information format should be consistent between entry and display • Format should distinguish each piece of data and highlight, not bury, important data • Special symbols (i.e., decimal places, dollar signs, and +/- signs) should be used as appropriate
Flexibility	• Information should be viewed and retrieved in a way most convenient to the user • Users should be given options for the sequence in which to enter or view data and for use of shortcut keystrokes, and the system should remember where the user stopped during the last use of the system

Characteristic	Consideration for Form and Report Design
User	• Issue related to experience, skills, motivation, education, and personality should be considered
Task	• Tasks differ in amount of information that must be obtained from or provided to the user • Task demands such as time pressure, cost of errors, and work duration will influence usability
System	• The platform on which the system is constructed will influence interaction styles and devices
Environment	• Social issues such as the users' status and role should be considered in addition to environment concerns such as lighting, sound, task interruptions, temperature, and humidity • The creation of usable forms and reports may necessitate changes in the users' physical work facilities

Implementation - activities that occur before the system is turned over to its users

Coding - the process where the physical design specifications developed by the analysis team are converted into computer code by the programming team

Testing - the process of examining a product to ascertain what defects it contains
- product can be tested through reviewing their construction and composition or through exercising their function and examining the results

Types of Testing
Unit Testing - also called module testing
- the process of testing individual code modules before they are integrated with other modules

Integration Testing - combining modules and testing them
- identify errors that were not or could not be detected by unit testing individual modules

System Testing
- the programs are integrated into systems
- performed first by developers or test personnel

INSY 55 (2/2)

- performed mostly at the end of each iteration to identify significant issues

Acceptance Testing - testing the system in the environment where it will eventually be used. Way for users to verify if the system meets their requirements the last round of testing before the system is handed over to its users

Two types:
- Alpha testing
- Beta testing

Alpha Testing
- User testing of a completed information system using simulated data

Types of tests done:
- Recovery testing
- Security testing
- Stress testing

Beta Testing - User testing of a completed information system using real data in the real user environment
- can be considered as a preparation of the installation phase

Installation - the process of moving from the current information system to the new one

Direct Installation - Changing over from the old information system to a new one by turning off the old system when the new one is turned on

Parallel Installation - Running the old information system and the new one at the same time until management decides the old system can be turned off
- Useful mainly when a system is large, complex, and composed of relatively independent subsystems

Documentation - provides information to users on how a system is operated and maintained
- provides information required for future modifications or re-implementation

Types of Documentation

1. **System Documentation** - records detailed information about a system's design specifications, its internal workings, and its functionality provide information to designers and developers who will maintain or re-implement the system

TYPES:
Internal documentation
External documentation

User Documentation - provides an ongoing support for end users of the system

Two aspects of an organization's computing infrastructure:
Training
Support

Computer Infrastructure - all of the resources and practices necessary to aid people adequately use computer systems to do their primary work

Classification of users:
End users

Methods for user support: On-line documentation and troubleshooting

Resident experts:
A help desk
Technical support

Maintenance - modification of a software product after delivery

Types of Maintenance

1. **Corrective Maintenance** - changes made to repair flaws in its design, coding, or implementation of the system
- remove errors or bugs from the system, procedures, hardware, network, data structures, and documentation
- adds a little or no value to the organization

2. **Adaptive Maintenance** - making changes to an information system to develop its functionality to changing business needs or to migrate it to a different operating environment
- usually a small part of an organization's maintenance effort

3. **Perfective Maintenance** - improve the system's efficiency, reliability, functionality, or maintainability
- considered to be proactive
- fix the system before it breaks

4. **Preventive Maintenance** - changes made to a system to lessen the chance of future system failure
- anticipate problems and correct them before they occur

Cost is an important expenditure in information system maintenance.
- The high costs associated with maintenance mean that the factors. influencing maintainability of systems should be understood.
- Maintainability

Approaches to Organizing System Maintenance

1. **Separate Approach** - maintenance group rejects new projects unless properly and thoroughly tested
- forces better documentation and formalizes the conversion from development to operations status and change procedures

2. **Combine Approach** - both groups form one major group of the information system
- users may be unable to distinguish work as for development or maintenance

3. **Functional Approach** - systems professionals are removed from IS and assigns them to business functions for both development and maintenance