DESIGN CONCEPTS

Lecture # 20

QUALITY ATTRIBUTES (FURPS)

- Functionality
  - Assessed by evaluating the feature set and capability of the program, generality of the functions delivered, Security of the overall system
- Usability
  - Assess by human factors, overall aesthetics, consistency, and documentation
- Reliability
  - Evaluated by measuring the frequency and severity of failure, the accuracy of output results, MTTF, ability to recovery from failure, and predictability of the program
- Performance
  - Measured by processing speed, response time, resources, throughput, and efficiency
- Supportability
  - Ability to extend the program, adaptability, serviceability

DESIGN PRINCIPLES

1. The design process should not suffer from ‘tunnel vision.’
2. The design should be traceable to the analysis model.
3. The design should not reinvent the wheel.
4. The design should “minimize the intellectual distance” between the software and the problem as it exists in the real world.
5. The design should exhibit uniformity and integration.
6. The design should be structured to accommodate change.
7. The design should be structured to degrade gently, even when aberrant data, events, or operating conditions are encountered.
8. Design is not coding, coding is not design.
9. The design should be assessed for quality as it is being created, not after the fact.
10. The design should be reviewed to minimize conceptual (semantic)errors.

DESIGN CONCEPTS

- Abstraction—data, procedure, control
- Architecture—the overall structure of the software
- Patterns—”conveys the essence” of a proven design solution
- Separation of concerns—any complex problem can be more easily handled if it is subdivided into pieces
- Modularity—compartmentalization of data and function
- Information Hiding—controlled interfaces
- Functional independence—single-minded function and low coupling
- Refinement—elaboration of detail for all abstractions
- Aspects—a mechanism for understanding how global requirements affect design
- Refactoring—a reorganization technique that simplifies the design

ABSTRACTION

- Abstraction allows designers to focus on solving a problem without being concerned about irrelevant lower level details.
- There are two types of abstraction available :
  - Procedural abstraction – a sequence of instructions that have a specific and limited function. E.g.: Word OPEN for a door.
  - Data abstraction – a named collection of data that describes a data object. Data abstraction for door would be a set of attributes that describes the door (e.g. door type, swing direction, weight, dimension

ARCHITECTURE

- "The overall structure of the software and the ways in which structure provides conceptual integrity for a system."
  - Structural properties
    - This aspect of the architectural design representation defines the components of a system (e.g., modules, objects, filters) and the manner in which those components are packaged and interact with one another.
  - Extra-functional properties
    - The architectural design description should address how the design architecture achieves requirements for performance, capacity, reliability, security, adaptability, and other system characteristics.
  - Families of related systems
    - The architectural design should draw upon repeatable patterns that are commonly encountered in the design of families of similar systems. In essence, the design should have the ability to reuse architectural building blocks.

ARCHITECTURE

- Some of the Architecture models are described below,
  - Structural models – architecture as organized collection of components
  - Framework models – attempt to identify repeatable architectural patterns
  - Dynamic models – indicate how program structure changes as a function of external events
  - Process models – focus on the design of the business or technical process that system must accommodate
  - Functional models – used to represent system functional hierarchy

PATTERN

- A design pattern describes a design structure and that structure solves a particular design problem in a specified content.
- Software engineer can use the design pattern during the entire software design process.
- Each pattern is to provide an insight to a designer who can determine the following-:
  - Whether the pattern can be reused.
  - Whether the pattern is applicable to the current project.
  - Whether the pattern can be used to develop a similar but functionally or structurally different design pattern.

SEPARATION OF CONCERNS

- A concern is a feature or behavior that is specified as part of the requirements model for the software
- Any complex problem can be more easily handled if it is subdivided into pieces that can each be solved and/or optimized independently
- By separating concerns into smaller, and therefore more manageable pieces, a problem takes less effort and time to solve.