SWEBOK KA #2: Software Design

The Software Engineering Body of Knowledge (SWEBOK) features 11 knowledge areas (KAs). The second KA is Software Design. The Software Design KA is focused on analyzing software requirements to develop software architectural (top-level) design and software detailed design. It includes six topics, as shown in Figure 1. These topics are Software Design Fundamentals, Key Issues in Software Design, Software Structure and Architecture, Software Design Quality Analysis and Evaluation, Software Design Notations, and Software Design Strategies and Methods.

The Software Design Fundamentals topic introduces concepts, notions, and technology that as a basis for software design. General design concepts are common design elements such as goals, constraints, alternatives, representations, and solutions. It provides the context of software design as well as the software design process, which is composed of two steps: architectural design and detailed design. It also covers enabling techniques, such as abstraction, coupling and cohesion, decomposition and modularization, encapsulation/information hiding, separation of interface and implementation, and sufficiency, completeness, and primitiveness.

The Key Issues in Software Design topic discusses some of the issues that must be addressed during software design. These issues include concurrency, control and handling of events, distribution of components, error and exception handling and fault tolerance, interaction and presentation, and data persistence.

The Software Structure and Architecture topic covers the aspects of architectural design. The first aspect of the topic is architectural structures and viewpoints, which addresses the multi-faceted nature of design. Design patterns can be used to describe details at a lower level. Finally, the concept of families of programs and frameworks is introduced as a potential approach to software reuse.

The Software Design Quality Analysis and Evaluation topic involves the quality and evaluation aspects of software design. Quality attributes include concepts such as maintainability, portability, testability, traceability, correctness, and robustness. Quality analysis and evaluation techniques involved tools and techniques such as software design reviews, static analysis, and simulation and prototyping. There are different measures to assess the software, grouped into function-oriented design measures and object-oriented design measures.

The Software Design Notations topic describes different notations used to represent software design. Structural descriptions represent the structural aspects of software. Class and object diagrams, deployment diagrams, and interface description languages are types of structural notation. Behavioral descriptions describe the behavior of software. Activity diagrams, flowcharts, sequence diagrams, and statechart diagrams are types of behavioral notation.

The Software Design Strategies and Methods topic includes strategies to guide the design process. There are general strategies, such as divide and conquer and stepwise refinement. Function-oriented design is a classic method in which decomposition centers on determining the main software functions. Object-oriented design is a method based on objects and their interactions. Data-structure-centered design is based on data structures used by the software. Component-based design is based on the integration of software components. There are also other design methods covered, such as formal and rigorous methods and transformational methods.

Figure 1. Breakdown of Topics for the Software Design Knowledge Area

