Software Design, Part I

abstract design: 1. A generic form that needs specialization (further design work) to produce concrete designs. 2. Design aimed at producing designs; for example, design patterns.

architectural design: In system/software system engineering: 1. The process of defining a collection of hardware and software components and their interfaces to establish a framework for developing a system/software system. 2. The result of the architectural design process. Also called high-level design, internal specification, preliminary design, system design, top-level design. [ANSI/IEEE Std 610.12-1990]

architectural design phase: The lifecycle phase in which a system's general architecture is developed, thereby fulfilling the requirements laid down by the software requirements document and detailing the implementation plan in response to it.

architectural design review: A joint acquirer-supplier review to evaluate the technical adequacies of the software architectural design as depicted in the software design descriptions.

architectural structure: A physical or logical layout of the components of a system design and their internal and external connections. Examples are function-oriented (structured) design, object-oriented design, and data-structure-oriented design.

architectural style: 1. Defines a family of systems in terms of a pattern of structural organization. Commonly used styles include pipes and filters, layers, rule-based systems, and blackboards. 2. Characterizes a family of systems that are related by sharing structural and semantic properties.

architectural view: A representation of a whole system from the perspective of a related set of concerns. [IEEE Std 1471-2000]

architectural viewpoint: 1. A specification of the conventions for constructing and using a view. 2. A pattern or template from which to develop individual views by establishing the purposes and audience for a view and the techniques for its creation and analysis. [IEEE Std 1471-2000]

baseline design: A system design that has been agreed on by all stakeholders interested in the system development.

bottom-up design: The process of designing a system by identifying low-level components, designing each component separately, and then designing a structure to integrate the low-level components into larger and larger subsystems until the design is finished.

cohesion: In software design, a measure of the strength of association of the elements within a module. Contrast with coupling.

coupling: In software design, a measure of the interdependence among modules in a computer program; the amount of information shared between two modules. Contrast with cohesion.

critical design review: A somewhat obsolete software development term for what is now called detailed design review.

database design specification: A document that describes the content and format of the permanent or semipermanent data necessary for the software to carry out its functions.

data-structure-oriented design: A design methodology used for business applications by basing the design on the logical data structures of the program specification. Examples include the Jackson System Design and Warnier-Orr methods.

design: 1. The process of defining the software architecture, components, modules, interfaces, and data for a software system to satisfy specified requirements. 2. The results of the design process.

design analysis: 1. The evaluation of a design to determine correctness with respect to stated requirements, conformance to design standards, system efficiency, and other criteria. 2. The evaluation of alternative design approaches. [ANSI/IEEE Std 830-1984]

design analyzer: An automated design tool that accepts information about a program's design and produces such outputs as module hierarchy diagrams, graphical representations of control and data structure, and lists of accessed data blocks.

design concept: A fundamental idea that can be applied to designing a system (for example, information hiding).