Software design description

A **software design description** (a.k.a. **software design document** or **SDD**; just **design document**; also Software Design Specification) is a written description of a software product, that a software designer writes in order to give a software development team overall guidance to the architecture of the software project. An SDD usually accompanies an architecture diagram with pointers to detailed feature specifications of smaller pieces of the design. Practically, the description is required to coordinate a large team under a single vision, needs to be a stable reference, and outline all parts of the software and how they will work.

Contents

Composition

The SDD usually contains the following information:

1. The *data design* describes structures that reside within the software. Attributes and relationships between data objects dictate the choice of data structures.
2. The *architecture design* uses information flowing characteristics, and maps them into the program structure. The transformation mapping method is applied to exhibit distinct boundaries between incoming and outgoing data. The data flow diagrams allocate control input, processing and output along three separate modules.
3. The *interface design* describes internal and external program interfaces, as well as the design of the human interface. Internal and external interface designs are based on the information obtained from the analysis model.
4. The *procedural design* describes structured programming concepts using graphical, tabular and textual notations.

These design mediums enable the designer to represent procedural detail, that facilitates translation to code. This blueprint for implementation forms the basis for all subsequent software engineering work.

IEEE 1016

IEEE 1016-2009, titled *IEEE Standard for Information Technology—Systems Design—Software Design Descriptions*,[1] is an IEEE standard that specifies "the required information content and organization" for an SDD.[2] IEEE 1016 does not specify the medium of an SDD; it is "applicable to automated databases and design description languages but can be used for paper documents and other means of descriptions."[3]

The 2009 edition was a major revision to IEEE 1016-1998, elevating it from recommended practice to full standard. This revision was modeled after IEEE Std 1471-2000, *Recommended Practice for Architectural Description of Software-intensive Systems*, extending the concepts of view, viewpoint, stakeholder, and concern from architecture description to support documentation of high-level and detailed design and construction of software. [IEEE 1016, *Introduction*]

Following the IEEE 1016 conceptual model, an SDD is organized into one or more design views. Each design view follows the conventions of its design viewpoint. IEEE 1016 defines the following design viewpoints for use:[4]

- Context viewpoint
- Composition viewpoint
- Logical viewpoint
- Dependency viewpoint
- Information viewpoint
- Patterns use viewpoint
- Interface viewpoint
- Structure viewpoint
- Interaction viewpoint
- State dynamics viewpoint
- Algorithm viewpoint
- Resource viewpoint

In addition, users of the standard are not limited to these viewpoints but may define their own.[5]

See also

- Game design document
- High-level design
- Low-level design

References

1. *IEEE Standard for Information Technology — Systems Design — Software Design Descriptions*. IEEE. 2009-07-20. doi:10.1109/IEEESTD.2009.5167255 (https://doi.org/10.1109%2FIEEESTD.2009.5167255). ISBN 978-0-7381-5925-6.
2. IEEE 1016, *Abstract*.
3. IEEE 1016, *Abstract*.
4. IEEE 1016, Clause 5.
5. IEEE 1016, sub-clause 4.5.

External links

- IEEE 1016 website (http://www.iso-architecture.org/ieee-p1016/)