WIKIPEDIA

Systems design

Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering.[1][2]

Contents

Overview

If the broader topic of product development "blends the perspective of marketing, design, and manufacturing into a single approach to product development,"[3] then design is the act of taking the marketing information and creating the design of the product to be manufactured. Systems design is therefore the process of defining and developing systems to satisfy specified requirements of the user.

Until the 1990s, systems design had a crucial and respected role in the data processing industry. In the 1990s, standardization of hardware and software resulted in the ability to build modular systems. The increasing importance of software running on generic platforms has enhanced the discipline of software engineering.

Architectural design

The architectural design of a system emphasizes the design of the system architecture that describes the structure, behavior and more views of that system and analysis.

Logical design

The logical design of a system pertains to an abstract representation of the data flows, inputs and outputs of the system. This is often conducted via modelling, using an over-abstract (and sometimes graphical) model of the actual system. In the context of systems, designs are included. Logical design includes entity-relationship diagrams (ER diagrams).

Physical design

The physical design relates to the actual input and output processes of the system. This is explained in terms of how data is input into a system, how it is verified/authenticated, how it is processed, and how it is displayed. In physical design, the following requirements about the system are decided.

1. Input requirement,
2. Output requirements,
3. Storage requirements,
4. Processing requirements,
5. System control and backup or recovery.

Put another way, the physical portion of system design can generally be broken down into three sub-tasks:

1. User Interface Design
2. Data Design
3. Process Design

User Interface Design is concerned with how users add information to the system and with how the system presents information back to them. Data Design is concerned with how the data is represented and stored within the system. Finally, Process Design is concerned with how data moves through the system, and with how and where it is validated, secured and/or transformed as it flows into, through and out of the system. At the end of the system design phase, documentation describing the three sub-tasks is produced and made available for use in the next phase.

Physical design, in this context, does not refer to the tangible physical design of an information system. To use an analogy, a personal computer's physical design involves input via a keyboard, processing within the CPU, and output via a monitor, printer, etc. It would not concern the actual layout of the tangible hardware, which for a PC would be a monitor, CPU, motherboard, hard drive, modems, video/graphics cards, USB slots, etc. It involves a detailed design of a user and a product database structure processor and a control processor. The H/S personal specification is developed for the proposed system.

Related disciplines

- Benchmarking – is an effort to evaluate how current systems perform
- Computer programming and debugging in the software world, or detailed design in the consumer, enterprise or commercial world - specifies the final system components.
- Design – designers will produce one or more 'models' of what they see a system eventually looking like, with ideas from the analysis section either used or discarded. A document will be produced with a description of the system, but nothing is specific – they might say 'touchscreen' or 'GUI operating system', but not mention any specific brands;
- Requirements analysis – analyzes the needs of the end users or customers
- System architecture – creates a blueprint for the design with the necessary structure and behavior specifications for the hardware, software, people and data resources. In many cases, multiple architectures are evaluated before one is selected.
- System testing – evaluates the system's actual functionality in relation to expected or intended functionality, including all integration aspects.

Alternative design methodologies

Rapid application development (RAD)

Rapid application development (RAD) is a methodology in which a system designer produces prototypes for an end-user. The end-user reviews the prototype, and offers feedback on its suitability. This process is repeated until the end-user is satisfied with the final system.

Joint application design (JAD)

Joint application design (JAD) is a methodology which evolved from RAD, in which a system designer consults with a group consisting of the following parties:

- Executive sponsor
- System Designer
- Managers of the system

JAD involves a number of stages, in which the group collectively develops an agreed pattern for the design and implementation of the system.

See also

- Arcadia (engineering)
- Architectural pattern (computer science)
- Configuration design
- Electronic design automation (EDA)
- Electronic system-level (ESL)
- Embedded system
- Graphical system design
- Hypersystems
- Modular design
- Morphological analysis (problem-solving)
- SCSD (School Construction Systems Development) project
- System information modelling
- System development life cycle (SDLC)
- System engineering
- System thinking
- TRIZ

References

1. This article incorporates public domain material from the General Services Administration document "Federal Standard 1037C" (http://www.its.bldrdoc.gov/fs-1037/fs-1037c.htm).
2. This article incorporates public domain material from the United States Department of Defense document "Dictionary of Military and Associated Terms".
3. Ulrich & Eppinger (2000). *Product Design & Development*. Irwin McGraw-Hill. ISBN 0-07-229647-X.

Further reading

- Bentley, Lonnie D., Kevin C. Dittman, and Jeffrey L. Whitten. *System analysis and design methods.* (1986, 1997, 2004).
- C. West Churchman (1971). *The Design of Inquiring Systems: Basic Concepts of Systems and Organization*. New York: Basic Books. ISBN 0-465-01608-1.
- William Gosling (1962). *The design of engineering systems*. New York: Wiley.
- Hawryszkiewycz, Igor T. *Introduction to system analysis and design.* Prentice Hall PTR, 1994.
- Levin, Mark Sh. *Modular system design and evaluation.* Springer, 2015.
- Maier, Mark W., and Rechtin, Eberhardt (2000). *The Art of System Architecting* (Second ed.). Boca Raton: CRC Press.
- Saltzer, J.H.; et al. (November 1984). "End-to-End arguments in System Design". *ACM Transactions in Computer Systems*. **2** (4): 277–288.
- Ulrich, Karl T.; Eppinger, Steven D. (2000). *Product Design and Development* (Second ed.). Boston: Irwin McGraw-Hill.