

Definitions

(Definitions selected from *Merriam-Webster 7th Intercollegiate Dictionary*)

- What is design?

Definitions

- What is *design*? **deliberative, purposive planning**

Definitions

- What is *design*? deliberative, purposive planning
- What is *engineering*?

Definitions

- What is *design*? deliberative, purposive planning
- What is *engineering*? skillful or artful contrivance
 - Role of scientific and mathematical principles

Definitions

- What is *design*? deliberative, purposive planning
- What is *engineering*? skillful or artful contrivance
 - Role of scientific and mathematical principles
- What is *craft*?

Definitions

- What is *design*? deliberative, purposive planning
- What is *engineering*? skillful or artful contrivance
 - Role of scientific and mathematical principles
- What is *craft*? **skilled occupation**

Definitions

- What is *design*? deliberative, purposive planning
- What is *engineering*? skillful or artful contrivance
 - Role of scientific and mathematical principles
- What is *craft*? skilled occupation
- What is *art*?

Definitions

- What is *design*? deliberative, purposive planning
- What is *engineering*? skillful or artful contrivance
 - Role of scientific and mathematical principles
- What is *craft*? skilled occupation
- What is *art*? the conscious use of skill, taste, and artistic imagination in the production of aesthetic objects

Software Design

- *Software design* is the process of building a program while satisfying a problem's functional requirements and without violating its non-functional constraints
 - Interesting design typically involves finding a *satisficing* tradeoff among the constraints
- Software design is normally broken into two phases: high level or *architectural* design and low level or *detail* design

Architectural Design

- *Architectural design* is the process of identifying and assigning the responsibility for aspects of functional behavior to various modules or components of a software system
 - Strategy for dealing with non-functional constraints must be determined
 - The communication interfaces among the components must also be specified
- We will devote significant effort later in the term to considering software architecture

Detail Design

- *Detail design* is the process of specifying the behavior of each of the components
 - Data structure representation
 - Procedural; algorithm selection

Detailed Design – Data Structures

The primary activity during data design is to select logical representations of data objects identified during the requirements definition and specification phase. The selection process may involve algorithmic analysis of alternative structures in order to determine the most efficient design or may simply involve the use of a set of modules that provide the operations upon some representation of an object.

-- Wasserman

Detailed Design - Procedures

- After data and program structure have been established, it becomes necessary to select algorithms and specify procedural details
- Design Notations
 - Pseudo code
 - Structured programming
 - Graphical design notation (like flowcharts)
 - Tables (decision tables)
 - Program design language (PDL)

Approaches to Design

- There are many approaches to design.
 - Some espouse a particular point of view as to how best to structure a system (for example, object-oriented design)
 - Some of them are intended for a particular class of application (e.g. real-time systems)
 - And some are intended for a specific part of the system structure (such as user-interface design).
- All approaches to design, however, contain three aspects that may be compared: method, representation and validation

Design Method

- A *method* is a systematic sequence of steps that a design team uses to solve a problem
- A method usually encourages a particular viewpoint on its users
 - For example, the object-oriented design method encourages designers to view problems in terms of their constituent objects before worrying about functional behavior
 - Other methods we will refer to: structured design, role-based design
- A design method acts as a discipline on the designers, forcing them to order their thoughts and activities in certain ways

Design Representation

- Non-trivial problems require design solutions to be expressed using some form of representation
- The design representation can be either graphical or textual, but it is sufficiently formal that it can be checked for adherence to certain properties
- A design representation serves two audiences
 - The designers themselves. That is, the discipline imposed by the method and the representation encourages early detection of missing or inconsistent aspects of a proposed solution
 - Design representations are also read by other stakeholders. These might be coders or testers. Notably, they might subsequently be maintainers trying to understand the designers' original intent

Design Validation

- Designs must be checked to assure that they accomplish their intended goals. A good design method will have an associated validation technique
 - For example, Structured Design measures the extent of *coupling* and *cohesion* to check the quality of its designs
- Design inspections, walk-throughs and reviews are general techniques for assessing design quality. They can provide a process substrate to method-specific validation techniques

Issues in Design - Content

- Functional behavior versus non-functional constraints
- Architectural versus detail design
- Specification/*what* versus design/*how*
- Application specificity

Issues in Design - Method

- Top-down versus bottom-up
- Verbs/functions versus nouns/objects
- Conceptual integrity versus cooperative work
- Long-term maintainability versus short-term schedule
- Role of tools

Issues In Design - Representation

- Discipline versus communication
- Text versus graphics
- Formal versus informal
- Syntax versus semantics
- Static/data/declaration versus dynamic
/control/procedure

Issues of Design - Validation

- Independence of validators
- Dependence on design method
- On-going versus after-the-fact
 - "Bad smells" and refactoring

Design Information - Traditional

- Subcomponents
 - Data / data flows
 - Processes / functions
- Communication flow
 - Control regime / control flows
- Performance
- Resources

Design Information - IEEE Standard (1016)

- Designer
- Source
- Type
- Purpose
- Function
- Subordinates
- Dependencies
- Resources
- Interfaces
- Processing
- Data
- Tradeoffs
- Assumptions
- References
- Standard views

Leonardo Objects

- **Stakeholders**: views/interests; work procedures
- **Issue bases**: issues, alternatives, analyses, resolutions (commitments, decisions), **rationale**, impacts, conflicts, suggestions
- Notes/comments; TBDs/gaps, questions
- Artifacts: code, documentation, design
- Virtual objects: views/abstractions, derived info

Leonardo Relationships

- Temporal relations: histories, schedules, transformations, **versions** (revisions, releases, **variants**)
- **Dependencies**: supports, usages, sharing
- **Constraints**: internal, external (requirements, specifications)
- **Aggregates**: configurations, packages, components
- Decompositions
- Working relationships

Design Rationale

- *Design decisions* are explicit choices of how to trade off two non-functional aspects of a design, such as speed versus size
- Design decisions should be explicitly documented
- Documentation of design decisions is called *design rationale*

Design Concepts

- Conceptual integrity
- Coupling / cohesion
- Information hiding
- Abstraction / refinement
- Aesthetics / elegance

Conceptual Integrity

Of these [thoughts] one of the very first that occurred to me was, that there is seldom so much perfection in works composed of many separate parts, upon which different hands had been employed, as in those completed by a single master.

Rene Descartes
Discourse on a Method

Conceptual Integrity - 2

I will contend that conceptual integrity is the most important consideration in system design. It is better to have a system omit certain anomalous features and improvements, but to reflect one set of design ideas, than to have one that contains many independent and uncoordinated ideas.

Fred Brooks

The Mythical Man-Month

© 2009 Spencer Rugaber

Conceptual Integrity - 3

My experience [...] is that design and art are rarely the product of a single mind. [...] There are two ways that the single mind idea doesn't work [...] The easy one to understand is the ways that others are constantly helping. As a writer I encounter help in all sorts of ways [...] What is common is that the thing being created [...] participated in its own creation. This is the other way that the single mind idea doesn't work. [...] A designer struggles to create a first draft of a design, then, not to be merely an important step in the design process, but in order to create a collaborator that will make the remainder of the design process easier [...]. A collaborator that triggers thoughts and directions that would never occur to the designer were the artifact not sitting there, staring back.

Richard Gabriel
Designed as Designer

Coupling / Cohesion

- *Coupling* - the extent to which two components depend on each other for successful execution
 - Low coupling is good
- *Cohesion* - the extent to which a component has a single purpose or function
 - High cohesion is good

Information Hiding

- Use of encapsulation to hide implementation details
- Reduces intercomponent coupling thereby supporting subsequent maintenance

Abstraction / Refinement

- All design methods support the idea of abstraction and refinement
- That is, designs are expressed at different levels of detail with correspondences between the levels
- Various conceptual devices (abstraction mechanisms) are used to refine a design at one level into one at a lower level

Abstractions Mechanisms

- Declarative: *what*, not *how*
- Aggregation: container, not contents
- Generalization: class, not individuals
- Parameterization binding details later
- Non-determinism: leaving choices unspecified

Aesthetics

- Aquinas: *Ad pulchritudinem tria requiruntur: Integritas, consonantia, claritas.*
 - James Joyce translates it as: *Three things are needed for beauty: wholeness, harmony and radiance.*
 - In terms of software design: completeness, consistency, and conceptual integrity
- Pascal: *Je n'ai fait celle-ci plus longue que parce que je n'ai pas eu le loisir de la faire plus courte.*
 - Literally: I made this [letter] very long, because I did not have the leisure to make it shorter.
 - Translation: I would have written a shorter letter, but I did not have the time.
 - In terms of software design: it takes time to understand a problem and solve it in an elegant way

Philosophy (Ehn)

- Descartes - analytical, use of models
- Marx - understanding the social context of design solutions
- Heidegger - tools for accomplishing goals
- Wittgenstein - language games

Observations

- Design is the most creative part of software development
- Consequently, overall system quality is highly dependent on the designs produced
- A key determinant of design quality is extent of experience on similar projects by members of the design team