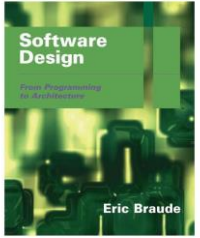
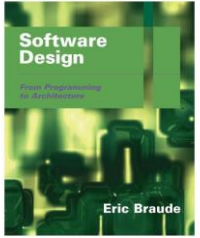

Lec 05 - Software Design

Topic covered

- ✧ Introduction to Software Design.
- ✧ Goals of software design.



What is Software Design?



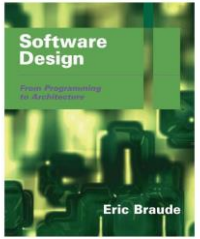
- ✧ A “**Software design**” is a set of documents on whose basis a software can be fully programmed.
- ✧ A complete design should be so explicit that a programmer could code the application from it without the need for any other documents.
- ✧ Software designs are like the blueprints of a building.

What is Software Design?

- ✧ The purpose of design is to produce a solution to a problem:
 - The problem: requirements specification.
 - The solution: your description of how the requirements are to be met.
- ✧ Design is the creative process of describing and transforming a problem into a solution.
- ✧ A set of documents on whose basis a software can be fully programmed.
- ✧ Software design describe:
 - The system well enough that coders can build it and deploy it without serious problems.
 - All the parts of the system and how they fit together (architecture, high-level design)
 - Each part in detail so that it can be coded.

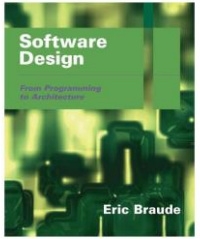
Goals of Software Design

- ✧ Correctness
- ✧ Robustness
- ✧ Flexibility
- ✧ Reusability
- ✧ Efficiency
- ✧ Reliability
- ✧ Usability

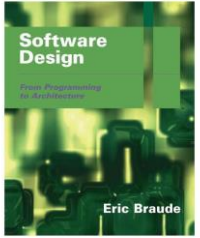


Correctness

- ✧ Software design must satisfy the requirements for the application
- ✧ We are primarily responsible for ensuring that our code does what it's intended to.



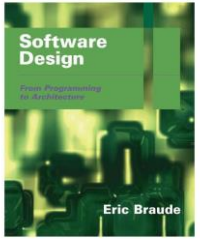
Robustness



✧ A design or implementation is robust if it is able to handle miscellaneous and unusual conditions such as:

- Bad data.
- User error.
- Programmer error.
- Environmental conditions.

Flexibility



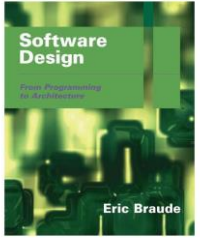
- ✧ Requirements of an application can change in many ways.
 - Design should be flexible to accommodate these changes.

- ✧ Aspects of flexibility
 - Obtaining more or less of what's already present
 - e.g. handle more kinds of account

 - Adding new functionality
 - Add withdraw to existing deposit function

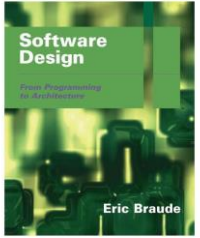
 - Change functionality
 - Allow withdrawal to create an overdraft

Reusability



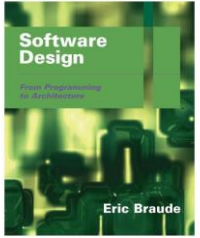
- ✧ The trend in software is to reuse parts among applications
 - Example, Java API --- a large, extensive body of widely reused classes
- ✧ Types of reusability
 - Object code
 - Example: sharing dll's between word processor and spreadsheet
 - Classes – in source code form
 - Example: *Customer* class used by several applications
 - Assemblies of Related Classes
 - Example: the java.awt package

Efficiency



- ✧ Efficiency refers to the use of available machine cycles and memory.
- ✧ Create designs and implementations that are as fast as required, and which make use of no more than available memory.

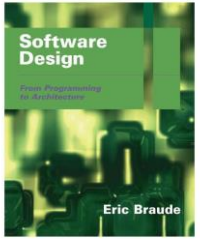
Reliability



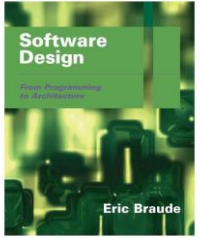
- ✧ An application is reliable if it is relatively defect free.
- ✧ Metric for reliability.
 - Average time between failures.
- ✧ Clean designs make it easier for developers to produce error-free applications.

Usability

- ✧ An application has high usability if users find it easy to use.
- ✧ Usability is attained through human-interface design.



Summary



- ✧ A “Software design” is a set of documents on whose basis a software can be fully programmed.
- ✧ Goals of software design
 - Correctness
 - Robustness
 - Flexibility
 - Reusability
 - Efficiency
 - Reliability
 - Usability