

Software Design

System Requirements

- Functional requirements
- Non-Functional
 - Efficiency
 - Quality
 - Process
 - Platform
- Cost / Budget

System Design - Top-Down approach or Bottom-Up approach

Top-Down <ul style="list-style-type: none">- Looks at actual goal of system- Creates sub problems- Works its way down to the smallest problems	Bottom-Up <ul style="list-style-type: none">- Looks at Utilities, Small components- Aggregates a large solution- You can build a program to be highly modular
---	--

Aspects of Design

Architecture <ul style="list-style-type: none">- Division of system into components- Designing the Problem	User Interface <ul style="list-style-type: none">- The way the program looks- How the user interacts with the system	Algorithms <ul style="list-style-type: none">- Handling of the data
--	--	--

Goals of a good design

- Increase Profit / Decrease cost / Increase Revenue
- Ensuring that all requirements are met
- Accelerating Development

Design Principle 1 - Divide and Conquer

Separate each part and work on small components. Delegate different responsibilities of the project. Parts can be modular.

Design Principle 2 - Increase Cohesion where possible

A System of modules has high cohesion if it keeps similar modules closely related.