Design

Alex Sverdlov
alex@theparticle.com

1 Introduction

Design is a process of turning user requirements into a blue-print for building software.

There are many dimensions to design. Essentially, we are trying to optimize these major factors:

- Abstraction: Cutting a big-problem into smaller-problems can be done at multiple levels. For every problem, there is an appropriate level of abstraction—going a bit too much into details or a few levels up does not present a clear view of the problem or the solution.
- Architecture: The big-picture items in a software project.
- Patterns: Often the same problems keep showing up over and over again, and there are well thought-out ways of solving them. There are multiple books on design patterns.
- Separation of Concerns: more independence is usually better.
- Modularity: a module is often a self-contained blob of code that implements some business logic.
- Information Hiding: modules should manage access to their internal state via APIs.
- Functional independence: the idea is to build modules that are as independent of other modules as possible—with little interaction with other modules.
- Factoring/Refactoring: the idea is to factor out commonalities as opposed to repeat them. For example, we can sprinkle security all over the place, or centralize it into a security module that everyone uses.