

# The Builder Pattern

John Sonmez

<http://simpleprogrammer.com>

John.sonmez@gmail.com



# Introduction

- **Overview of Builder**
- **Problem 1: Too many parameters**
- **Problem 2: Order dependent**
- **Problem 3: Different constructions**
- **What We Built**
- **Variations**

# Overview of Builder

- Separates the construction of a complex object from its representation so that the same construction process can create different representations.



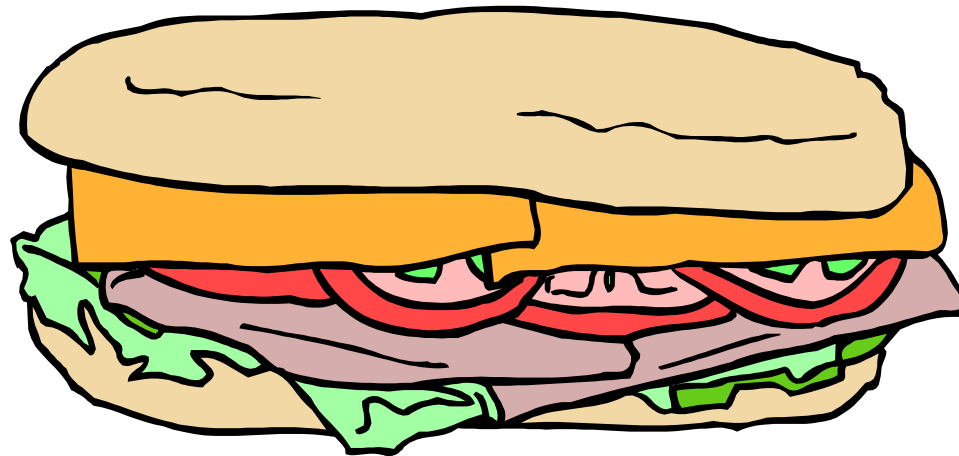
# A Dialog

- **Hey!**
  - Yeah?
- **I want a sandwich!**
  - Ok, first what kind of bread?
- **Wheat**
  - Ok, what size?
- **1 foot long**
  - Ok, what kind of meat and cheese?
- **Turkey and Swiss**
  - Do you want it toasted?
- **Yes**
  - What kind of vegetables?
- ...

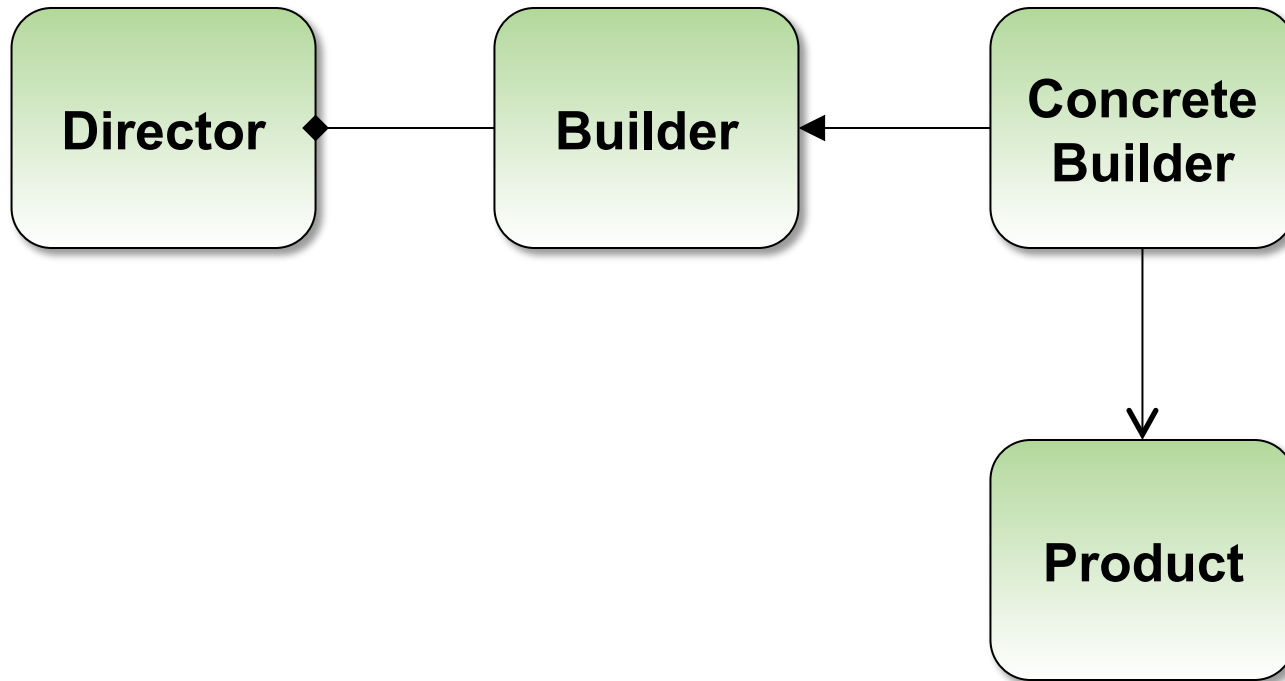


# Another Dialog

- **Hey!**
  - Yeah?
- **I want a sandwich that I have designed called the super turkey pickler**
  - Do you have instructions?
- **Yes, here is a list of the ingredients**
  - Ok, I can follow my steps and use those ingredients



# The Builder Pattern



# Builder Pattern Roles

- **Director**
  - Uses the **Concrete Builder**
  - Knows how to build
  - Client code calls directly



# Builder Pattern Roles

- **Builder**
  - Abstract interface or class
  - Defines steps
  - Holds instance of **Product**

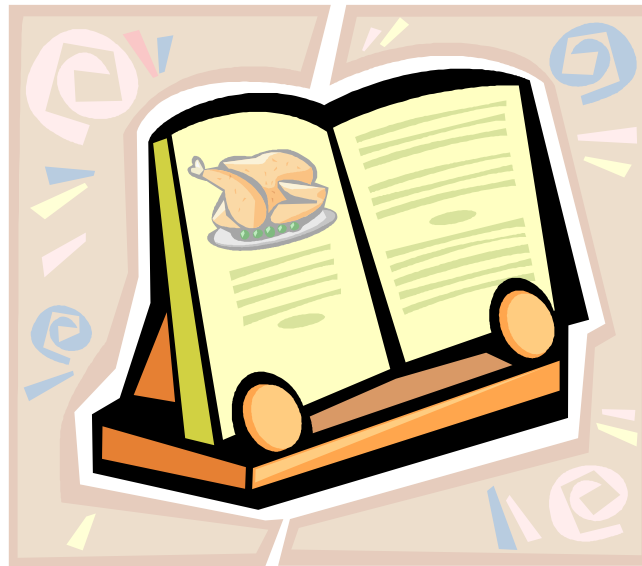




# Builder Pattern Roles

- **Concrete Builder**

- Should be more than one of these
- Provides an implementation for interface defined by the **Builder**
- A recipe



# Builder Pattern Roles

- **Product**

- What is being built
- Not a different type, but different data



# Summary

- **Overview of Builder**
- **Problem 1: Too many parameters**
- **Problem 2: Order dependent**
- **Problem 3: Different constructions**
- **What We Built**
- **Variations**