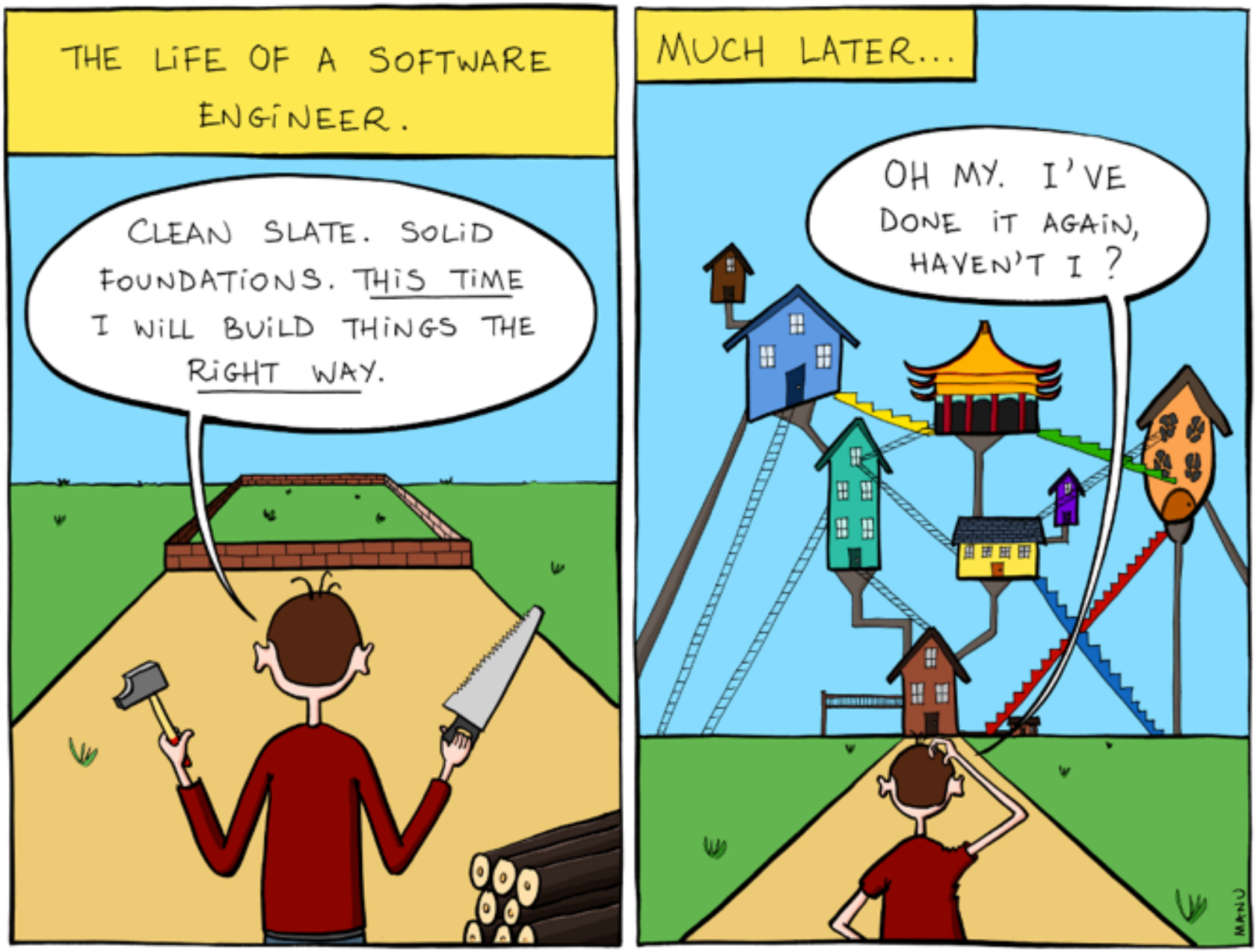
Advanced Programming
(with python)

David Beazley
@dabeaz
http://www.dabeaz.com



http://www.bonkersworld.net

What is Programming?

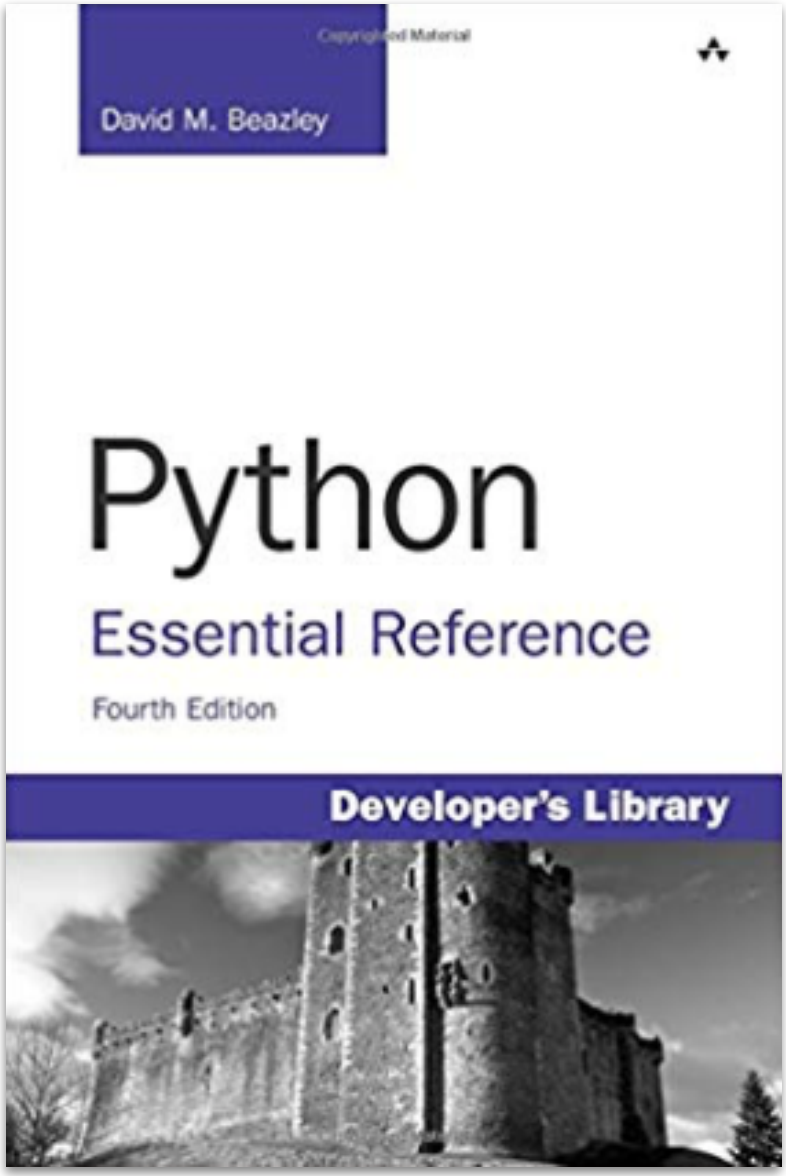
- Problem solving
- Software design/architecture
- Composition of components
- Abstraction
- Decision making
- ???? Probably many other things...

This Course

- It's mostly about programming
- We'll be using Python
- And to be sure, we'll say a lot about Python
- Maybe more than wanted about Python
- But the overall principles are more general

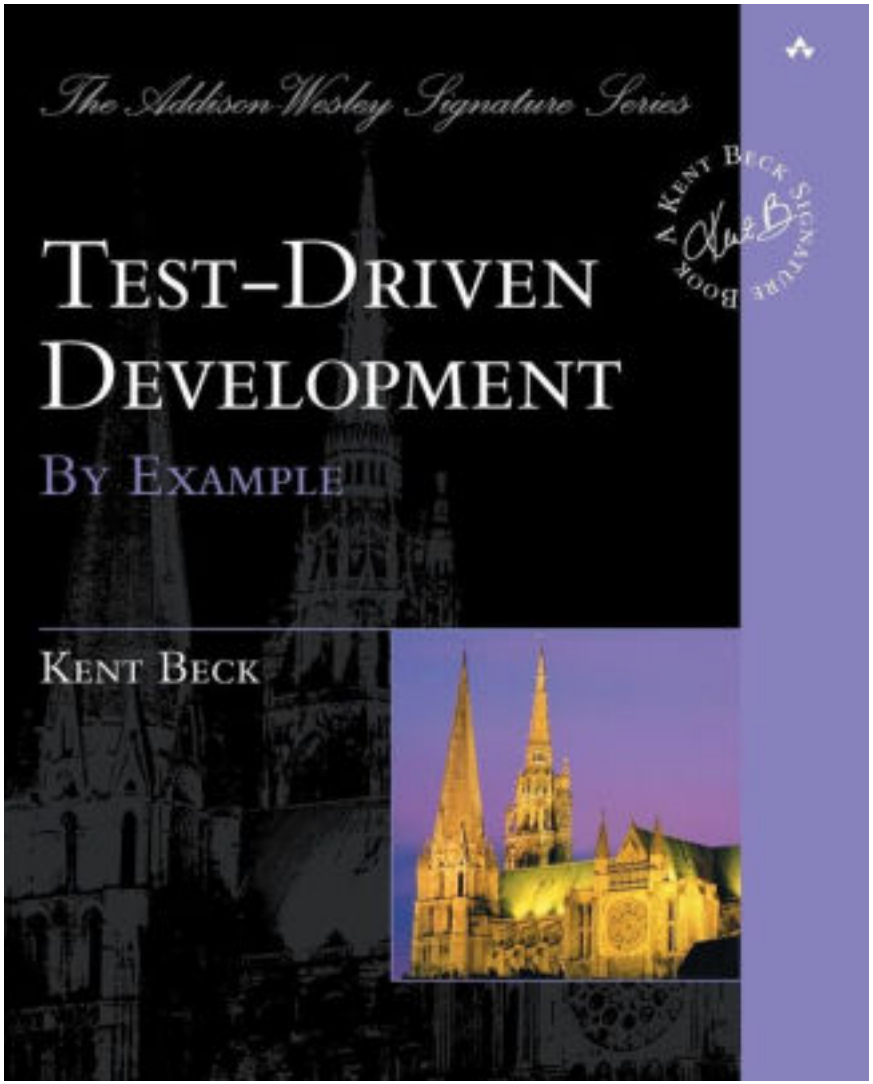
Not a Python Reference

- Not focused on "features"
- Not "Advanced Python"
- Not about the library
- Or third-party modules
- Or project organization



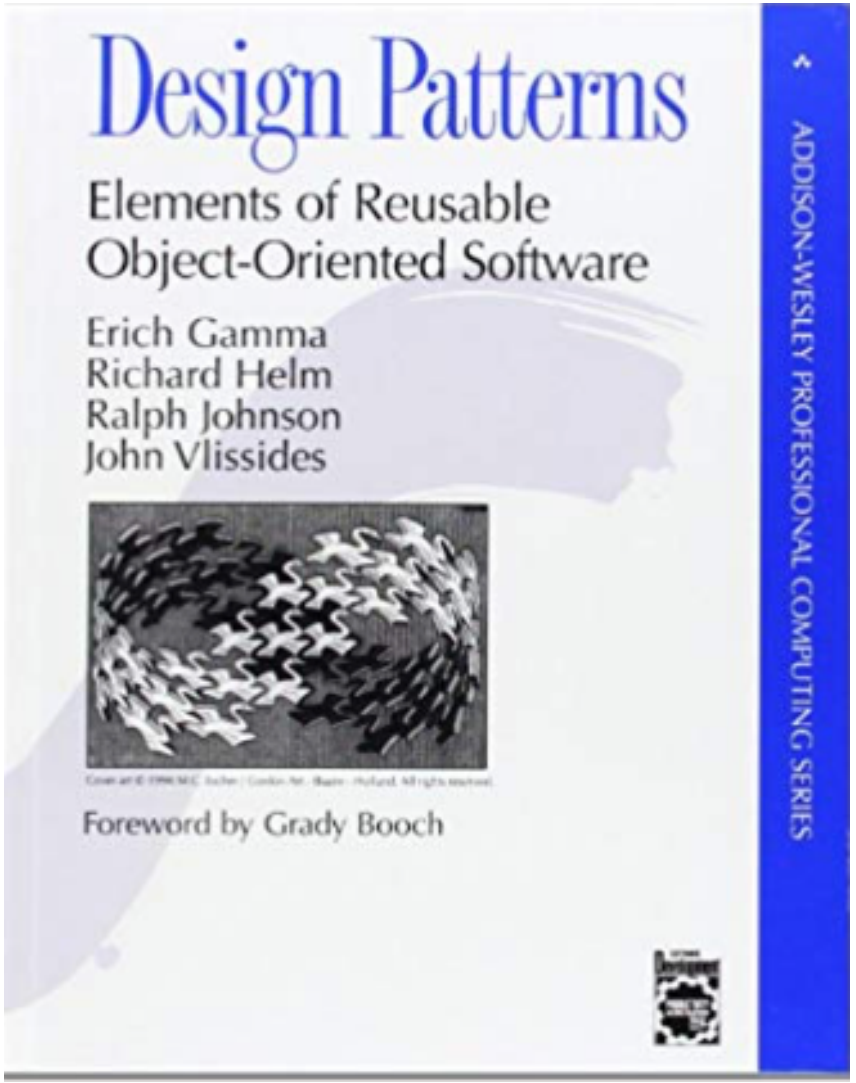
Not Software Process

- Not about TDD
- Or Agile
- Or Waterfall
- Or project management...



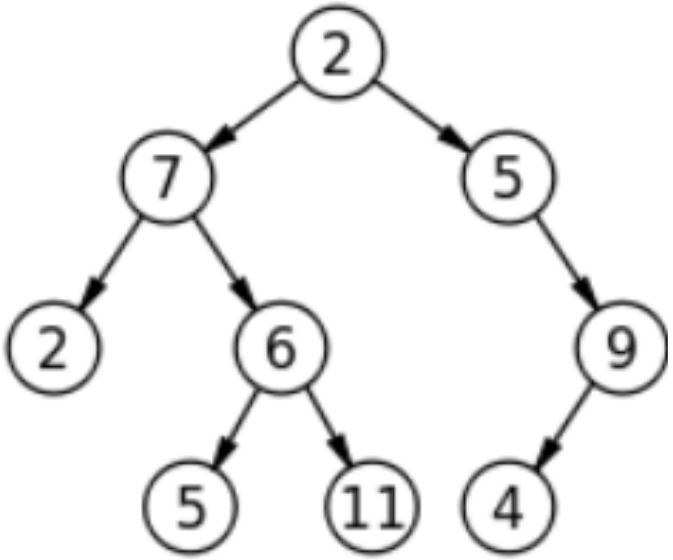
Not Recipes

- This is a fine book...
- But most Python devs don't think like this
- Very C++/Java focused



Not Algorithms

- When was the last time you had to implement a classic "algorithm"?
- You'd use a library
- Maybe multiple libraries
- And glue parts together



Overview

- This is a course about the "edges"
- Breaking problems down into parts
- Thinking about how the parts fit together
- Thinking about design
- Thinking about complexity (avoiding it)

Format

- Entire course is organized around coding
- There are no formal lecture slides
- There are no given "solutions"
- There is often no "right" answer
- Driven by group discussion
- You will get more out of the course by participation (questions, discussion, etc.)

Core Topics

- Data Abstraction
- Object-Oriented Programming
- Functional Programming
- Concurrency
- Linguistic abstraction
- Generally: Better idea of how Python works in relationship to the above topics

Recurring Theme

- Composition
- Much of what we do as programmers is we compose parts together (functions, objects, modules, packages, etc.)
- Throughout this course, we'll ask a lot of questions about this topic in particular.
- Exploring consequences of decisions.

Let's Begin