Software Architecture Tool Belt: Design Patterns and Code Smells

Nico Rehwaldt @nikku

Context: Refactoring

We continuously ensure that our software stays helthy, do we?

Context: Build for Understandability

We spend most of our time reading (and understanding) software, not writing it.

We better build towards understandability, too.

Context: Why do humans build software?

- Building software is hard (still requires humans to do so)
- There is no silver bullet, software can work in so many ways
- **Goal: Aim for the good enough and keep it**

Levels of Understanding

- line of code
- code unit
- cross component
- systems

Design Pattern: What is it?

> In software engineering, a software design pattern is a general, reusable solution to a commonly occurring problem within a given context in software design.
>
> – Wikipedia

Design Pattern: What is it?

Defines roles and responsibilities in cross component communication pattern as well as the result being achieved / problem being solved.

Design Pattern: Why should I care?

A common knowledge and visibility of design patterns in the code base promotes understandability.

Example: Say *we use the observer pattern here* and the recipient immediately understands the problem being solved.

Design Pattern: Resources

- Refactoring Guru
- Wikipedia: Software design pattern
- Book: Gang of Four - Elements of Reusable Object-Oriented Software

Code Smell: What is it?

> **In computer programming, a code smell is any characteristic in the source code of a program that possibly indicates a deeper problem.** Determining what is and is not a code smell is subjective, and varies by language, developer, and development methodology.
>
> – Wikipedia

Code Smell: Why should I care?

Gives your feeling *Something smells like 💩 here* a name.

You still need to evaluate the impact and deal with it yourself.

Code Smell: Resources

- Anti Pattern
- Martin Fowler on Code Smells
- How to write unmaintainable code

Thanks