

## Refactoring, what, when,how?

- Programs are hard to get right the first time
  - Similar to an essay paper? Rewrite?
  - Code isn't fast enough, but it works ...
  - Code isn't general enough, but it solves the problem ...
  - Code passes initial tests, then becomes problematic ...
- Refactoring: change how the program works, not what it does
  - If we have a suite of tests, we ensure that they stay passed
  - Change the code, retest, change the code, retest
- Only works if we have testable code, otherwise ...?

## When to refactor: code smells

Highly experienced and knowledgeable developers have a "feel" for good design. Having reached a state of "Unconscious Competence," where they routinely practice good design without thinking about it too much, they find that they can look at a design or the code and immediately get a "feel" for its quality, without getting bogged down in extensive "logically detailed arguments".

<http://c2.com/cgi/wiki?CodeSmell>

- Note that a CodeSmell is a *hint* that something *might* be wrong, not a certainty. A perfectly good idiom may be considered a CodeSmell because it's often misused, or because there's a simpler alternative that works in most cases. Calling something a CodeSmell is not an attack; it's simply a sign that a closer look is warranted. so CodeSmell is more instinctive than intuitive? [spelling corrected by ola, see c2.com link]

## Things that merit sniffing: smells

- Duplicate code: once ok, twice ok, three times: refactor
  - Why is duplicate code a potential problem?
- Method is too long (what's too long?)
  - How do we fix, why a problem?
- Classes with too much code
  - Each class should have one responsibility, *cohesiveness*
- Code that's commented out, not used
  - Why is the code there? Why is keeping it a problem?

## Martin Fowler

- Guru of refactoring
  - My primary areas of involvement are in object-oriented development, refactoring, patterns, agile methods, enterprise application architecture, domain modeling, and extreme programming.
- I see refactoring as a very specific technique to do the more general activity of restructuring. Restructuring is any rearrangement of parts of a whole. It's a very general term that doesn't imply any particular way of doing the restructuring.

