Good Code

Perdita Stevens

School of Informatics
University of Edinburgh

Plan

- What does it mean for code to be good?
- Why is it important?
- Why isn't it enough?

Context (personal, you)

I said that you are assumed to be competent at programming in Java.

i.e. you can write code.

Context (personal, you)

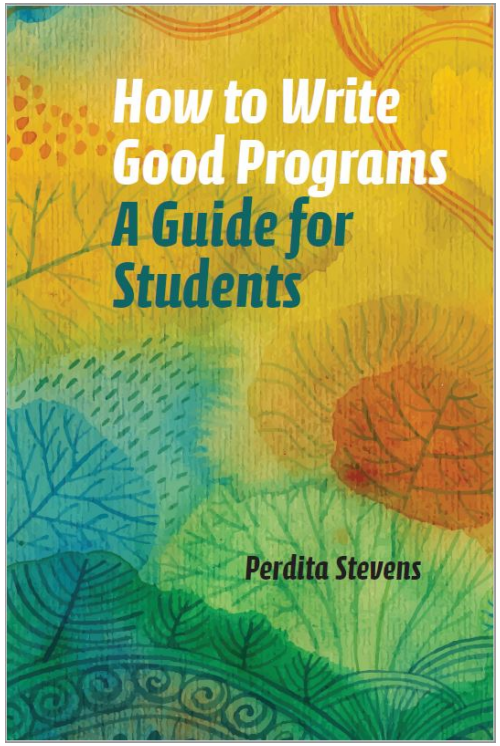
I said that you are assumed to be competent at programming in Java.

i.e. you can write code.

How do you know when you've written good code?

Context (personal, me)

The topic of good code, and how people learn to write it, has been on my mind a lot lately... my book



came out last year

Context (discipline)

Coding
Programming
Software engineering

... overlaps? differences? connotations?

Context (discipline)

Coding
Programming
Software engineering

... overlaps? differences? connotations?

increasing responsibility; increasing awareness of big picture

Good programs

1. do what they are supposed to do (are *correct*)
2. are clearly written

But what do these mean, and why are they important? How can we tell? How can we ensure them? Are other qualities important?

Correctness

well, if the code doesn't do what it's supposed to, it may be useless (or worse).

- $64K question: correct with respect to what?
- specification...
- contracts (more later)
- what about users' actual needs?

  Verification: have we built the software right?
  Validation: have we built the right software?

Key difficulty: any other artefact, with respect to which the code is supposed to be correct, may itself be "incorrect".

Agile as a "back to basics" movement

Coding may be a small part of what a typical software engineer spends their time doing...

Agile as a "back to basics" movement

Coding may be a small part of what a typical software engineer spends their time doing...

... but it's the code that actually runs in the end, so it has to be good

Agile as a "back to basics" movement

Coding may be a small part of what a typical software engineer spends their time doing...

... but it's the code that actually runs in the end, so it has to be good

... and maybe software engineers ought to spend of their more time coding?

Agile as a "back to basics" movement

Coding may be a small part of what a typical software engineer spends their time doing...

... but it's the code that actually runs in the end, so it has to be good

... and maybe software engineers ought to spend of their more time coding?

More on agile later, but for now, how does agile handle correctness?

Tests as specification

- tests as specification
- Test-Driven Development (TDD)

Tests for all production code, always up to date, is a part of "agile" that tends to get left out. It's hard.

Requirements still change. When they do, tests and production code have to change.

Tests as specification

- tests as specification
- Test-Driven Development (TDD)

Tests for all production code, always up to date, is a part of "agile" that tends to get left out. It's hard.

Requirements still change. When they do, tests and production code have to change.

Key challenge: make change cheaper.

Tests as specification

- tests as specification
- Test-Driven Development (TDD)

Tests for all production code, always up to date, is a part of "agile" that tends to get left out. It's hard.

Requirements still change. When they do, tests and production code have to change.

Key challenge: make change cheaper.

Key element: clearly written code.

Clearly-written code

Often seen – by the inexperienced! – as unimportant.

But the code–human interface is as important as the code–computer one, because of this need for change.

Clearly-written = Easy to change when necessary

- layout
- conventions, e.g. capitalisation
- consistency, in general, e.g. about how errors are handled
- names
- structure

Local improvements

You can improve code a lot by looking at just what's in front of you and optimising that...

... but sometimes you need a more bird's eye view. E.g. if you want to rename a class, what captures what it means to every client?

Sometimes high-level conventions and frameworks help.

Sometimes you need a way to get high-level structure into a human's head, and that means...

Local improvements

You can improve code a lot by looking at just what's in front of you and optimising that...

... but sometimes you need a more bird's eye view. E.g. if you want to rename a class, what captures what it means to every client?

Sometimes high-level conventions and frameworks help.

Sometimes you need a way to get high-level structure into a human's head, and that means...

MODELLING