Testing

Objectives of a software project

Build the right product.
Build the product right.
Build it as quickly and cheaply as possible.

Software Methodology

HOW TO:

Build the right product.
Build the product right.
Build it as quickly and cheaply as possible.

Wicked problem, no silver bullet, waffle principle, etc.

Practices, Principles, Patterns

Software Methodology

WHEN AND HOW TO ANSWER
- Are we building the right project?
- Are we building the project right?

Life Cycle Models

When

waterfall → iterative → agile

Test at the end. Test early and often. Test first.

How

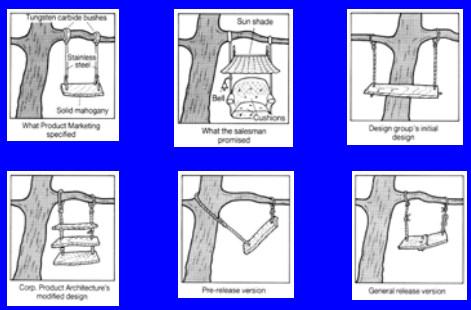
Are we building the right product?

Are we building the product right?

How

Are we building the right product?

Are we building the product right?



How

Are we building the right product?

Customer evaluation
Use case realizations
Acceptance testing

How

Are we building the right product?

Are we building the product right?

Rephrased

- Design: Is the design maintainable and extensible?
- Implementation: Is the code clean and correct?

Bugs

- Avoid
- Detect & repair

Bugs

What is the
the big deal about bugs?

Abstraction

our talent for abstraction comes part and parcel with our talent for overlooking error

slashdot

Aoccdrnig to a rscheearch at Cmabrigde Uinervtisy, it deosn't mttaer in waht oredr the ltteers in a wrod are, the olny iprmoetnt tihng is taht the frist and lsat ltteer be at the rghit pclae.

The rset can be a total mses and you can sitll raed it wouthit porbelm.

Tihs is bcuseae the huamn mnid deos not raed ervey lteter by istlef, but the wrod as a wlohe.

Amzanig

bugs

```
for (i=0; i<=n, i++);

while (i=0) i++;

double d = i/20;
```

bugs cont.

- failure to test input
- memory leaks
- weird c++ shenanigans

bugs cont.

- code != intent
- intent wrong

exercise 1

- In groups of 2 discuss
  - A really bad bug you had in a CS70 project
  - How it manifested itself
  - What you did to try to find it
  - What finally worked
- In ten minutes your group will present one case study to the class.

exercise 2

- brainstorm with your partner to come up with a list (as long and creative as possible) of ways to track down bugs
- in 15 minutes you'll present your results to the class

exercise 3

- with your partner, create a list of things to try the next time you have a bug
  - the list should be ordered (as well as you can) with most promising approaches listed first
  - indicate how long you should stick to one approach before going on
- in 10 minutes you'll present your results