

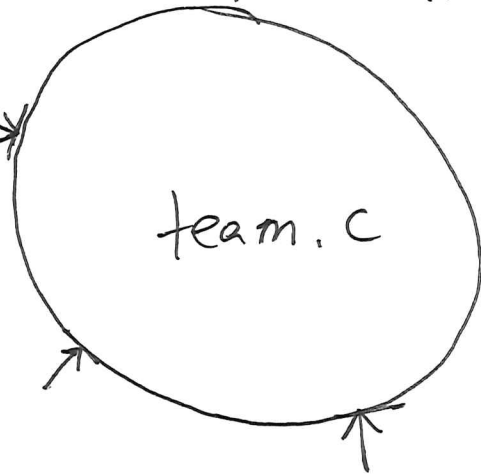
Testing: Confirm if code meets spec.

1) unit testing - Try a little functionality at a time. Verify everything.

2) interactive
↳ manual testing

3) assertions
ie. assert(...)

Implementation



assert

Only things that must always be true

- Not external args
- Not system status
- Not file contents

public interface is the contract of what code should do.

- documentation

- header file(s)

(.h files)

→ aq.gs/264/hw10

→ team.h

"candy machine interfaces"

How to think of test cases

- Edge cases
 - extremes / boundaries
 - 0, ~~1~~ many
- Corner case
 - special
- Functionality
 - Main Cases
 - all functions
 - Special case
 - weird cases
 - Nits (eg. output format, etc.)
- print-integer(0, ...)
 - team of odd or even size
 - first person's favorite is second person

Testing Hw 10

team size

{
0,
1,
many odd
many even
}

x

functions

{
fprint-team
add-person
:
:
:
}

x

corner/special/main case

{
person[i].fav
= person[i+k]
person[i].fav
= person[i-k]
}

team name
size

x

{
1,
many
}

x

.....