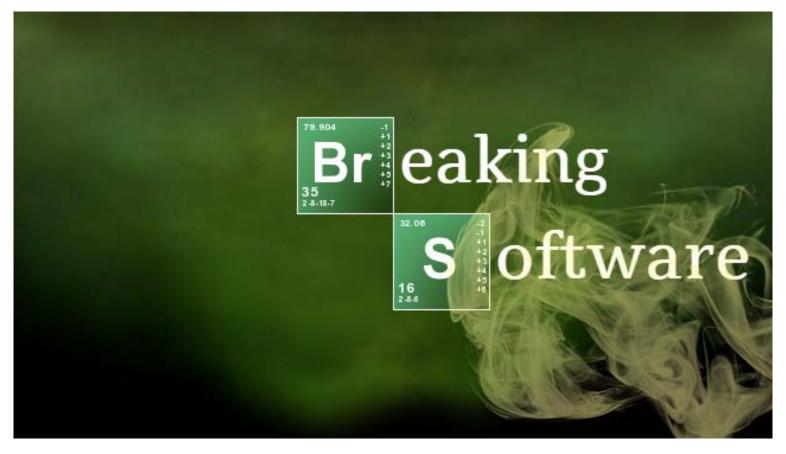
CS1632, Lecture 6



Wonsun ahn

Software tends not to break much on the "happy path"

- Happy Path: a case where user inputs valid, usual values; expected usage
- It breaks on the unexpected cases.
 - Corner cases.
 - Systems going down.
 - Malicious users.
 - When you're off in the wilderness.

Logic Errors: The logic of the program is incorrect

- Requirement: Driving while drunk shall get a student a DUI.
- Code:

```
if (student.isDrunk() || student.isDriving()) {
    student.setDUI(true);
} else {
    student.setDUI(false);
}
```

OFF-by-one error: a subset of logic errors where values are specified incorrectly by one unit

- Requirement: The minimum drinking age for student shall be 21.
- Code:

```
if (student.getAge() > 21) {
    student.setCanDrink(true);
} else {
    student.setCanDrink(false);
}
```

Rounding/floating point errors: rounding or floating point display give incorrect results.

```
double oneVal = 1.0 / 857.0;
double total = oneVal * 857.0;

System.out.println("Should be 1.0, actually = " + total);
boolean areEqual = (total == 1.0);
System.out.println("Are equal? " + areEqual);
```

Integration errors: Errors at boundaries between systems/subsystems.

```
int startDistanceInKilometers = 14;
spacecraft.setDistance(startDistanceInKilometers);
public class Spacecraft
   public void setDistance(int distanceInMiles) {
```

Errors of assumption:

developer or system makes an assumption which turns out to be incorrect, or at odds with other assumptions.

```
OutputFile.write(TAB_DELIMITED);
...
InputFile.read(COMMA DELIMITED);
```

MISSING DATA ERRORS:

An error occurs because needed data is missing and the system cannot operate properly without it.

```
public static void main(String[] args) {
    System.out.println(args[3]);
}
```

BAD DATA ERRORS: System cannot handle improperly formatted or invalid data.

```
Enter two numbers to divide: 7 0
Exception in thread "main"
java.lang.ArithmeticException: / by zero
```

DISPLAY ERRORS: The data is correct but not displayed properly.

```
double pi = Math.PI;
System.out.printf("Pi is equal to %.1f", pi);
```

Null pointer error: The program dereferences a null pointer.

```
String oneILove = null;
oneILove = oneILove.toUpperCase();

System.out.printf("This one goes out to
the one I love," + oneILove);
```

I/O Errors:

The system encounters an unexpected state of disk, network, or other I/O and cannot handle it.

```
try {
    // read in file
} catch (FileNotFoundException e) {
    // AAARGH WHAT DO I DO
    System.exit(1);
}
```

Configuration error:

The system could work correctly, but it was not configured to work correctly.

'javac' is not recognized as an internal or external command, operable program or batch file.

The list goes on...

- Data type errors
 - Error arising due to incorrect implicit / explicit data type conversion
- Permission errors
 - No permission to access a required resource (file, database table, etc.)
- Version mismatch errors
 - Library version not the same version software was intended to be used with
- Distributed system errors
 - Error while communicating between different parts of distributed system
 - E.g. Error in data marshalling / demarshalling between client / server
- Interface errors
 - Error arising from developer misunderstanding behavior of an API

Now Please Read Textbook Chapter 7