COM 3529
Software Testing & Analysis

Professor Phil McMinn

1.1 Introduction

Beizer’s Maturity Model

0. There’s no difference between testing and debugging

1. The purpose of software testing is to show that software works

2. The purpose of software testing is to show that software doesn't work

“Excellent testing can make you unpopular with almost everyone!”

Beizer’s Maturity Model

0. There’s no difference between testing and debugging

1. The purpose of software testing is to show that software works

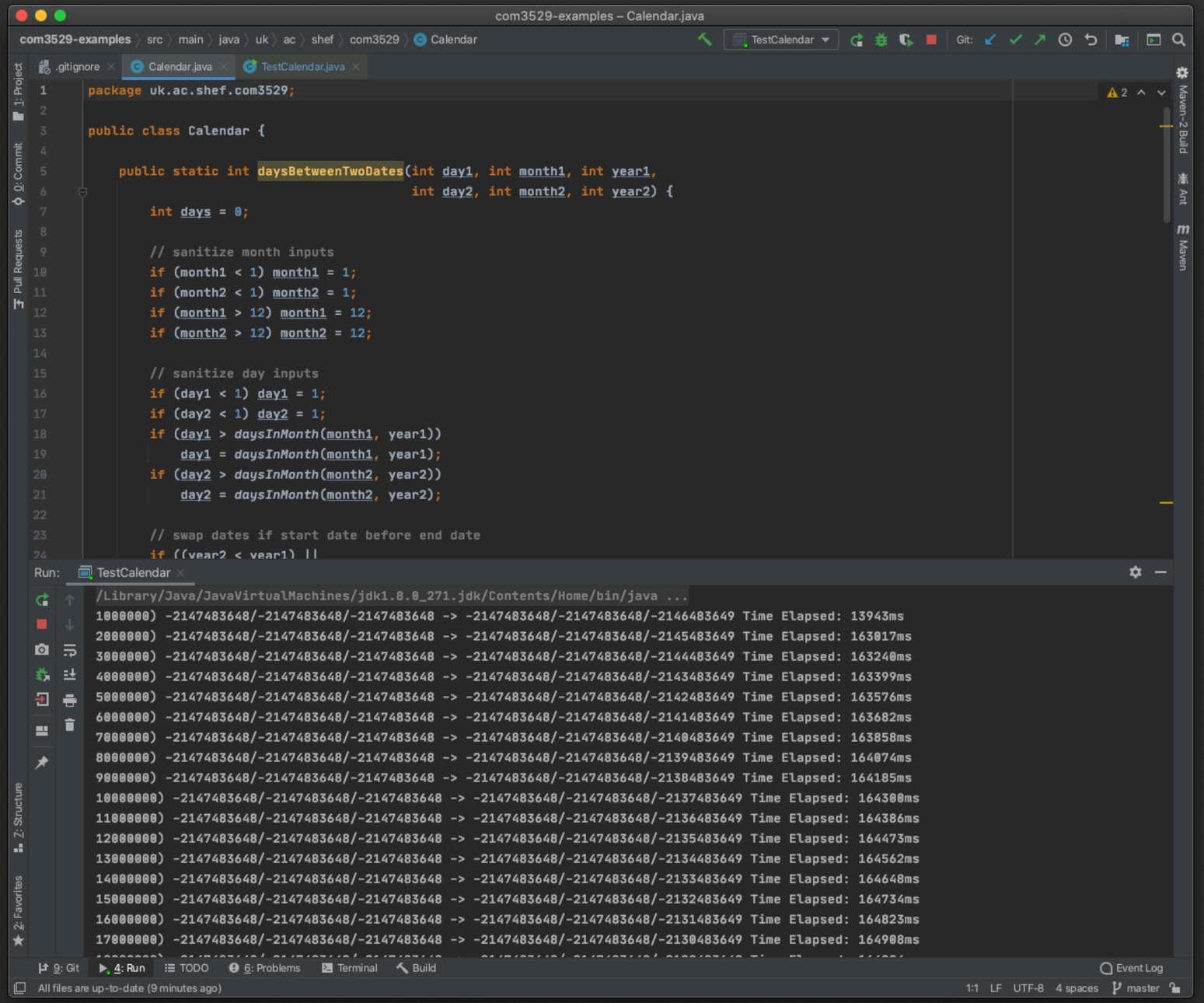
2. The purpose of software testing is to show that software doesn't work

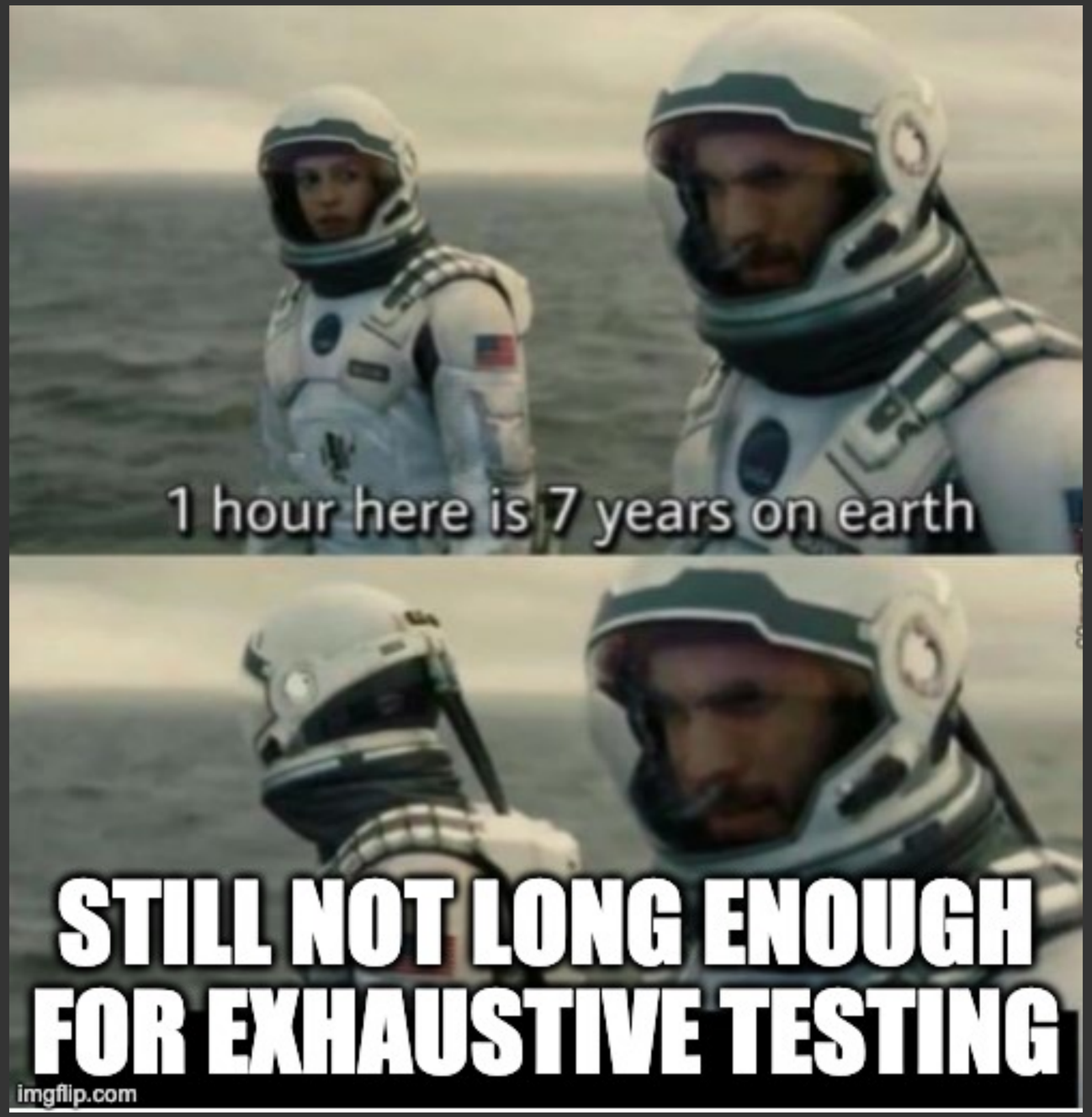
3. The purpose of software testing is not to show anything in particular, but to reduce the risk of using software

4. Testing is a mental discipline that helps all IT professionals develop higher quality software

Why Software Testing is Hard

	Tractable problems (*P*)	**Intractable problems (*NP*)**	**Uncomputable problems**
Description	Can be solved efficiently	Method for solving exists, but is hopelessly time consuming	Cannot be solved by any computer program
Computable in theory	😎	😎	😭
Computable in practice	😎	🤔	😭
Example	Find the shortest route on a map	Decryption	*Finding all bugs in computer programs*


com3529-examples – Calendar.java
com3529-examples 〉 src 〉 main 〉 java 〉 uk 〉 ac 〉 shef 〉 com3529 〉 Calendar
TestCalendar
Git:
.gitignore
Calendar.java
TestCalendar.java
1: Project
0: Commit
Pull Requests
Maven-2 Build
Ant
Maven
2 ^ v
1 package uk.ac.shef.com3529;
2
3 public class Calendar {
4
5 public static int daysBetweenTwoDates(int day1, int month1, int year1,
6 int day2, int month2, int year2) {
7 int days = 0;
8
9 // sanitize month inputs
10 if (month1 < 1) month1 = 1;
11 if (month2 < 1) month2 = 1;
12 if (month1 > 12) month1 = 12;
13 if (month2 > 12) month2 = 12;
14
15 // sanitize day inputs
16 if (day1 < 1) day1 = 1;
17 if (day2 < 1) day2 = 1;
18 if (day1 > daysInMonth(month1, year1))
19 day1 = daysInMonth(month1, year1);
20 if (day2 > daysInMonth(month2, year2))
21 day2 = daysInMonth(month2, year2);
22
23 // swap dates if start date before end date
24 if ((year2 < year1) ||
Run:
TestCalendar
/Library/Java/JavaVirtualMachines/jdk1.8.0_271.jdk/Contents/Home/bin/java ...
1000000) -2147483648/-2147483648/-2147483648 -> -2147483648/-2147483648/-2146483649 Time Elapsed: 13943ms
2000000) -2147483648/-2147483648/-2147483648 -> -2147483648/-2147483648/-2145483649 Time Elapsed: 163017ms
3000000) -2147483648/-2147483648/-2147483648 -> -2147483648/-2147483648/-2144483649 Time Elapsed: 163240ms
4000000) -2147483648/-2147483648/-2147483648 -> -2147483648/-2147483648/-2143483649 Time Elapsed: 163399ms
5000000) -2147483648/-2147483648/-2147483648 -> -2147483648/-2147483648/-2142483649 Time Elapsed: 163576ms
6000000) -2147483648/-2147483648/-2147483648 -> -2147483648/-2147483648/-2141483649 Time Elapsed: 163682ms
7000000) -2147483648/-2147483648/-2147483648 -> -2147483648/-2147483648/-2140483649 Time Elapsed: 163858ms
8000000) -2147483648/-2147483648/-2147483648 -> -2147483648/-2147483648/-2139483649 Time Elapsed: 164074ms
9000000) -2147483648/-2147483648/-2147483648 -> -2147483648/-2147483648/-2138483649 Time Elapsed: 164185ms
10000000) -2147483648/-2147483648/-2147483648 -> -2147483648/-2147483648/-2137483649 Time Elapsed: 164300ms
11000000) -2147483648/-2147483648/-2147483648 -> -2147483648/-2147483648/-2136483649 Time Elapsed: 164386ms
12000000) -2147483648/-2147483648/-2147483648 -> -2147483648/-2147483648/-2135483649 Time Elapsed: 164473ms
13000000) -2147483648/-2147483648/-2147483648 -> -2147483648/-2147483648/-2134483649 Time Elapsed: 164562ms
14000000) -2147483648/-2147483648/-2147483648 -> -2147483648/-2147483648/-2133483649 Time Elapsed: 164648ms
15000000) -2147483648/-2147483648/-2147483648 -> -2147483648/-2147483648/-2132483649 Time Elapsed: 164734ms
16000000) -2147483648/-2147483648/-2147483648 -> -2147483648/-2147483648/-2131483649 Time Elapsed: 164823ms
17000000) -2147483648/-2147483648/-2147483648 -> -2147483648/-2147483648/-2130483649 Time Elapsed: 164908ms
2: Favorites
Z: Structure
9: Git
4: Run
TODO
6: Problems
Terminal
Build
Event Log
All files are up-to-date (9 minutes ago)
1:1 LF UTF-8 4 spaces master


1 hour here is 7 years on earth
STILL NOT LONG ENOUGH FOR EXHAUSTIVE TESTING
imgflip.com

> Software testing can only show the presence, not the absence of bugs

The Oracle Problem

The First Bug

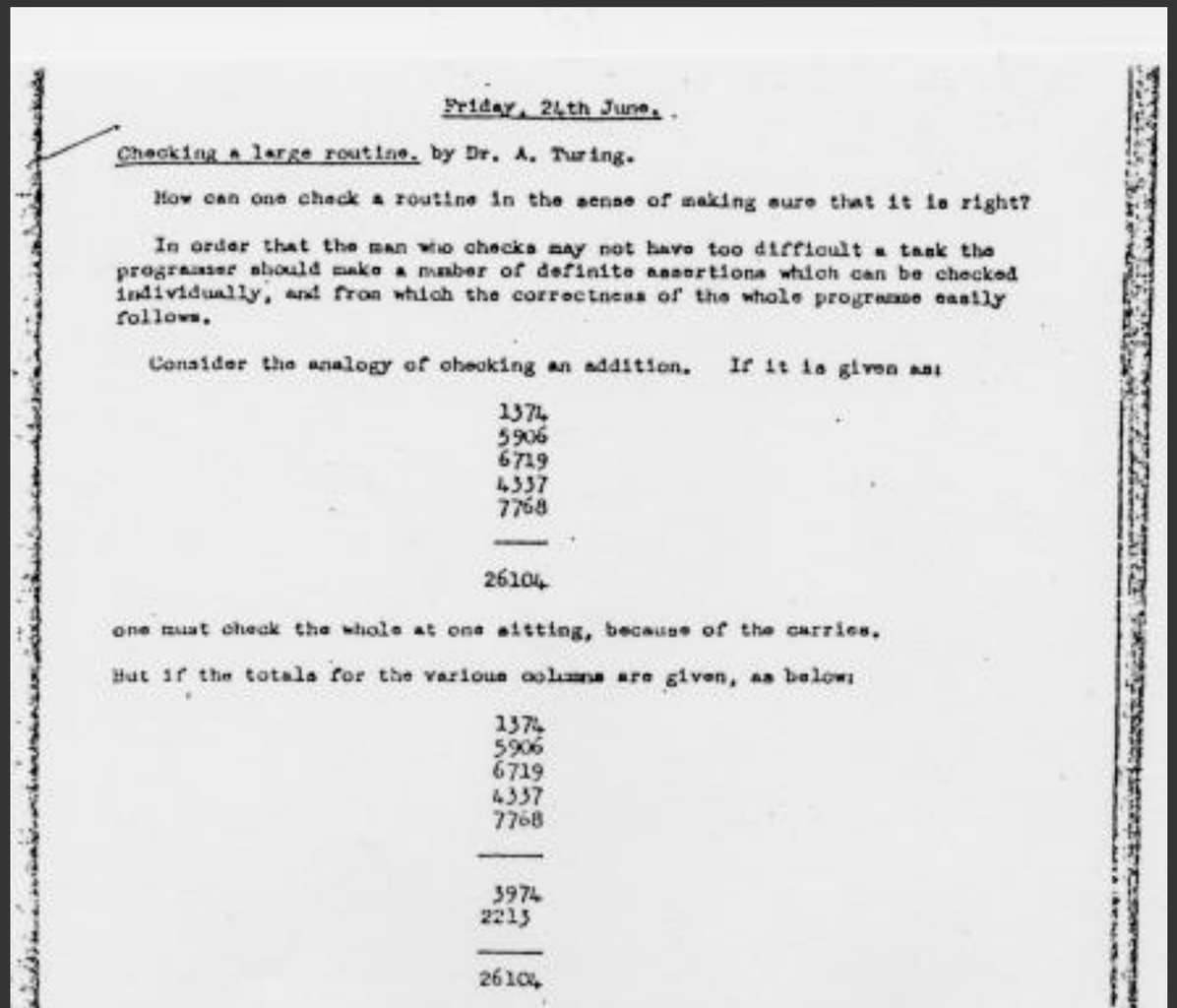
Photo # NH 96566-KN (Color) First Computer "Bug", 1947

9/9

0800 Arctan started
1000 " stopped - arctan ✓ { 1.2700 9.037 847 025
9.037 846 ?95 correct
13" vc (032) MP - MC ~~2.130476415 (-3)~~ 4.615925059(-2)
(033) PRO 2 2.130476415
correct 2.130676415
Relays 6-2 in 033 failed special speed test
In relay " " 11.00 test.
Relays changed
1100 Started Cosine Tape (Sine check)
1525 Started Mult + Adder Test.
1545  Relay #70 Panel F
(moth) in relay.
First actual case of bug being found.
~~1700~~ 1630 Arctangent started.
1700 closed down.

The First Research Paper on Software Testing

“Checking a large routine” by A. Turing



Friday, 24th June.

Checking a large routine. by Dr. A. Turing.

How can one check a routine in the sense of making sure that it is right?

In order that the man who checks may not have too difficult a task the programmer should make a number of definite assertions which can be checked individually, and from which the correctness of the whole programme easily follows.

Consider the analogy of checking an addition. If it is given as:

1374
5906
6719
4337
7768

26104

one must check the whole at one sitting, because of the carries.

But if the totals for the various columns are given, as below:

1374
5906
6719
4337
7768

3974
2213

26104

What You Will Get From This Module

In the first half of this module, you will gain:

1. A deeper understanding of how failures happen, and the difference between faults and failures (Week 1)

2. An understanding of systematic approaches to Software Testing, underpinned by coverage criteria (Weeks 2–4)

3. Insights into some of the latest automated techniques for test case generation, with hands-on practical experience (Week 5)

Examples Repository

https://github.com/philmcminn/com3529-code