# Approval Testing Introduction

Emily Bache
@emilybache

PAGERO

Bache
Consulting

# Approval Testing

A different paradigm for Assertion

# Parts of a Test Case

```
15    @Test
16    public void getSpeedOfAfricanParrot_With_One_Coconut() {
17        Parrot parrot = new Parrot(ParrotTypeEnum.AFRICAN, 1, 0, false);
18        assertEquals(parrot.getSpeed(), 3.0);
19    }
```

16 - Test Case Name
17 - Arrange
18 - Act
18 - Assert

# Potential problems?

```
15    @Test
16    public void getSpeedOfAfricanParrot_With_One_Coconut() {
17        Parrot parrot = new Parrot(ParrotTypeEnum.AFRICAN, 1, 0, false);
18        assertEquals(parrot.getSpeed(), 3.0);
19    }
```

- We only check one aspect of the parrot: the speed.

- If the speed changes to 3.1 the test will fail, and we have to change code to fix it.

# Another way to test this?

```
32
33          @Test
34   ⊕▶     public void getSpeedNorwegianBlueParrot_nailed() {
35              Parrot parrot = new Parrot(ParrotTypeEnum.NORWEGIAN_BLUE, 0, 0, true);
36              assertEquals(parrot.toString(), "Parrot(type=NORWEGIAN_BLUE, speed=2.0)");
37          }
```

7 tests done: 1 failed — 33ms

```
/usr/lib/jvm/java-8-oracle/bin/java ...

junit.framework.ComparisonFailure: null
Expected :Parrot(type=NORWEGIAN_BLUE, speed=0.0)
Actual   :Parrot(type=NORWEGIAN_BLUE, speed=2.0)
  <Click to see difference>




    at junit.framework.Assert.assertEquals(Assert.java:81)
    at junit.framework.Assert.assertEquals(Assert.java:87)
    at parrot.ParrotTest.getSpeedNorwegianBlueParrot_nailed(ParrotTest.java:36) <27 internal calls>
```

# Why don't we do this?

```
32
33        @Test
34   ⬤▶  public void getSpeedNorwegianBlueParrot_nailed() {
35            Parrot parrot = new Parrot(ParrotTypeEnum.NORWEGIAN_BLUE, 0, 0, true);
36            assertEquals(parrot.toString(), "Parrot(type=NORWEGIAN_BLUE, speed=2.0)");
37        }
```

- Test isolation - many tests with similar strings

- Maintenance costs - updating comparison strings

- A lot of 'toString' methods needed

- TDD - in practice you don't write the assertions up front any more - copy and paste from actual output

# Minesweeper
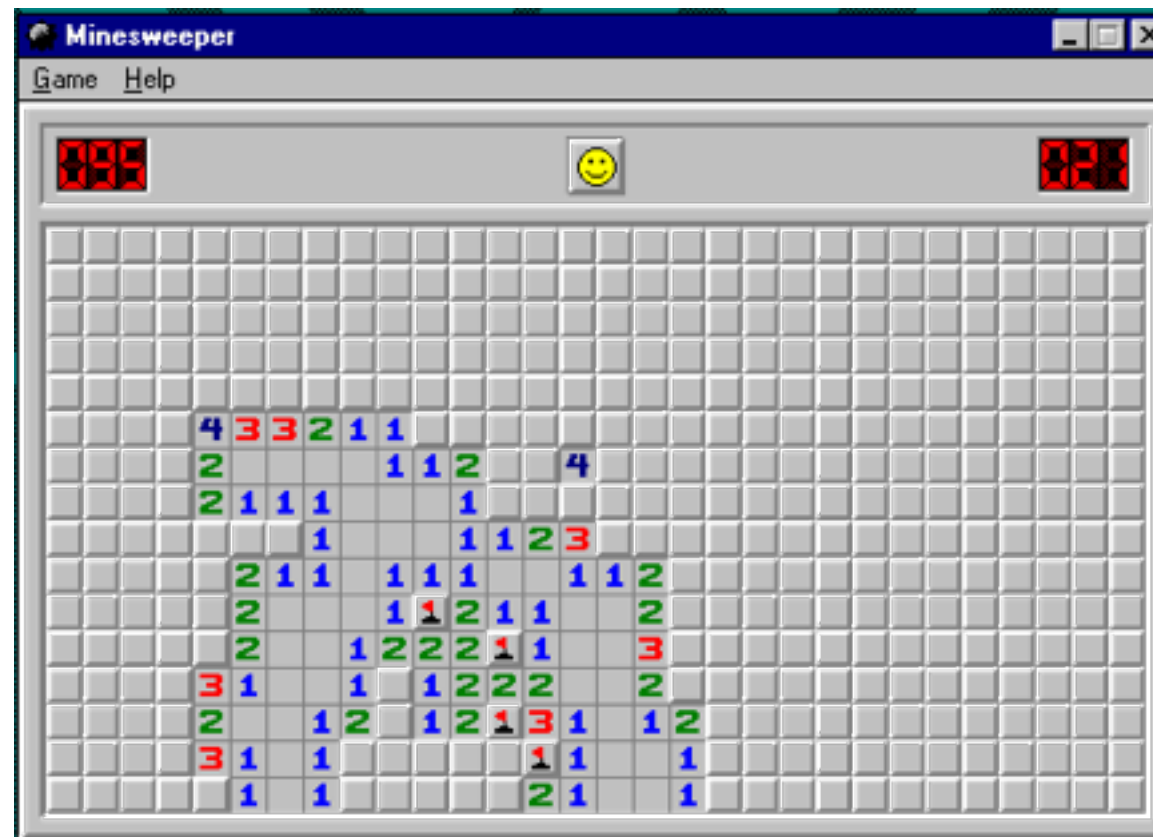
# Minesweeper as a
# Command Line Program



Minesweeper
program
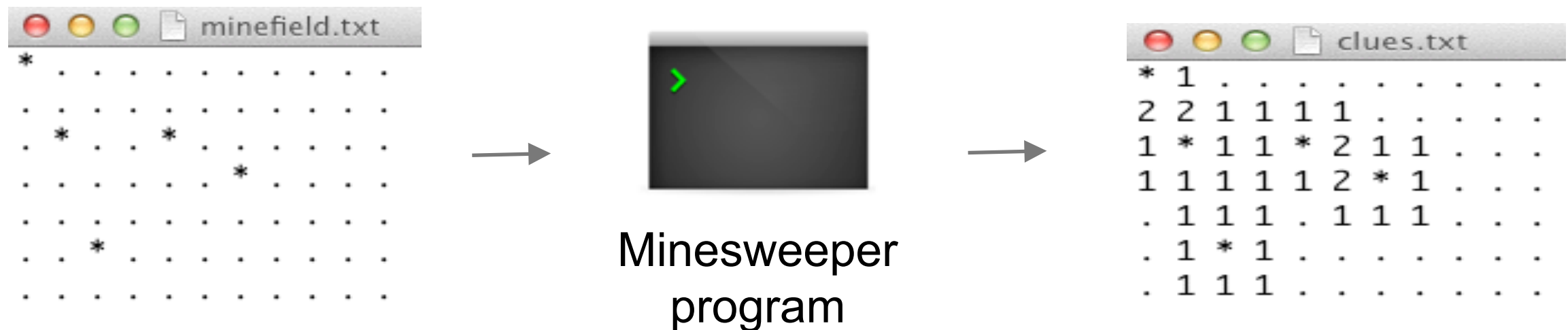
# How to check for correct behaviour?



Minesweeper
program
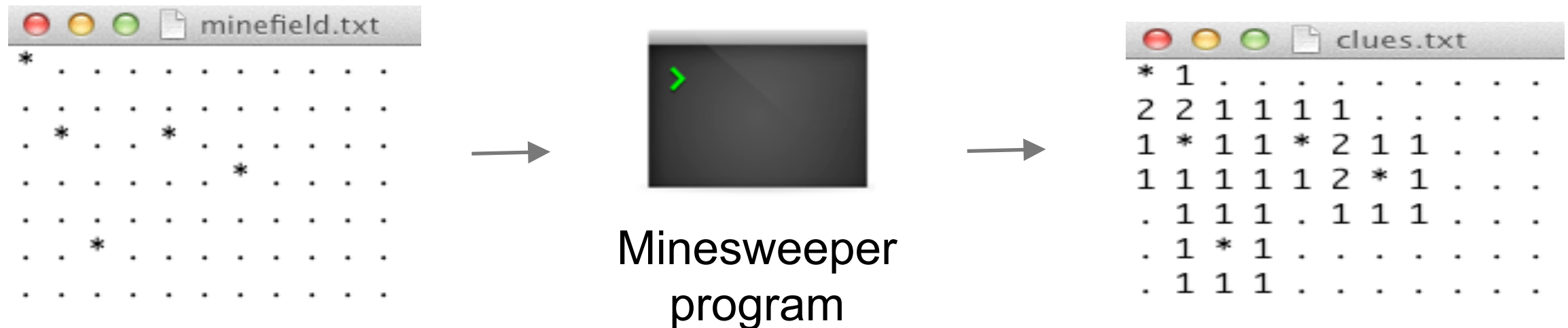
```
assertEquals(2, clueAt(0,1))
assertEquals(1, clueAt(1,0))
...
```

# Approval Testing
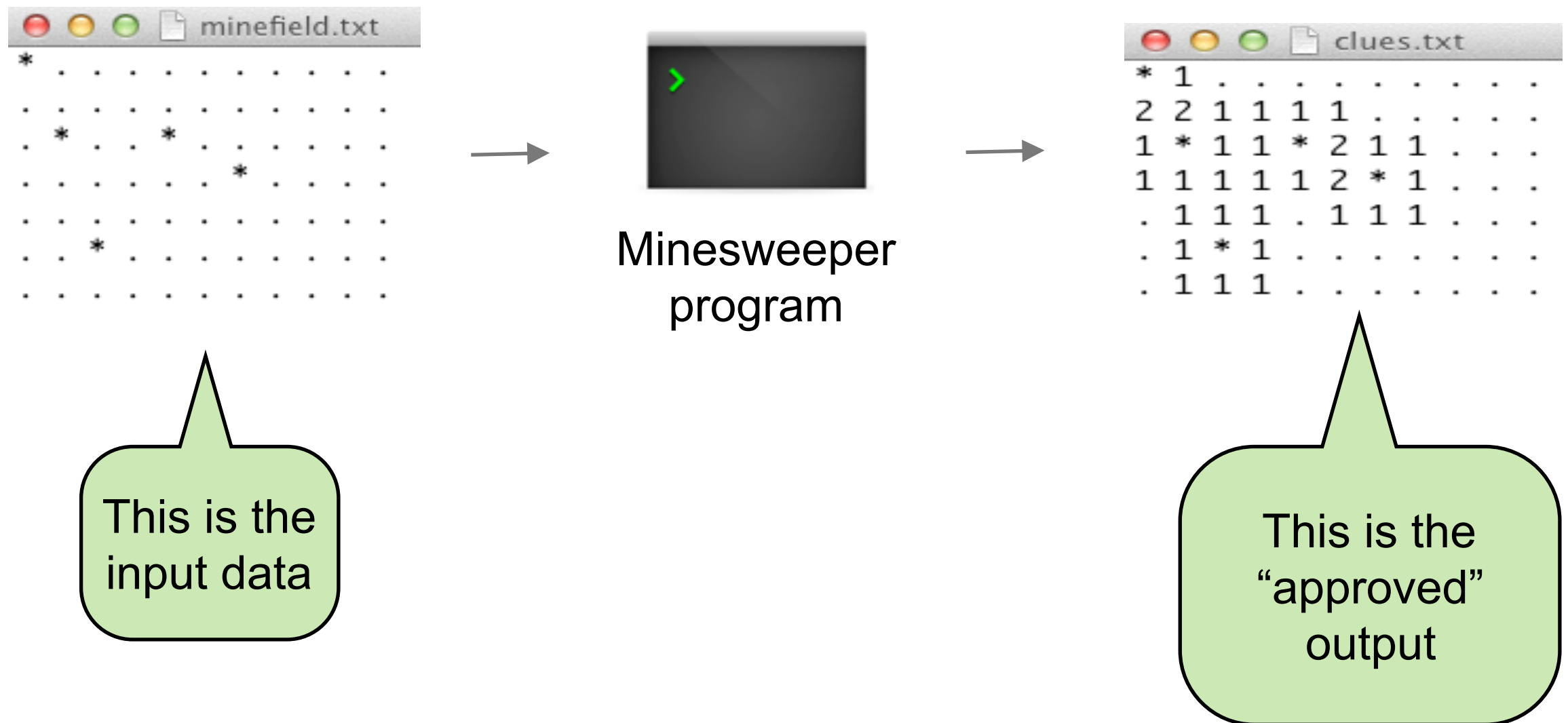
# Test Failure: Textual Diff

**TextTest**

- Open Source
- Approval Testing
- Data-Driven Testing
- Rich-Client Capture-Replay Testing

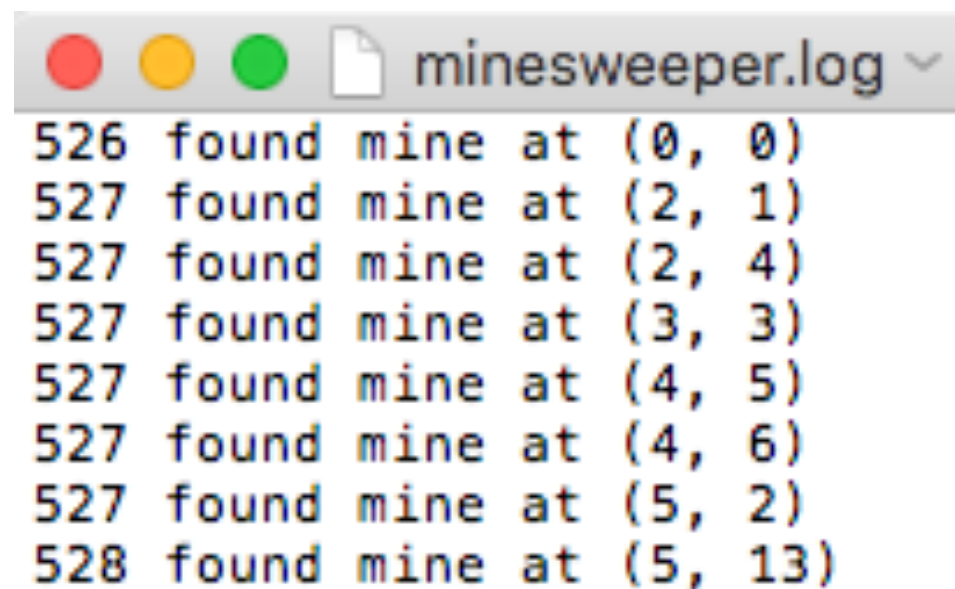http://texttest.org

# Demo

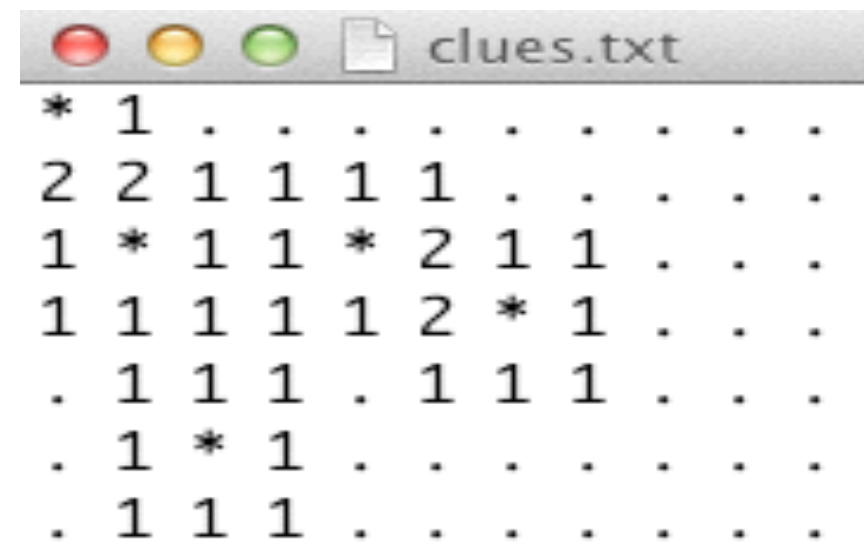# Addressing the problems

```java
33    @Test
34    public void getSpeedNorwegianBlueParrot_nailed() {
35        Parrot parrot = new Parrot(ParrotTypeEnum.NORWEGIAN_BLUE, 0, 0, true);
36        assertEquals(parrot.toString(), "Parrot(type=NORWEGIAN_BLUE, speed=2.0)");
37    }
```

- Test isolation - many tests with similar strings — **Tool Support**

- Maintenance costs - updating comparison strings — **Tool Support**

- A lot of 'toString' methods needed — **Test fixtures, data driven testing**

- TDD - in practice you don't write the assertions up front any more - copy and paste from actual output — **Iterative TDD, Tool Support**

# Assert or Filter?



```
526 found mine at (0, 0)
527 found mine at (2, 1)
527 found mine at (2, 4)
527 found mine at (3, 3)
527 found mine at (4, 5)
527 found mine at (4, 6)
527 found mine at (5, 2)
528 found mine at (5, 13)
```
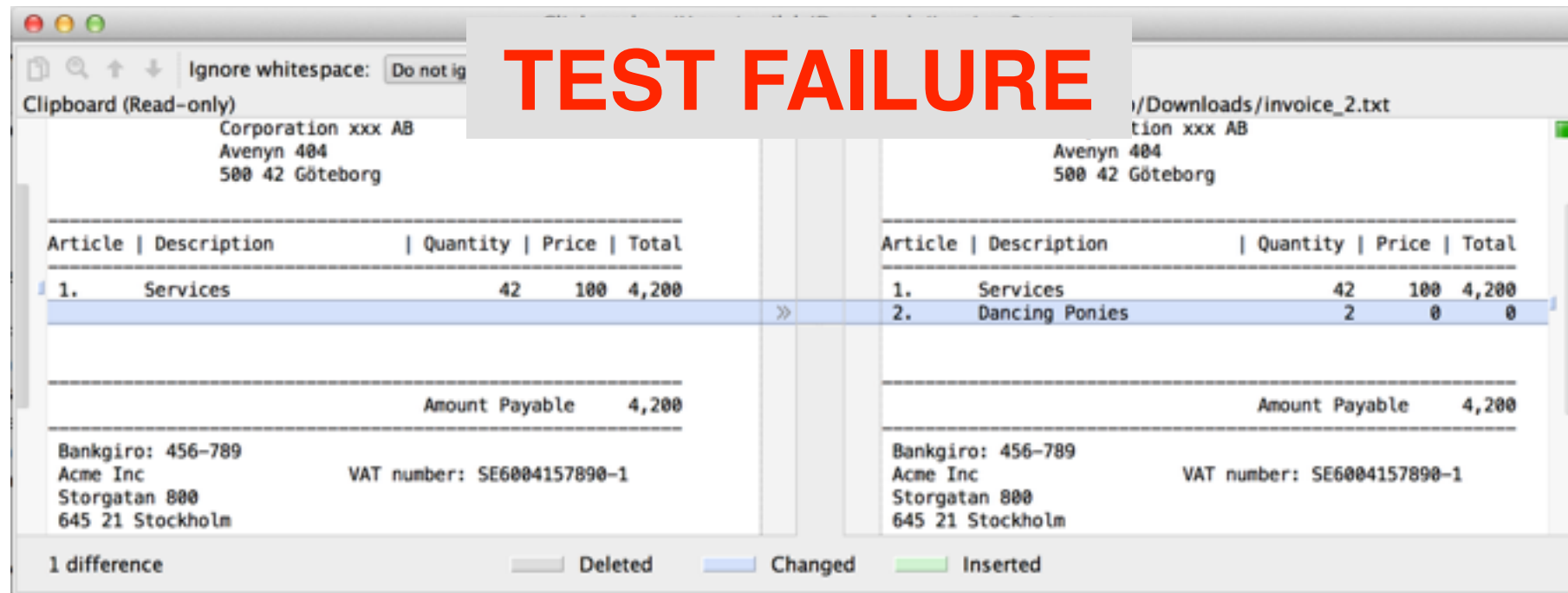


```
* 1 . . . . . . . . . . .
2 2 1 1 1 1 . . . . . . .
1 * 1 1 * 2 1 1 . . . . .
1 1 1 1 1 2 * 1 . . . . .
. 1 1 1 . 1 1 1 . . . . .
. 1 * 1 . . . . . . . . .
. 1 1 1 . . . . . . . . .
```

```
[run_dependent_text]
log:^\d+{REPLACE <timestamp>}
[end]
```

```
assertEquals(2, clueAt(0,1))
assertEquals(1, clueAt(1,0))
...
```
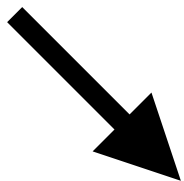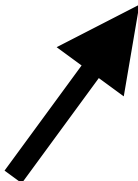
# Approval tests: advantages



- Find defects you didn't anticipate

- Less work to create new test cases

- When behaviour changes, easy to update many tests

# Real-World examples

Pagero Online
Cloud Service



Invoice
Issuer

Invoice
Recipient

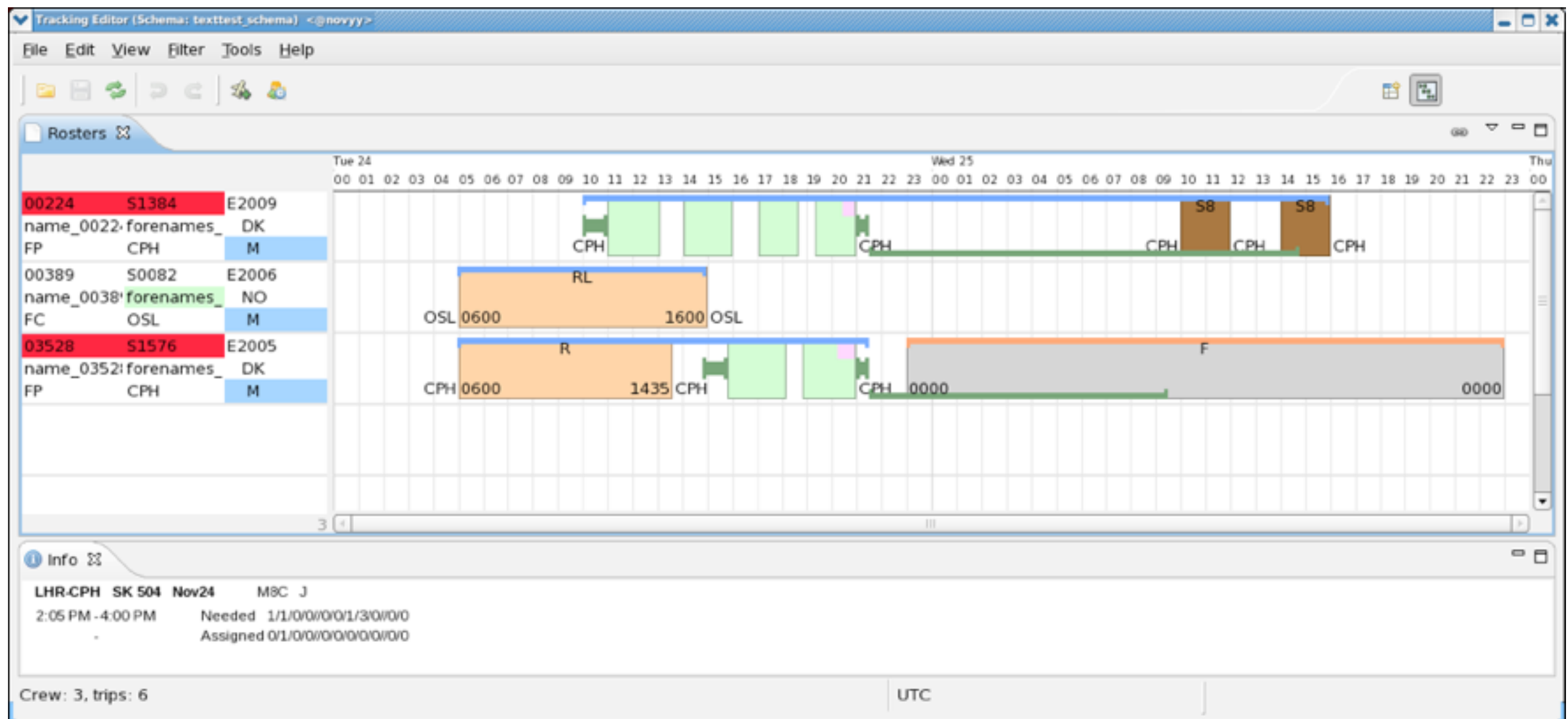# Change in Invoice received

Approved Output          New Output

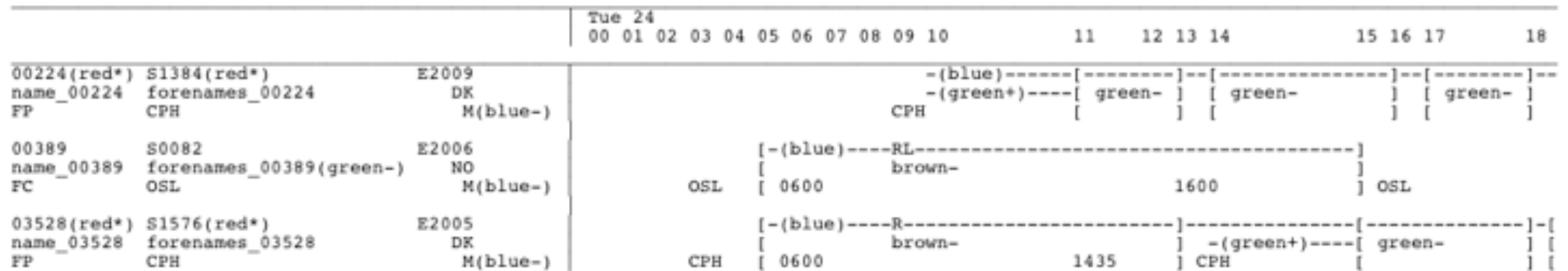# Crew Management System

# Use-Case Recording

```
open default plan and show rosters
change service type of activity JPO 2537 for name_0352 to F
undo
wait for completion of undo
close and discard changes
```

# ASCII-art