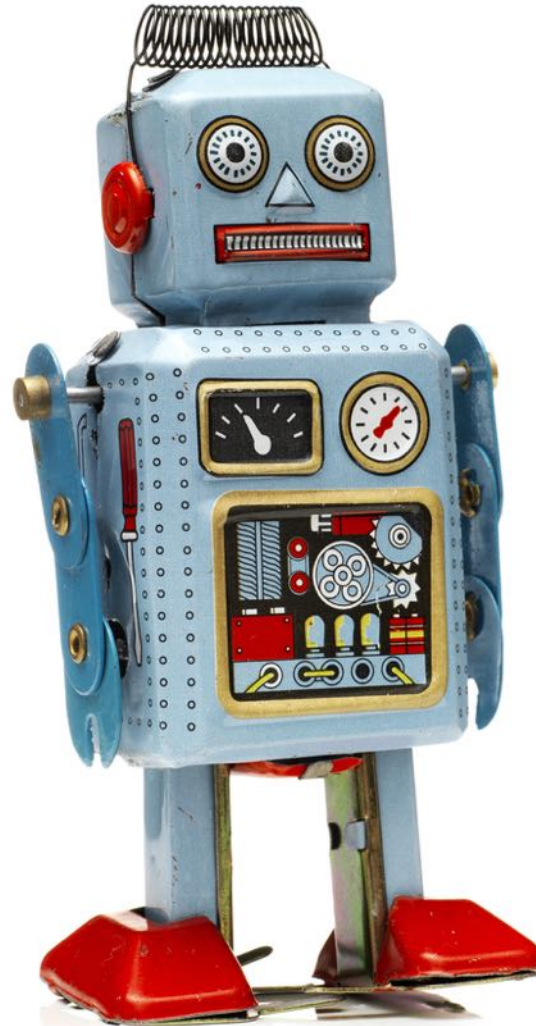


Automatic Bug Fixing



Bug Fixes

Added new code segment in a commit:

```
always_comb
begin : MUX
  if (sel == 1'b0) beggin ← Syntax Error
    mux_out = din_0;
  end else begin
    mux_out = din_1;
  end
end
```

2 Ways to Fix

Unroll Entire Commit

```
always_comb
begin : MUX
  if (sel == 1'b0) beggin
    mux_out = din_0;
  end else begin
    mux_out = din_1;
  end
end
```

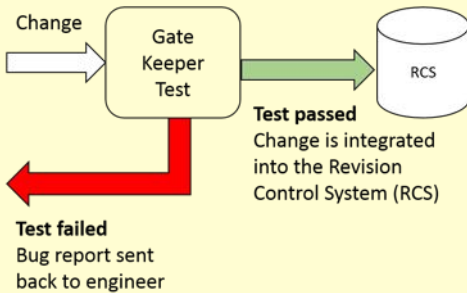
Fix Root Cause

```
always_comb
begin : MUX
  if (sel == 1'b0) begin
    mux_out = din_0;
  end else begin
    mux_out = din_1;
  end
end
```

Regression Test Setups

Continuous Integration

- Gate keeper
- Short directed tests



Post-Integration Testing

- Large Test Suites
- Random/Directed Tests



Automatic Debug

PinDown

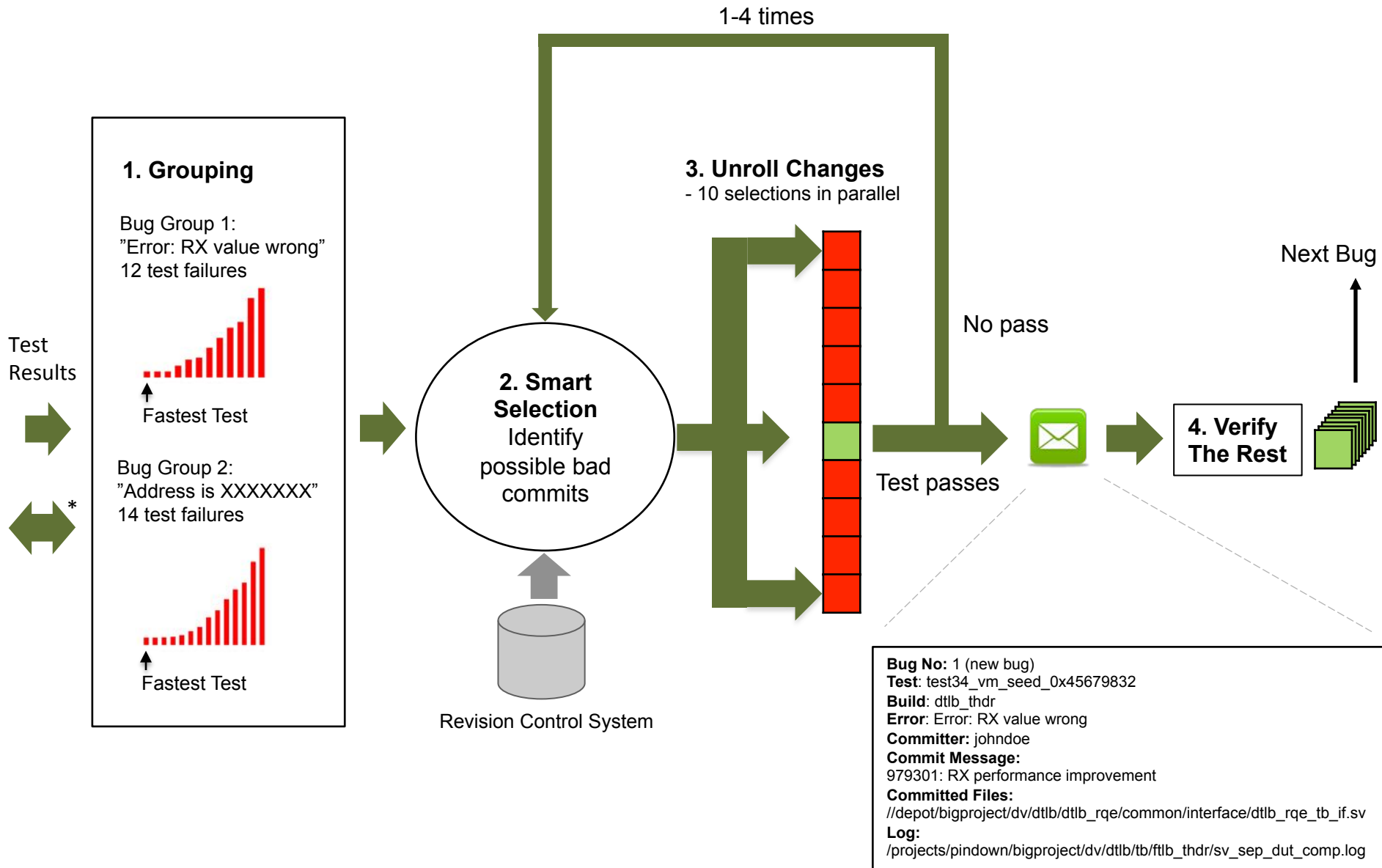
Test Results Open Bugs Diagnose Settings

Config	Test	maj 11 2012 8:51	maj 11 2012 8:52	maj 11 2012 8:52
config_1	Build Result	Green	Green	Green
config_1	t1	Green	Red	Green
config_1	t2	Red	Green	Green
config_1	t3	Green	Green	Red
config_2	Build Result	Green	Green	Green
config_2	t1	Red	Green	Green
config_2	t2	Green	Green	Green
config_2	t3	Green	Green	Green

or

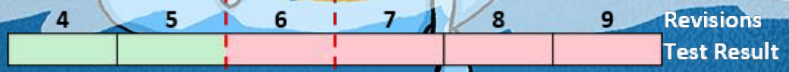
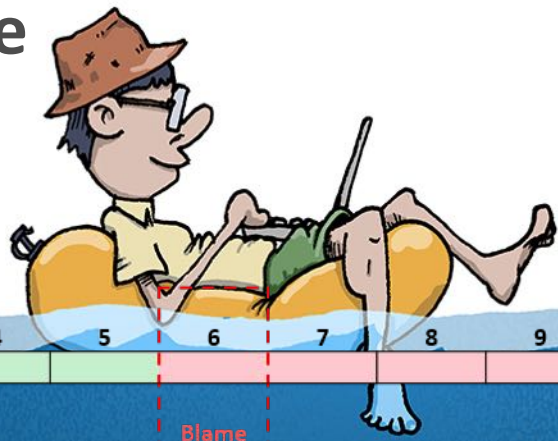
Inhouse Solution (e.g using
GIT Bisect)

PinDown Debug Algorithm

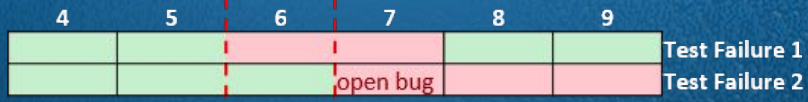


```
Bug No: 1 (new bug)  
Test: test34_vm_seed_0x45679832  
Build: dtlb_thdr  
Error: Error: RX value wrong  
Committer: johndoe  
Commit Message:  
979301: RX performance improvement  
Committed Files:  
//depot/bigproject/dv/dtlb/dtlb_rqe/common/interface/dtlb_rqe_tb_if.sv  
Log:  
/projects/pin-down/bigproject/dv/dtlb/tb/ftlb_thdr/sv_sep_dut_comp.log
```

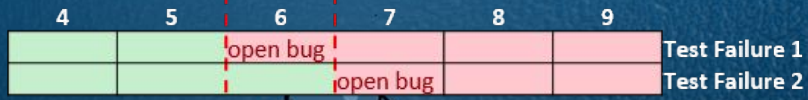
The Principle is Simple



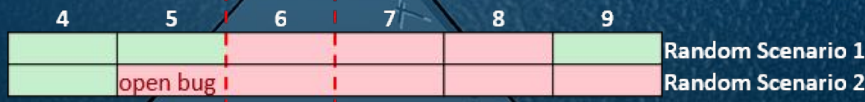
Overlapping Failures 1



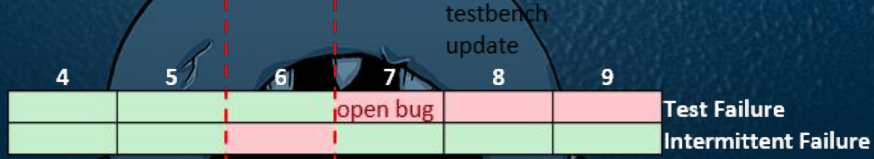
Overlapping Failures 2



Random Instability



Intermittent Failures



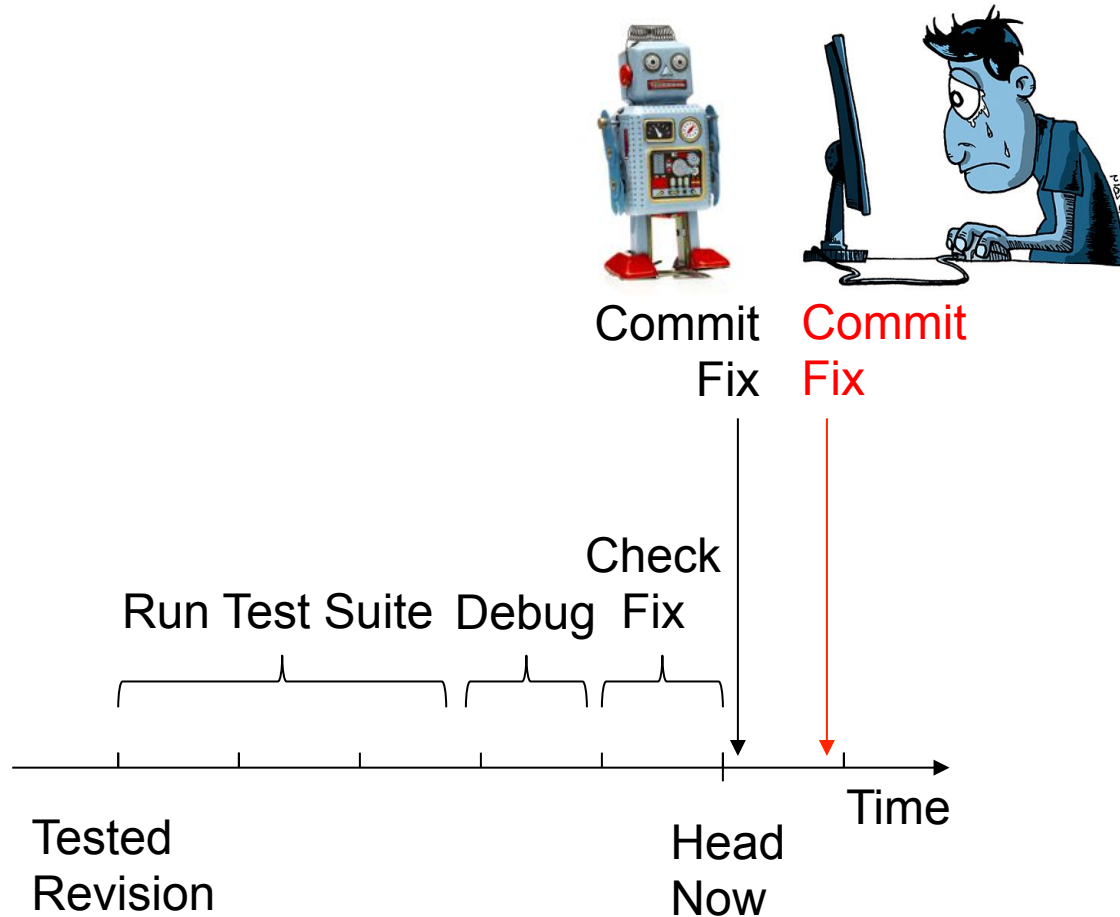
Build Failures



...but the reality is difficult

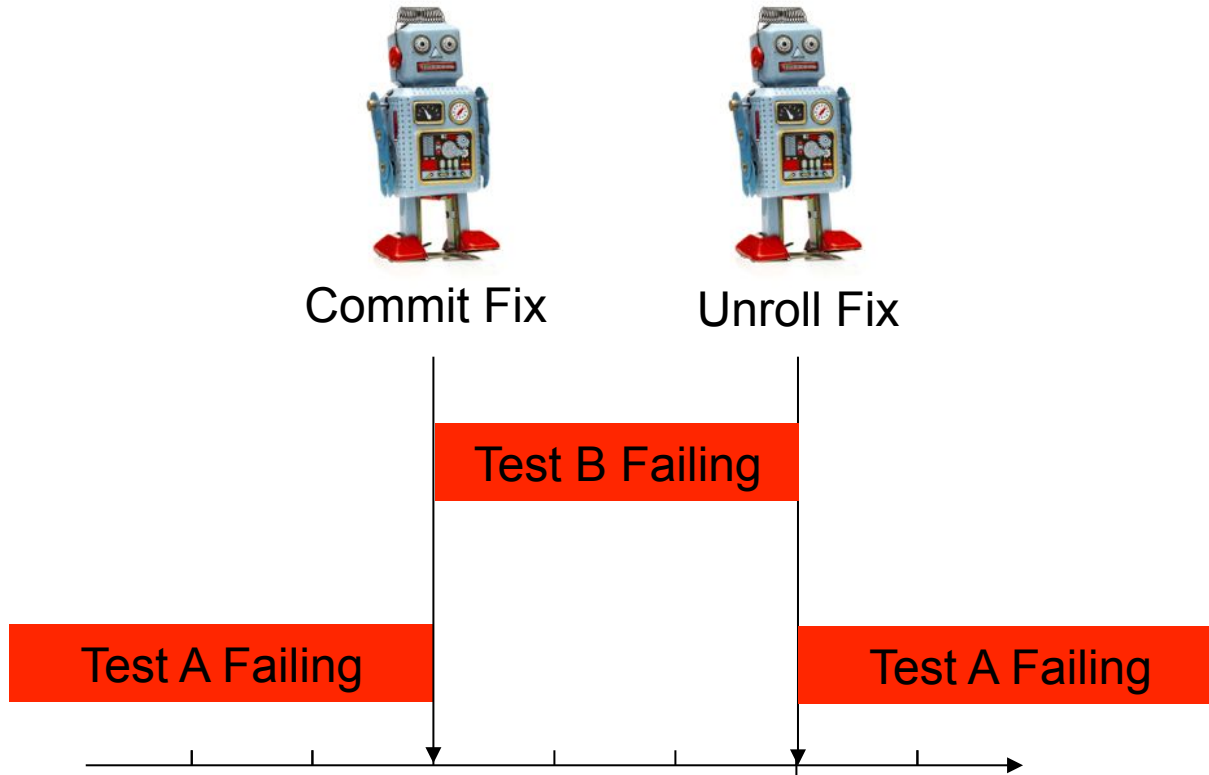
PinDown supports all scenarios

Human – Tool Race Condition



Solution: Before committing a fix, check if a fix has already been committed. Both humans and tools need to do this.

Fault Oscillation



Solution: The tool must not unroll its own fixes. In those cases just send a bug report.

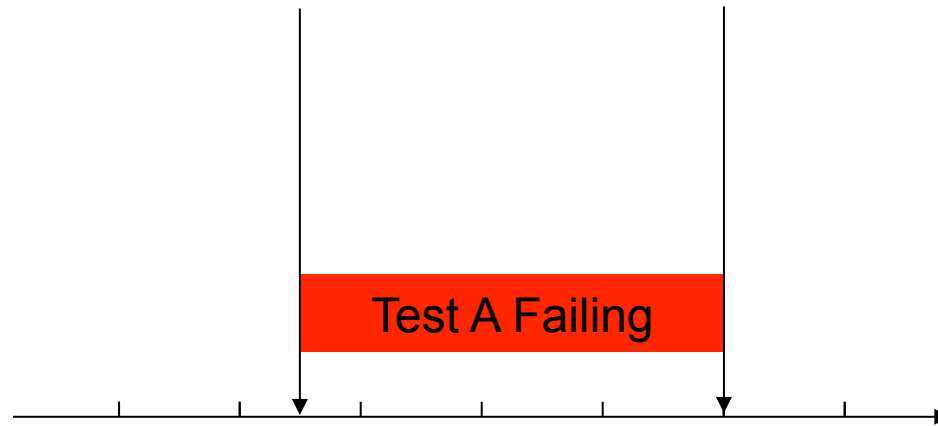
Partial Introduction of a New Feature



Introduce Feature



Remove it

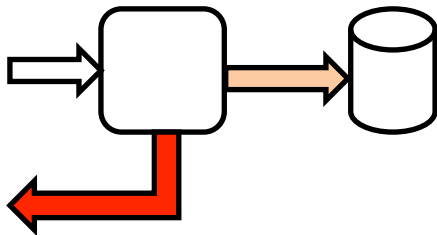


Solution: Don't allow partial introduction of a feature which makes a test fail. This rule applies to a continuous integration gate keeper as well.

Constrained random tests may find issues weeks after inserting

Summary

Continuous Integration Gate Keeper



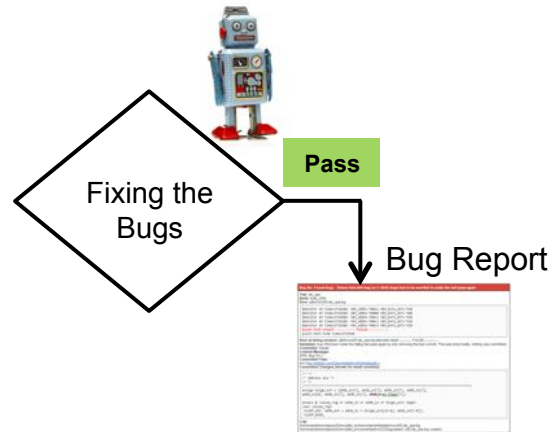
Test Suites

Short Test Suites
Directed Tests Only

Policy Implications

No partial checkins that causes tests to fail

Automatic Bug Fix Local



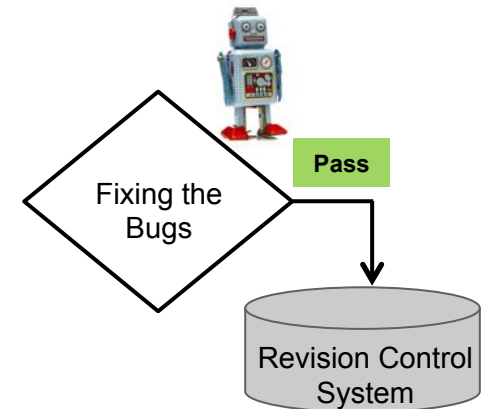
Test Suites

Any Test Suite Size
Random, Directed Tests

Policy Implications

None

Automatic Bug Fix with Commit



Test Suites

Any Test Suite Size
Random, Directed Tests

Policy Implications

- No partial checkins that causes tests to fail
- Check tool/RCS before commit