Improving Debug Productivity using latest AI & ML Techniques

Amod Khandekar

Sr Principal Customer Engagement Engineer

Cadence

Amod has been working with Cadence for past 20 years. He has worked as Pre Sales and Customer Support Engineer supporting all the verification related tools. Currently he is part of Product Engineering team with Focus on Verisium Apps.

Sundararajan Ananthakrishnan

Application Engineer Architect

Cadence

Sundar has been at Cadence for 13 years, with an overall Experience of 23+ years in the field of SoC & IP Verification. Sundar leads the Cadence AVS field team for India region and drives the Advanced ML/AI and Verification Solutions in the region.

Amit Verma

Software Engineering Director

Cadence

Amit has been at Cadence for 18 years, and until 2021 he was managing multiple coverage related Formal apps for Cadence’s Formal product, IEV earlier and currently Jasper. For 2 years he has been managing innovative software developments for Verisium bug-localization apps.

He is passionate about exploring how the ML/AI and Verification technologies can be used together to accelerate the design verification.

Improving Debug Productivity using latest AI & ML Techniques

Amod Khandekar, Amit Verma, Sundararajan A
Cadence Design Systems

Narasimha Rao Chinni
Samsung

Agenda

What

Challenges & related asks for failures debug

What

Solutions to failures debug related problems

How

Run apps to faster failures debug

What

Productivity gains for failures debug time

Agenda

What

Challenges & related asks for failures debug

What

Solutions to failures debug related problems

How

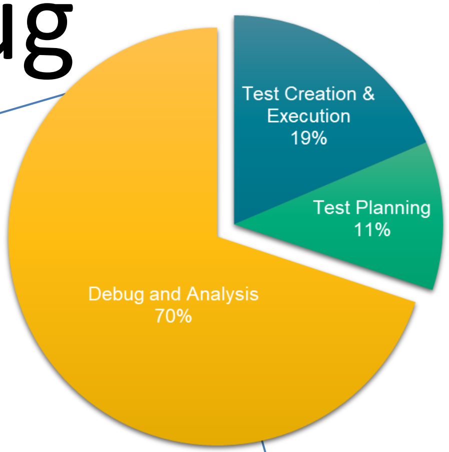
Run apps to faster failures debug

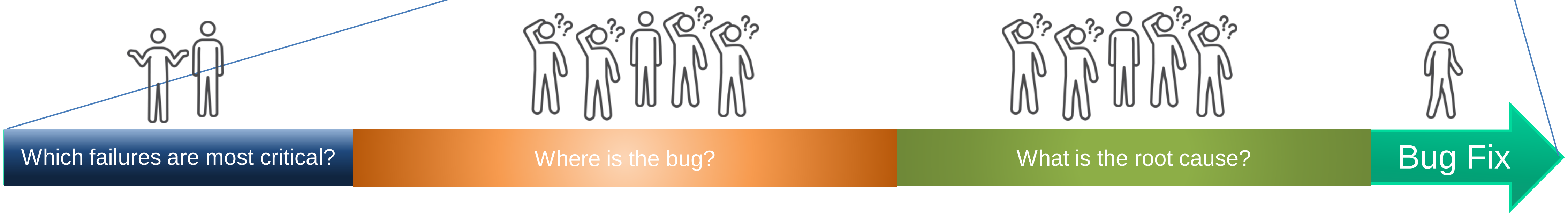
What

Productivity gains for failures debug time

Verification Challenge – SoC Debug

Major Pain Points – Large debug time

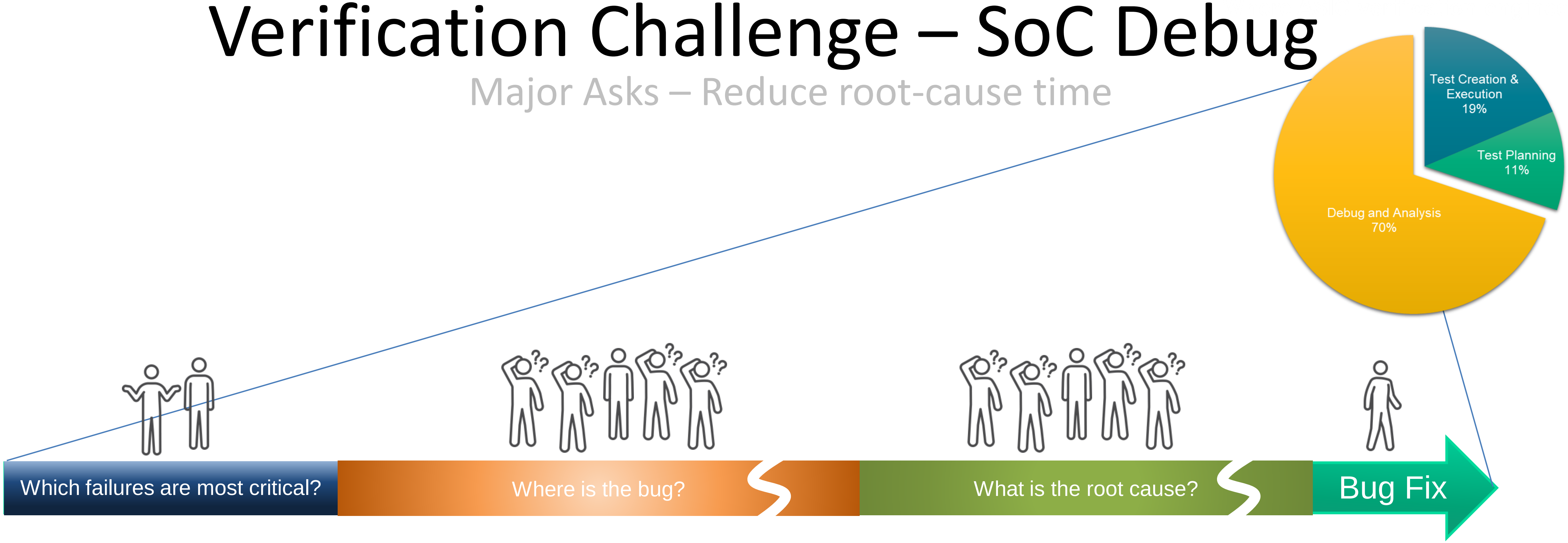




- SoCs integrate hundreds of IP
- Each of the IP is constantly changing, evolving, improving
- Week to week, SoC-level testing results in several test failures
- 2 major pain points
  - Determining the **root cause (source code)** of the failure takes significant time/resources
  - Determining the **test** to reproduce the failure in shortest time takes significant time/resources

Verification Challenge – SoC Debug

Major Asks – Reduce root-cause time



- Provide the Semantic behavior changes (Structural) for quick analysis for design changes
- Provide the Functional behavior changes for accurate analysis for design changes
- Provide the failure causing repository version from large chain of versions for filtered analysis of failure
- Provide the shortest failing test per failure for reproducing faster a failure

Agenda

What

Challenges & related asks for failures debug

What

Solutions to failures debug related problems

How

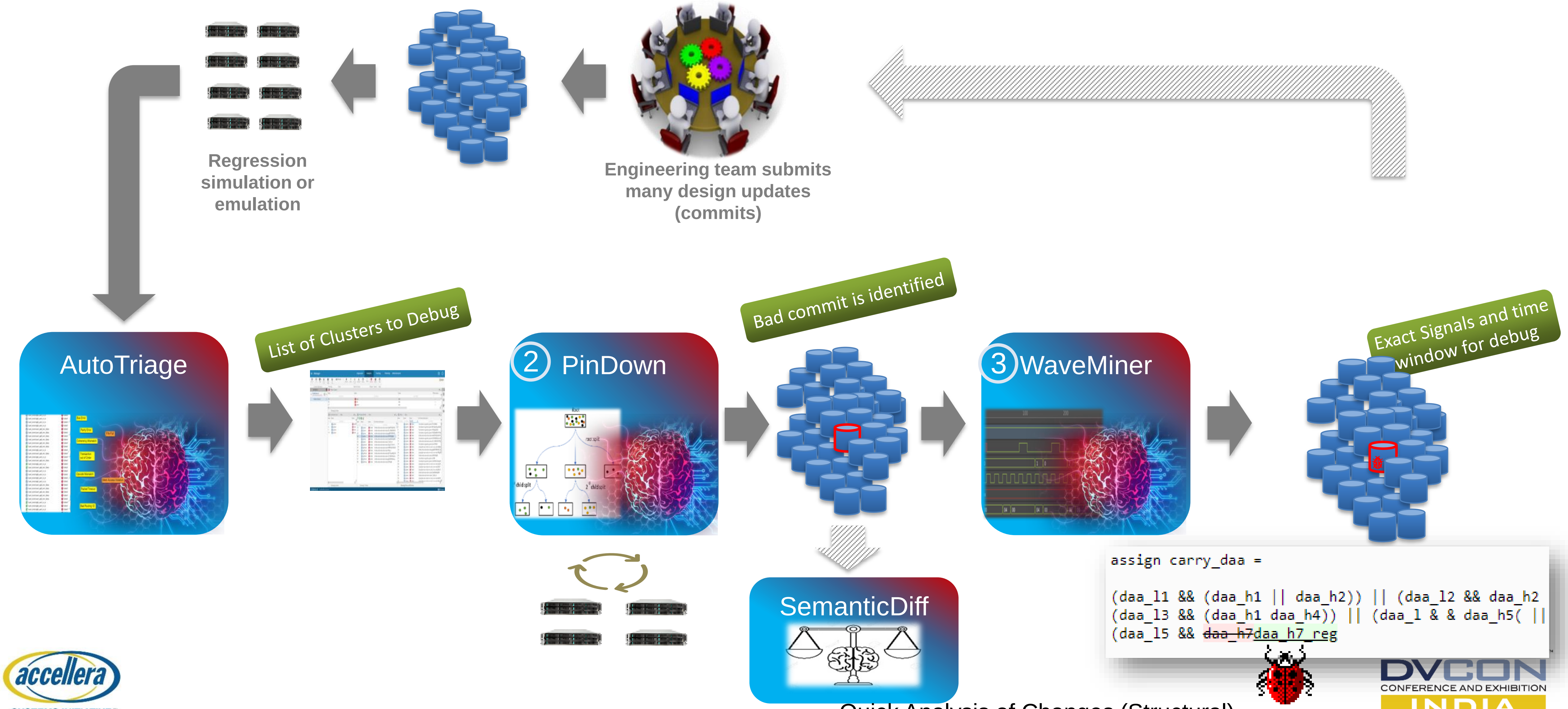
Run apps to faster failures debug

What

Productivity gains for failures debug time

Automatic Bug Localization Solution & Apps

Given **two design versions** and **regression failure**,
predict/locate with high accuracy the **root-cause at source code**



Regression simulation or emulation
Engineering team submits many design updates (commits)
AutoTriage
List of Clusters to Debug
2 PinDown
Bad commit is identified
3 WaveMiner
Exact Signals and time window for debug
SemanticDiff

```
assign carry_daa =

(daa_l1 && (daa_h1 || daa_h2)) || (daa_l2 && daa_h2
(daa_l3 && (daa_h1 daa_h4)) || (daa_l & & daa_h5( ||
(daa_l5 && ~~daa_h7~~daa_h7_reg
```

Quick Analysis of Changes (Structural)

Agenda

What

Challenges & related asks for failures debug

What

Solutions to failures debug related problems

How

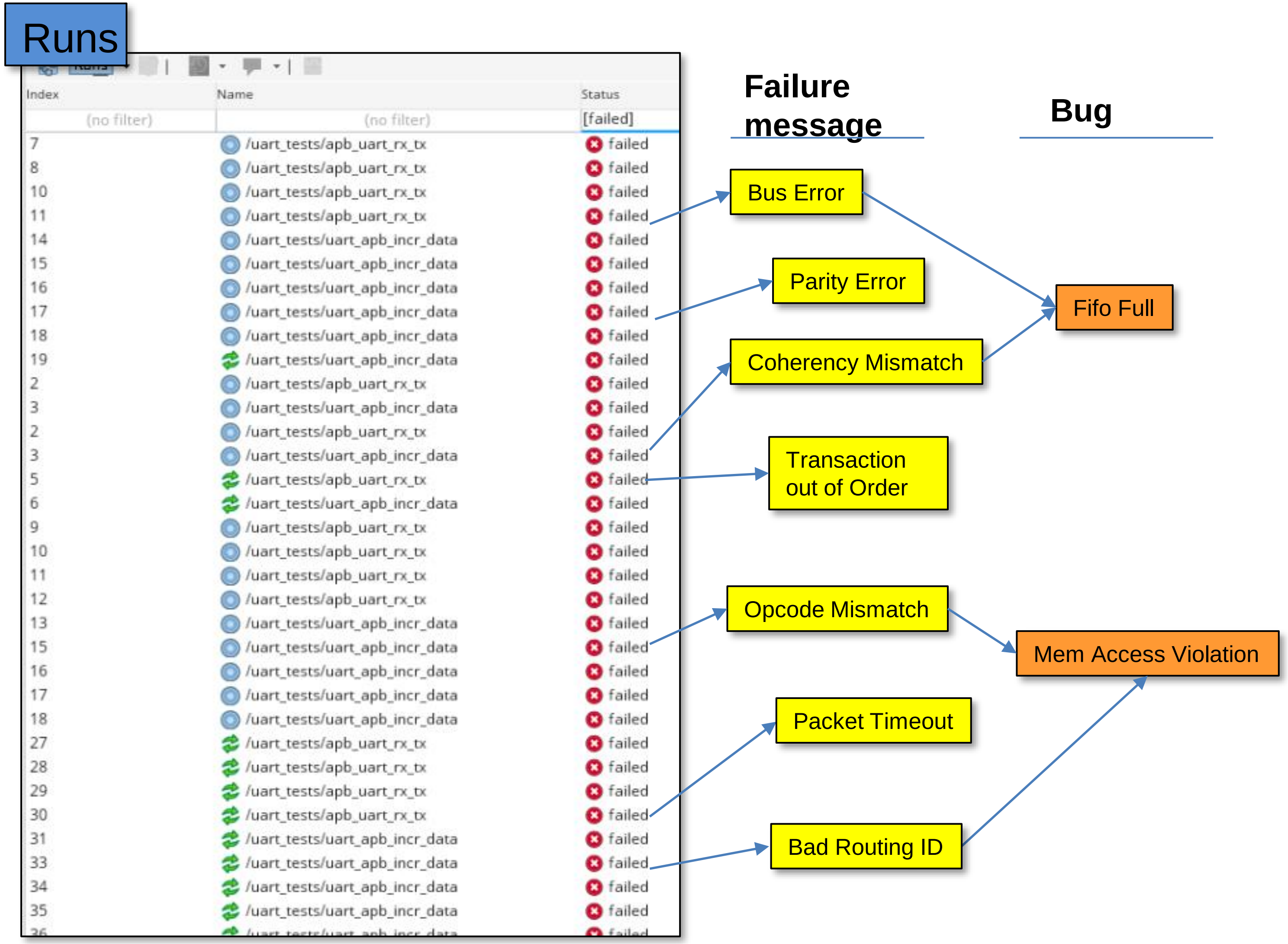
Run apps to faster failures debug

What

Productivity gains for failures debug time

Verisium AutoTriage

Vanilla Failure Triage Flow



Runs

Index	Name	Status
(no filter)	(no filter)	[failed]
7	/uart_tests/apb_uart_rx_tx	failed
8	/uart_tests/apb_uart_rx_tx	failed
10	/uart_tests/apb_uart_rx_tx	failed
11	/uart_tests/apb_uart_rx_tx	failed
14	/uart_tests/uart_apb_incr_data	failed
15	/uart_tests/uart_apb_incr_data	failed
16	/uart_tests/uart_apb_incr_data	failed
17	/uart_tests/uart_apb_incr_data	failed
18	/uart_tests/uart_apb_incr_data	failed
19	/uart_tests/uart_apb_incr_data	failed
2	/uart_tests/apb_uart_rx_tx	failed
3	/uart_tests/uart_apb_incr_data	failed
2	/uart_tests/apb_uart_rx_tx	failed
3	/uart_tests/uart_apb_incr_data	failed
5	/uart_tests/apb_uart_rx_tx	failed
6	/uart_tests/uart_apb_incr_data	failed
9	/uart_tests/apb_uart_rx_tx	failed
10	/uart_tests/apb_uart_rx_tx	failed
11	/uart_tests/apb_uart_rx_tx	failed
12	/uart_tests/apb_uart_rx_tx	failed
13	/uart_tests/uart_apb_incr_data	failed
15	/uart_tests/uart_apb_incr_data	failed
16	/uart_tests/uart_apb_incr_data	failed
17	/uart_tests/uart_apb_incr_data	failed
18	/uart_tests/uart_apb_incr_data	failed
27	/uart_tests/apb_uart_rx_tx	failed
28	/uart_tests/apb_uart_rx_tx	failed
29	/uart_tests/apb_uart_rx_tx	failed
30	/uart_tests/apb_uart_rx_tx	failed
31	/uart_tests/uart_apb_incr_data	failed
33	/uart_tests/uart_apb_incr_data	failed
34	/uart_tests/uart_apb_incr_data	failed
35	/uart_tests/uart_apb_incr_data	failed
36	/uart_tests/uart_apb_incr_data	failed

Failure message

Bug

Bus Error

Parity Error

Coherency Mismatch

Transaction out of Order

Opcode Mismatch

Packet Timeout

Bad Routing ID

Fifo Full

Mem Access Violation

<u>Today's failure triage</u>

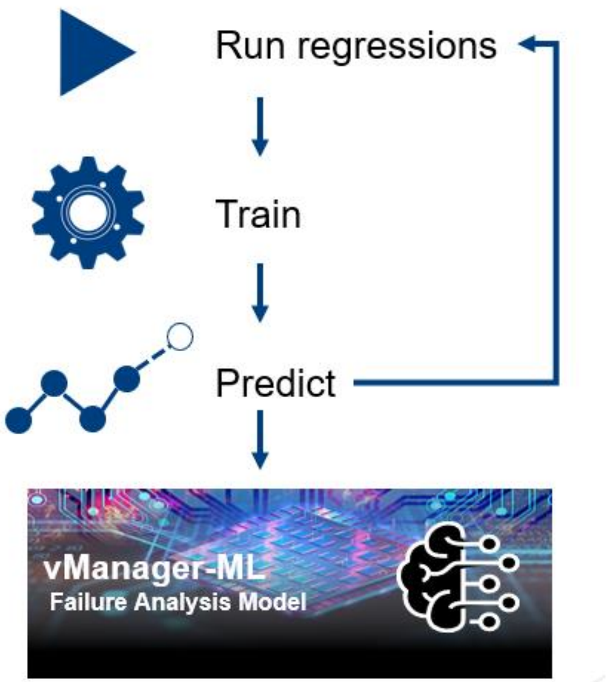
- Group runs by their failure description
- Are they really from the same bug?
- **Many** different failures messages could be associated from **one** bug
- **One** failure message can be caused by **many** bugs

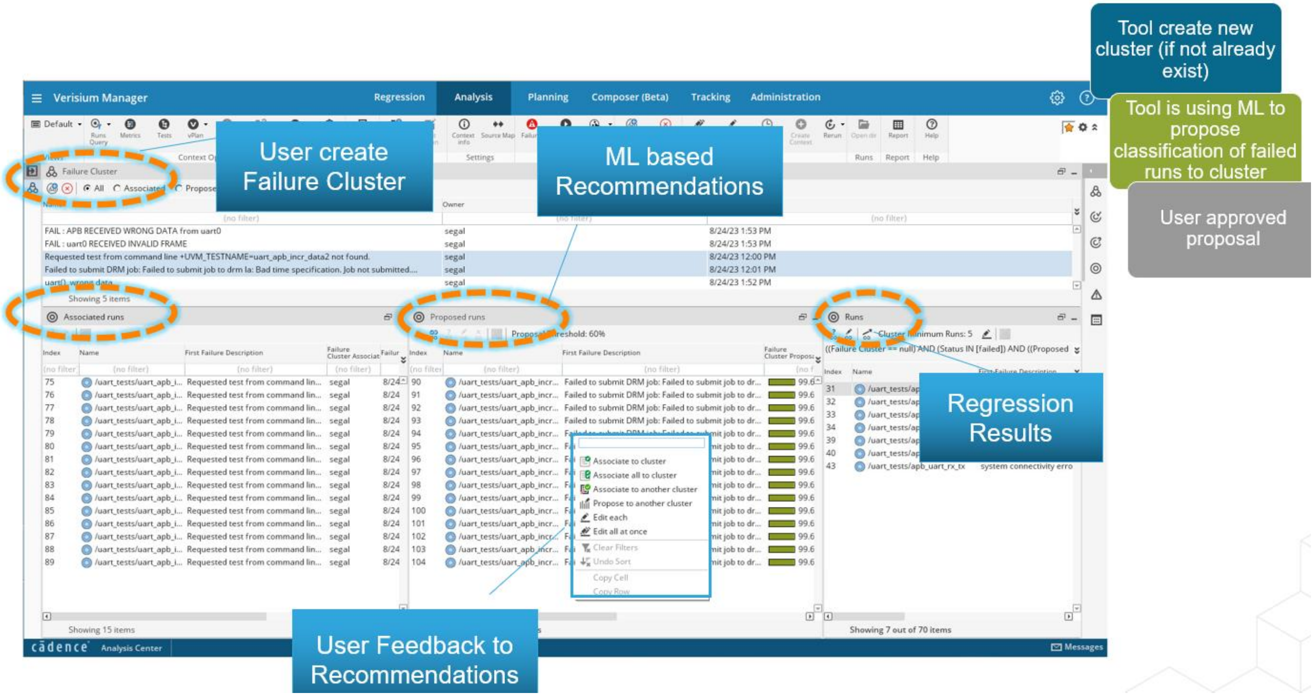
<u>Impact</u>

- Manual triage is difficult
- Bugs could be characterized by a variety of conditions
- Automated scripted solutions are not smart enough

AutoTriage – Automated ML Bucketing of Regression Failures

- Solution
  - Automate the failure analysis/classification using Supervised Learning ML
  - Initial results show success rate of **~95% prediction**





Failure Cluster

Name	Owner	
(no filter)		(no filter)
FAIL : APB RECEIVED WRONG DATA from uart0	segal	8/24/23 1:53 PM
FAIL : uart0 RECEIVED INVALID FRAME	segal	8/24/23 1:53 PM
Requested test from command line +UVM_TESTNAME=uart_apb_incr_data2 not found.	segal	8/24/23 12:00 PM
Failed to submit DRM job: Failed to submit job to drm la: Bad time specification. Job not submitted....	segal	8/24/23 12:01 PM
uart0 wrong data	segal	8/24/23 1:52 PM

Showing 5 items

Associated runs

Index	Name	First Failure Description	Failure Cluster Associat	Failur
(no filter)	(no filter)	(no filter)	(no filter)	
75	/uart_tests/uart_apb_i...	Requested test from command lin...	segal	8/24
76	/uart_tests/uart_apb_i...	Requested test from command lin...	segal	8/24
77	/uart_tests/uart_apb_i...	Requested test from command lin...	segal	8/24
78	/uart_tests/uart_apb_i...	Requested test from command lin...	segal	8/24
79	/uart_tests/uart_apb_i...	Requested test from command lin...	segal	8/24
80	/uart_tests/uart_apb_i...	Requested test from command lin...	segal	8/24
81	/uart_tests/uart_apb_i...	Requested test from command lin...	segal	8/24
82	/uart_tests/uart_apb_i...	Requested test from command lin...	segal	8/24
83	/uart_tests/uart_apb_i...	Requested test from command lin...	segal	8/24
84	/uart_tests/uart_apb_i...	Requested test from command lin...	segal	8/24
85	/uart_tests/uart_apb_i...	Requested test from command lin...	segal	8/24
86	/uart_tests/uart_apb_i...	Requested test from command lin...	segal	8/24
87	/uart_tests/uart_apb_i...	Requested test from command lin...	segal	8/24
88	/uart_tests/uart_apb_i...	Requested test from command lin...	segal	8/24
89	/uart_tests/uart_apb_i...	Requested test from command lin...	segal	8/24

Showing 15 items

Proposed runs

Proposal Threshold: 60%

Index	Name	First Failure Description	Failure Cluster Propos
(no filter)	(no filter)	(no filter)	(no f
90	/uart_tests/uart_apb_incr...	Failed to submit DRM job: Failed to submit job to dr...	99.6
91	/uart_tests/uart_apb_incr...	Failed to submit DRM job: Failed to submit job to dr...	99.6
92	/uart_tests/uart_apb_incr...	Failed to submit DRM job: Failed to submit job to dr...	99.6
93	/uart_tests/uart_apb_incr...	Failed to submit DRM job: Failed to submit job to dr...	99.6
94	/uart_tests/uart_apb_incr...	Failed to submit DRM job: Failed to submit job to dr...	99.6
95	/uart_tests/uart_apb_incr...	Fai mit job to dr...	99.6
96	/uart_tests/uart_apb_incr...	Fai mit job to dr...	99.6
97	/uart_tests/uart_apb_incr...	Fai mit job to dr...	99.6
98	/uart_tests/uart_apb_incr...	Fai mit job to dr...	99.6
99	/uart_tests/uart_apb_incr...	Fai mit job to dr...	99.6
100	/uart_tests/uart_apb_incr...	Fai mit job to dr...	99.6
101	/uart_tests/uart_apb_incr...	Fai mit job to dr...	99.6
102	/uart_tests/uart_apb_incr...	Fai mit job to dr...	99.6
103	/uart_tests/uart_apb_incr...	Fai mit job to dr...	99.6
104	/uart_tests/uart_apb_incr...	Fai mit job to dr...	99.6

Runs

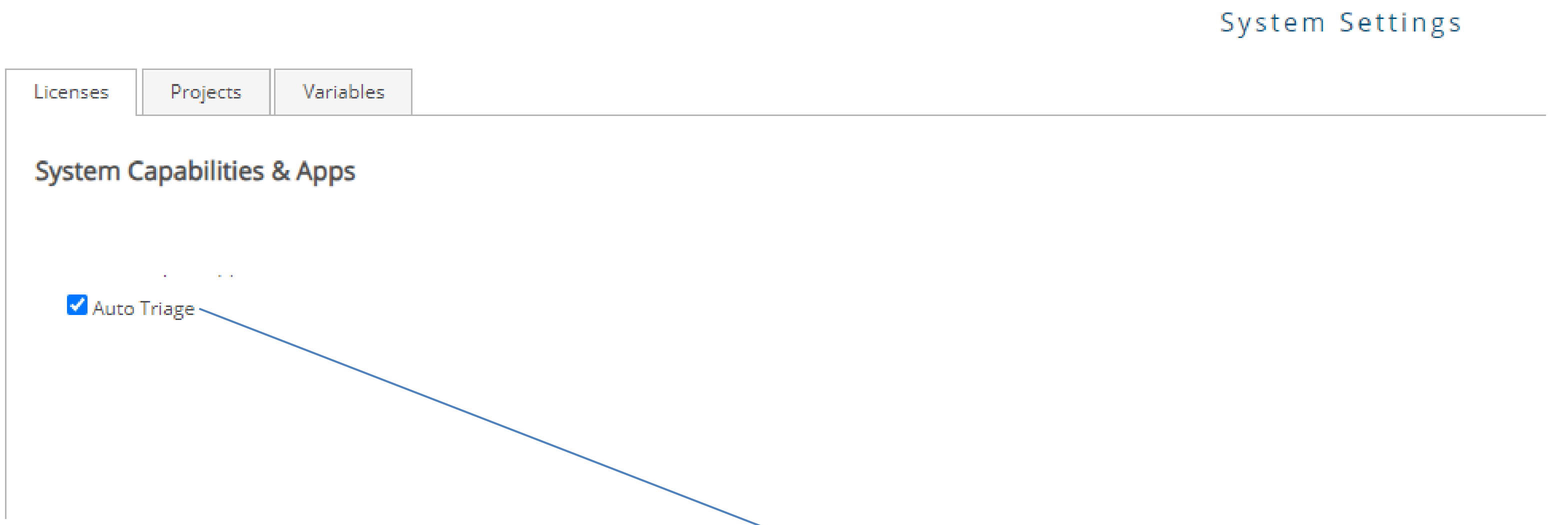
Cluster Minimum Runs: 5

((Failure Cluster == null) AND (Status IN [failed]) AND ((Proposed

Index	Name	First Failure Description
31	/uart_tests/ap	
32	/uart_tests/ap	
33	/uart_tests/ap	
34	/uart_tests/ap	
39	/uart_tests/ap	
40	/uart_tests/ap	
43	/uart_tests/apb_uart_rx_tx	system connectivity erro

Showing 7 out of 70 items

Configuring Verisium Manager for AutoTriage



System Settings

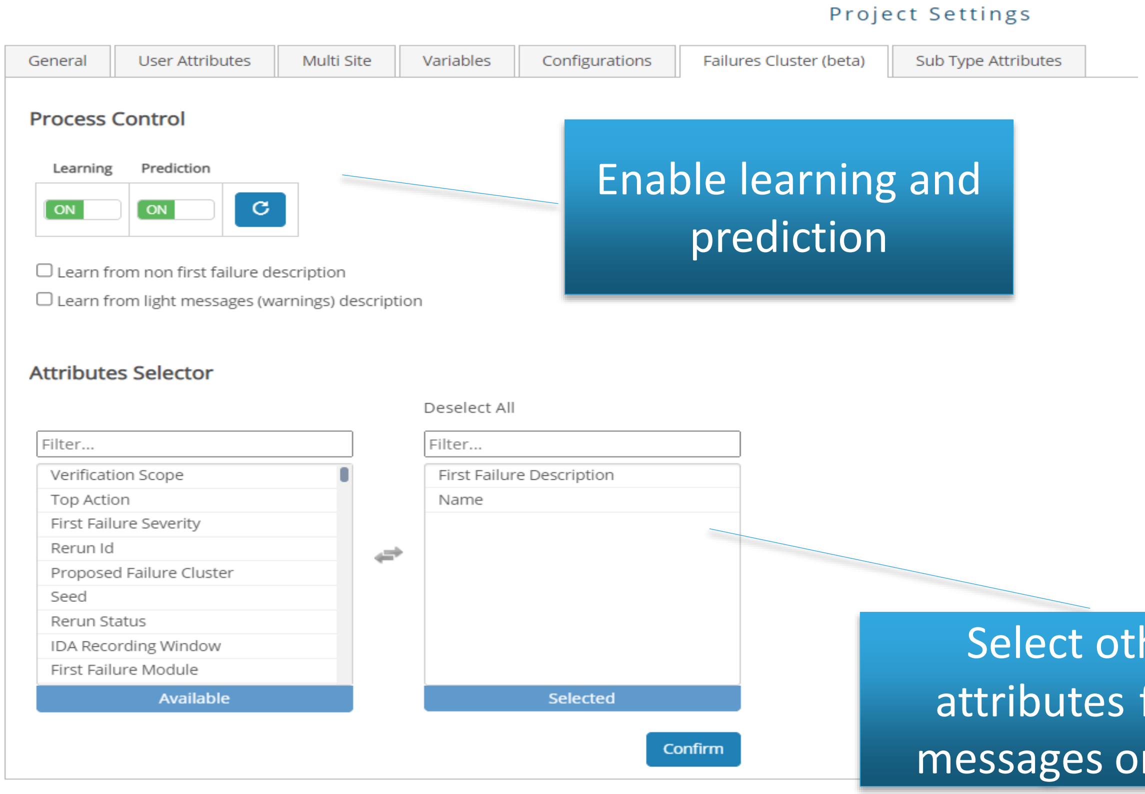
System Capabilities & Apps

☑ Auto Triage

This is very easy to enable and setup. User can start triaging failures in a very short time by enabling this in the vManager Web Admin Portal

Run Attributes used in learning

- By default, only the Run name and first failure description are used in the ML learning process however if other attributes are useful, they should be added **before** the learning process begins
  - Changing the attributes will **reset the learning**, however after reset if the previous data (failed run to cluster association) exist the tool learning curve is very fast



Project Settings

Process Control

Learning: ON

Prediction: ON

☐ Learn from non first failure description

☐ Learn from light messages (warnings) description

Attributes Selector

Available:

Filter...

- Verification Scope
- Top Action
- First Failure Severity
- Rerun Id
- Proposed Failure Cluster
- Seed
- Rerun Status
- IDA Recording Window
- First Failure Module

Selected:

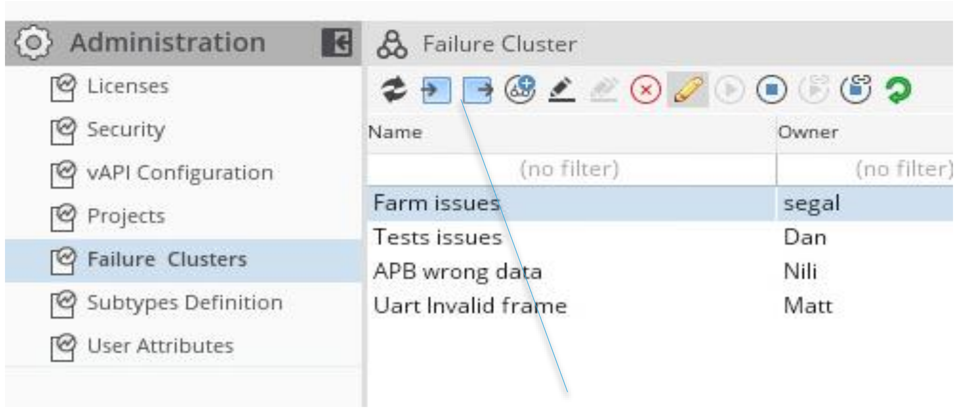
Deselect All

Filter...

- First Failure Description
- Name

Enable learning and prediction

Select other attributes from messages or runs



Name	Owner
(no filter)	(no filter)
Farm issues	segal
Tests issues	Dan
APB wrong data	Nili
Uart Invalid frame	Matt

Ability to export/import the clusters and the learning from one project to another

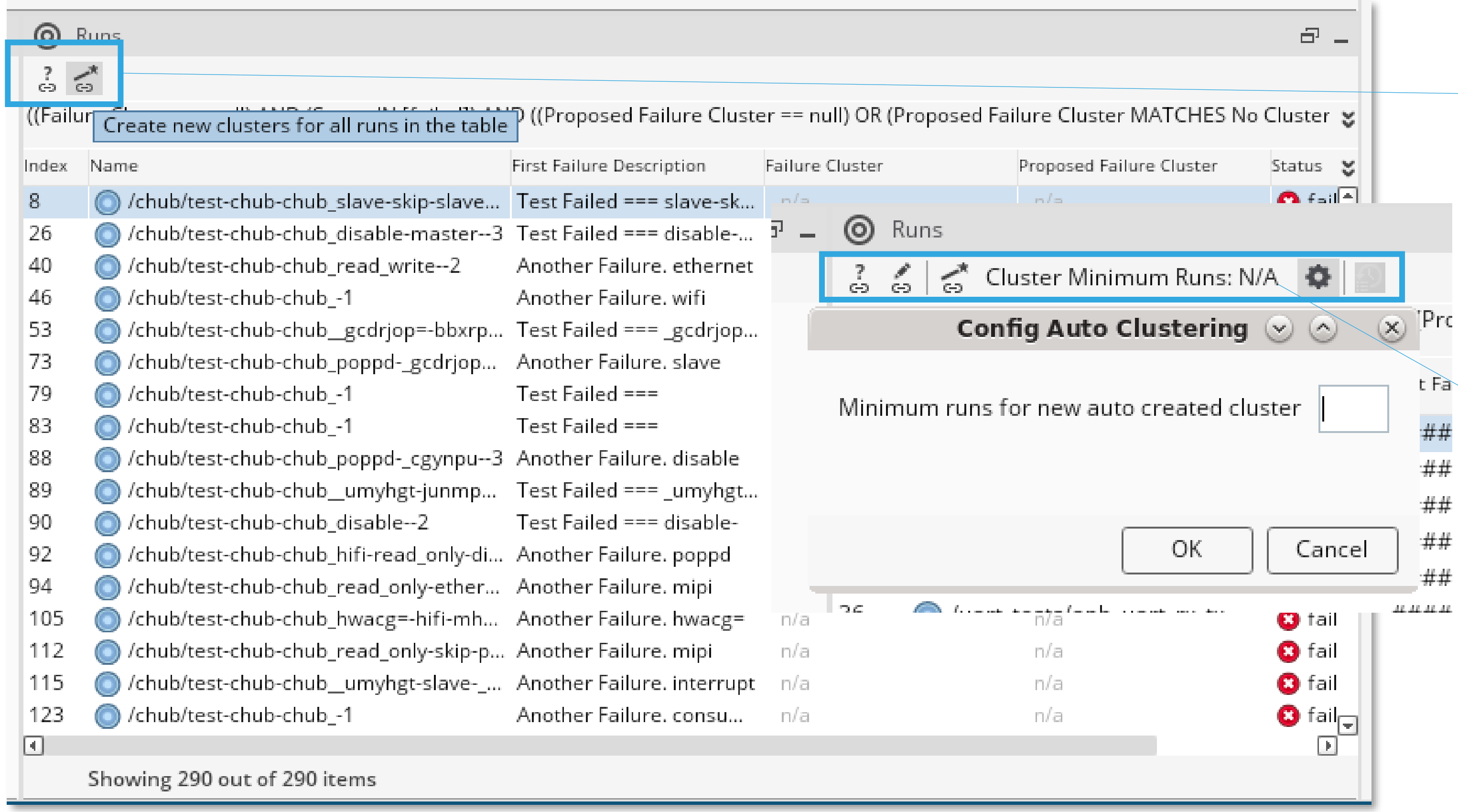
Enhancements to Create New Clusters Unsupervised ML

Unrecognized pattern (Unsupervised ML)

ML create new clusters, and proposed failed runs to the new Cluster

(trigger by user)

- New Button added to Failure Cluster Analysis Context to create Automatic Clusters



Runs
Create new clusters for all runs in the table
Runs
Cluster Minimum Runs: N/A
Config Auto Clustering
OK
Cancel

((Failur …) ((Proposed Failure Cluster == null) OR (Proposed Failure Cluster MATCHES No Cluster

Index	Name	First Failure Description	Failure Cluster	Proposed Failure Cluster	Status
8	/chub/test-chub-chub_slave-skip-slave...	Test Failed === slave-sk...	n/a	n/a	fail
26	/chub/test-chub-chub_disable-master--3	Test Failed === disable-...			
40	/chub/test-chub-chub_read_write--2	Another Failure. ethernet			
46	/chub/test-chub-chub_-1	Another Failure. wifi			
53	/chub/test-chub-chub__gcdrjop=-bbxrp...	Test Failed === _gcdrjop...			
73	/chub/test-chub-chub_poppd-_gcdrjop...	Another Failure. slave			
79	/chub/test-chub-chub_-1	Test Failed ===			
83	/chub/test-chub-chub_-1	Test Failed ===			
88	/chub/test-chub-chub_poppd-_cgynpu--3	Another Failure. disable			
89	/chub/test-chub-chub__umyhgt-junmp...	Test Failed === _umyhgt...			
90	/chub/test-chub-chub_disable--2	Test Failed === disable-			
92	/chub/test-chub-chub_hifi-read_only-di...	Another Failure. poppd			
94	/chub/test-chub-chub_read_only-ether...	Another Failure. mipi			
105	/chub/test-chub-chub_hwacg=-hifi-mh...	Another Failure. hwacg=	n/a	n/a	fail
112	/chub/test-chub-chub_read_only-skip-p...	Another Failure. mipi	n/a	n/a	fail
115	/chub/test-chub-chub__umyhgt-slave-_...	Another Failure. interrupt	n/a	n/a	fail
123	/chub/test-chub-chub_-1	Another Failure. consu...	n/a	n/a	fail

Showing 290 out of 290 items

Minimum runs for new auto created cluster

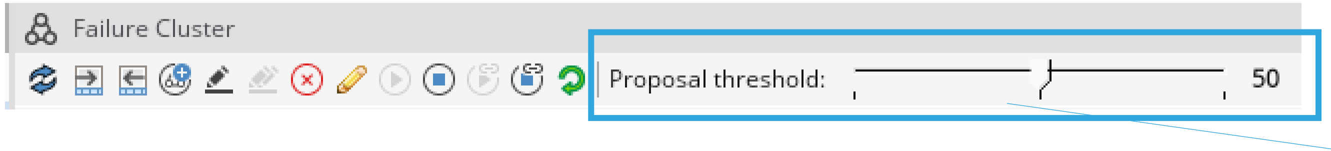
Create new clusters for all runs in the table

Advance option to control how tolerant the algorithm is towards noise

Enhancements to Control Automatically Proposed Existing Clusters, Supervised ML

Recognized pattern (Supervised ML)

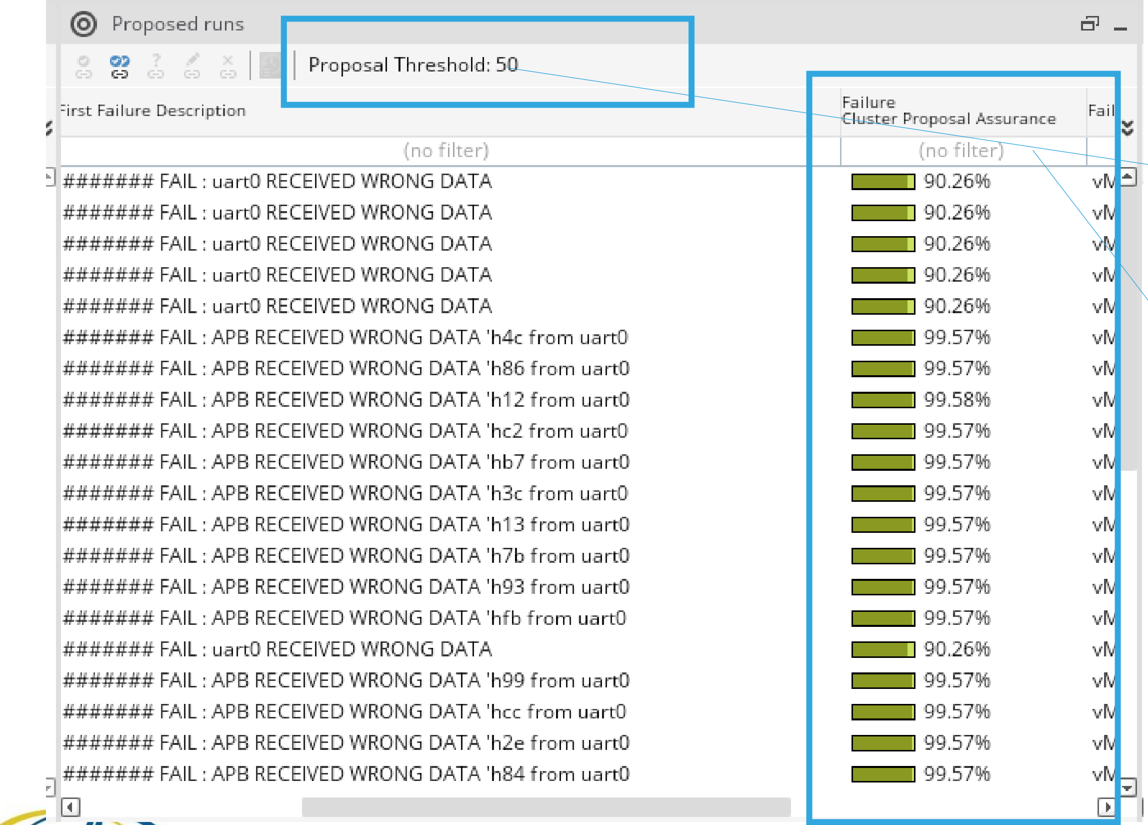
ML proposed failed runs to existing Clusters (automatically)



Failure Cluster

Proposal threshold: 50

The threshold can be configured in the admin page



Proposed runs

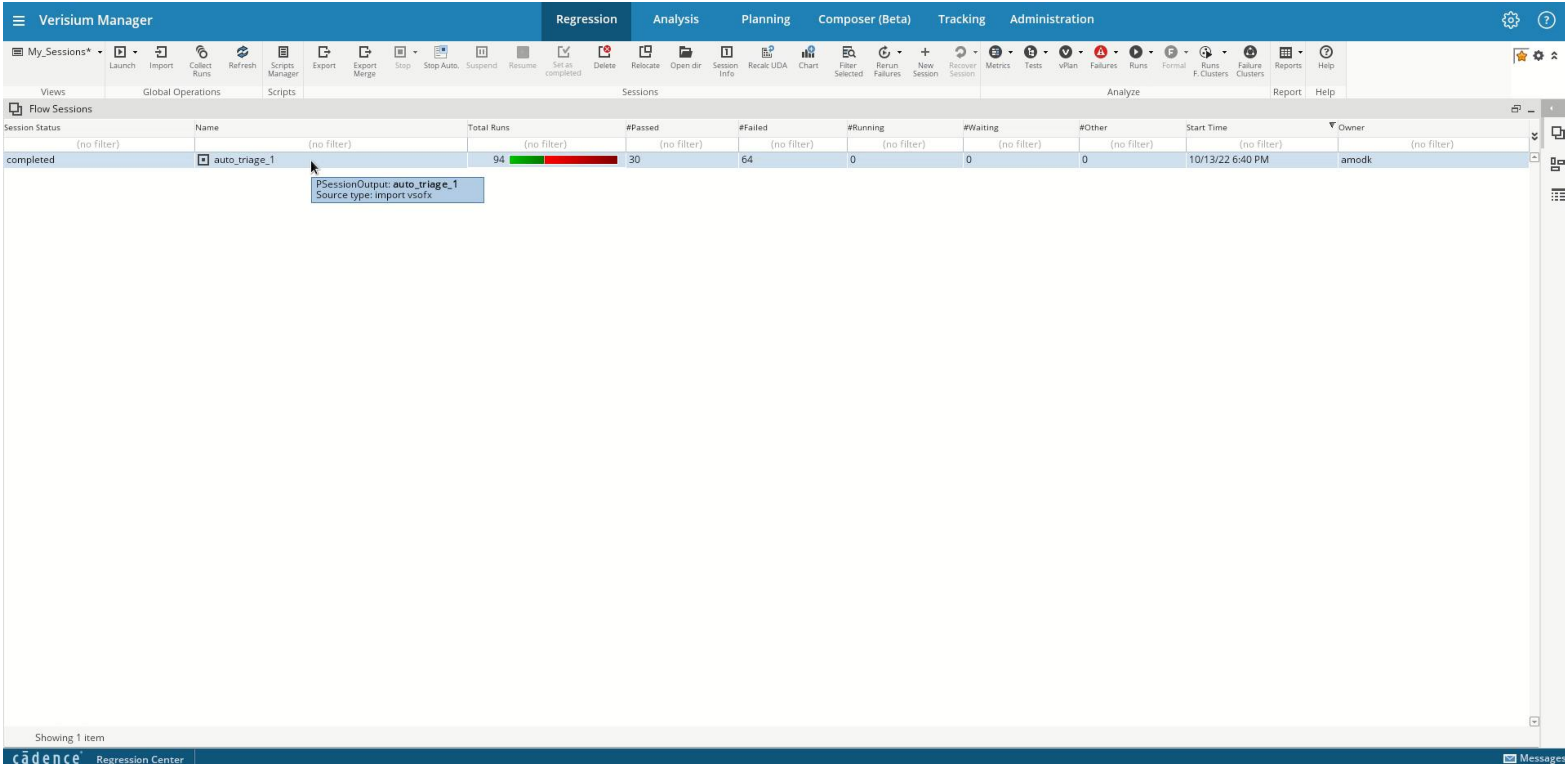
Proposal Threshold: 50

First Failure Description	Failure Cluster Proposal Assurance	Fail
(no filter)	(no filter)	
####### FAIL : uart0 RECEIVED WRONG DATA	90.26%	vM
####### FAIL : uart0 RECEIVED WRONG DATA	90.26%	vM
####### FAIL : uart0 RECEIVED WRONG DATA	90.26%	vM
####### FAIL : uart0 RECEIVED WRONG DATA	90.26%	vM
####### FAIL : uart0 RECEIVED WRONG DATA	90.26%	vM
####### FAIL : APB RECEIVED WRONG DATA 'h4c from uart0	99.57%	vM
####### FAIL : APB RECEIVED WRONG DATA 'h86 from uart0	99.57%	vM
####### FAIL : APB RECEIVED WRONG DATA 'h12 from uart0	99.58%	vM
####### FAIL : APB RECEIVED WRONG DATA 'hc2 from uart0	99.57%	vM
####### FAIL : APB RECEIVED WRONG DATA 'hb7 from uart0	99.57%	vM
####### FAIL : APB RECEIVED WRONG DATA 'h3c from uart0	99.57%	vM
####### FAIL : APB RECEIVED WRONG DATA 'h13 from uart0	99.57%	vM
####### FAIL : APB RECEIVED WRONG DATA 'h7b from uart0	99.57%	vM
####### FAIL : APB RECEIVED WRONG DATA 'h93 from uart0	99.57%	vM
####### FAIL : APB RECEIVED WRONG DATA 'hfb from uart0	99.57%	vM
####### FAIL : uart0 RECEIVED WRONG DATA	90.26%	vM
####### FAIL : APB RECEIVED WRONG DATA 'h99 from uart0	99.57%	vM
####### FAIL : APB RECEIVED WRONG DATA 'hcc from uart0	99.57%	vM
####### FAIL : APB RECEIVED WRONG DATA 'h2e from uart0	99.57%	vM
####### FAIL : APB RECEIVED WRONG DATA 'h84 from uart0	99.57%	vM

The configurable threshold for proposals

New attribute show the percentage of the assurance level we have for the cluster proposal

VERISIUM AUTOTRIAGE DEMO



Session Status	Name	Total Runs	#Passed	#Failed	#Running	#Waiting	#Other	Start Time	Owner
(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)
completed	auto_triage_1	94	30	64	0	0	0	10/13/22 6:40 PM	amodk

PSessionOutput: **auto_triage_1**
Source type: import vsofx

Showing 1 item

Verisium SemanticDiff

SemanticDiff – Meaningful Diff Analysis

- An advanced AI Driven RTL design comparison tool
  - Handles both DUT and TB
- Compares two snapshot versions of the same RTL design
  - Determines the meaningful semantic differences between them
- Generates diff metrics for the analysis
  - CSV and summary log, useful for postprocessing
  - Module and design-hierarchy metrics
- Analysis completes very fast
  - in 30% of the time it takes to compile + elaborate entire design

SemanticDiff – Meaningful Diff Analysis

Identify and rank semantic changes between two RTL versions

Ignore harmless changes

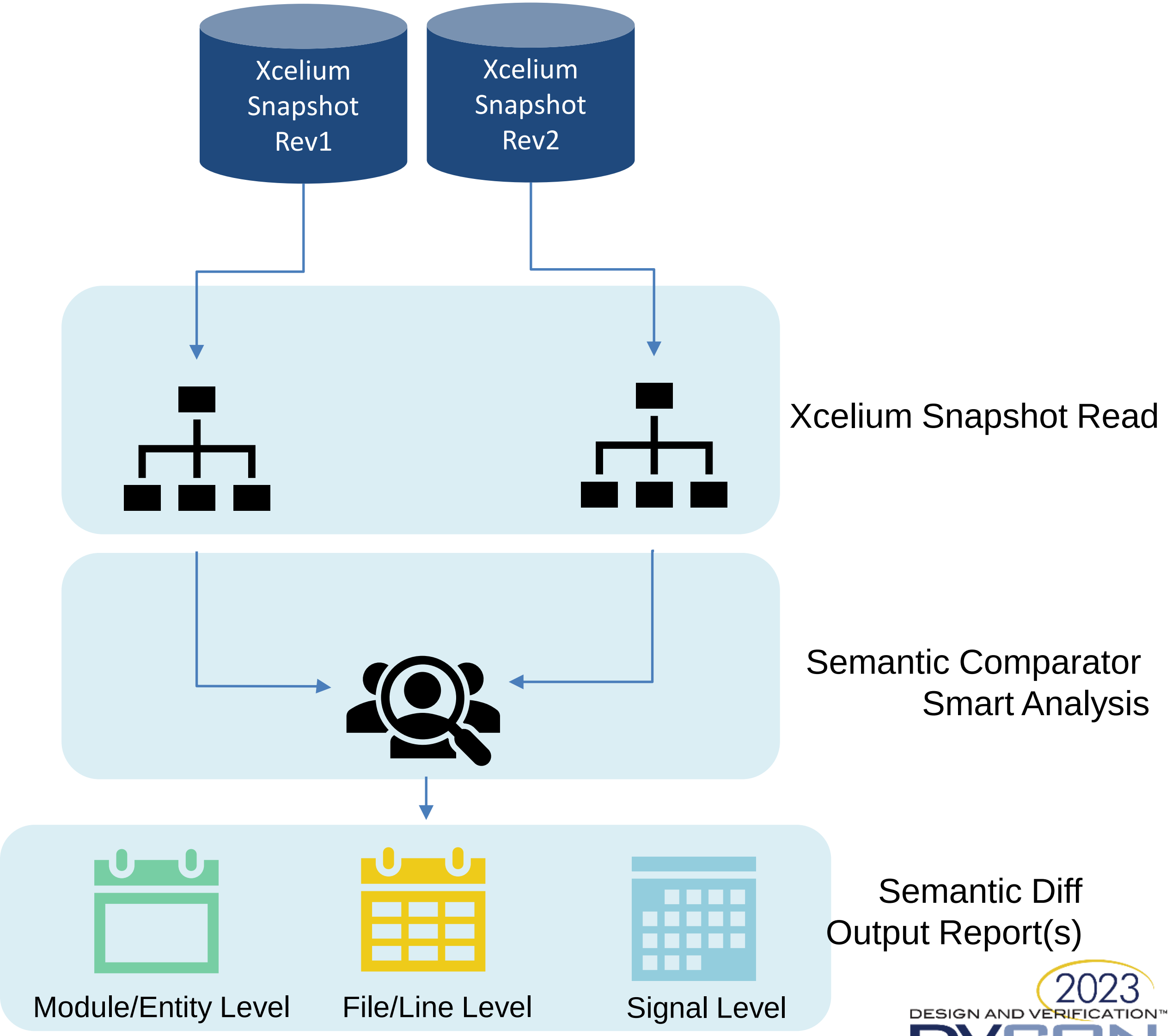
Rank “complexity” of genuine logic changes

```
module cg (d, clk);
  input d, clk;
  reg orig;
  reg clone;
  reg g_latch;
  wire w = orig ^ d;
  wire gclk = clk & g_latch;

  always @(clk or w)
     if (~clk) g_latch <= w;

  always @(posedge gclk)  clone <= d;

  always @(posedge clone) orig <= d;

  fd : assert property (
         @(posedge clk) orig == clone
  );
Endmodule
```

?

=

```
module cg (d, clk);
  input d, clk;
  reg orig, clone, g_latch;

  // Comments …
  wire w = orig ^ d;
  wire gclk = clk & g_latch;

  always @(clk or w)
     if (clk) g_latch <= w;

  always @(posedge gclk)
       clone <= d;

  always @(posedge clone)
       orig <= d;

  fd : assert property (
         @(posedge clk) orig == clone
  );
endmodule
```



Xcelium Snapshot Rev1

Xcelium Snapshot Rev2

Xcelium Snapshot Read

Semantic Comparator Smart Analysis

Semantic Diff Output Report(s)

Module/Entity Level

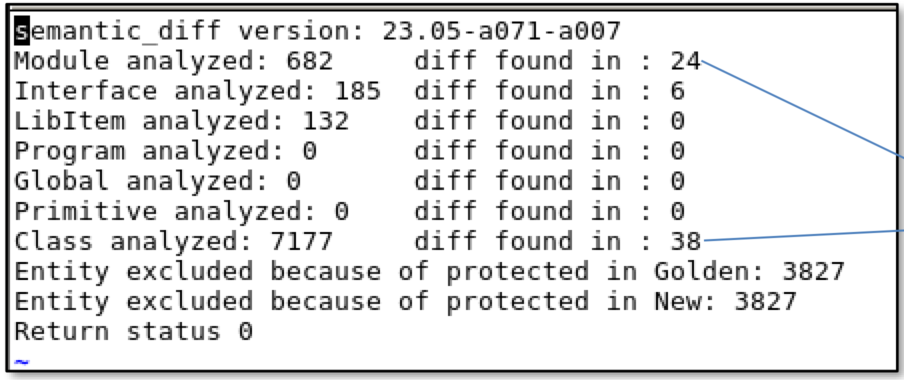
File/Line Level

Signal Level

Launching SemanticDiff

```
verisium –semanticdiff
–xmlibdirpath_golden <path of reference snapshot>
-xmlibdirpath_new <path of the new snapshot>
<other user configurable options>
```

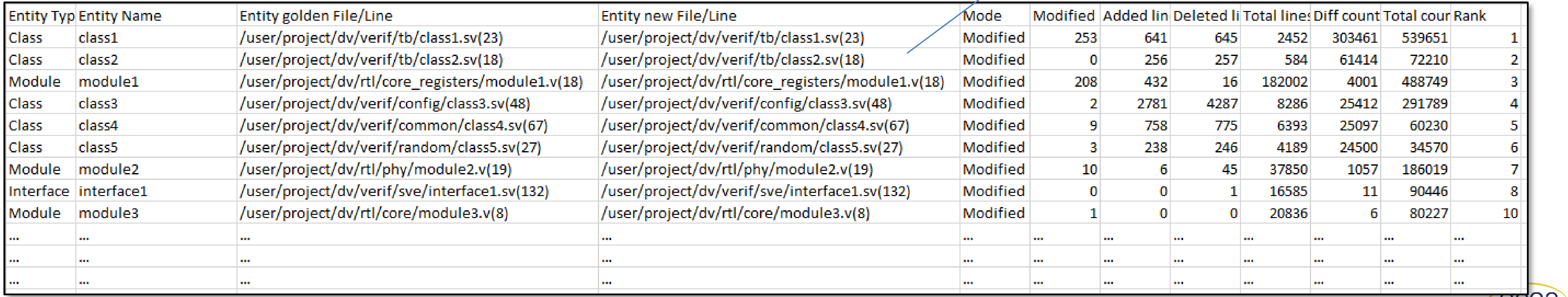
Verisium SemanticDiff Results



```
semantic_diff version: 23.05-a071-a007
Module analyzed: 682     diff found in : 24
Interface analyzed: 185  diff found in : 6
LibItem analyzed: 132    diff found in : 0
Program analyzed: 0      diff found in : 0
Global analyzed: 0       diff found in : 0
Primitive analyzed: 0    diff found in : 0
Class analyzed: 7177     diff found in : 38
Entity excluded because of protected in Golden: 3827
Entity excluded because of protected in New: 3827
Return status 0
~
```

The summary log shows how many entities we analyzed and where we found semantic differences

The detailed csv report gives individual statistics about each file where semantic differences were found



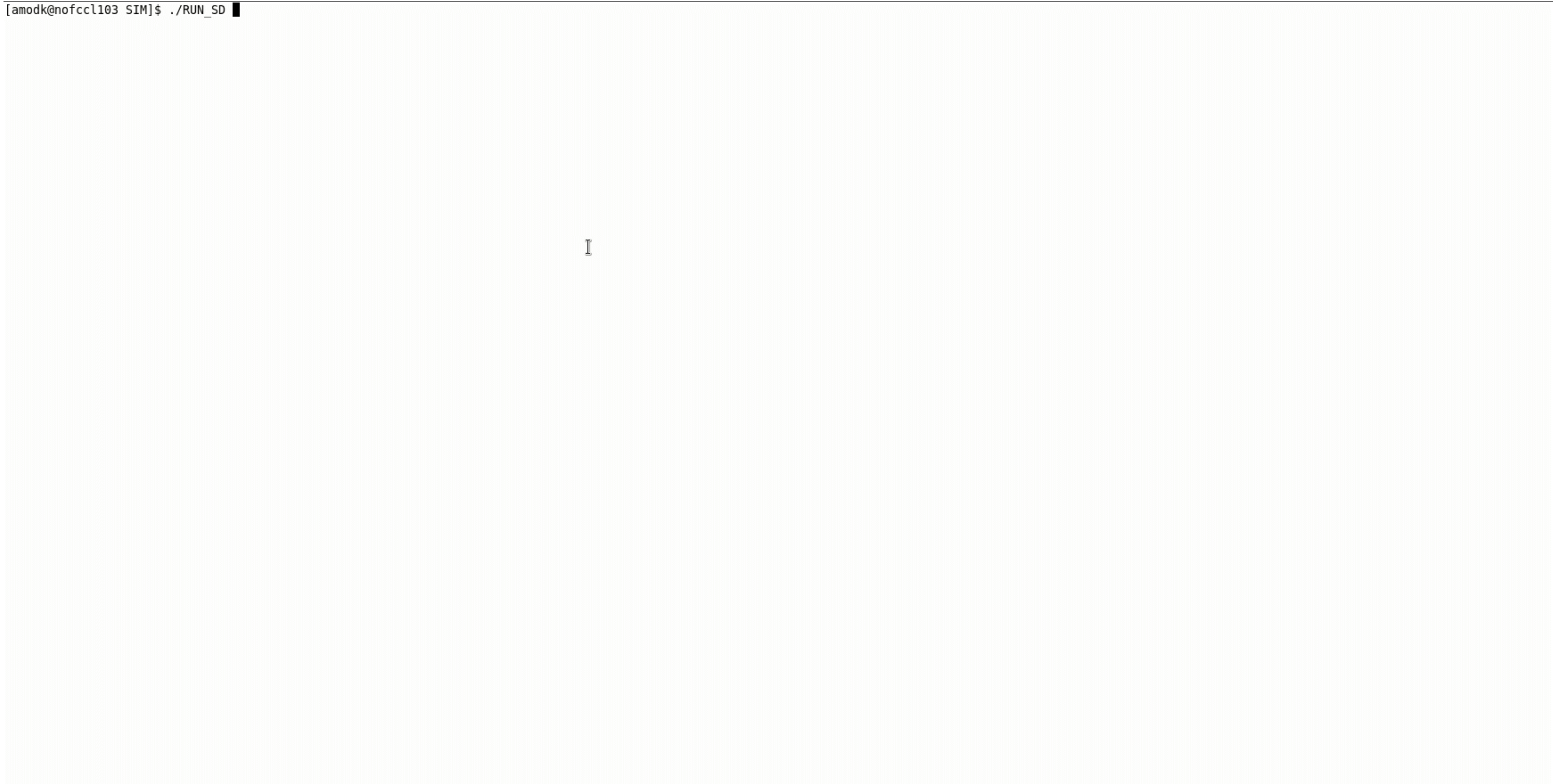
Entity Typ	Entity Name	Entity golden File/Line	Entity new File/Line	Mode	Modified	Added lin	Deleted li	Total lines	Diff count	Total cour	Rank
Class	class1	/user/project/dv/verif/tb/class1.sv(23)	/user/project/dv/verif/tb/class1.sv(23)	Modified	253	641	645	2452	303461	539651	1
Class	class2	/user/project/dv/verif/tb/class2.sv(18)	/user/project/dv/verif/tb/class2.sv(18)	Modified	0	256	257	584	61414	72210	2
Module	module1	/user/project/dv/rtl/core_registers/module1.v(18)	/user/project/dv/rtl/core_registers/module1.v(18)	Modified	208	432	16	182002	4001	488749	3
Class	class3	/user/project/dv/verif/config/class3.sv(48)	/user/project/dv/verif/config/class3.sv(48)	Modified	2	2781	4287	8286	25412	291789	4
Class	class4	/user/project/dv/verif/common/class4.sv(67)	/user/project/dv/verif/common/class4.sv(67)	Modified	9	758	775	6393	25097	60230	5
Class	class5	/user/project/dv/verif/random/class5.sv(27)	/user/project/dv/verif/random/class5.sv(27)	Modified	3	238	246	4189	24500	34570	6
Module	module2	/user/project/dv/rtl/phy/module2.v(19)	/user/project/dv/rtl/phy/module2.v(19)	Modified	10	6	45	37850	1057	186019	7
Interface	interface1	/user/project/dv/verif/sve/interface1.sv(132)	/user/project/dv/verif/sve/interface1.sv(132)	Modified	0	0	1	16585	11	90446	8
Module	module3	/user/project/dv/rtl/core/module3.v(8)	/user/project/dv/rtl/core/module3.v(8)	Modified	1	0	0	20836	6	80227	10
...	...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...	...

Verisium SemanticDiff Performance

DESIGN	Compile + Elaboration Time(sec)	SemanticDiff Analysis Time(sec)
Design 1	921	220
Design 2	2596	716
Design 3	8558	2901
Design 4	14480	3120

- The numbers here represent SemanticDiff Analysis on the entire snapshot
- Semanticdiff completes in 20-30% of compile+elab time.
- User has flexibility to run SemanticDiff on portion of the snapshot for faster turnaround time

VERISIUM SEMANTICDIFF DEMO



```
[amodk@nofccl103 SIM]$ ./RUN_SD
```

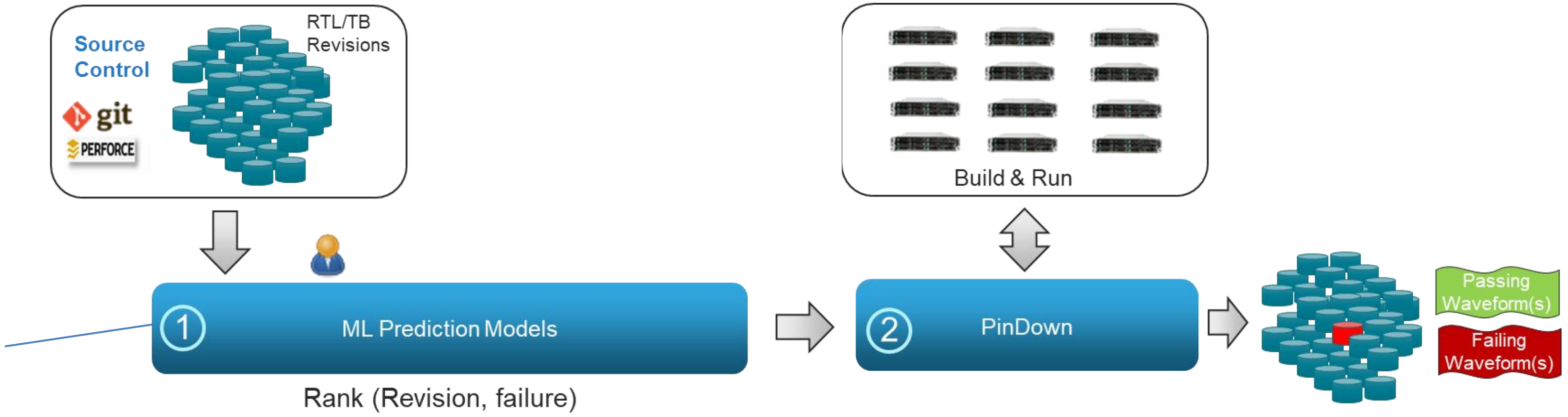
Verisium PinDown

PinDown - ML Based Search

ML Risk Prediction is based on:

- code complexity, commit info, design/tb structure, revision history

100+ features implemented



“Baseline" revision

“Failing" revision

	✓														✗
Revision#	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122
Risk Prediction	0.05	0.14	0.1	0.1	0.02	0.25	0.07	0.83	0.1	0.01	0.01	0.13	0.38	0.01	0.22

Validation Step

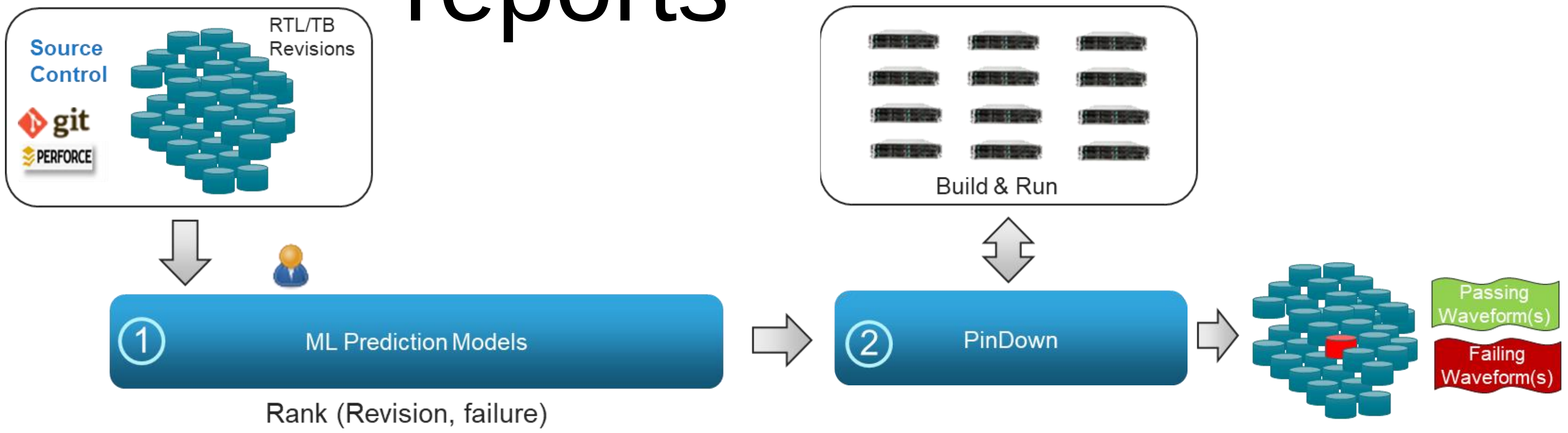
								✗							✓
Revision#	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122

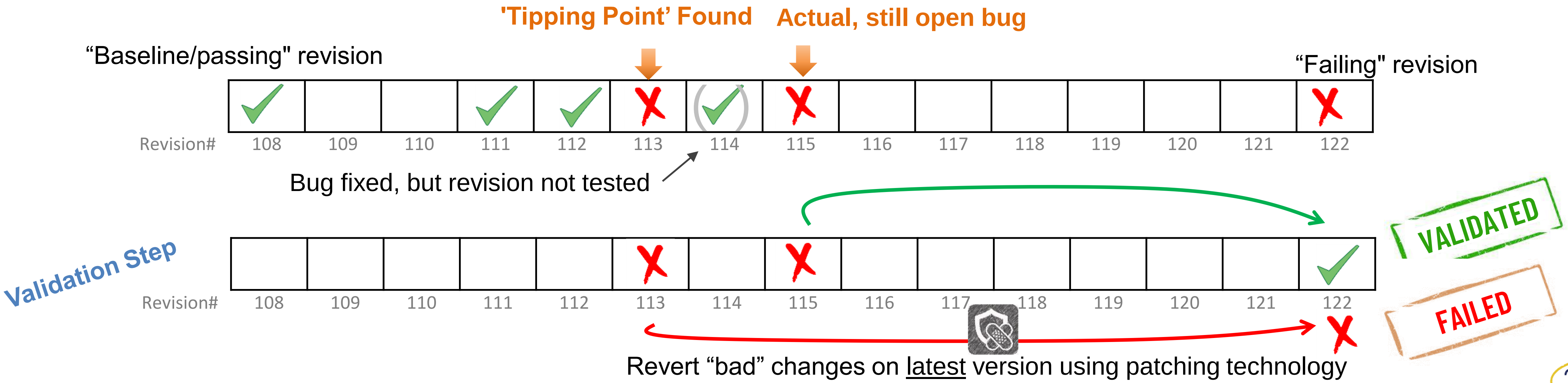
VALIDATED

Revert “bad” changes on latest version using patching technology

One single recompile/rerun of the failing test to validate it is passing without bad commit

PinDown: Validation to ensure correct bug reports





Revision#	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122
	✓			✓	✓	✗	(✓)	✗							✗

Revision#	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122
						✗		✗							✓

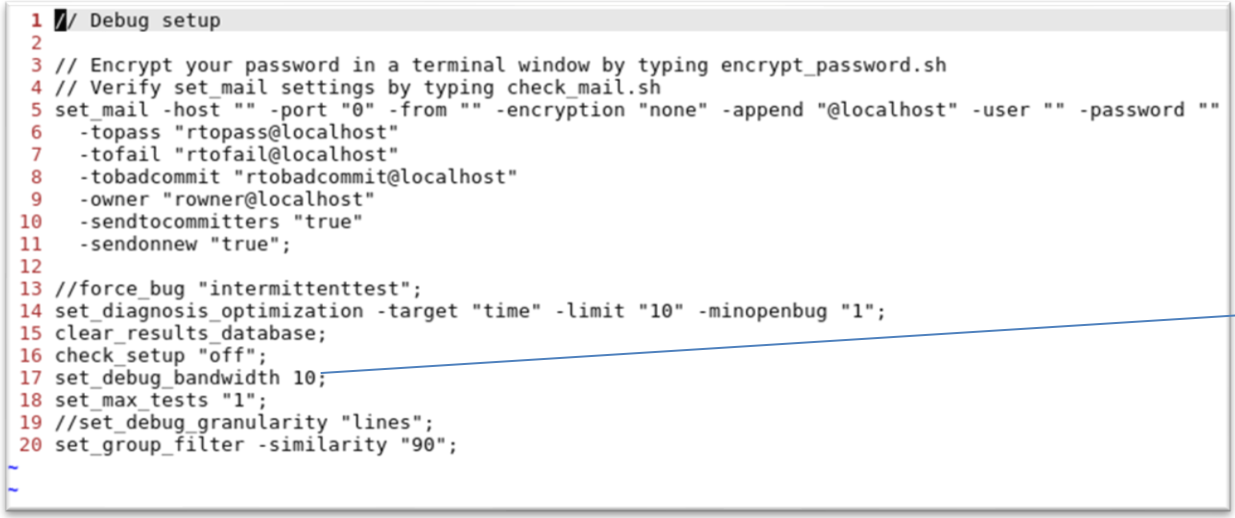
Revert "bad" changes on latest version using patching technology

Running PinDown

```
verisium –pindown
-config <path to pindown_config file >
--session_name <session to run PinDown analysis>
--vmanager_server <server on which PinDown should run analysis>
```

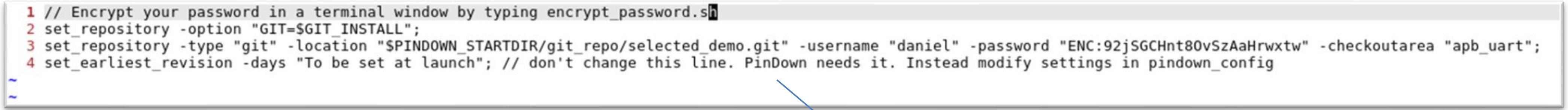
- Config file configures the PinDown Debug Analysis
    - Defines debug options for PinDown
    - Provides version control information
    - Defines how long should PinDown Debug run

Revision Control and Debug Options



```
// Debug setup

// Encrypt your password in a terminal window by typing encrypt_password.sh
// Verify set_mail settings by typing check_mail.sh
set_mail -host "" -port "0" -from "" -encryption "none" -append "@localhost" -user "" -password ""
  -topass "rtopass@localhost"
  -tofail "rtofail@localhost"
  -tobadcommit "rtobadcommit@localhost"
  -owner "rowner@localhost"
  -sendtocommitters "true"
  -sendonnew "true";

//force_bug "intermittenttest";
set_diagnosis_optimization -target "time" -limit "10" -minopenbug "1";
clear_results_database;
check_setup "off";
set_debug_bandwidth 10;
set_max_tests "1";
//set_debug_granularity "lines";
set_group_filter -similarity "90";
~
~
```

Runs that PinDown can run in parallel

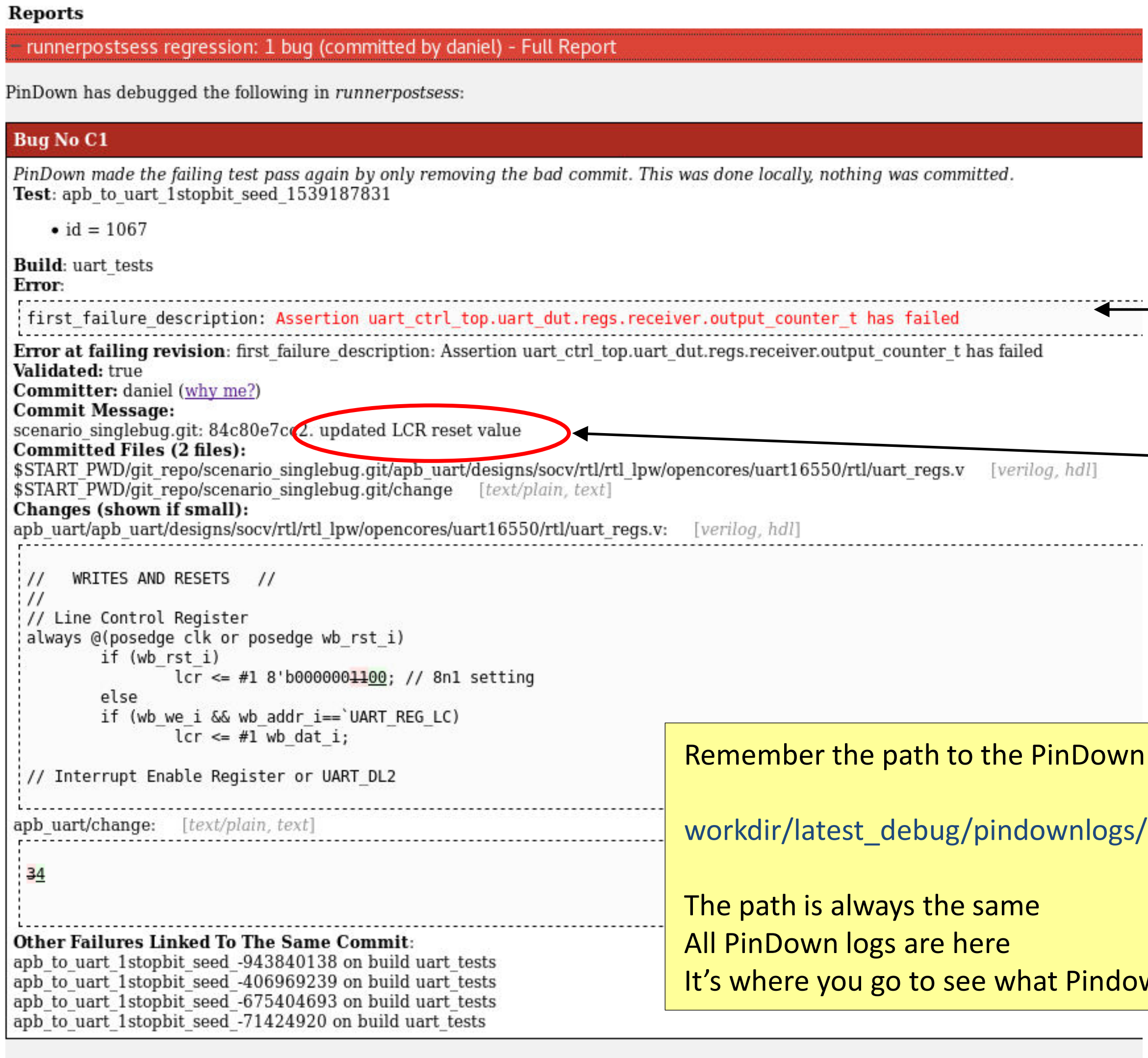


```
// Encrypt your password in a terminal window by typing encrypt_password.sh
set_repository -option "GIT=$GIT_INSTALL";
set_repository -type "git" -location "$PINDOWN_STARTDIR/git_repo/selected_demo.git" -username "daniel" -password "ENC:92jSGCHnt8OvSzAaHrwxtw" -checkoutarea "apb_uart";
set_earliest_revision -days "To be set at launch"; // don't change this line. PinDown needs it. Instead modify settings in pindown_config
~
~
```

Revision Control information that PinDown uses for debug.

PinDown Bug Report in Demo

Open workdir/latest_debug/pindownlogs/text/pindown_debug.html and go to ”Reports” section:



Reports

− runnerpostsess regression: 1 bug (committed by daniel) - Full Report

PinDown has debugged the following in *runnerpostsess*:

Bug No C1

PinDown made the failing test pass again by only removing the bad commit. This was done locally, nothing was committed.
Test: apb_to_uart_1stopbit_seed_1539187831

- id = 1067

Build: uart_tests
Error:

```
first_failure_description: Assertion uart_ctrl_top.uart_dut.regs.receiver.output_counter_t has failed
```

Error at failing revision: first_failure_description: Assertion uart_ctrl_top.uart_dut.regs.receiver.output_counter_t has failed
Validated: true
Committer: daniel (why me?)
Commit Message:
scenario_singlebug.git: 84c80e7cc2. updated LCR reset value
Committed Files (2 files):
$START_PWD/git_repo/scenario_singlebug.git/apb_uart/designs/socv/rtl/rtl_lpw/opencores/uart16550/rtl/uart_regs.v [*verilog, hdl*]
$START_PWD/git_repo/scenario_singlebug.git/change [*text/plain, text*]
Changes (shown if small):
apb_uart/apb_uart/designs/socv/rtl/rtl_lpw/opencores/uart16550/rtl/uart_regs.v: [*verilog, hdl*]

```
//   WRITES AND RESETS   //
//
// Line Control Register
always @(posedge clk or posedge wb_rst_i)
        if (wb_rst_i)
                lcr <= #1 8'b000000~~11~~00; // 8n1 setting
        else
        if (wb_we_i && wb_addr_i==`UART_REG_LC)
                lcr <= #1 wb_dat_i;

// Interrupt Enable Register or UART_DL2
```

apb_uart/change: [*text/plain, text*]

```
~~3~~4
```

Other Failures Linked To The Same Commit:
apb_to_uart_1stopbit_seed_-943840138 on build uart_tests
apb_to_uart_1stopbit_seed_-406969239 on build uart_tests
apb_to_uart_1stopbit_seed_-675404693 on build uart_tests
apb_to_uart_1stopbit_seed_-71424920 on build uart_tests

Error message

Commit message for bad commit

Remember the path to the PinDown logs:

workdir/latest_debug/pindownlogs/text

The path is always the same
All PinDown logs are here
It’s where you go to see what Pindown did

pindown.log:

```
Debug result accepted:
                Bug ID: C1
                Run ID: 1067
                Debug Status: pilot_validated_bug

                Committer: daniel

Generating waveminer results for bug: C1 (new bug)
waveminer results for bug C1 (new bug) are ready under run
directory:
vmrunner_pindownprerunscript/manual_checkout/apb_uart/apb_
uart/my_sessions/uart_ctrl/uvm_regression_25_10_22_14_06_2
9/chain_0/uart_tests/run_3/debug

Debug is completed.
```

For the WaveMiner analysis of signals go the folder shown at the end of pindown.log. Here you find 2 important files: waveminer_top_signals and show_waveminer

waveminer_top_signals:
1. uart_ctrl_top.uart_dut.regs.lcr
2. uart_ctrl_top.uart_dut.regs.block_value
3. uart_ctrl_top.uart_dut.regs.counter_t
4. uart_ctrl_top.uart_dut.regs.block_cnt

Ranked most problematic signal

show_waveminer:
Run this script to show the wave forms in WaveMiner

Customer Use-Case: Bug Report in 1 iteration in 51% of cases

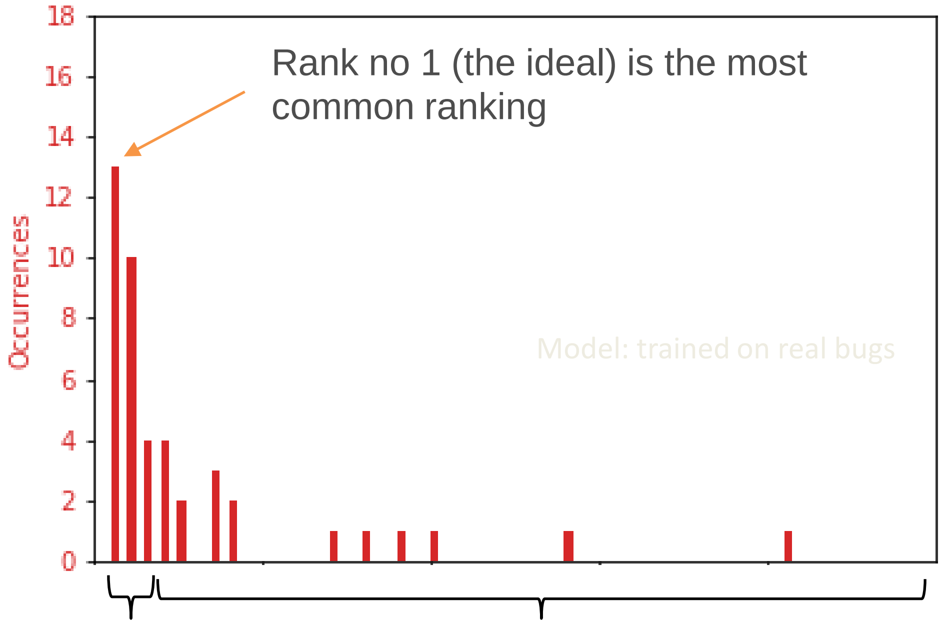
PinDown Accuracy

- Due to Validation
- Each Bug Prediction is Validated
- If you want to automate blame, you better be right

PinDown Efficiency

- Due to ML-based Bug Prediction
- Saves slots on the farm
- Bug reports issued faster

Bug prediction ranking for 53 validated bugs June 5th to Sep 27th 2020



Rank no 1 (the ideal) is the most common ranking

1 iteration
(rank 1-3)

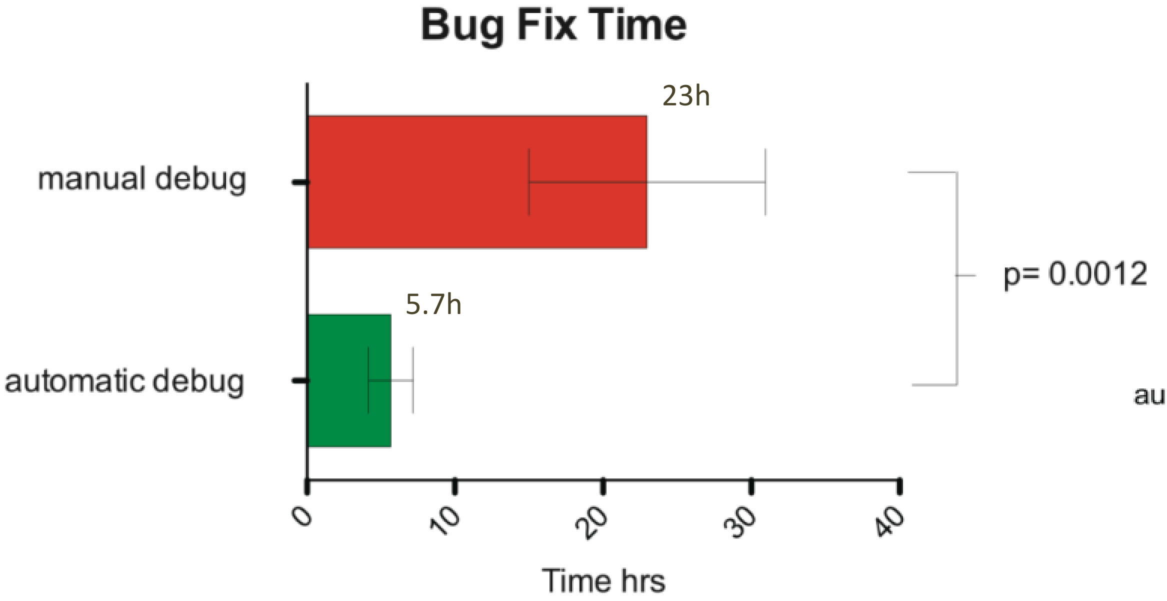
51%

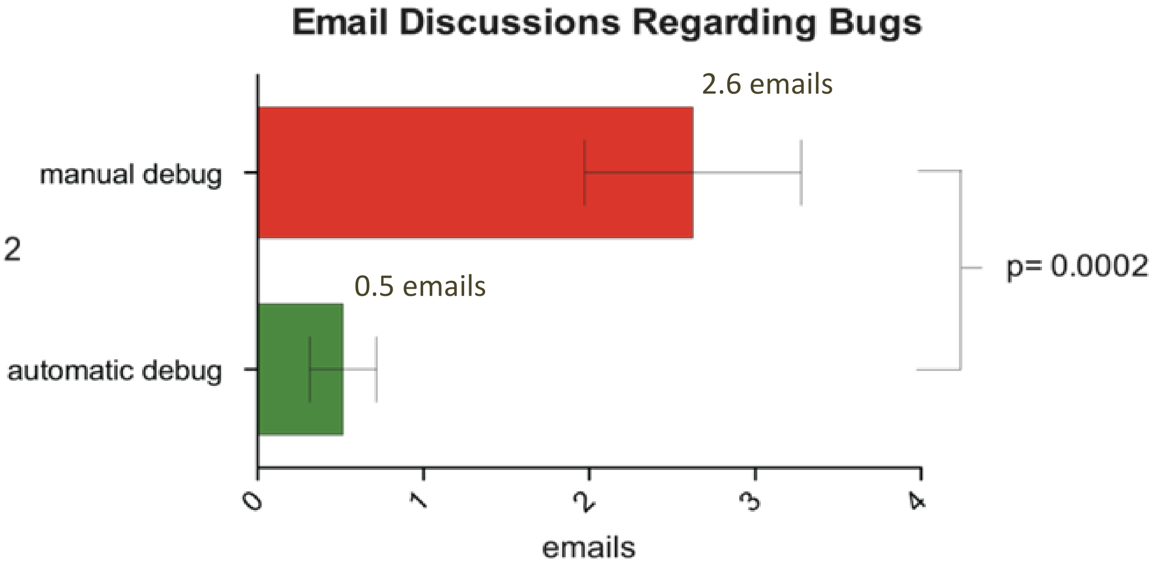
👍

2-7 iterations
(rank 4 or more)

Customer Use-Case:
Bugs Fixed 4x Faster, 5x Less Discussion

Project Details
- ASIC IP Project (Microprocessor)
- About 40 people (ASIC designers plus DV engineers)
- Multi-site
- Measured over 3 months



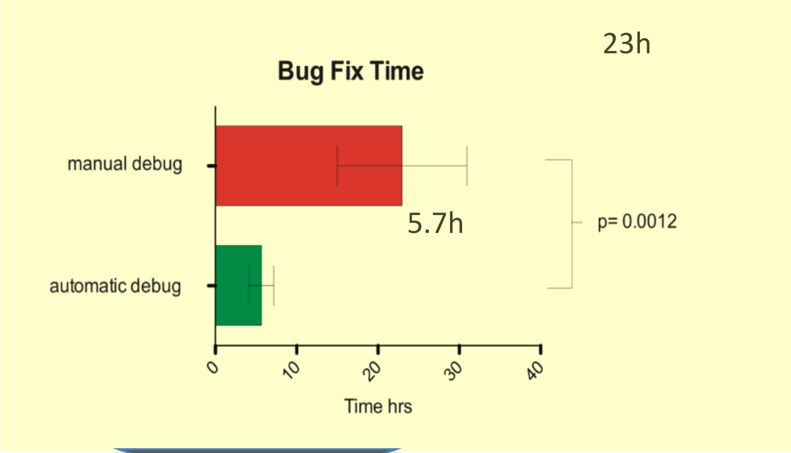


Measured Bug Fix Time
- from the time the bug was reported
- to the time the fix was submitted to the revision control system

There was more discussion about who and what needs to be fixed when there was just an error message vs. a PinDown bug report

Customer Use-Case: 11% reduced project time

Time to correct bugs 75% shorter (4x) with PinDown...



...has a direct impact on the total project leadtime

Time saving per bug (17.3h/bug) x Number of bugs (39/quarter) x Degree of blocking (35%)

Time saving per bug (17.3h/bug)	x	Number of bugs (39/quarter)	x	Degree of blocking (35%)
measured		measured		estimated

Measured project time: 3 months = 2184 hours

Freed up Verification Lead

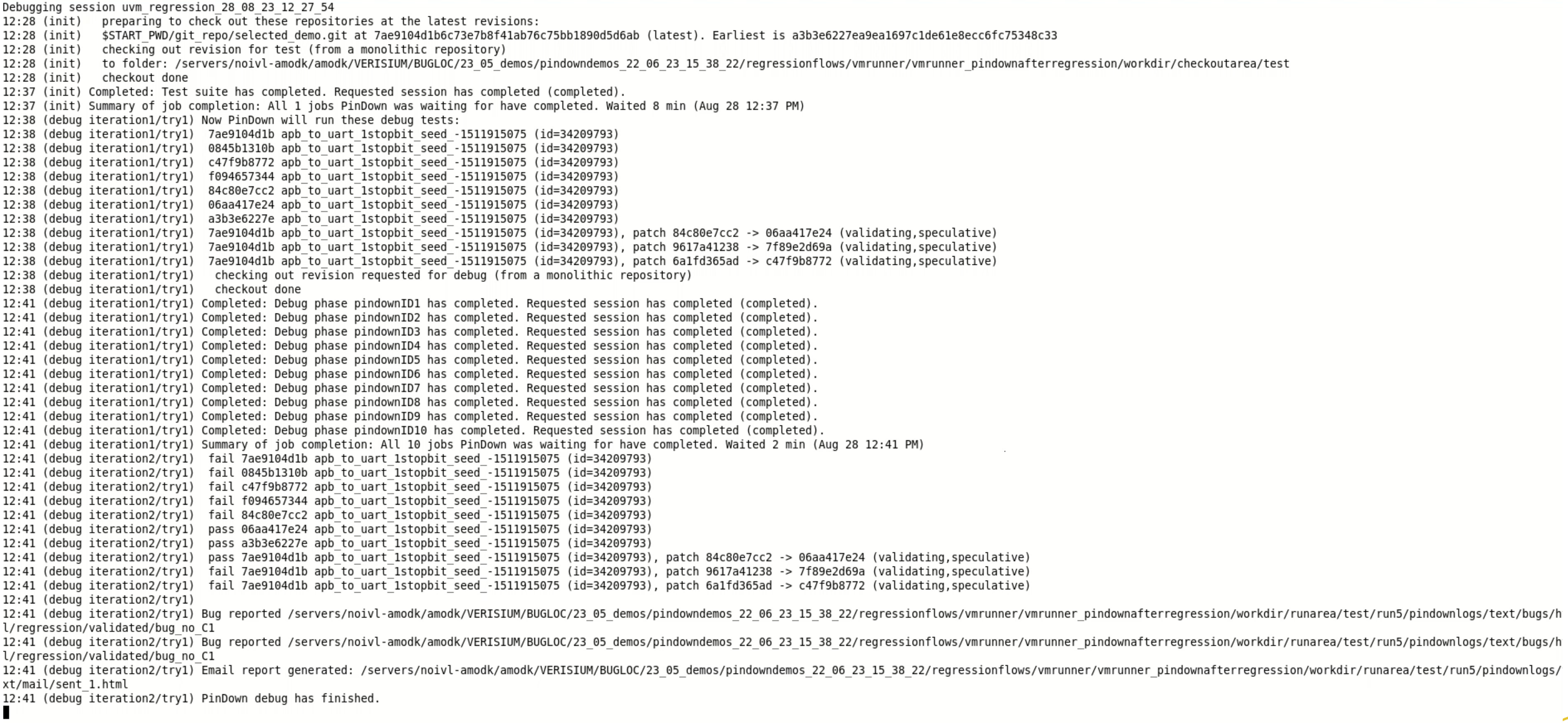
- PinDown took over the job of chasing down engineers to fix issues
- Half of the verification lead's time was saved (0.5 engineer years)

+

11% shorter total project lead time!

- Faster time-to-market
- Major cost savings: 4.4 engineer years (40 engineers in project)

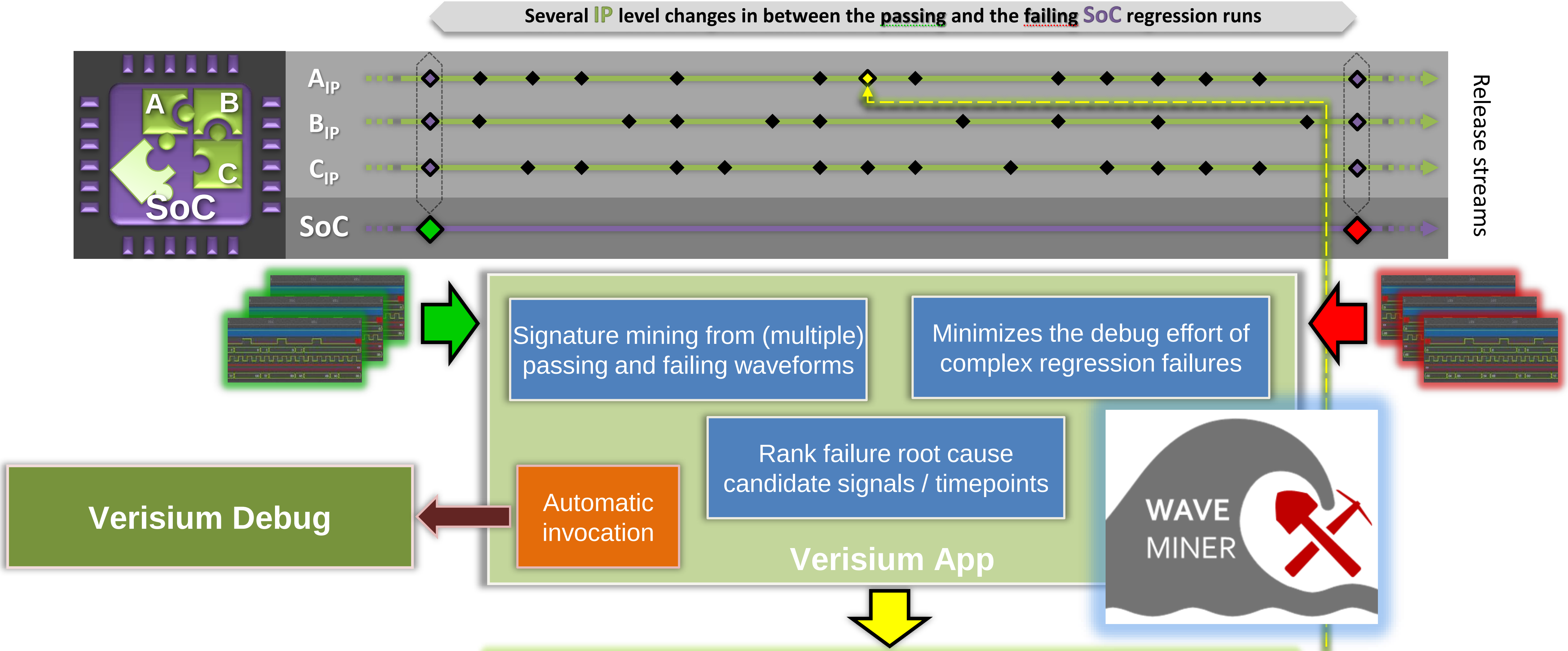
Verisium PinDown Demo



```
Debugging session uvm_regression_28_08_23_12_27_54
12:28 (init)   preparing to check out these repositories at the latest revisions:
12:28 (init)   $START_PWD/git_repo/selected_demo.git at 7ae9104d1b6c73e7b8f41ab76c75bb1890d5d6ab (latest). Earliest is a3b3e6227ea9ea1697c1de61e8ecc6fc75348c33
12:28 (init)   checking out revision for test (from a monolithic repository)
12:28 (init)   to folder: /servers/noivl-amodk/amodk/VERISIUM/BUGLOC/23_05_demos/pindowndemos_22_06_23_15_38_22/regressionflows/vmrunner/vmrunner_pindownafterregression/workdir/checkoutarea/test
12:28 (init)   checkout done
12:37 (init) Completed: Test suite has completed. Requested session has completed (completed).
12:37 (init) Summary of job completion: All 1 jobs PinDown was waiting for have completed. Waited 8 min (Aug 28 12:37 PM)
12:38 (debug iteration1/try1) Now PinDown will run these debug tests:
12:38 (debug iteration1/try1)  7ae9104d1b apb_to_uart_1stopbit_seed_-1511915075 (id=34209793)
12:38 (debug iteration1/try1)  0845b1310b apb_to_uart_1stopbit_seed_-1511915075 (id=34209793)
12:38 (debug iteration1/try1)  c47f9b8772 apb_to_uart_1stopbit_seed_-1511915075 (id=34209793)
12:38 (debug iteration1/try1)  f094657344 apb_to_uart_1stopbit_seed_-1511915075 (id=34209793)
12:38 (debug iteration1/try1)  84c80e7cc2 apb_to_uart_1stopbit_seed_-1511915075 (id=34209793)
12:38 (debug iteration1/try1)  06aa417e24 apb_to_uart_1stopbit_seed_-1511915075 (id=34209793)
12:38 (debug iteration1/try1)  a3b3e6227e apb_to_uart_1stopbit_seed_-1511915075 (id=34209793)
12:38 (debug iteration1/try1)  7ae9104d1b apb_to_uart_1stopbit_seed_-1511915075 (id=34209793), patch 84c80e7cc2 -> 06aa417e24 (validating,speculative)
12:38 (debug iteration1/try1)  7ae9104d1b apb_to_uart_1stopbit_seed_-1511915075 (id=34209793), patch 9617a41238 -> 7f89e2d69a (validating,speculative)
12:38 (debug iteration1/try1)  7ae9104d1b apb_to_uart_1stopbit_seed_-1511915075 (id=34209793), patch 6a1fd365ad -> c47f9b8772 (validating,speculative)
12:38 (debug iteration1/try1)   checking out revision requested for debug (from a monolithic repository)
12:38 (debug iteration1/try1)   checkout done
12:41 (debug iteration1/try1) Completed: Debug phase pindownID1 has completed. Requested session has completed (completed).
12:41 (debug iteration1/try1) Completed: Debug phase pindownID2 has completed. Requested session has completed (completed).
12:41 (debug iteration1/try1) Completed: Debug phase pindownID3 has completed. Requested session has completed (completed).
12:41 (debug iteration1/try1) Completed: Debug phase pindownID4 has completed. Requested session has completed (completed).
12:41 (debug iteration1/try1) Completed: Debug phase pindownID5 has completed. Requested session has completed (completed).
12:41 (debug iteration1/try1) Completed: Debug phase pindownID6 has completed. Requested session has completed (completed).
12:41 (debug iteration1/try1) Completed: Debug phase pindownID7 has completed. Requested session has completed (completed).
12:41 (debug iteration1/try1) Completed: Debug phase pindownID8 has completed. Requested session has completed (completed).
12:41 (debug iteration1/try1) Completed: Debug phase pindownID9 has completed. Requested session has completed (completed).
12:41 (debug iteration1/try1) Completed: Debug phase pindownID10 has completed. Requested session has completed (completed).
12:41 (debug iteration1/try1) Summary of job completion: All 10 jobs PinDown was waiting for have completed. Waited 2 min (Aug 28 12:41 PM)
12:41 (debug iteration2/try1)  fail 7ae9104d1b apb_to_uart_1stopbit_seed_-1511915075 (id=34209793)
12:41 (debug iteration2/try1)  fail 0845b1310b apb_to_uart_1stopbit_seed_-1511915075 (id=34209793)
12:41 (debug iteration2/try1)  fail c47f9b8772 apb_to_uart_1stopbit_seed_-1511915075 (id=34209793)
12:41 (debug iteration2/try1)  fail f094657344 apb_to_uart_1stopbit_seed_-1511915075 (id=34209793)
12:41 (debug iteration2/try1)  fail 84c80e7cc2 apb_to_uart_1stopbit_seed_-1511915075 (id=34209793)
12:41 (debug iteration2/try1)  pass 06aa417e24 apb_to_uart_1stopbit_seed_-1511915075 (id=34209793)
12:41 (debug iteration2/try1)  pass a3b3e6227e apb_to_uart_1stopbit_seed_-1511915075 (id=34209793)
12:41 (debug iteration2/try1)  pass 7ae9104d1b apb_to_uart_1stopbit_seed_-1511915075 (id=34209793), patch 84c80e7cc2 -> 06aa417e24 (validating,speculative)
12:41 (debug iteration2/try1)  fail 7ae9104d1b apb_to_uart_1stopbit_seed_-1511915075 (id=34209793), patch 9617a41238 -> 7f89e2d69a (validating,speculative)
12:41 (debug iteration2/try1)  fail 7ae9104d1b apb_to_uart_1stopbit_seed_-1511915075 (id=34209793), patch 6a1fd365ad -> c47f9b8772 (validating,speculative)
12:41 (debug iteration2/try1)
12:41 (debug iteration2/try1) Bug reported /servers/noivl-amodk/amodk/VERISIUM/BUGLOC/23_05_demos/pindowndemos_22_06_23_15_38_22/regressionflows/vmrunner/vmrunner_pindownafterregression/workdir/runarea/test/run5/pindownlogs/text/bugs/h
l/regression/validated/bug_no_C1
12:41 (debug iteration2/try1) Bug reported /servers/noivl-amodk/amodk/VERISIUM/BUGLOC/23_05_demos/pindowndemos_22_06_23_15_38_22/regressionflows/vmrunner/vmrunner_pindownafterregression/workdir/runarea/test/run5/pindownlogs/text/bugs/h
l/regression/validated/bug_no_C1
12:41 (debug iteration2/try1) Email report generated: /servers/noivl-amodk/amodk/VERISIUM/BUGLOC/23_05_demos/pindowndemos_22_06_23_15_38_22/regressionflows/vmrunner/vmrunner_pindownafterregression/workdir/runarea/test/run5/pindownlogs/
xt/mail/sent_1.html
12:41 (debug iteration2/try1) PinDown debug has finished.
█
```

Verisium Waveminer

WaveMiner *simplifies* Regression Debug



Several IP level changes in between the passing and the failing SoC regression runs

Signature mining from (multiple) passing and failing waveforms

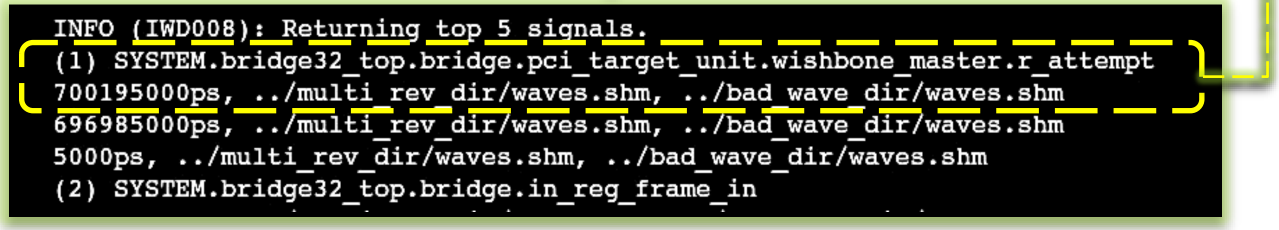
Minimizes the debug effort of complex regression failures

Rank failure root cause candidate signals / timepoints

Automatic invocation

Verisium App

Verisium Debug

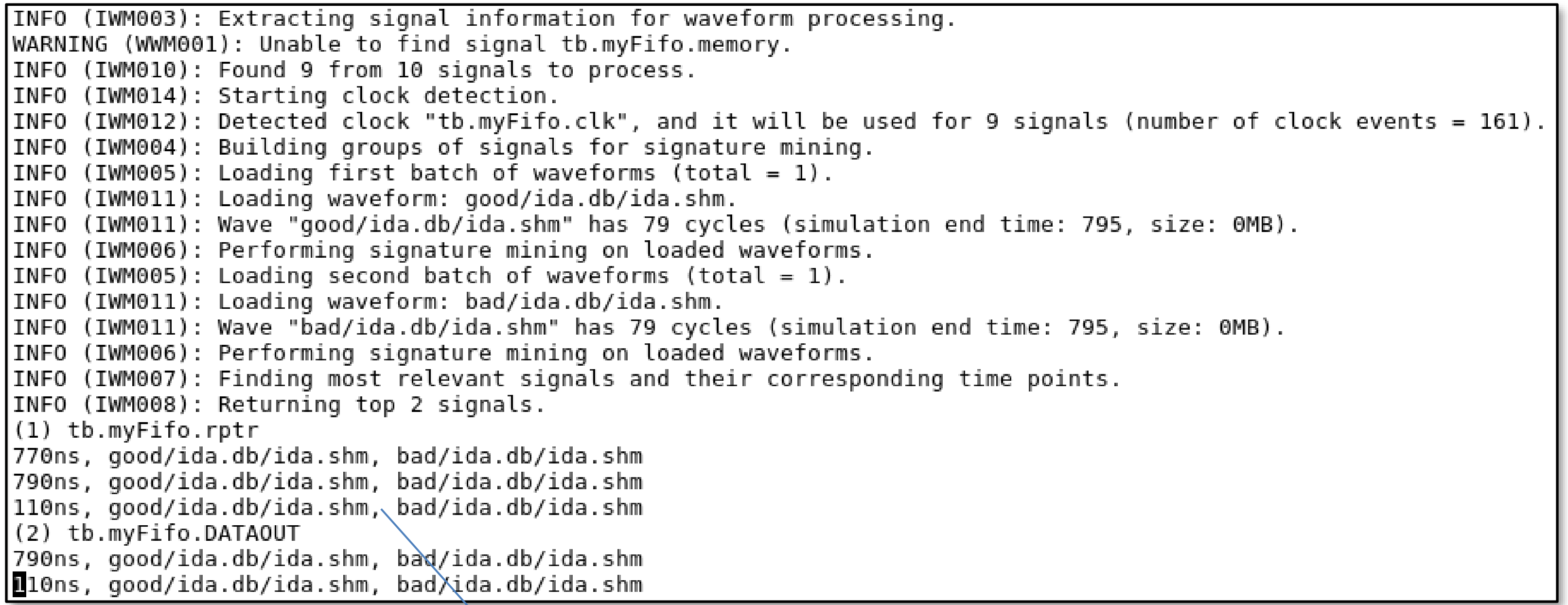


```
INFO (IWD008): Returning top 5 signals.
(1) SYSTEM.bridge32_top.bridge.pci_target_unit.wishbone_master.r_attempt
700195000ps, ../multi_rev_dir/waves.shm, ../bad_wave_dir/waves.shm
696985000ps, ../multi_rev_dir/waves.shm, ../bad_wave_dir/waves.shm
5000ps, ../multi_rev_dir/waves.shm, ../bad_wave_dir/waves.shm
(2) SYSTEM.bridge32_top.bridge.in_reg_frame_in
```

Launching Verisium WaveMiner

```
verisium –waveminer
-wavepath_new <path to the new waveform>
-xmlibdirpath_new <path to the new snapshot>
-wavepath_golden <path to reference waveform>
-xmlibdirpath_golden <path to the reference snapshot>
```

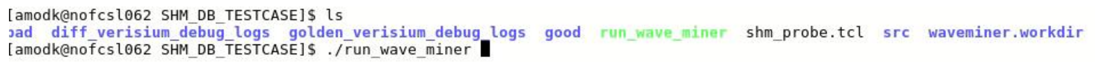
Verisium WaveMiner Report



```
INFO (IWM003): Extracting signal information for waveform processing.
WARNING (WWM001): Unable to find signal tb.myFifo.memory.
INFO (IWM010): Found 9 from 10 signals to process.
INFO (IWM014): Starting clock detection.
INFO (IWM012): Detected clock "tb.myFifo.clk", and it will be used for 9 signals (number of clock events = 161).
INFO (IWM004): Building groups of signals for signature mining.
INFO (IWM005): Loading first batch of waveforms (total = 1).
INFO (IWM011): Loading waveform: good/ida.db/ida.shm.
INFO (IWM011): Wave "good/ida.db/ida.shm" has 79 cycles (simulation end time: 795, size: 0MB).
INFO (IWM006): Performing signature mining on loaded waveforms.
INFO (IWM005): Loading second batch of waveforms (total = 1).
INFO (IWM011): Loading waveform: bad/ida.db/ida.shm.
INFO (IWM011): Wave "bad/ida.db/ida.shm" has 79 cycles (simulation end time: 795, size: 0MB).
INFO (IWM006): Performing signature mining on loaded waveforms.
INFO (IWM007): Finding most relevant signals and their corresponding time points.
INFO (IWM008): Returning top 2 signals.
(1) tb.myFifo.rptr
770ns, good/ida.db/ida.shm, bad/ida.db/ida.shm
790ns, good/ida.db/ida.shm, bad/ida.db/ida.shm
110ns, good/ida.db/ida.shm, bad/ida.db/ida.shm
(2) tb.myFifo.DATAOUT
790ns, good/ida.db/ida.shm, bad/ida.db/ida.shm
110ns, good/ida.db/ida.shm, bad/ida.db/ida.shm
```

WaveMiner generates ranked list of signals based on waveform analysis and ranks the timepoints also to help narrow down the debug and opens Verisium Debug Window for further debugging

Verisium WaveMiner Demo



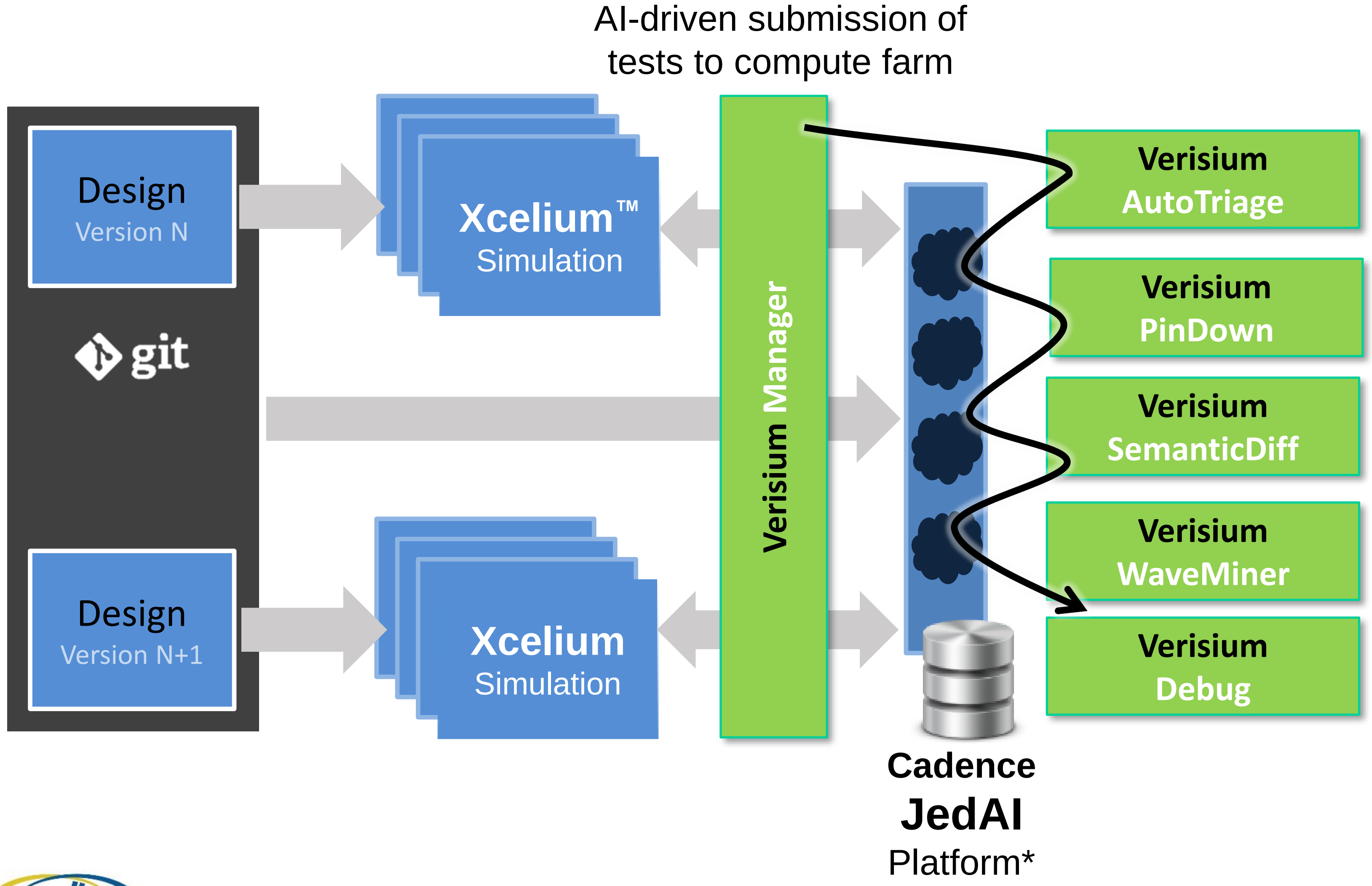
```
[amodk@nofcsl062 SHM_DB_TESTCASE]$ ls
ɔad  diff_verisium_debug_logs  golden_verisium_debug_logs  good  run_wave_miner  shm_probe.tcl  src  waveminer.workdir
[amodk@nofcsl062 SHM_DB_TESTCASE]$ ./run_wave_miner █
```

Verisium WaveMiner Performance

Waveform DB’s	Size	WaveMiner Analysis Time
Waveform DB 1	Golden – 1GB, Diff – 850MB	~ 20 min
Waveform DB 2	Golden – 26 GB , Diff – 25GB	~ 35 min
Waveform DB 3	Golden - 46GB, Diff - 43GB	~ 15 min
Waveform DB 4	Golden – 648MB, Diff – 702MB	~57 min

The analysis time taken for generating results is dependent on multiple factors like size of the waveform, number of signals to analyze and total activity happening on each signal of interest.

Verisium Platform 1.0 in Action



Automatically groups tests failing due to the same underlying bug

Automatically identifies code differences between design versions N and N+1

Analyzes waveforms and automatically identifies root cause of bug (signals + time)

Interactive side-by-side display of passing vs failing tests with bug root cause highlighted

Agenda

What

Challenges & related asks for failures debug

What

Solutions to failures debug related problems

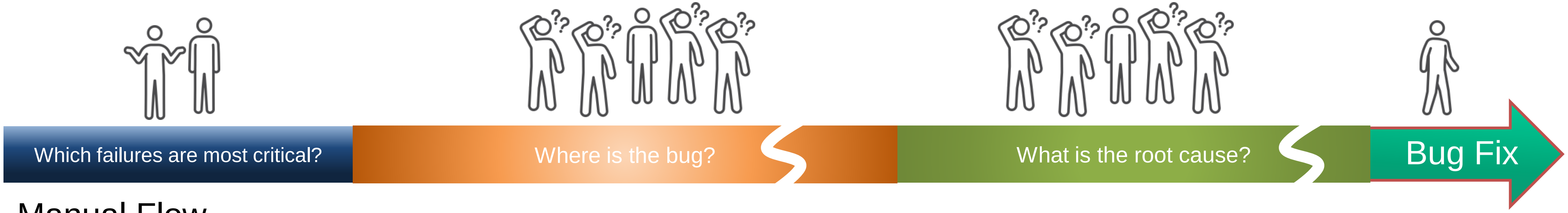
How

Run apps to faster failures debug

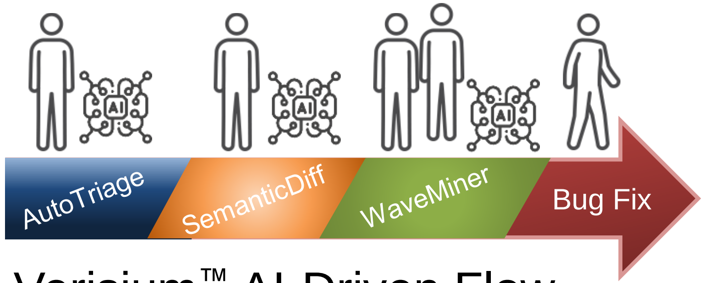
What

Productivity gains for failures debug time

Summary - SoC Debug with Verisium Platform



Manual Flow



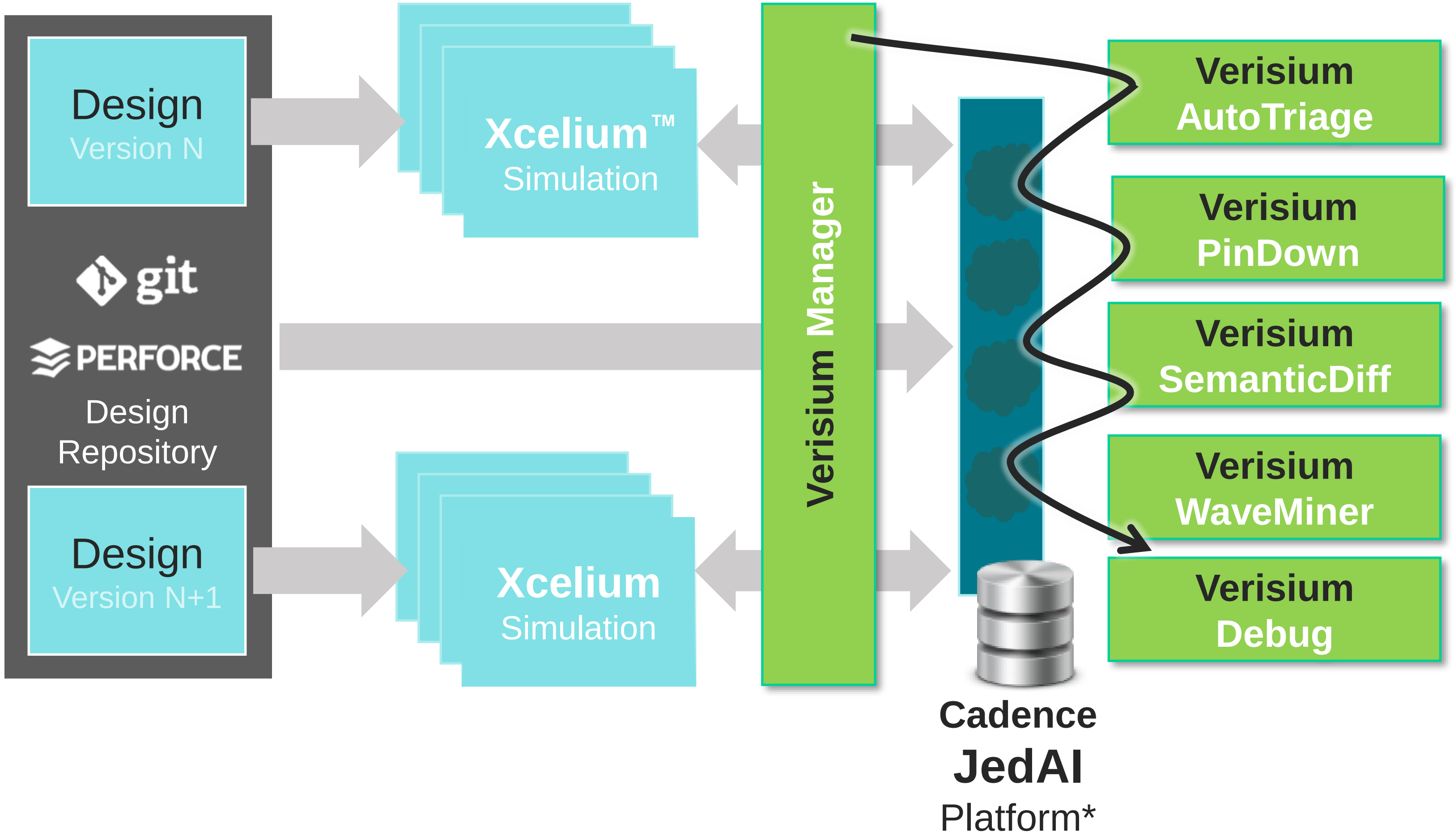
Verisium™ AI-Driven Flow

Potential for 10X improvement in debug productivity

Samsung Collaboration

VERISIUM APPS Collaboration with Samsung

AI-driven submission of tests to compute farm



Used in SoC and IP Regressions

Ongoing setup at SoC level

Used by different teams

Debugging and Analysis

Verisium Results

- To validate the tool – Mid Level Complexity IP(~100k gate count)
- Results

Complexity of Failure	Debug time without tool(Minutes)	Debug time with tool(Minutes)	Debug gain
Less	30	20	1.5X
Less	40	25	1.6X
Moderate	60	40	1.5X
Moderate	80	45	1.7X
Hard	130	70	1.85X
Hard	135	80	1.7X

Q &A