



Connecting Verification Tools To Enhance Their Power

Avi Ziv

Raviv Gal

Verification and Quality Technologies

IBM Research - Haifa



The Verification Evolution



Early verification days



Modern verification – highly automated
Large teams, random generation, coverage



Motivation

The verification process utilizes many tools, each handling a specific aspect of the process

- Verification and work plans
- Job submission
- Bug tracking
- Coverage handling
- Version Control

Each tool can provide its view on the state and progress of the process

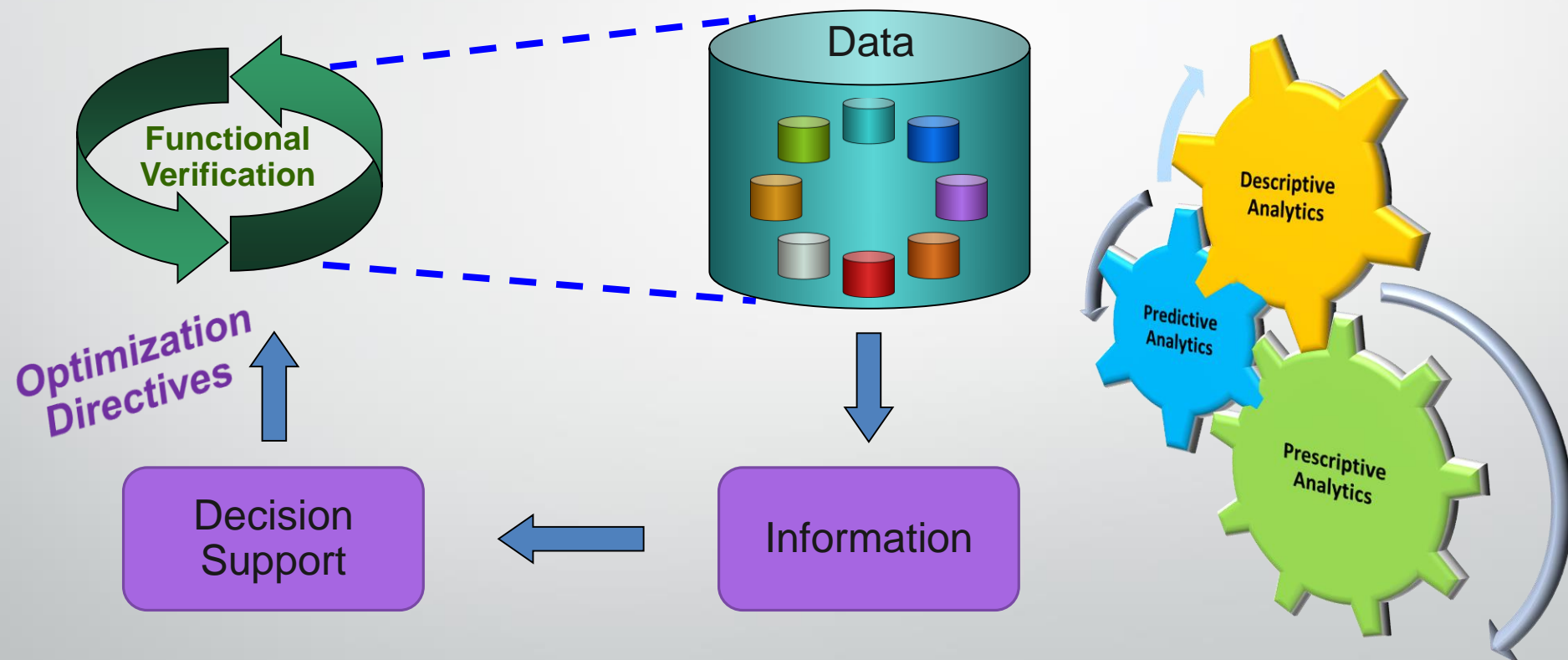
- Work plan burn down
- Job fail rate
- Coverage status

But these views do not necessarily show the entire picture

- Does the introduction of a new feature have the expected effect on bug / coverage?
- What is the most effective test?

Vision

A consolidated platform for planning, tracking, analysis, and optimization of a large scale verification project



Challenges

Many independent tools - data sources

- Each with its own data repository (or without one)
- Each with a different API
- Each uses its own vocabulary

→ Need to connect the independent sources and translate to a common vocabulary

Tools are optimized for operational work

- Optimized for insert-update-delete
- Less so for analytics – mainly data retrieval and manipulation

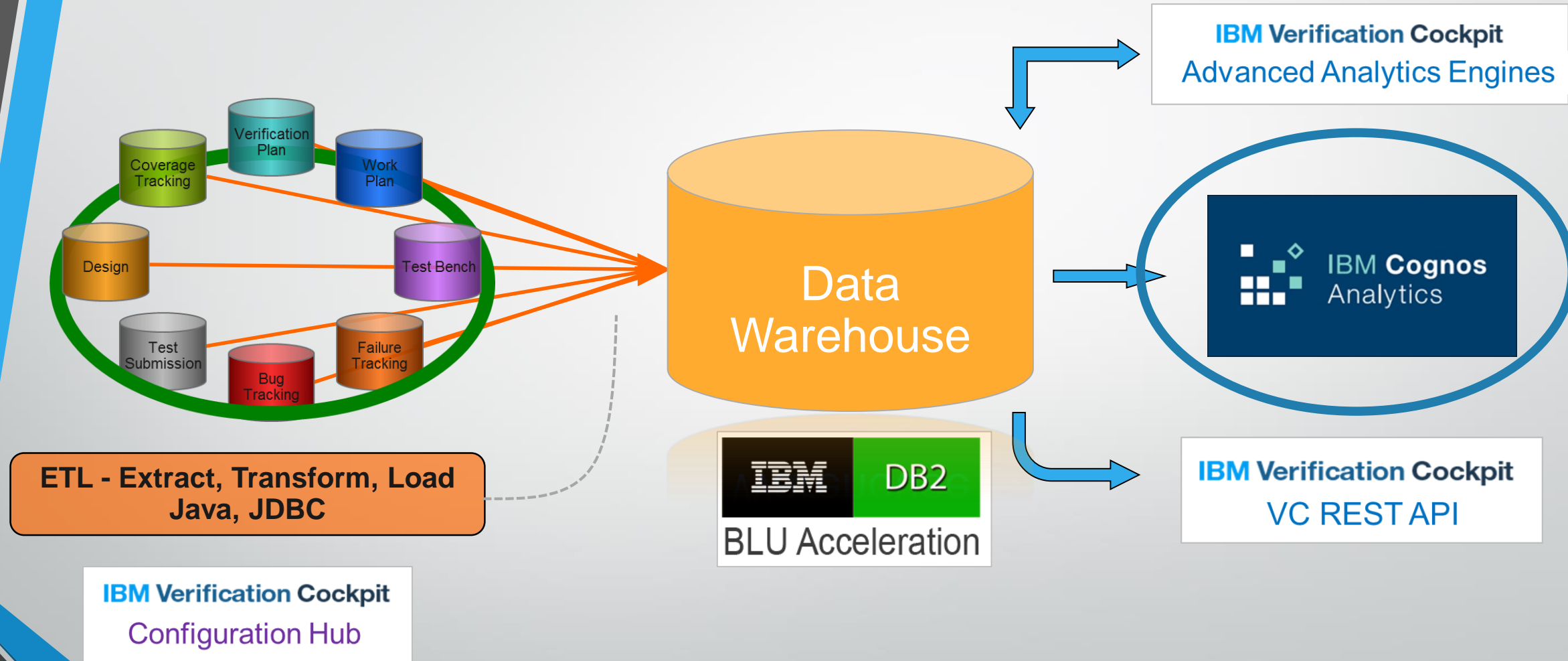
→ Need efficient ways to retrieve and manipulate data

Vast amount of data

- Billions of cycles, millions of tests, hundreds of thousands of coverage events
- Structured and unstructured data...

→ Trade-offs between data stored and granularity (and accuracy) of analysis

Our Platform

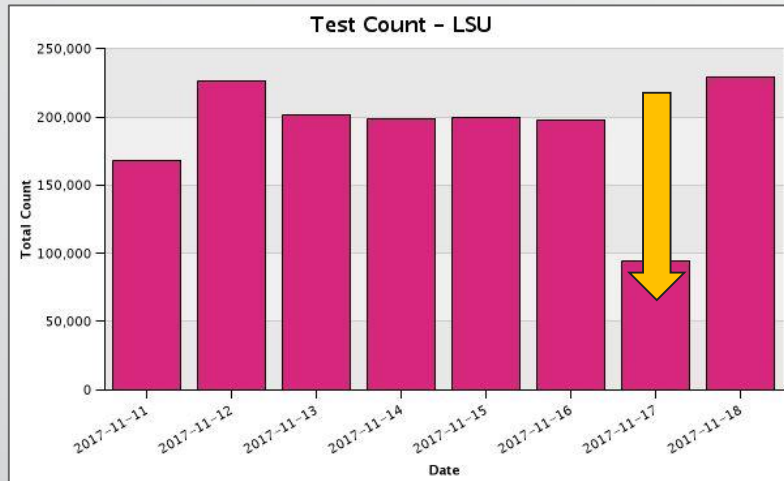


Verification Dashboard

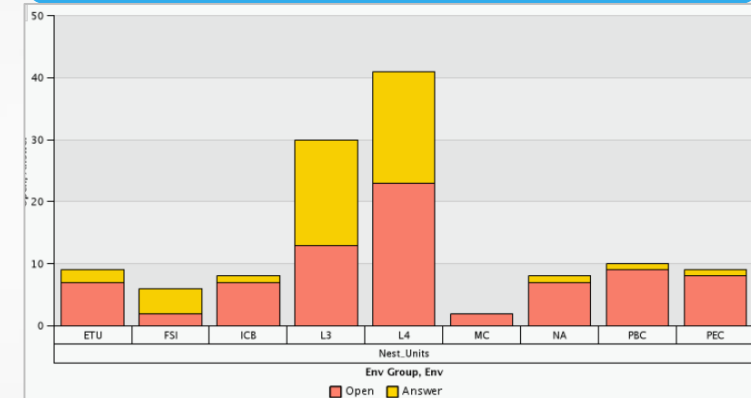
- We all like DNNs, but don't underestimate descriptive analytics
- Different views on status and trends is a powerful feature
- Used by verification lead, team, and managers

Verification Dashboard Snippets

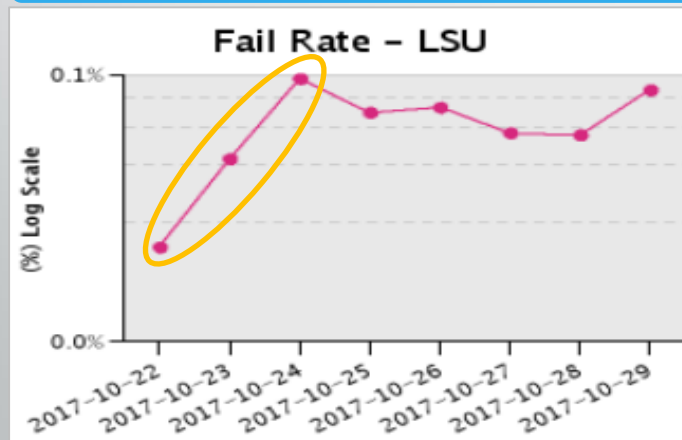
How many tests did I run?



How big is my backlog – open and answered issues?



What is my fail rate?



Show me the coverage status by model, with drill down to events

		target:Unit										
#	Entity	S	Total	Covered		Lightly Covered		Zero Hit		Aged Out	Never Hit	Newly Hit
#	name		#	#	% Δ	#	%	#	%	#	#	#
5	xu_app_zmac		0	0	0	0	0	0	0	0	0	0
13	xu_tlb2_c_zmac		40	17	42.5	2	5	21	52.5	2	19	1
17	xu_tlb2_lru_zmac		73	33	45.2	0	0	40	54.8	23	17	0
21	xu_ztop		3,244	2,050	63.2	23	0.7	1,171	36.1	0	1,171	53
16	xu_tlb2_lkup_zmac		8	6	75	0	0	2	25	0	2	0
14	xu_tlb2_estation_zmac		335	283	84.5	22	6.6	30	9.0	3	25	18
6	xu_async_quiesce_ztop		2,014	1,802	89.5	199	9.9	13	0.6	12	0	0
20	xu_xlate_ztop		431	390	90.5	4	0.9	37	8.6	36	1	0
11	xu_tlb2_a_zmac		109	101	92.7	0	0	8	7.3	0	8	2
9	xu_misc_ztop		644	632	98.1	2	0.3	10	1.6	10	0	1

Combining Multiple Sources

Combining data from multiple sources can provide a different and better view into the verification process

Deepen and widen the view provided by a single source

- Number of fails and number of bugs are two aspects of test quality

Explain anomalies seen in a single source

- Decrease in defect opening rate can be explained by increased activity in code writing

Identify flaws and leaks in the verification process

Example – The “Stupid” Report

Compares the number of passing tests in each environment to the number of tests reporting coverage in the coverage database

Written more to test our ability to create such reports and assess their performance

Found several bugs in the verification environment and its infrastructure

EnvGroup	Env	Date	Test Count			Cycles		
			HDWB	BugSpray	Diff [%]	HDWB	BugSpray	Diff [%]
Core_Units	ISU	2018-10-14	96,359	93,669	3%	4,905M	4,764M	3%
		2018-10-15	110,689	103,699	6%	5,666M	5,302M	6%
		2018-10-16	119,572	103,132	14%	6,123M	5,264M	14%
		2018-10-17	129,570	119,841	8%	6,623M	6,126M	8%
		2018-10-18	106,891	101,406	5%	5,407M	5,153M	5%
		2018-10-19	90,557	92,236	-2%	4,615M	4,695M	-2%
		2018-10-20	93,761	90,666	3%	4,762M	4,603M	3%
		2018-10-21	98,228			5,012M		
	LSU	2018-10-14	232,365	222,065	4%	10,665M	10,446M	2%
		2018-10-15	128,885	101,530	21%	6,401M	5,034M	21%
		2018-10-16	3,603	3,170	12%	129M	107M	18%
		2018-10-17	199	1	99%	5M	51,503	99%
		2018-10-18	143,913	146,074	-2%	7,244M	7,373M	-2%
		2018-10-19	280,666	288,439	-3%	14,263M	14,685M	-3%
		2018-10-20	295,406	288,134	2%	15,022M	14,656M	2%
		2018-10-21	292,882			14,889M		

Regression Test Quality Analysis

Goal

- Run optimized regression in quality (bugs) and resources

Descriptive

- Rank the regression tests according to their ability to fulfill verification goals
 - Target areas in the design that changed recently
 - Find bugs
 - Improve coverage

Analysis method – simple statistics on measures related to the tests

- Multi sources – Job submission, defect tracking, coverage database, version control
- Cross sources – Probability of hitting rare events, defects per million tests

Prescriptive

- Use the analysis results (manually or automatically) to direct the job submission system

TAC/CDG
Out of today's scope

Test Quality Report

Rank	Menu	List	CQ Per M Tests	HDWB				Tracker			ClearQuest		
				Total Count	Pass Rate	Fail Rate	zFail Rate	Avg Bone Pile	Sim Errors	CQs	Opened	Answered	Closed
1	lsu_dev	lsu_uldrikis	4,065.041	246	44.715%	46.748%	8.537%	12	12	1	1	1	1
2	ls_misc	lsu_stressExceptions	2.615	764,963	99.932%	0.024%	0.043%	14	4	1	2	1	1
3	ls_main	ls_gold_vec_rev	0.880	3,409,594	99.986%	0.012%	0.002%	13	14	2	3	2	2
4	ls_main	ls_gold	0.307	9,785,930	99.989%	0.005%	0.006%	23	15	1	3	2	1
5	ls_core	ls_core_vec_rev_directed	0.000	219	0.000%	100.000%	0.000%	0	0	0	0	0	0
5	ls_main	ls_gold_mvcr1	0.000	1,297,118	99.996%	0.000%	0.004%	20	4	1	0	0	0
5	ls_misc	lsu_clkGatingOnVsOff	0.000	12,615	99.849%	0.055%	0.095%	2	2	1	0	0	0
5	ls_misc	lsu_disables	0.000	346,978	99.973%	0.002%	0.025%	2	4	1	0	0	0
5	ls_misc	lsu_hw_prefetch	0.000	1,004,005	99.990%	0.005%	0.005%	6	9	1	0	0	0
5	ls_misc	lsu_setpAggr	0.000	1,034,860	99.963%	0.011%	0.026%	6	8	1	0	0	0

Summary

- 
- The verification process utilizes many different tools to handle its many aspects and facets
 - Combining the operational data of these tools can create views and reveal information that is not easily seen by looking at the data of each tool separately
 - We presented a solution that is based on collecting and combining data from different tools in a centralized data warehouse and using reporting and analysis engines on the combined data
 - Results show that simple reports and deeper analysis of the combined data can help improve the verification process in many ways

