Software Quality Dashboard for Agile Teams

Alexander Bogush

Apr 11th 2014

Schlumberger

Agenda

- Code quality metrics and their importance
- Lean thinking
- Quality Dashboard building blocks
- Green screens
- Software quality attributes
  - Internal, external, trends, ranges & tools
- Visual code maps
- Summary

Quality..

..is the degree to which a product fulfills requirements

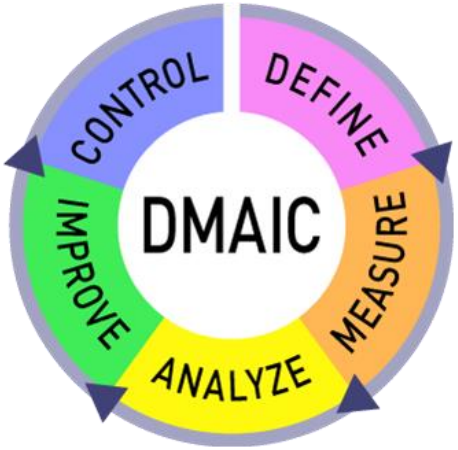
- Higher grade doesn't mean higher quality

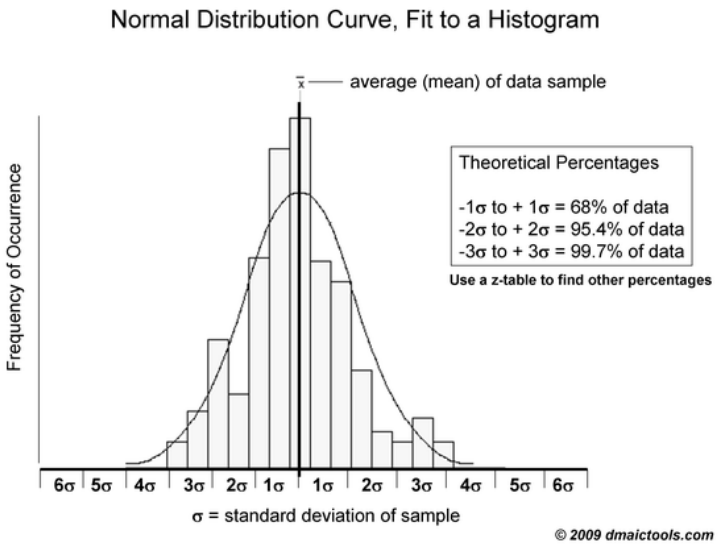
www.laresidencemykonos.com, lvivalive.com/old-city-hostel

Quality Metrics

Quality metrics distinguish good engineering product sample from faulty

- tightly coupled with numerical measurements
- acceptable ranges, % and distribution of defects explicitly defined





www.siliconbeachtraining.co.uk/blog/reduce-queue-times-six-sigma, ecouterre.com, dmaictools.com

Quality Metrics – Why We Care?

Extreme values are an indicator of (hidden) defects, not only “pure” technical debt

- Refactor bad code before it deteriorates further is a sure way to avoid numerous bugs and wasted effort
- Good design is cost effective way to build-in and assure quality

To make our jobs more enjoyable, predictable, and sustainable

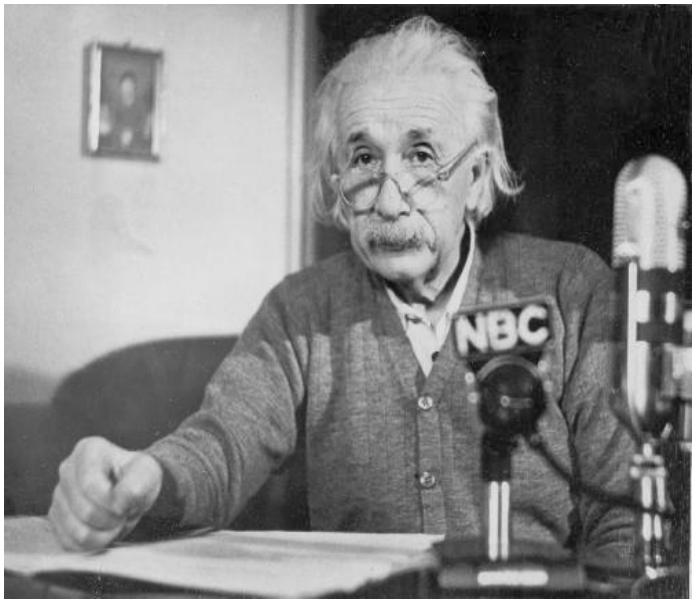
- We spend at least half a day at work
- Spend more time being creative
  - i.e. design new component vs.
  - clean rubbish created by ourselves and others

www.sportscoachingbrain.com

Code Quality Metrics – Keep Design Simple

Make and keep good design to sustain and deliver quality software by good

- Abstraction
- Cohesion and Coupling
- Information Hiding
- Structural Complexity
- …etc

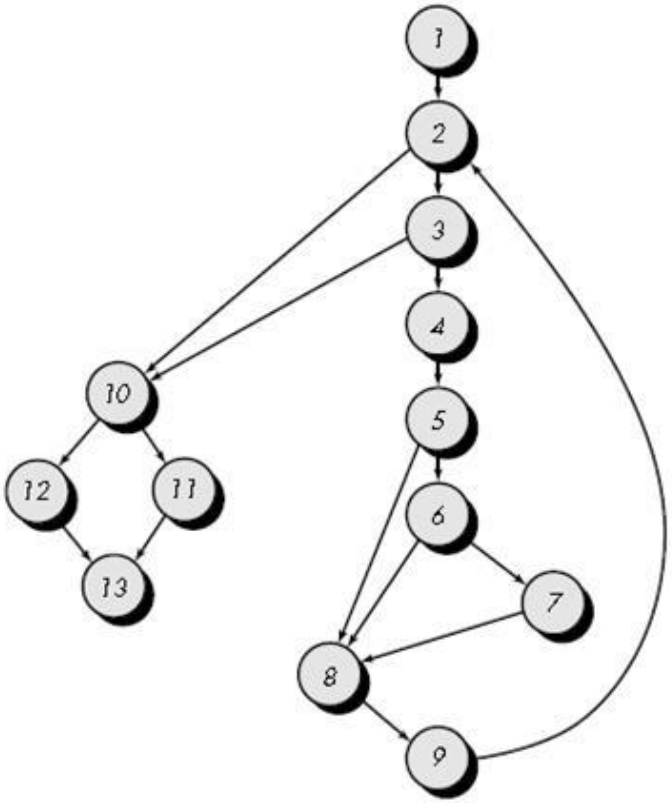


"Make things as simple as possible, but no simpler"
- Albert Einstein, interview at NBC

thedemocraticdaily.com, thepridepost.com

Cyclomatic Complexity

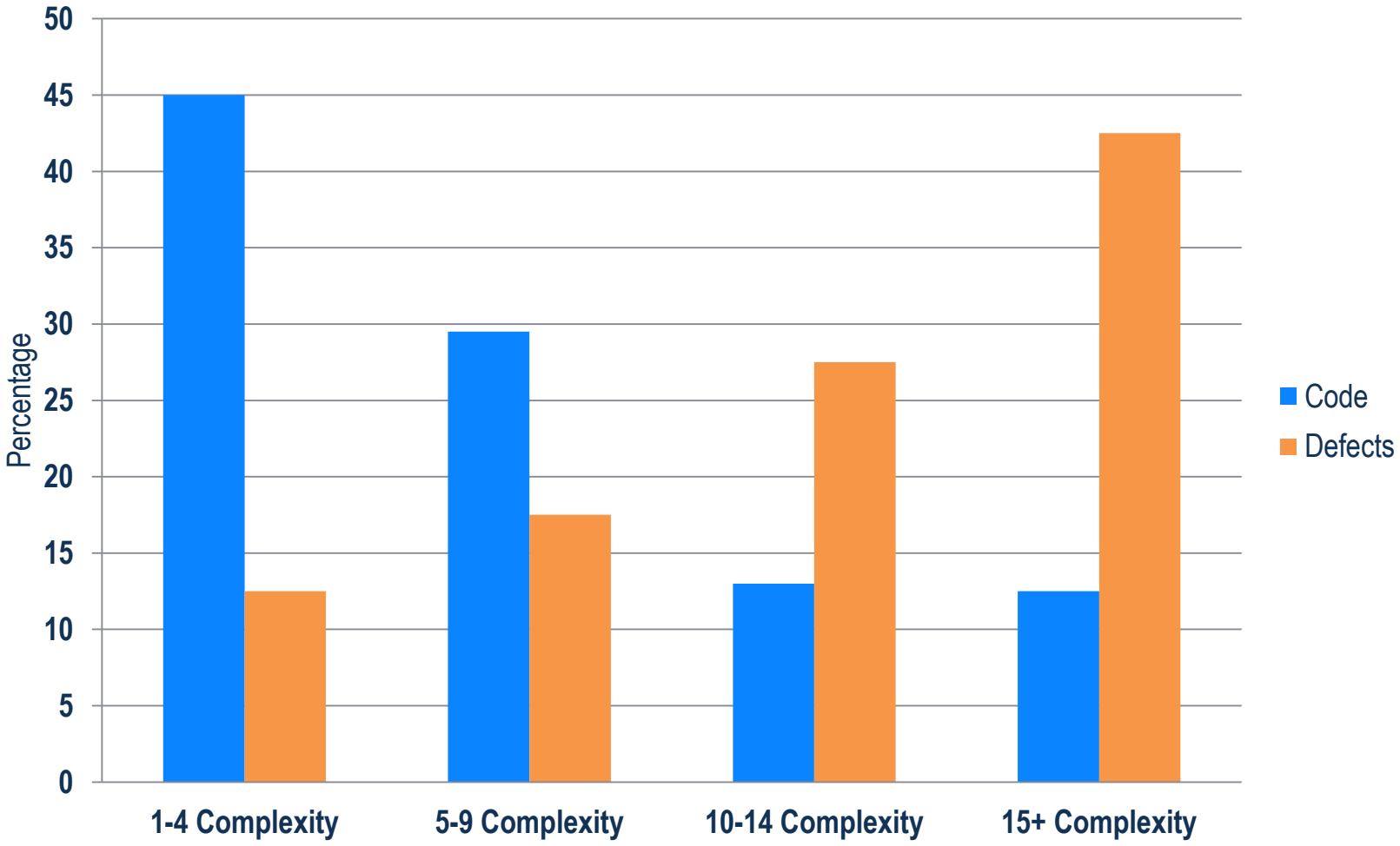
Control Flow graph



- # of independent paths through the code
- each new decision (*if, elseif, switch, for, do while, catch, etc)* adds one point

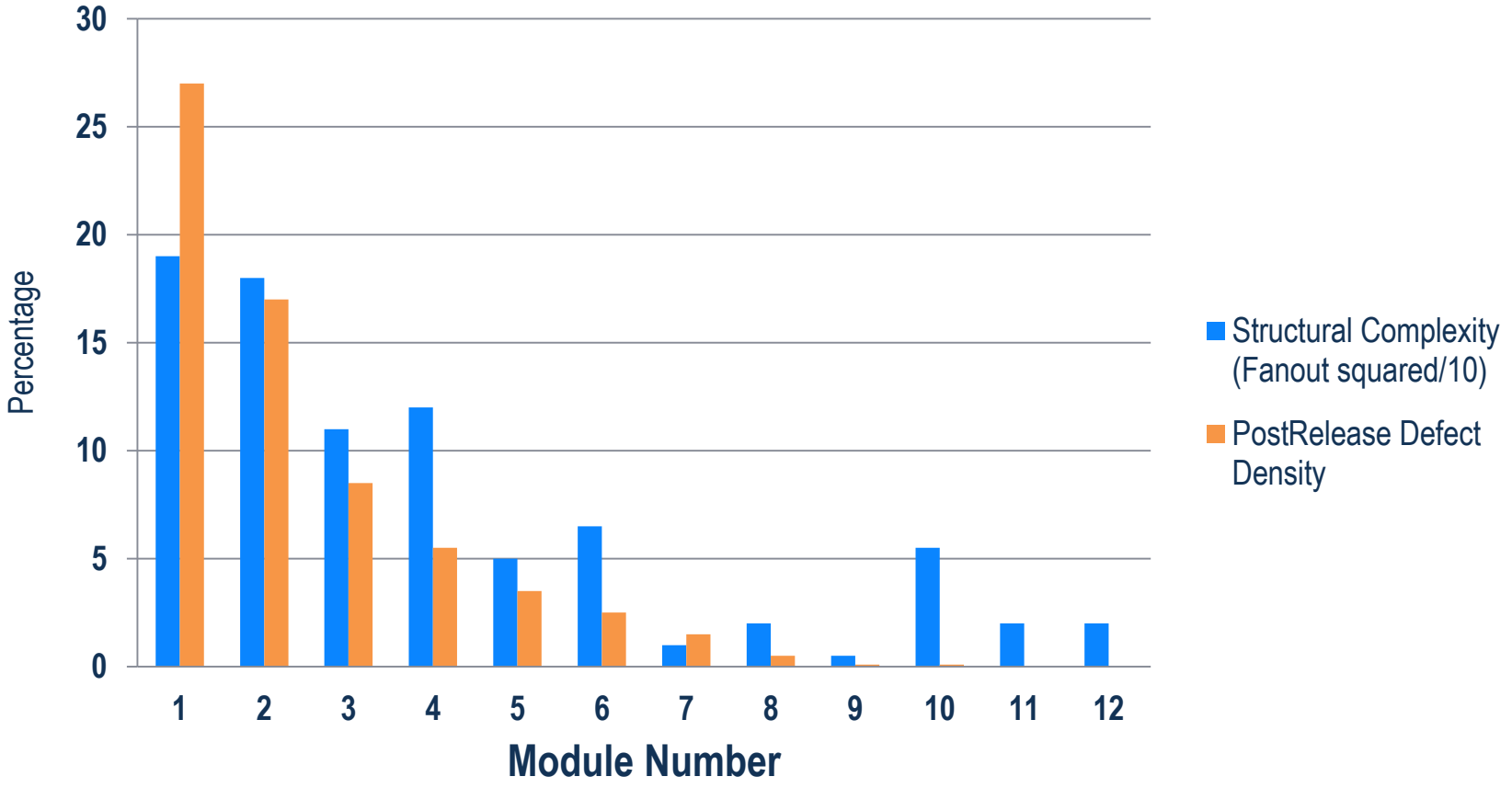
http://www.codeproject.com/Articles/37111/White-Box-Testing-Technique

Cyclomatic Complexity vs Defect Rate



Mark Schroeder, "A Practical Guide to Object-Oriented Metrics", IT Pro, Nov/Dec 1999.

Structural Complexity vs Defect Rate



Robert Grady, *Practical Software Metrics for Project Management and Process Improvement*, Prentice Hall PTR, 1992.

Defect probability vs Fan Out and SLOC

Fan Out	FO^2/10	5	10	15	20	25	30	35	40	45	50	75	100
1	0.1	0.10%	0.10%	0.20%	0.20%	0.30%	0.30%	0.40%	0.40%	0.50%	0.50%	0.80%	1.00%
2	0.4	0.20%	0.40%	0.60%	0.80%	1.00%	1.20%	1.40%	1.60%	1.80%	2.00%	3.00%	4.00%
3	0.9	0.50%	0.90%	1.40%	1.80%	2.30%	2.70%	3.20%	3.60%	4.10%	4.50%	6.80%	9.00%
4	1.6	0.80%	1.60%	2.40%	3.20%	4.00%	4.80%	5.60%	6.40%	7.20%	8.00%	12.00%	16.00%
5	2.5	1.30%	2.50%	3.80%	5.00%	6.30%	7.50%	8.80%	10.00%	11.30%	12.50%	18.80%	25.00%
6	3.6	1.80%	3.60%	5.40%	7.20%	9.00%	10.80%	12.60%	14.40%	16.20%	18.00%	27.00%	36.00%
7	4.9	2.50%	4.90%	7.40%	9.80%	12.30%	14.70%	17.20%	19.60%	22.10%	24.50%	36.80%	49.00%
8	6.4	3.20%	6.40%	9.60%	12.80%	16.00%	19.20%	22.40%	25.60%	28.80%	32.00%	48.00%	64.00%
9	8.1	4.10%	8.10%	12.20%	16.20%	20.30%	24.30%	28.40%	32.40%	36.50%	40.50%	60.80%	81.00%
10	10	5.00%	10.00%	15.00%	20.00%	25.00%	30.00%	35.00%	40.00%	45.00%	50.00%	75.00%	100.00%
11	12.1	6.10%	12.10%	18.20%	24.20%	30.30%	36.30%	42.40%	48.40%	54.50%	60.50%	90.80%	121.00%
12	14.4	7.20%	14.40%	21.60%	28.80%	36.00%	43.20%	50.40%	57.60%	64.80%	72.00%	108.00%	144.00%
13	16.9	8.50%	16.90%	25.40%	33.80%	42.30%	50.70%	59.20%	67.60%	76.10%	84.50%	126.80%	169.00%
14	19.6	9.80%	19.60%	29.40%	39.20%	49.00%	58.80%	68.60%	78.40%	88.20%	98.00%	147.00%	196.00%
15	22.5	11.30%	22.50%	33.80%	45.00%	56.30%	67.50%	78.80%	90.00%	101.30%	112.50%	168.80%	225.00%
20	40	20.00%	40.00%	60.00%	80.00%	100.00%	120.00%	140.00%	160.00%	180.00%	200.00%	300.00%	400.00%
25	62.5	31.30%	62.50%	93.80%	125.00%	156.30%	187.50%	218.80%	250.00%	281.30%	312.50%	468.80%	625.00%

Robert Grady, *Practical Software Metrics for Project Management and Process Improvement*, Prentice Hall PTR, 1992.

Lean Principles

Long-term strategy

Flow (eliminate waste)

Pull

Less variability

Stop & Fix

Standardize

Simple visual management

Good technology

Leaders-teachers

Develop people

Help partners

Go See

Consensus

Continuous improvement

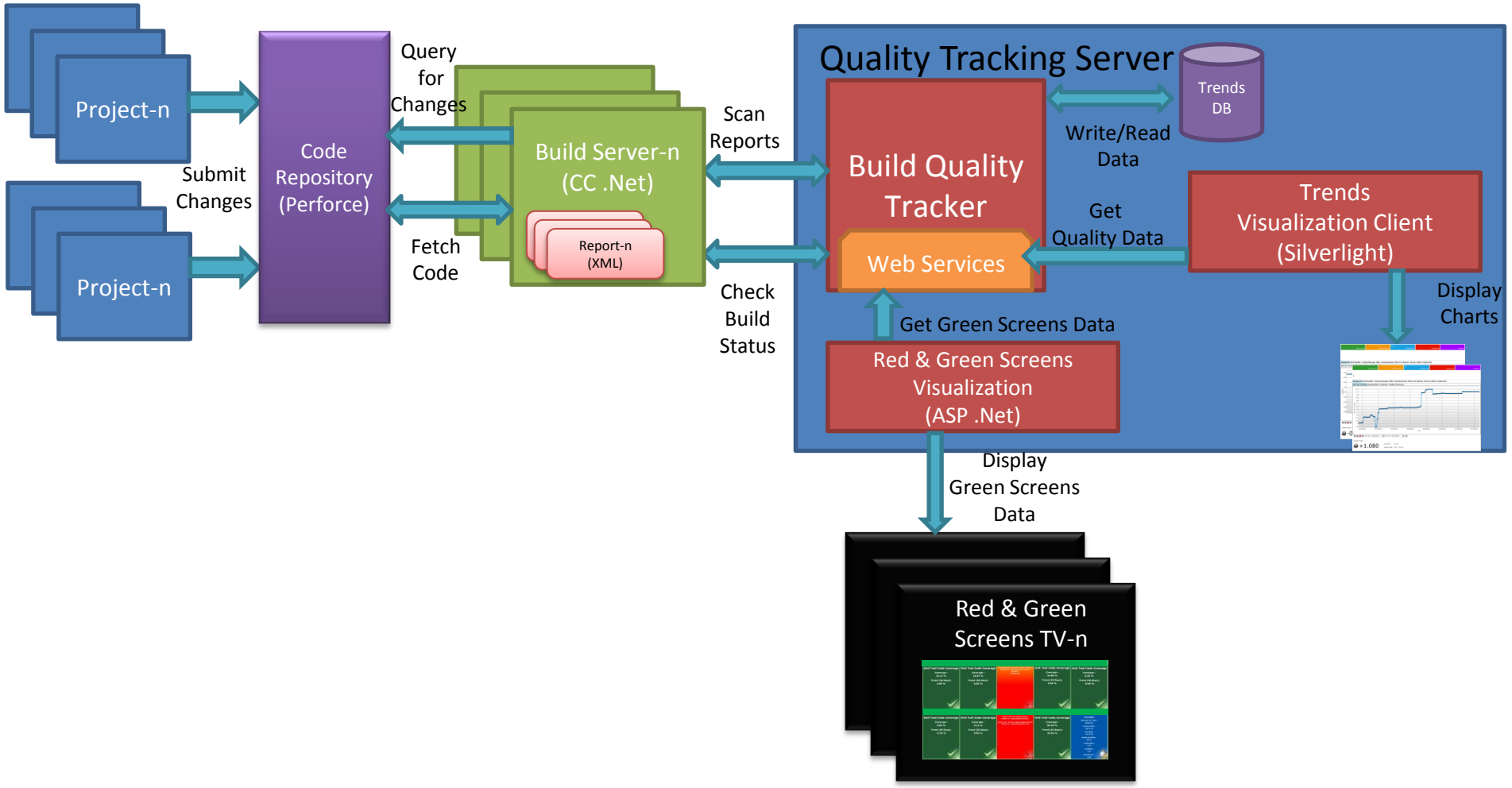
The Toyota Way 2001, Wikipedia

Lean Spring Clean Summary

- 1-2 hours each, 40% of engineers

- Typical projects:
  - Personal efficiency (email, laptop, space, etc)
  - Reducing technical debt
  - Team efficiency (Clean-up bug tracking, TFS, etc)

Quality Dashboard Building Blocks

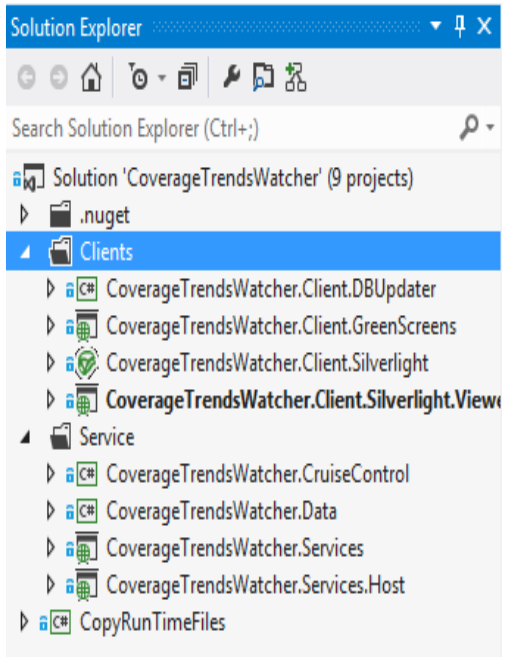


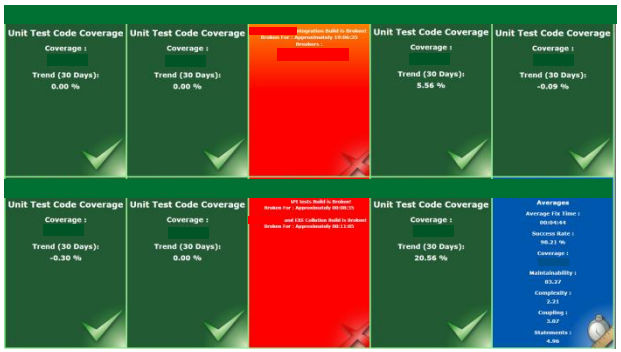
Piotr Szewello, Schlumberger internal report, 2013

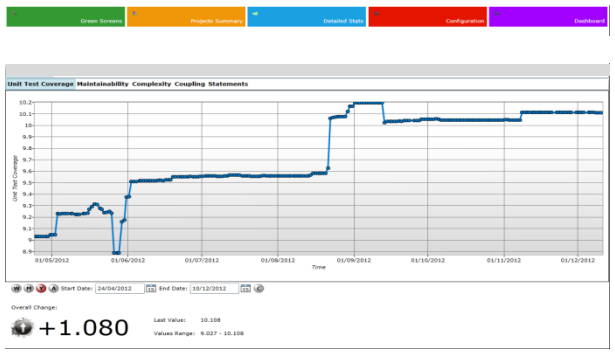
Quality Trends Watcher

Server Side Solution that

- communicates with CC .NET build servers
- every day scans build logs to extract quality data
- persists this data in a repository
- provides WCF services for Visualization Clients
  - Silverlight client for Trends Visualization
  - HTML client for “Red & Green” Screens

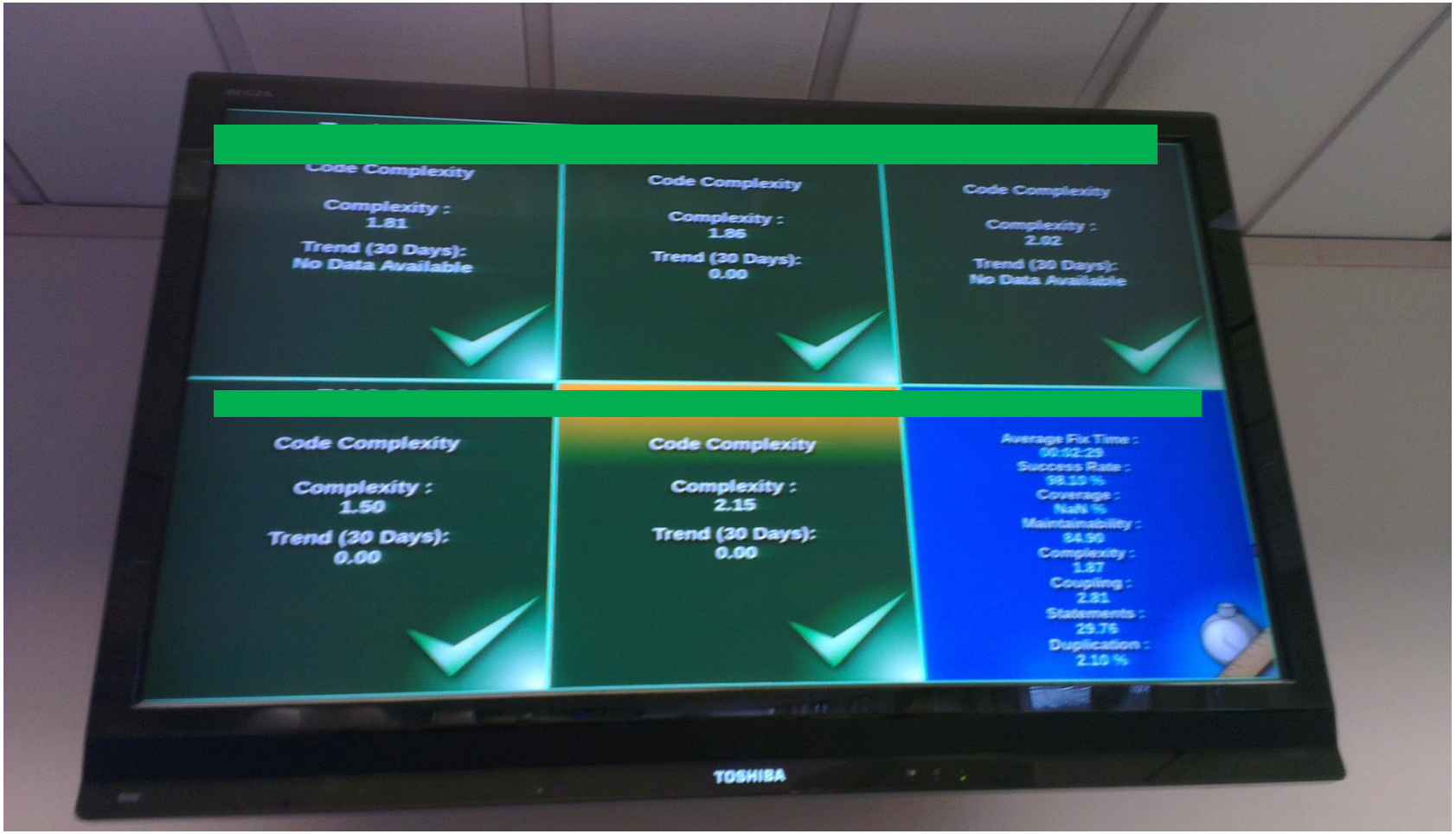






Piotr Szewello, Schlumberger internal report, 2013

Green Screens



Software Quality Attributes

External ("in the eyes of customer" and others)

- Performance
- Correctness
- Scalability
- Usability
- Efficiency
- Reliability
- Integrity
- Robustness
- Adaptability

Internal (team's ability to deliver/ rate of returns /integrity)

- Maintainability
- Flexibility
- Portability
- Reusability
- Readability
- Testability
- Understandability

Code Complete 2 edition, Steve McConnell, Microsoft Press, 2004

Internal Quality Trends & Tools

Basic yet efficient set of metrics

- **Test Coverage** (NCover/NUnit, VSTest/NUnit/CppUnit)
  - TDD, ability to release frequently with confidence, etc.
- **Cyclomatic Complexity** (FxCop, SourceMonitor)
  - Directly related to number of defects if exceeds good range
- **Coupling** (FxCop)
  - Global complexity, can be extended with OO metrics
- **% of Duplicated code** (Atomiq)
- **Number of Statements** (FxCop, SourceMonitor)
  - Secondary to CC
- **Maintainability** (FxCop)
  - Turned out to be not very useful

Metrics Objectives Tailored per Project: Avg & Trends

	Coverage	Delta	Maintain.	Delta	C.Compl.	Delta	Coupling	Delta	Stat.	% Dupl
Pr. A	10.1%	1.1%	81.8%	0.12%	2.352	-0.01	3.26	-0.01	6.22	17.3%
Pr. B	12.3%	3.2%	82.1%	0.45%	2.439	0	3.25	-0.06	5.73	6.4%
Pr. C	60.5%	12.5%	84.9%	-0.01%	1.826	0	2.9	0	4.21	19.2%
Pr. D	21.6%	9.7%	85.6%	0.34%	1.513	-0.05	2.46	0.07	4.34	2.2%
Pr. E	7.4%	3.3%	79.6%	-0.36%	2.965	0.03	3.86	0.03	7.07	4.5%
Pr. F	6.7%	-14.6%	84.7%	-0.94%	2.571	0.42	3.28	0.23	4.57	21.7%
Pr. G	33.5%	0.0%	N/A	N/A	1.875	0.03	N/A	N/A	11.83	41.1%
Pr. H	60.8%	1.3%	85.3%	-0.34%	2.089	0.08	2.43	0.08	5.19	2.3%
Pr. I	N/A	N/A	84.4%	0.18%	2.111	0.04	3.14	-0.06	7.20	N/A

- Red boxes are ready candidates for improvement
- Make it a build failure if they deteriorate further

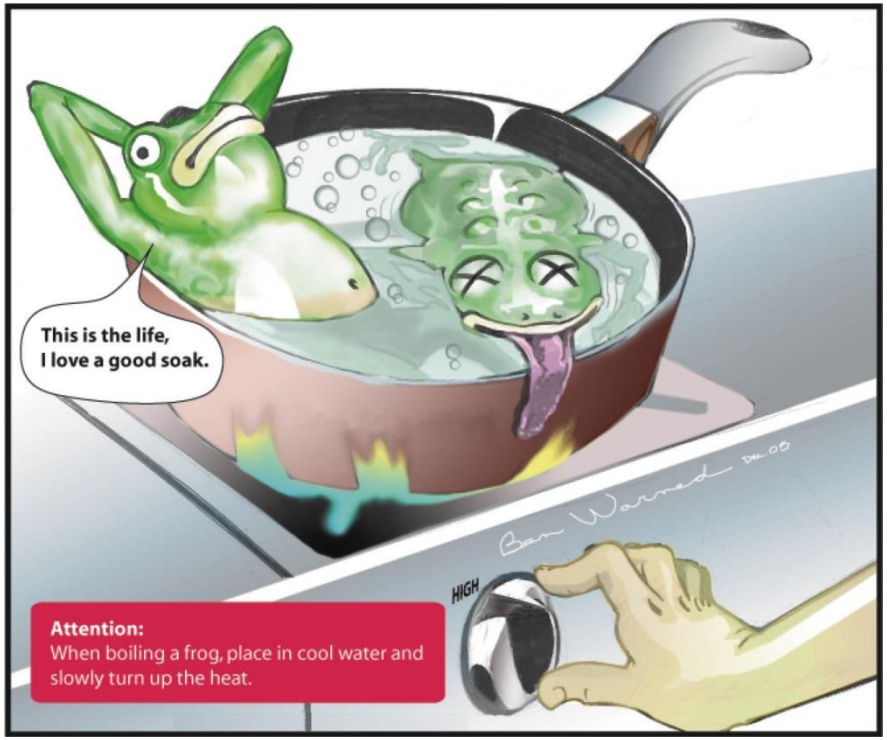
Internal Quality Metrics per Project: 15 worst extremes & their trends

	Maintainability	Delta	C.Complexity	Delta	Coupling	Delta
Product A	9.73	0.13	81.87	0.07	54.47	0.07
Product B	16.20	0.73	63.8	1.33	44.2	-0.67
Product C	22.00	-0.07	25.27	0	35.47	0
Product D	21.87	0.73	10.8	-3.07	25.4	-0.8
Product E	17.13	-0.01	58.2	0.4	43.47	0.2
Product F	21.47	-1.00	86.33	6.6	31.33	-0.47
Product G	N/A	N/A	119.6	2.73	N/A	N/A
Product H	22.60	-0.07	47.67	1.4	31.87	-0.33
Product I	23.00	-0.13	21.47	0.67	32	0.13

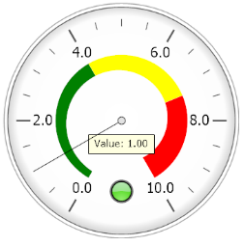
All red boxes are candidates for monthly/quarterly/annual targets for improvements

Creeping Normality - Frog Missing Temperature Sensor

It is surprisingly easy for code to become so complex that it can no longer support any significant enhancement



dT



T

http://allaboutfrogs.org/stories/boiled.html, http://www.offroaders.com devexpress.com

Boundary Thresholds/Healthy Ranges

- Test Coverage [>80 green, 50-80 warning zone, <50 red]
- Cyclomatic Complexity [10 15]
- Coupling [10, 80]
- % of Duplication [2,5]
- Number of Statements [300, 600]
- Maintainability [82, 20]
- Methods per Class [10,22]
- Calls per Method [5, 15]
- Maximum block depth [3, 6]
- Depth of Inheritance [3, 6]

Software Quality Attributes

External ("in the eyes of customer" and others)

- Performance
- Correctness
- Scalability
- Usability
- Efficiency
- Reliability
- Integrity
- Robustness
- Adaptability

Internal (team's ability to deliver/ rate of returns /integrity)

- Maintainability
- Flexibility
- Portability
- Reusability
- Readability
- Testability
- Understandability

Code Complete 2 edition, Steve McConnell, Microsoft Press, 2004

External Quality Attributes Report

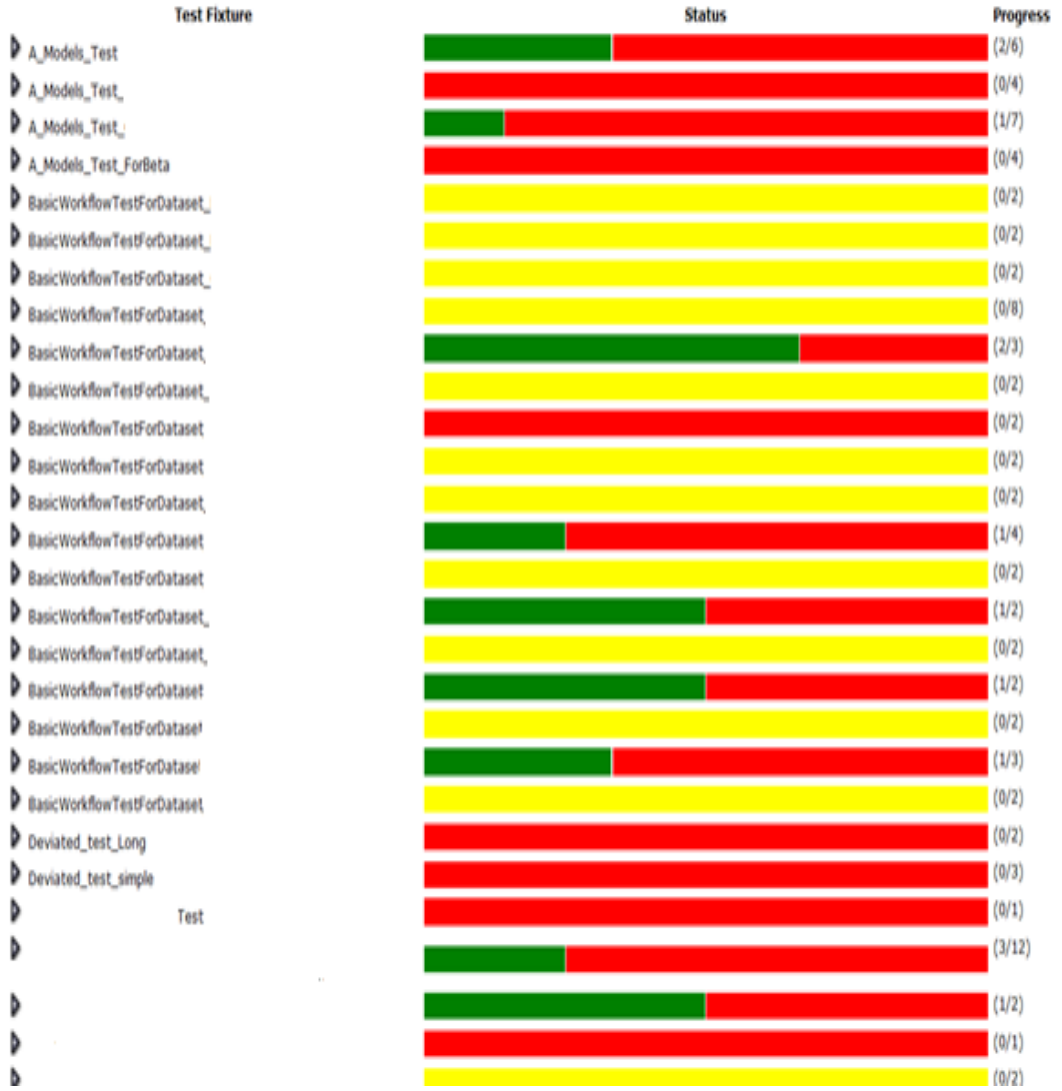
- Specific attributes and target values identified through ATAM process
- Derived from nightly build test suite
- Aggregated to give min/max/average values and compare with targets/goals

Quality Attributes Summary

Categories	Range		
Project Performance Parameters (In Seconds)	Max	Average	Min
ModelCreationTime	2	1	0
UpscalingTime	54	28	3
OptimizationSessionTime	168	83	42
ScenarioTime	168	83	42
TrialTime	53	8	0
SimulationTime	27	4	0
Project Scalability Parameters	Max	Average	Min
Cells	55419	28909	2400
CellsUsed	28557	15098	1639
Slabs	19	10	2
	10	2	0
	10	2	0
	3	0	0
	3	0	0
	7	1	0
	7	1	0
	2	1	1
	14	4	1
Project Correctness Parameters	Max	Average	Min
	9	4	1
	6	2	1
	3	1	0
	438053695.10184717	357523638	278906419.295025

Workflows Status – Client's Value Flow

Comparison with Baseline			
PROGRESSION	N/A	N/A	PASS DFC: | BDFC: 0 | FDFC: 0 | BFDFC: 0
EQUAL	PASS DFC: 36 | BDFC: 36 | FDFC: 0 | BFDFC: 0	N/A	N/A
EQUAL	N/A	PASS DFC: 140 | BDFC: 86 | FDFC: 0 | BFDFC: 0	N/A
EQUAL	PASS DFC: 33 | BDFC: 33 | FDFC: 0 | BFDFC: 0	PASS DFC: 85 | BDFC: 4 | FDFC: 0 | BFDFC: 0	N/A
PROGRESSION	PASS DFC: 6 | BDFC: 0 | FDFC: 0 | BFDFC: 0	PASS DFC: 31 | BDFC: 0 | FDFC: 0 | BFDFC: 0	N/A
PROGRESSION	PASS DFC: 11 | BDFC: 0 | FDFC: 0 | BFDFC: 0	N/A	PASS DFC: 36 | BDFC: 25 | FDFC: 0 | BFDFC: 0
EQUAL	PASS DFC: 2 | BDFC: 2 | FDFC: 0 | BFDFC: 0	N/A	N/A
EQUAL	PASS DFC: 88 | BDFC: 95 | FDFC: 0 | BFDFC: 0	N/A	PASS DFC: 67 | BDFC: 53 | FDFC: 0 | BFDFC: 0
PROGRESSION	N/A	N/A	N/A
EQUAL	PASS DFC: 469 | BDFC: 500 | FDFC: 0 | BFDFC: 0	PASS DFC: 168 | BDFC: 161 | FDFC: 0 | BFDFC: 0	PASS DFC: 437 | BDFC: 508 | FDFC: 0 | BFDFC: 0
EQUAL	N/A	N/A	PASS DFC: 11 | BDFC: 11 | FDFC: 0 | BFDFC: 0
EQUAL	PASS DFC: 37 | BDFC: 37 | FDFC: 0 | BFDFC: 0	N/A	N/A
EQUAL	N/A	N/A	PASS DFC: 10 | BDFC: 13 | FDFC: 0 | BFDFC: 0
EQUAL	PASS DFC: 4 | BDFC: 4 | FDFC: 0 | BFDFC: 0	PASS DFC: 16 | BDFC: 16 | FDFC: 0 | BFDFC: 0	MINOR DFC: 39 | BDFC: 39 | FDFC: 2 | BFDFC: 2



Code “Soundness” Index

JIT(JavaScript InfoVis ToolKit) API http://philogb.github.io/jit/

Code “Soundness” Index – zoom out

Summary and Acknowledgements

- All the trademarks mentioned and pictures used are of their respective owners
- Big thanks to engineers and interns from Abingdon Answer Products who participated in the development of this system
- Special thanks to Steve Tockey for inspiring me to look at complexity and Nigel Lester for reviewing and promoting it to ACCU
- Contact information

Alexander Bogush abogush@slb.com
Senior software engineer at Schlumberger
Abingdon, UK

Thank you

Any questions?