Reducing Software Vulnerabilities
The “Vital Few” Product and Process Metrics

NDIA

20th Annual Systems Engineering Conference

Girish Seshagiri

19735

Topics

- Why We Are Here
- Ubiquitous Software Defects, “Patch and Pray”, What is at Stake
- Unique Point in Time
- State of the Practice – Software Development
- Managing Secure Software Development
- “Vital Few” Metrics
- Case Study

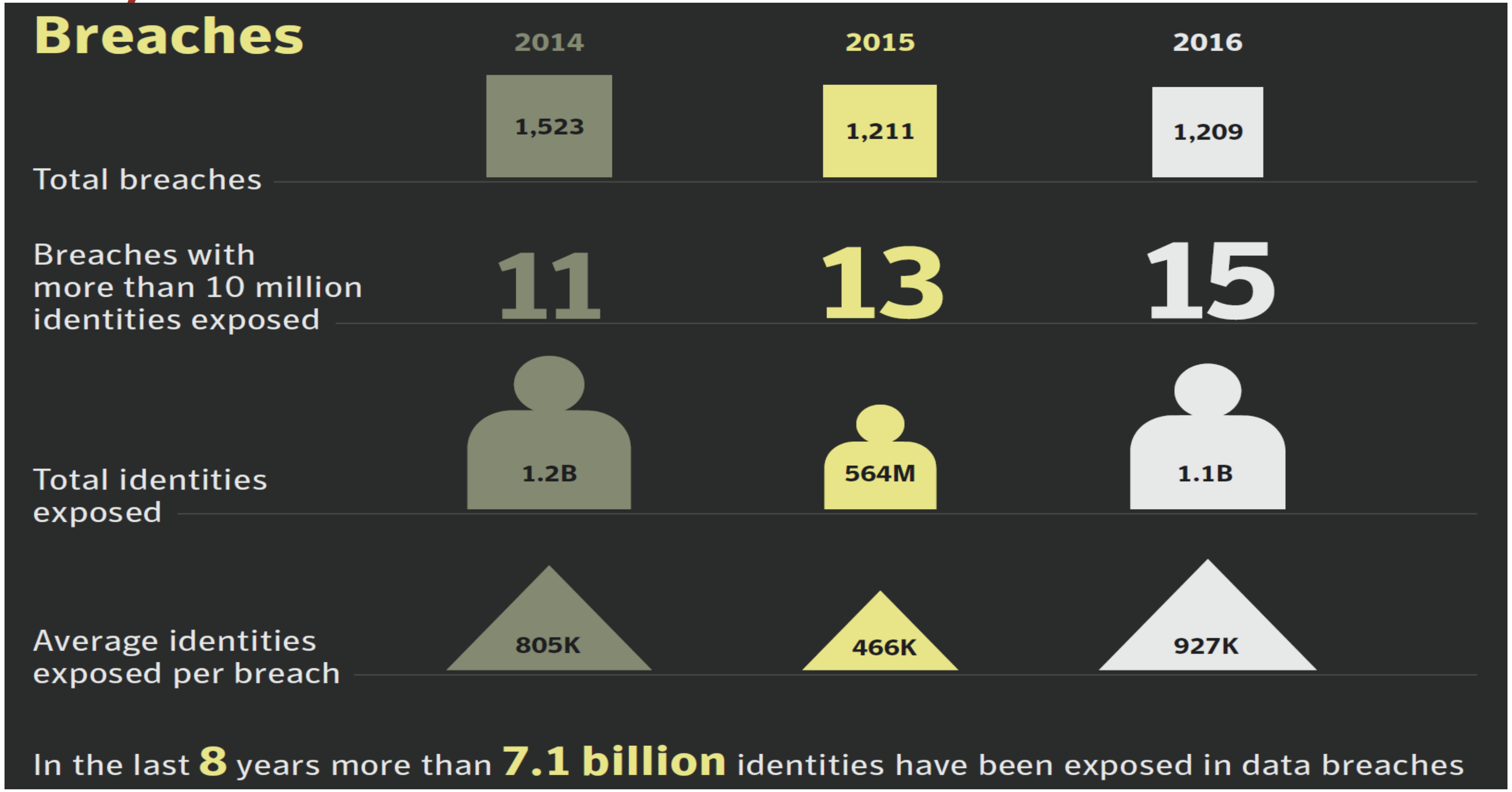
Main Points

- **Defective software** is insecure
  - “If you have a quality problem, you have a security problem”
- Cannot **rely on testing alone** to find and remove software defects
  - Common misconception – “if it passes test, it must be OK”
  - Root cause of “Deliver now, Fix later” culture, technical debt, increase in total ownership cost in many agile projects
- Move from reactive to proactive – **threat detection to threat prevention**
- **Reducing vulnerabilities** - number one goal for every agile software team
  - Manage software by managing software quality
- Need **transformation** – organization, team, individual
  - Combine high maturity CMMI Maturity Level 5 with agile team process to create secure software by self-managed teams and empowered developers with certifications

Why We Are Here

- Increasing number of cyber attacks against critical infrastructure
- Current cybersecurity measures are reactive
- 90% of attacks are successful by exploiting defects in software
- Software developers not trained to deliver software with fewer vulnerabilities
- 1.5 million cybersecurity jobs currently unfilled

Why We Are Here
Personal Identity Breaches

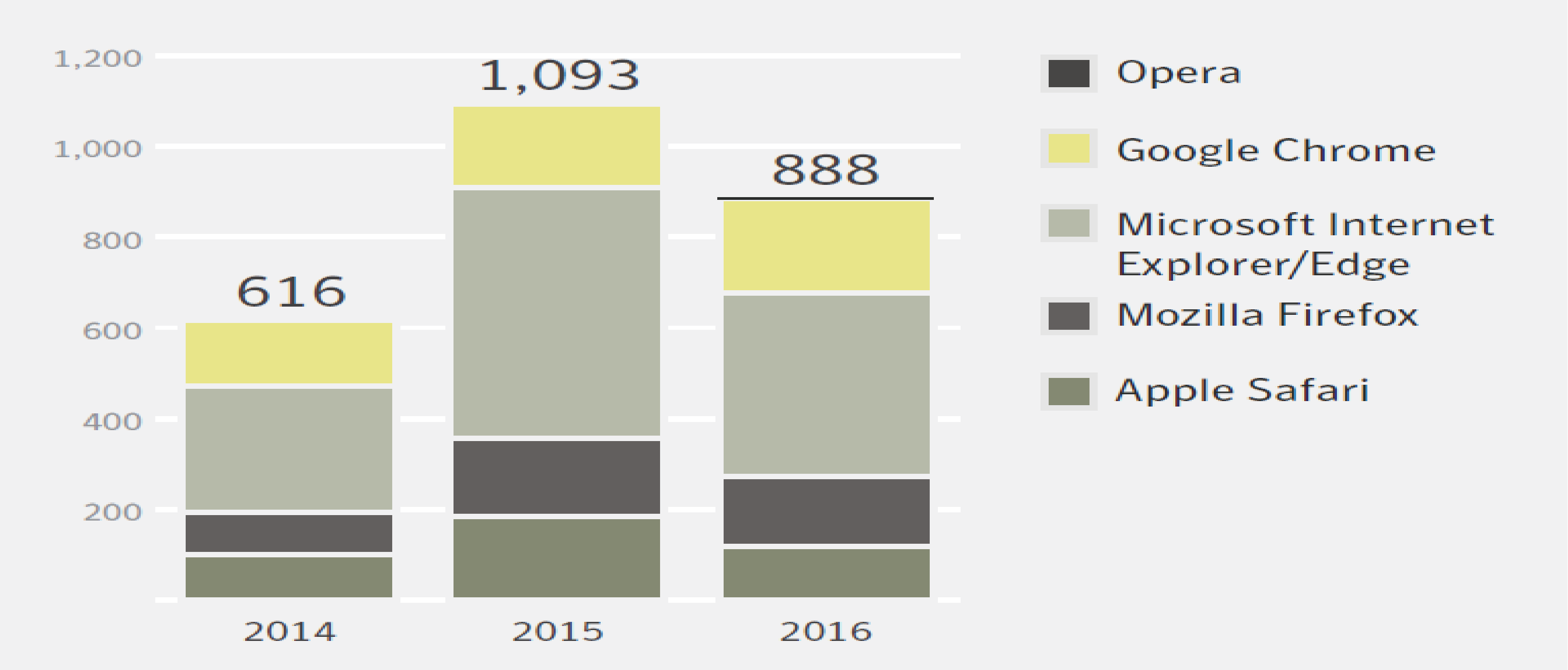


Breaches

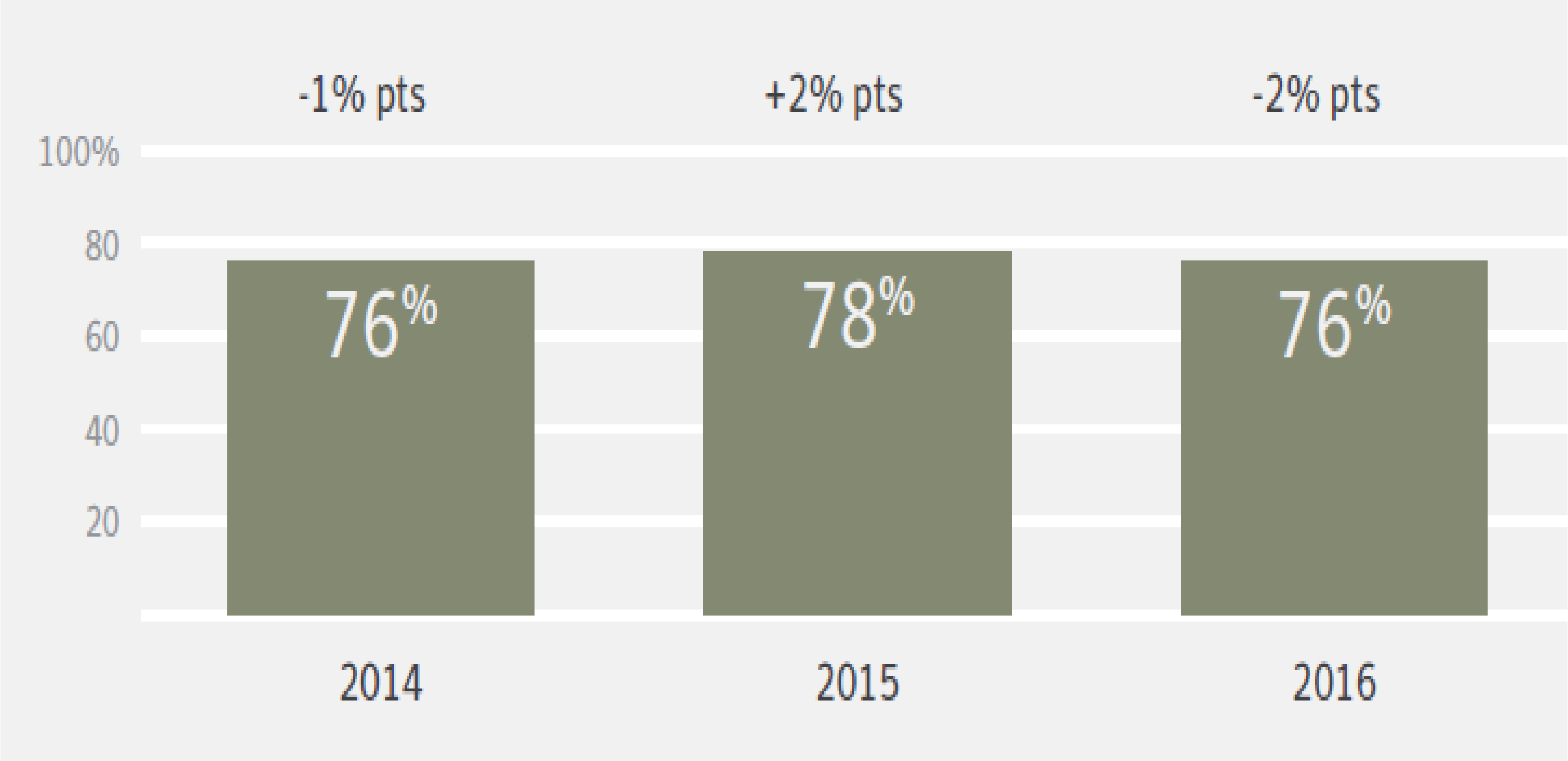
	2014	2015	2016
Total breaches	1,523	1,211	1,209
Breaches with more than 10 million identities exposed	11	13	15
Total identities exposed	1.2B	564M	1.1B
Average identities exposed per breach	805K	466K	927K

In the last **8** years more than **7.1 billion** identities have been exposed in data breaches

Why We Are Here
Browser Vulnerabilities



Why We Are Here
Websites with Vulnerabilities



Patch and Pray - 1

Microsoft Patch Tuesday

On Tuesday, September 12, Microsoft released fixes for more than 80 security issues in multiple products, including Windows, Office, Microsoft .NET Framework, Flash, Internet Explorer, and Edge.

Patch and Pray - 2

Apache Struts Vulnerability Exploited in Equifax Breach

Equifax has acknowledged that the massive breach that exposed personal information of as many as 143 million people was due to a failure to apply a patch for a vulnerability in Apache Struts. A patch for the flaw was released on March 6, 2017. The Equifax breach occurred in "mid-May" 2017.

Patch and Pray - 3

Adobe Security Updates

Adobe has released updates to address security issues in Flash Player, ColdFusion, and RoboHelp for Windows. The Flash updates, available for Windows, Mac, Linux, and Chrome OS, address two critical memory corruption flaws. The ColdFusion update includes fixes for four flaws, and the RoboHelp update fixes two flaws

Patch and Pray - 4

Patches dominate the Top of the News this week.

Bill Murray speaks for many when he writes: "The cost to tolerate or remediate a design, recording, or coding error goes up exponentially with the time to its discovery. **There is something fundamentally wrong with an industry in which the toleration, indeed the institutionalization, of late discovery and remediation of error is as it is in ours."**

And John Pescatore offers a path toward fixing that fundamental flaw in the software industry, describing a future where larger buyers of software (government, for example, along with the Business Roundtable) set a much higher bar for application security testing at multiple stages of the development life cycle, both for custom software they develop and for every package they buy, with **substantial contractual penalties for vendors who fail**.

What is at Stake - 1

Some US States Are Going Back to Paper Ballots

In the wake of rising concerns about the security of electronic voting systems, several US states are returning to the use of paper ballots for their elections. Georgia will pilot a paper-ballot system in elections this fall.

FDA Approves Pacemaker Patch, Announces Recall of Abbott/St. Jude Medical Devices

The US Food and Drug Administration (FDA) has announced a recall of more than 450,000 pacemakers because they require a firmware update to address several security issues. The recall applies to several models of pacemakers manufactured by Abbott, formerly known as St. Jude Medical. Patients must visit their doctor's office where the update can be installed while the device is in backup mode. The flaws could be exploited to gain unauthorized access to vulnerable devices and issue commands to modify the pacemaker's settings and functionality.

What is at Stake - 2

Car Safety Vulnerability Lies in the Way CAN Handles Error Messages

A vulnerability in the Controller Area Network (CAN) that exists in most new automobiles could be exploited to shut down components of the car, including safety systems. Any component connected to the car's CAN bus could be affected. The issue is not one that can simply be patched because it lies in the CAN bus messaging protocol standard. Components that send too many error messages are disconnected from the CAN, so if attackers can spoof error messages to appear to be coming from a targeted component, that component could be shut off from the CAN.

What is at Stake -3

National Infrastructure Advisory Council Report - A Pre 9-11 Moment

"There is a narrow and fleeting window of opportunity before a watershed, 9/11-level cyber attack, [for the nation] to organize effectively and take bold action," said the US National Infrastructure Advisory Council report. The report lists 11 recommendations, including "establish separate, secure communications networks specifically designated for the most critical cyber networks; ... identify best-in-class scanning tools and assessment practices; ... [and] establish clear protocols to rapidly declassify cyber threat information."

Unique Point in Time – 1

- Software is the most important technology contributing to global high standards of living
- Cyber-attacks are increasing; 90% of attacks result from exploiting defects in software
- The cybersecurity spend estimated to grow to $170 billion by 2020
- 50 billion devices coming online during the next 5 years (IOT)
- Government is dependent on legacy code which is insecure

Unique Point in Time – 2

- GAO has added the Management of Information Technology (IT) Acquisitions and Operations to high risk list
- Many agencies are convinced that agile/Scrum is the silver bullet solution
- Majority of the government's IT spend is for operations & maintenance; majority of the maintenance spend is for corrective maintenance
- There is no evidence that software project management capability has improved during the past 50 years

Software Engineering's Persistent Problems – 1

- Exponential rise in cybersecurity vulnerabilities due to defective software
- Unacceptable cost, schedule, and quality performance of Enterprise Resource Planning (ERP) and legacy systems modernization projects
- The number one cost driver in software projects – cost of finding and fixing bugs (i.e. scrap and rework)
- Arbitrary and unrealistic schedules leading to a culture of "deliver now, fix later"

Software Engineering's Persistent Problems – 2

- Inability to scale software engineering methods even for medium size systems
- Lack of understanding of the impact of variation in individual productivity
- Absence of work place democracy and joy in work

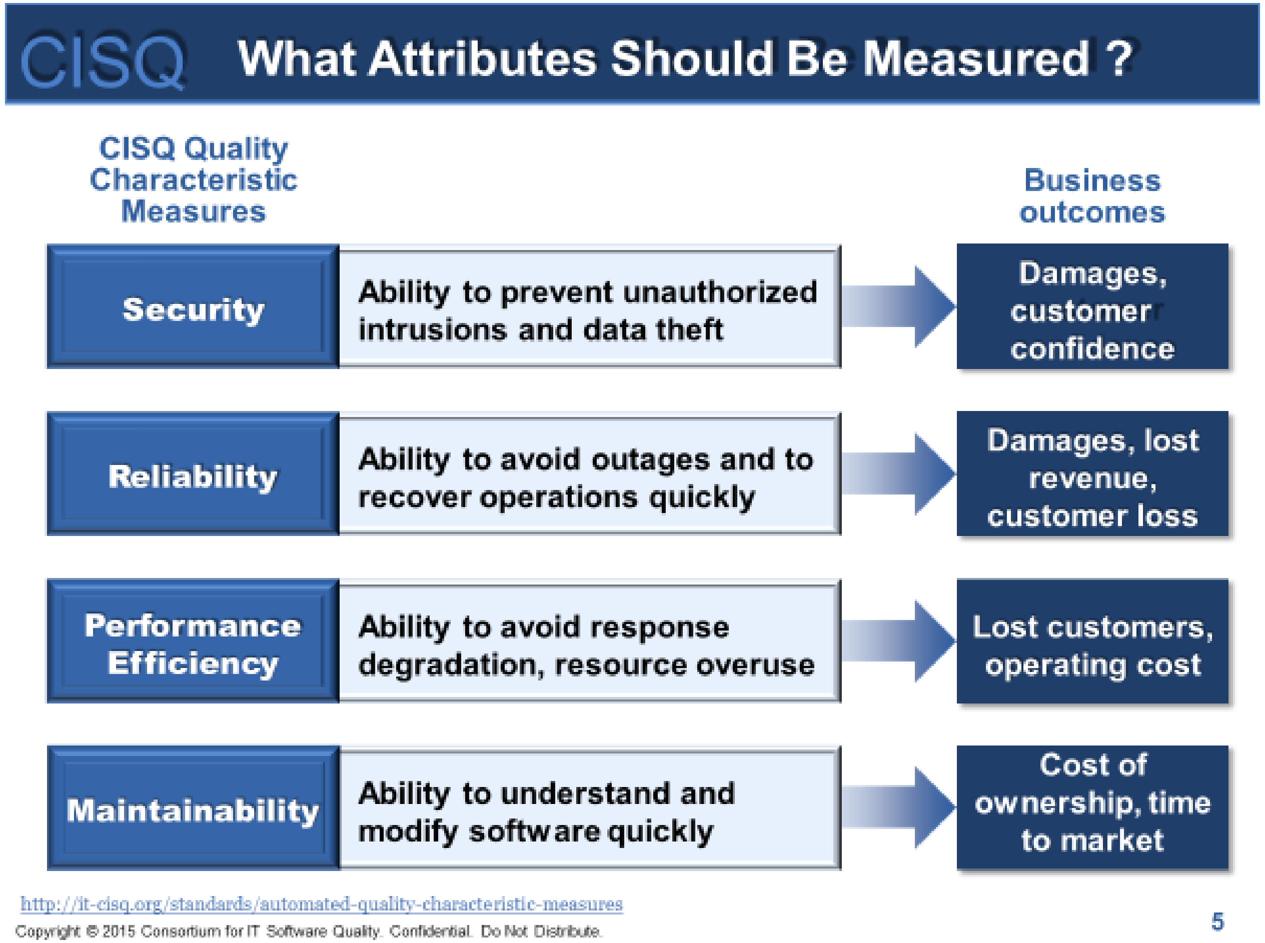
Managing for Secure Software Development – 1

- Staff the projects with team members who are formally trained in estimating, planning and tracking, and measuring and managing quality
- Start each project right with a jelled team capable of making disciplined commitments that the team knows it can meet
- Ensure team members are trained to conduct personal and peer reviews of all design and code artifacts in order to put the highest-quality code components into test, striving for 90 – 100% defect-free components with zero cybersecurity vulnerabilities

Managing for Secure Software Development – 2

- Ensure the review checklists incorporate specific checks for the types of vulnerabilities identified within “The OWASP Top Ten and CWE/SANS Top 25 Most Dangerous Programming Errors”
- Support the teams in collecting, analyzing, and reporting product and process data—size, effort, schedule, and quality
- Require teams to report status weekly
  - Precise and accurate data on planned versus actual size, effort, schedule, earned value and defects injected and removed

CISQ Quality Characteristic Measures

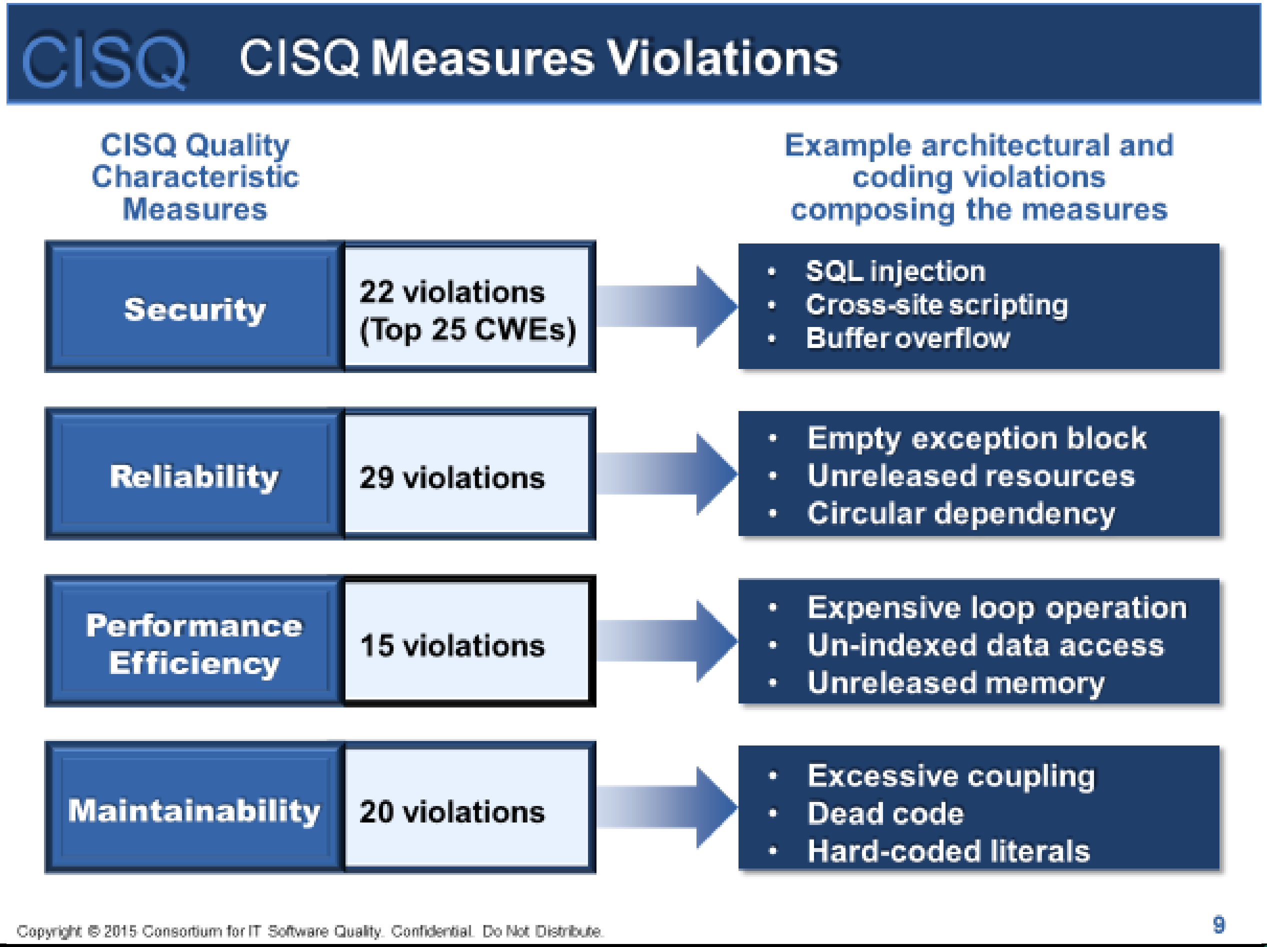


CISQ What Attributes Should Be Measured ?

CISQ Quality Characteristic Measures		Business outcomes
Security	Ability to prevent unauthorized intrusions and data theft	Damages, customer confidence
Reliability	Ability to avoid outages and to recover operations quickly	Damages, lost revenue, customer loss
Performance Efficiency	Ability to avoid response degradation, resource overuse	Lost customers, operating cost
Maintainability	Ability to understand and modify software quickly	Cost of ownership, time to market

http://it-cisq.org/standards/automated-quality-characteristic-measures

CISQ Quality Characteristic Violations



CISQ Measures Violations

CISQ Quality Characteristic Measures		Example architectural and coding violations composing the measures
Security	22 violations (Top 25 CWEs)	• SQL injection • Cross-site scripting • Buffer overflow
Reliability	29 violations	• Empty exception block • Unreleased resources • Circular dependency
Performance Efficiency	15 violations	• Expensive loop operation • Un-indexed data access • Unreleased memory
Maintainability	20 violations	• Excessive coupling • Dead code • Hard-coded literals

Principles for Successful Metrics Program

- Metrics derived from business and team goals
  - What do we need to know to make sure we are achieving the goals?
- Software teams must be focused on the "vital few" metrics and not the "trivial many"
- Size metric is essential to normalize data
- Operational definitions, precision, and accuracy of data are of utmost importance
- Unless data is collected and used by trained team members, it won't be useful
- Never use defect data for performance appraisal

Operational Definitions

Term	Definition
% schedule deviation	Actual schedule divided by planned schedule (by months) times 100 minus 100
Acceptance test defect	Defect found during the acceptance test of the product. This defect may be identified by the customer as well as the ISHPI team. Acceptance test period starts after the completion of system test of the product and ends when the code is deployed to production.
Cost of Quality (COQ)	$\text{COQ} = \frac{\text{effort in appraisal tasks + effort in prevention tasks + effort in failure tasks}}{\text{Total effort towards the commitment (including project management)}} \times 100$ • Appraisal tasks: Personal reviews, peer reviews, and first-time test execution • Prevention tasks: Training, postmortems, and causal analysis • Failure tasks: Analyzing and fixing defects found in reviews and testing
Defect density	The number of defects identified in a product divided by the size of the product component, expressed in standard measurement terms for that product (e.g., 1000 lines of new and changed code)
Earned value	Percentage of effort for tasks completed divided by total effort of the tasks
First Time Right (in acceptance test)	Deployed software changes, accepted by the customer the first time, with no further rework
Peer review yield	The percentage of the total defects that are found and removed in peer review
Personal review yield	The percentage of the total defects that are found and removed in personal review

Recommended Team Base Measures

Measure	Unit
Size	Lines of code
Task time	Minutes
Defects	Number
Task Completion	Yes/No

“Vital Few” Metrics to Reduce Vulnerabilities

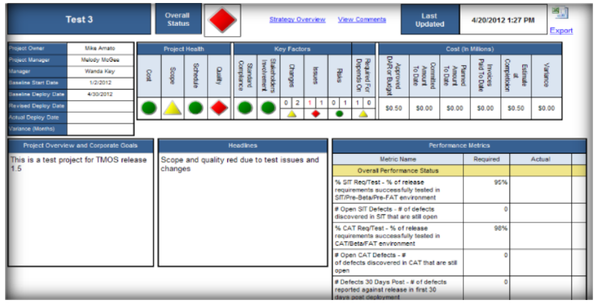
Performance Metrics	Leading Indicator?	Lagging Indicator?	Organization	Project	Increment	Component
Planned vs. actual size	Y		✓	✓	✓	✓
Planned vs. actual effort	Y		✓	✓	✓	✓
Planned vs. actual schedule	Y		✓	✓	✓	✓
Planned vs. actual earned value	Y				✓	
Planned vs. actual defect profile	Y	Y	✓	✓	✓	✓
Total Cost of Quality (COQ)	Y	Y	✓	✓	✓	
First Time Right in acceptance test		Y	✓	✓	✓	
Acceptance test defect density in delivered code		Y	✓	✓	✓	

Decision Process to Reduce Vulnerabilities

- Size, effort, schedule deviation
  - Review process adherence and test and rework due to poor quality
- Earned value deviation
  - Review hours worked against plan, task dependencies
  - "What if" scenario using process performance model
- Planned versus actual defect profile
  - Review personal and peer review yields – design, code
  - Estimate defect counts and defect densities for downstream steps using quality projection model

Project Management

Metrics SW



Test 3	Overall Status		Strategy Overview	View Comments	Last Updated	4/20/2012 1:27 PM	Export

Project Owner	Mike Amato
Project Manager	Melody McGee
Manager	Wanda Key
Baseline Start Date	1/2/2012
Baseline Deploy Date	4/30/2012
Revised Deploy Date	
Actual Deploy Date	
Variance (Months)	

Project Health				Key Factors						Cost (In Millions)					
Cost	Scope	Schedule	Quality	Standard Compliance	Stakeholders Involvement	Changes	Issues	Risks	Required For Depends On	Approved D/A/R or Budget	Committed Amount To Date	Planned Amount To Date	Invoices Paid To Date	Estimate at Completion	Variance
●	▲	●	◆	●	●	0 2 ▲	1 1 ◆	0 1 ●	1 0 ▲	$0.50	$0.00	$0.00	$0.00	$0.50	$0.00

Project Overview and Corporate Goals

This is a test project for TMOS release 1.5

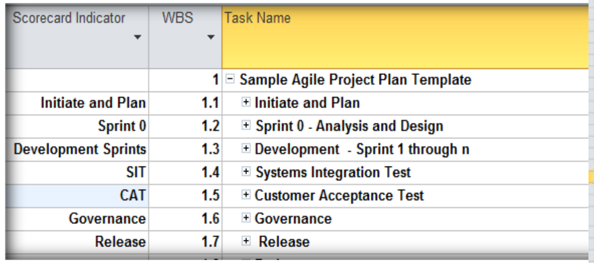
Headlines

Scope and quality red due to test issues and changes

Performance Metrics

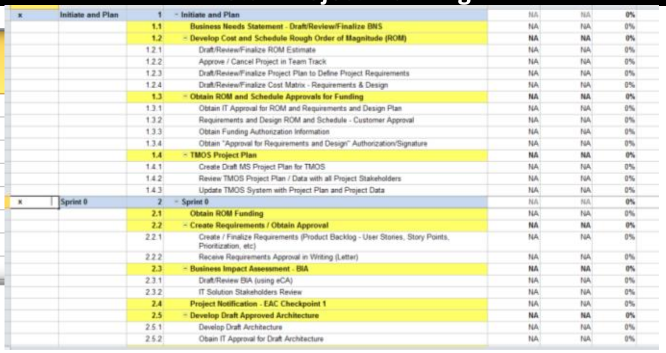
Metric Name	Required	Actual
Overall Performance Status		
% SIT Req/Test - % of release requirements successfully tested in SIT/Pre-Beta/Pre-FAT environment	95%	
# Open SIT Defects - # of defects discovered in SIT that are still open	0	
% CAT Req/Test - % of release requirements successfully tested in CAT/Beta/FAT environment	98%	
# Open CAT Defects - # of defects discovered in CAT that are still open	0	
# Defects 30 Days Post - # of defects reported against release in first 30 days post deployment	0	

Agile Milestone Tracking



Scorecard Indicator	WBS	Task Name
	1	⊟ Sample Agile Project Plan Template
Initiate and Plan	1.1	⊞ Initiate and Plan
Sprint 0	1.2	⊞ Sprint 0 - Analysis and Design
Development Sprints	1.3	⊞ Development - Sprint 1 through n
SIT	1.4	⊞ Systems Integration Test
CAT	1.5	⊞ Customer Acceptance Test
Governance	1.6	⊞ Governance
Release	1.7	⊞ Release

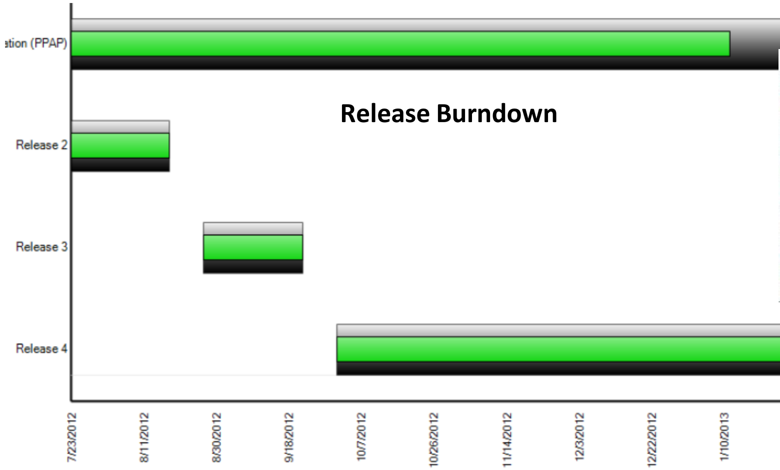
Detail Project Planning



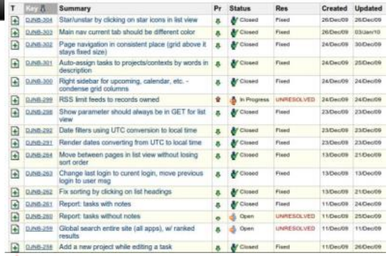
x	Initiate and Plan	1	- Initiate and Plan	NA	NA	0%
		1.1	Business Needs Statement - Draft/Review/Finalize BNS	NA	NA	0%
		1.2	- Develop Cost and Schedule Rough Order of Magnitude (ROM)	NA	NA	0%
		1.2.1	Draft/Review/Finalize ROM Estimate	NA	NA	0%
		1.2.2	Approve / Cancel Project in Team Track	NA	NA	0%
		1.2.3	Draft/Review/Finalize Project Plan to Define Project Requirements	NA	NA	0%
		1.2.4	Draft/Review/Finalize Cost Matrix - Requirements & Design	NA	NA	0%
		1.3	- Obtain ROM and Schedule Approvals for Funding	NA	NA	0%
		1.3.1	Obtain IT Approval for ROM and Requirements and Design Plan	NA	NA	0%
		1.3.2	Requirements and Design ROM and Schedule - Customer Approval	NA	NA	0%
		1.3.3	Obtain Funding Authorization Information	NA	NA	0%
		1.3.4	Obtain "Approval for Requirements and Design" Authorization/Signature	NA	NA	0%
		1.4	- TMOS Project Plan	NA	NA	0%
		1.4.1	Create Draft MS Project Plan for TMOS	NA	NA	0%
		1.4.2	Review TMOS Project Plan / Data with all Project Stakeholders	NA	NA	0%
		1.4.3	Update TMOS System with Project Plan and Project Data	NA	NA	0%
x	Sprint 0	2	- Sprint 0	NA	NA	0%
		2.1	Obtain ROM Funding	NA	NA	0%
		2.2	- Create Requirements / Obtain Approval	NA	NA	0%
		2.2.1	Create / Finalize Requirements (Product Backlog - User Stories, Story Points, Prioritization, etc)	NA	NA	0%
		2.2.2	Receive Requirements Approval in Writing (Letter)	NA	NA	0%
		2.3	- Business Impact Assessment - BIA	NA	NA	0%
		2.3.1	Draft/Review BIA (using eCA)	NA	NA	0%
		2.3.2	IT Solution Stakeholders Review	NA	NA	0%
		2.4	Project Notification - EAC Checkpoint 1	NA	NA	0%
		2.5	- Develop Draft Approved Architecture	NA	NA	0%
		2.5.1	Develop Draft Architecture	NA	NA	0%
		2.5.2	Obain IT Approval for Draft Architecture	NA	NA	0%

Senior Management View

Agile Project Team View

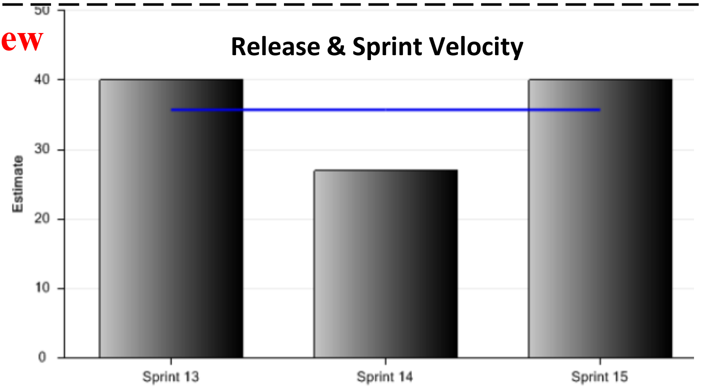


Release Burndown

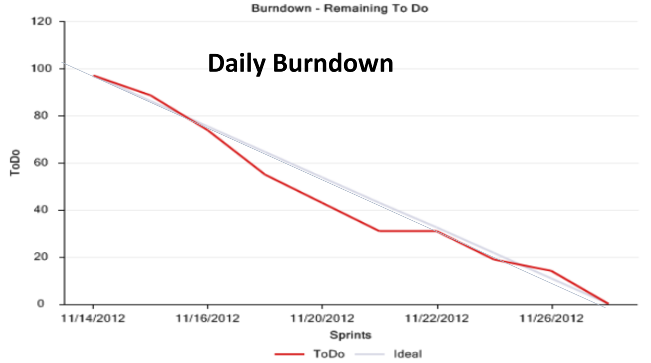


T	Key	Summary	Pr	Status	Res	Created	Updated
⊞	DJNB-304	Star/unstar by clicking on star icons in list view	⇩	Closed	Fixed	26/Dec/09	26/Dec/09
⊞	DJNB-303	Main nav current tab should be different color	⇩	Closed	Fixed	26/Dec/09	03/Jan/10
⊞	DJNB-302	Page navigation in consistent place (grid above it stays fixed size)	⇩	Closed	Fixed	24/Dec/09	30/Dec/09
⊞	DJNB-301	Auto-assign tasks to projects/contexts by words in description	⇩	Closed	Fixed	24/Dec/09	25/Dec/09
⊞	DJNB-300	Right sidebar for upcoming, calendar, etc. - condense grid columns	⇩	Closed	Fixed	24/Dec/09	24/Dec/09
⊞	DJNB-299	RSS limit feeds to records owned	⇧	In Progress	UNRESOLVED	24/Dec/09	24/Dec/09
⊞	DJNB-298	Show parameter should always be in GET for list view	⇩	Closed	Fixed	23/Dec/09	23/Dec/09
⊞	DJNB-292	Date filters using UTC conversion to local time	⇩	Closed	Fixed	23/Dec/09	23/Dec/09
⊞	DJNB-291	Render dates converting from UTC to local time	⇩	Closed	Fixed	23/Dec/09	23/Dec/09
⊞	DJNB-284	Move between pages in list view without losing sort order	⇩	Closed	Fixed	13/Dec/09	21/Dec/09
⊞	DJNB-263	Change last login to curent login, move previous login to user msg	⇩	Closed	Fixed	13/Dec/09	13/Dec/09
⊞	DJNB-262	Fix sorting by clicking on list headings	⇩	Closed	Fixed	13/Dec/09	21/Dec/09
⊞	DJNB-261	Report: tasks with notes	⇩	Closed	Fixed	11/Dec/09	24/Dec/09
⊞	DJNB-260	Report: tasks without notes	●	Open	UNRESOLVED	11/Dec/09	25/Dec/09
⊞	DJNB-259	Global search entire site (all apps), w/ ranked results	⇩	Open	UNRESOLVED	11/Dec/09	11/Dec/09
⊞	DJNB-258	Add a new project while editing a task	⇩	Closed	Fixed	11/Dec/09	26/Dec/09

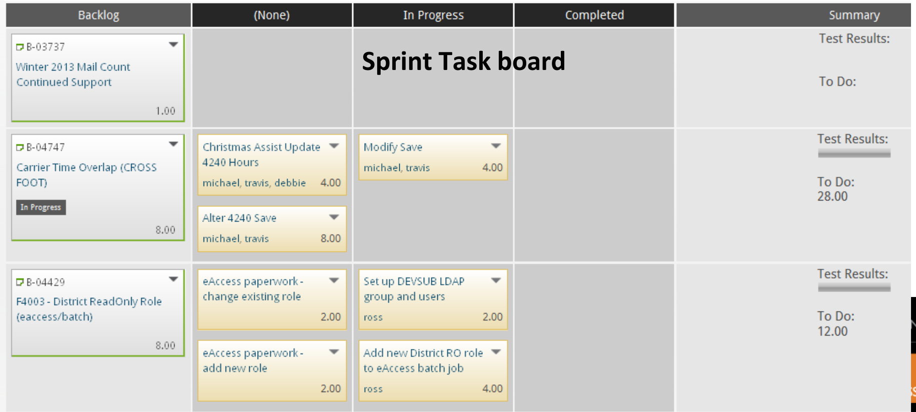
Issues / Risks



Release & Sprint Velocity



Daily Burndown

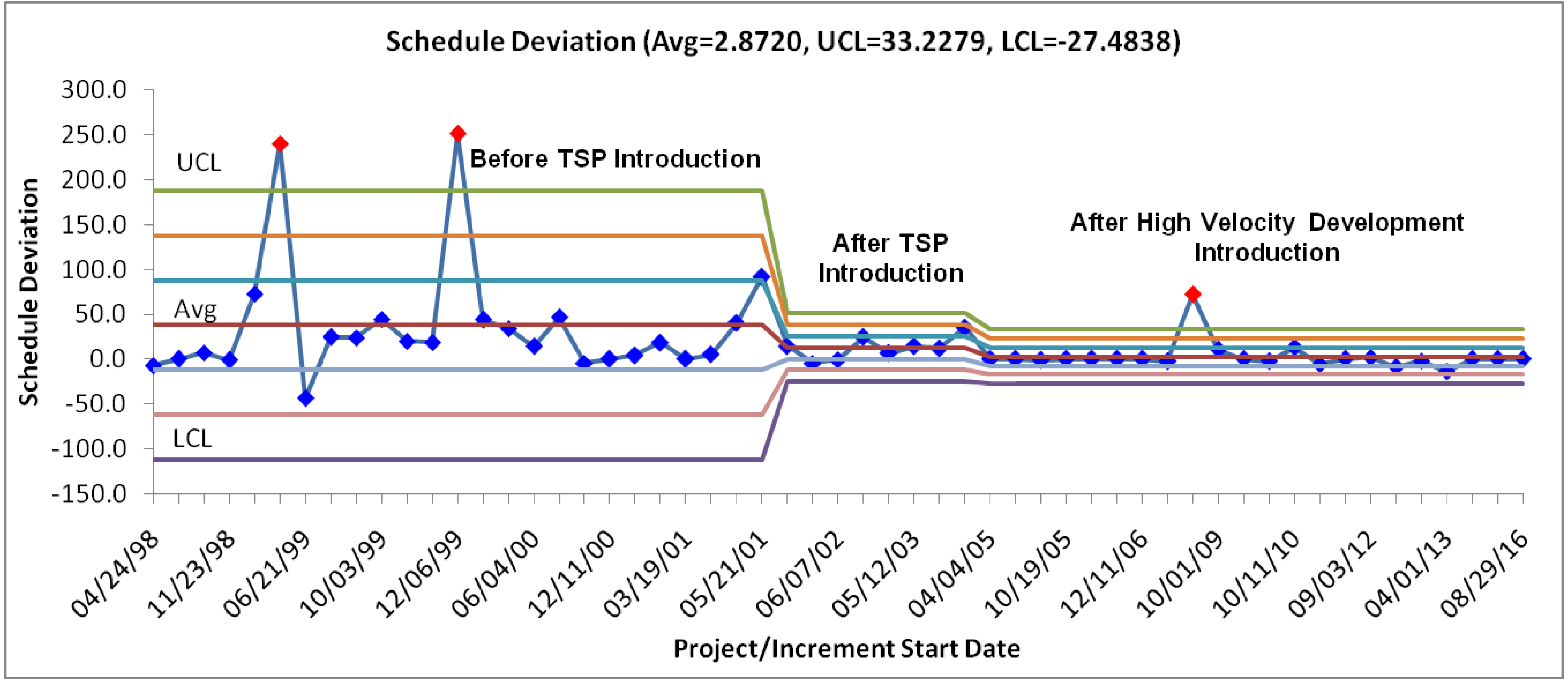


Backlog	(None)	In Progress	Completed	Summary
B-03737 Winter 2013 Mail Count Continued Support 1.00				Test Results: To Do:
B-04747 Carrier Time Overlap (CROSS FOOT) In Progress 8.00	Christmas Assist Update 4240 Hours michael, travis, debbie 4.00 Alter 4240 Save michael, travis 8.00	Modify Save michael, travis 4.00		Test Results: To Do: 28.00
B-04429 F4003 - District ReadOnly Role (eaccess/batch) 8.00	eAccess paperwork - change existing role 2.00 eAccess paperwork - add new role 2.00	Set up DEVSUB LDAP group and users ross 2.00 Add new District RO role to eAccess batch job ross 4.00		Test Results: To Do: 12.00

Sprint Task board

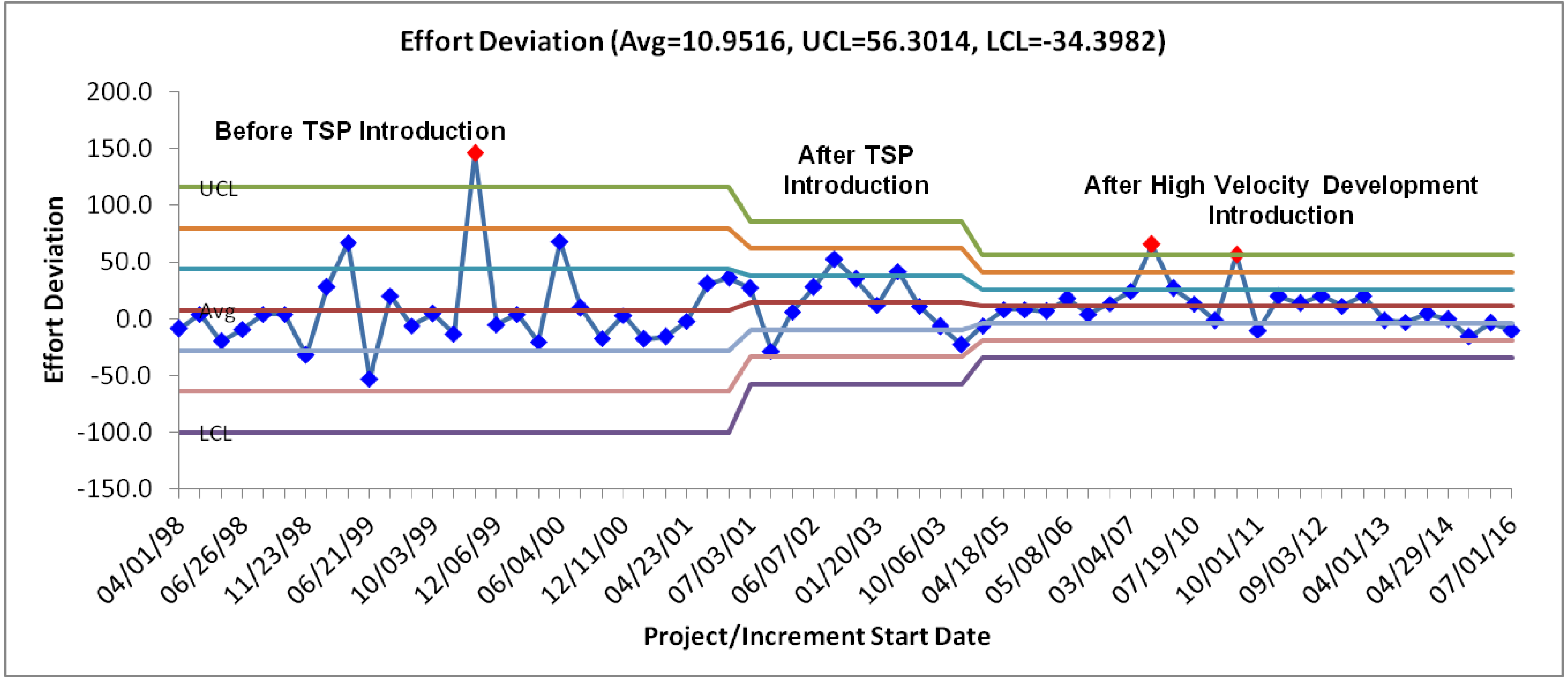
Organization Capabilities – 1

Schedule Deviation – Development Phases



Organization Capabilities – 2

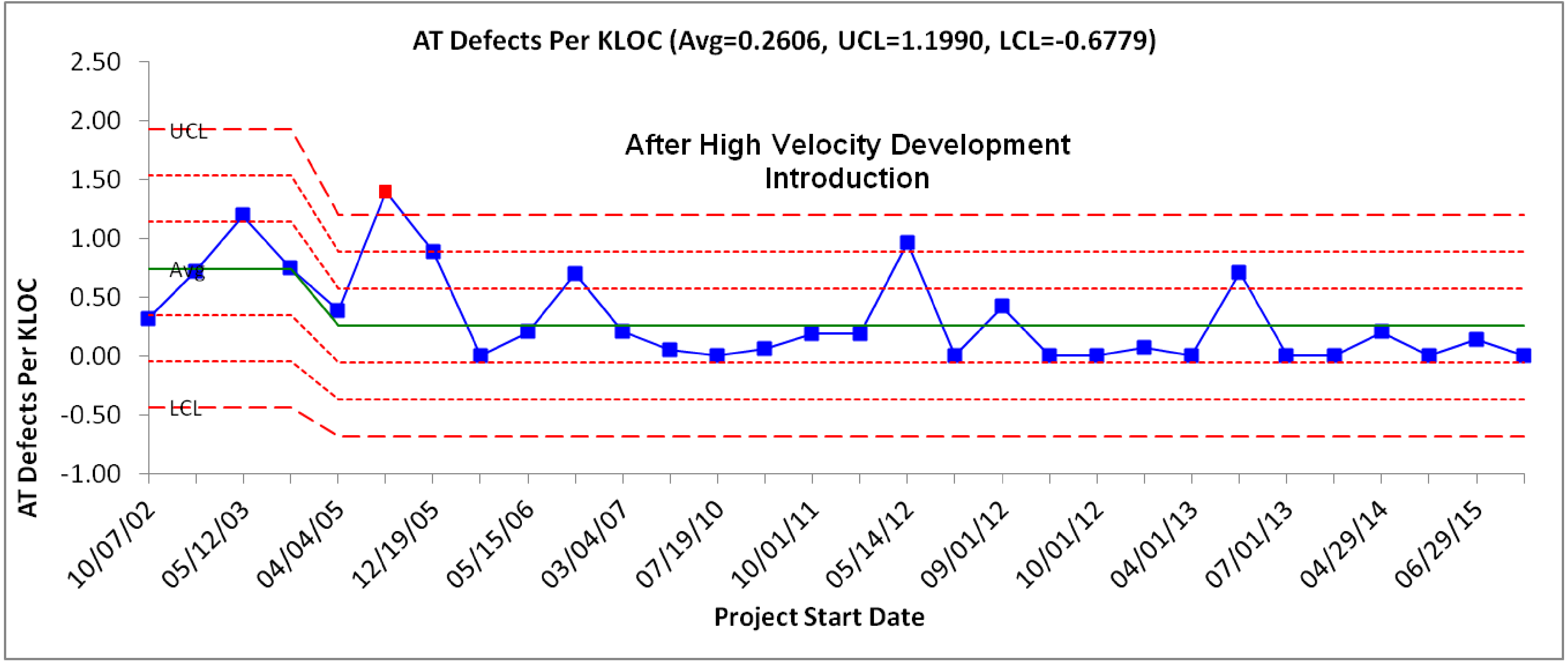
Effort Deviation – Development Phases



advanced technology • native know-how

Organization Capabilities – 3

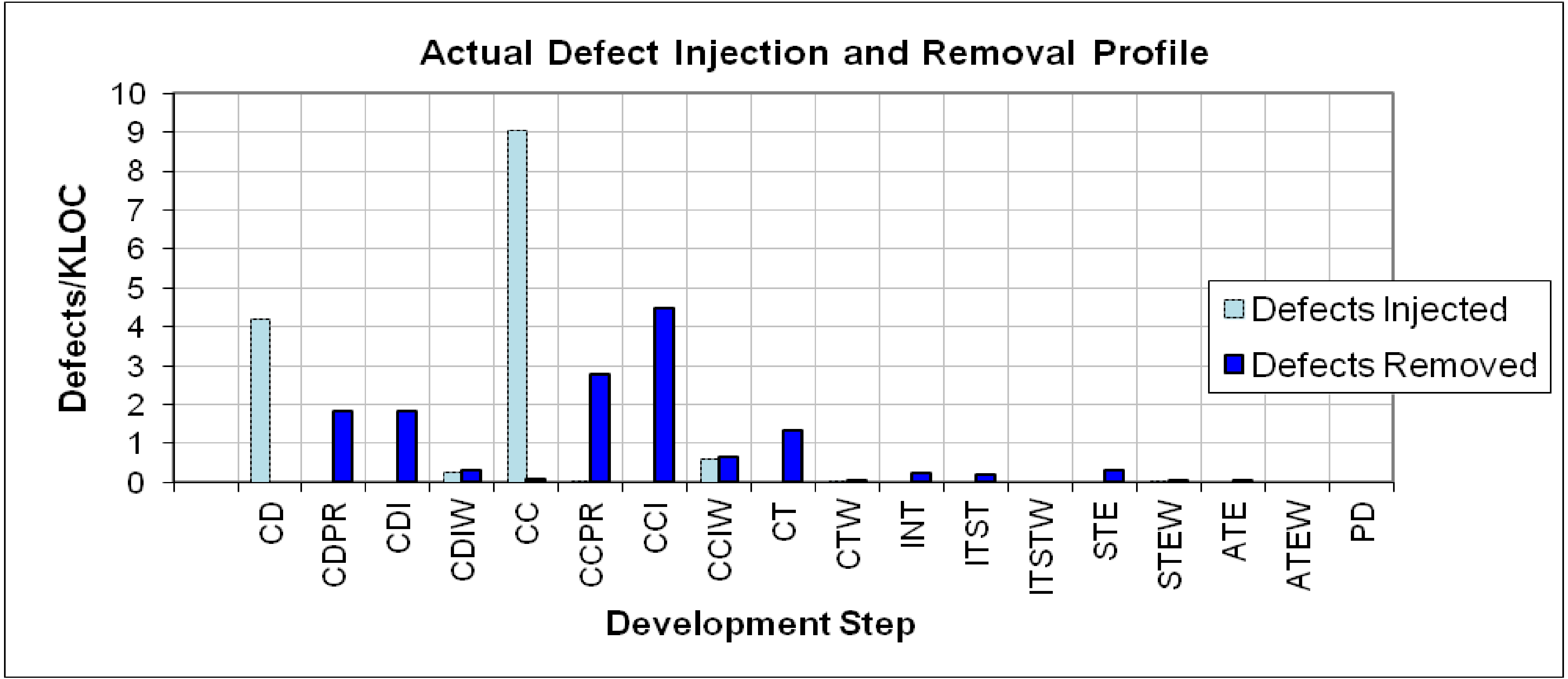
Acceptance Test Defect Density – Maintenance/Enhancement



Exp. 2017-03-07 / Appraisal #21757

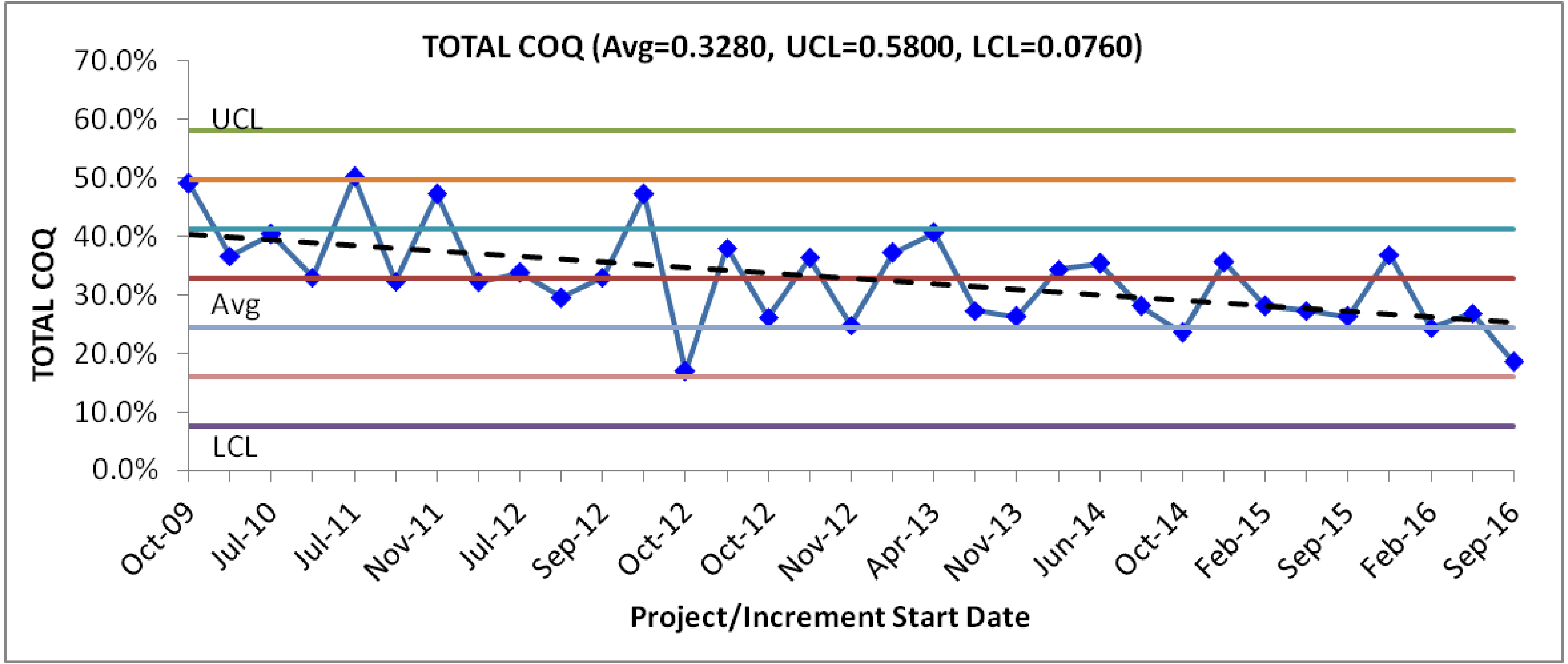
Organization Capabilities – 4

Defect Injection and Removal Profile



Organization Capabilities – 5

Total Cost of Quality



Appraisal
+ Failure
+ Prevention
= COQ

Appraisal

- Personal reviews
- Peer reviews
- All levels of first-time test execution

Failure

- Rework from reviews
- Rework from tests
- Regression testing

Organization Capabilities Summary

Industry, Best in Class, Organization

Performance Metrics	Industry Average	Best in Class Average	Org. Average
Schedule deviation	> 50%	6%	< 3%
Acceptance test defects in delivered product (100,000 Source Lines of Code)	> 100	55	< 27
% of design and code inspected	< 100%	100%	100%
Customer’s time to accept 100,000 LOC product	> 4 months	3.4 weeks	< 5 weeks
% of defects removed prior to system test	< 60%	70 – 95%	> 85%
% of development time in rework fixing system test defects	> 33%	2.7 – 10%	< 10%
Cost of quality	> 50%	29%	< 33%

Customer Benefits of Optimizing Process and Agile Development

- Frequent incremental deliveries that meet customer business goals
- Dramatically reduces and practically eliminates cyber security incidents attributable to poor quality software code
- Reduces software operations & maintenance (O&M) costs by more than half
- Fewer bugs to fix in production software making available a larger percentage of O&M spend for new features and enhancements
- Enables lifetime warranty

Main Points

- **Defective software** is insecure
  - "If you have a quality problem, you have a security problem"
- Cannot **rely on testing alone** to find and remove software defects
  - Common misconception – "if it passes test, it must be OK"
  - Root cause of "Deliver now, Fix later" culture, technical debt, increase in total ownership cost in many agile projects
- Move from reactive to proactive – **threat detection to threat prevention**
- **Reducing vulnerabilities** - number one goal for every agile software team
  - Manage software by managing software quality
- Need **transformation** – organization, team, individual
  - Combine high maturity CMMI Maturity Level 5 with agile team process to create secure software by self-managed teams and empowered developers with certifications

Takeaways/“ASKs”

- Act with a **sense of urgency** to move from reactive “patch and pray” to proactive secure software development
- Begin **transformation** of organization, teams, and individuals to consistently deliver software which is secure from cyber attacks
- Manage secure software development by **managing software quality**
- Focus on the **“vital few”** metrics and not the “trivial many”
- Support teams in **reducing vulnerabilities** as the number one project goal
- Develop industry/government/academic coalition led by industry to address cybersecurity **“skills gap”** and talent pipeline

References

NIST Publication NISTIR 8151

http://nvlpubs.nist.gov/nistpubs/ir/2016/NIST.IR.8151.pdf

Dramatically Reducing Software Vulnerabilities

Report to the White House Office of Science and Technology Policy

ISHIPI Technical Report, 22 July 2016

Barti Perini, Stephen Shook and Girish Seshagiri

Reducing Software Vulnerabilities – The Number One Goal for Every Software Development Organization, Team, and Individual

Contact

Girish Seshagiri

girish.seshagiri@ishpi.net

(703) 426-2790