Performance Metrics That Matter: Eliminating Surprises in Agile Projects

Girish Seshagiri, Ishpi Information Technologies, Inc.

Software Solutions Conference 2015

November 16–18, 2015

“If We Eliminate the Monthly Status Report What Do We Replace It With?”

Agenda

Software Engineering’s Persistent Problems

Common Misconceptions of Software

Immutable Laws of Software Development

Performance Metrics That Matter

Main Points

We do awesome things in IT. Our **problems persist**.

Status quo is not acceptable with the threat of **cyber attacks**

We need

- to shift our focus to the **individual developer/engineer** trained in quality methods
- to cease dependence on test as the principal defect removal method
- the **“vital few” performance metrics** that really matter and help us manage the software work by managing quality

IT Acquisition Failures Land On High-risk List

GAO Report

"**Too frequently**, federal IT investments fail to be completed or incur cost overruns and schedule slippages while contributing little to mission-related outcomes,"

"Unfortunately, **fairly consistently**, we find problems with these projects. And these seem to center on a lack of **discipline** and effective **management practices**, the need for **improvements in project planning**, and poor program oversight in governance."

Software Engineering's Persistent Problems - 1

Exponential rise in cybersecurity vulnerabilities due to **defective software**

Unacceptable cost, schedule, and quality performance of legacy systems **modernization** and Enterprise Resource Planning (**ERP**) projects

Software Engineering's Persistent Problems - 2

Cost of finding and fixing software bugs (i.e. **scrap and rework**) the number one cost driver in software projects

Arbitrary and **unrealistic schedules** leading to a culture of "**deliver now, fix later**"

Software Engineering's Persistent Problems - 3

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**

The Appetite for Assured Software

The organizational appetite for assured software is driven by the net losses realized from compromised software

The consumer has been living with nearly **60 years of poorly developed** and incompetent software.

Hundreds of millions of dollars are spent annually on post software compromise and incident recovery, lost opportunities and productivity (ask me).

Insecure software represents a pervasive kinetic threat to critical infrastructure and our way of life…..make no mistake about it.

The prudent approach is to take a proactive one. That is, software assurance measures must be a top integration priority in the enterprise cyber security risk management schema.

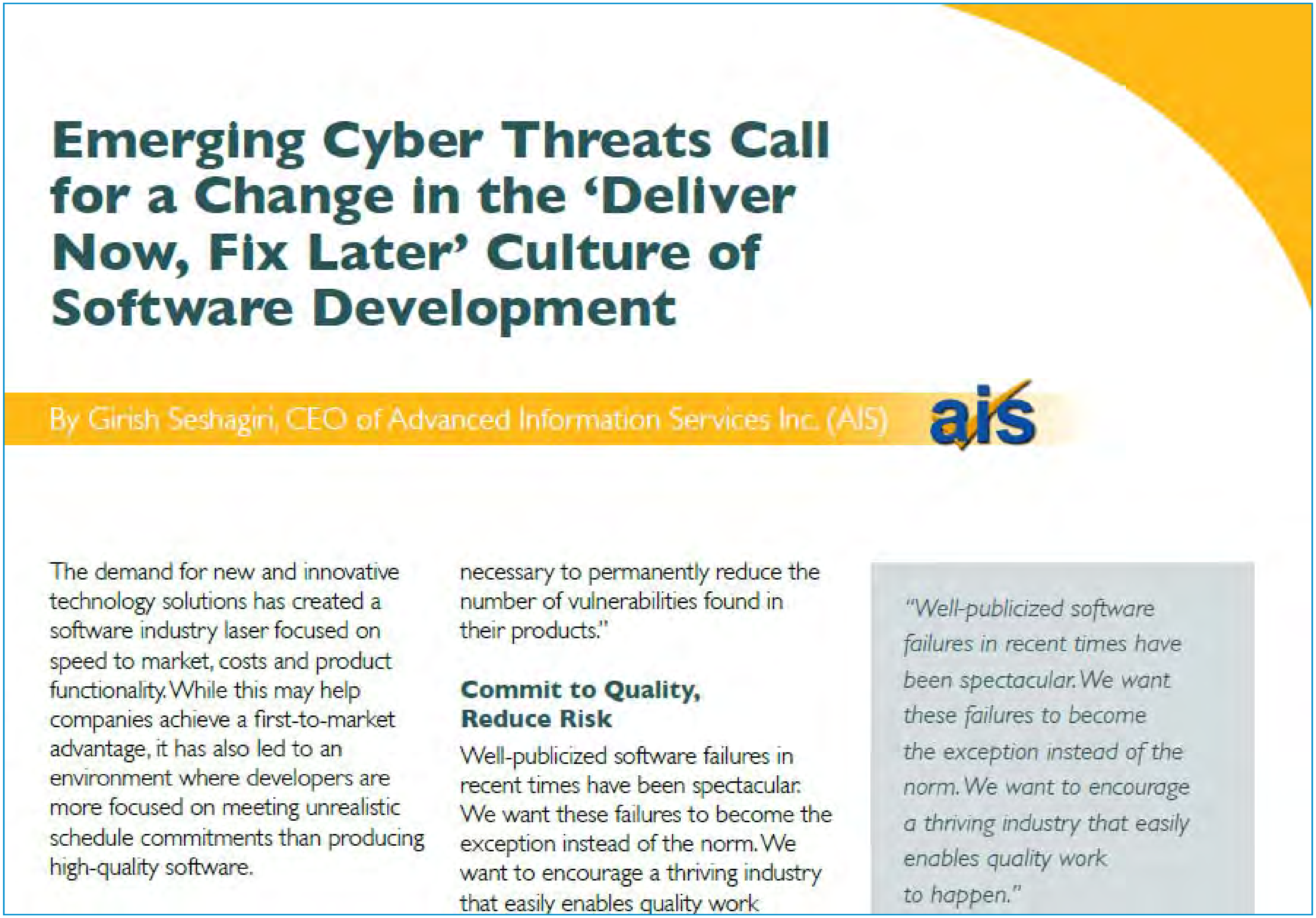
SWAMP Webinar – Jerry L. Davis, Chief Information Officer, NASA

By the Numbers

Feel my pain. Lack of a good software assurance program is a painful experience

At one time – 127 applications were tested and;

- 81 (64%) contained high vulnerabilities that facilitated exposure of sensitive data or **system take over**;
- 45 applications (36%) exposed **Personally Identifiable Information** (PII)
- At another time – 50 applications were tested and;
- 41 applications (82%) hosted **OWASP top 10 defects**
- 5 applications (10%) taken offline due to high risk
- 19 (38%) contained high vulnerabilities that facilitated exposure of sensitive data or system take over
- 12 applications (24%) exposed PII



Emerging Cyber Threats Call for a Change in the 'Deliver Now, Fix Later' Culture of Software Development

By Girish Seshagiri, CEO of Advanced Information Services Inc. (AIS)

The demand for new and innovative technology solutions has created a software industry laser focused on speed to market, costs and product functionality. While this may help companies achieve a first-to-market advantage, it has also led to an environment where developers are more focused on meeting unrealistic schedule commitments than producing high-quality software.

necessary to permanently reduce the number of vulnerabilities found in their products."

Commit to Quality, Reduce Risk

Well-publicized software failures in recent times have been spectacular. We want these failures to become the exception instead of the norm. We want to encourage a thriving industry that easily enables quality work

> *"Well-publicized software failures in recent times have been spectacular. We want these failures to become the exception instead of the norm. We want to encourage a thriving industry that easily enables quality work to happen."*

The Application Security Industry
Is Now Bigger Than
The Application Development Industry

Common Misconceptions -1

We must start with firm requirements

If it passes test, it must be OK

Software quality can’t be measured

The problems are technical

We need better people

Software management is different

***Managing the Software Process,* Watts Humphrey, Addison**

Common Misconceptions – 2

Maturity levels guarantee results

Maturity level 3 is all that is needed

Higher maturity levels add to cost

Higher maturity levels are needed only for safety critical or business mission critical systems

If it is “agile” or “lean”, it is good

What we need are lean processes

Maturity level 5 is the end

The Real Question

Whose Process Is It?

Why? - 1

Why do development teams agree to **delivery schedule they know they can’t meet?**

Why don’t C-level executives realize that poor **quality performance is the root cause** of most software cost and schedule problems?

Why doesn’t the government **hold contractors liable** for software defects and vulnerabilities?

Why? - 2

Why does the software applications development industry believe that **quality increases costs and schedule?**

Why do we continue to rely on **test as the principal defect removal** method?

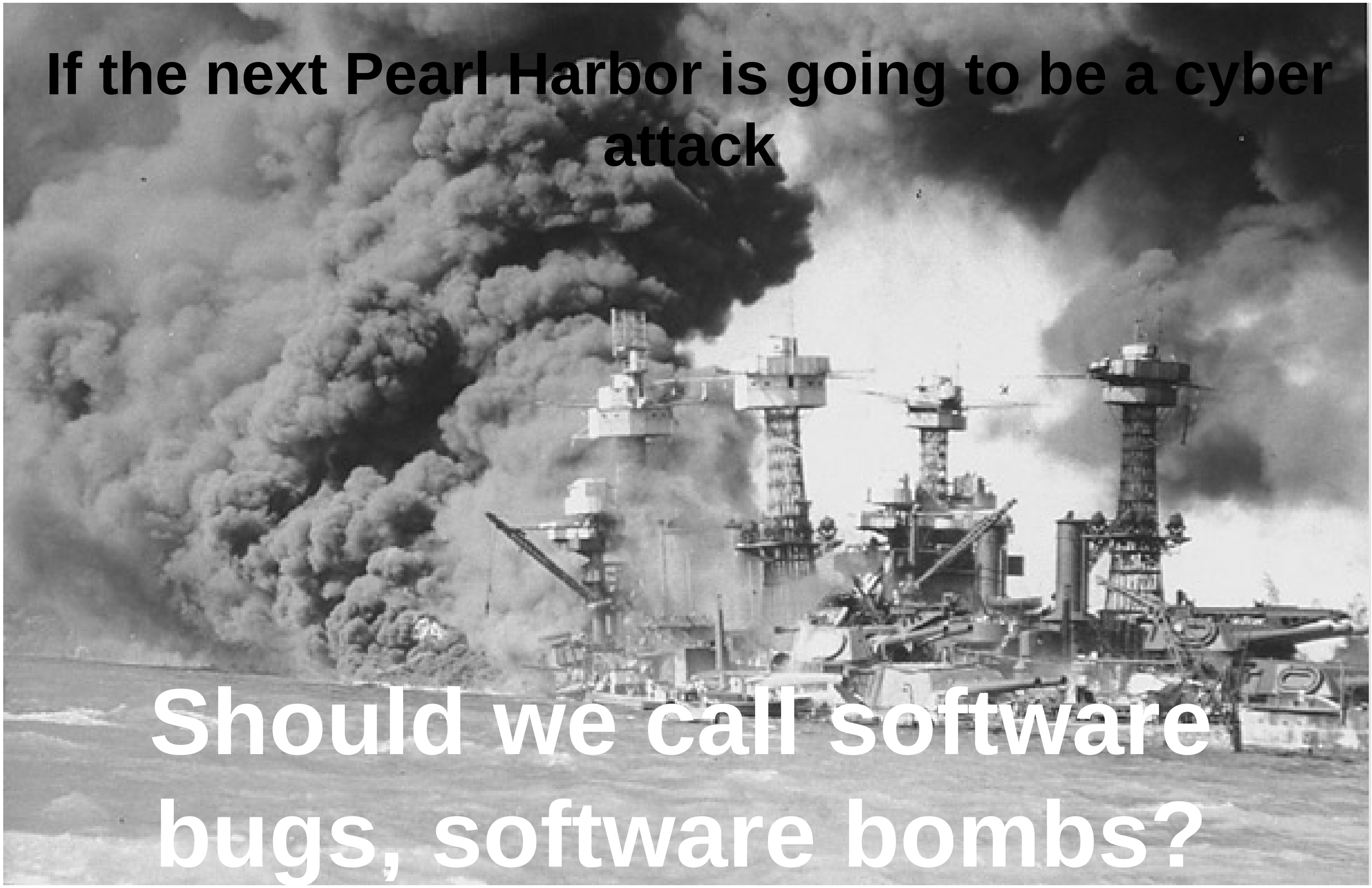
Why do we continue to rely on monthly status reporting when projects get to be **one year late one day at a time?**

Why? - 3

Why don’t we call technical debt for what it really is, **“malpractice”?**

Why don’t we charge the **cost** of post release bug fixing (corrective **maintenance**) to **development** where it belongs?

Why do we approach software and supply chain assurance as a technical problem and not the **management problem** that it is?



If the next Pearl Harbor is going to be a cyber attack

Should we call software bugs, software bombs?

Have You Considered?

Quality work is more **predictable**

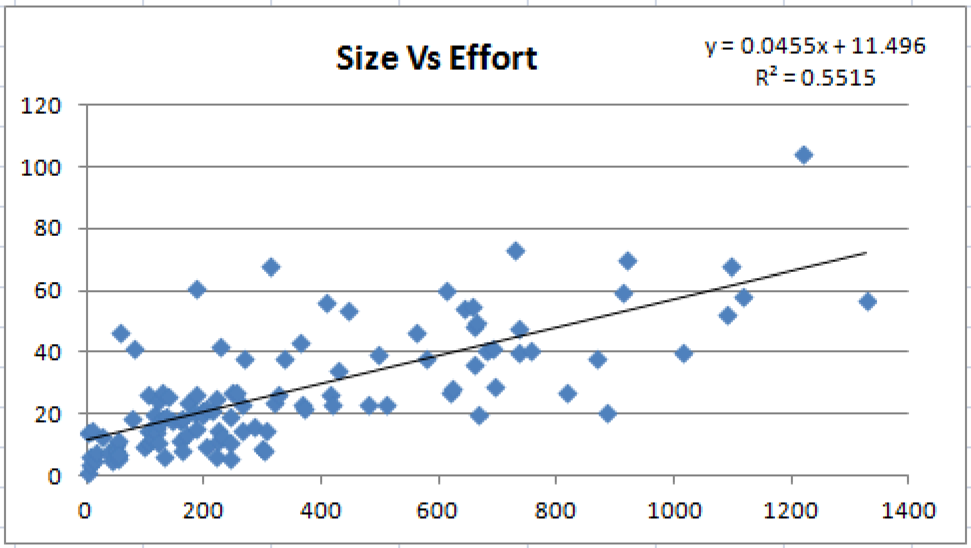
Unhappy people rarely do quality work

Without quality, agility is in **name only**

Quality **without numbers** is just talk

Immutable Laws of Software Development – 1

The number of development hours will be directly proportional to the size of the software product



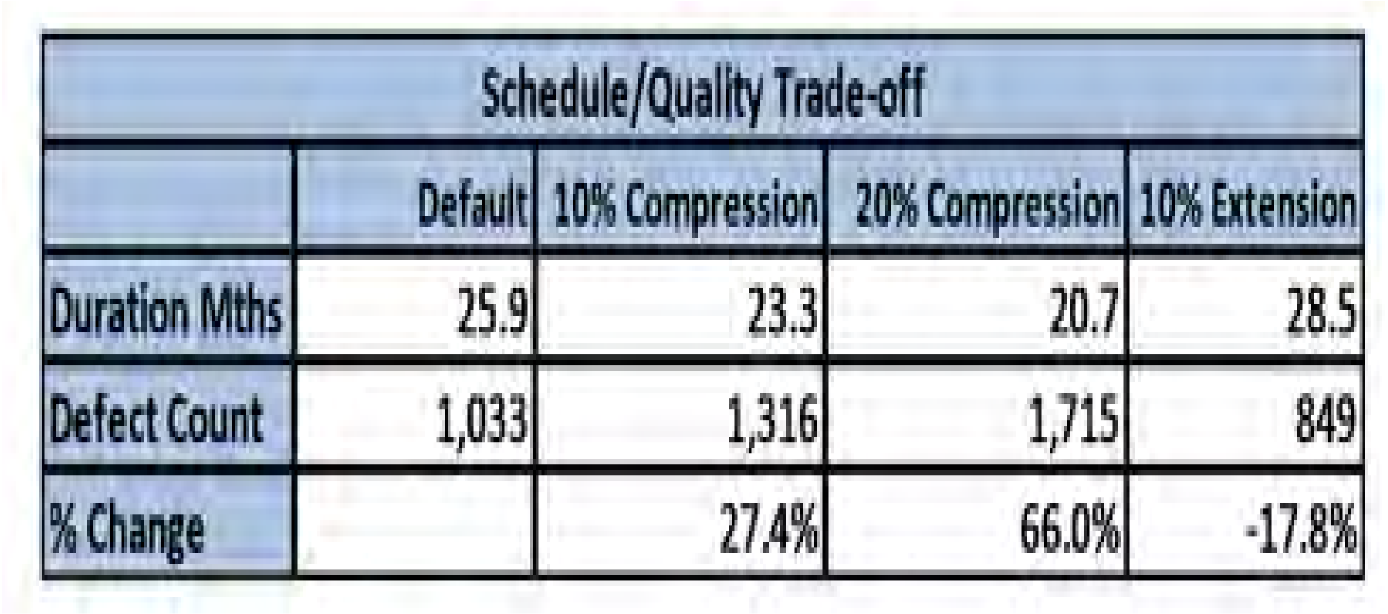
Immutable Laws of Software Development – 2

When acquirers and vendors both “guess” as to how long a project should take, the acquirers’ “guess” will always win

- Customers’ Dilemma
  - Want their product now at zero cost.
  - Due to time-to-market pressures, time frames are arbitrary and unrealistic for the software team to produce a product that works.
- Developers’ Choices
  - Try to “guess” what it would take to win the business.
  - Or make a commitment based on a plan and what the organization can do based on organization historic data.

Immutable Laws of Software Development – 3

When management compresses schedule arbitrarily, the project will end up taking longer



Schedule/Quality Trade-off				
	Default	10% Compression	20% Compression	10% Extension
Duration Mths	25.9	23.3	20.7	28.5
Defect Count	1,033	1,316	1,715	849
% Change		27.4%	66.0%	-17.8%

Immutable Laws of Software Development – 5

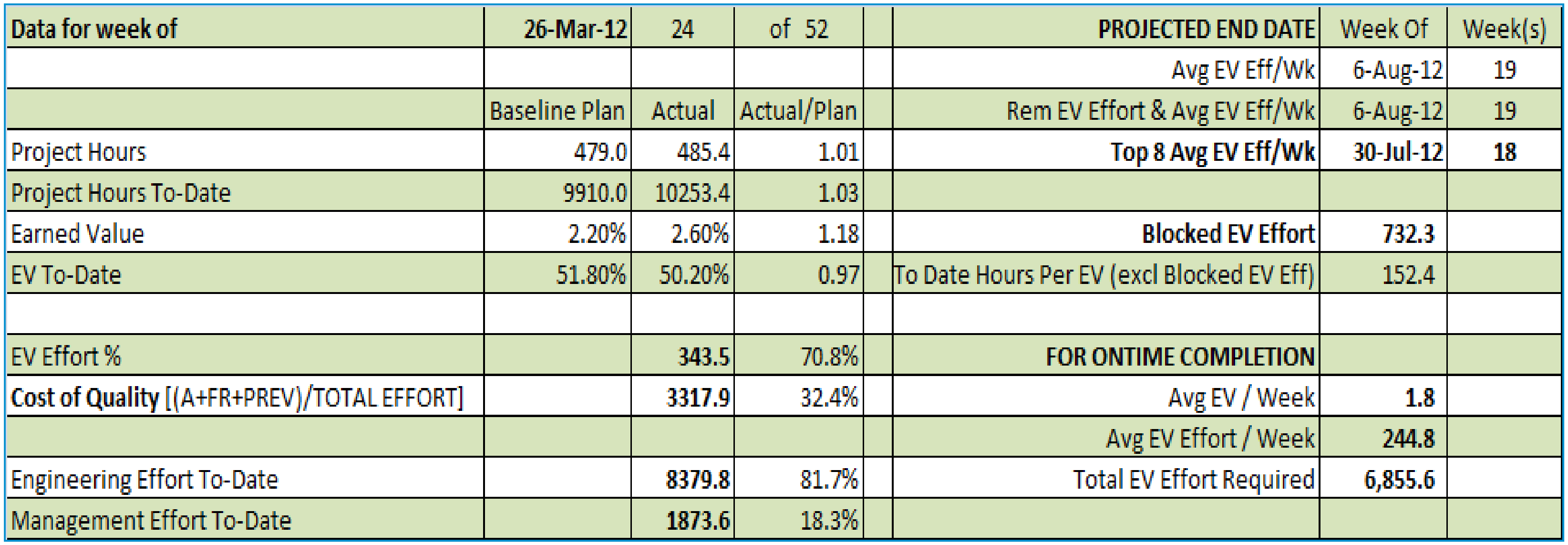
When poor quality impacts schedule, schedule problems will end up as quality disasters

Maryland officials were warned for a year of problems with online health-insurance site

"We didn't know it would be broken when we turned it on"

Immutable Laws of Software Development – 6

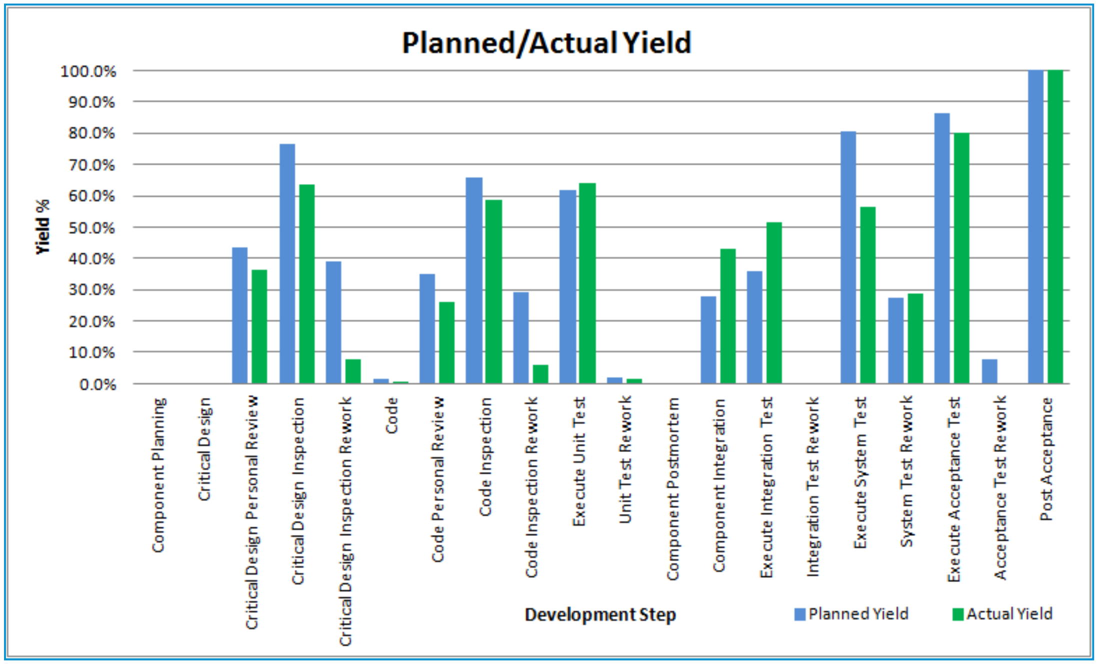
The less you know about a project during development, the more you will be forced to know later



Data for week of	**26-Mar-12**	24	of 52		**PROJECTED END DATE**	Week Of	Week(s)
					Avg EV Eff/Wk	6-Aug-12	19
	Baseline Plan	Actual	Actual/Plan		Rem EV Effort & Avg EV Eff/Wk	6-Aug-12	19
Project Hours	479.0	485.4	1.01		**Top 8 Avg EV Eff/Wk**	**30-Jul-12**	**18**
Project Hours To-Date	9910.0	10253.4	1.03				
Earned Value	2.20%	2.60%	1.18		**Blocked EV Effort**	**732.3**	
EV To-Date	51.80%	50.20%	0.97		To Date Hours Per EV (excl Blocked EV Eff)	152.4	
EV Effort %		**343.5**	70.8%		**FOR ONTIME COMPLETION**		
Cost of Quality [(A+FR+PREV)/TOTAL EFFORT]		**3317.9**	32.4%		Avg EV / Week	**1.8**	
					Avg EV Effort / Week	**244.8**	
Engineering Effort To-Date		**8379.8**	81.7%		Total EV Effort Required	**6,855.6**	
Management Effort To-Date		**1873.6**	18.3%				

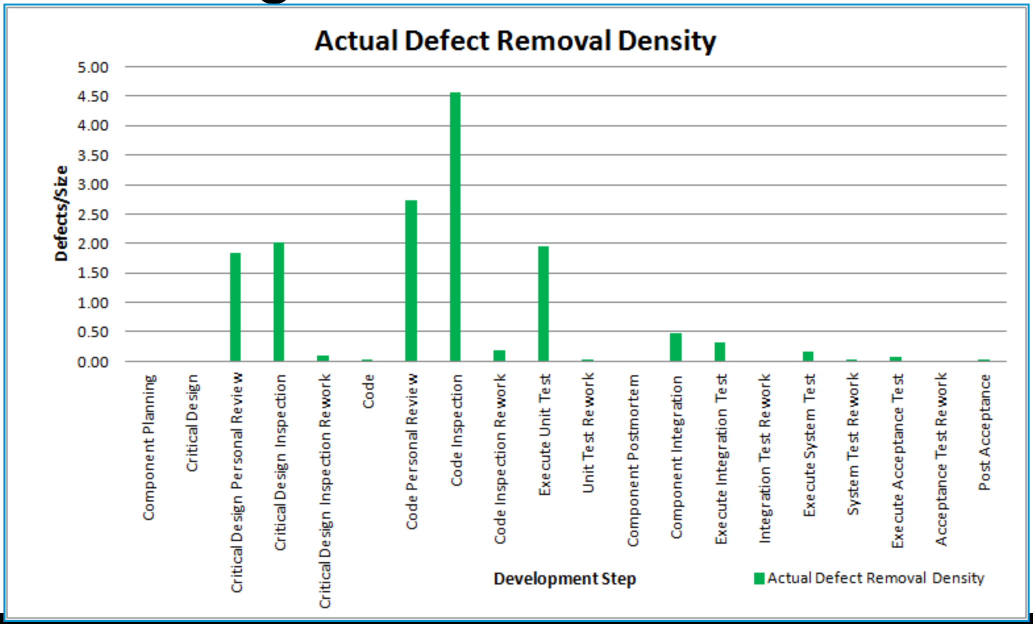
Immutable Laws of Software Development – 7

When test is the principal defect removal method during development, corrective maintenance will account for the majority of the maintenance spend



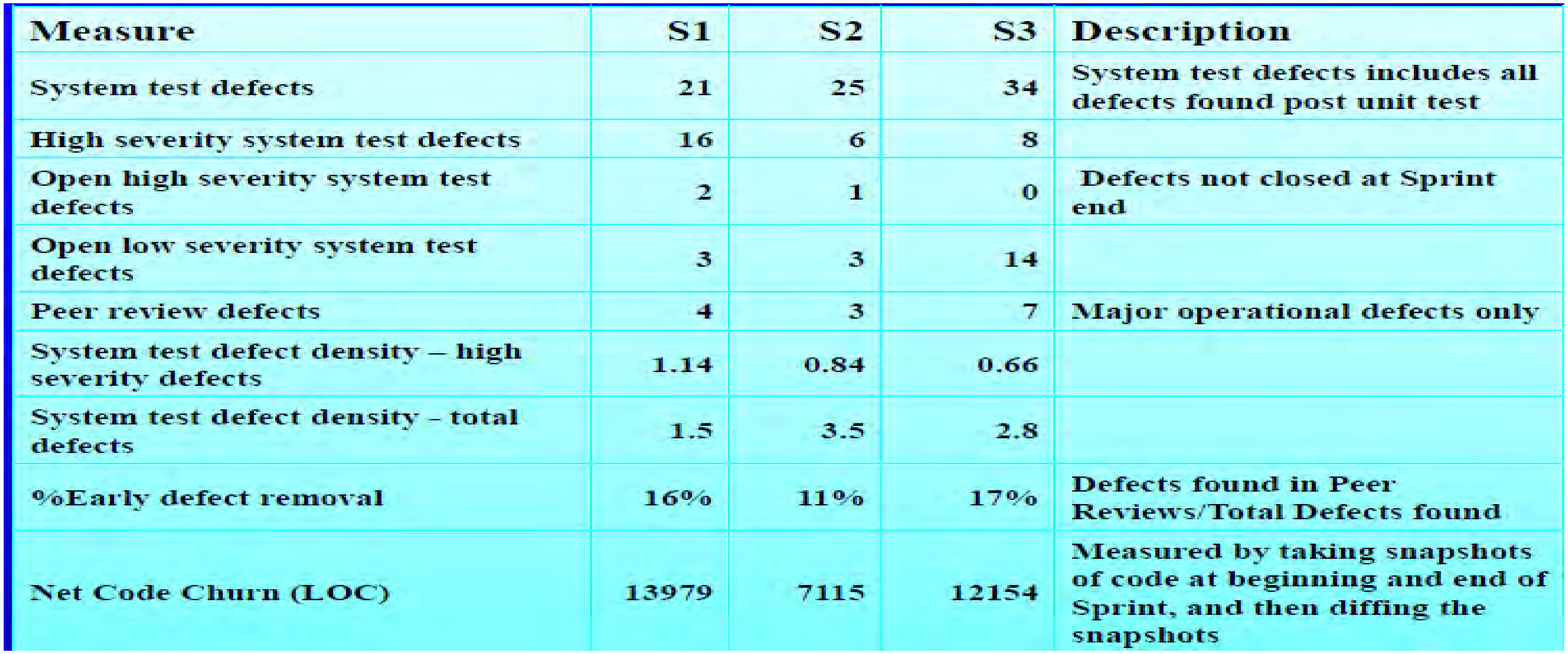
Immutable Laws of Software Development – 8

The number of defects found in production use will be inversely proportional to the percent of defects removed prior to integration, system, and acceptance testing



Immutable Laws of Software Development – 9

The amount of technical debt is inversely proportional to the length of the agile sprint

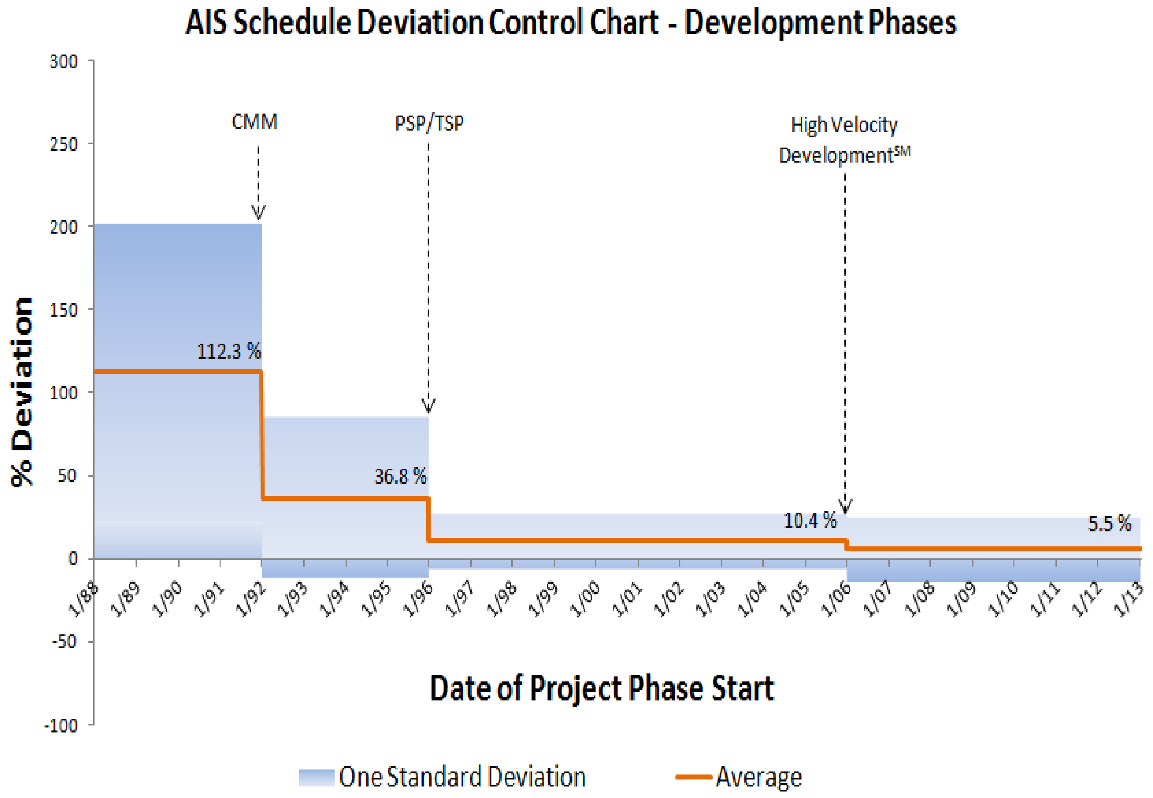


Measure	S1	S2	S3	Description
System test defects	21	25	34	System test defects includes all defects found post unit test
High severity system test defects	16	6	8	
Open high severity system test defects	2	1	0	Defects not closed at Sprint end
Open low severity system test defects	3	3	14	
Peer review defects	4	3	7	Major operational defects only
System test defect density – high severity defects	1.14	0.84	0.66	
System test defect density - total defects	1.5	3.5	2.8	
%Early defect removal	16%	11%	17%	Defects found in Peer Reviews/Total Defects found
Net Code Churn (LOC)	13979	7115	12154	Measured by taking snapshots of code at beginning and end of Sprint, and then diffing the snapshots

Immutable Laws of Software Development – 12

Insanity is doing the same thing over and over and firing the project manager or the contractor when you don't get the results you expected

Results
Organization History



Constancy of Purpose

- Make schedule and quality predictable. Since the introduction of HVD, average schedule deviation has been less than 5%

Focus on quality:

- Removal of defects at the earliest opportunity, before test where they are the least costly to remove
- Quality is more predictable
- Unhappy people rarely do quality work

On the project for the Selective Service System, we were able to deliver 680,000 lines of source code where:

- Zero security vulnerabilities were found in pen testing
- Production deployment 2 weeks ahead of schedule
- Schedule deviation less than 2% throughout 150 weeks of development
- Zero system downtime in over 3 years of production use due to software defects

Results
Recently Completed Project

Component yield: 92.3%

- Percent of defects introduced during development that were removed during development (before integration or system test)

Cost of Quality: 34.9% [Industry average: >50%]

- Effort in Appraisal, Failure and Prevention tasks

Time to Accept Deliverables:

- 1.3 Weeks per 100,000 SLOC [Industry average: >16 Weeks]
- 0.21 Defects/KLOC [Industry average: 4.73]

Schedule deviation: 4 weeks ahead of schedule

- 2.5% ahead [Industry average: 27% behind]

Results
New Team Member

43 Components

Size estimate error: 9%

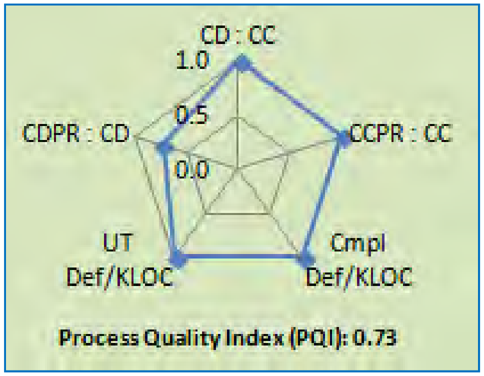
Effort estimate error: 13%

Process Quality Index (PQI): 0.73

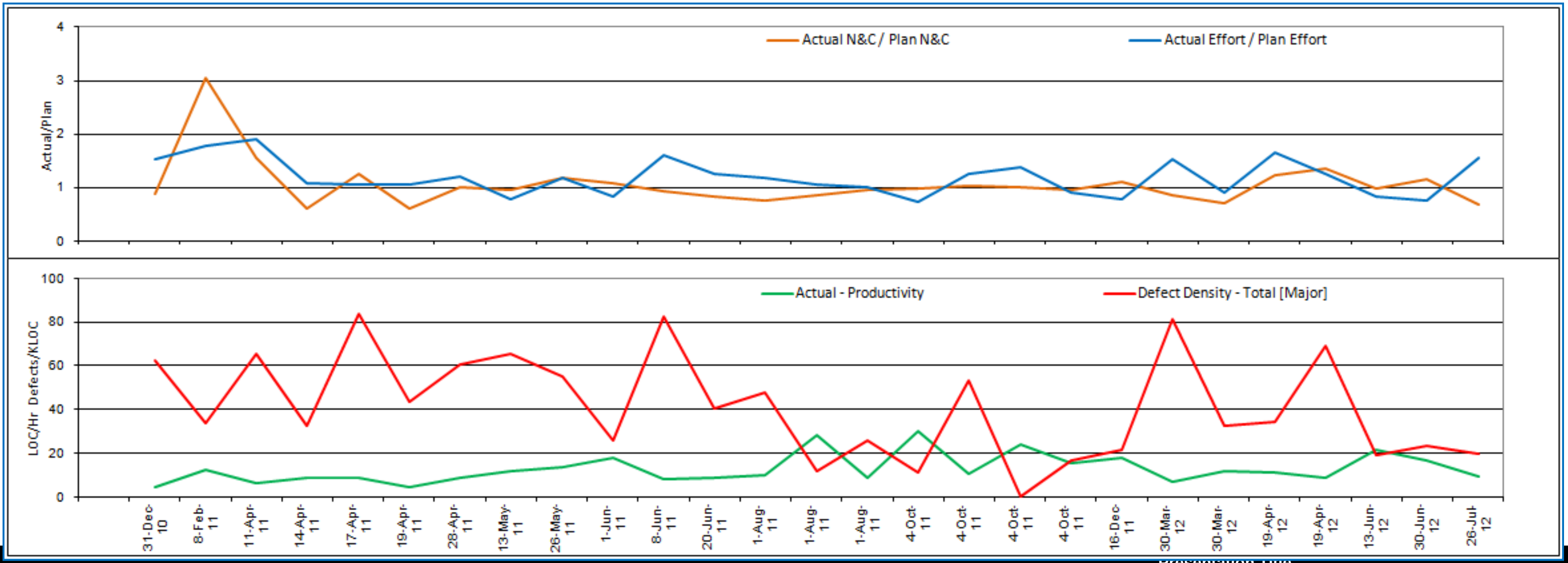
SEI data: PQI > 0.4 indicates high quality component

Component yield: 93.5%

Percent of defects introduced during development that were removed during development (before integration or system test)



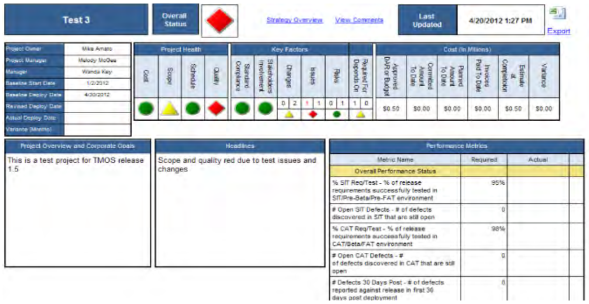
Process Quality Index (PQI): 0.73



Performance Metrics That Matter Benchmarking

	Industry Average	Company Average
Schedule deviation	> 50%	< 6%
No. of defects in delivered product (Size: 100,000 Source Lines of Code)	> 100	< 15
Customer's time to accept 100,000 SLOC product	> 4 Months	< 5 Weeks
% of design and code inspected	100	100
% of defects removed prior to system test	< 60%	> 85%
% of development time fixing system test defects	> 33%	< 10%
Cost of quality	> 50%	< 35%
Warranty on products	?	Lifetime

Agile Project Management Example



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 DAR 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

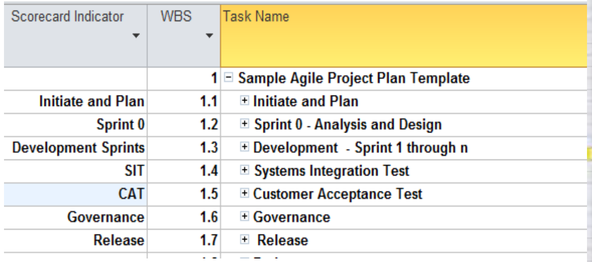
This is a test project for TMOS release 1.5

Headlines

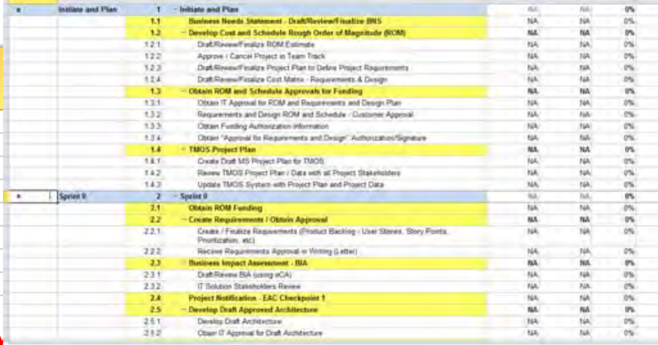
Scope and quality red due to test issues and changes

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	



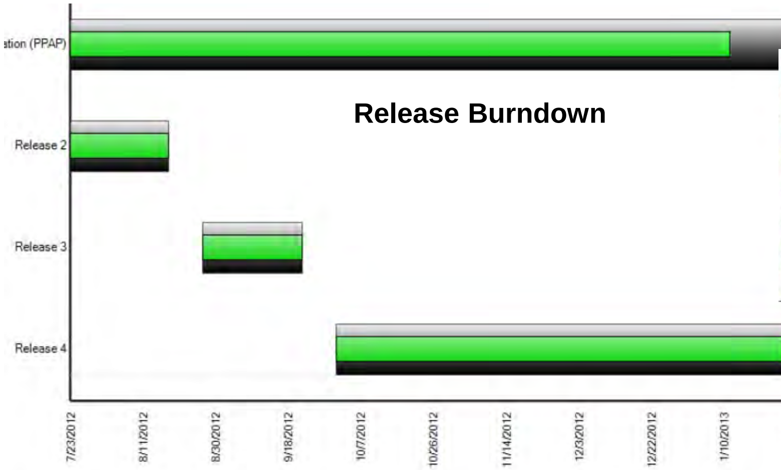
Scorecard Indicator	WBS	Task Name
	1	⊟ Sample Agile Project Plan Template
Initiate and Plan	1.1	⊞ Initiate and Plan
Sprint 0	1.2	⊞ Sprint 0 - Analysis and Design
Development Sprints	1.3	⊞ Development - Sprint 1 through n
SIT	1.4	⊞ Systems Integration Test
CAT	1.5	⊞ Customer Acceptance Test
Governance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



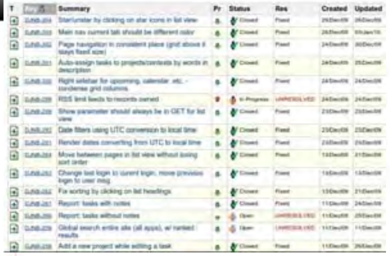
●	Initiate and Plan	1	− Initiate and Plan	NA	NA	0%
		1.1	Business Needs Statement - Draft/Review/Finalize BNS	NA	NA	0%
		1.2	− Develop Cost and Schedule Rough Order of Magnitude (ROM)	NA	NA	0%
		1.2.1	Draft/Review/Finalize ROM Estimate	NA	NA	0%
		1.2.2	Approve / Cancel Project in Team Track	NA	NA	0%
		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%
●	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	Recieve 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%
		2.1.2	Obtain IT Approval for Draft Architecture	NA	NA	0%

Senior Management View

Agile Project Team View



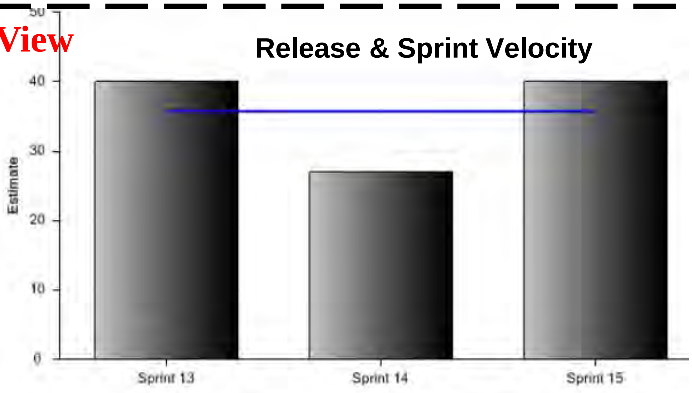
Release Burndown

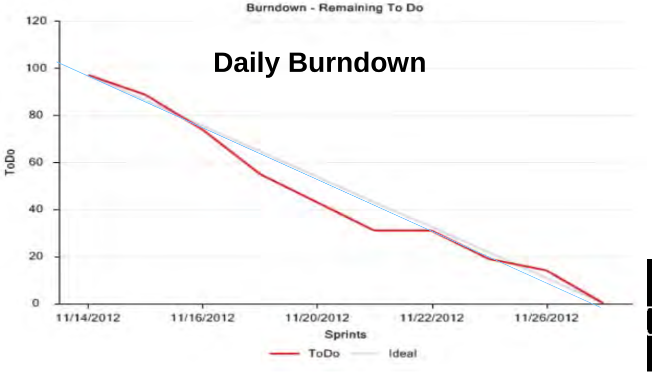


T	Key	Summary	Pr	Status	Res	Created	Updated
⊞	[illegible]	Star/unstar by clicking on star icons in list view		Closed	Fixed	[illegible]	[illegible]
⊞	[illegible]	Main nav current tab should be different color		Closed	Fixed	[illegible]	[illegible]
⊞	[illegible]	Page navigation in consistent place (grid above it stays fixed size)		Closed	Fixed	[illegible]	[illegible]
⊞	[illegible]	Auto-assign tasks to projects/contexts by words in description		Closed	Fixed	[illegible]	[illegible]
⊞	[illegible]	Right sidebar for upcoming, calendar, etc. - condense grid columns		Closed	Fixed	[illegible]	[illegible]
⊞	[illegible]	RSS limit feeds to records owned		In Progress	UNRESOLVED	[illegible]	[illegible]
⊞	[illegible]	Show parameter should always be in GET for list view		Closed	Fixed	[illegible]	[illegible]
⊞	[illegible]	Date filters using UTC conversion to local time		Closed	Fixed	[illegible]	[illegible]
⊞	[illegible]	Render dates converting from UTC to local time		Closed	Fixed	[illegible]	[illegible]
⊞	[illegible]	Move between pages in list view without losing sort order		Closed	Fixed	[illegible]	[illegible]
⊞	[illegible]	Change last login to current login, move previous login to user msg		Closed	Fixed	[illegible]	[illegible]
⊞	[illegible]	Fix sorting by clicking on list headings		Closed	Fixed	[illegible]	[illegible]
⊞	[illegible]	Report: tasks with notes		Closed	Fixed	[illegible]	[illegible]
⊞	[illegible]	Report: tasks without notes		Open	UNRESOLVED	[illegible]	[illegible]
⊞	[illegible]	Global search entire site (all apps), w/ ranked results		Open	UNRESOLVED	[illegible]	[illegible]
⊞	[illegible]	Add a new project while editing a task		Closed	Fixed	[illegible]	[illegible]

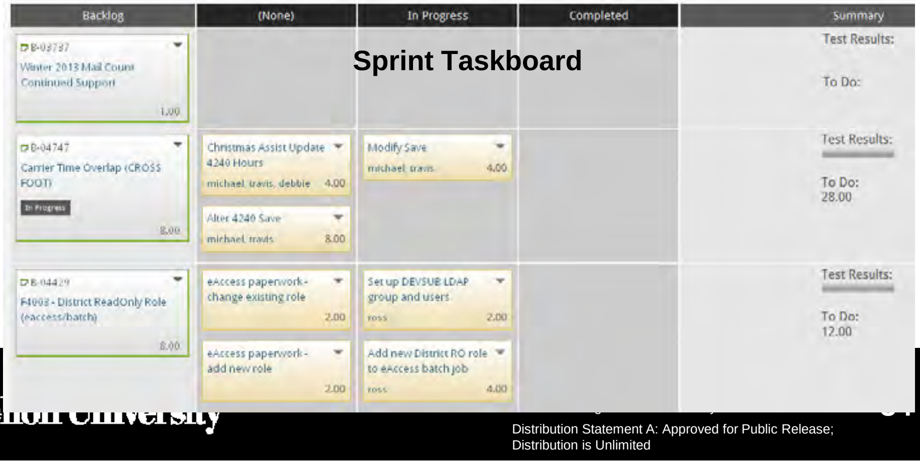
Issues / Risks

Release & Sprint Velocity





Daily Burndown



Sprint Taskboard

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

“Vital Few” Performance Metrics

Metric	Increment	Sprint	Compone nt
Planned vs. Actual size, effort, schedule, earned value	√	√	√
Cost of quality - % development effort in defect prevention, pre-test defect removal, testing defect removal, post-release defect removal	√		
% defects removed prior to system test	√	√	
Time in User Acceptance Test	√	√	
% with zero post-unit test defects			√
% design, code inspected	√	√	√
Process improvement proposals	√		

Government Expect More

Make **quality the number one goal**

Hold contractors liable for software defects or vulnerabilities

Acquire **Lowest Price Guaranteed Quality (LPGQ)** offers rather than Lowest Price Technically Acceptable (LPTA) or Best Value offers

Trust contractors but verify

Industry
Be Responsible for Quality

Make **quality the number one goal**

Cease dependence on test and rework for **defect removal**

Provide **quality guarantees** while continually improving cost and schedule performance

Support **2013 NDAA Sec 933**

Empower Developers

End the practice of imposing **arbitrary and unrealistic** schedules

Trust and support the teams

Train software developers to negotiate **realistic and aggressive** schedule

Have Fun on the Job

Joy in Work

“There is a square; there is an oblong. The players take the square and place it upon the oblong. They place it very accurately; they make a perfect dwelling place. Very little is left outside. The structure is now visible; what was inchoate is here stated; we are not so various or so mean; we have made oblongs and stood them upon squares. This is our triumph; this is our consolation.”

The players in Virginia Woolf’s *The Waves*

What does

“FUN ON THE JOB”

Mean to you?

“ If I have made myself too clear,
you must have misunderstood me”
Alan Greenspan

Questions?

Contact

Girish Seshagiri

girish.seshagiri@ishpi.net

703 426-2790

Performance Metrics That Matter:

Eliminating Surprises in Agile Projects

Girish Seshagiri, Ishpi Information Technologies,Inc.

Software Solutions Conference 2015
November 16–18, 2015

Software Engineering Institute | Carnegie Mellon University