Agility, Quality, Innovation, Joy in Work

Is the Two-Week Agile Sprint, the Worst Software Idea Ever?

Management Issues in Software Assurance and Information Security

Advanced Information Services Inc.

April 2, 2014

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.

Source: Shaping Your Approach – the Executive's role in software Assurance, 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

Source: Shaping Your Approach – the Executive's role in software Assurance, SWAMP Webinar
Jerry L. Davis, Chief Information Officer, NASA

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."

Growth Industries - 1

Information Assurance

Certification & Accreditation

Testing, Test Automation

Code Analyzers

Certifications

PMP, ITIL, CMMI, Agile Scrum

Growth Industries - 2

The Application Security Industry

Is Now Greater Than

The Applications Development Industry

CMMI - 1

CMMI defines characteristics of an effective process

Sets organizational standards and baseline procedures for use by all projects in an organizational unit

Organizations use CMMI to enforce repeatable processes and work in a predictable way

CMMI – 2

CMMI is extremely helpful in stabilizing an organization and getting a level of statistical control

CMMI levels build the foundation needed for real improvement

CMMI level 5 is the beginning and not the end

CMMI Improves Quality

	For Projects of 5000 Function Points in Size		
CMM Level	Defect Potential per Function Point	Defect Removal Efficiency	Delivered Defects per Function Point
SEI CMM 1	5.50	73.00%	1.49
SEI CMM 2	4.00	90.00%	0.40
SEI CMM 3	3.00	95.00%	0.15
SEI CMM 4	2.50	97.00%	0.08
SEI CMM 5	2.25	98.00%	0.05

Capers Jones, STN 13-1 April 2010: Software Quality, Reliability, and Error Prediction

CMMI Implementation Issues

Relying on artifacts to make sure process is being followed

Artifacts are produced by organizational bureaucracy and not the developers

Artifacts may have no relationship to the actual work being done

Easier to pass appraisals than to change engineering behavior

The Real Question

Whose Process Is It?

Agile Manifesto Assumptions

Large projects are unsuccessful or canceled

Planning and tracking project progress is non-value adding

- Requirements change throughout the project

Requirements defects are #1 reason projects fail

Big up-front design causes more problems than it solves

- Creates larger attack surfaces vulnerable to security incidents

Agile Manifesto

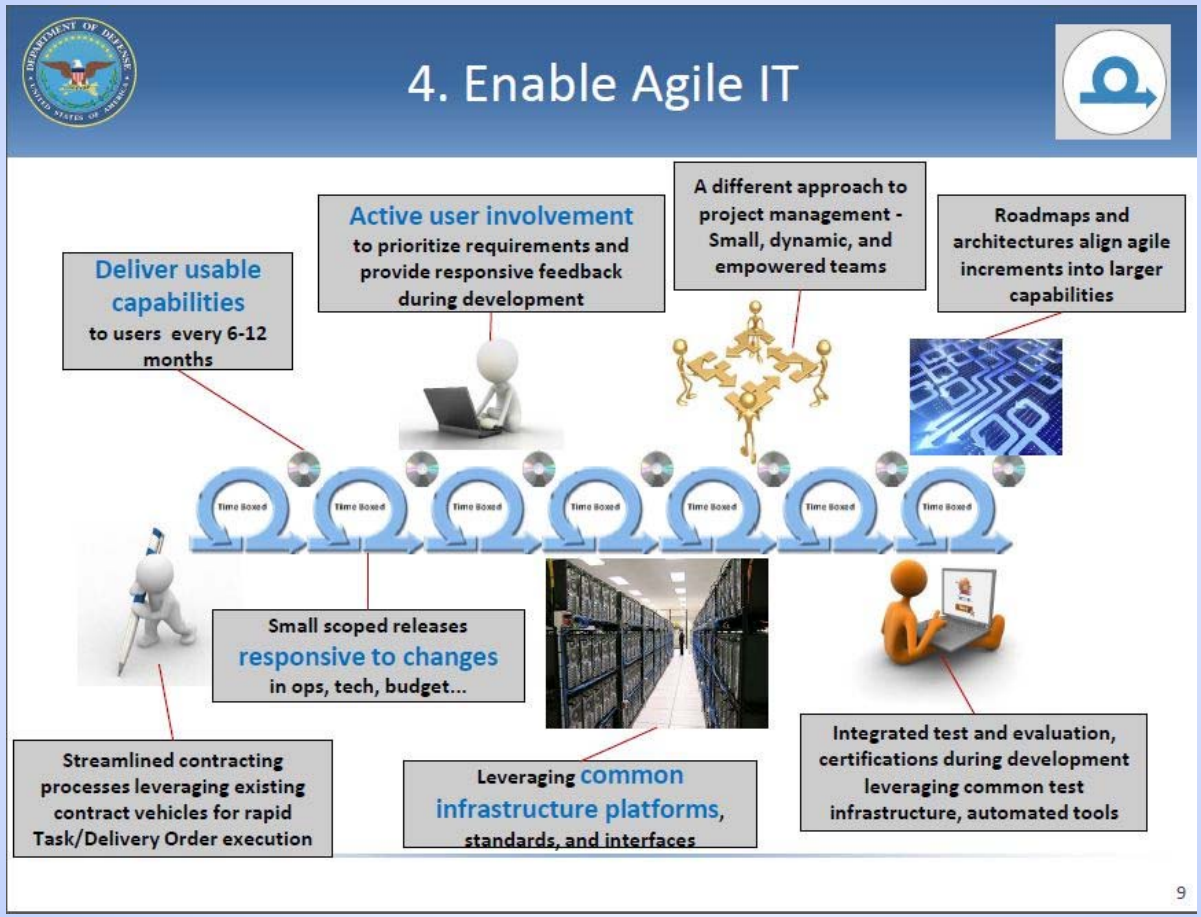
Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

DoD and Agile



DoD CIO's 10-Point Plan for IT Modernization, Ms. Teri Takai, DoD CIO

Agile Sprints

Example

- Two year modernization project to deliver 600,000 lines of code
- Two week sprints
- Working software 12,000 LOC every two weeks

Issues

- Defect density of the delivered working software
- “Deliver now, Fix later”?
- Technical debt
- Fixed management overhead every sprint
- Customer acceptance testing time
- Work-Life balance

Agile Manifesto 2.0?

Doing over Plan, Do, Check, Act

Guesswork over facts

Speed over optimum results

Accumulating technical debt over institutionalizing disciplined approach

If customers want it in the worst way,

that's how we will deliver it

Software Engineering's Persistent Problems

1. Exponential rise in cybersecurity vulnerabilities due to defective software
2. Unacceptable cost, schedule, and quality performance of legacy systems modernization and Enterprise Resource Planning (ERP) projects
3. Cost of finding and fixing software bugs (i.e. scrap and rework) the number one cost driver in software projects
4. Arbitrary and unrealistic schedules leading to a culture of "deliver now, fix later"
5. Inability to scale software engineering methods even for medium size systems
6. Understanding the impact of variation in individual productivity
7. Absence of work place democracy and joy in work

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

Immutable Laws of Software Development – 1

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

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

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 – 2

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

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

Successful cost, schedule, and quality outcomes depend on the degree of convergence between the organization's official process, the teams' perceived process and the individuals' actual processes

The Way Forward – 1

CONNECT THE DOTS

AGILITY, QUALITY, INNOVATION,

AND

JOY IN WORK

The Way Forward – 2

Quality work is more predictable

Unhappy people rarely do quality work

Without quality, agility is in name only

Quality without numbers is just talk

Culture of innovation is possible only in environments where people are respected, self-aware and practice agility with discipline

We can get there by applying the principles of Deming, Drucker, and Humphrey to software technical work

What They Said

Dr. Deming

- Constancy of purpose brings innovation
- Extrinsic motivation leads to the destruction of the individual
- Remove barriers that rob people of pride of workmanship

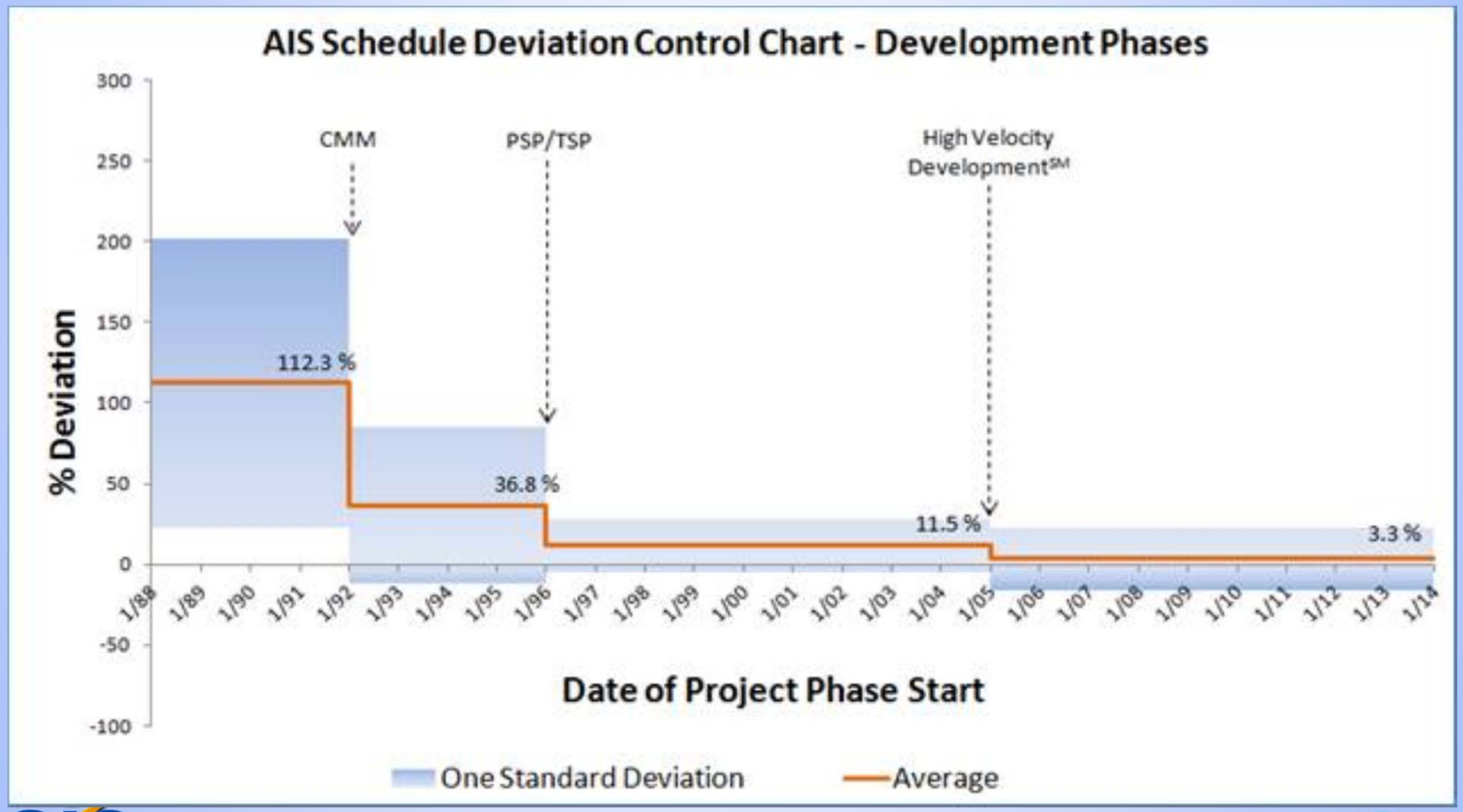
Dr. Drucker

- That one can truly manage another person is by no means adequately proven

Dr. Humphrey

- Quality is more important than schedule, since poor quality performance is the root cause of most software cost and schedule problems

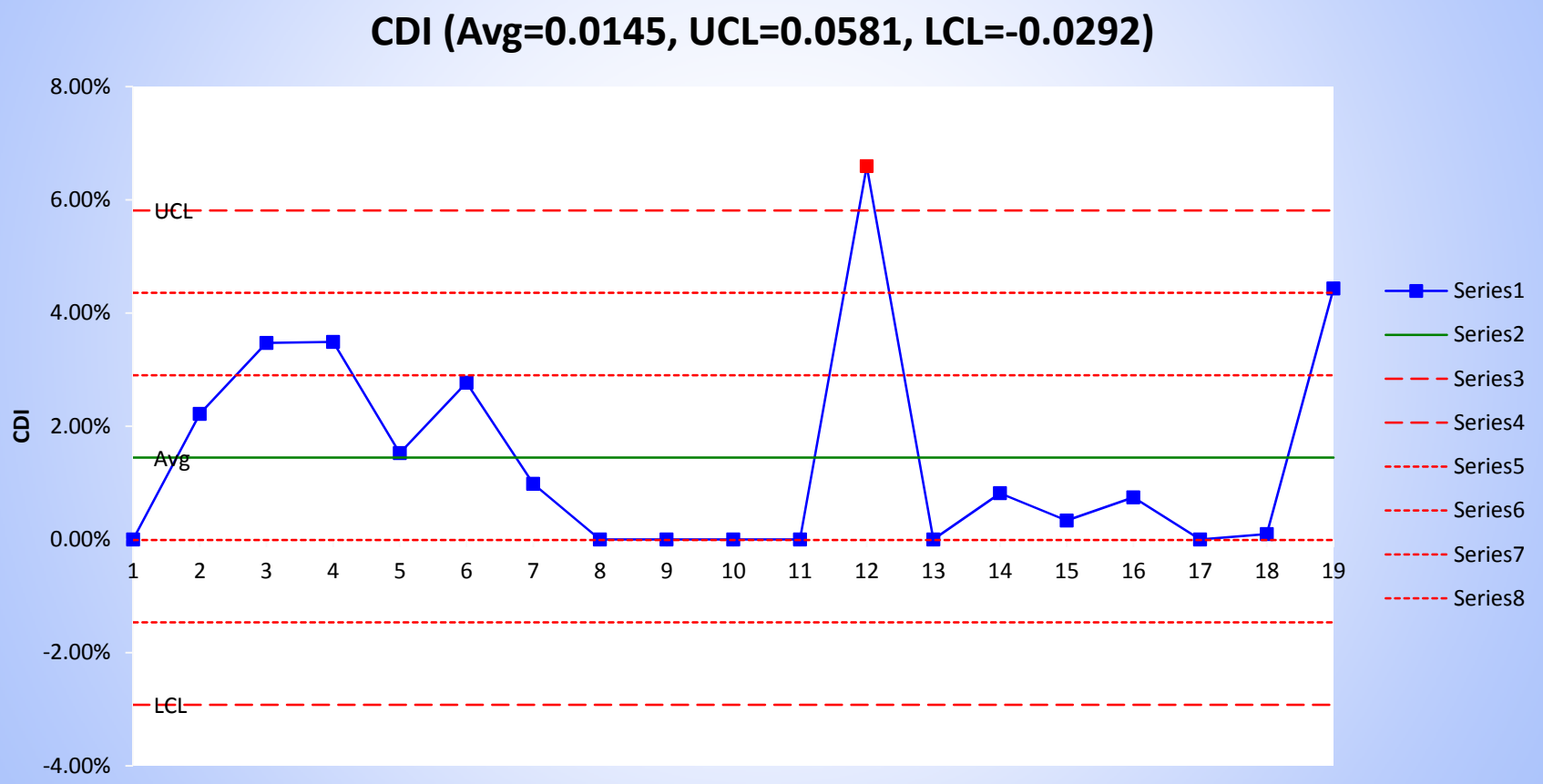
Constancy of Purpose
Schedule Performance



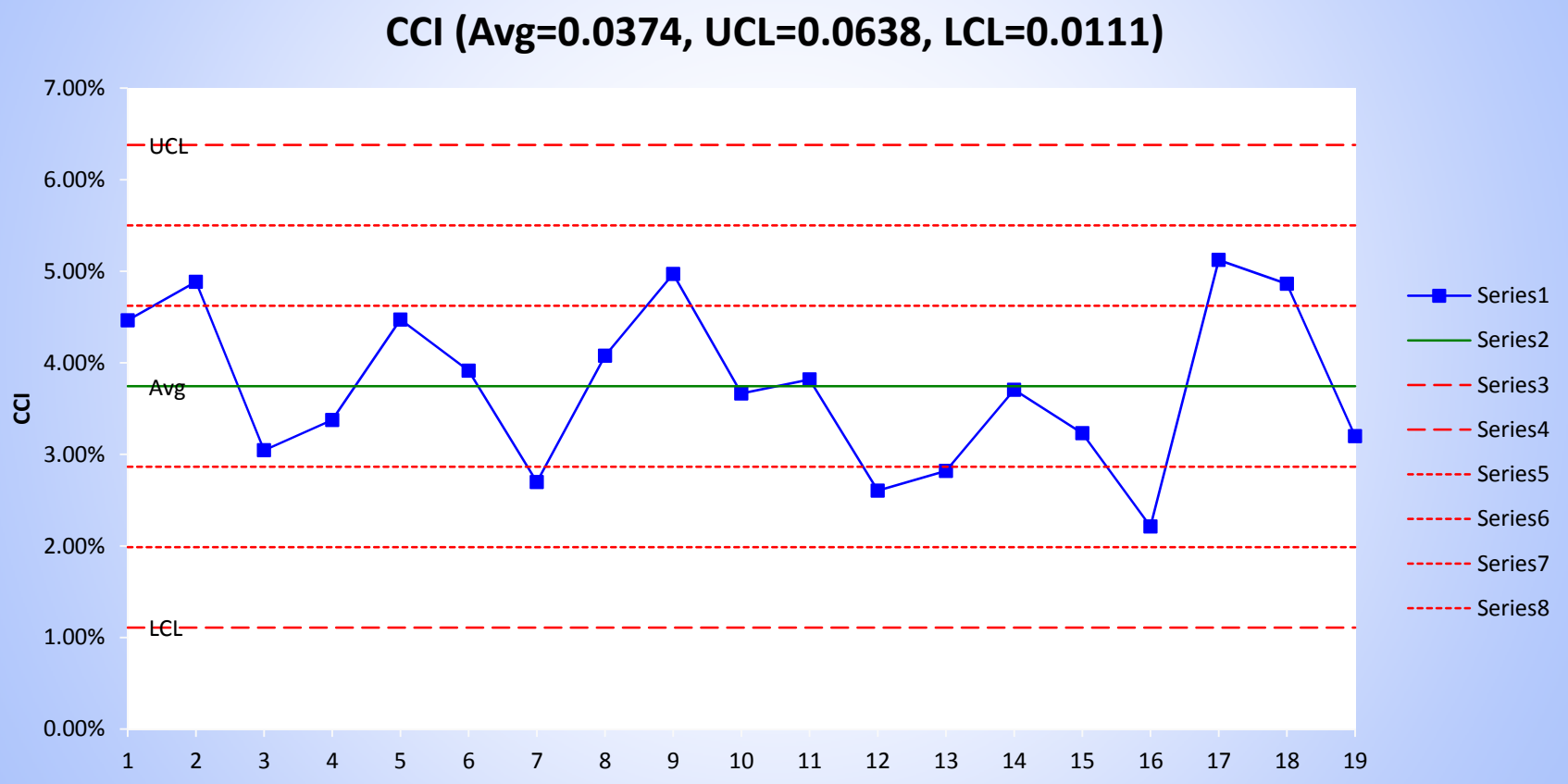
Performance Metrics That Matter Benchmarking

	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

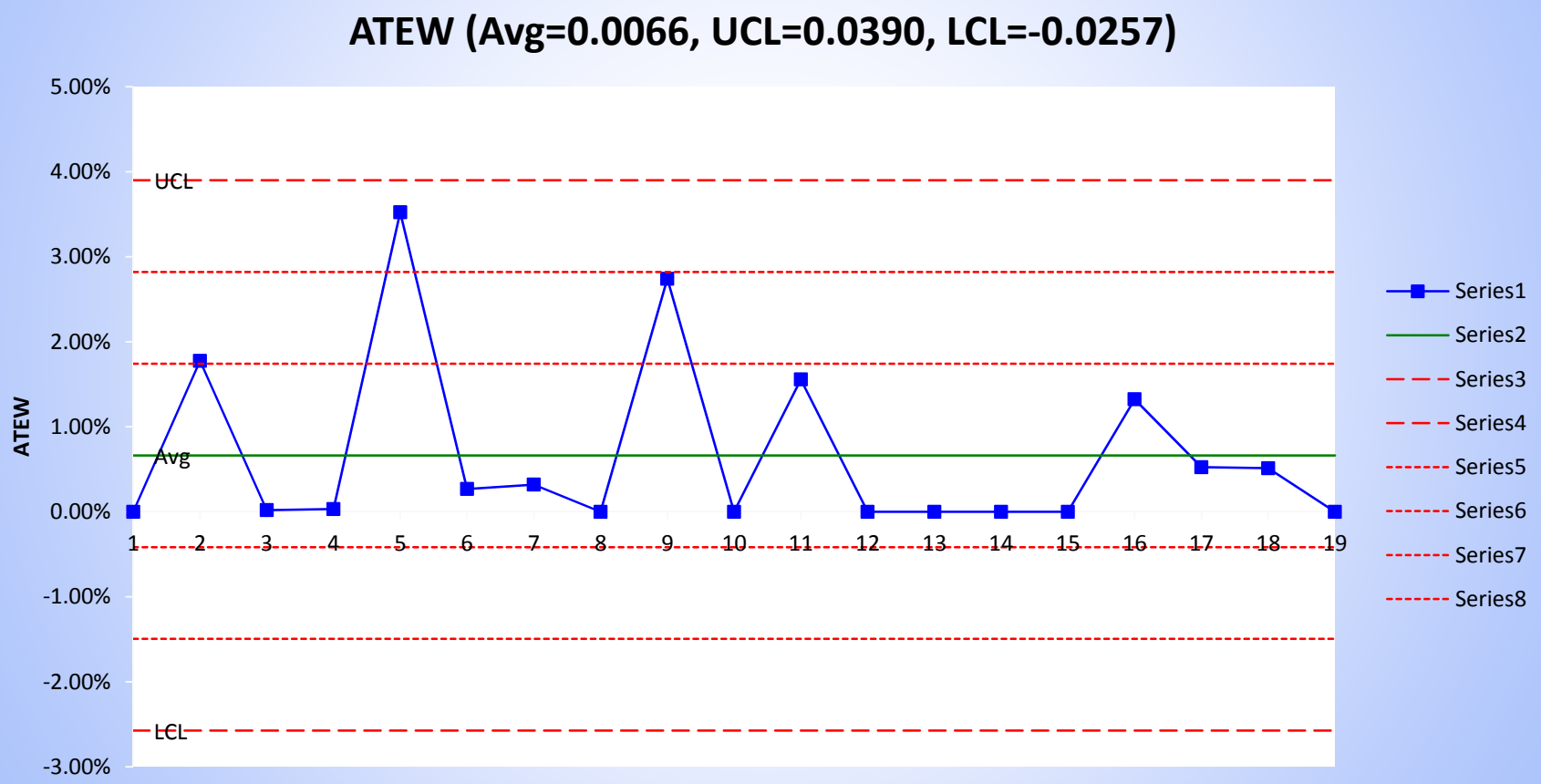
Design Inspection Rework



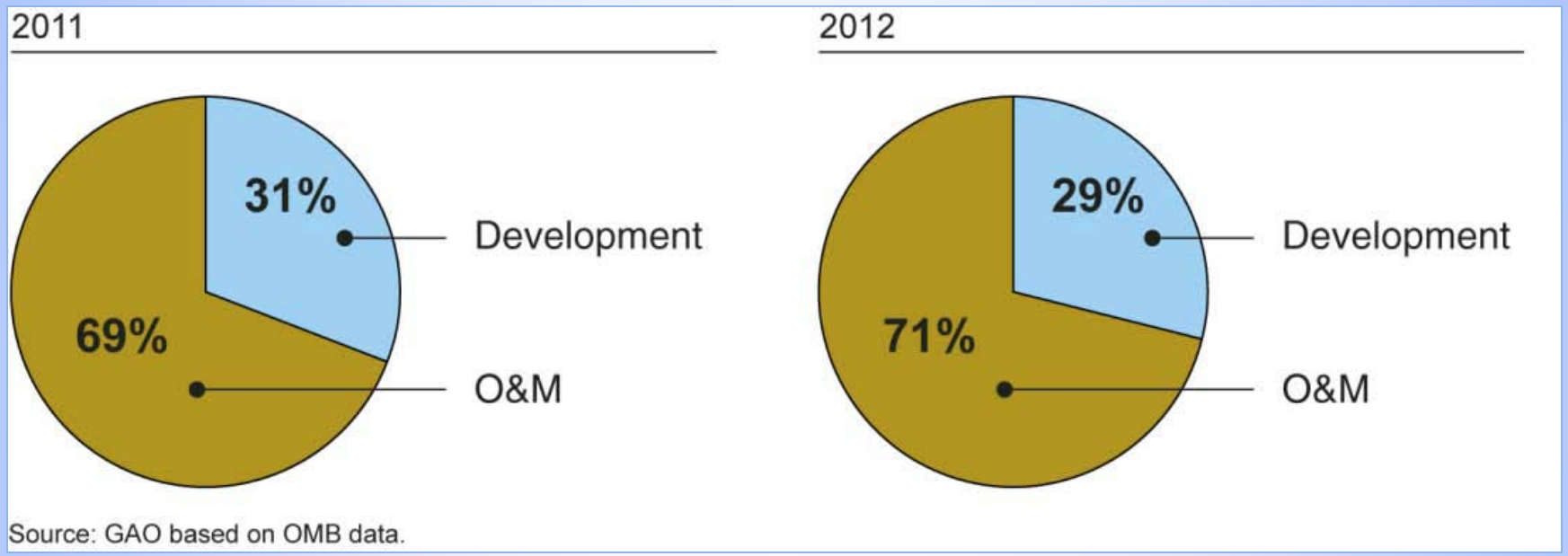
Code Inspection Rework



Acceptance Test Rework



The $80 Billion IT Spend



Percentages of Total IT Spending for Fiscal Years 2011 and 2012 for 26 Key Federal Agencies

The $592 Billion Opportunity

Category	%	Spend	Waste	%	Annual Savings
Development	30.0	24.0	Scrap and Rework	60.0	14.4
O & M	70.0	56.0	Corrective Maintenance	80.0	44.8
Annual Spend	100.0	80.0			**59.2**

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?

Girish Seshagiri

girish.seshagiri@advinfo.net

703 426-2790