How to Predict Software Quality Early in the Software Development Lifecycle and Produce Secure Software

Tutorial by
Girish Seshagiri

Software Assurance Forum
“Game Changing Tools and Practices”
September 27, 2010

Trademarks And Service Marks

The following are service marks of Carnegie Mellon University.

- CMMI℠
- Team Software Process℠
- TSP℠
- Personal Software Process℠
- PSP℠

The following are registered trademarks of Carnegie Mellon University.

- Capability Maturity Model®
- CMM®
- Capability Maturity Model® Integration
- CMMI®
- CERT®

The following are registered service marks of Advanced Information Services Inc.

- No Surprise Project Management℠

The following are registered trade marks of Advanced Information Services Inc.

- CPIW®
- SOLONsys®

ais

© AIS 2009

How to Predict Software Quality Early in the Software Development Lifecycle and Produce Secure Software

Module 1

Introduction And Tutorial Overview

ais

© AIS 2009

Module Overview

Introductions, expectations, and tutorial overview

Managing the software work

Why are we here?

Tutorial objectives

ais

© AIS 2009

Introductions

Who we are

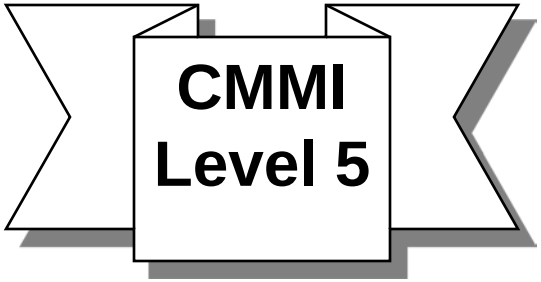
- Names
- Jobs

Who you are

- Names
- Jobs
- What you expect to get out of this tutorial

© AIS 2009

ais



Winner IEEE Software Process Achievement Award

http://www.sei.cmu.edu/managing/ieee-award/ieee.award.html

Preamble

Don’t think of business as a life without greatness

Unless the distant goals of meaning, greatness, and destiny are addressed, we can’t make an intelligent decision about what to do tomorrow morning – much less set the long-term strategy of the company

First decision must be to commit to an ethical world, a civilized existence, a moral order

Nothing is more practical than for people to deepen themselves

- Peter Koestenbaum (pkipeter@ix.netcom.com)

ais

© AIS 2009

Tutorial Overview

This is a tutorial for C-Level executives, software managers, team leaders, supervisors and professionals who are responsible for improving their organization's capability to manage the software work, and produce secure software on predictable cost and schedule

The material for this tutorial is drawn from

- *Books, writings and lectures by Watts Humphrey*
- *Books and writings by Peter Drucker, Edwards Deming, Capers Jones*
- The instructor's 30+ years experience managing the software work

© AIS 2009

Managing the Software Work State of the Practice - 1

In spite of increased adoption of PMP, Six Sigma, Lean, Agile, CMMI, ITIL etc, industry performance outcomes for cost, schedule and quality continue to be mixed

Changing managers, procurement regulations, acquisition procedures, and contracting provisions including oversight reporting have not resolved the cost, schedule, and quality problems of most software projects

Even with all of the emphasis on writing software with security in mind, most software applications remain riddled with security holes

© AIS 2009

Managing the Software Work State of the Practice - 2

Current software quality methods rely on testing, and are expensive, unpredictable and ineffective

Heroic efforts rescue troubled projects; heroes are in short supply

High-performance teams are the exceptions and not the norm

© AIS 2009

Why Are We Here? - 1

The adverse impact of software vulnerabilities caused by defective software is far-reaching

The defects that escape testing are exploited by hackers to launch cyber attacks

The current method of dealing with the increasing number of cyber attacks is reactive

© AIS 2009

Why Are We Here? - 2

We need a rapid transformation of the U.S. software industry from the current “Deliver now, fix later” culture to one capable of delivering substantially defect free code within predictable cost and schedule

This is a national high-priority need

If we continue with current methods, the U.S. taxpayer will pay billions of dollars for fixing defects in delivered products

ais

© AIS 2009

Why are We Here? - 3

First step is to make quality the number one priority and recognize that in order to manage the software work we must learn to manage quality

By adopting proven principles of managing knowledge work, software development quality and productivity can be increased by orders of magnitude

The leadership challenge is to build a cohesive and rewarding team environment where most people can do much better work than they are currently doing and produce truly amazing results

© AIS 2009

Tutorial Objectives - 1

Help attendees understand

- The common misconceptions about software work, and how our contracts and management systems are largely built on them, leading to a culture of “deliver now, fix later”
- The enormous non-value added cost of current practices that rely on testing as the principal software defect removal method
- Characteristics of procedural processes such as CMMI and how organizations can leverage their substantial investment in CMMI by adopting operational processes such as Agile and Team Software Process (TSP)

© AIS 2009

Tutorial Objectives - 2

Help attendees understand

- Performance metrics that really matter such as the cost of quality, time in test, defect density of delivered product, phase yields, and effectiveness of reviews and inspections etc.

- How to calculate a Process Quality Index metric to predict whether components that have been unit tested will have down-stream defects in integration, system, and customer acceptance testing

- Principles of managing knowledge work

ais

© AIS 2009

Tutorial Objectives - 3

Help attendees understand

- How to promote, build and sustain a Software Center of Excellence executing at CMMI Maturity Level 5

- Critical business value drivers: agility, quality and joy in work

© AIS 2009

Agenda

Module 1 Introduction	15
Module 2 Current Methods – "Deliver now, fix later"	25
Module 3 Software Process Evolution	35
Break	30
Module 4 Predicting Component Quality	15
Module 5 Real Project Examples	15
Module 6 Software Center of Excellence	30
Module 7 Joy in Work	15

ais

© AIS 2009

Summary

Defective software is insecure

Managing the software work is difficult because of the way it is currently done

Now, there is a national high priority need to change the way we manage software

To manage software, you need to manage quality

© AIS 2009

How to Predict Software Quality Early in the Software Development Lifecycle and Produce Secure Software

Module 2

Current Methods – “Deliver now, Fix later”

© AIS 2009

Module Overview

Common misconceptions about the software work

Contracts and management systems

Relationship between quality and schedule

Defect injection and removal

Managing software quality

ais

© AIS 2009

Common Misconceptions About The Software Work - 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

Source: *Managing the Software Process*, Watts Humphrey

© AIS 2009

Common Misconceptions About The Software Work – 2

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 levels guarantee results

Maturity level 5 is the end

ais

© AIS 2009

Contracts and Management Systems - 1

Too often, our contracts and management systems are based on the common misconceptions

Time and Materials and Cost Plus Fee contracts end up rewarding poor cost, schedule and quality performance

CMMI levels 2 or 3 specified as pass/fail criteria

Emphasis on product technical reviews which are important, but do not motivate change and are not helpful to improve the process or track status

ais

© AIS 2009

Contracts and Management Systems - 2

Not trusting teams to estimate the job, prepare aggressive and realistic plans, and negotiate responsible commitments

Lack of precision and timeliness in reporting contract performance and job status

Multi-contractor teams lack sense of purpose, goals and mission and do not act as a real team

© AIS 2009

The Top 15 U.S. Software Cost Drivers in Rank Order Circa 2009

1. **The cost of finding and fixing bugs**
2. The cost of cancelled projects
3. The cost of producing paper documents and English words
4. The cost of recovery from security flaws and attacks
5. The cost of requirements changes during development
6. The cost of programming or coding
7. The cost of customer support
8. The cost of meetings and communication
9. The cost of project management
10. The cost of application renovation
11. **The cost of innovation and new kinds of features**
12. The cost of litigation for cancelled projects
13. The cost of training and learning software applications
14. The cost of avoiding security flaws
15. **The cost of acquiring reusable components**

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

© AIS 2009

Schedule Compression and Quality

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%

Source: *Donald M. Beckett and Douglas T. Putnam, STN 13-1 April 2010: Software Quality, Reliability, and Error Prediction*

© AIS 2009

Adding Staff and Quality

Staff/Quality Trade-off			
	Peak Staff 16	Peak Staff 32	% Change
Duration Mths	26	22.6	-13.1%
Defect Count	1,043	1,411	35.3%
Effort Months	225	392.0	74.2%

Source: Donald M. Beckett and Douglas T. Putnam, STN 13-1 April 2010: Software Quality, Reliability, and Error Prediction

© AIS 2009

Defect Injection Rate and Removal Percent - Sample

Phase	Injection Rate – Defects Per Hour	Percent Defects Removed - Yield
Requirements	0.5	
High Level Design	0.5	
Detailed Design	1.5	
Code	2.0	
Unit Test		50%
Integration Test		55%
System Test		35%

Your organization/projects have such data?

ais

© AIS 2009

Defects Per Function Point

Defect Origins	Defect Potentials	Removal Efficiency	Delivered Defects
Requirements	1.00	77%	0.23
Design	1.25	85%	0.19
Coding	1.75	95%	0.09
Document	0.60	80%	0.12
Bad Fixes	0.40	70%	0.12
Total	*5.00*	*85%*	*0.75*

Source: *Capers Jones, STN 13-1 April 2010: Software Quality, and Software Economics*

© AIS 2009

Deliver now, fix later - 1

Why do competent software professionals agree to delivery dates when they have no idea how to meet them?

Why do rational managers accept schedule commitments when engineers offer no evidence that they can meet the commitments?

© AIS 2009

Deliver now, fix later - 2

If it doesn’t have to work any body can deliver on time

If you want the product in the worst way, that’s how you will get it

If the situation looks truly impossible, it probably is

Schedule is what must happen; quality determines what will happen

© AIS 2009

Negotiated Commitments - Developers

When pressed for early deliveries, the responsible team members say

> "I understand your requirements, I will do my utmost to meet it, but until I make a plan, I can not responsibly commit to a date"

© AIS 2009

Negotiated Commitments - Managers

When pressed for early deliveries, the responsible managers say

> "I trust you to create an aggressive and realistic plan, I will review the plan, but I will not commit you to a date that you can not meet"

© AIS 2009

Managing Software Quality - 1

To meet schedule and cost commitments consistently, you must manage software quality

Quality without numbers is just talk

The common ways to manage software quality are with testing and reuse

Testing is now relied upon and is not sufficient

For reuse, the parts must be initially of high quality or the quality problems will be worse

ais

© AIS 2009

Managing Software Quality - 2

Software-intensive products typically have many defects

The three defect removal strategies

- Test, test, test
- Inspect and test
- Review, inspect and test

Time to find and fix test defects can vary from a few hours to few weeks

ais

© AIS 2009

Managing Software Quality - 3

Quality work is more predictable

If you do not manage quality, your schedule problems will end up as quality disasters

Software professionals must be trained to make plans and negotiate commitments

© AIS 2009

Summary

Managing the software work is difficult, because of the way it is currently done

Poor quality is the principal cause of cost and schedule problems

To manage the software work, you need to manage software quality

© AIS 2009

How to Predict Software Quality Early in the Software Development Lifecycle and Produce Secure Software - Module 3

Software Process Evolution

Module Overview

Managing the software work

CMMI, a procedural process

Need for operational processes

Building and maintaining self-directed teams – CMMI, TSP

Performance metrics that matter

© AIS 2009

Managing the Software Work - 1

Software and systems development is knowledge work

The first rule for knowledge work is that managers can’t manage it - the workers must manage themselves

© AIS 2009

Managing the Software work - 2

The second rule is that developers and their teams

- Must know how to manage themselves
- Negotiate their commitments with management
- Manage with data
- Own their own work

The third rule is that management must trust the development teams to plan and manage their own work

Source: Acquiring Quality Software, Watts Humphrey

© AIS 2009

Current Methods

Command and control management system not suitable for managing knowledge work

Workers and managers have different objectives and different views of project success

Lack of trust between management and teams

Teams unable to negotiate schedule and cost commitment

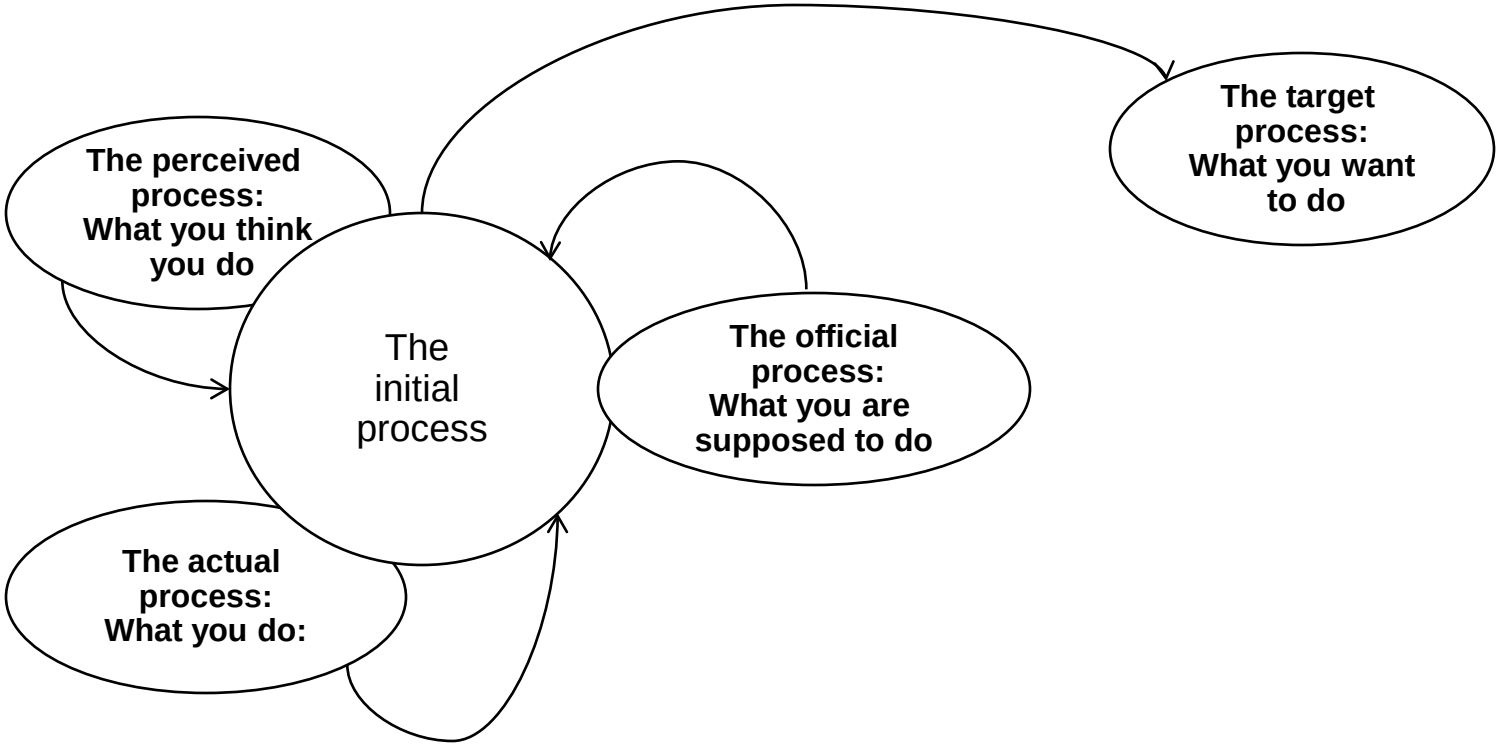
Arbitrary, unrealistic schedules creating culture of "Deliver now, fix later"

Lack of precise project status measures

ais

© AIS 2009

The Four Processes



You can not usefully evolve your process definition until it reasonably represents what you do

- Watts Humphrey, A Discipline for Software Engineering

© AIS 2009

CMMI, A Procedural Process - 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

ais

© AIS 2009

CMMI, A Procedural Process - 2

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

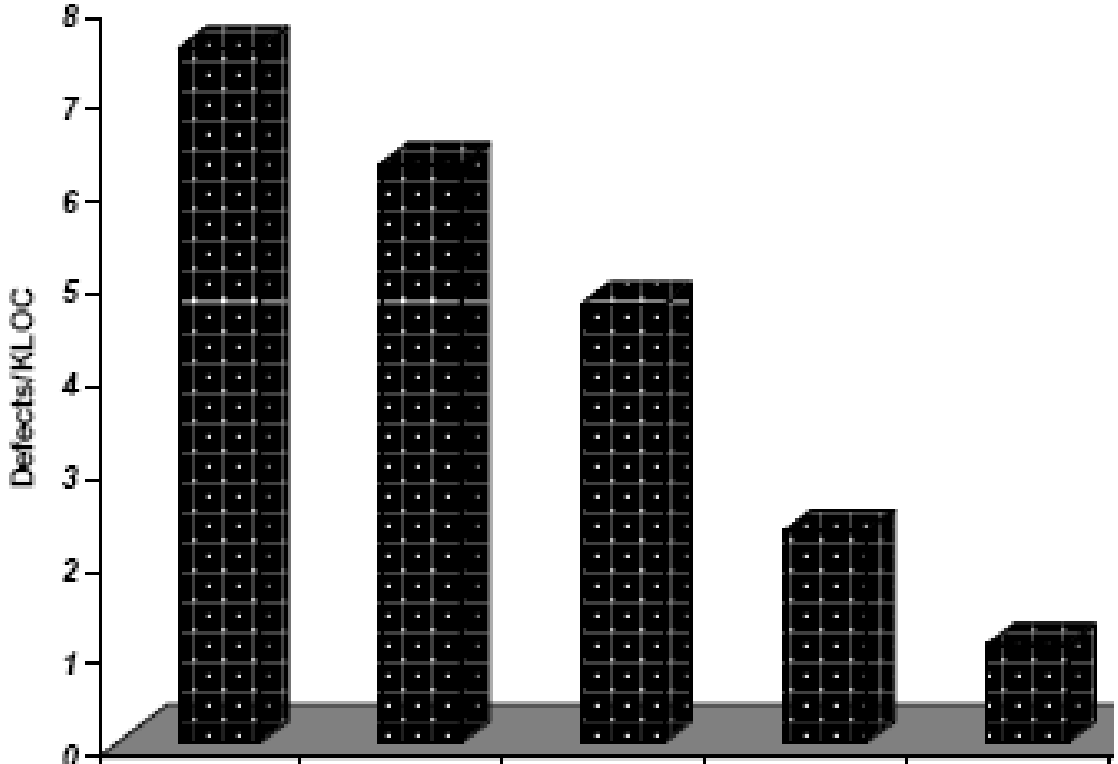
CMMI levels build the foundation needed for real improvement

CMMI level 5 is the beginning and not the end

ais

© AIS 2009

CMMI Improves Quality - 1



	CMM Level 1	CMM Level 2	CMM Level 3	CMM Level 4	CMM Level 5
Defects/KLOC	7.5	6.24	4.73	2.28	1.05

© AIS 2009

CMMI Improves Quality - 2

	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

© AIS 2009

Issues with Procedural Processes

Procedural processes rely 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

© AIS 2009

Need for an Operational Process

The need is not for lots of process data and artifacts but for engineers who develop and use artifacts and data

What would happen if software professionals used sound engineering practices?

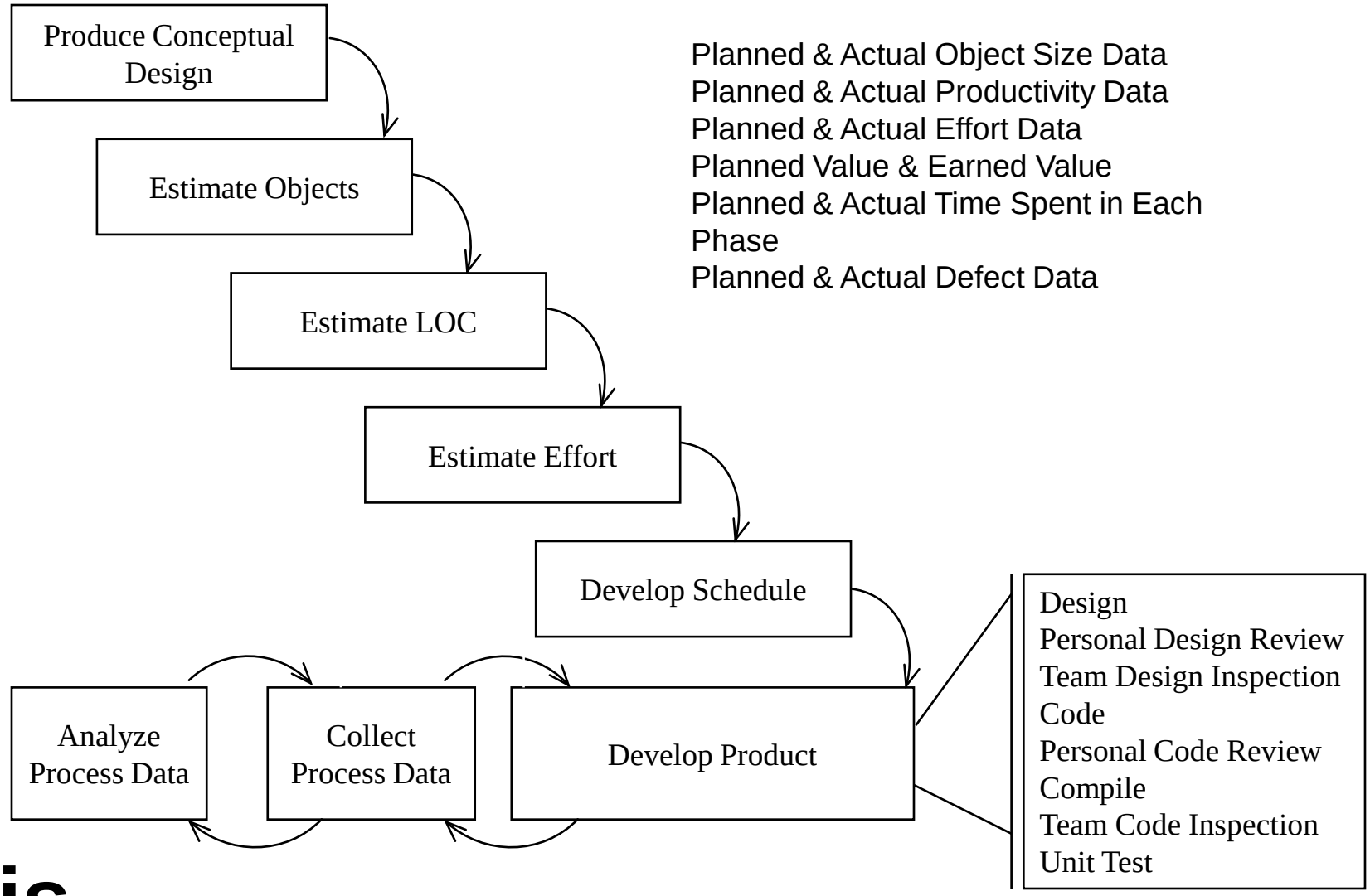
- Made and followed detailed plans
- Gathered and used historical data
- Measured and managed quality
- Analyzed and improved their processes

The need is for an operational process at the team and individual levels

ais

© AIS 2009

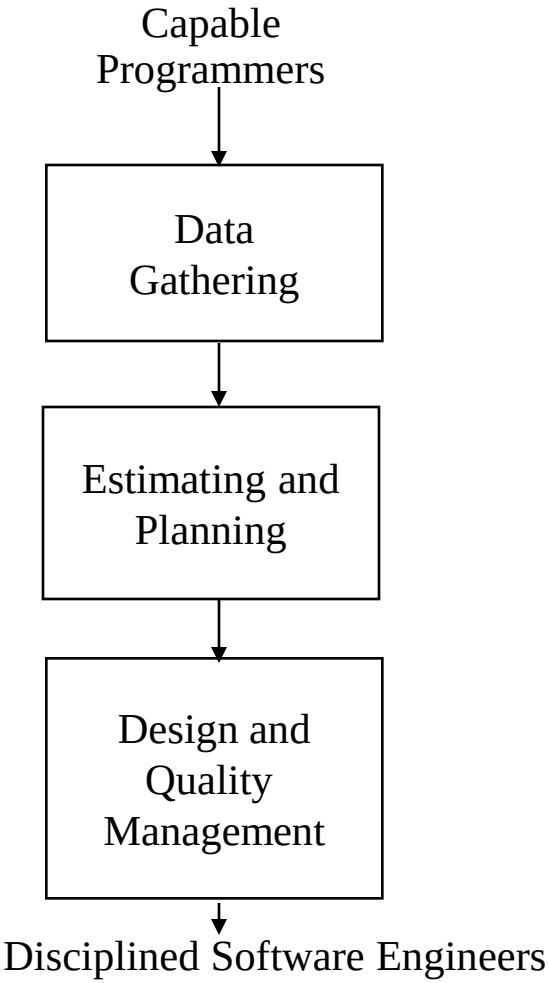
Component Development Process



Planned & Actual Object Size Data
Planned & Actual Productivity Data
Planned & Actual Effort Data
Planned Value & Earned Value
Planned & Actual Time Spent in Each Phase
Planned & Actual Defect Data

© AIS 2009

Self Improvement
Personal Software Process Training -1



© AIS 2009

Management Principles for Knowledge Work

Trust the knowledge workers

Build trustworthy teams

Rely on facts and data

Manage quality

Provide leadership to support knowledge workers to manage themselves

Source: "Why Can't We Manage Large Projects", Watts Humphrey, Crosstalk, July/August 2010

© AIS 2009

Leading Versus Managing

Management uses resources to accomplish results

Leaders motivate people to accomplish objectives

Management is impersonal and can be demeaning

Leadership is transformational where the real objective is the achievement

Ubuntu – “I am because we are”

© AIS 2009

Building Teamwork Capability - 1

Provide a management environment where the engineers are encouraged and rewarded for doing quality work

Create a mechanism to guide teams through defining their processes and making complete, precise, and detailed plans

© AIS 2009

Building Teamwork Capability - 2

Build an environment where everybody planned and tracked their work and measured and managed the quality of their products

Provide a trained coach

Create an organization environment where Level 5 behavior is the norm

ais

© AIS 2009

Team Improvement
Self-directed Teams

Characteristics of self-directed teams

- Sense of membership and belonging
- Commitment to a common team goal
- Ownership of the process and plan
- The skill to make a plan, the conviction to defend it, and the discipline to follow it
- Dedication to excellence

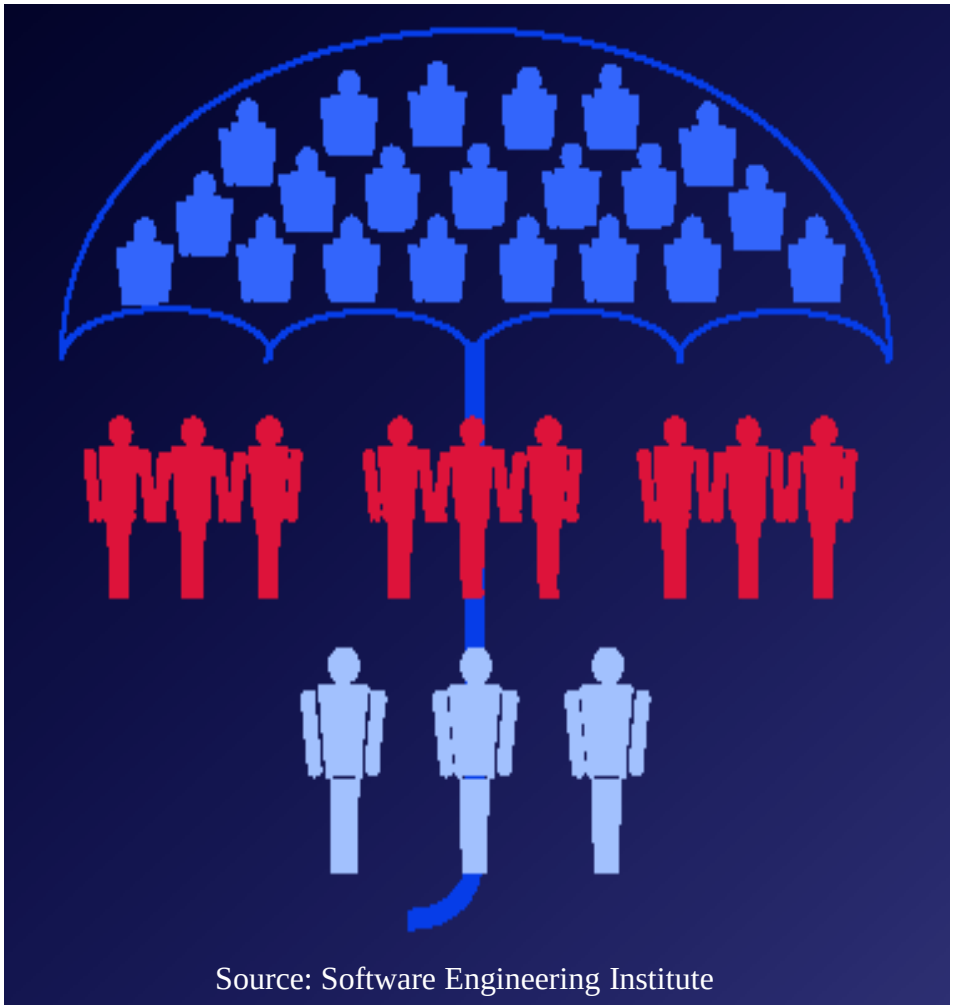
© AIS 2009

Transforming The World Of Software Models Of Excellence

CMMI – Builds organizational capability

TSP – Builds quality products on cost and schedule

PSP – Builds individual skill and discipline



© AIS 2009

Goals and Measurement Myths and Facts - 1

Dr. Deming

- Numerical goals accomplish nothing
- Extrinsic motivation leads to the destruction of the individual
- Rewards motivate people to work for the rewards
- Various components should work together for optimization of profit and joy in work
- System must create something of value, in other words, results
- Life is variation

© AIS 2009

Goals and Measurement Myths and Facts - 2

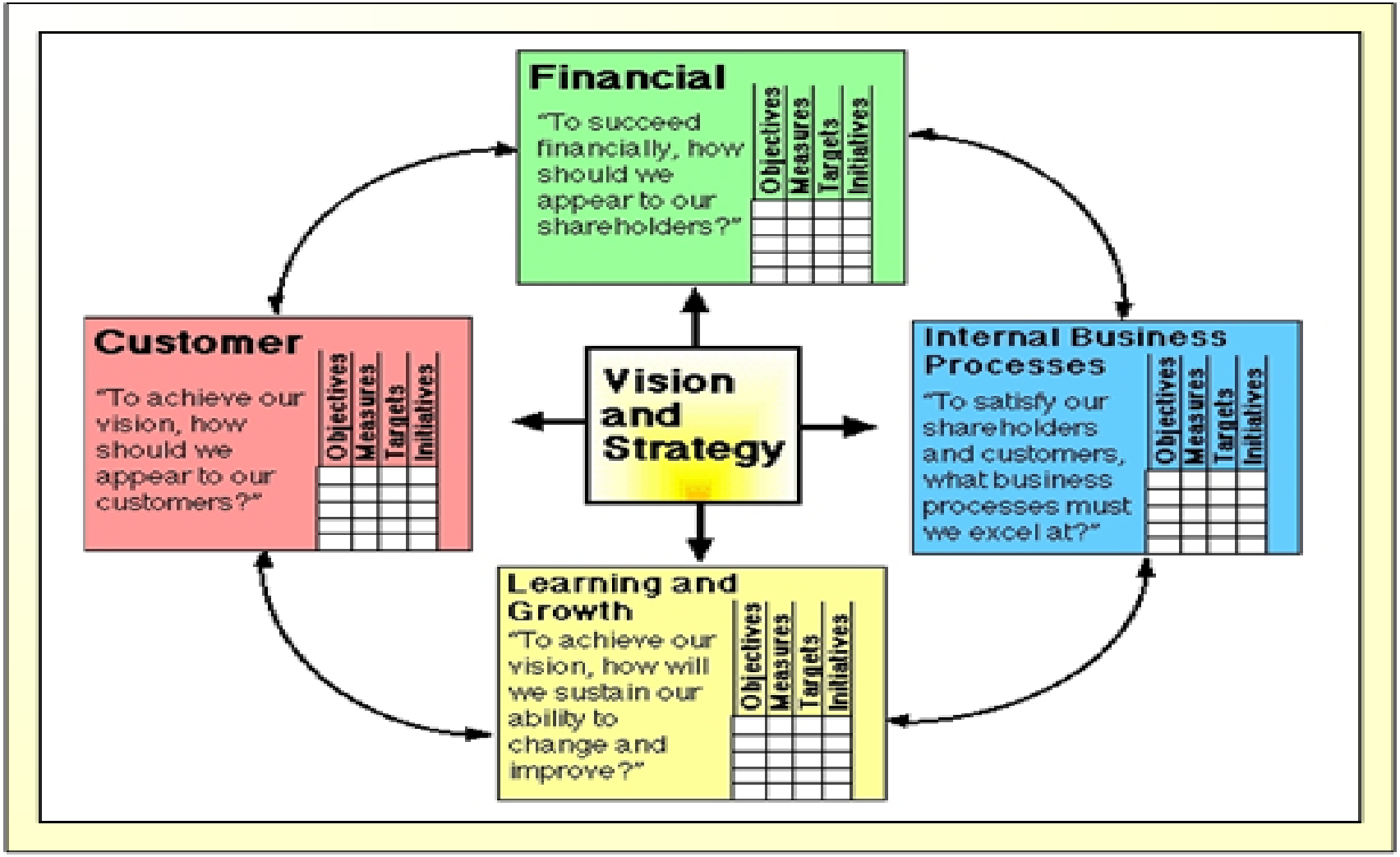
Watts Humphrey

- Undisciplined or unmotivated people can not do timely or predictable intellectual work
- Disciplined and motivated people need aggressive goals; support goals with specific programs and plans
- You can’t easily tell the quality of a program, but you can ask if it was properly developed
- If measures can not detect one-day slip, you can not anticipate problems and prevent them
- Defining measures is not always easy, but it is almost always possible
- Planning is the first step for everything you do
- Business is prediction

ais

© AIS 2009

Performance Metrics That Matter
The Balanced Scorecard



ais

© AIS 2009

Performance Metrics That Matter
Leading, Lagging Indicators - 1

- Customer feedback indicating met or exceeded customer expectations for:
  - Quality
  - Timeliness
  - Value for Products & Services

- Work products with less than or equal to targeted rework effort
- Work products with zero post development defects
- Commitments with actual effort equal to or less than committed effort
- Commitments with actual schedule equal to or less than committed schedule

- Value for Products & Services

committed effort
- Commitments with actual schedule equal to or less than committed schedule

PERFORMANCE DRIVERS – LEADING INDICATORS

- Monthly percent of revenue from targeted customer
- Monthly percent of revenue from targeted services
- Monthly percent of employees meeting or exceeding their budgeted revenue target

- Defect free delivery
- On-time delivery

- Assessment indicating targeted P-CMM Key Process Areas fully satisfied

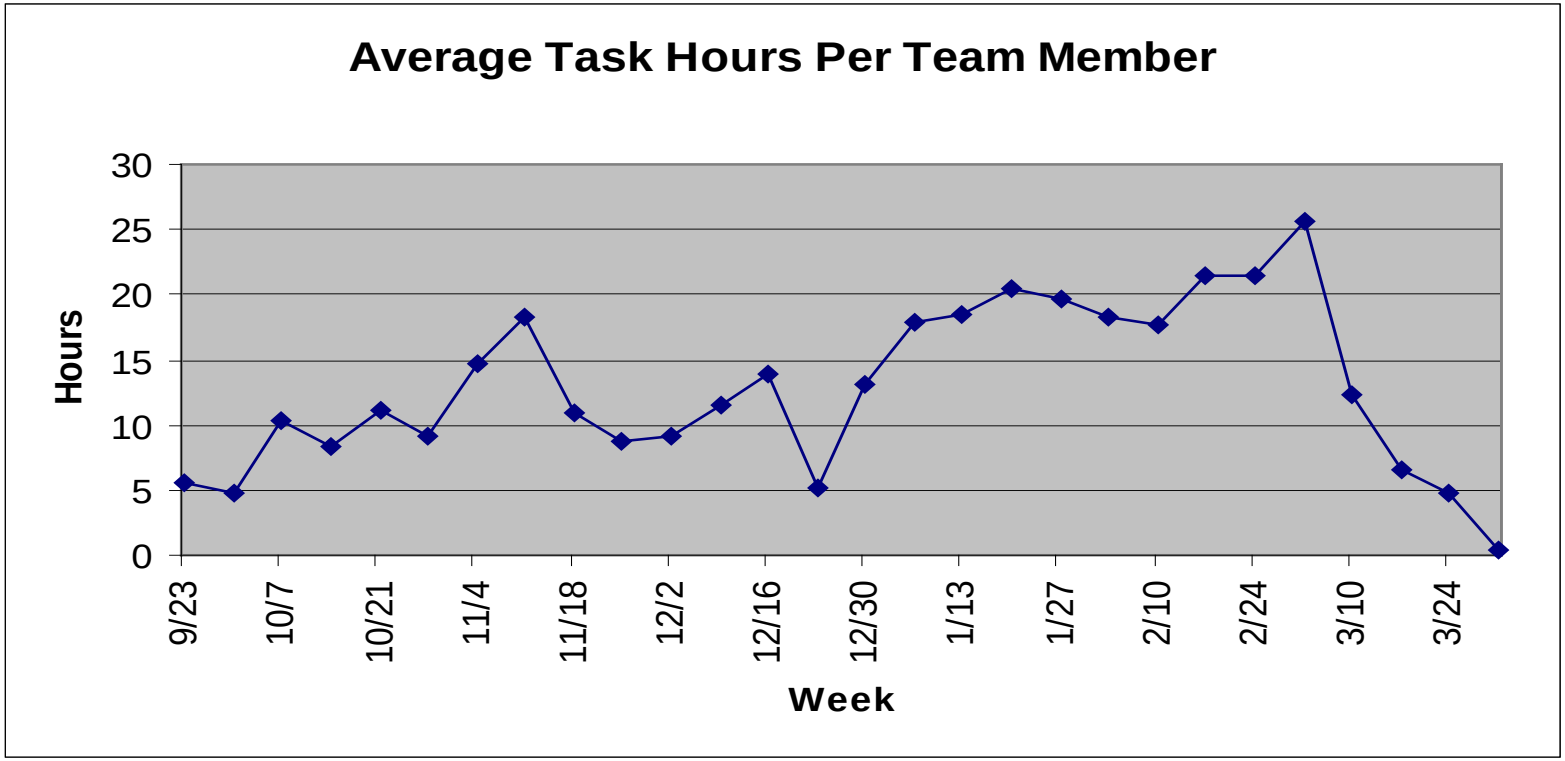
- Individuals following the PSP
- Work products with targeted percent of defects removed before peer review

- Teams using the TSP

Performance Metrics That Matter
Leading, Lagging Indicators - 2

FINANCIAL	CUSTOMER	EMPLOYEE	INTERNAL BUSINESS PROCESS	LEARNING & GROWTH
CORE OUTCOMES – LAGGING INDICATORS				
•Quarterly revenue increase •Annual valuation increase	•Customer feedback indicating met or exceeded customer expectations for: - Quality - Timeliness - Value for Products & Services	•Targeted employee turnover rate	•Work products with less than or equal to targeted rework effort •Work products with zero post development defects •Commitments with actual effort equal to or less than committed effort •Commitments with actual schedule equal to or less than committed schedule	
PERFORMANCE DRIVERS – LEADING INDICATORS				
•Monthly percent of revenue from targeted customers •Monthly percent of revenue from targeted services •Monthly percent of employees meeting or exceeding their budgeted revenue target	•Defect free delivery •On-time delivery	•Assessment indicating targeted P-CMM Key Process Areas fully satisfied	•Individuals following the PSP •Work products with targeted percent of defects removed before peer review	•Teams using the TSP •Employees achieving training goals •Internal products being developed •Identified market needs

Performance Metrics That Matter
Average No. of Task Hours Per week



© AIS 2009

Performance Metrics That Matter Weekly Status – Team Estimate At Completion

Task Status Summary

Status Summary	Plan	Actual	Pln/Act	Act/Pln
Project Hours	450.3	468.3	0.96	1.04
Project Hours To-date	3173.0	3255.3	0.97	1.03
Earned Value	4.3	1.7	2.51	0.40
EV To-date	28.5	26.9	1.06	0.94
To-date Hours for EV Tasks Closed	2482.0	1977.1	1.26	0.80
To-date Hours for Non-EV Tasks		429.6		

Current Week	7 of 29	
	Week(s)	
Current Status Based On EV	0.4	Behind
Projected Completion Based On		
Earned Value Earn Rate	3.0	Ahead
Remaining Plan Hours and EV		
Project Hours and Estimate Accuracy	4.9	Ahead

Blocked EV Effort	848.6
Avg EV Hours Per Week	465.0
Avg EV Per Week	3.8
EVHours reqd for 100% EV	5219.9
For ontime completion	Per Week
Required EV	3.3
Required EV Hours	237.3

© AIS 2009

Performance Metrics That Matter Weekly Status – Individual A Estimate At Completion

Task Status Summary

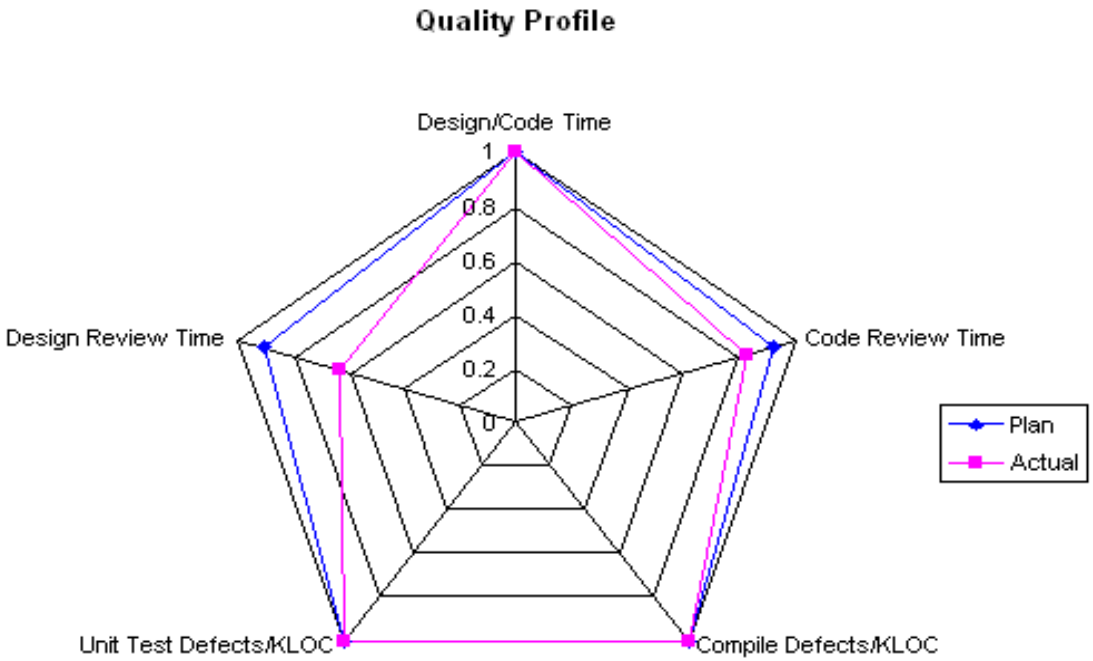
Status Summary	Plan	Actual	Pln/Act	Act/Pln
Project Hours	38.0	41.1	0.92	1.08
Project Hours To-date	269.7	298.8	0.90	1.11
Earned Value	8.1	6.7	1.22	0.82
EV To-date	62.6	36.9	1.70	0.59
To-date Hours for EV Tasks Closed	141.9	269.4	0.53	1.90
To-date Hours for Non-EV Tasks		20.7		

Current Week	7 of 29	
	Week(s)	
Current Status Based On EV	4.9	Behind
Projected Completion Based On		
Earned Value Earn Rate	10.0	Ahead
Remaining Plan Hours and EV		
Project Hours and Estimate Accuracy	15.7	Behind

Blocked EV Effort	8.7
Avg EV Hours Per Week	42.7
Avg EV Per Week	5.3
EVHours reqd for 100% EV	460.4
For ontime completion	Per Week
Required EV	2.9
Required EV Hours	20.9

© AIS 2009

Performance Metrics That Matter
Quality Profile



The above chart shows our 5 measures to achieve our quality goals:

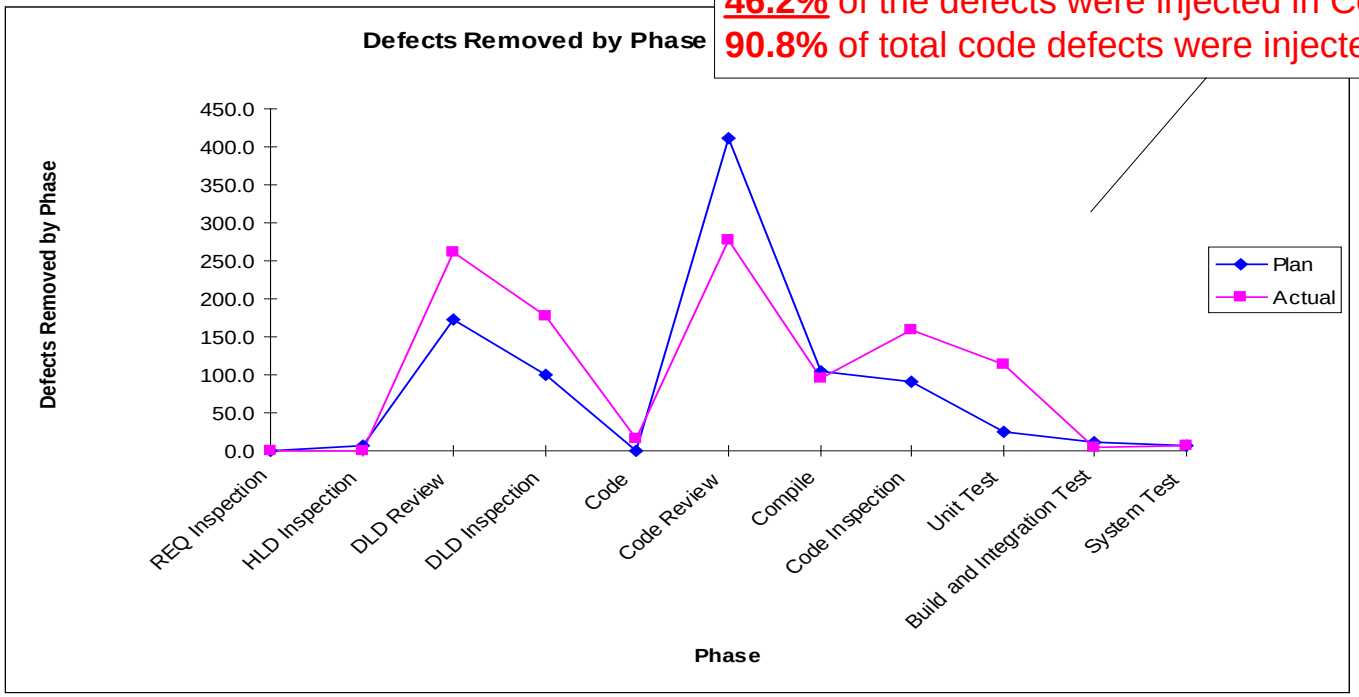
1) Design time to Code time comparison
2) Design Review Time as a % of design time (should be 50% or greater)
3) Code Review Time as a % of code time (should be 50% or greater)
4) Compile Defects per KLOC (Compile should find less than 10 defects per KLOC)
5) Unit Test Defects per KLOC (UT should find less than 5 defects per KLOC)

© AIS 2009

Performance Metrics That Matter
Defects Removed by Phase

The project Injected and Removed a total of
1112 code defects throughout the project lifecycle

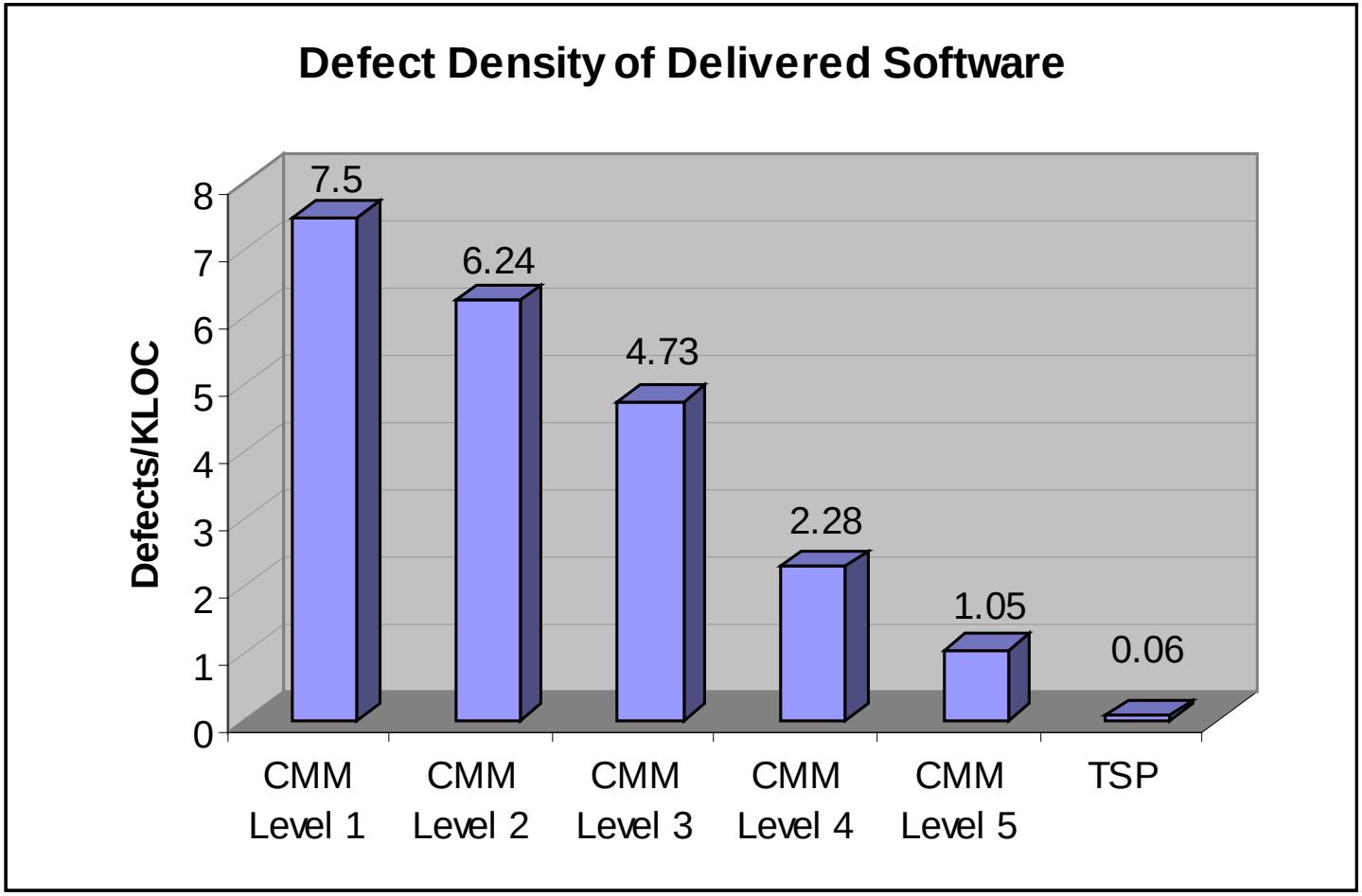
44.6% of the defects were injected in Detailed Design
46.2% of the defects were injected in Code
90.8% of total code defects were injected Design & Code



ais

© AIS 2009

Performance Metrics That Matter
Post-delivery Defects



Source: Davis, N., Mullaney, J. The Team Software Process (TSP) in Practice: A Summary of Recent Results. CMU/SEI-2003-TR-014. September 2003

© AIS 2009

Performance Metrics That Matter Benchmarking

	Industry Average	AIS Average
Schedule deviation	>50%	<10%
No. of defects in delivered product 100,000 LOC	>100	<15
% of design and code inspected	<100	100
Time to accept 100,000 LOC product	10 Months	5 Weeks
% 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

ais

© AIS 2009

Summary

CMMI is a procedural process

Need operational processes at the team and individual levels

Knowledge workers need to be trained to manage themselves

Build an environment where everybody planned and tracked their work and measured and managed the quality of their products

Performance metrics that really matter come from the developers themselves and provide early warning

ais

© AIS 2009

How to Predict Software Quality Early in the Software Development Lifecycle and Produce Secure Software - Module 4

Predicting Component Quality

Module Overview

Level 5 component development process

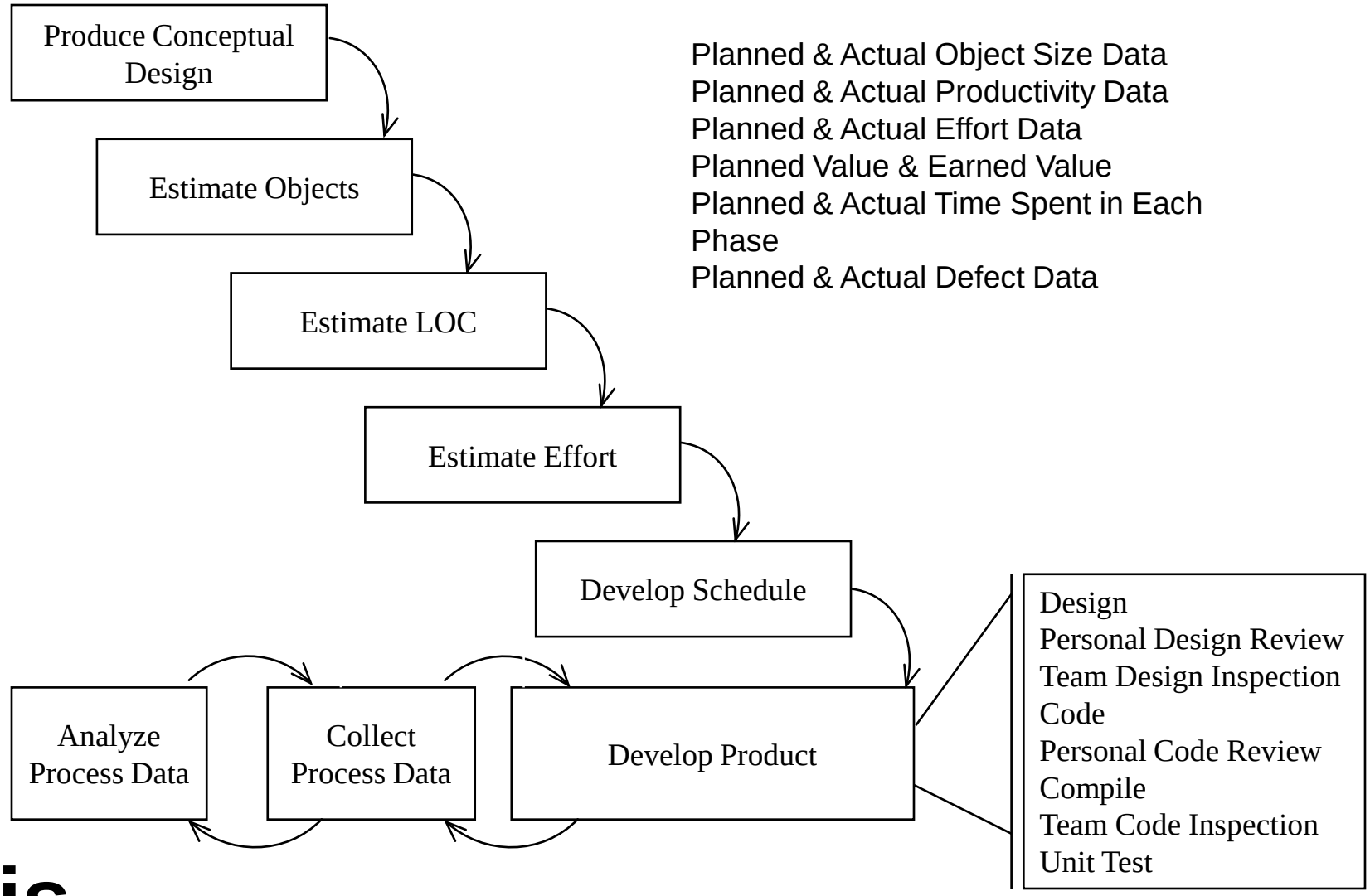
Early defect removal

Phase yields

Process Quality Index (PQI)

© AIS 2009

Component Development Process



Planned & Actual Object Size Data
Planned & Actual Productivity Data
Planned & Actual Effort Data
Planned Value & Earned Value
Planned & Actual Time Spent in Each Phase
Planned & Actual Defect Data

© AIS 2009

Performance Metrics That Matter
Early Defect Removal

Developers measure and manage the quality of their individual components and remove defects early through personal reviews and team inspections

Include OWASP Top Ten and CWE/SANS Top 25 Most Dangerous Programming Errors in review and inspection checklists

Developers strive to get the highest quality product into test

© AIS 2009

Performance Metrics That Matter
Team Goals for Phase Yields

Phase Yields (% defect removed)	Goal
Requirements Inspections	70%
Design reviews and inspections	70%
Code reviews and inspections	70%
Compiling	50%
Unit test at 5 or less defects/KLOC	90%
Integration and system test at <1.0 defects/KLOC	80%
Before compile	>75%
Before unit test	>85%
Before integration test	>97.5%
Before system test	>99%

© AIS 2009

Software Components - 1

To manage software product quality, we need to manage quality of the components

Components are what developers build

Error-prone components account for a disproportionate number of defects found in integration, system and user acceptance tests

ais

© AIS 2009

Software Components - 2

20% of the modules in a system typically account for 80% of the defects

It is extremely useful to know which components are likely to be error-prone in later testing so that we can take corrective actions pro-actively

© AIS 2009

Performance Metrics That Matter
Process Quality Index

PQI is a leading indicator of overall product quality

PQI gives ability to predict whether components that have been unit tested will have down-stream defects in integration, system, and user acceptance testing

Teams can take corrective action and reduce test and rework time

ais

© AIS 2009

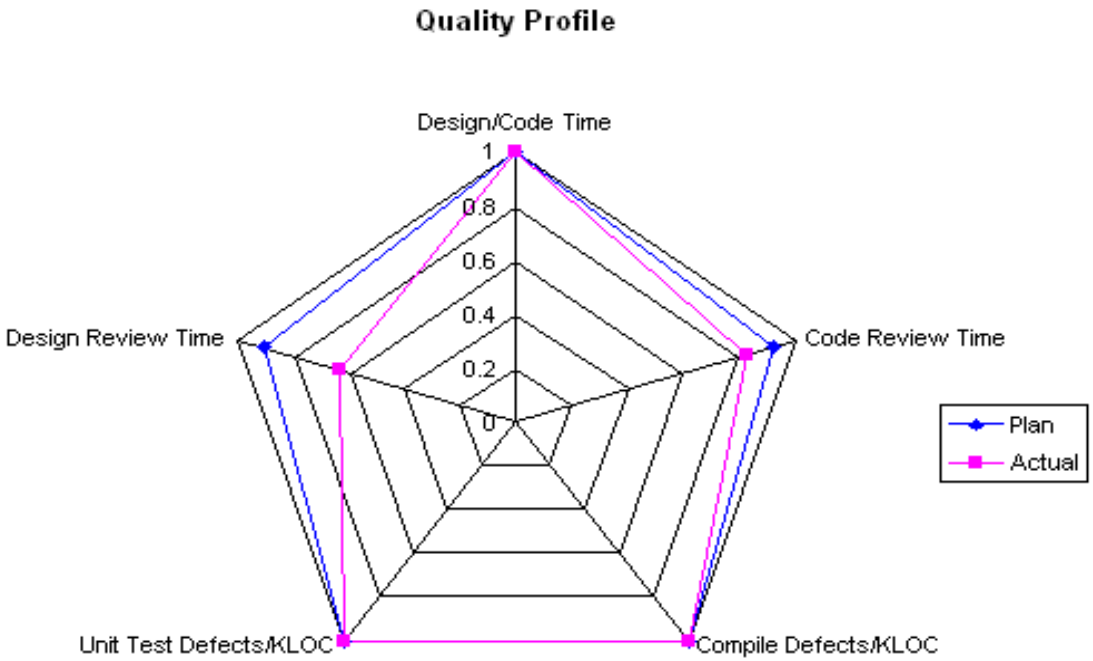
PQI Calculation

1. Design time to Code time comparison
2. Design Review Time as a % of design time (should be 50% or greater)
3. Code Review Time as a % of code time (should be 50% or greater)
4. Compile Defects per KLOC (Compile should find less than 10 defects per KLOC)
5. Unit Test Defects per KLOC (UT should find less than 5 defects per KLOC)

ais

© AIS 2009

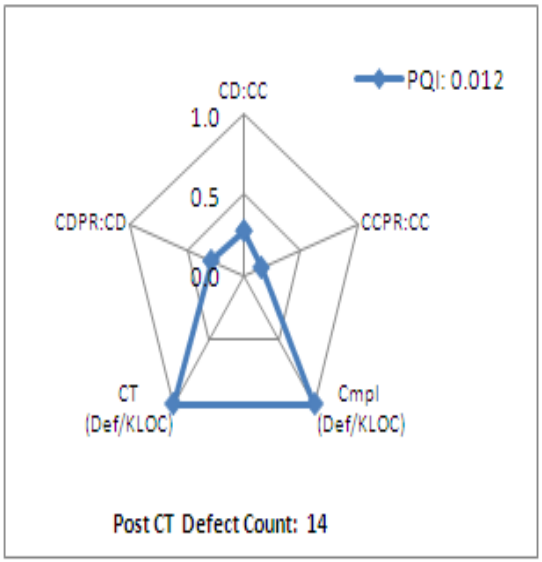
Component Quality Profile

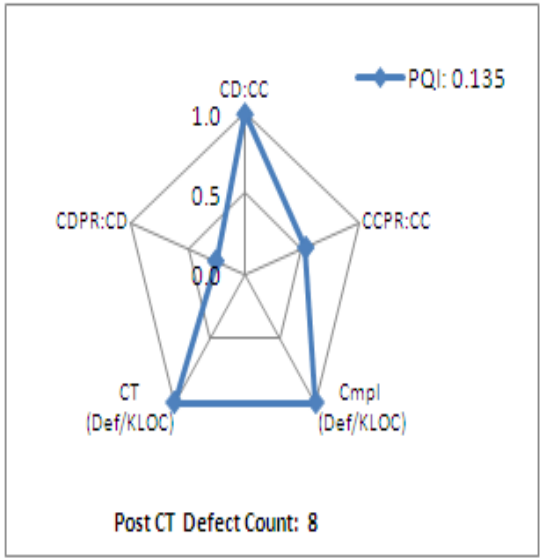


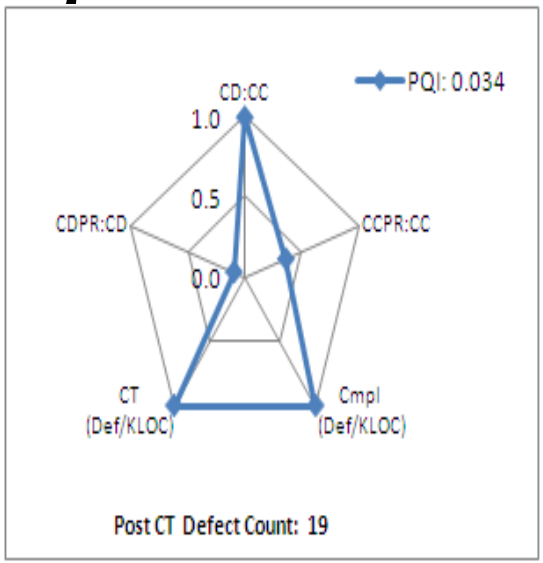
By plotting the five PQI values on radar charts, a component's quality profile can be visually identified in time to take corrective action

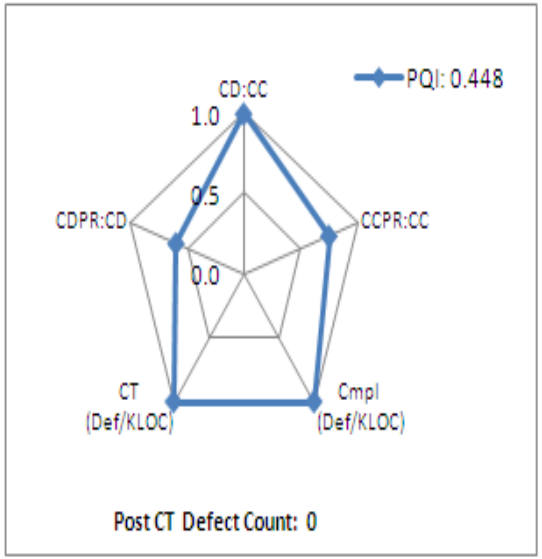
© AIS 2009

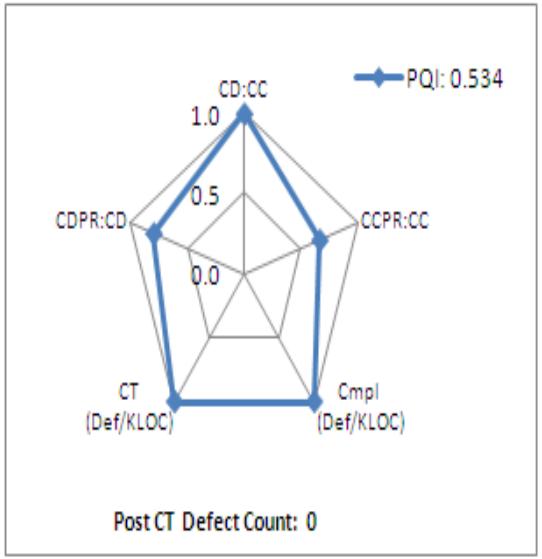
AIS Federal Project

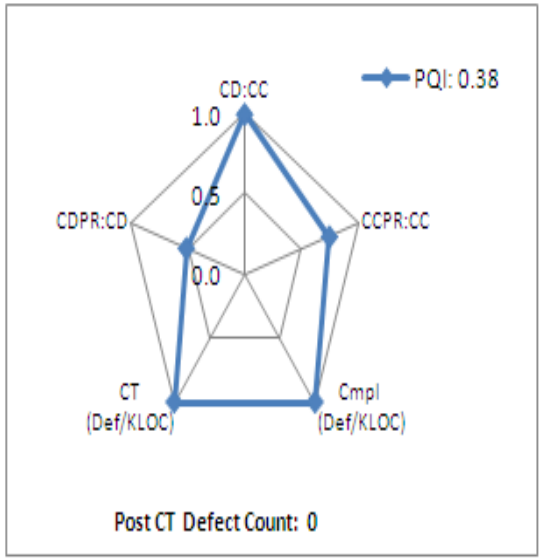












© AIS 2009

Performance Metrics That Matter Percent Defect Free Components

When knowledge workers have been trained and know how to manage themselves, they are capable of giving early warning to management when problems arise

A key metric is the percent of components in the product that are defect free

Management can motivate the technical teams to strive for 100% defect free components when both parties view project success the same way

© AIS 2009

Summary - 1

Developers gather, analyze and report size, time, and defect data

To manage software quality, developers need to manage quality of the software components built by them

Developers practice early defect removal to put the highest quality product into test

© AIS 2009

Summary -2

Process Quality Index (PQI) is a metric that is used to predict whether a component is likely to have downstream defects in integration, system, and acceptance tests

Teams should strive for 100 percent defect free components in the products they build

© AIS 2009

How to Predict Software Quality Early in the Software Development Lifecycle and Produce Secure Software - Module 5

Real Project Examples

Module Overview

Real project data from Microsoft, Intuit, Adobe and AIS using TSP/PSP

Capers Jones ranking of the methods

© AIS 2009

Microsoft IT Project

Weekly Data Summary at Code Complete:

At Code Complete we had spent 1630.8 hours in design and development

Weekly Data	Plan	Actual
Schedule hours for this week	15.0	5.8

Weekly Data Summary at Project End:

At Project End we had spent just 19 additional hours (from code complete) fixing defects and post mortem.

Schedule hours this cycle to date	2052.0	1649.3
Earned value for this week	0.0	0.0

Source: Microsoft's Pilot of TSP Yields Dramatic Results, News at SEI, 2004

© AIS 2009

Microsoft Fixed Schedule Project - 1

Teams negotiated features and resources to fit the schedule

Team members were empowered to stop a project from going into Code, SIT, UAT or Production

Predicted software quality early using metrics

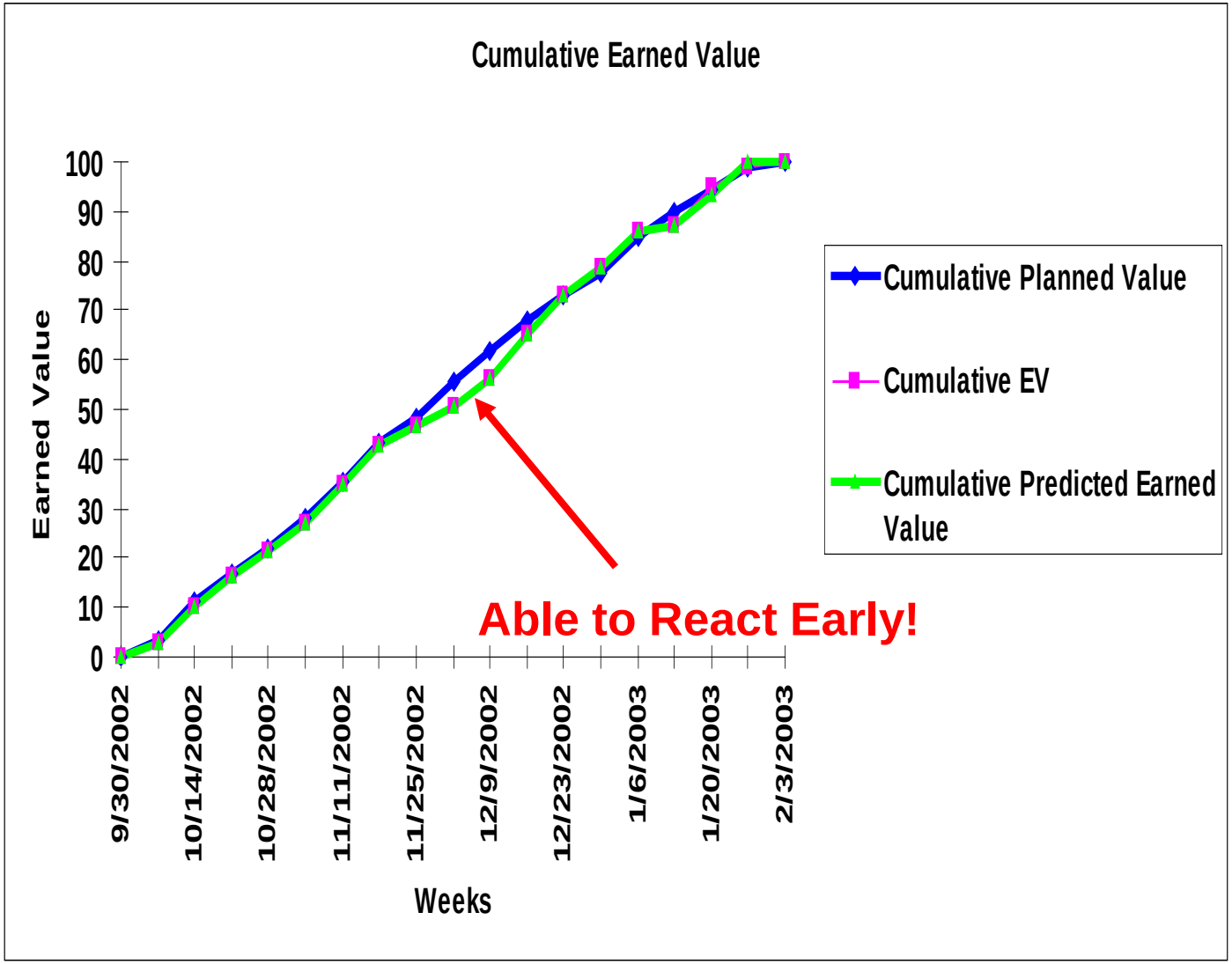
Achieved work life balance

Developers released early from projects

Low budget for post production defects

© AIS 2009

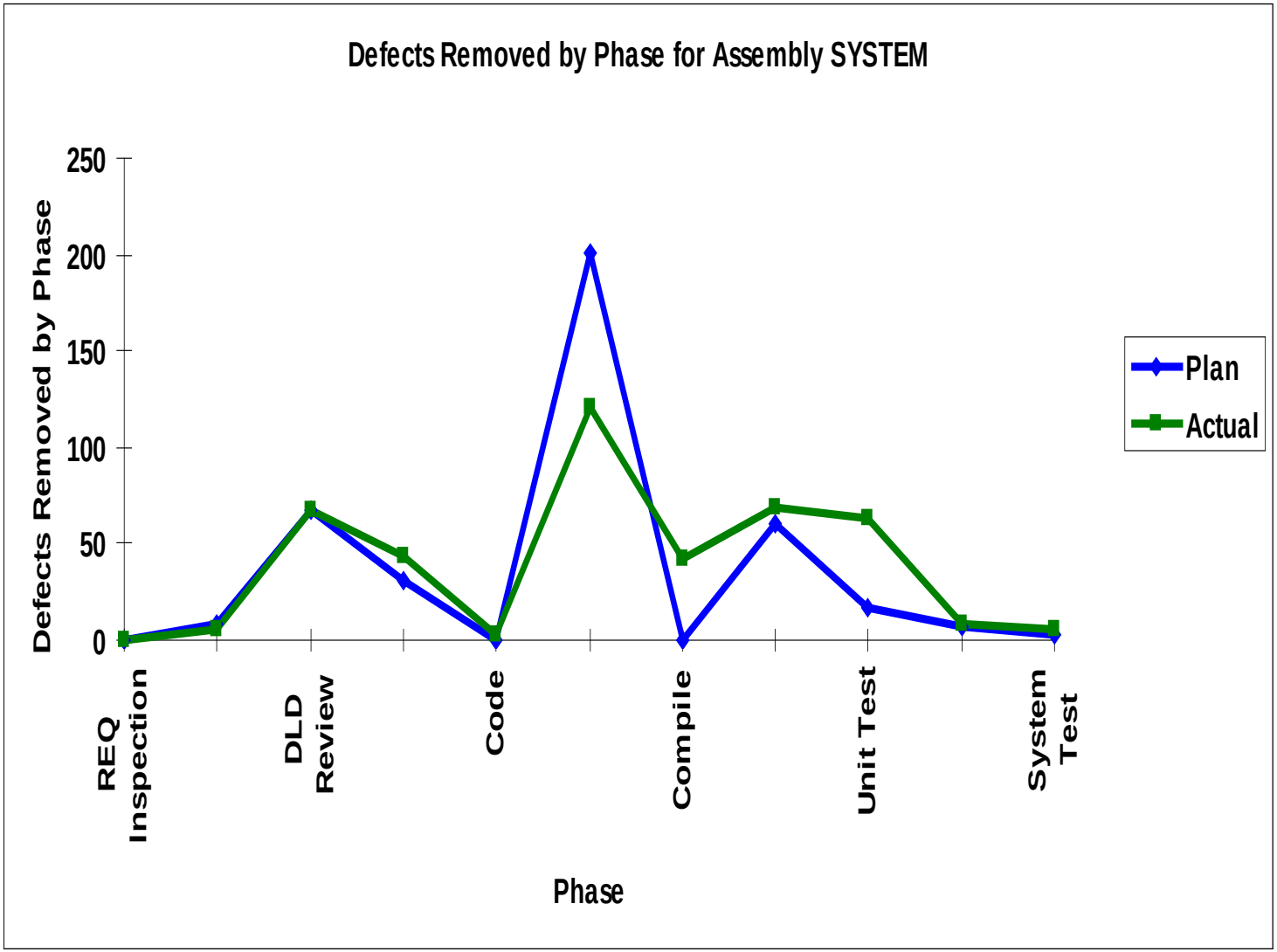
Microsoft Fixed Schedule Project - 2



ais

© AIS 2009

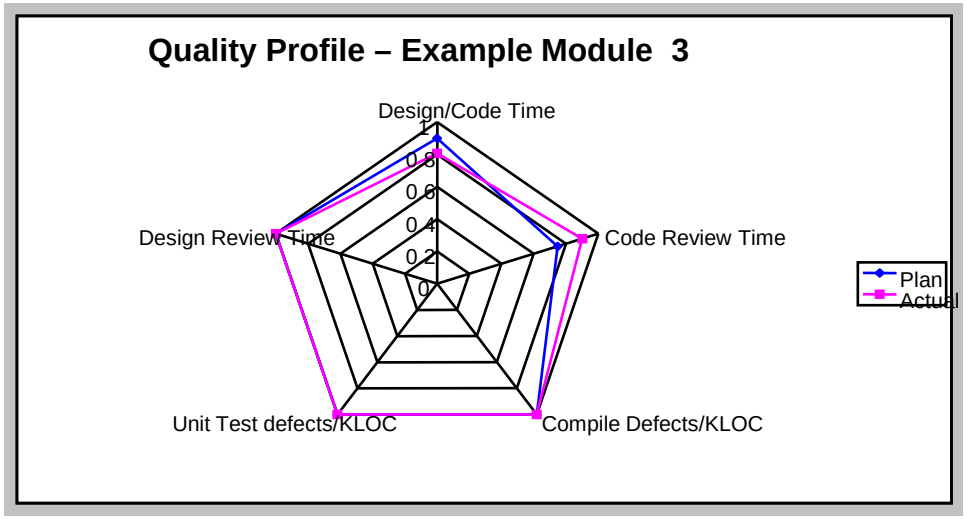
Microsoft Fixed Schedule Project - 3

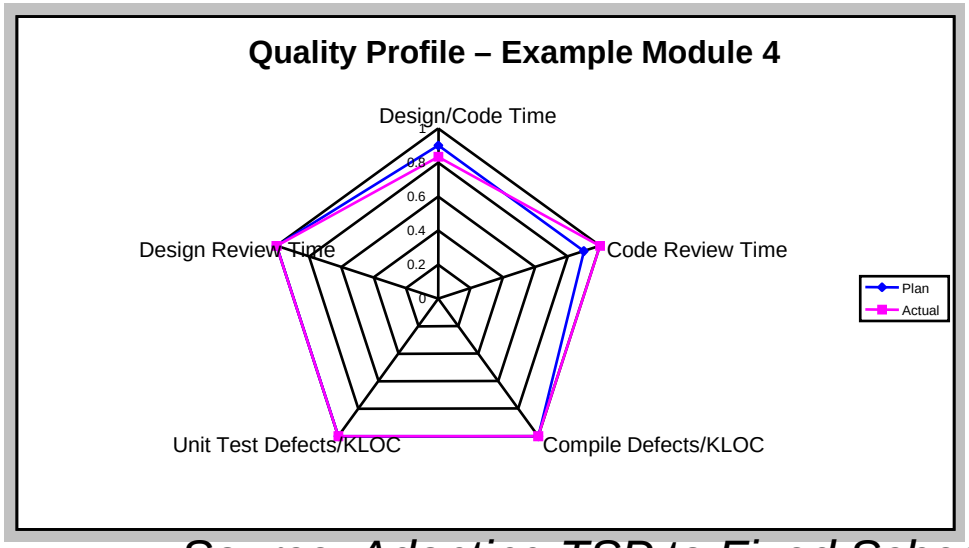


ais

© AIS 2009

Microsoft Fixed Schedule Project - 4





Planned correctly

- Executed correctly
- Planned & Actual PQI > 0.7
- Zero system test defects

Source: Adopting TSP to Fixed Schedule Projects, Microsoft Presentation, TSP Users Group Meeting 2005, Chennai, India

© AIS 2009

Intuit

Reduced project risk by finding 86% of defects before System Testing even started

Engineers doubled available time for developing features by reducing rework and testing time to <10%

Improved quality and time-to-market, while eliminating work that adds no value at Intuit

“Engineers love it… Once they adopt it they can’t imagine going back”

Source: Jim Sartain, Director, Intuit

ais

© AIS 2009

Adobe

Teams spent four times less rework than typical Adobe software projects

Teams removed 90% of total bugs prior to systems test

Teams plan, track, measure, and improve

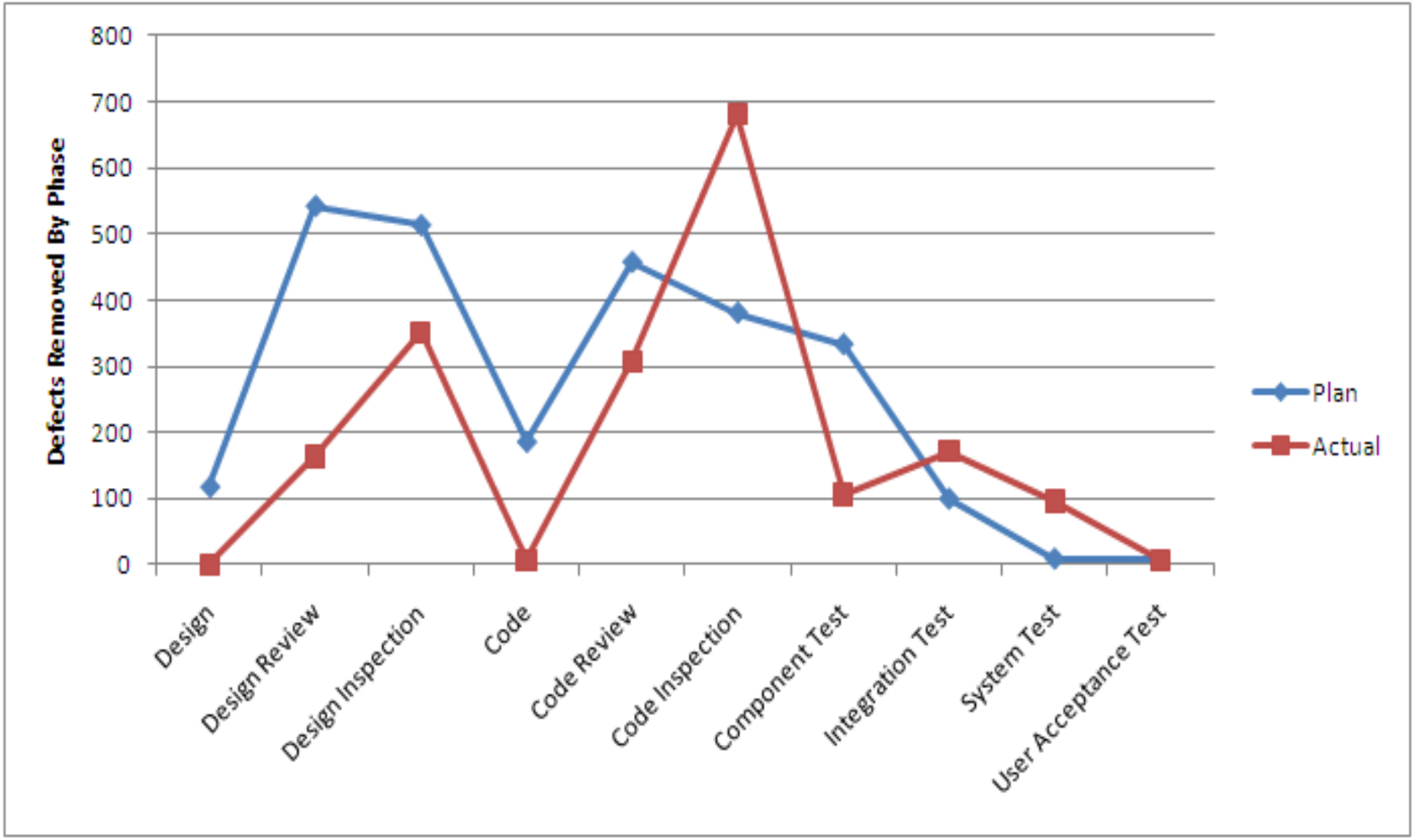
Measurement has encouraged other improvements

- Improved design practices
- Defect analysis to determine common causes and implement improvement methods
- Better estimation methods

Source: The Evolution of Quality at Adobe, SEPG North America, March 20

© AIS 2009

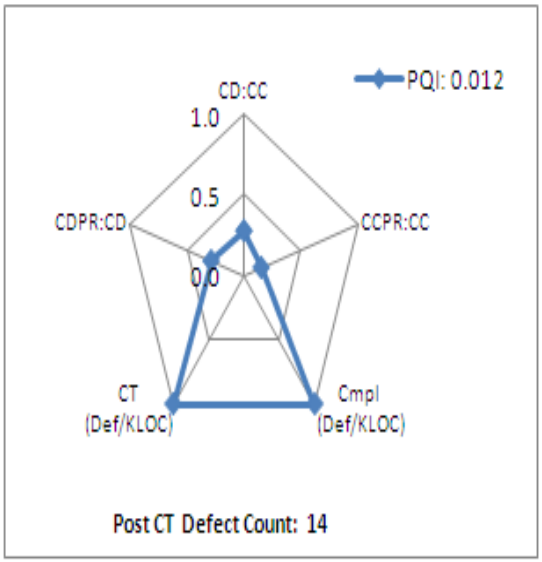
AIS Federal IT Project Results - 1

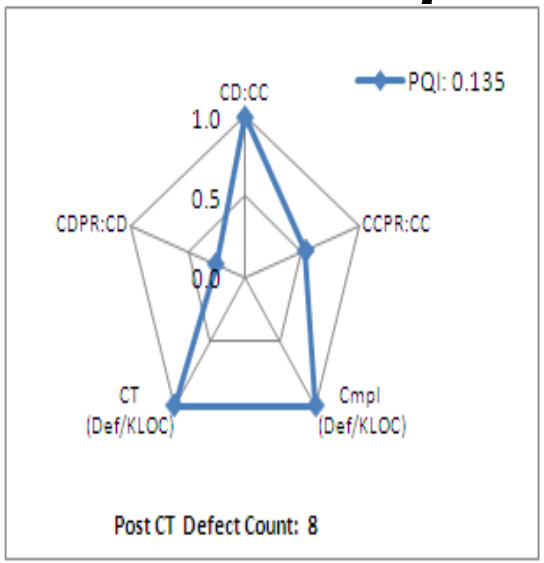


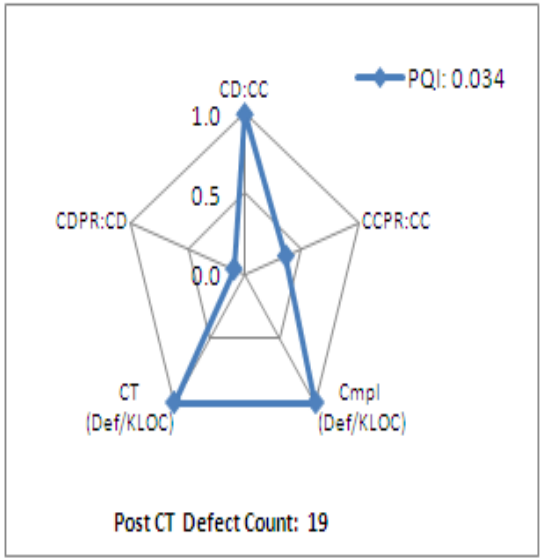
ais

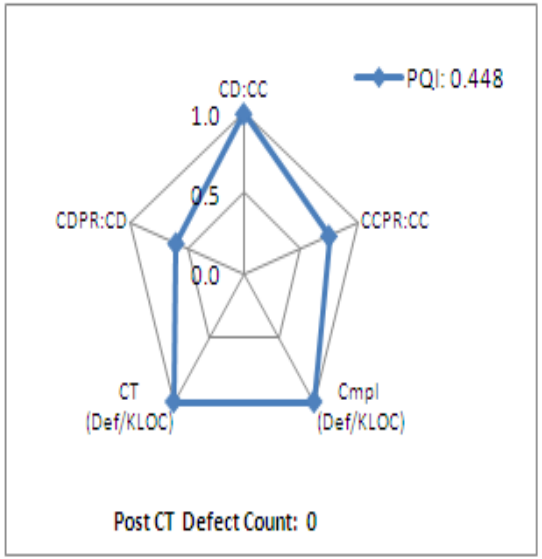
© AIS 2009

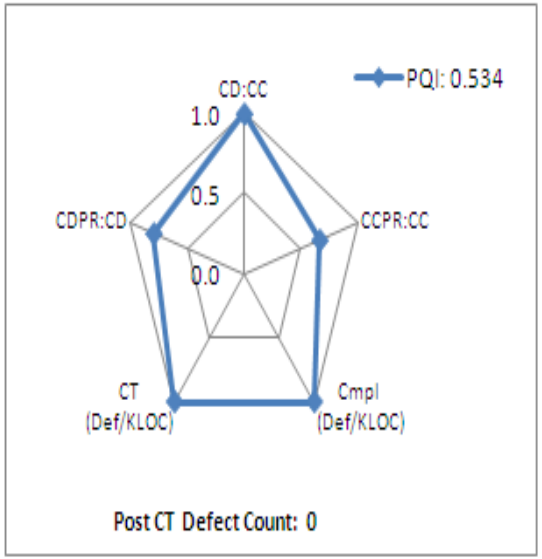
AIS Federal IT Project Results - 2

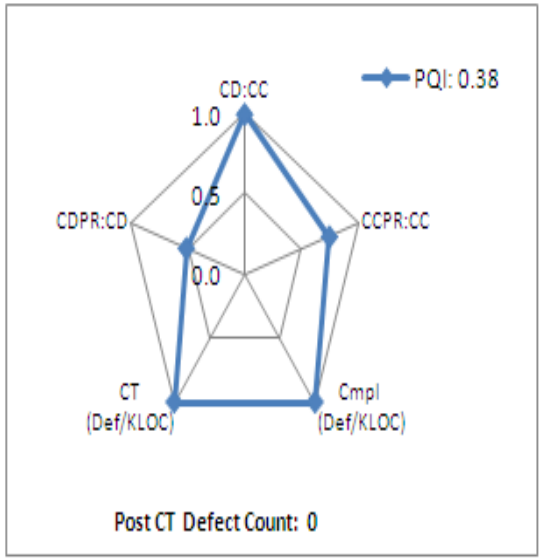












© AIS 2009

Summary

Some of the world's biggest software organizations have reported successful results by empowering developers and teams to successfully negotiate commitments and deliver high quality on schedule and cost

All projects practiced early defect removal and did not rely on testing to find defects

© AIS 2009

How to Predict Software Quality Early in the Software Development Lifecycle and Produce Secure Software - Module 6

Software Center of Excellence

Module Overview

Characteristics of a Software Center of Excellence

Assets of a Software Center of Excellence

Competitive value of Software Center Of Excellence

Costs and savings

Implementation approach and timeline

© AIS 2009

Characteristics of a Software Center of Excellence - 1

- Quality above all
- Process as a key enabler for business success
- Continuous improvement as a way of life
- Level 5 behavior is normal
- Defined roles to support project teams
- Engineers trained to
  - Plan and estimate effort and schedule
  - Measure and manage quality

© AIS 2009

Characteristics of a Software Center of Excellence - 2

- Coaching to help teams and individuals achieve excellence while maintaining product focus
- Aggressive and realistic commitments based on historical data
- Data accuracy and precision
  - Time tracked in minutes
  - All defects recorded
- Tool support for
  - Process repository and maintenance
  - Individual, team and organization historical data – size, time, defects

© AIS 2009

Characteristics of a Software Center of Excellence - 3

Trusting environment

Low staff turnover

Democratic teams having fun on the job

© AIS 2009

© AIS 2009

“Business has only two functions: marketing and innovation. Marketing and innovation produce results. All the rest are costs.”

Peter Drucker

© AIS 2009

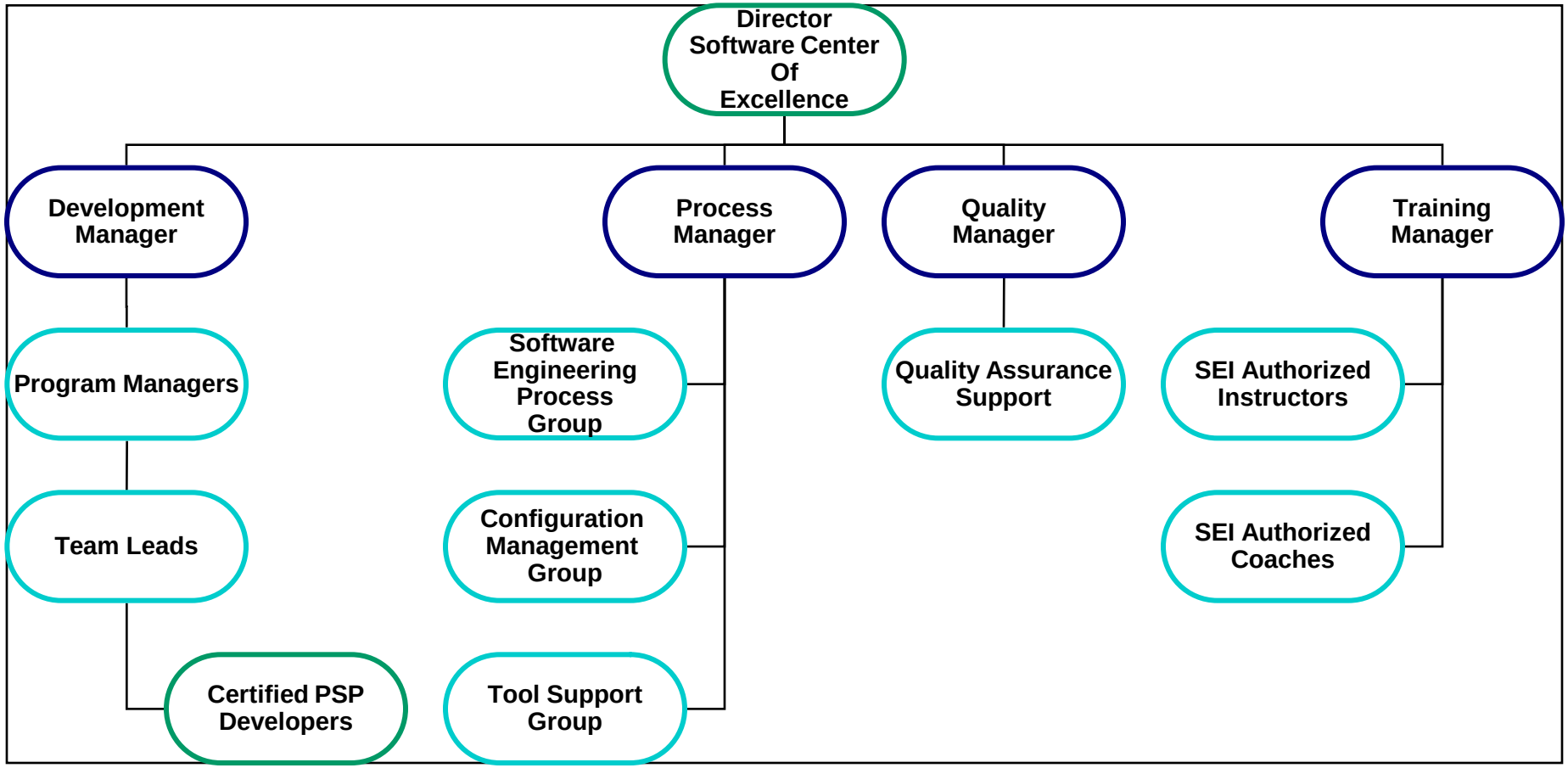
The Top 15 U.S. Software Cost Drivers in Rank Order Circa 2019

1. **The cost of innovation and new kinds of features**
2. **The cost of acquiring reusable components**
3. The cost of requirements changes during development
4. The cost of programming or coding
5. The cost of training and learning software applications
6. The cost of avoiding security flaws
7. The cost of producing paper documents and English words
8. The cost of customer support
9. The cost of meetings and communication
10. The cost of project management
11. The cost of application renovation
12. The cost of litigation for cancelled projects
13. **The cost of finding and fixing bugs**
14. The cost of recovery from security flaws and attacks
15. The cost of cancelled projects

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

© AIS 2009

Organization Chart
Software Center of Excellence



ais

© AIS 2009

Types of Process Repository Assets Software Center of Excellence

Type	Number
Form	59
Policy	44
Process	115
Roadmap	9
Checklists	90
Standard/Guideline	99
Template	97
Tool	60
Other	91
Total	**664**

ais

© AIS 2009

Process Assets by Maturity Level
Software Center of Excellence

CMMI Level	No. of Process Assets
Level 2	266
Level 3	314
Level 4	48
Level 5	36
Total	**664**

ais

© AIS 2009

TSP/PSP Tool Asset
Software Center of Excellence

Features

- Support PSP PROBE (Proxy Based Estimating)
- Earned Value – Individual, Team
- Time tracking
- Defect tracking
- Quality profile – Process Quality Index
- Tracking reports – Quality, Schedule, Goals

© AIS 2009

Competitive Value of the Software Center of Excellence - 1

Intangibles

- Joy in work
- Work life balance
- Reduced employee turnover
- Historical data – individual, team, organization
- Learning organization
- Ability to negotiate responsible commitments
- Brand recognition for quality and dependability

© AIS 2009

Competitive Value of the Software Center of Excellence - 2

Proven results from CMMI Level 5

Category	**CMMI ML3**	**CMMI ML5**
Test time as % of development	30 to 35%	20% to 25%
Acceptance test defects/KLOC	4.73	1.05

ais

© AIS 2009

Competitive Value of the Software Center of Excellence - 3

Proven results from TSP projects

Category	**Average**	**Range**
Effort deviation	-4%	-36% to +26%
Schedule deviation	6%	-8% to +23%
System test duration (days/KLOC)		0.1 to 1.1
System test defect density (defects/KLOC)		0.1 to 1.1
Acceptance test defect density		0 to 0.35

Source: Watts Humphrey, Wining with Software

ais

© AIS 2009

Cost Savings
Software Center of Excellence

Reduced engineering days in test

Increased engineering development days without increasing head count

Increased Lines of Code or Function Points developed per year

Reduced engineering repair/rework days

ais

© AIS 2009

Investment and Expenses
Software Center of Excellence

Investment

- Center startup
- PSP engineer training
- Manager training

On-going expense

- Center management overhead
- Coaching support
- Configuration management, quality assurance and tool support

ais

© AIS 2009

Return on Investment

For an example organization with 100 software engineers, 20 managers, and 4 coaches the 5-year present-value costs and savings can be as high as $2.8 Million and $11.1 Million respectively

At 6% annual interest rate, the ROI can be as high as 397%

Source: Watts Humphrey, Winning with Software

© AIS 2009

Software Center of Excellence Implementation Approach - 1

Establish a quality policy

- “Doing poor-quality work to meet a tight schedule is not acceptable”
- Implement Process Improvement Proposal (PIP) mechanism

Align the business strategic objectives with internal process and learning and growth objectives

- Use Balanced Scorecard, Goal Questions Metrics or other methods to show the linkage
- Determine the “vital few” data for tracking progress

ais

© AIS 2009

Software Center of Excellence Implementation Approach - 2

Sponsor Software Center of Excellence as the single most important business strategic initiative

Establish organization-wide consensus and acceptance of the center's goals

- Capability to deliver substantially defect free products on predictable schedule, cost
- Achieve CMMI Maturity Level 5 in 24 to 36 months
- Joy in work

© AIS 2009

Software Center of Excellence Implementation Approach - 3

Train the center managers

Select and install tools

Train engineers in PSP and get them certified

- SEI official courses
- SEI-authorized PSP instructors
- SEI PSP Developer Certification
- Level 5 individuals

ais

© AIS 2009

Software Center of Excellence Implementation Approach - 4

Launch TSP teams

- SEI TSP site license
- SEI-authorized TSP coach
- Level 5 teams
- TSP Organization Certification (Forthcoming from SEI)

Train and authorize internal PSP instructors and coaches

Accelerate TSP adoption and team launches

ais

© AIS 2009

Software Center of Excellence Implementation Approach - 5

Support the teams to execute at CMMI Level 5

- Management oversight through quarterly review
- Support for process management, CM, QA
- Tool support

Conduct gap analysis

- Completed projects
- CMMI process areas

Implement action plan

ais

© AIS 2009

Software Center of Excellence Implementation Approach - 6

Conduct mini-assessment

- Confirm SCAMPI readiness

Sponsor SCAMPI A

- Select SEI-authorized high maturity lead appraiser
- Train assessment team

Conduct SCAMPI A

Achieve ML5 and celebrate !!!!

ais

© AIS 2009

Training

Course	Attendees	Days
Strategy Seminar	Senior Managers	1.0
Managing TSP Teams	Team leads, Managers	3.0
PSP for Engineers	Engineers	10.0
Software Requirements	Engineers	2.0
Software Inspections	Engineers	2.0
Tools	Team Leads, Engineers	1.0

© AIS 2009

Software Center of Excellence Implementation Time Line - 1

Kickoff	Month 1
Establish quality policy	
Align business and process objectives	
Establish Software Center of Excellence goals	
Announce Software Center of Excellence initiative	
Phase 1	
Executives/Managers trained	Month 2
Engineers trained – Batch 1	Month 2
TSP teams 1 & 2 launched	Month 3
Quarterly review	Month 6

© AIS 2009

Software Center of Excellence Implementation Time Line - 2

Phase 2

Engineers trained - Batch 2	Month 7
TSP teams 3 & 4 launched	Month 8
Quarterly review	Month 9
SEPG chartered	Month 10
Postmortem of completed projects	Month 11
Integrate CMMI, TSP	Month 12
Quarterly review	Month 12

© AIS 2009

Software Center of Excellence Implementation Time Line - 3

Phase 3

Accelerate TSP/PSP adoption	Month 13
Quarterly review	Month 15
Postmortem of completed projects	Month 16
CMMI L5 Gap Analysis	Month 17
Implement action items	Month 18
Quarterly review	Month 18

© AIS 2009

Software Center of Excellence Implementation Time Line - 4

Phase 4

Quarterly review	Month 21
Mini-assessment	Month 22
Sponsor SCAMPI A	Month 23
Conduct SCAMPI A	Month 24

© AIS 2009

Summary - 1

A Software Center of Excellence is a competitive asset

Significant savings and return on investment are achievable due to improved quality, reduced time in test, reduced rework

Our suggested implementation approach and timeline include TSP introduction, CMMI integration and SCAMPI A for Maturity Level 5

© AIS 2009

Summary - 2

SEPG must take the leadership to improve the organization's competitive position, and achieve business strategic objectives for financial, customer, and employee perspectives

The time to get started is now and begin the journey towards joy in work

ais

© AIS 2009

How to Predict Software Quality Early in the Software Development Lifecycle and Produce Secure Software - Module 7

Joy in Work

Main Points - 1

Unhappy people rarely do quality work

People generally get more satisfaction by doing the job the proper way

Need transformation to a new management model – “people jointly working for themselves in democratic teams”

© AIS 2009

Main Points - 2

Transformation begins with the individual

The Software Center of Excellence based on SEI CMMI, TSP, and PSP enables self managed teams and facilitates the management obligation to realize the full potential of each and every professional

© AIS 2009

Job Satisfaction – Oxymoron ?

Conference Board Survey

	1995	2000	2007
Percent saying they are satisfied with their current jobs	59	51	49

Source: The Conference Board , February 2007

© AIS 2009

Human Motivation

“ What motivates technical professionals is what motivates volunteers, who have to get more satisfaction from their work than paid employees, precisely because they do not get a paycheck. ”

Peter Drucker

© AIS 2009

Negotiated Commitments - Developers

When pressed for early deliveries, the responsible team members say

"I understand your requirements, I will do my utmost to meet it, but until I make a plan, I can not responsibly commit to a date"

© AIS 2009

Negotiated Commitments - Managers

When pressed for early deliveries, the responsible managers say

> "I trust you to create an aggressive and realistic plan, I will review the plan, but I will not commit you to a date that you can not meet"

© AIS 2009

The Executive Transformation - 1

Pre-TSP	TSP
"If we don't meet the customer's schedule expectations, we will lose the business"	"Show me a plan to meet customer's aggressive schedule. I will not commit you to a date, you can't meet"
"I will gamble on a vague promise"	"We will win with a firm commitment based on a plan"
"I will tell you what must happen"	"Let us analyze the data and determine what will happen"

ais

© AIS 2009

The Executive Transformation - 2

Pre-TSP	TSP
"Where are you on the schedule?"	"How is the quality in reviews, inspections, and test?"
"What have you done for me lately?"	"What are you learning today that will help us to meet future challenges?"
"I will authorize overtime"	"Have fun doing this project"

ais

© AIS 2009

Do You Agree?

"The best engineers and scientists don't work for a company, a university, or a laboratory; they really work for themselves."

Watts Humphrey

"Successful careers are not planned. They develop when people are prepared for opportunities because they know their strengths, their method of work, and their values."

Peter Drucker

"Anyone that enjoys his work is a pleasure to work with."

Edwards Deming

© AIS 2009

Necessary Conditions For Self-Motivation	The Source Of Self-Motivation	Outcome of Fulfilled Motivation
Individual Emotional and Mental Health Safety and Trust Dignity and Self-Esteem Autonomy and Freedom Perspective	Calling Service Purpose Connectedness Interesting work Working with others **MEANING** Challenge Excelling Creating Learning Beauty Love	**Joy**

Source: Abolishing Performance Appraisals, Coens and Jenkins, 2000

© AIS 2009

What does

“FUN ON THE JOB”

Mean to you?

© AIS 2009

Contact Information

Girish Seshagiri

Advanced Information Services Inc.

(703) 426 2790

Email: girish.seshagiri@advinfo.net

Website: www.advinfo.net

© AIS 2009

For More Information - 1

Visit the SEI Website for CMMI, TSP

http://www.sei.cmu.edu/cmmi

http://www.sei.cmu.edu/tsp

See the books

Winning with Software, by Watts Humphrey, Addison-Wesley, 2002

PSP: A Self-Improvement Process for Software Engineers, by Watts Humphrey, Addison-Wesley, 2005.

TSP: Leading a Development Team, by Watts Humphrey, Addison-Wesley, 2006

TSP: Coaching Development Teams, by Watts Humphrey, Addison-Wesley, 2006

© AIS 2009

For More Information - 2

Contact a TSP transition partner

http://www.sei.cmu.edu/collaborating/partners/trans.part.psp.html

Contact SEI customer relations

Phone, voice mail, and on-demand FAX: 412/268-5800

E-mail: customer-relations@sei.cmu.edu

© AIS 2009