

---

# A PSP Commercial Project

Rob Tonneberger

# This presentation is in cooperation with



**Carnegie Mellon  
Software Engineering Institute**

Some material in this presentation is © Carnegie Mellon University, used with permission.

All other material is © 2006 Northern Horizons Inc., Brookline, NH 03033

The Software Engineering Institute (SEI) of Carnegie Mellon University is a federally funded research and development center sponsored by the U.S. Department of Defense through the Office of the Under Secretary of Defense for Acquisition, Technology, and Logistics.

® Capability Maturity Model, Carnegie Mellon, CMM, CMMI, are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

<sup>SM</sup> CMM Integration, Personal Software Process, PSP, SEI, SEPG, Team Software Process, and TSP are service marks of Carnegie Mellon University.

# Present State of Software Development

---

**Now, let's test this bridge . . . .**



# Present State of Software Development

---

Oops !



# Personal Software Process

---

- PSP is a self-improvement process that helps you to control, manage and improve the way you work. A personally developed process that guides your work.
- It is a structured framework of forms, guidelines and procedures for developing software.
- Properly used, the PSP provides the data you need to make and meet commitments, and it makes the routine elements of your job more predictable and efficient.
- The PSP's sole purpose is to help you improve your software engineering skills.

*A Self-Improvement Process for Software Engineers* by Watts Humphrey

# PSP Principles

---

- Engineers must plan their work and they must base their plans on personal data.
- Engineers must measure their work and use their results to improve.
- Engineers must feel personally responsible for the quality of their work. Superior products are not produced by accident; engineers must strive to do quality work.
- It costs less to find and fix defects earlier in a process than later.
- It is more efficient to prevent defects than to find and fix them.
- The right way is always the fastest and cheapest way to do a job.

*A Discipline for Software Engineering* by Watts Humphrey

# TOPICS

---

- Project Description
- Development
- PSP Results
- Summary

# PROJECT DESCRIPTION

---

- Product
- Organization
- F/W Developer Selection



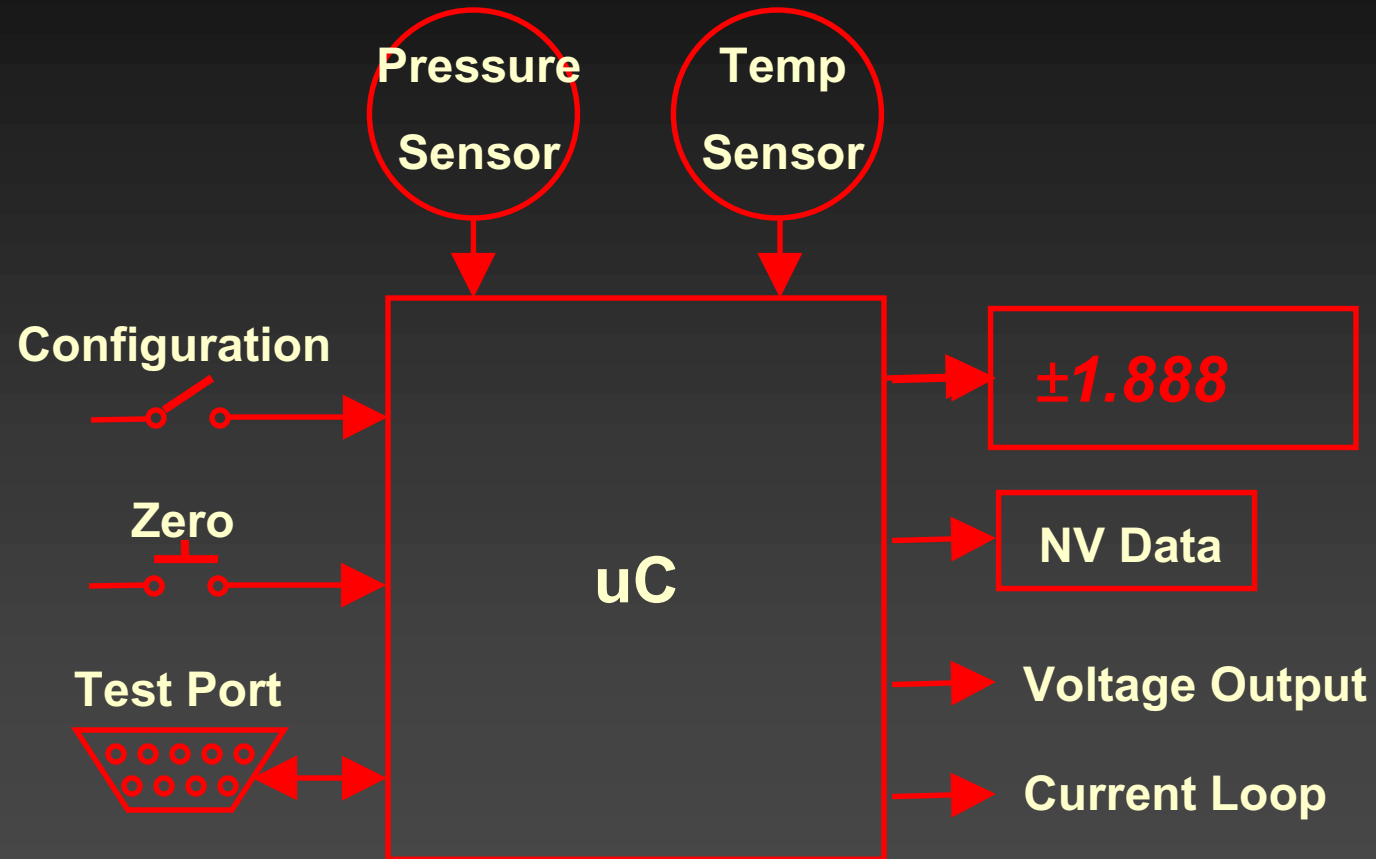
# The Product

---

A small air temperature and pressure monitor that is installed into HVAC ductwork.



# The Product



# Project Organization

---



# F/W Developer Selection

---

- Contract firmware developer
  - PSP and TSP trained by the SEI
  - Personal productivity and defect densities
- Project lead researched PSP/TSP before arriving at decision to hire
- Project lead mandates PSP use on project

# DEVELOPMENT

---

- Planning
  - Estimating
  - Tracking
  - Status
- Requirements
- Implementation
- System Test
- Postmortem

# Planning Estimating

---

## Planning Tool for estimating

- User enters estimated document and code sizes
- User enters work rates (i.e., 25 LOC/hr.)
- User enters estimated defects injected per phase
- User enters his workdays
- Tools estimates defects removed per phase
- Tool prepares a schedule

# Planning Tracking

---

## Planning Tool for tracking

- User enters time on each task
- User enters defects found by phase
- User enters date task is completed and its actual size

# Planning Status

---

## Planning Tool for weekly status:

- Planned vs. Earned value\*
- Planned vs. Actual hours per week\*\*
- Defect Density by phase (defects/KLOC)
- Review Rates (pages/hour or LOC/hour)
- etc.

\* Each task is estimated to take a percentage of the total estimated project hours (PV). You receive the EV when the task is completed. A completed program has 100 EV.

\*\* On task hours do not include emailing, phone, meetings, breaks, etc.



# DEVELOPMENT

---

- Planning
- Requirements
  - Software Design Specification\* (SDS)
  - Software Development Process (SDP)
- Implementation
- System Test
- Postmortem

\* Typically this information is in a separate requirements document

# Requirements SDS

---

## Software Design Specification

- Interfaces
- Communication
- Architectural requirements
  - Operating Modes
  - Data Flow
  - Timing
  - Exception Handling

# Requirements SDP

---

## Software Development Process

- Best practices
- Development process
- Development procedures & forms
- Defect removal phases

# Requirements SDP

---

## Some Best Practices

- Estimates based on personal history
- Designing software before coding
- Coding Standard
- Personal review of work products
- Peer inspection of work products
- Checklists for reviews and inspections
- Unit testing of each component

# Requirements SDP

---

## Development Process

- Planning
- Requirements
- High-level design
- Low-level designs
- Reviews and Inspections
- Unit, Integration & System tests
- Customer acceptance

# Requirements SDP

## Process Procedures

- Planning Script
- Code Review Script
- etc.

Step	Purpose	To guide the development of small programs
	Entry Criteria	<ul style="list-style-type: none"><li>• Requirements</li><li>• ...</li><li>• Defect Type Standard</li></ul>
1	Design	<ul style="list-style-type: none"><li>• ...</li></ul>
2	Design Review	<ul style="list-style-type: none"><li>• Review design by following Design Review Checklist</li><li>• Fix all defects found</li><li>• Record defects in Defect Recording Log</li><li>• Record time in Time Recording Log</li></ul>

# Requirements SDP

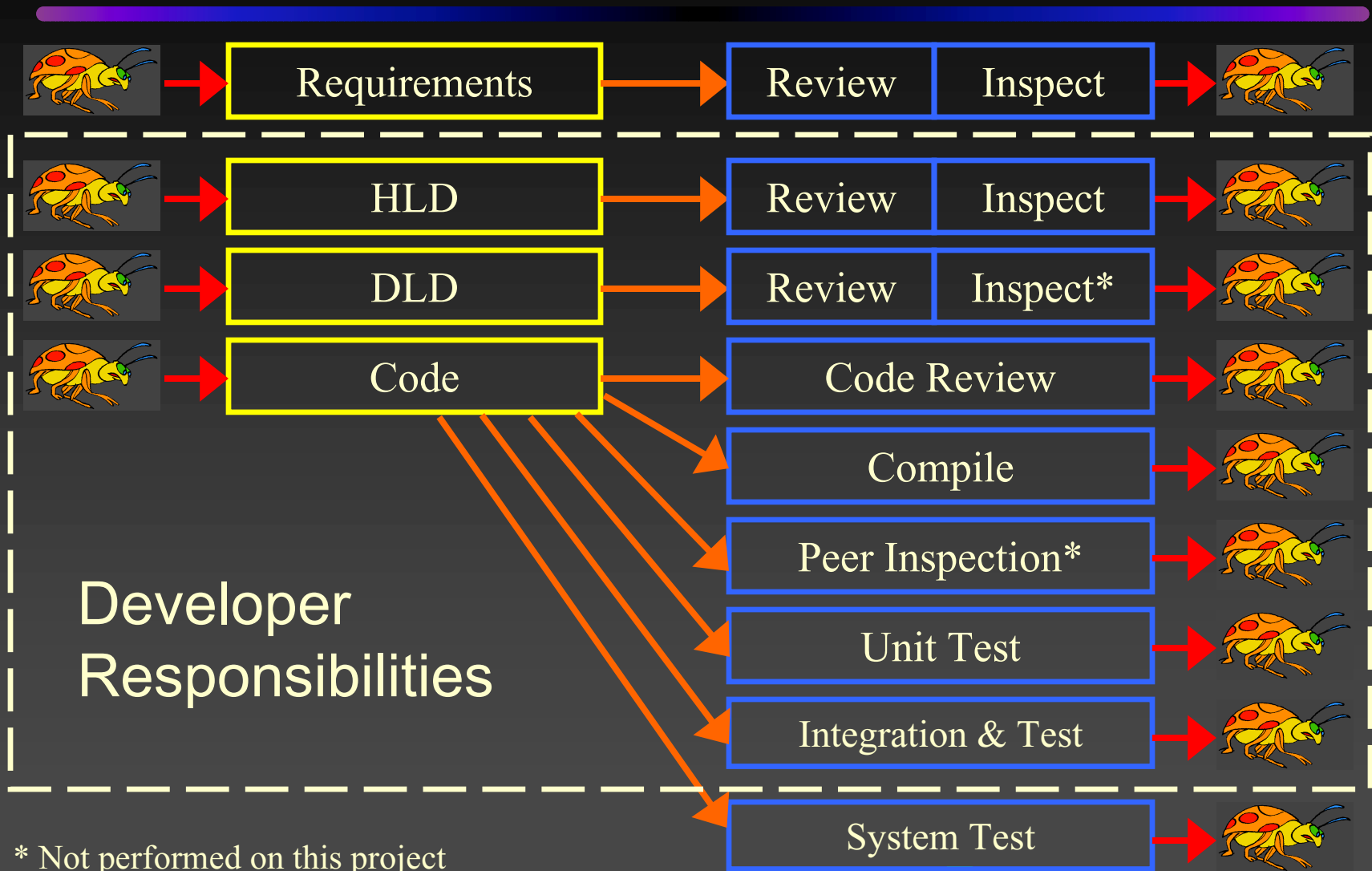
## Process Forms

- Project Plan Summary
- Code Review Checklist
- etc.

--- Functions ---

		Lcd	Timer	Uart
<b>Initialization</b>	<b>Check variable &amp; parameter initialization</b> <ul style="list-style-type: none"> <li>•at program initiation</li> <li>•at start of every loop</li> <li>•at function entry</li> </ul>	v v v	v v v	
<b>Calls</b>	<b>Check function call formats:</b> <ul style="list-style-type: none"> <li>•pointers</li> <li>•parameters</li> <li>•use of "*" and "&amp;"</li> </ul>	v v v	v	

# Requirements SDP



\* Not performed on this project



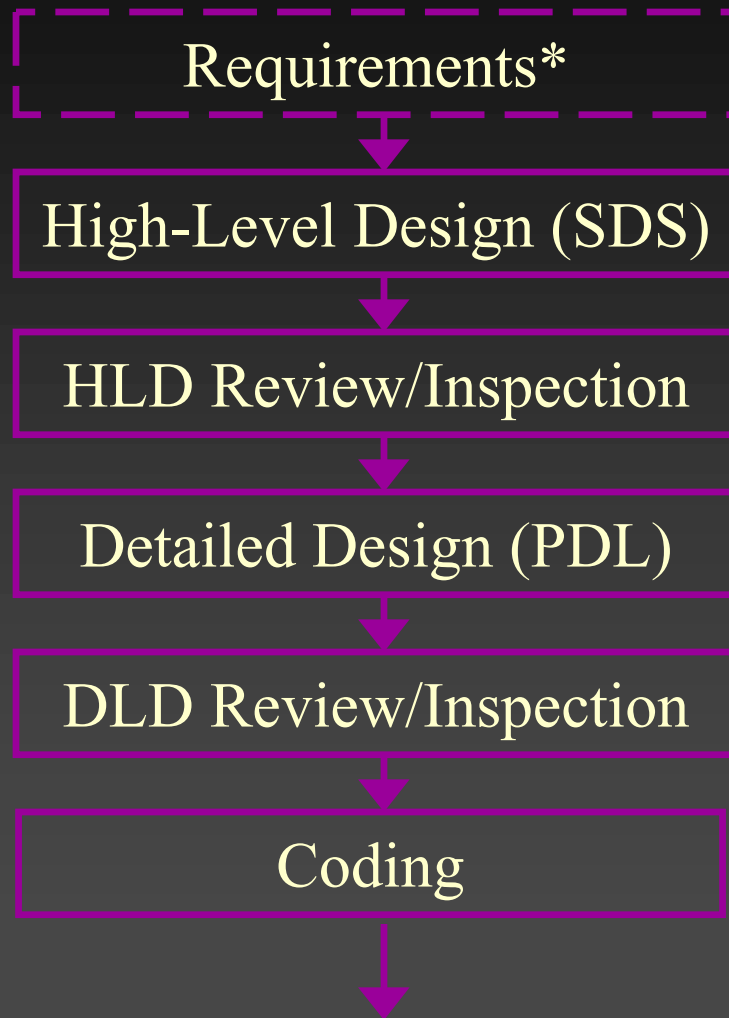
# DEVELOPMENT

---

- Planning
- Requirements
- Implementation
- System Test
- Postmortem

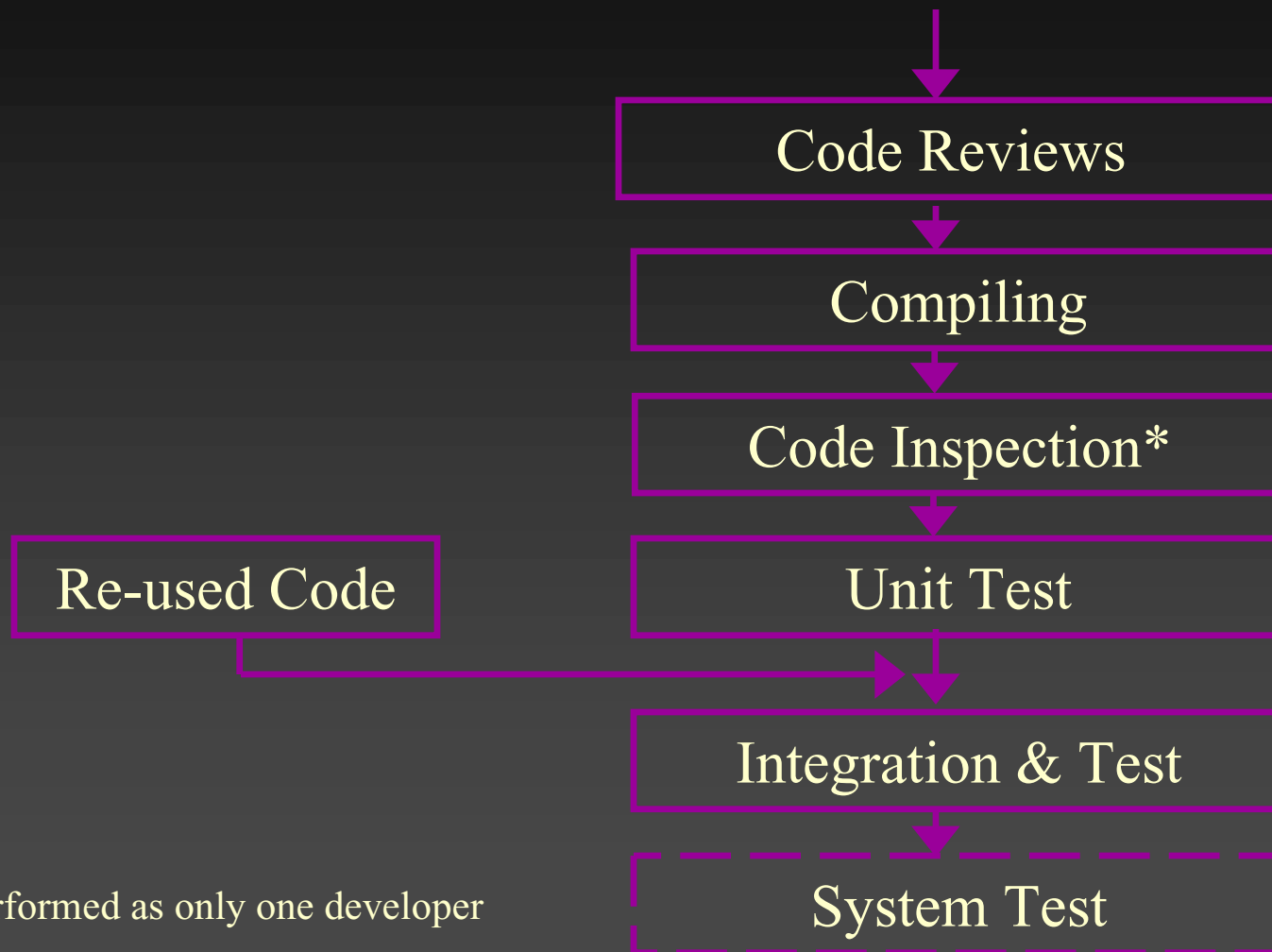
# Implementation

- Architecture
- Module descriptions



\* Initial SDS

# Implementation



\* Not performed as only one developer

# DEVELOPMENT

---

- Planning
- Requirements
- Implementation
- System Test
- Postmortem

# System Test

---

- Validate product
  - Against requirements
  - Against design documents
  - Environmental
  - Calibration
- In-system testing

With his own thorough reviews, inspections and testing, the PSP developer enters system test confident that his software is virtually defect free.

# DEVELOPMENT

---

- Planning
- Requirements
- Implementation
- System Test
- Postmortem

The PSP developer asks himself, “From the collected data and my experience in this project, how can I improve?”

# Postmortem

---

- Ensure all time and defect data are collected
- Print summaries and graphs from planning tool
- Review process for improvements
  - Process scripts
  - Review/inspection checklists
- Review performance improvements
  - Estimating
  - Productivity
  - Defect reductions
- Set goals for next project

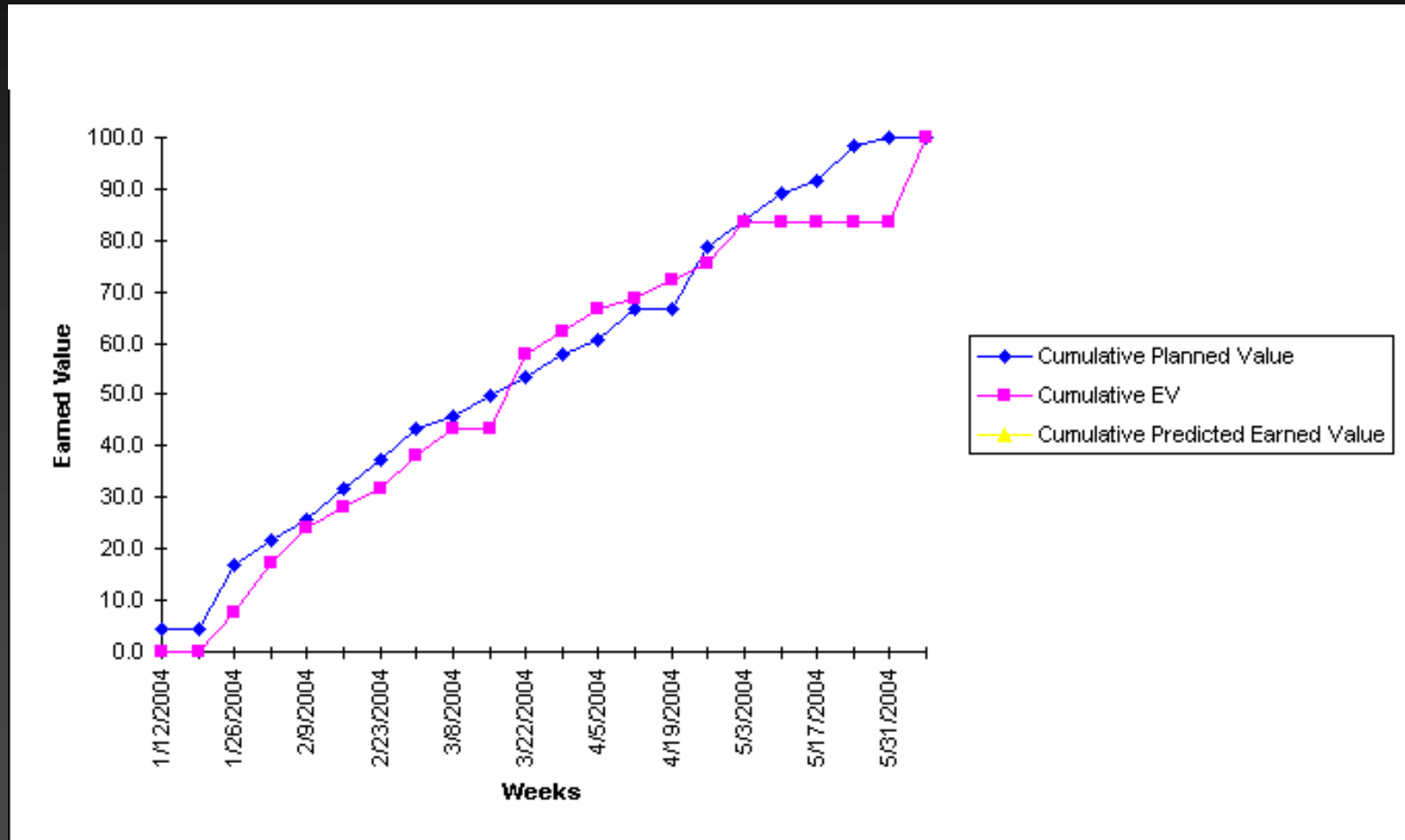
# PSP RESULTS

---

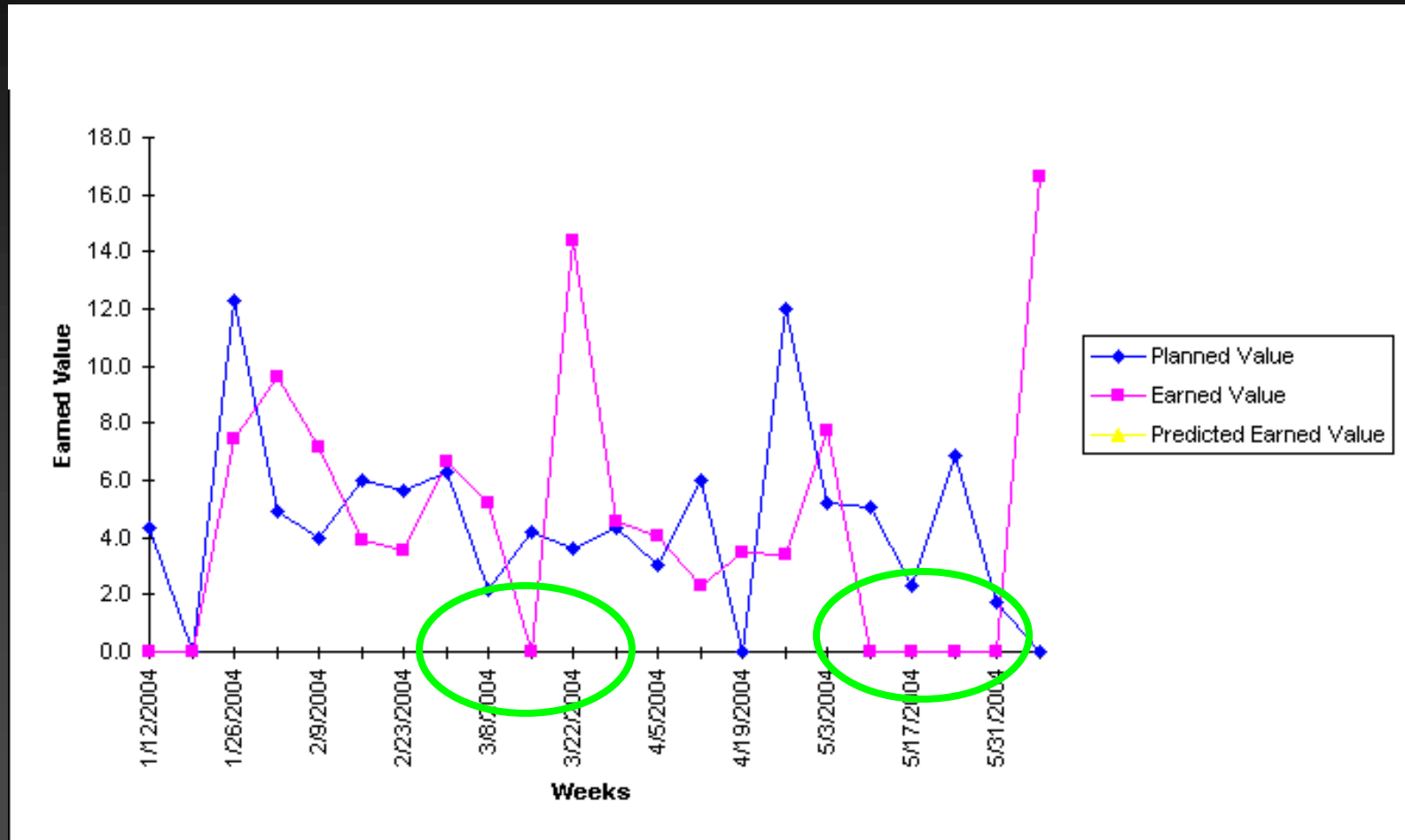
- Planning
  - Cumulative Earned value
  - Earned value per week
  - Cumulative Hours
  - Hours per Week
  - Time in Phase
- Productivity
- Quality



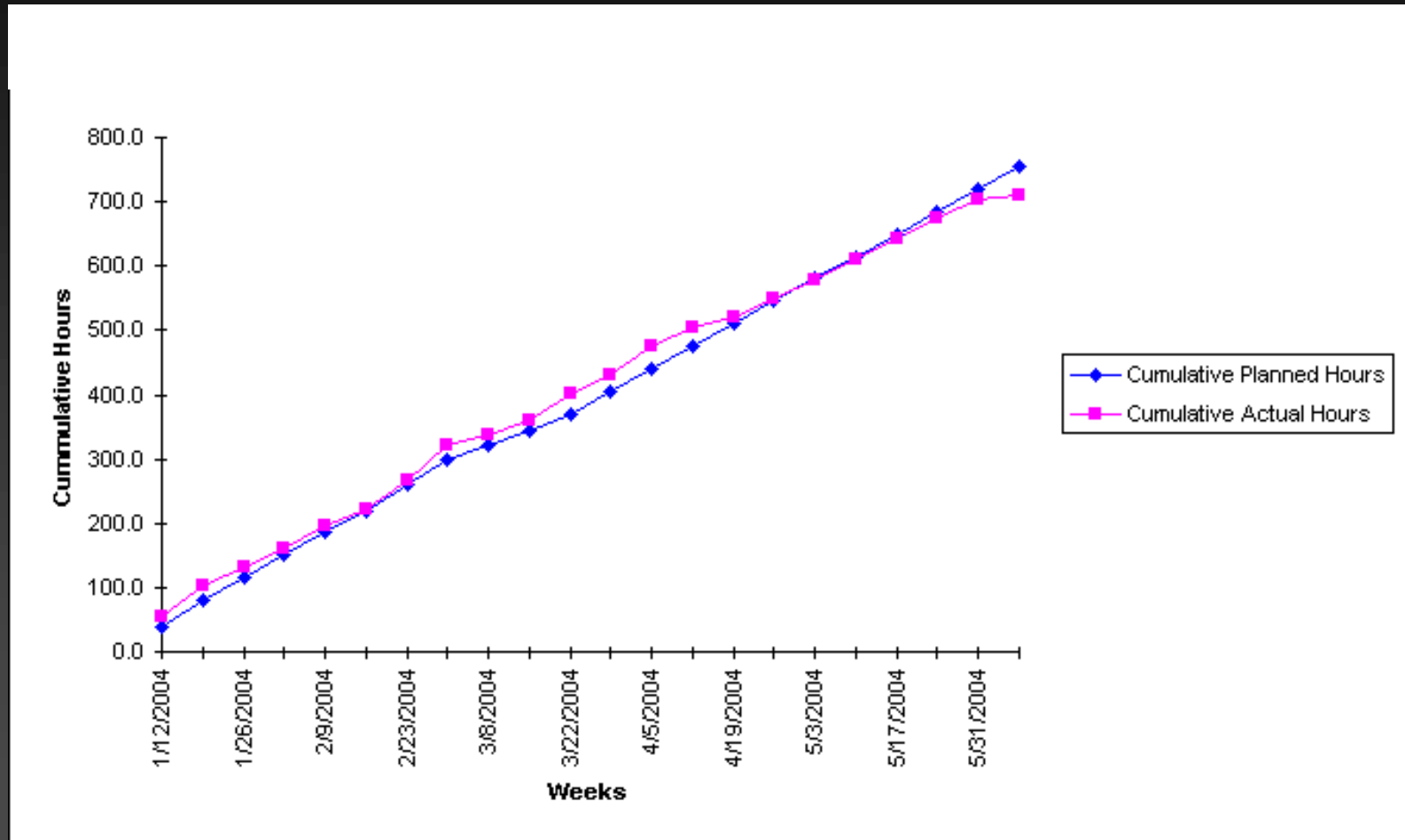
# Planning Cumulative Earned Value



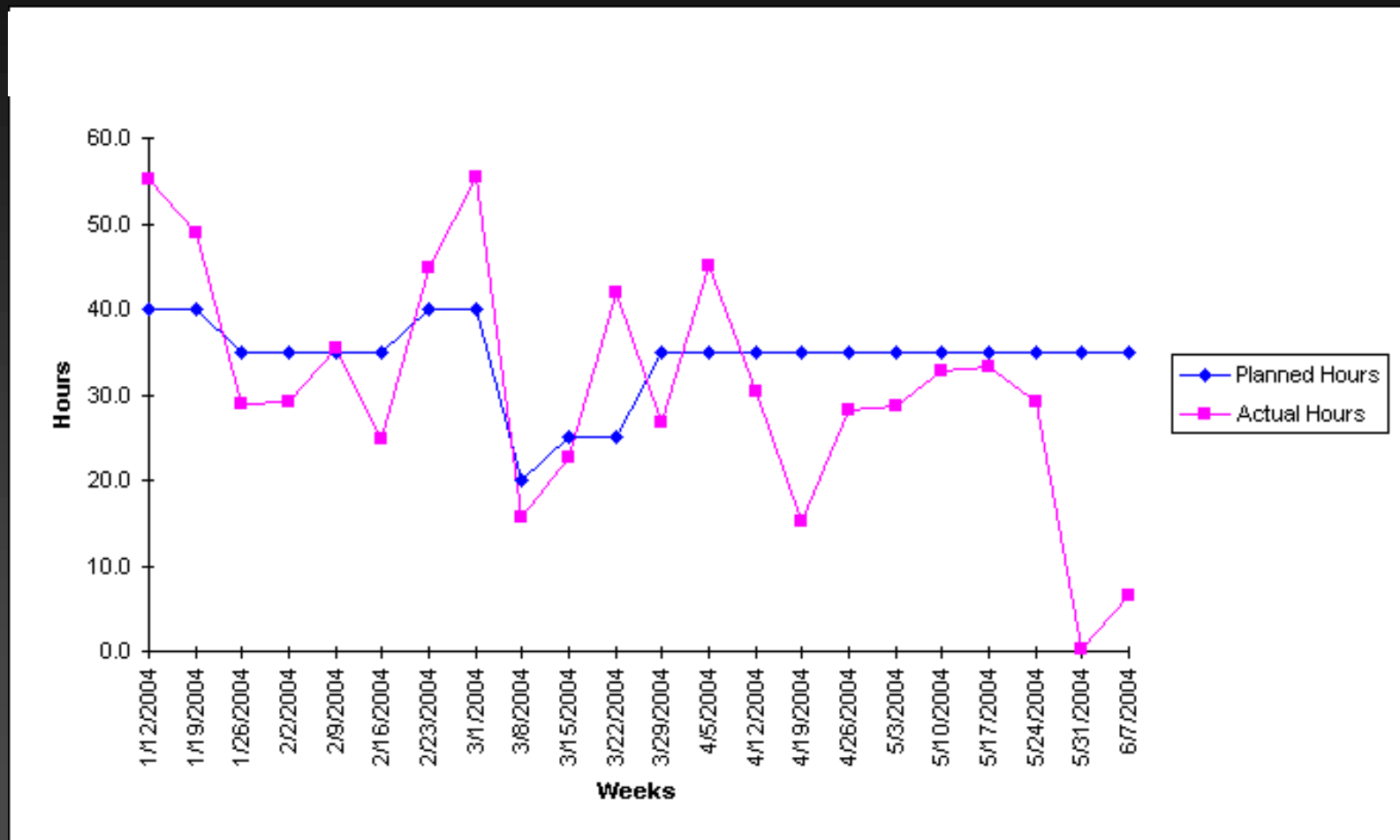
# Planning Earned Value per Week



# Planning Cumulative Hours

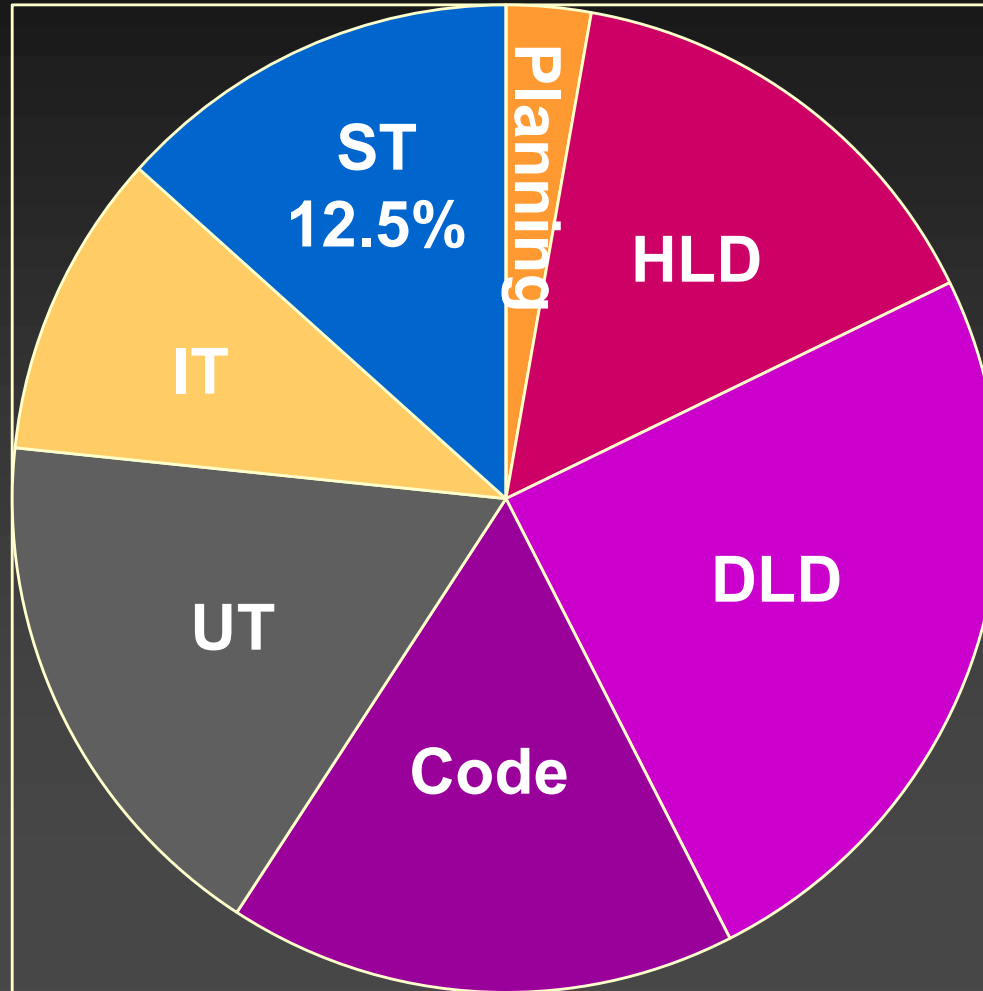


# Planning Hours per Week



# Planning Time in Phase

---



# PSP RESULTS

- Planning
- Productivity

	Plan		Actual		
	Size	Hours	Size	Hours	Rate
Docs	40	85	60	94	0.64/Hr.
Code	4106	605	5137	585	8.8/Hr.

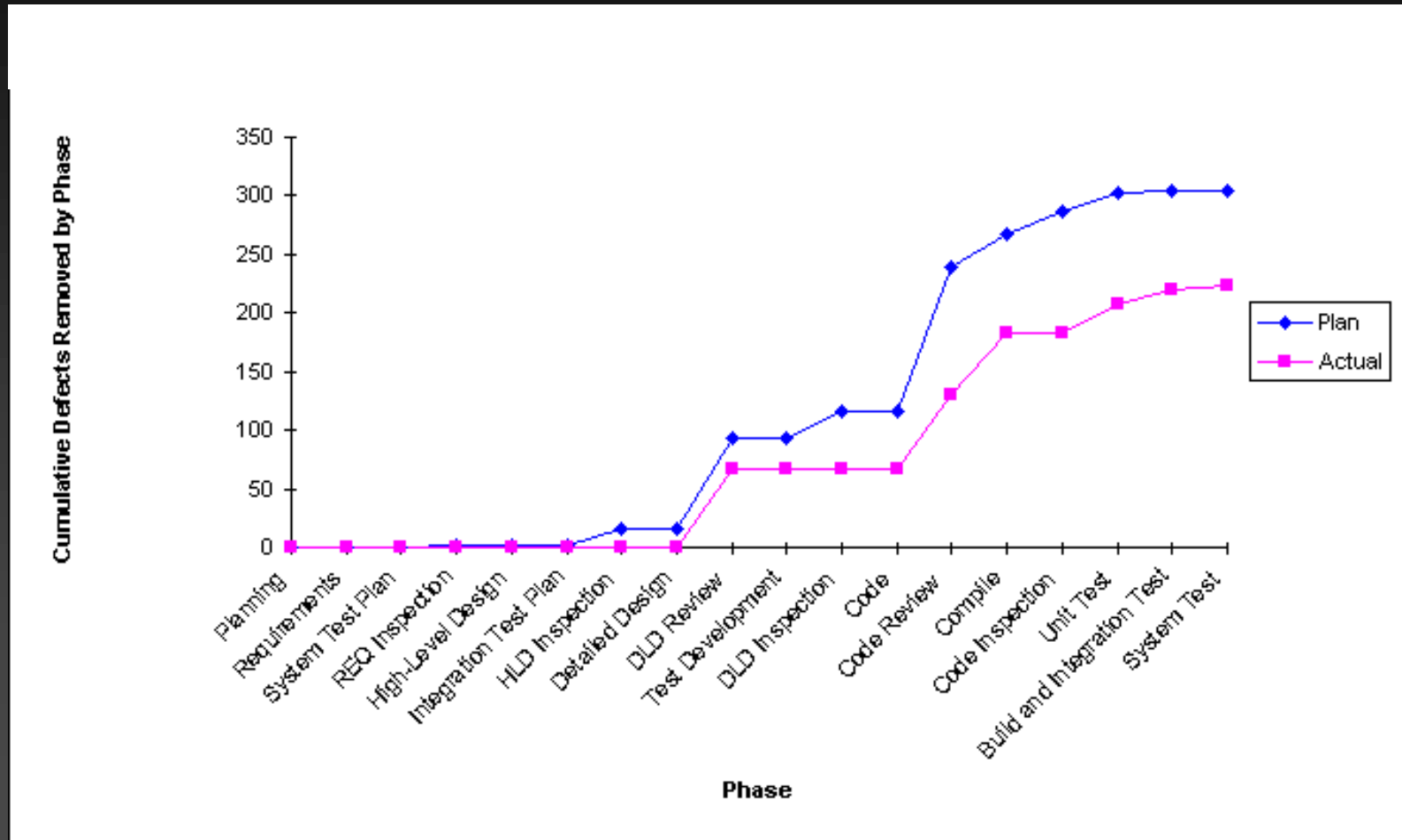
- Quality

# PSP RESULTS

---

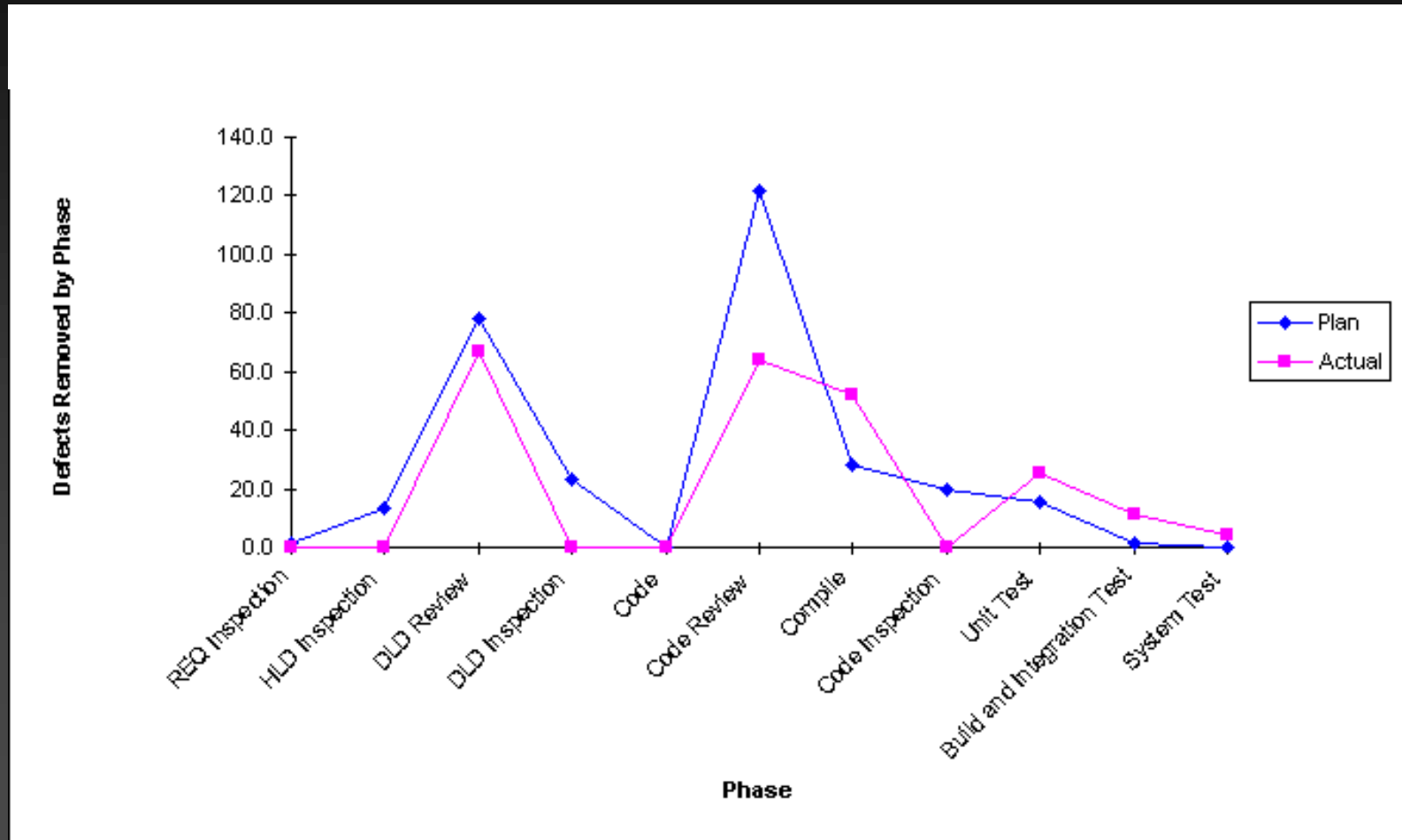
- Planning
- Productivity
- Quality
  - Cumulative Defects Removed by Phase
  - Defects Removed by Phase
  - Defect Density by Phase
  - Inspection & Review Rates
  - Quality Profile
  - Process Yield

# Quality Cumul. Defects Removed by Phase

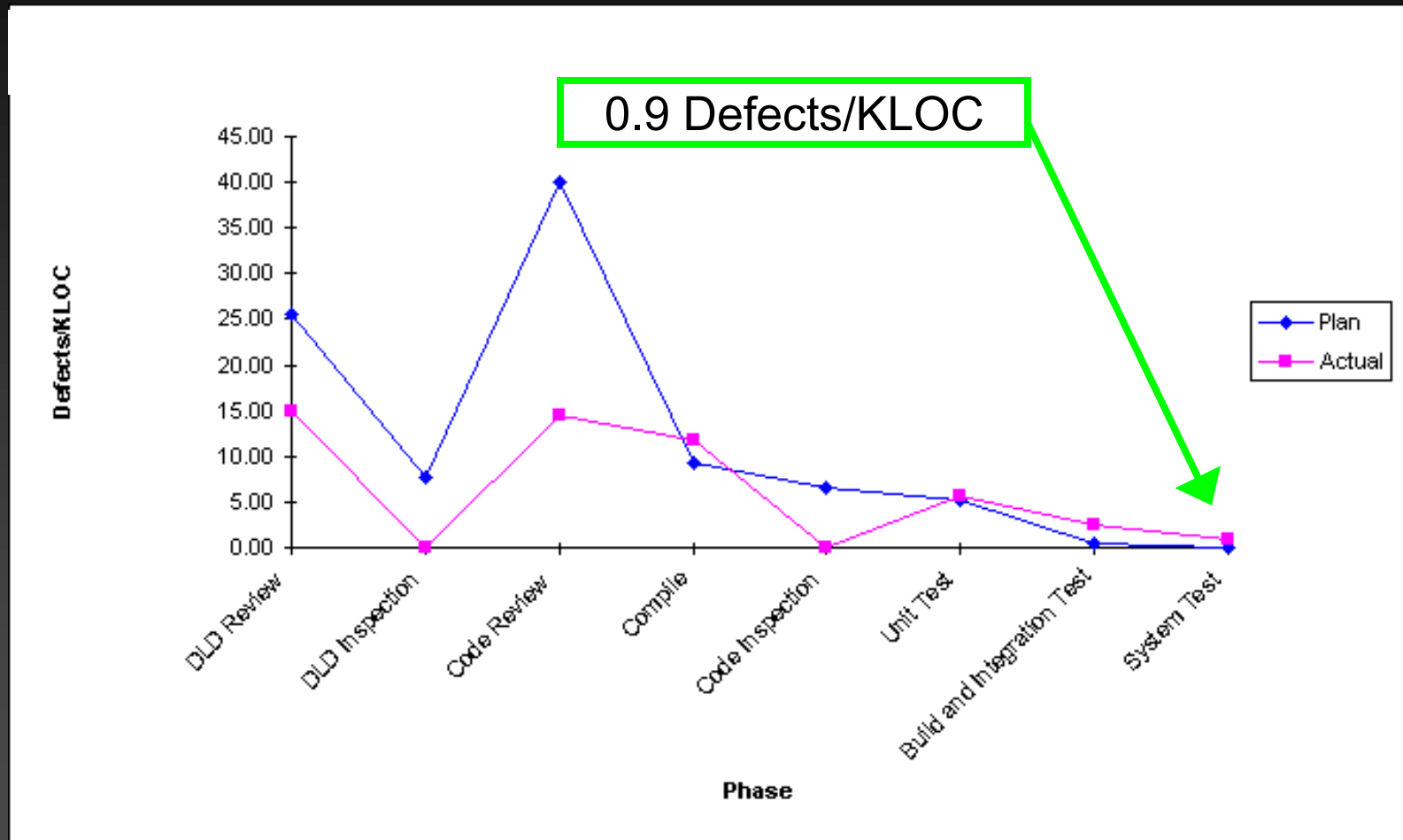




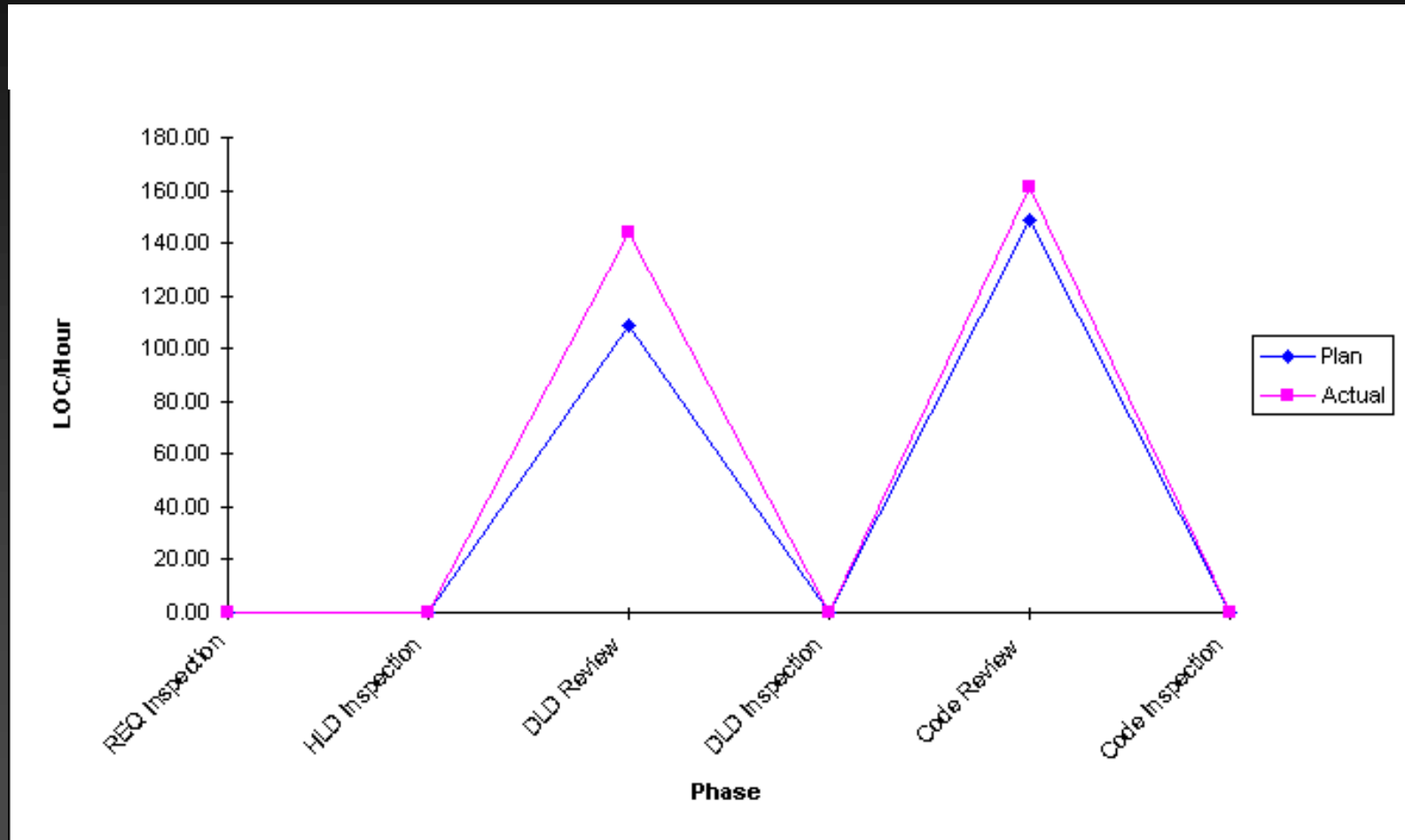
# Quality Defects Removed by Phase



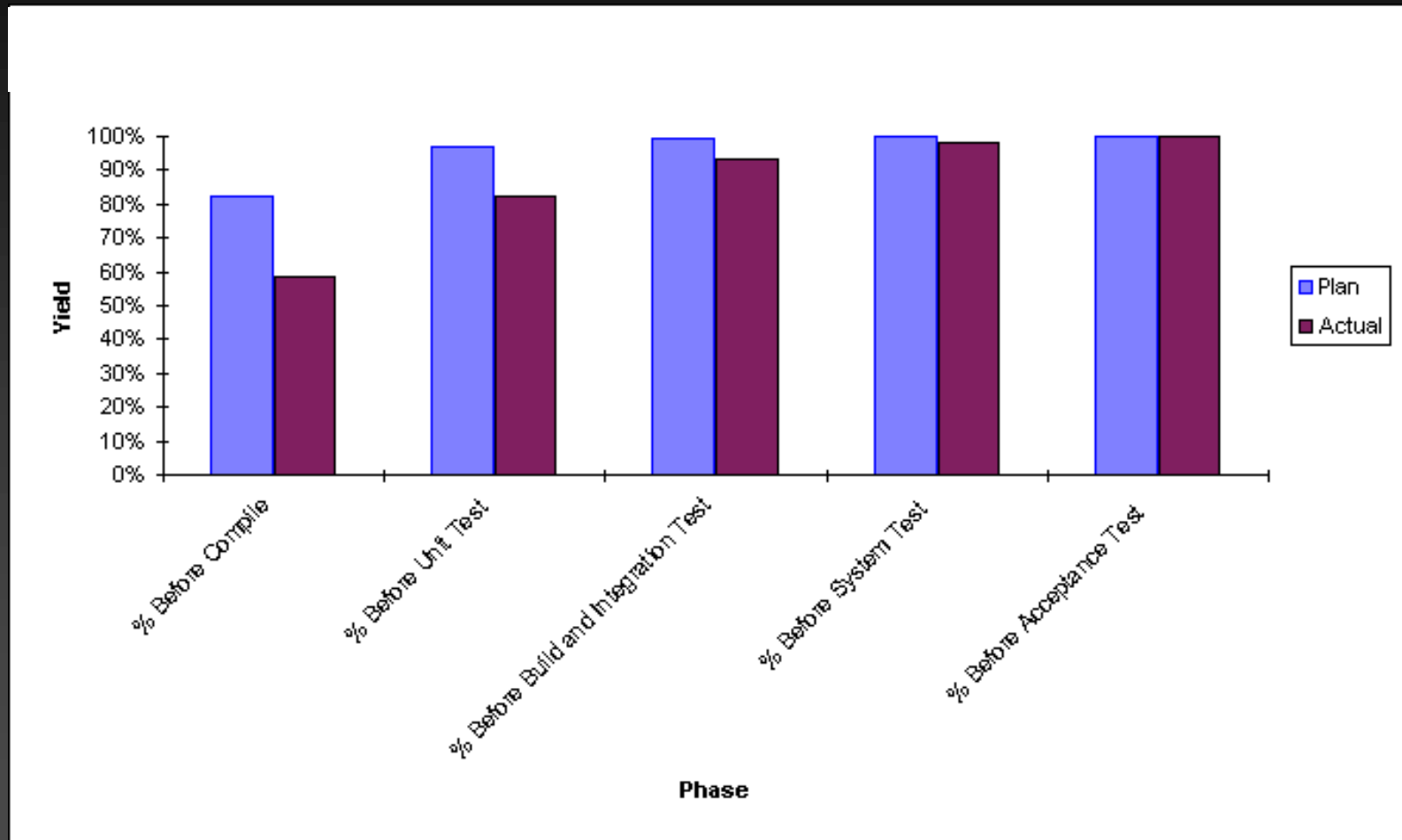
# Quality Defect Density by Phase



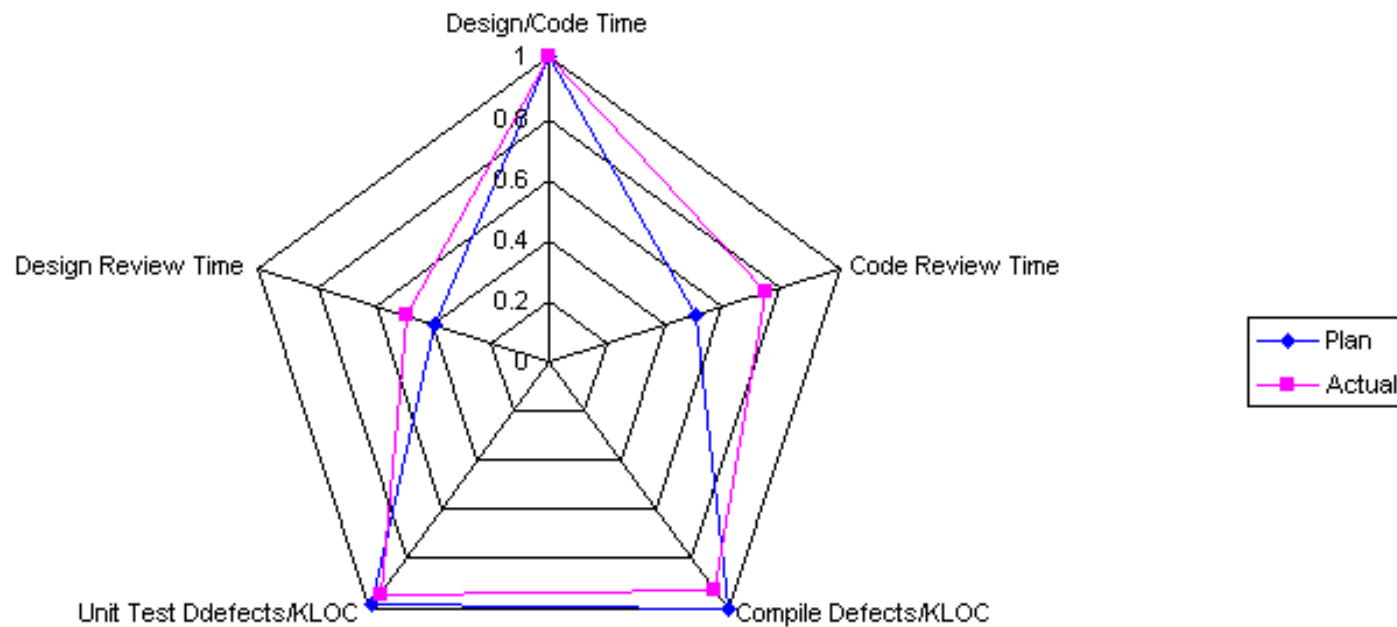
# Quality Inspection & Review Rates



# Quality Process Yield



# Quality Quality Profile



# SUMMARY

---

- Project Lead comments
- A Team of One
- Developer Challenges
- Further information

# Project Lead Comments

---

“Complex functions ... were often completely specified, written, reviewed, compiled and tested with virtually no defects found. The first time it happened, I was suspicious... 'just lucky' I thought. But by the second, third, and fourth times, I'd just resigned myself to the fact that Rob (and the SEI) had apparently figured out how to do it.”

Craig Surprise – Setra Systems

# A Team of One

---

## Act like a Professional Software Engineer

- Plan and manage your personal work
- Use effective personal methods
- Recognize your strengths and weaknesses
- Practice, practice, practice
- Learn from history
- Find and learn new methods

*A Self-Improvement Process for Software Engineers* by Watts Humphrey



# Developer Challenges

---

## Management support

- Negotiate commitments
  - Make a plan based on “your” performance history
  - Trade-offs: cost vs. schedule, reducing features
- Maintain project control
  - Follow and keep your plan updated
  - Manage change
- Deliver quality products
  - Low quality impacts cost, schedule and product value
  - High quality becomes your reputation

*A Self-Improvement Process for Software Engineers* by Watts Humphrey

# Further Information

---

- Read *Winning with Software - An Executive Strategy* by Watts S. Humphrey
- Read *PSP – A Self-Improvement Process for Software Engineers* by Watts S. Humphrey
- Attend the Software Engineering Process Group (SEPG<sup>SM</sup>) held each March by the SEI
- Investigate the PSP and TSP course offerings

# Contact Information

---

Rob Tonneberger  
Northern Horizons Inc.  
(603) 673-2808  
rtonneberger@northhorizons.com

This presentation & other information is available at:  
[www.northhorizons.com](http://www.northhorizons.com)



Northern Horizons Inc. is an authorized SEI Partner and is certified to teach PSP and TSP courses and the coaching of TSP projects. Our staff are trained directly at the Software Engineering Institute of Carnegie Mellon University, Pittsburgh, PA.

