SAND2006-3336P

A family of practices to deliver products on schedule, on budget, and high quality

May 24, 2006

Tom Cuyler
Team Coach
Sandia National Laboratories

Trademarks and Service Marks

The following are service marks of Carnegie Mellon University

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

The following are registered trademarks of Carnegie Mellon University

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

Science & Engineering Info Systems Group 4510

- Provide high-impact, value-added software and system solutions to Sandia's core mission programs
  - Nuclear Weapons (NW) SMU
  - Defense Systems and Assessments (DS&A) SMU
  - Homeland Security and Defense (HS&D) SMU
  - Integrated Enabling Services (IES) SMU
- Quality emphasis
  - Enterprise Architecture
  - Capability Maturity Model (CMM) Level 3
  - Target of CMMi Level 4
  - High rigor for classified systems as needed

Science & Engineering Info Systems Group 4510

- Core Competencies include:
  - Enterprise Architecture (includes Data & S/W architectures)
  - Production software solutions
    - Used across the Nuclear Weapons Complex
    - Need To Know access component w/API
    - High complexity systems (modeling & simulation, waveform analyses, security, reusable components)
  - Data Services
    - High rigor data warehousing (integration)
    - Data analyses, decision support systems
    - Data mining, pattern recognition, natural language processing

Science & Engineering Info Systems Group 4510

- **Experience with Applications Development**
  - **Currently developing new systems:**
    - **Nuclear Weapons (NW) Stockpile Surveillance**
  - **Releasing new versions**
    - **Classified NW Portal**
    - **NW Record Of Assembly**
    - **Materials Properties Analyses software**
    - **Stockpile Dismantlement software**
    - **Other classified systems**

The Goals of the PSP^{sm} and TSP^{sm}

The principal objective of the PSP and TSP is to make ***quality work*** normal and natural.

To do this, developers must

- Plan and track their personal work
- Measure and manage the quality of their products
- Follow their defined processes

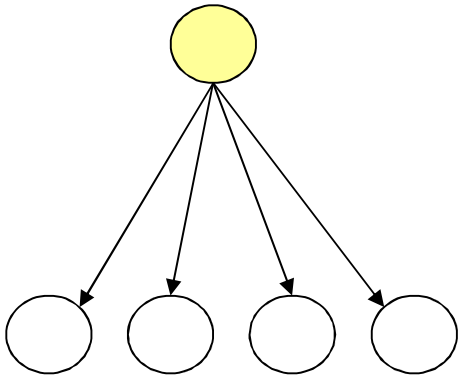
This, in turn, requires that the developers

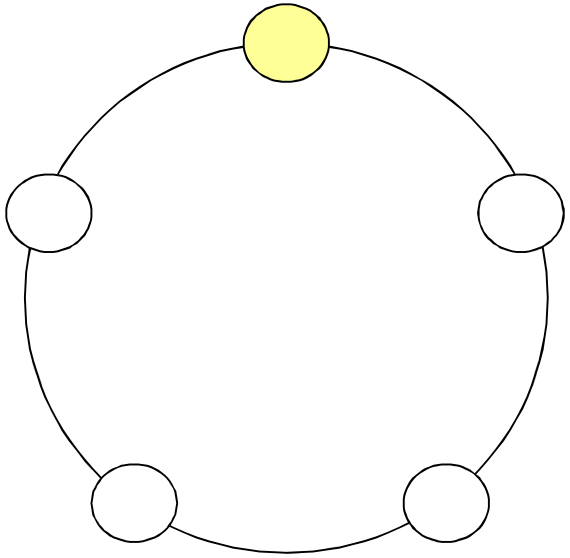
- "own" their own processes
- Believe in and be committed to their plans
- Strive to produce quality products

Successful Team Practices

- **When development teams work well, they generally have successful projects.**
- **To be successful, teams need**
  - clear goals
  - established roles
  - defined processes
  - agreed-upon plans
- **Teams also must be motivated and personally committed.**
- **Without these, teams will rarely be successful.**

Self-directed Teams





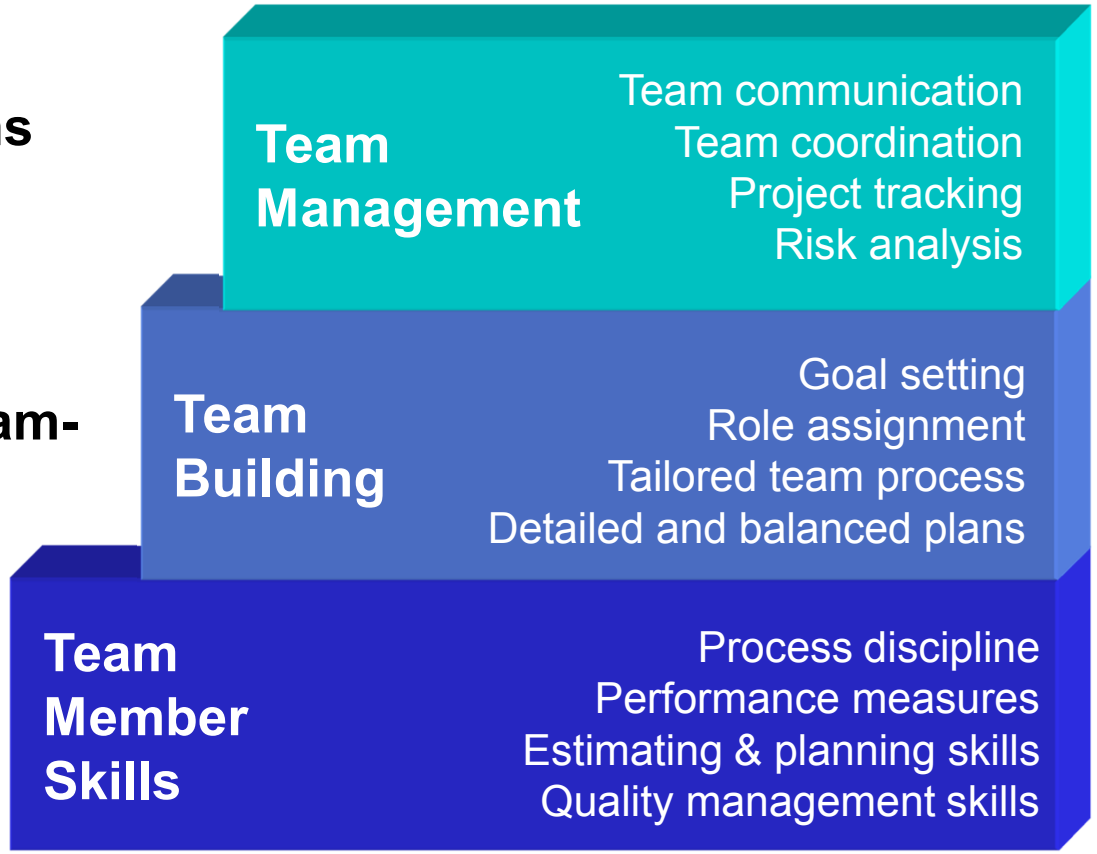
Traditional team
The leader plans, directs, and tracks the team's work.

Self-directed team
The team members participate in planning, managing, and tracking their own work.

Building Self-directed Teams

Self-directed teams must be built.

This requires a team-building process.



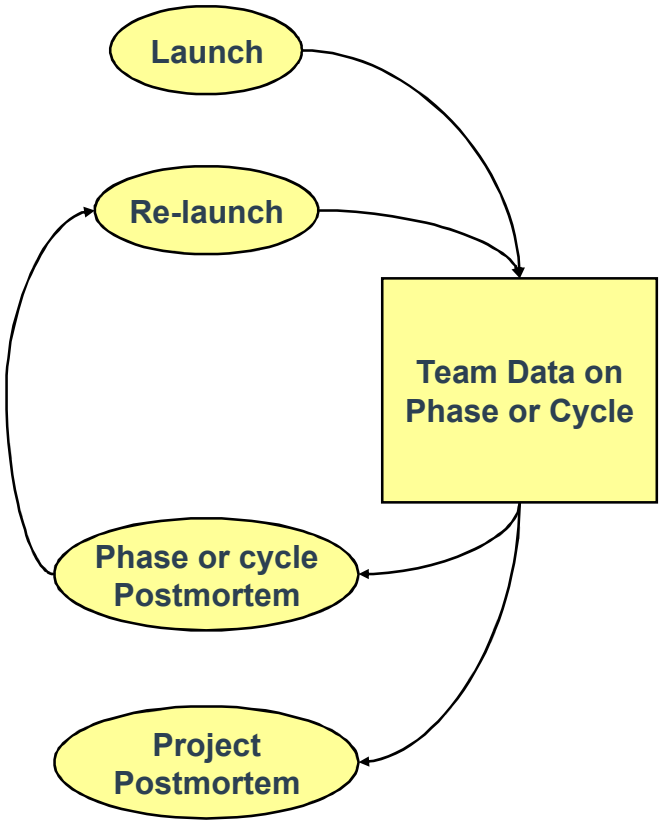
TSP Planning

TSP projects can start on any phase or cycle.

Each phase or cycle starts with a launch or re-launch.

The strategy is to
- develop in increments
- use multiple cycles
- work-ahead

TSP permits whatever process structure makes the most business and technical sense.



The TSP^{sm} Launch/Relaunch

The Launch

- The purpose of the launch is to produce a ***Team*** plan and get management agreement to that plan
- The principal objective of the launch is to build a team that is ready to jell
- The launch builds the team's confidence in their ability to do quality work

The TSP^{sm} Launch/Relaunch

In the four-day TSP team-building process, teams

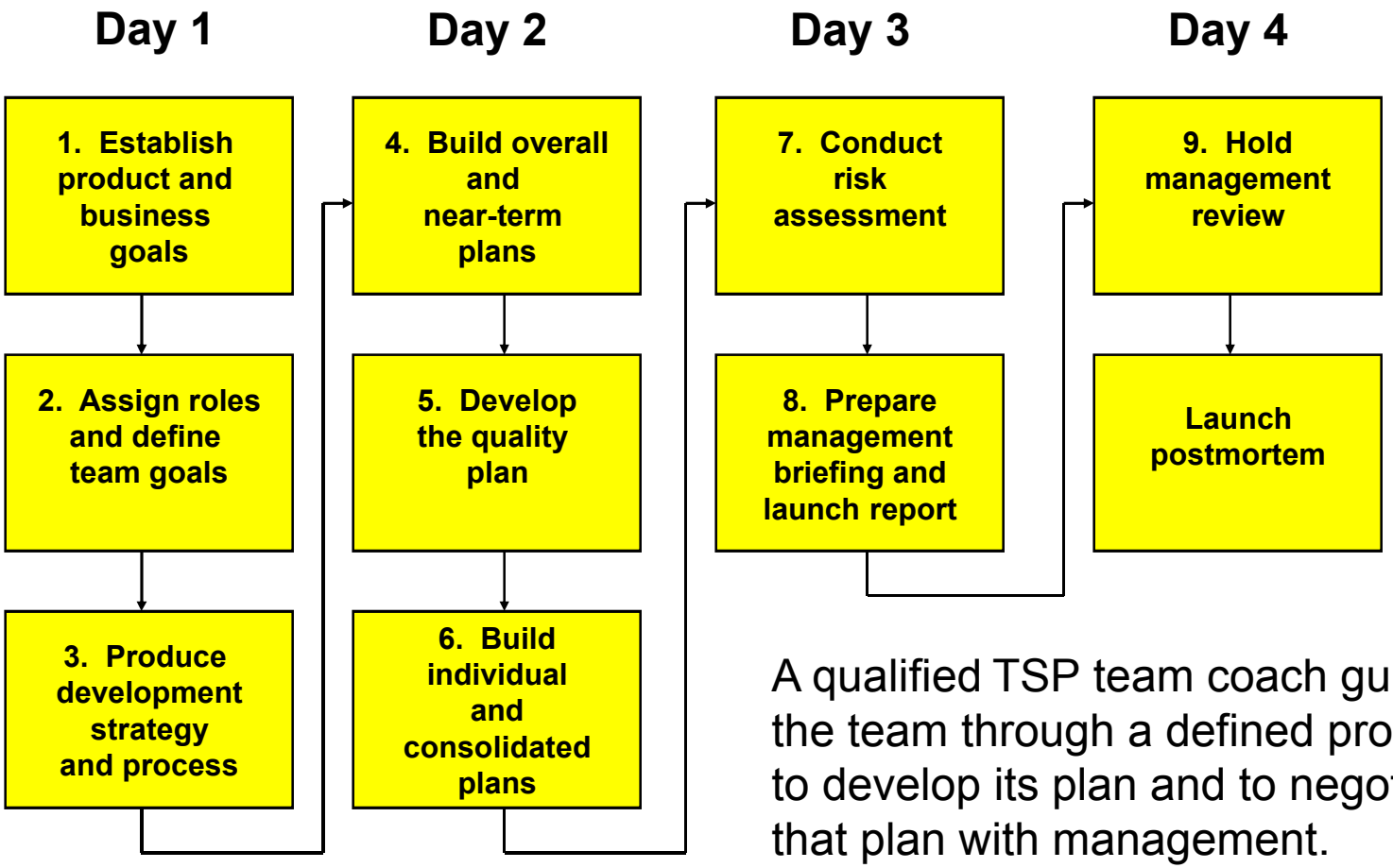
- establish their own goals and roles
- define their own strategies and processes
- produce their product development plans
- defend their plans to management

TSP teams

- own their own processes and plans
- understand why their goals are important
- have voluntarily committed to their goals
- believe that their commitments are achievable

All team members participate in the launch.

The TSP Launch Process



A qualified TSP team coach guides the team through a defined process to develop its plan and to negotiate that plan with management.

TSP^{sm} Metrics: Managing with Data

TSP^{sm} teams **use data** to help manage the project. The four basic metrics:

- Schedule
- Time
- Size
- Defects

From these the team **tracks**:

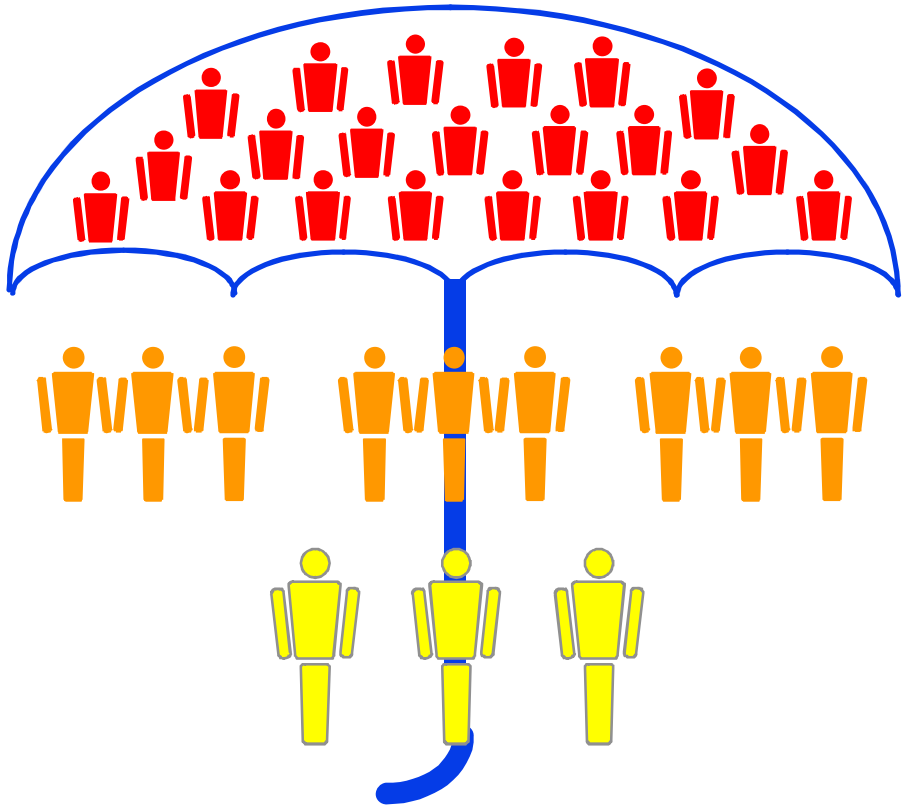
- Earned value
- Defect removal
- Component quality profile.

Vertically Aligned Capabilities: TSP and CMMI

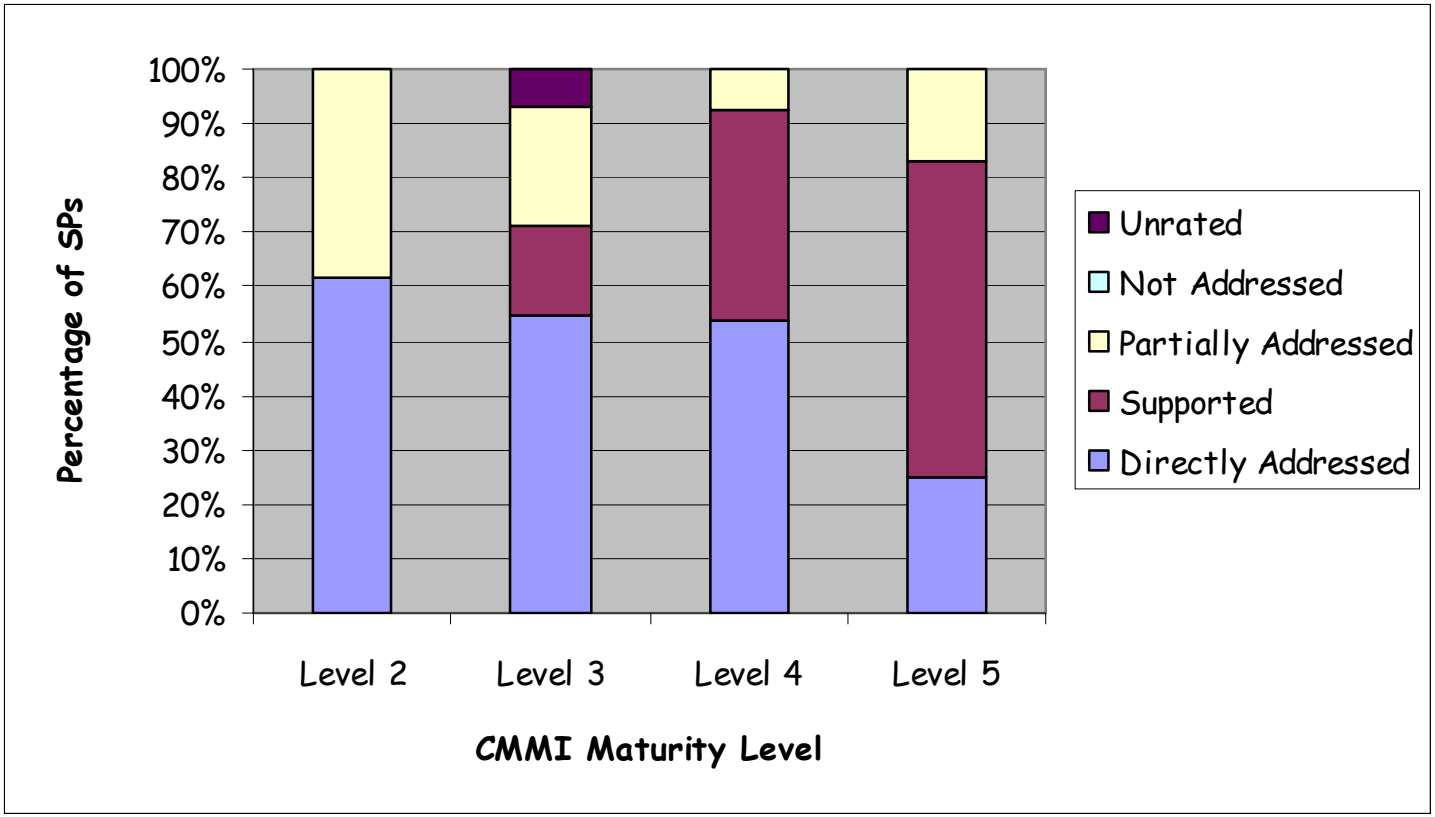
CMM/CMMI: ***for organizational capability***

TSP - ***for quality products on cost and schedule***

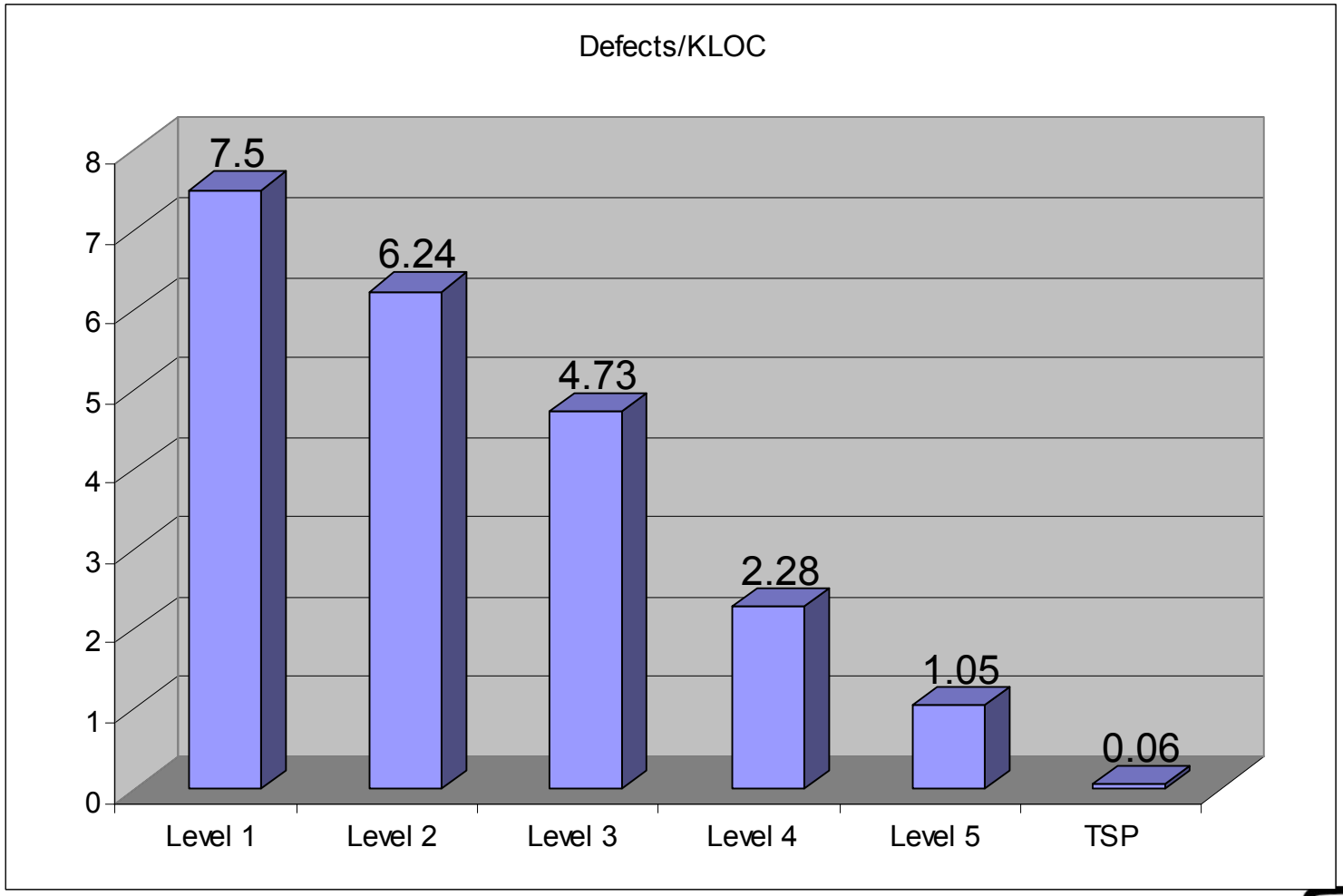
PSP - ***for individual skill and discipline***



TSP-CMMI by Maturity Level



Process Improvement Pays Off

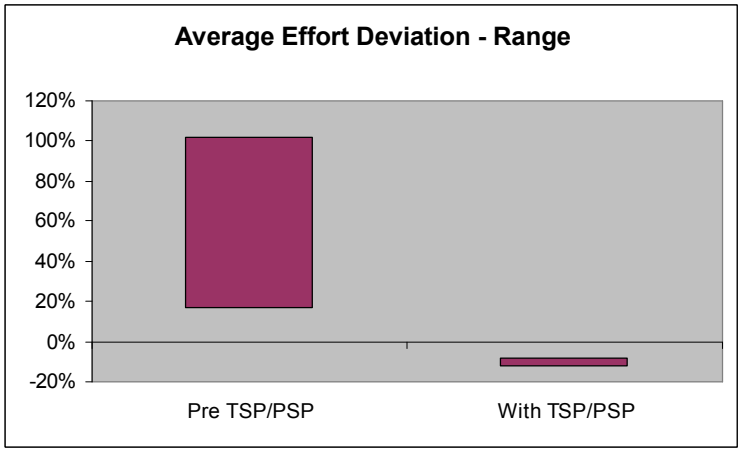


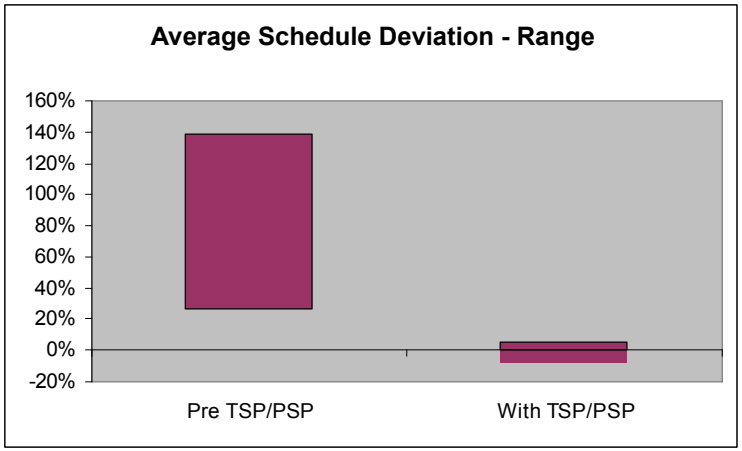
TSP Performance Summary

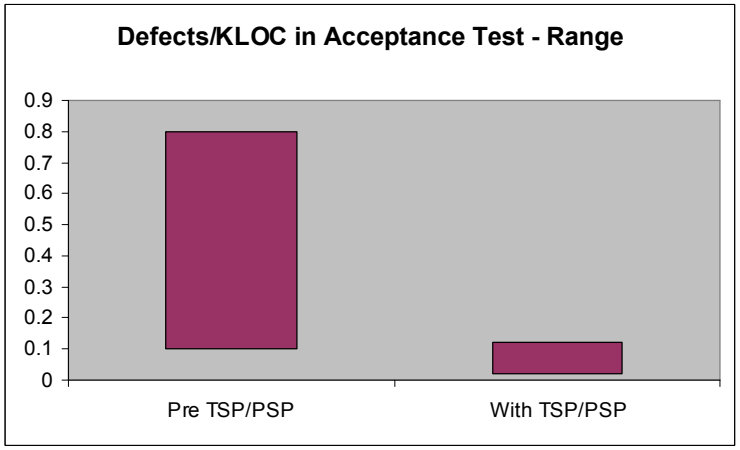
Performance Category	TSP Impact Study (2003)*	Typical Industry Performance
System test defects per thousand instructions	**0.4 avg. 0.0 to 0.9**	**2 to 14**
Released defects per thousand instructions	**0.06 avg. 0.0 to 0.2**	**1 to 7**
System test effort (% of total effort)	**4% avg. 2% to 7%**	**40%**

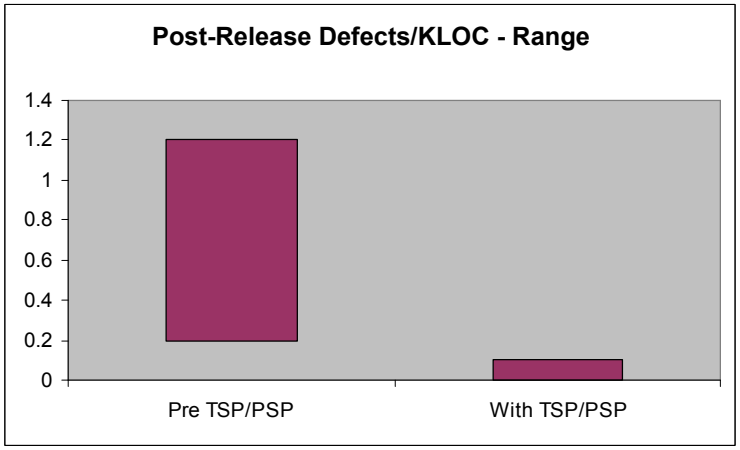
*From a study of 20 projects in 13 organizations

TSP Performance Comparison*









*From a study of 18 projects in 4 organizations

What is Quality Management?

- Most would reply Strong Testing?
- Achieving Quality Goals
  - Gathering data
  - Monitoring process quality
- Quality Management Methods

What is Quality Management?

- With software
  - every team member's work is important
  - any poor-quality module can crash the system
  - only a few defective modules will delay entire programs
- Testing finds only a fraction of a product's defects.
- The number of defects in a delivered product is proportional to the number found in development testing.
- To consistently produce quality products, the defects must be removed before testing

What is Quality Software?

- High-quality products must meet all user requirements.
- Quality software must be essentially free of defects.
- A defect is something in a program that is incorrect and must be fixed.
- Defects result in
  - incorrect, insecure, or unsafe operation
  - installation problems
  - confusing or incorrect documentation
  - error-prone modification and enhancement

Quality Products

- High-quality processes produce high-quality products.
- The cornerstone of a high-quality software process is early defect removal.
- Quality work saves time and money.
- Quality work is not done by accident; it requires discipline, commitment, management, and measurement.
- TSP shows teams how to efficiently remove defects at the earliest possible point in the process.

Importance of High Quality

- To produce high quality systems, every part must be of high quality.
- This is possible only if every engineer consistently follows a quality process.
- To consistently follow a quality process, engineers must
  - be properly trained (with the PSP)
  - work on a disciplined team (with the TSP)
  - have coaching support and guidance

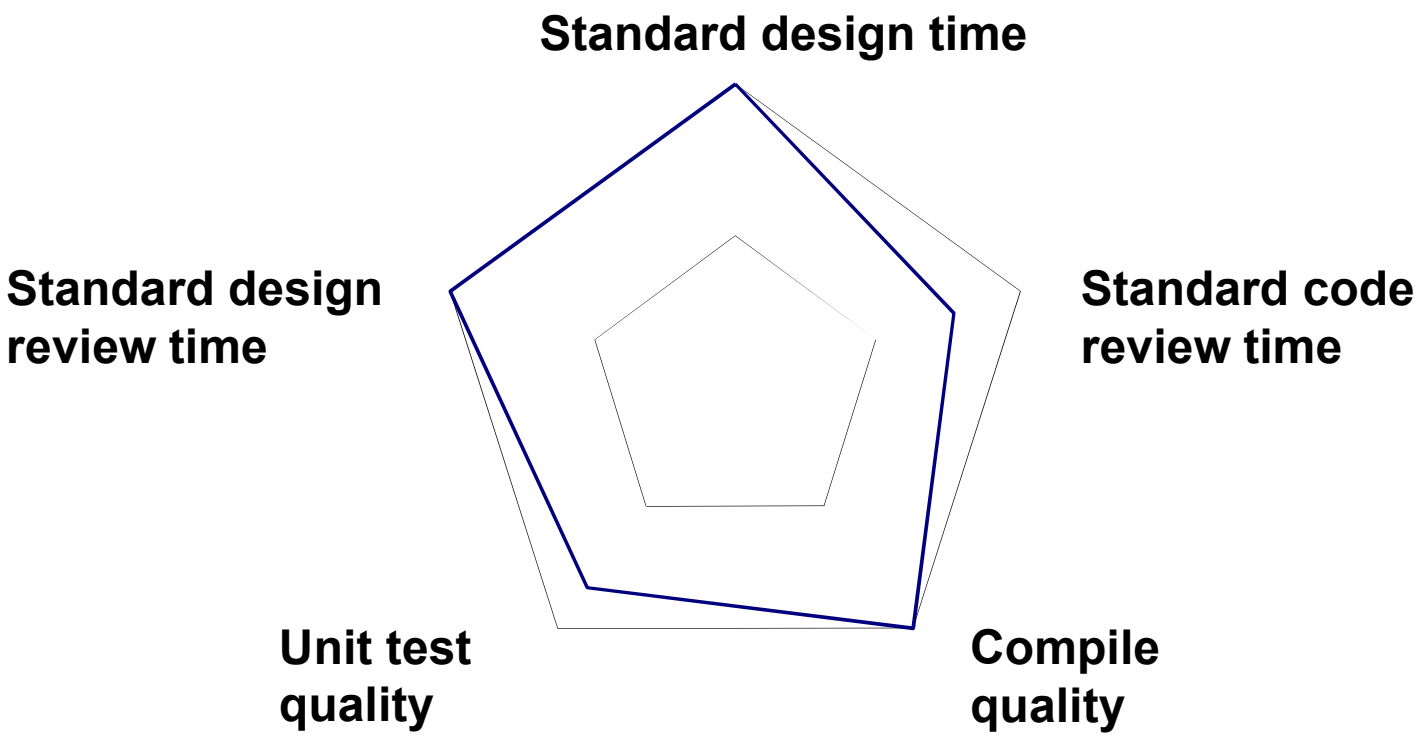
Coaching and Guiding Quality Work

- The first step is to ensure that the planned process is likely to produce a high-quality product.
  - inspections
  - reviews
  - tests
- Next, the team must examine the process data as the work is being done.
  - Adjust the process if it is not adequate.
  - Correct the product elements that were produced with the faulty process.
- Then, continue reassessing and correcting.

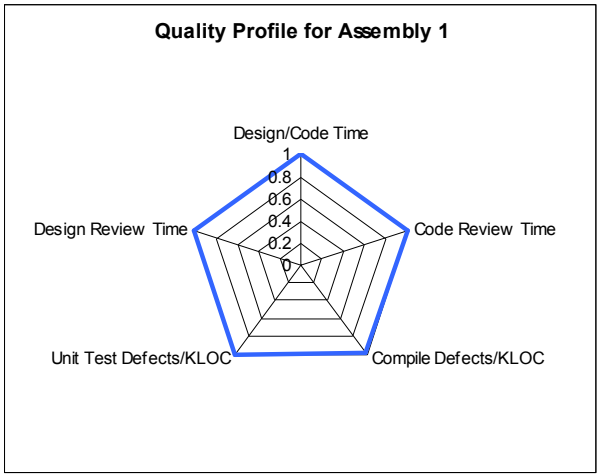
Gathering Data to assess Quality

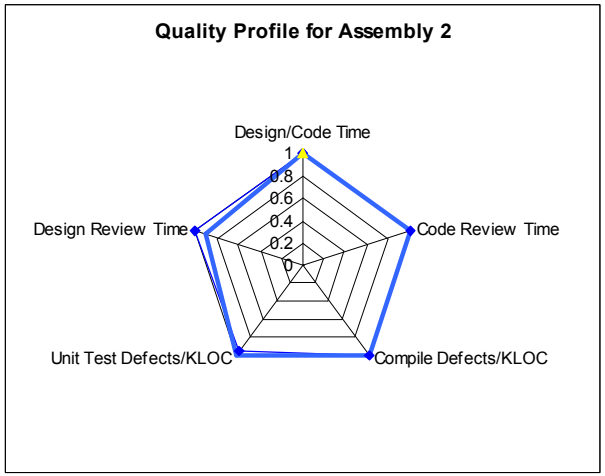
- Size, time and defect data are used to track progress towards team quality goals for
  - individual yield
  - component defect densities
  - defect rate for the system
- Individuals collect their own defect data at the completion of
  - design reviews
  - code reviews
  - compiles
  - unit tests
- Teams collect defect data at the completion of
  - design inspection
  - code inspection
  - integration test
  - system test

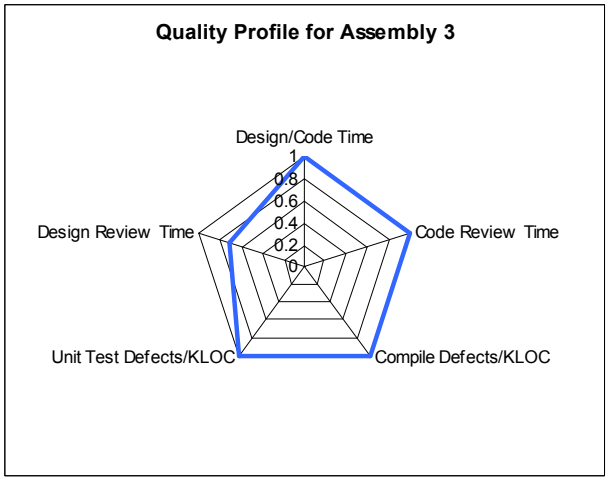
PQI Defined (or Component quality profile)

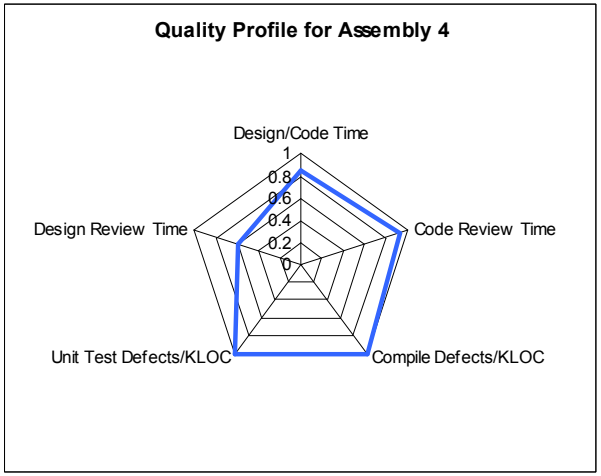


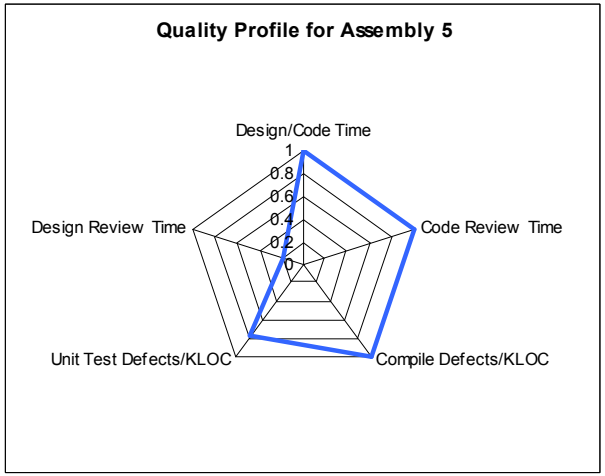
Selected TSP Quality Profiles

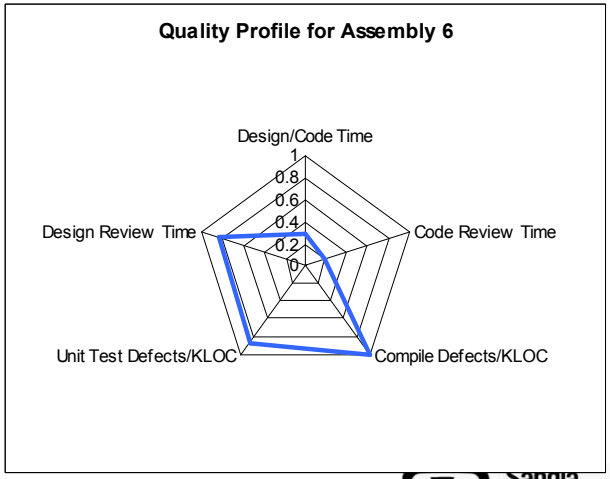












Using Quality Profiles

- The combination of the five profile dimensions provides a reasonably comprehensive view of the development process.
- When the quality profile indicates a high-quality program, that program is unlikely to have remaining defects.
- By examining the quality profiles for many programs, teams can quickly spot programs that are most likely to have quality problems.
- From the quality profile, teams can see which process area was most troublesome and presents an opportunity for improvement.
- The profiles should be used only as guides.

Test Data Analysis

- Test defect should be used to assess upstream processes
- Look at test data and try to identify the most troublesome components.
  - The number of defects found in any test is the best indicator of the number of defects likely to remain after that test.
  - Test yields can be 40-60%, so roughly as many defects remain as were found.
  - If the engineer follows a good process, and defect density in unit test are under 5 def/KLOC, test yields can be 70-90%.

- Have the team reinspect and fix the components that are likely to have remaining defects, as time allows. This will
  - produce higher-quality products
  - reduce integration and test time

Interpreting Quality Data

- To maintain a consistent focus on quality, it is important to regularly analyze and report defect data.
  - need to be looked at in context
  - provide an indication of good and bad quality components
  - provide hints on what may be done to reduce the risk of poor quality products
- Data can be examined for the system or at the individual level.
  - Team-level system data may be reported to management.
  - Team data may indicate troublesome phases or processes.
  - Individual data may indicate components with specific problems.
- Be careful how data are handled.
- As data begin to stabilize, consider statistical analysis.

TSP^{sm} Success Factors

- Clear *Sustained* Management Support
- Focus is on Process and Product Quality improvement
- Manage quality with data and strive to remove defect early in the process (Injected, removed, cause)
- Reduce test time by striving for no test defects
- Manage and make project improvements during the project
- Leverage data to make overall process improvements for the next project

Any Questions?