Carnegie Mellon
Software Engineering Institute

Pittsburgh, PA 15213-3890

Personal Software Process SM for Engineers: Part I

Introduction to PSP and TSP

Sponsored by the U.S. Department of Defense

Lecture Topics

The need for change

PSP[SM] and TSP[SM] principles and objectives

What is the TSP?

The need for management support

What is the PSP and how does it help?

Course results

[SM] Personal Software Process, PSP, Team Software Process, and TSP are service marks of Carnegie Mellon University.

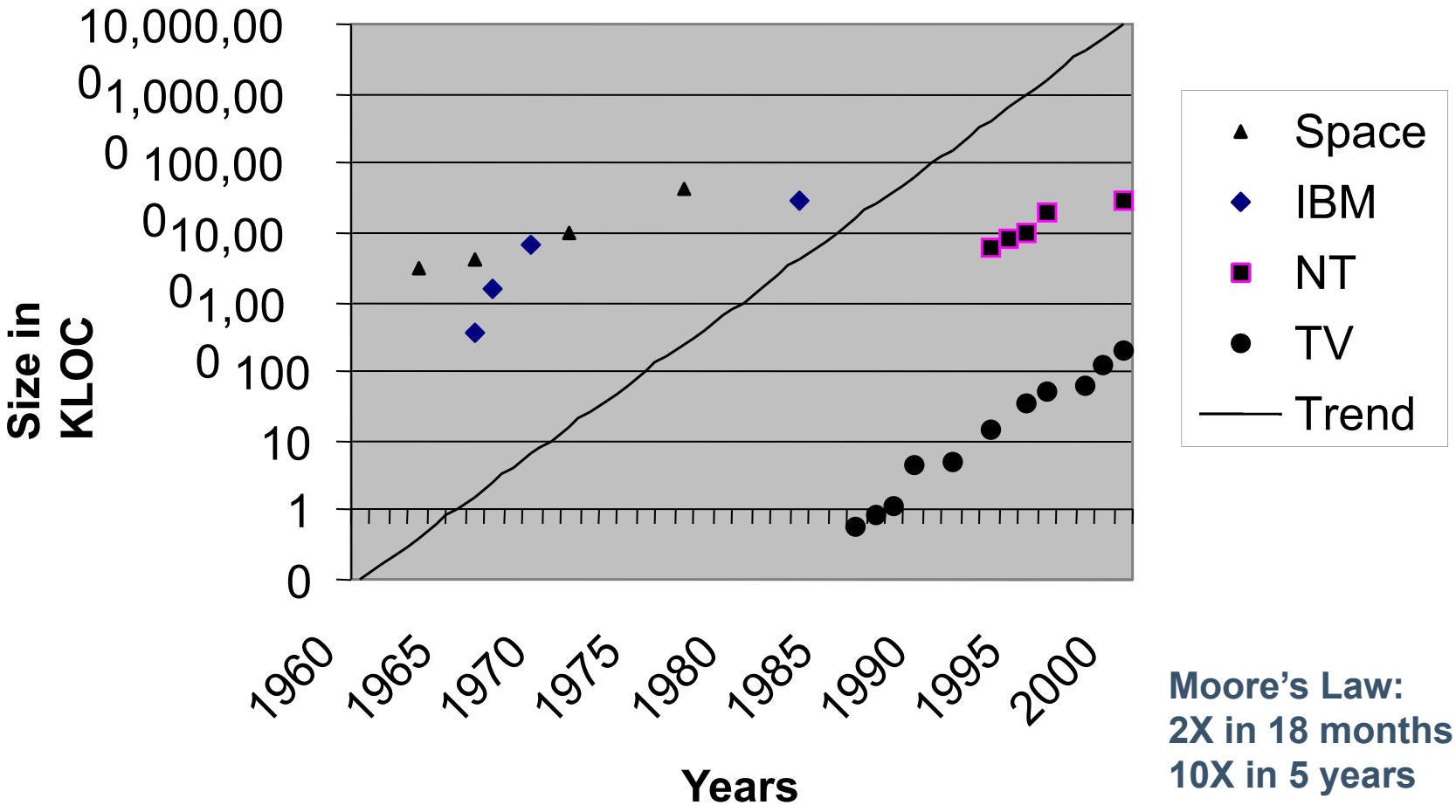
The Changing World of Software

Software now controls most business, government, and military systems.

- Factories are managed by software.
- Most advanced products are controlled by software.
- Finance, administrative, and business operations are largely run by software.

The cost, schedule, and quality of software is now a critical business concern.

Software Products are Bigger



Moore's Law:
2X in 18 months
10X in 5 years

Big Software Projects Usually Fail

With increased size, projects are more troubled.

Project Size	People	Time (Months)	Success Rate
Less than $750K	6	6	55%
$750K to $1.5M	12	9	33%
$1.5M to $3M	25	12	25%
$3M to $6M	40	18	15%
$6M to $10M	+250	+24	8%
Over $10M	+500	+36	0%

This is a problem of scale: current software practices do not scale up.

Standish: Chaos Reports, 1999

Why Projects Fail -1

Large and small software projects fail for four reasons.

Project commitments are often unrealistic.

- The larger the project, the less influence we have.
- If we don't have anything to say, nobody will listen.

Larger projects are harder to control.

- Today, few developers have personal plans.
- Without a plan, you cannot know job status.
- If you don't know where you are, management can't understand job status.
- If management doesn't understand job status, they can't manage projects.

Why Projects Fail -2

Quality problems get worse with project size.

- In software systems, if any part has quality problems, the system will have quality problems.
- If the developers do not manage quality, their teams cannot manage quality.
- When unmanaged, quality will always be poor.

To be effective, teams need leadership and coaching.

- Leaders build team motivation and commitment.
- Coaching develops team cohesion.
- Cohesive, motivated, and committed teams do the best work.

The Need for Change

Many lives and businesses now depend on software.

We now need larger, more complex, and safer software systems on predictable schedules.

Without different software practices, this will not happen.

The Team Software Process (TSP) addresses this need.

The PSP provides the knowledge and skill that developers need to work on TSP teams.

Management Support -1

The initial TSP objective is to convince management to let your team be self directed.

A self-directed team

- sets its own goals
- establishes its own roles
- decides on its own development strategy
- defines its own processes
- develops its own plans
- measures, manages, and controls its own work

Self-directed teams do the best work.

Management Support -2

Management will support you as long as you

- strive to meet their needs
- provide regular reports on your work
- convince them that your plans are sound
- do quality work
- respond to changing needs
- come to them for help when you have problems

Management Support -3

Self-directed teams are a bargain.

Management will agree to your managing your own work as long as they believe that you are doing a superior job.

To convince them of this, you must

- maintain precise and accurate plans
- measure and track your work
- regularly show management that you are doing superior work

The PSP shows you how to do this.

PSP Principles -1

The quality of a software system is determined by the quality of its worst components.

The quality of a software component is governed by the individual who developed it.

The quality of a software component is governed by the quality of the process used to develop it.

The key to quality is the individual developer's skill, commitment, and personal process discipline.

PSP Principles -2

As a software professional, you are responsible for your personal process.

You should measure, track, and analyze your work.

You should learn from your performance variations.

You should incorporate lessons learned into your personal practices.

What Does a PSP Provide?

A stable, mature PSP allows you to

- estimate and plan your work
- meet your commitments
- resist unreasonable commitment pressures

You will also

- understand your current performance
- be better equipped to improve your capability

What Does the PSP Provide?

The PSP provides

- a proven basis for developing and using an industrial-strength personal process
- a discipline that shows you how to improve your personal process
- the data to continually improve the productivity, quality, and predictability of your work

What is the PSP?

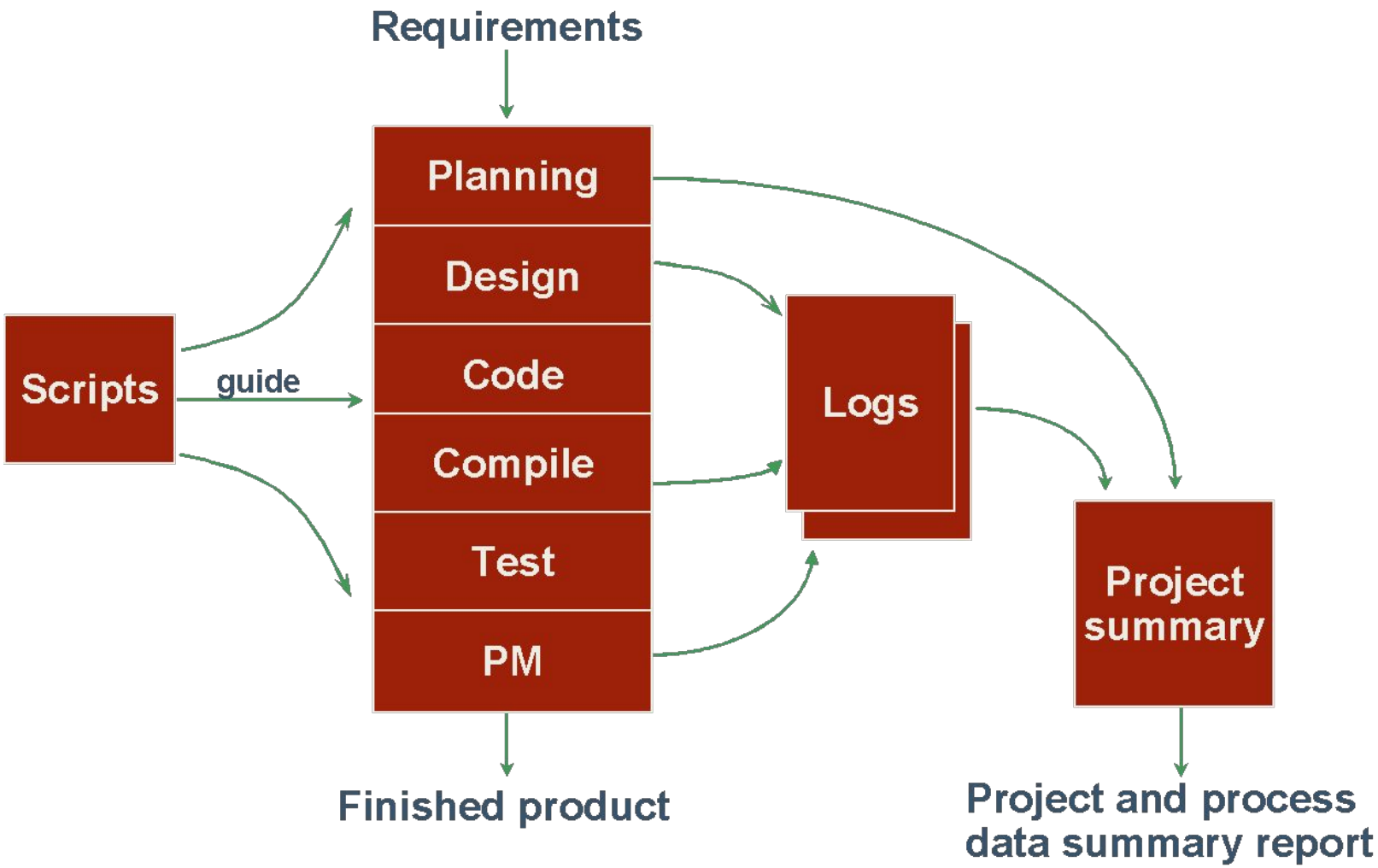
The PSP is a personal process for developing software or for doing any other defined activity. The PSP includes

- defined steps
- forms
- standards

It provides a measurement and analysis framework for characterizing and managing your personal work.

It is also a defined procedure that helps you to improve your personal performance.

The PSP Process Flow



The Personal Software Process

The PSP process is designed for individual use.

It is based on scaled-down industrial software practice.

The PSP course demonstrates the value of using a defined and measured process.

It helps you and your organization meet the increasing demands for high quality and timely software.

Learning the PSP -1

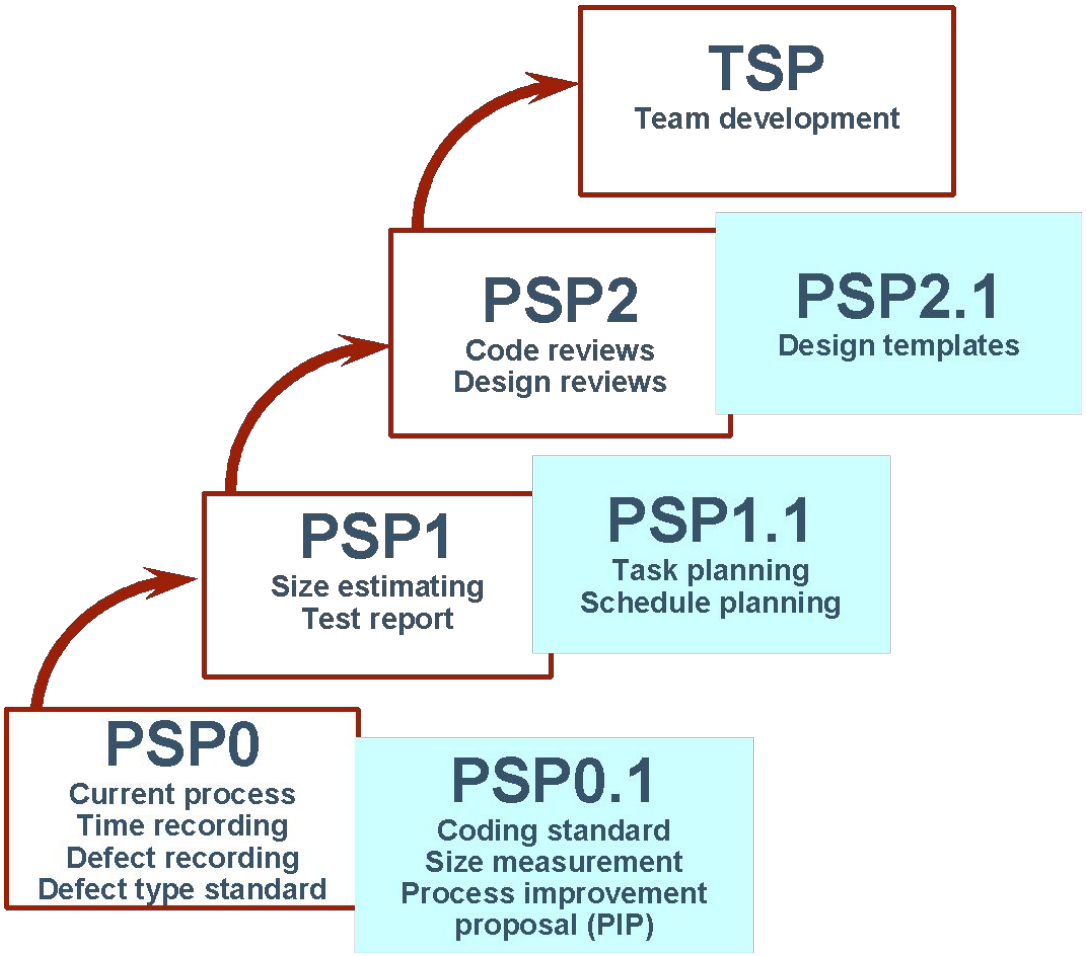
The PSP is introduced in six upward-compatible steps.

You write one or more module-sized programs at each step.

You gather and analyze data on your work.

You use the results to improve your personal performance.

Learning the PSP -2



Learning the PSP -3

PSP0: You establish a measured performance baseline.

PSP1: You make size, resource, and schedule plans.

PSP2: You practice defect and yield management.

At Course Conclusion

You will have practiced the key elements of an industrial-strength software process.*

You will understand which methods are most effective for you.

You will do better work.

You will have long-term improvement goals.

*These are generally called CMMI level 5 processes.

Course Results

We now have data on over 30,000 programs written using the PSP.

The following charts show how others have improved during the PSP course.

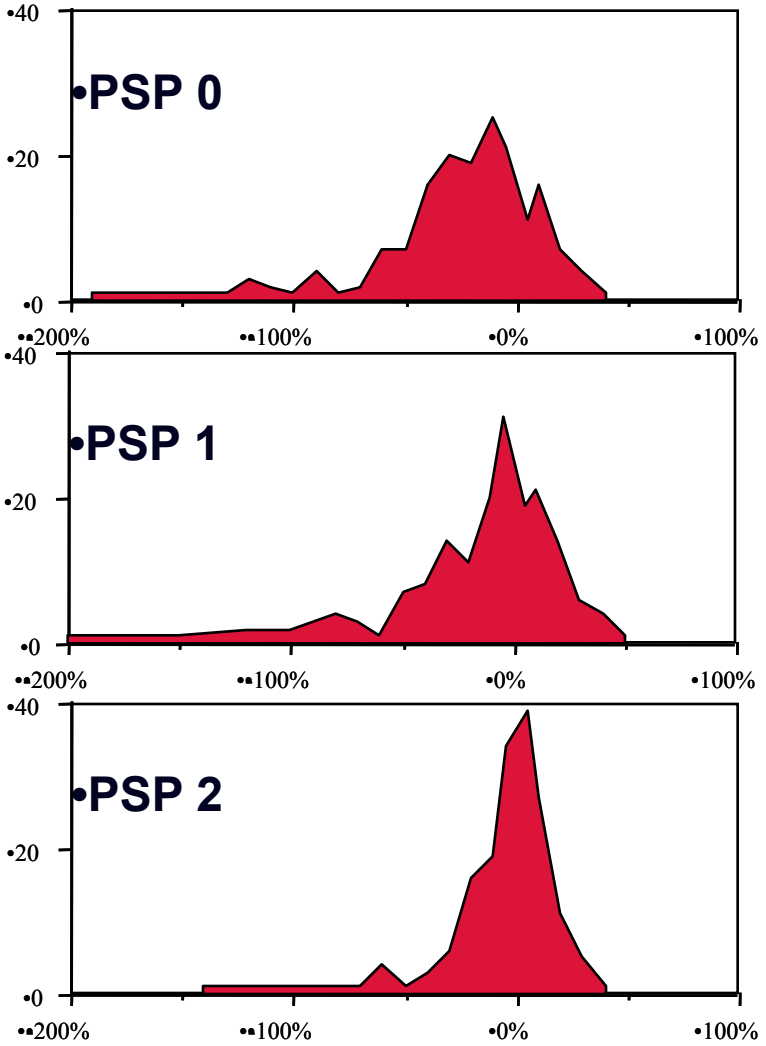
- effort estimating
- compile and test time
- productivity

PSP Effort Estimating Accuracy

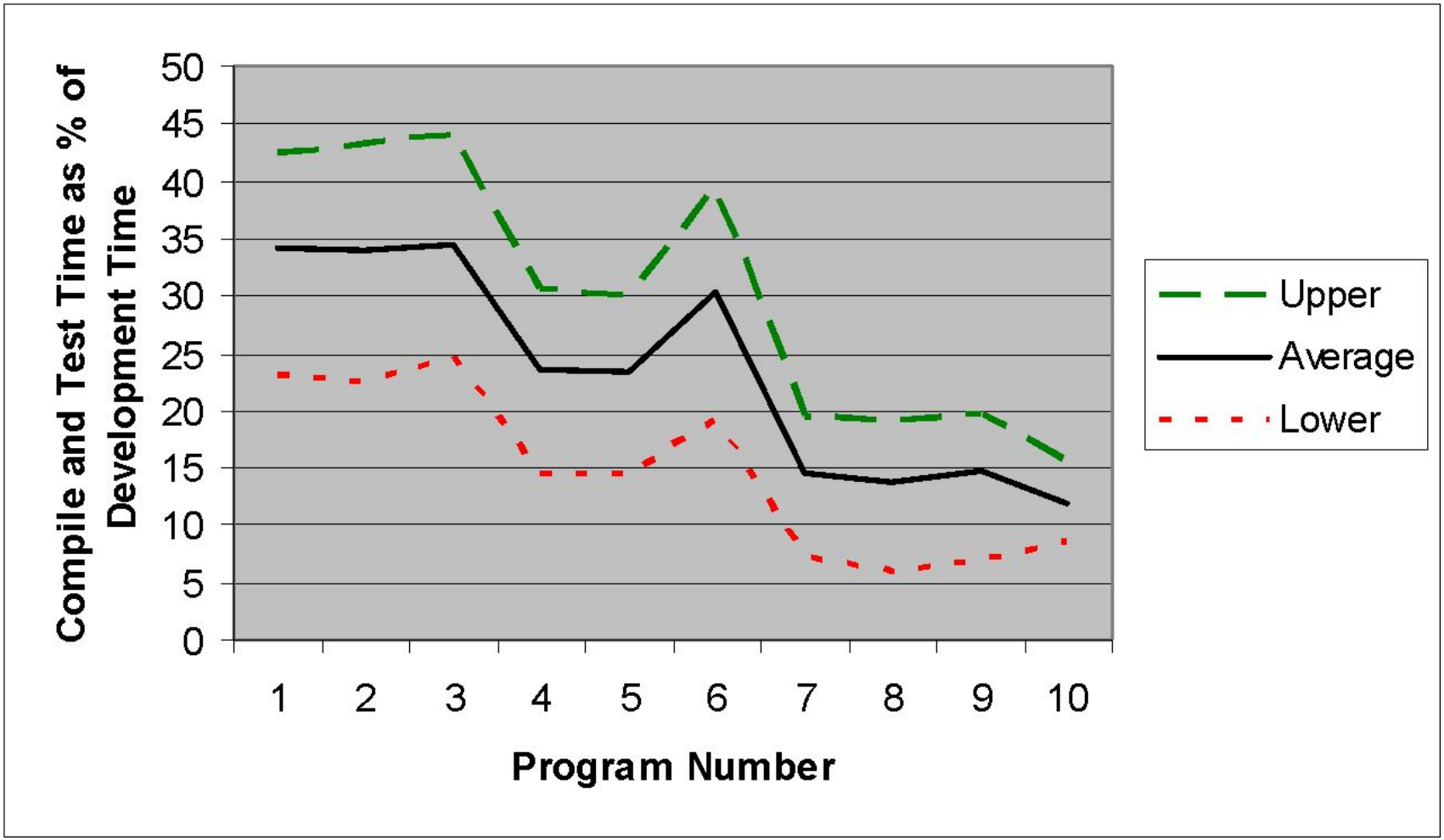
Majority are under-estimating

Balance of over- and underestimates

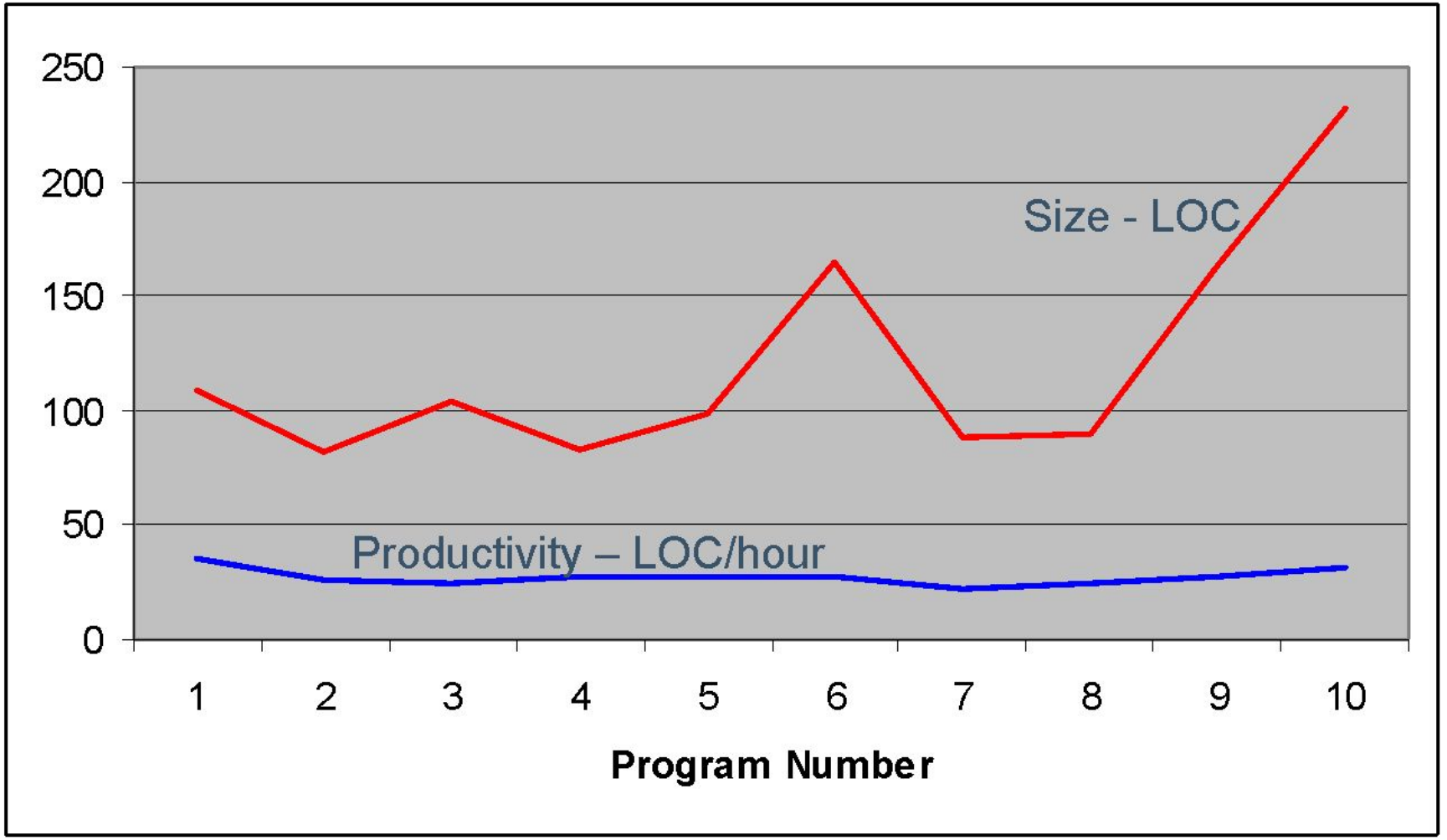
Much tighter balance around zero



Compile and Test Time – 810 Engineers



Size and LOC/hour – 810 Engineers



Messages to Remember

The PSP is a defined process that helps you do better work.

Once you have completed the course, you will know how to apply the PSP to your personal needs.

You will have the knowledge and skill to be on a TSP team.

With PSP0, the objective is to gather accurate and complete data on your work.