Using Computer-Based Performance Models to Address Resistance to PSP

Jim Hart

Software Process Dynamics, LLC

jim.hart@spdynamics.com

www.spdynamics.com

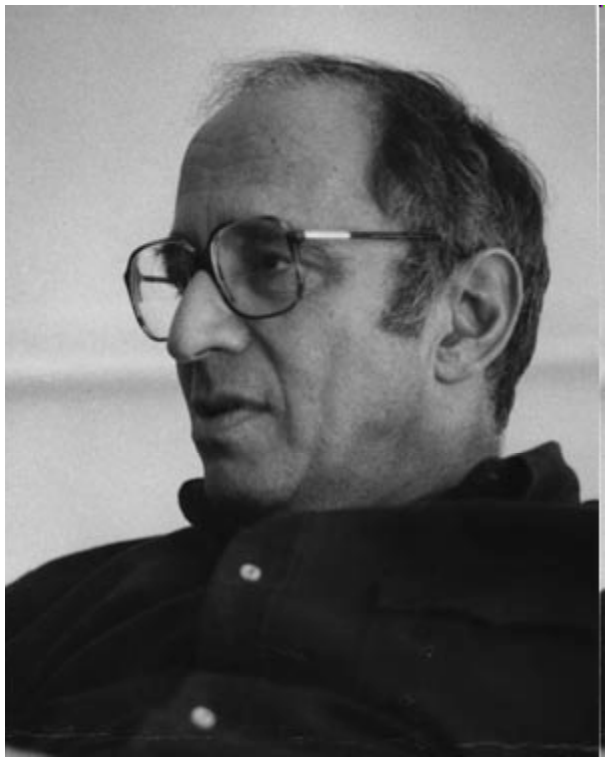
704-277-7444

Presentation Agenda

I. **Paradigms in Conflict**
II. Exposing and Working with Paradigms
III. PSP Computer Model
IV. Status and Future Work
V. Questions and Answers

Thomas Kuhn

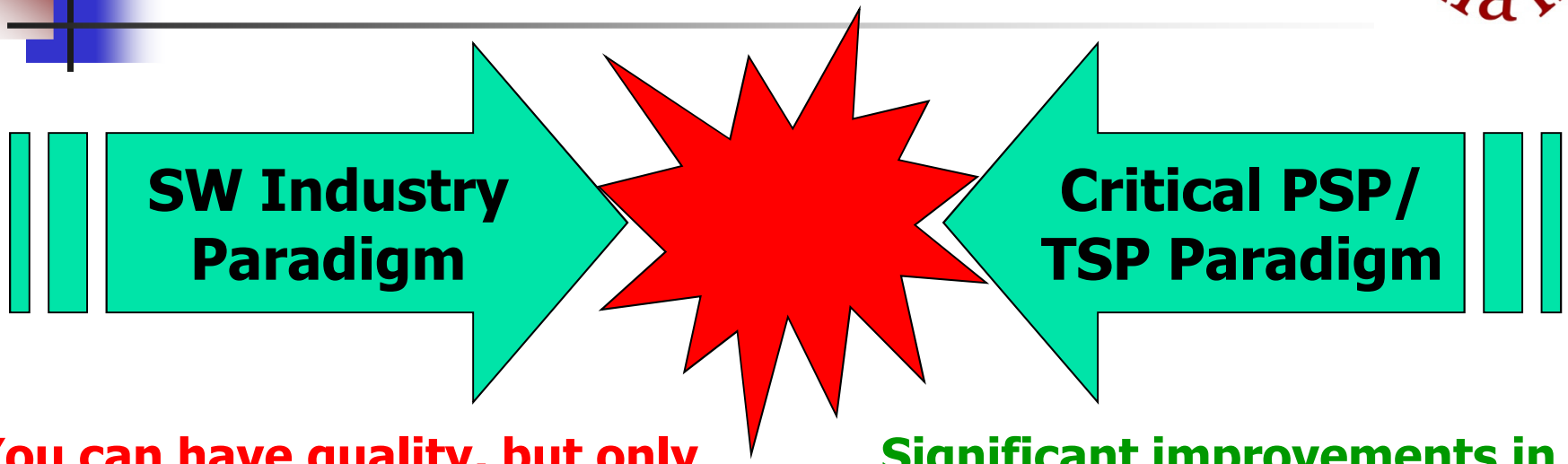
- Coined the term "paradigm" to describe the set of beliefs or truths that experts hold about their reality
- Discovered that when we are given factual evidence that conflicts with our existing paradigms, we are more likely to refute or ignore the data than we are to alter our paradigms



Technology Transition

- What is important is noting the ***effect*** paradigms have on us when we attempt to adopt technologies that are inconsistent with our beliefs.
- PSP/TSP technology deployment is challenging not so much because it is not valuable nor especially complex, but because it conflicts with deeply engrained but incomplete or inaccurate paradigms.
- **Having more data won't solve the PSP/TSP transition problem.**

Paradigms in Software



You can have quality, but only at the expense of productivity.

Time spent logging and analyzing detailed data about a developer's process is not productive time.

Compiling and unit testing code are highly efficient steps in catching most defects.

Significant improvements in quality are possible without loss of productivity.

Analysis of personal data gives significant insights into a developer's process

Reviews/inspections are more efficient at catching defects than testing.

"Overcoming" Paradigms

- Dealing with conflicting paradigms is handled by:
  - Ignoring them completely
  - Driving through them with force or coercion
  - Working with people by "bribing" them
- Problems with these approaches:
  - Expensive
  - Slow
  - Superficial
  - Risky

Presentation Agenda

I. Paradigms in Conflict

II. Exposing and Working with Paradigms

III. PSP Computer Model

IV. Status and Future Work

V. Questions and Answers

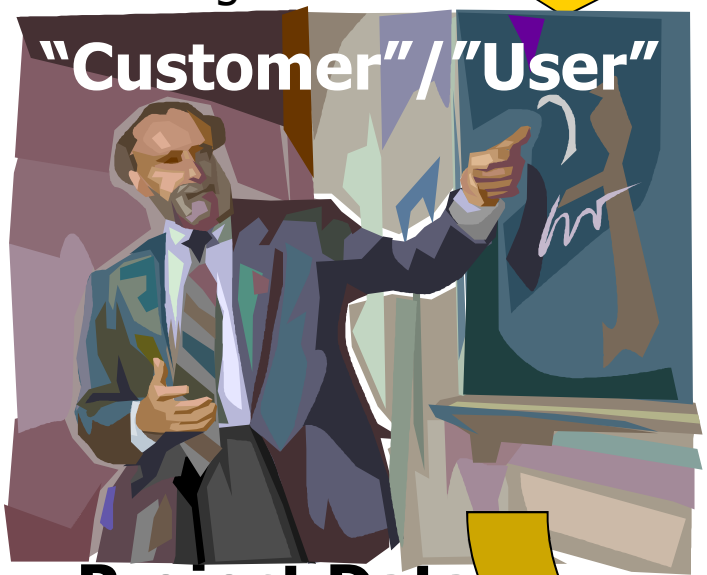
A Working Alternative

- Computer-based performance models
  - Have been effectively used to training new software project managers on how to manage a software project
  - Go well beyond the conceptual notion of managing a project by getting "students" to enact effective decision-making and behaviors desired in good managers
- Learning Laboratories
  - Expands on simulators to create comprehensive learning environments

Project Management Learning Laboratory

Aligns Plan to Actuals
- Change functionality
- Change costs
- Change schedule

Aligns Actuals to Plan
- Adjust staffing
- Work overtime
- Assign/shift tasking



Project System

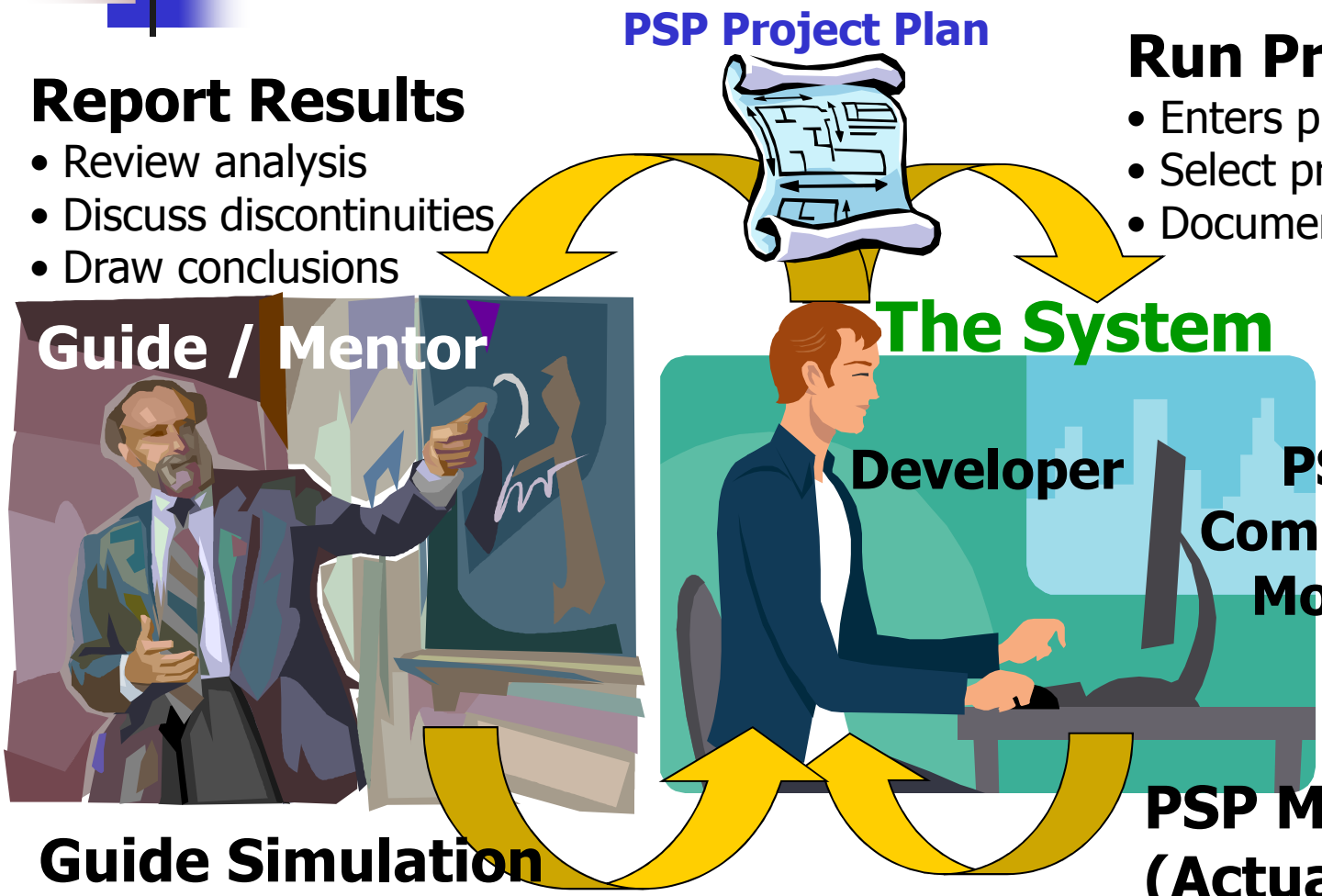
Project Data
- Requirements
- Budget
- Schedule
- Resources

Progress Indicators
- Staffing Profiles
- Production Levels
- Costs
- Defects

Simulating Counter-Intuitive Behaviors in Project System

- Brooks' Law:
  - Adding more people to a late project can make it later.
- 90% Complete Syndrome:
  - The project is half done when 90% of the software system is coded.
- The Mythical Man-Month:
  - Just because a woman can have a baby in 9 months doesn't mean 9 women can have a baby in 1 month.

PSP Learning Laboratory



Report Results
- Review analysis
- Discuss discontinuities
- Draw conclusions

Run Programs
- Enters planning data
- Select processes to use
- Document expectations

Guide Simulation
- Program requirements
- Training on scripts and use of PSP model
- Assist and mentor

PSP Measures (Actuals)
- Hours expended by phase
- Defects Injected/Removed
- Summary Graphs

Possible "What If" Laboratory Scenarios

- Run the set of PSP exercises with the following variations:
  1. Use prescribed PSP levels and performance measures to improve process capability
  2. Use prescribed PSP levels but do **not** use the performance measures to improve process capability
  3. Allow the user to set their own PSP levels and use performance measures to improve process capability (when available)
  4. Allow the user to set their own PSP levels and **not** use performance measures
- Because the entire set of PSP "A" programs can be simulated in several hours, users can try different scenarios to observe variations

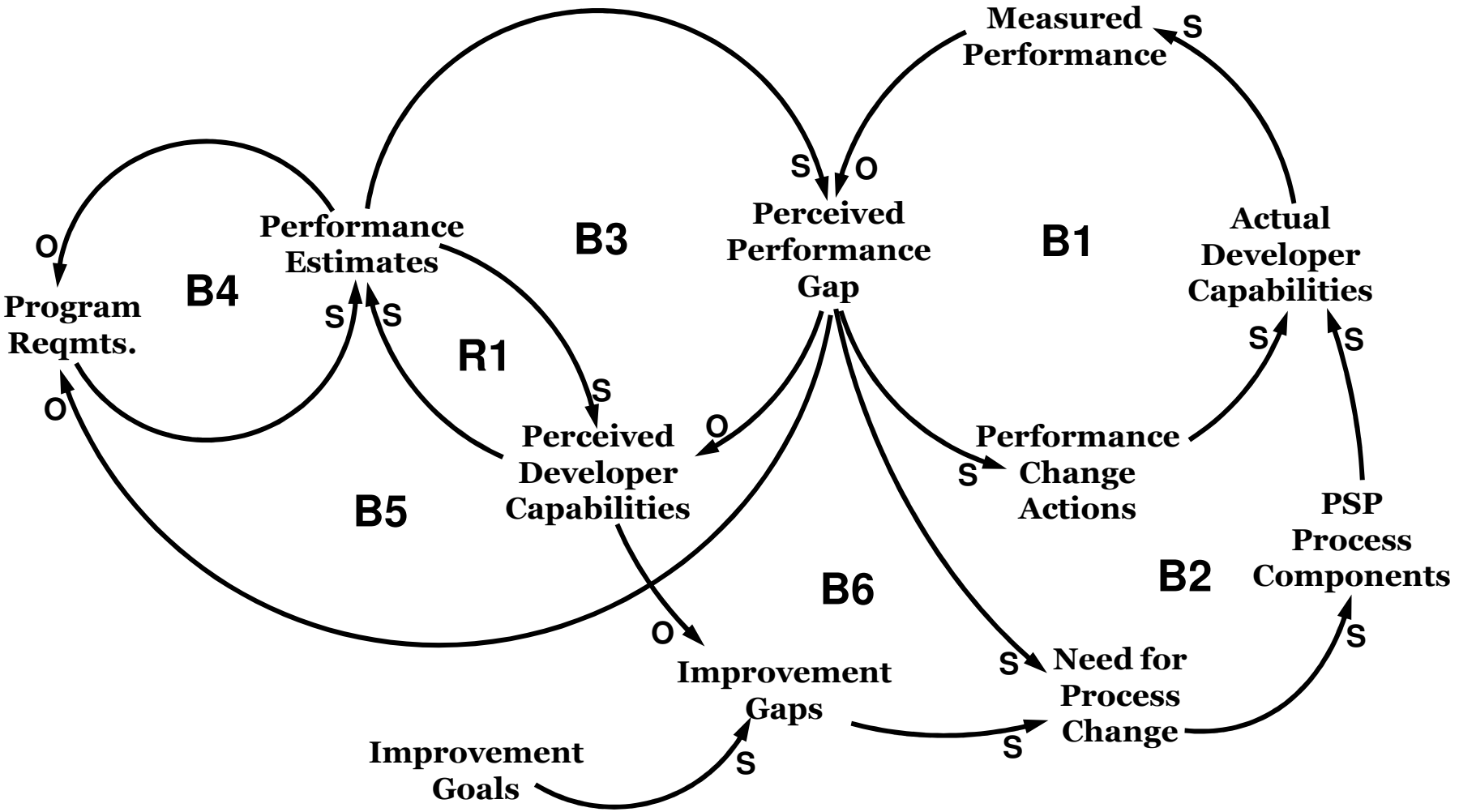
Presentation Agenda

I. Paradigms in Conflict
II. Exposing and Working with Paradigms
III. PSP Computer Model
IV. Status and Future Work
V. Questions and Answers

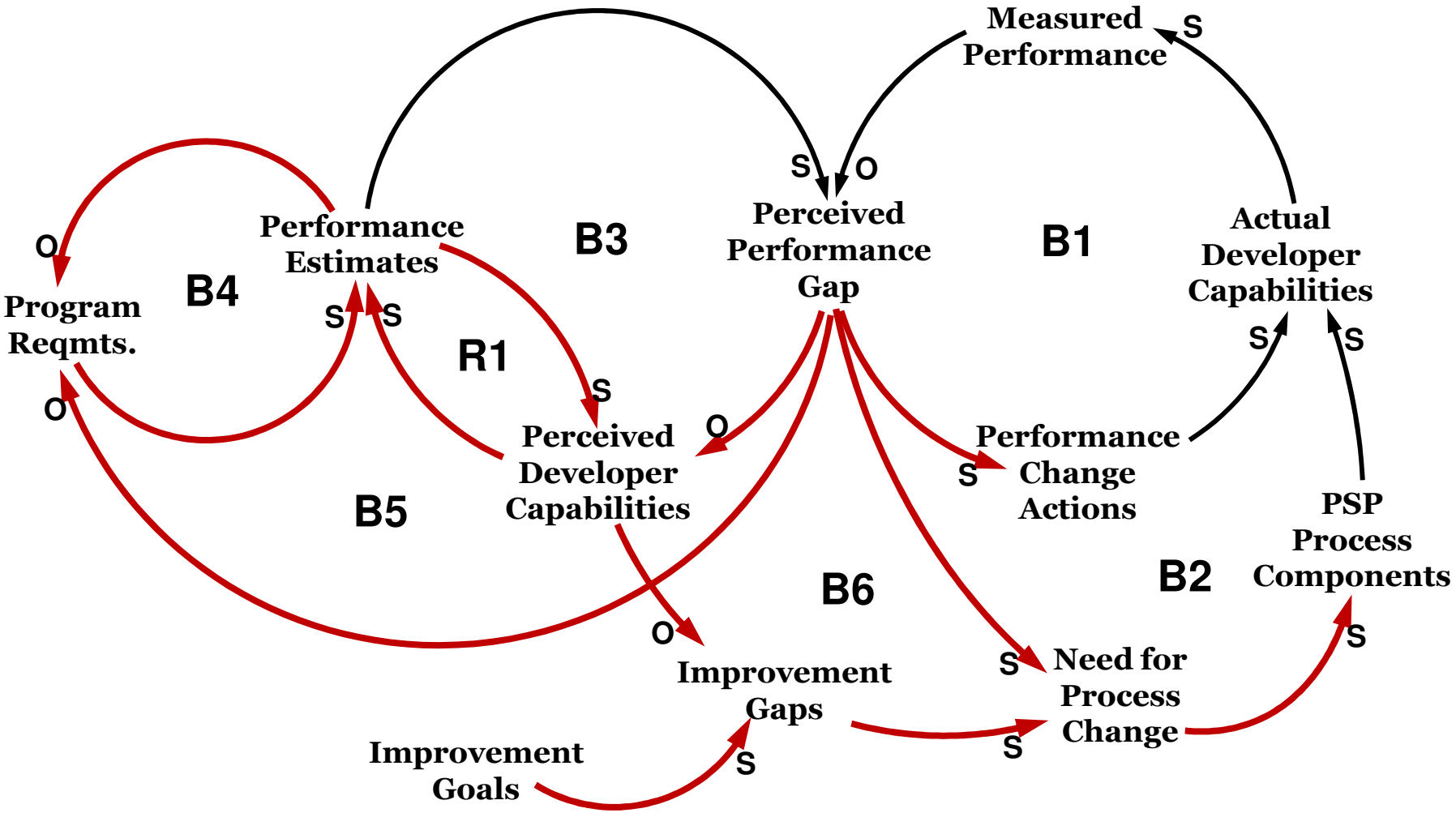
Value of a Dynamic PSP Model

- A dynamic model:
  - More accurately reflects what is really happening (that is, software development **is** a dynamic process)
  - Emphasizes behavior over procedure, thus better explaining why PSP/TSP gets the performance results that it does
  - Defines areas of feedback, which is the basis for learning

Causal Loop Diagram for Programmer/Model System

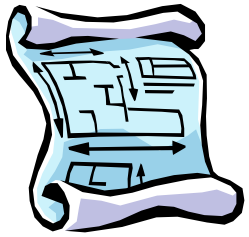


Decision Making Points in the Causal Loop Diagram



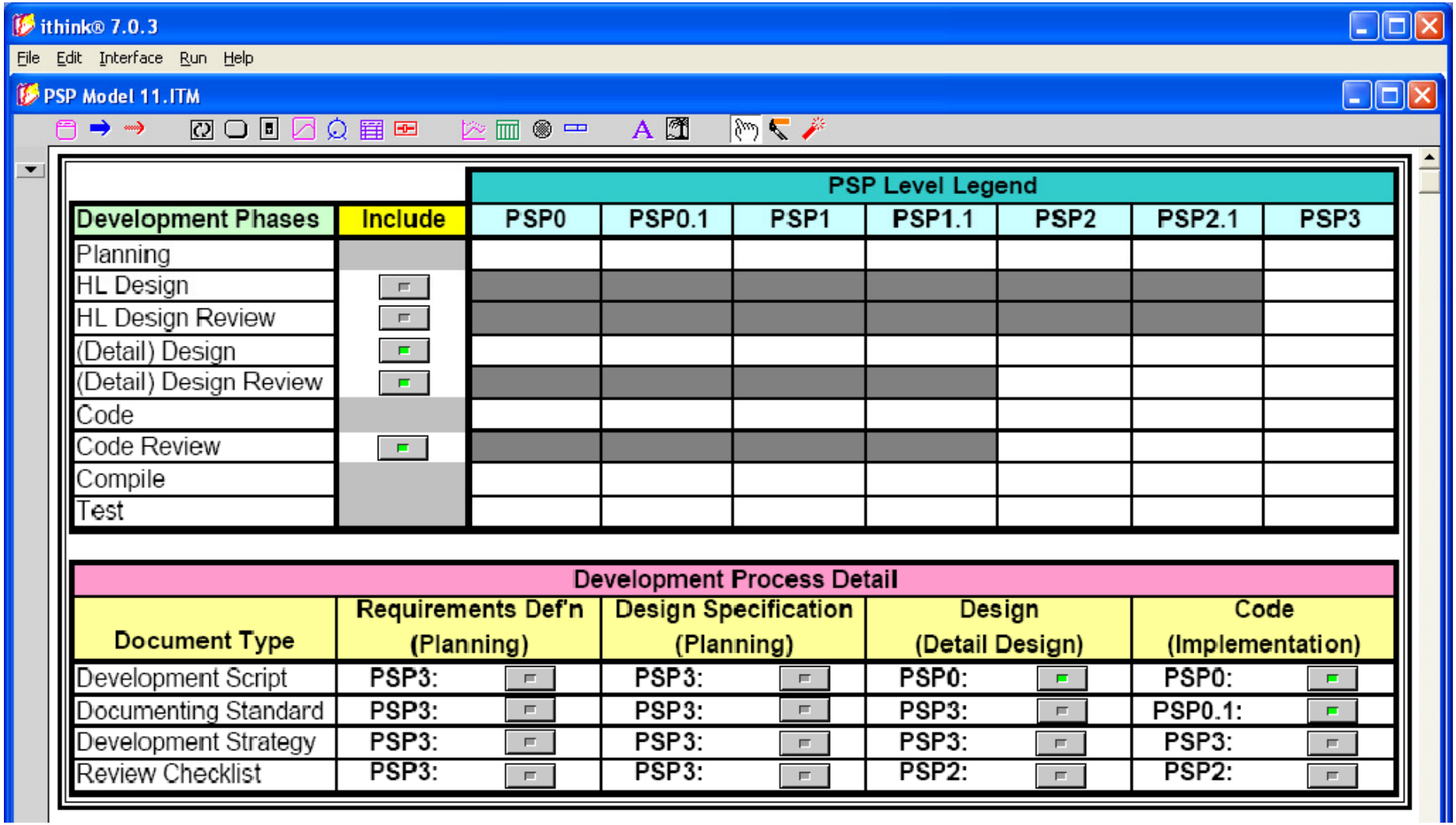
Developer Actions with the Model

- Planning data are limited to those essential in the PSP Project Plan
  - Program Size: B, D, M, R, N, Total New Reused from Estimation Process
  - Time in Phase: Total Time
  - Defects Injected: Total Development
  - Defects Removed: Total Development



- PSP process levels are selected and settings are potentially overridden (ON or OFF)

Example: Selecting PSP Process Components



Developer Capabilities

- Developer capabilities cover:
  - Productivity rates (development and defect fix times)
  - Defect Injection rates
  - Defect Detection rates
- The model adds variability (randomness) to the programmer capabilities
- The model includes "learning" from
  - developing programs
  - applying PSP process components

Developer Proficiencies

- Proficiencies are the basis for determining a programmer's capabilities
- A proficiency is a way to measure the developer's knowledge and skill in performing a task effectively and efficiently
- Proficiencies can improve over time. This is simulated in the model based on time spent on the processes used to develop the PSP programs

Developer Proficiencies

- Proficiencies identified and used in the model to cover all PSP Phases:
  - Estimating
  - Requirements Specification
  - Design Specification
  - Design (Detailed – Non-language specific)
  - Coding (Language Specific)
  - Verification: Reviewing
  - Verification: Testing
  - Process and Process Improvement

Developer Proficiencies

- A developer is given a proficiency rating of:
  - Beginner – New to performing the skill
  - Novice– Has a basic knowledge, but frequently requires time to look up information
  - Skilled – Having a working knowledge
  - Proficient – Above average knowledge and skill
  - Expert – A "wiz" or guru
- Proficiencies improve according to a bell curve.

Learning Cycle

Actual Developer Capabilities

Program Enactment

Performance Measures

Perceived Developer Capabilities

Developer Proficiencies

Important Points (1)

- The model is NOT a prediction technique and should not be used as one!
- A PSP Learning Laboratory is a way to address resistance through understanding, and is NOT a replacement for PSP training.
- A Learning Lab addresses resistance by:
  - Introducing PSP terminology and process levels
  - Demonstrating the value of processes, and gathering and using measures to improve them
  - Comparing possible effects of variations in processes used to develop software, e.g.,
    - Can implementing half of the PSP processes potentially give you half the gains? Less? More?
    - What are possible effects of not enacting PSP3?

Important Points (2)

- If you want to create an effective learning environment, allow people to:
  - experiment with real-world scenarios without fear of failure, injury or threat of reprisal
  - adequately reflect on what they observe
  - expose their paradigms by allowing them access to model details under the simulator
  - develop skills by doing, not listening
  - conduct self-directed learning, emphasizing the learners' personal responsibilities in the learning process

Presentation Agenda

I. Paradigms in Conflict
II. Exposing and Working with Paradigms
III. Computer-Based Model Concepts
IV. Status and Future Work
V. Questions and Answers

Current Status

- I have completed a working PSP model (generating the essential graphs).
- A NDA is in place with the SEI to give me access to the comprehensive PSP/TSP data.
- I am currently reviewing and documenting the model.
- I am writing grant proposals to continue this work effort into creating a TSP model.

Future Work

- I will be submitting the final PSP model to the SEI for review and verification.
- Once the model has been adequately reviewed, I will release a beta version of the model for public use and feedback.
- I will begin working on a TSP model that expands on the basics of the PSP model.
- I am looking for opportunities to pilot PSP/TSP learning labs in industry.

Presentation Agenda

I. Paradigms in Conflict
II. Exposing and Working with Paradigms
III. Computer-Based Model Concepts
IV. Status and Future Work
V. Questions and Answers

My Contact Information

- Please give me feedback and comments regarding this presentation

  jim.hart@stise.com

- Feel free to email me regarding:
  - How to become a reviewer of the PSP and TSP models
  - How to receive the PSP model (and documentation) when it is released
  - How to construct a PSP/TSP Learning Lab for potential executives, managers and development teams