

Custom Fitting TSPSM To A Maintenance Project: Patience and Perseverance Pay Off

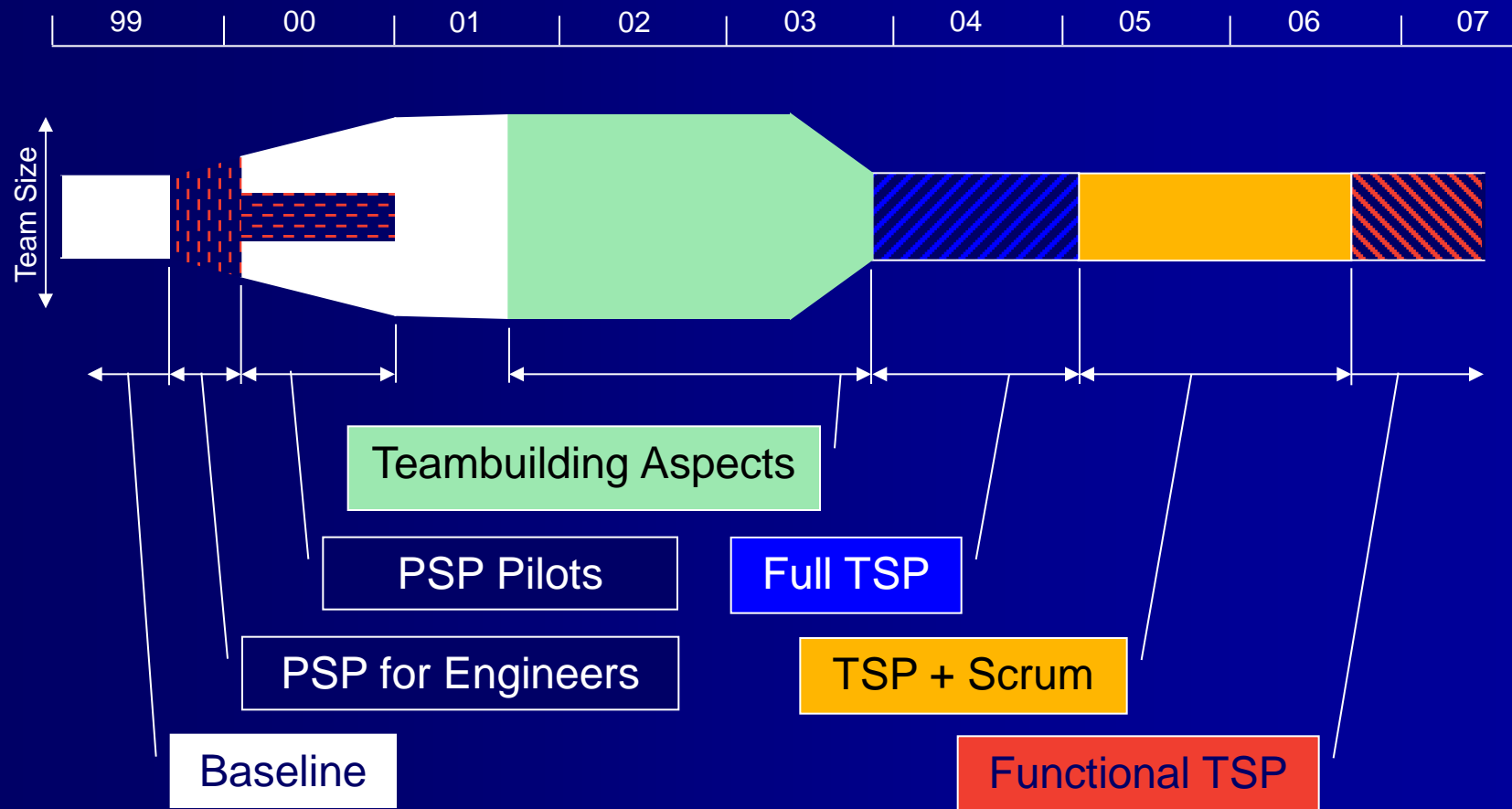
John Stark, Pam Skotak

TSP Symposium, 18 Sep 2007
Orlando

Project Background

- Weapon simulations and data processing
 - About 500 kLOC in three languages
 - Lots of third party source code
 - Work in a classified lab
- Software maintenance for over 20 years
 - Much of legacy code is poorly documented
 - Based on Change Requests (CRs)

History with PSP and TSP



PSP Pilots

- Used by a subset of the developers
- Benefits
 - Began methodical culture change
 - Discipline, metrics and tight teamwork
 - Project record-breaking schedule accuracy, quality, productivity

No PSP/TSP Activity

- Rapid team growth – diverted attention from PSP and TSP
- Project bogged down – team lost motivation

Teambuilding Aspects of TSP

- Implemented only the easier portions of TSP
 - No PSP training
- Benefits
 - Common understanding of situation
 - Continually good communication
 - Restored motivation
- Major challenges
 - Lack understanding of TSP fundamentals
 - Unpredictable quality and schedule

Challenges

- ✓ Motivation
- ☐ Limited TSP Understanding
- ☐ Unpredictable quality & sched

Full TSP Implementation

- TSP by the book; fully trained
- Benefits
 - Clearly visible performance
 - Enabled manager to lead
 - Boosted CMMI® progress
- Major challenges
 - Task definition and category
 - Data collection consistency and timeliness
 - Retention and use of historical data
 - Requirements volatility and validation defects
 - Some members having trouble creating and following plans

Challenges

- ✓ Motivation
- ✓ Limited TSP Understanding
- ❑ Unpredictable quality & sched
- ❑ Task definition and category
- ❑ Data collection consistency
- ❑ Use of historical data
- ❑ Rqmts volatility / val defects
- ❑ Creating & following plans

TSP With Scrum

- Added several Scrum ideas
- Benefits
 - Ability to rapidly change direction
 - Greatly reduced requirements volatility
 - Best team communication and understanding
 - Kept all individual plans in sync; frequent learning cycle
- Major challenges
 - Product size definition
 - Progress monitoring difficult
 - Analysis of defect data

Challenges

- ✓ Motivation
- ✓ Limited TSP Understanding
- ❑ Unpredictable quality & sched
- ❑ Task definition and category
- ❑ Data collection consistency
- ❑ Use of historical data
- ✓ Rqmts volatility / val defects
- ✓ Creating & following plans
- ❑ Product size definition
- ❑ Progress monitoring difficult
- ❑ Analysis of defect data

Functional TSP

- Broke into sub-teams with independent mini-launches
- Benefits
 - Straightforward monitoring of each release

Challenges

- ✓ Motivation
- ✓ Limited TSP Understanding
- ❑ Unpredictable quality & sched
- ❑ Task definition and category
- ❑ Data collection consistency
- ❑ Use of historical data
- ✓ Rqmts volatility / val defects
- ✓ Creating & following plans
- ❑ Product size definition
- ✓ Progress monitoring difficult
- ❑ Analysis of defect data

Team Dashboard

- Adopted the Team Dashboard inside the lab
- Benefits
 - Consistent and timely data collection
 - Accessibility and use of personal historical data
 - Better able to analyze defect data
 - Much shorter launches; better focus

Challenges

- ✓ Motivation
- ✓ Limited TSP Understanding
- ❑ Unpredictable quality & sched
- ❑ Task definition and category
- ✓ Data collection consistency
- ✓ Use of historical data
- ✓ Rqmts volatility / val defects
- ✓ Creating & following plans
- ❑ Product size definition
- ✓ Progress monitoring difficult
- ✓ Analysis of defect data

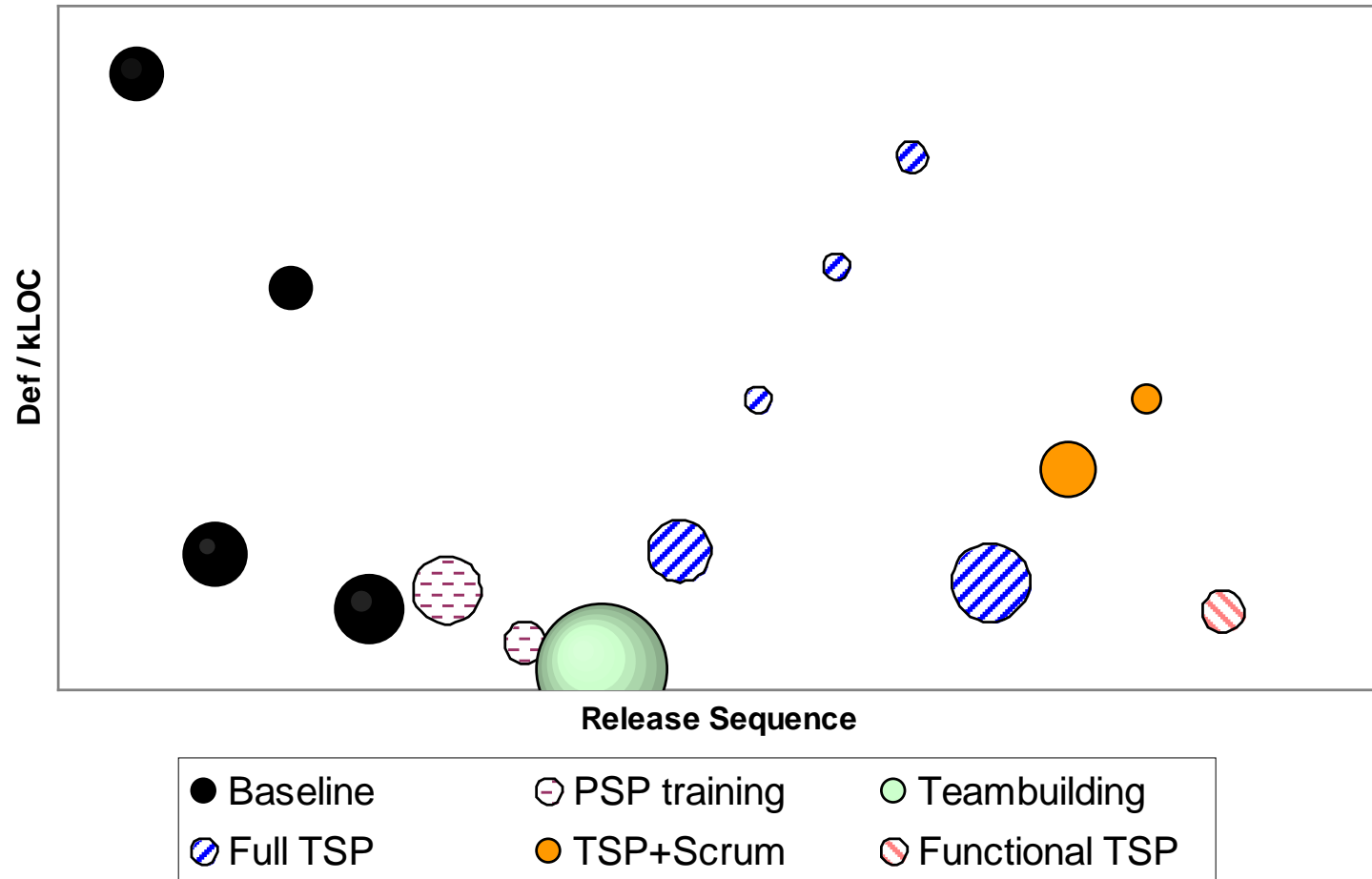
Revised Metrics Framework

- Created six different life cycles
 - Each CR implementation has its own life cycle
 - Several CRs bundled into a release (sub-project)
 - Team Dashboard support for pre-defined workflows
- Benefits
 - More relevant task plans
 - More meaningful size measures
- Major challenges
 - Quantifying quality

Challenges

- ✓ Motivation
- ✓ Limited TSP Understanding
- ❑ Unpredictable quality & sched
- ✓ Task definition and category
- ✓ Data collection consistency
- ✓ Use of historical data
- ✓ Rqmts volatility / val defects
- ✓ Creating & following plans
- ✓ Product size definition
- ✓ Progress monitoring difficult
- ✓ Analysis of defect data
- ❑ Quantifying quality

Post-Release Defect Density



Baselining Quality Measures

- To compare projects (sub-projects)
- Normalize defect densities
- LOC alone is insufficient
 - One recent sub-project:
 - 0 N&C LOC
 - 800 merged files
 - 300 hr
 - 4 acceptance test defects
- Current size measures are:
 - LOC
 - Merged files
 - Design pages
 - User guide pages
 - Test cases
- Still working on this one...

Lessons Learned

■ General

- Collect metrics where most of the work is done
- Engineers will use personal data if it is accessible and reliable

■ Maintenance-specific

- Frequent customer involvement
 - Use artificial cycle boundaries judiciously
- Manage each component/release (sub-project) with its own plan
- Use best life cycle and size measure for each change

Summary

- Many major problems resolved
- Many qualitative benefits
 - Better communication
 - Motivated, self-directed team
 - Performance is visible to management
 - Strong support for CMMI
- TSP is now highly tailored
 - Menu of life cycles
 - Appropriate size measures for each
- Data now consistent, reliable

Challenges

- ✓ Motivation
- ✓ Limited TSP Understanding
- ❑ Unpredictable quality & sched
- ✓ Task definition and category
- ✓ Data collection consistency
- ✓ Use of historical data
- ✓ Rqmts volatility / val defects
- ✓ Creating & following plans
- ✓ Product size definition
- ✓ Progress monitoring difficult
- ✓ Analysis of defect data
- ❑ Quantifying quality

Thank You!

- John Stark
starkj@saic.com
540-310-4956
- Pam Skotak
skotakp@saic.com
817-577-1122

SM TSP and PSP are service marks of Carnegie Mellon University

® CMMI is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.