What Works Best with TSPi for Small Team Productivity and Quality

William L. Honig, Ph.D.
Associate Professor, Department of Computer Science
TSPi Effectiveness with Small Teams

• TSPi impact on software teams
  – 23 teams of 7 to 12 graduate students on real world developments
  – Software process awareness and impact
    • Productivity coupled with quality
    • Result of planning and analysis
    • Extensive data collection

• Bringing real world software experience to the classroom
  – R&D leadership in communications companies
  – Land line, wireless, satellite, private and public networks
    • Voice, data, land line, mobile, satellite, network management
What Results?

• **Data Summary – Productivity**
  – Source Lines of Code (LOC) per Person Hour
    • High 47.4
    • Average 13.5
    • Low 1.8
    (complete Cycle 2 development, including reuse – all phases)

• **Data Summary – Quality**
  – Defects Injected per Total KLOC
    • Low 2.8
    • Average 24.1
    • High 86.3
How do the teams work?

• Team composition
  • Students assigned to Team
    » Based on From INFO
  • Roles matched to background
  • Demographic mixture
  • Well trained individual programmers

• Learning environment
  • 14 to 17 weeks of class
  • Strict enforcement of team discipline
  • Face to face team meetings required
  • Students ? Employees, but can be “fired”

Team Phoenix
Fall 2001
Team Roles at a Glance
(Five Specialized Roles)

• Support Manager
• Quality/Process Manager
• Planning Manager
• Development Manager
• Team Leader
Team Productivity – Cycle 2
Source Lines Of Code (LOC) per Hour

Productivity

<table>
<thead>
<tr>
<th>Teams</th>
<th>Productivity of Each Team</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rambler</td>
<td></td>
</tr>
<tr>
<td>Ice Cool</td>
<td></td>
</tr>
<tr>
<td>Lucid</td>
<td></td>
</tr>
<tr>
<td>Phoenix</td>
<td></td>
</tr>
<tr>
<td>Blue Bee</td>
<td></td>
</tr>
<tr>
<td>Dim Sum</td>
<td></td>
</tr>
<tr>
<td>Doc Max</td>
<td></td>
</tr>
<tr>
<td>Socrates</td>
<td></td>
</tr>
<tr>
<td>Kites</td>
<td></td>
</tr>
<tr>
<td>Titans</td>
<td></td>
</tr>
<tr>
<td>G10</td>
<td></td>
</tr>
<tr>
<td>eUphoria</td>
<td></td>
</tr>
<tr>
<td>Beta</td>
<td></td>
</tr>
<tr>
<td>Seals</td>
<td></td>
</tr>
<tr>
<td>Bees</td>
<td></td>
</tr>
<tr>
<td>Sharp</td>
<td></td>
</tr>
<tr>
<td>Silicon Raiders</td>
<td></td>
</tr>
<tr>
<td>Evolution</td>
<td></td>
</tr>
<tr>
<td>Avalanche</td>
<td></td>
</tr>
<tr>
<td>T3</td>
<td></td>
</tr>
<tr>
<td>Code Warriors</td>
<td></td>
</tr>
<tr>
<td>Phoenix II</td>
<td></td>
</tr>
<tr>
<td>Volki</td>
<td></td>
</tr>
</tbody>
</table>

Good
How is TSPi used in the classroom?

• Student teams complete two cycles of development
  • Same team assignment for both cycles
  • Some switch roles for cycle two
• “Customer” provides starting point
  • Product Needs Statement (not full requirements)
  • 2 to 4 meetings with customer to clarify needs and review requirements and plans
• Teams present key milestones and demonstrate product to faculty, research assistants, customer
The Process at a Glance (TSPi)

A controlled, data driven, step-by-step process for software life cycle

Launch → Strategy → Plan → Requirements → Design → Implementation → Test → Postmortem → Repeat
How do students learn PSP first?

• Personal Software Process (PSP)
  – Required for individuals
  – Prerequisite for TSPi

• PSP trial introduction
  – Undergraduate programming course
  – Plan (estimate time), track defects, record time spend

• Only some TSPi student teams have this experience before TSPi begin
  – Quick two day introduction
  – One programming project
Development Projects

• “Real World” Development
  – University staff groups as customers
    • working system or,
    • prototype or,
    • requirements clarification,…

• Wide range of applications
  – Prospect tracking for Graduate School
  – Summer visit registration for College of Arts and Sciences
  – Student Portal for Information Technology
  – Grant Approval and Tracking for VP Research

• Many technologies
  – C++, Java, XML, ColdFusion,…
How are data collected?

• Textbook: Watts S. Humphrey, *Introduction to the Team Software Process*™

• Key data entered weekly into 21 forms:
  – Product Summary (SUMP)
  – Quality Summary (SUMQ)
  – Work Tasks/Effort (TASK)
  – Schedule and Earned Value (SCHEDULE)
  – Defect Identification and Correction (LOGD)
  – Inspection Reports (INS)
  – Time Recording Log (LOGT)
<table>
<thead>
<tr>
<th>Name</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team</td>
<td>Instructor</td>
</tr>
<tr>
<td>Part/Level</td>
<td>Cycle</td>
</tr>
</tbody>
</table>

**Product Size**

<table>
<thead>
<tr>
<th>Requirements pages (SRS)</th>
<th>Plan</th>
<th>Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Other text pages</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High-level design pages (SDS)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Detailed design lines</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Base LOC (B) (measured)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deleted LOC (D)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modified LOC (M)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Added LOC (A)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reused LOC (R)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total New and Changed LOC (N)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total LOC (T)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Total New Reuse LOC**

<table>
<thead>
<tr>
<th>Estimated Object LOC (E)</th>
<th></th>
</tr>
</thead>
</table>

**Upper Prediction Interval (70%)**

<table>
<thead>
<tr>
<th>Time in Phase (hours)</th>
<th>Plan</th>
<th>Actual</th>
<th>Actual %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management and miscellaneous</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Launch</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strategy and planning</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirements</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>System test plan</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirements inspection</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>High-level design</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integration test plan</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>High-level design inspection</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Implementation planning</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Detailed design</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Detailed design review</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test development</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Detailed design inspection</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
What Results?
Defects Injected per LOC

Quality--a

Teams

Quality(a) of Each Team

Good
Quality Results from Cycle Testing ONLY

• In Cycle Testing determines the quality numbers
  – No “production” use recorded

• “Testing can only show the presence of bugs, not their absence”
  – Fault Seeding
  – Bug Density / Arrival Rate Analysis
Where are the Hours Used?

Total Time by Phase

Total Cycle 2 Hours by Phase

- Mgmt&Misc: 15%
- Launch: 5%
- Strat&Plan: 20%
- Requirements: 3%
- Design: 9%
- Implementation: 8%
- Test: 16%
- PostMortem: 24%

6396 Total Hours to Date
Student Outcomes

• Student Perceptions – Popular Course
  • Team work experiences very positive learning
  • Understand process – appreciation varies
  • Data collection a struggle
    – Volume of data needed
    – Needed for timely team cooperation

• My Viewpoint
  • Students well equipped to join industrial teams; larger team sizes work well
  • TSPi textbook is great on metrics and quality, limited on coverage of design, testing,…
  • Volume of “paper work” can lead to cybercrud
Students “Value” Forms

Greatest perceived value in forms that manage change and defects (red) and project plan creation and tracking (blue)

Student Survey: Choose the forms useful to your team.

Question 15

Greatest perceived value in forms that manage change and defects (red) and project plan creation and tracking (blue)
How do these findings apply to industry?

• Student teams approximate small industry task teams / development groups
  – Importance of (self) policing team behavior
  – Specialized roles help (in addition to developer role)

• Training / Coach / Observer role is critical to rapid introduction of process such as TSPi
  – Get through one cycle quickly to speed learning
  – Need Process Coach / Facilitator

• Face to face regular meetings
  – Weekly cycle of data, analysis, action
  – Emphasis on analysis and quality is key
    • Lead teams to analysis (not just data generation)

• Historical data a real help for getting started
  – If none, BEGIN NOW!
What about TSPi and Small Teams?

- Team data for 23 student teams show industry level productivity early in learning TSPi
  - Quality *always* needs focus
- TSPi can be learned efficiently and applied rapidly
  - Team composition and coaching
- The “academic” learning approach likely applicable to other types of organizations
  - Value of discipline, data collection, metrics
LOC Vary Greatly

Total LOC and Its Max. Min. and Avg.


Total LOC:
- Max: 14,000
- Min: 0
- Average

Locograms:
- Total LOC of Each Team
- Max
- Min
- Average
Larger Team Size Works

• Flexibility in Roles:
  – Some ability to switch roles
  – Easier to recover from “drop outs”

• Student Feedback:
  – Students identified the problems their team encountered
  – 20% felt a smaller team size of 5 would lessen the problems
What are some next steps?

Expand Focus on Analysis Metrics for In cycle Quality Improvement

Ease Data Gathering Travail Mobile Tool

Incorporate Teaching Materials on Technique Best Practices

Effectiveness of TSPi to Accelerate Transition to CMMI

Questions, follow-ups, ideas… contact

William L. Honig
whonig@luc.edu, 1-312-915-7988