Applying Functional TSP[SM] to a Maintenance Project

SEPG 2004

Ellen George
Steve Janiszewski
PS&J Software Six Sigma

Agenda

- The Maintenance Team
- Preparation
  - Setting goals
  - Collecting data
- Tailoring the TSP[SM] Launch
- Results
- Lessons Learned

The Maintenance Team Dilemma

- Organizations often have difficulty planning and managing a maintenance project because of the inherent unpredictability associated with responding to anomaly reports
  - Many anomalies don't even require a change
  - There is no correlation between the length of an investigation and the size of the resulting change
  - There is a feeling that any attempt at estimating the effort for the investigation task is futile
- This becomes a psychological barrier to using PSP[SM] and TSP[SM]

The Functional Team

The *functional* team has a functional, rather than a product, mission. While all the members may do similar work, they do not develop a single product and their individual tasks are usually quite independent. … [An] example would be a maintenance group where each member handles the repair and enhancement of a product. While several of the members might occasionally work on elements to be integrated into a common release, they would usually work alone. [Humphrey]

Project Characteristics

- Maintenance team residing partly in the USA and partly in Europe
- Tasks included upgrading the software to support new requirements, addition of new features, defect investigations & fixes
- 1 Manager (USA)
  - Attended all launch meetings but remained quiet
- 2 Leads (1 USA + 1 Europe)
- 9 Developers
  - Approximately half resided in USA and half in Europe
  - European team members tend to be very quiet during launches

Is It a Team?

- The current work activity is all maintenance.
- Each developer considered themselves responsible for a different product (turned out that they were each responsible for a component of single product)
- Each product is built and released separately.
- No task or schedule dependencies between developers
- Defects are generally localized to a component of the product
- ★ Individuals had only worried about their own component in the past

Questions

- **Maintenance Project**
  - **What are the project goals?**
  - **Why are these developers being launched as a team?**
  - **The nature of the maintenance work is highly interruptive, so how do we plan for “unplanned”, high priority interruptions?**
  - **What do we use as a size metric?**
  - **Is there a conceptual design?**
  - **What does quality mean?**

Launch Strategy

- The coach always needs a strategy for a launch.
- ★ For a “non-traditional” team, having a strategy is a key success criteria.
- Focus on *COMMONALITY* and *REPEATABILITY*
  - Find a means to get the individuals to gel as a team
  - Get the team to collect enough data prior to the launch so that they can create a budget for high priority interruptions
  - Understand how to “tailor” the launch to the needs of the maintenance team
  - Have the team collect time, defect and size data suitable for re-planning

Preparing the Project Manager

- *COMMONALITY* → Focus Program Manager on GOALS
  - Why is this team being launched?
  - What do you want to get out of this TSP launch?
  - How would you define a successfully operating team?
  - What compelling reason does this team have to work together?
- Suggestion:
  - Develop a long term vision.
  - Identify a series of short term (~1 month increments) goals that support the vision.
  - Focus on only 3 to 4 goals in the first month.
  - Make sure that the first incremental goals are achievable with low risk.

Preparing the Team Members

- ***REPEATABILITY*** → **Focus Team Members on DATA**
  - **Asked team to gather time on task data for two weeks prior to the launch**
  - **Objective of data gathering was to determine number of hours each week spent on planned vs unplanned activity**
  - **This data would be used to determine:**
    - **Time on planned tasks**
    - **Budget for time on unplanned tasks**
  - **Used an automated data collection tool to minimize overhead**

More on Goals

- Why are these developers being launched as a team?
  - To get control of their activity.
  - To become less reactive and more proactive.
  - To fill in holes in the knowledge/skills matrix.
  - Want to document the system architecture.
  - There is a large amount of “cut and paste” duplicated code in the system. Duplication could be eliminated.
  - With a good understanding of the system architecture, the team can use a reported defect as an opportunity to look for similar defects elsewhere in the system.

Tailoring

- Meeting 1,2: Goals
- Meeting 3: Conceptual Design
- Meeting 3: Process Plan
- Meeting 4: Top Level Plan
- Meeting 5: Quality Plan
- Meeting 6: Detailed Plan
- Meeting 8: Prepare Outbrief

Tailoring – Goals

- Goals
  - ★For each goal, ask the team why it was important to them, what would happen if they missed it, and how they would know if they were making progress against the goal.
  - –All goals were defined at the team level.
  - –The team decided to break down their goals into prioritized tasks, to be interleaved with their product tasks.

Tailoring – Conceptual Design

- Team didn't think of themselves as part of a bigger project, so they were hesitant to spend time reviewing the conceptual design
- ★ We decided to project a diagram of the system and have the design manager lead a discussion in which they visited each component and the "most knowledgeable person" told the others what they knew about it
- Each team member got a much better appreciation for the relationships in the system, the size of the components, and areas of risk (where nobody had knowledge, or knowledge was only one person deep)

Tailoring – Process Plan

- Maintenance process was somewhat defined prior to meeting.
- ★ Discussion focused around defining explicit entry criteria, exit criteria, and required approvals for moving from one process step to the next.
- Also discussed process and approvals required for handling priority interruptions.
- Little discussion or pushback when the “defined” process was presented. Upon discussion of entry/exit criteria and process for handling priority interruptions, even the more quiet team members got involved.

Tailoring – Top Level Plan

- Reviewed a list of known defects and service requests and identified owners.
- Owners led discussion on:
  - The amount of material that would have to be examined in the investigation phase to analyze each defect or service request (size)
  - An estimate of time for each applicable phase.
- ★ As a team, agreed on average hours on planned task per week based on data collected.
- Included previous goal tasks created, balanced the plan.
- ★ The team started to realize that much of what they previously thought was “unplanned” activity was actually known and “plannable”, and everything else was “budgetable”.

Tailoring – Quality Plan

- Didn't know how to measure quality for the legacy code – Defects per problem report? Defects per new and changed LOC? Defects per total LOC?
- ★ Spent considerable time discussing what quality means to their customer … what things annoy their customer … how to measure and track quality … what feedback they need from their customer.
- Put a plan in place to collect customer quality metrics.

Tailoring – Detailed Plan

- Each person went back to their desk and created their task lists in the TSP^{SM} tool.
- ★ Regrouped and generated multiple EV Plans.
  - Priority 0,1 Plan
  - Six Week Plan
  - Full Plan
- Tool made it easy to roll up several different plans

Tailoring – Prepare Outbrief

- Reviewed the launch script for preparing the outbrief
- This team was concerned that management would not let them re-launch as a team … made them work harder to justify their existence.
- ★ After outlining the standard slides, the team decided to add a few slides to explain why they are different from normal development teams and why they should be kept together as a team.
- 11 team members:
  - 8 generated the slides for the outbrief
  - ★ 3 used the time to write the detailed scripts for their new life cycle phases (investigation, solution tradeoff, implementation)

Postmortem – What Worked

- Team synergy was improved
- Setting team goals
- Meetings were well organized
- Role assignment
- ★ Increase in understanding of team roles and goals
- ★ Good job of balancing work load
- ★ Time spent defining what quality is
- Exposure of everyone else's jobs
- Good participation and contribution from everyone
- ★ Good to know that some troubleshooting procedures will be described

Postmortem – What Worked *(cont'd)*

- Started defining our common processes
- Management was informed about issues that cause concern
- Distributing responsibility
- ★ Process improvement tasks prioritized with regular work
- ★ Have a mechanism for better professional communication
- Start using our PSP℠ training
- Personal tasks may keep us from flailing
- ★ This seems like the birthday of this team.

Postmortem - Opportunities

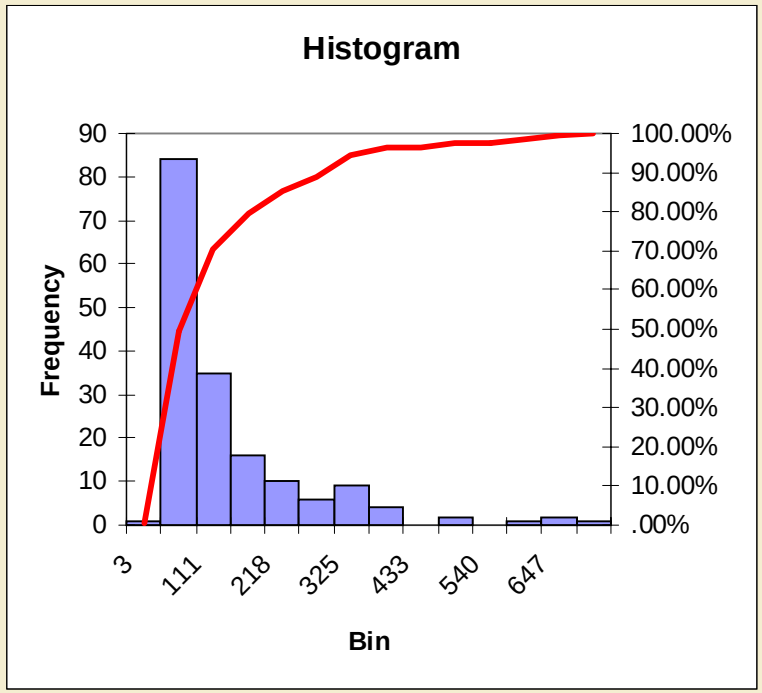
- We were here too late too many nights
- Should have made sure everyone had a computer with the TSP^{SM} tool installed before launch
- I should have done a better job of reading pre-launch materials.
- Documentation seems so repetitive that I don't seem to know what is new.

Estimating

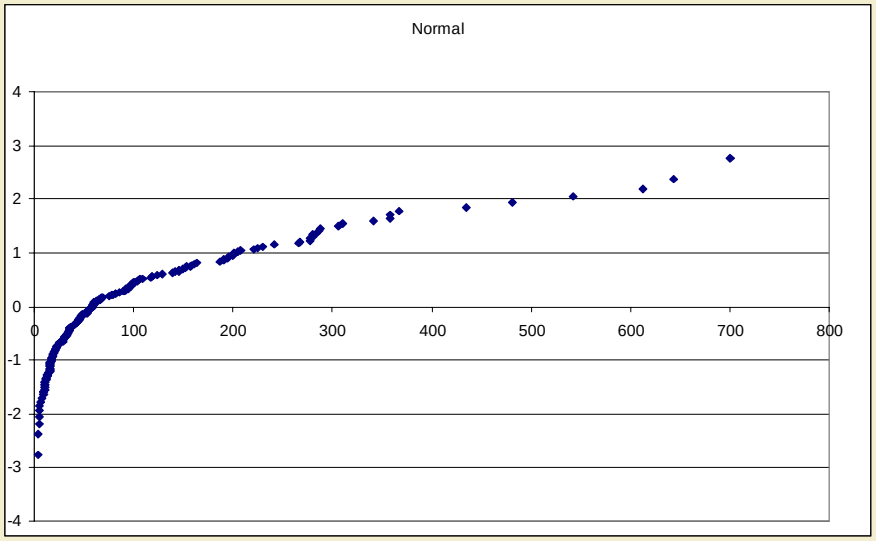
- Team was skeptical about ability to estimate anomaly investigation tasks during the launch
- Agreed to measure associated effort and attempt to develop an estimating algorithm from the new data
- Estimates during launch were “best guesses”
  - Without initial data, the team was able to estimate to within 41%
- Estimating algorithm was developed and effectiveness assessed during the postmortem
  - If the team used the new algorithm they would have been able to estimate with a 23% error, an improvement of 2x.

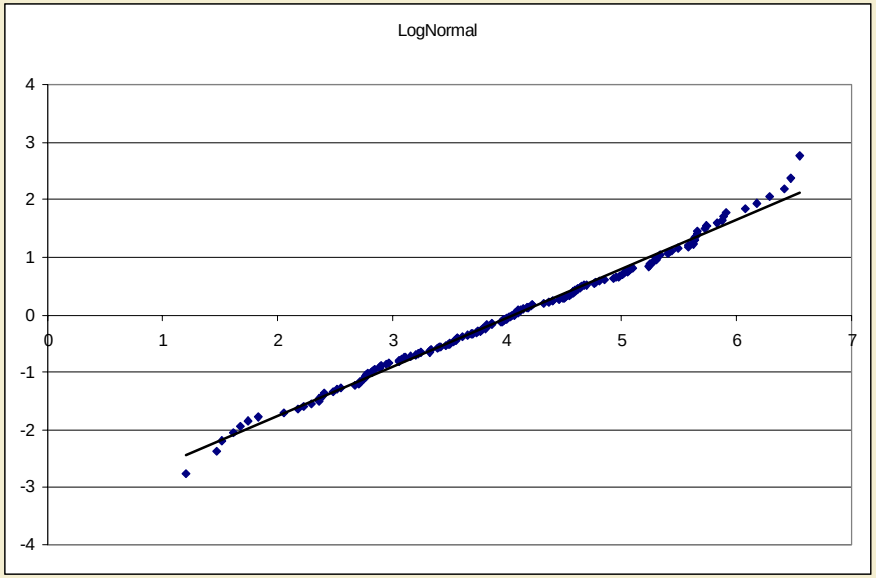
Anomaly Investigation Effort

- Collected data on the durations of anomaly investigation tasks on a legacy program for several months
- Generating a histogram suggests that a skewed-distribution function like a log normal would provide a good model of duration variability



Probability Plots





- Log normal does provide a reasonable model of the distribution function for anomaly investigations
- Size range indicates a mean length of 59 minutes
- ~ 70% of all investigations required between 18 and 187 minutes

	Duration
Very Small	6
Small	18
Medium	59
Large	187
Very Large	596

Estimating Algorithm

- This result leads directly to a simple technique for estimating anomaly investigations that can be employed during a TSP^{SM} launch
- For the tasks that are part of the current anomaly investigation backlog, categorize each one as very small, small, medium, large, or very large and use the estimated time provided by the table
- Pro-rate the total for all backlogged tasks to account for the unplanned investigations by using the historical percentage of time devoted to unplanned anomaly investigations
- In our case this was typically between 80% – 85%

Re-Estimating the Launch

- During the launch, we estimated the effort of each identified investigation as small, medium or large
- Assessment of effort was made by the task owner, based on familiarity with the functionality and amount of code that would need to be reviewed
- During postmortem, the original S/M/L estimate from the launch was used along with the calculated values of small, medium, large to re-estimate tasks

	Actual Time	Est Time (Launch)	Est Time (S/M/L)
Minutes	2549	4301	3313
% Error		- 41%	- 23%

- The re-estimate based on the calculated size ranges reduced the estimation error by a factor of 2 to a total error of 23%

Lessons Learned

- Prepare for the launch. Don't "wing it".
  - Focus on commonality to get the team members to feel and act like a team.
- Pick 3 to 4 very specific and achievable goals that will unite the team
- Focus on data to show the team that their work week is repeatable.
  - Team up developers for tasks such as generating the outbrief … builds camaraderie and moves things along faster
- Launch coach needs to have a project independent reporting path
  - Don't be afraid to use PSP 1.0 until you get some data

Contact Information

Look through the presentation CD for our talk on
"*Using PSP to Develop Software Requirements and Architectural Design*"
presented earlier in this conference on Tuesday, March 9 at 11:20 AM.

For additional information visit our web site or contact us at:

Ellen George 201- 358-8828
EllenGeorge@SoftwareSixSigma.com

Steve Janiszewski 201- 947-0150
SteveJaniszewski@SoftwareSixSigma.com

www.SoftwareSixSigma.com