AGENDA

• Objective

• Software Problem Report Prediction Modeling in the Department of Defense Environment
  • Data Collection
  • Analyze
  • Forecast
  • Control

• Pros and Cons

• Take Away/Lessons Learned
Objective

• “What if your Operation and Sustainment (O&S) budget was cut by X dollars?”, “What would be the impact?” During the maintenance phase of the program lifecycle that question is challenging to answer in a way that is meaningful; impactful.

• This brief aims to give a program office a tool to develop a defensible estimate for Software Maintenance. Through the alignment of historical Software data with planned lifecycle events; one can derive event driven factors and use trend analysis to forecast future Software Problem Reports (bugs).

• This method is a departure from calculations based on general industry factors to a more complete product-specific calculation.

• Provides a graphical depiction of predicted Software Maintenance costs across the life cycle. Helps answer the question “What does X Sustainment dollars buy me?” and “What level of Sustainment funding best meets user needs and is financially viable?”.

• Addresses the balancing of acceptable backlog (Backlog Management Index BMI) and affordability.

5/12/2015
What is a Problem Report

• A Problem Report (PR) can have many names: Trouble Ticket, Software Anomaly Report, Software Trouble Report, Software Deficiency Report, Defect, Bug, etc.

• It is typically submitted by a consumer/user to describe a perceived deficiency in a Software product.
  • For the purposes of this study, a formal Problem Report would only be recorded after the product has completed a formal verification test and has been approved for release by the program’s Configuration Control Board.

• Multiple avenues of detection: 1) Test Process: lab, regression, Design Verification Test (DVT), System Operational Verification Testing (SOVT), Development Test (DT/OT/FOT&E), System of Systems (SOSI), End to End (E2E), etc.; 2) Fielding Process: users or technical assistance.

• PRs are normally categorized by severity level, per IEEE 1044; ranging from 1 (Critical) to 5 (Aesthetic).

• In aligning with the Department of Defense Cost Element Template, the costs associated with the correction of Problem Reports fall under Software Change Product for Army, and Sustainment/Interim Contractor Support/Software Maintenance for Navy and AF.

• The following slides will dive into the details of how; Data Collection, Analysis, Forecast, and Control play a role in this approach.
SW Problem Reports- Data Collection

Collection: ✓ Analyze: Forecast: Control:

• Collect historical SW maintenance Data: Problem Reports
  • PRs Opened
  • PRs Closed
  • Severity Levels
  • Program Schedule
  • Software Version Releases
  • Major Test Events driving PR identifications

• Program Configuration Control Boards (CCB) usually maintain robust databases. Typically reports already exist that you can start with, and then fill the holes in from there.
SW Problem Report- Analyze

- Analyze historical data
  - PRs Opened
    - Plot PRs opened over time
    - Overlay key Events
    - Identify Special Cause Variations
    - Determine Upper and Lower Bounds for Common Cause Variation
      - Run Charts
      - Control Charts
      - Frequency Distribution
      - Probability Charts
    - Determine if data makes sense to breakout by SW version or at a top level
  - PRs Closed
    - Repeat for PRs Closed

Collection: ✓ Analyze: ✓ Forecast: Control:
SW Problem Report - Analyze

Step 1: Scatter Plot

- Use a scatter plot to depict number of PRs opened each month (# PRs-Opened over Time (Months)).
**SW Problem Report- Analyze**

**Step 2: Overlay Events**

- By overlaying the product lifecycle events, once can see if there is a correlation in spikes of PR counts with respect to specific events.

5/12/2015
• Common Cause Variation is defined as “a constant system that captures the expected amount of natural variation in the process”.
• Special Cause Variations are “caused by a source of variation that is not part of the constant system”.
• “Variation from common-cause systems should be left to chance, whereas special causes of variation should be identified”. The Special Cause Variations are identified by these event driven spikes.
By removing the Special Cause data points, the resulting stratified data set allows one to ascertain the typical average monthly PR Opened rate. This stratification is Common Cause Variation. Common Cause Variation accounts for typical variance in monthly count.
### SW Problem Report - Analyze

#### Step 5: Analyze Common Cause Variation

- **Identify special cause variations and determine control limits for the Common Cause variation to ensure the data is correctly bounded.**

  - Use only the Common Cause data points
  - Use all data points

**Product A Common Cause Run Chart**

**Product A Common Cause Control Chart**

**PR Frequency**

**Product A Probability**

5/12/2015
SW Problem Report - Analyze

Step 6: PRs Opened Trend Analysis

- Final scatterplot of PRs-Opened over Time, should now highlight trends discovered.

5/12/2015
SW Problem Report - Forecast

Collection: ✓ Analyze: ✓ Forecast: ✓ Control:

• Model Predictive Forecasts
  • PRs Opened
    • Use data derived from previous analysis steps to model forecasts
      • Use Common Cause Variation for average monthly rate
      • Add in Special Cause Variation
        • Identify important future schedule events within Product Life Cycle
          • Testing, Version Releases, Fielding
        • Identify known future enhancements and proportionally scale
        • Understand Product Life Cycle to determine appropriate Decay Point and Rate
  
  • PRs Closed
    • Use data derived from previous analysis steps to model forecasts
    • Look at different scenarios, and determine appropriate scenario to move forward with
    • Balance between acceptable backlog (Backlog Management Index BMI) and affordability
    • Apply cost per PR (computed from program historical data) to derive estimated Software Maintenance Costs
SW Problem Report - Forecast

Decay Curve

- Once Product reaches maturity, should see a decline in PRs Opened (Decay).
- Identify what the events are that determine maturity
  - Full Operational Capability (all users have had a chance to exploit the product; SOVT-derived PRs will fade)
  - The expectation that the full capability of the SW will be utilized
  - All known enhancements will be realized
- Assumes the PR Opened rate to decline at a rate of x% per year until it settles at a minimum average monthly rate of n PRs opened per month
- Assumes that the exception is when an enhancement is added to the waveform, in which the cycle repeats itself but at a proportionally lesser scale
• PR Opened Forecast Curve, uses the Common Cause Monthly average, and accounts for Special Cause Variations, Point of Decay, and any known future enhancements.
Graph illustrates Quantity of PRs. To derive costs, apply a cost per PR.

Scenarios are endless, chose one that balances acceptable Backlog Management Index (BMI) with affordability.

Sample Scenarios

- **Scenario 1)** Current Budget: Quantity PRs that can be closed within the current Maintenance Budget (fiscally constrained);

- **Scenario 2)** Historical Rate: Quantity PRs closed based on past performance (still partially fiscally constrained);

- **Scenario 3)** Closing PRs only with a Severity level of 1-3 (requirements perspective);

- **Scenario 4)** Achieving and maintaining a desired BMI level (done at either the total level, or for specific severity levels) (fiscally aware & requirements perspective).
• Backlog displayed by Severity Level.
SW Problem Report - Forecast:
Backlog Management Index (BMI)

If BMI is larger than 100, the backlog size has been reduced.
This metric illustrates repair responsiveness; how well am I managing my backlog.
Can be used as a driver for a forecast scenario, as well as a management tool.

PDF Image of Chart:

\[ BMI = \frac{\text{Number of problems closed during the month}}{\text{Number of problem arrivals during the month}} \times 100\% \]
Severity Levels defined by IEEE Standard 1044 (1 Critical; 5 Aesthetic).

Understanding the composite of Backlog by severity levels, can tailor modeling a scenario based upon targeting specific severity level BMIs.

Ideally, a program would want to keep Critical Severity Levels at a BMI of 100.
SW Problem Report - Forecast: Events

- Identifying which events are driving Problem Report Findings, improves the ability to estimate.
- Take into consideration how those drivers may alter through the Product’s Life Cycle, some may be triggers for Decay points/rates.
SW Problem Report- Control

- Realize ways to improve on collecting meaningful metrics. Metrics can be used to illustrate Measure of Effectiveness (MOEs).
- Investigate current processes in place for handling PRs and determine ways to streamline or tighten up procedures.
  - Is valuable data missing? Is some data unnecessary? What is the flow chart of a Problem Report from Start to Finish?
- Examine how SW Maintenance is enacted contractually.
  - How is a PR tracked? Is cost reporting required, and at what level?
- Routine updates to the forecast analysis should include rolling in new actuals and fine-tuning assumptions/factors. Forecast improves over time as more historical data is incorporated and special cause variations are identified and analyzed.
• Inputs into the PR database should be at a level that allows for easy generation of Metric Reports.

5/12/2015
SW Problem Reports- Pros and Cons

**Pros**
- Departs from generic high level constructs based on significant assumptions of commonality.
- Based on actual programmatic data, and aligns with an individual program’s unique lifecycle sustainment profile.
- Provides a way to quantify the SW maintenance effort, which enables the ability to create meaningful impact statements aimed at protecting Operation & Sustainment funds.
- Able to be continuously fine-tuned with periodic updates incorporating additional programmatic data
- Practical, defensible, data-derived, and adaptable.

**Cons**
- Requires actual programmatic data (post MS C / post sell-off testing).
  - If acceptable data exists from predecessor/analogous system, could develop CERs and use with appropriate applied risk.
- Data collection/reporting needs to be under configuration management.
- Forecast (data out) is only as good as data collection (data in).
- Requires knowledge of program (software release cycle).
- Actual cost and/or hours per PR can be difficult to accurately capture.
Defensible Impact Statements

• An X% budget cut, impacts the Program Office’s sustainment portfolio by slowing defect resolution (Y% PRs will go unfixed) and increasing the backlog (or BMI) by Z*.

• Reduced support yields increased RISK of R% to product sustainment.

• It will delay/deny key software patch(es) by M Months, limiting ongoing support to the Fleet/Warfighter/User.

• Reduced sustainment adversely impacts ability to rapidly respond to emerging threats and IA risk reduction.

* The ability to quantify a dollar impact to Sustainment is vastly beneficial in the existing fiscally-constrained environment. With PR analysis in place, impact statements can be even more detailed to call out specific PRs that will go unfixed, and how that will impede the warfighter. The “ripple” effect can be shown in the forecast models.
Parting Thoughts

• Per Dr. Boehm, Software maintenance cost estimates in 1976 ranged from 50 to 75 percent of the overall software life-cycle costs.
• The weight of maintenance sits heavy on a program’s budget, more emphasis should be placed on estimating such an extensive amount of a program’s lifecycle cost.
• Software Maintenance is a causality dilemma of which came first: the budget or the amount of Software Maintenance Problems to be resolved; one can only close as many PRs as one has funds to do the work.
• Consequently, when developing a prediction model there are continuous scenarios, what matters is finding the trade-off that best balances affordability and the amount of acceptable PR backlog (BMI).
• To achieve desired BMI, base your estimate on desired BMI level, and tailor from there.

Lessons Learned
  • Understand your product.
  • Understand your data.
  • Continue to fine-tune your analysis.
  • Incorporate needed improvements to data collection/analysis/ and control.
  • Best way to defend your budget is with actual data that allows the quantification of the projected Software Maintenance effort.

Practical, defensible, data-derived, product-specific, and adaptable
Back Up
Severity Definitions

• Severity 1- Critical: Prevents Essential Capability
  • Issue likely to result in a failure that affects/jeopardizes safety of operator, personnel, or the system itself. The issue prevents the accomplishment of an operational or mission essential capability. This severity level also includes most security-related issues.

• Severity 2- Major: No workaround
  • Issue results in failure of one or more channels within the system (application failure). Affects the accomplishment of an operational or mission-essential capability and no work-around is known.

• Severity 3- Adverse: Workaround
  • Issue does not result in a system failure, but causes system to produce incorrect, incomplete, or inconsistent results; adversely affects the accomplishment of an operational or mission-essential function. System usability is impaired, but a documented work-around is available.

• Severity 4- Minor: Operational Inconvenience
  • Issue does not cause a failure or impair usability; results in user/operator or development/support personnel inconvenience or annoyance but does not affect required operational or mission essential capabilities or accomplishment of development/support responsibilities.

• Severity 5- Exception: Other Effort
  • Non-conformance to standards, generally aesthetic issues. Also includes changes to documentation. Some enhancement requests may be designated with this severity.
Statistical Analysis

- **Histograms**
  - Gives a better view of the center, distribution, and shape of the data, as well as how much variation there is.
  - Make a histogram from the same data set as the run chart is based upon.

- **Cumulative Frequency Charts (Probability Chart)**
  - Represent the sum of all frequencies up to and including that value.

- **Stratification**
  - Used during data analysis to see if theories about causes or patterns are supported by data.
  - Dividing data points into separate sets based on different attributes.
  - Any time you think there maybe a difference between subsets of your data you can stratify it.

- **Variation**
  - There are two types of variation:
    - **Common Cause:** a constant system that captures the expected amount of natural variation in the process.
    - **Special Cause:** variation caused by a source of variation that is not part of the constant system.
  - It’s important to identify Special Cause variations first, so can then look at Common Cause variations.
  - Variation from common-cause systems should be left to chance, where as special causes of variation should be identified.
  - If any system exhibits only common cause variation, it is called “statistically controlled”.

- **Central Limit Theorem**
  - Stable distributions produce normally distributed averages, even when the individual data are not normally distributed.
Statistical Charts

• Run Charts
  • Identifying problems/opportunities (trends/patterns/variation).
  • Determining potential root causes.
  • Follow-up and verification of results.
  • They don’t identify exactly what the cause is, they just help you decide whether to look for something special or common in the process.
  • Since time based, they can tell you when special causes seem to be in effect.
  • Can show random variation associated with a process driven by common cause variation.

• Control Charts
  • An advanced run chart.
  • A run chart may not show any patterns that display special causes, but a control chart can show the expected amount of natural variation in the process.
  • It can identify data points that fall beyond the control limits, which means it can not be explained by common cause variation.
  • Divide the data into zones of standard deviations
    • Within +- 1 STD of average: results in 34% of data in each zone (68% total)
    • Within +- 2 STD of average: results in an additional 14% of data in each zone (96% total)
    • Within +- 3 STD of average: results in an additional 4% of data in each zone (100% total)
  • Control Limits are set at +- 3 STD.
    • If control limits don’t make sense, i.e. the lower control limit is negative, and your data does not allow a negative number, then use a control chart based on averages instead of on individual data.
      • The central limit theorem states that stable (no special causes) distributions produce normally distributed averages, even when the individual data are not normally distributed.