Mining Technical Issue Memorandums for Knowledge: *Challenges and Possibilities*

William Stanton

Annual NASA IV&V Workshop

September 11, 2013

Overview

- Background
- Potential Value of Mining TIMs
- Problem Description
- Solution Needs
- Available Technology
- Possible Mining Scenarios
- Potential Next Steps towards a Solution
- Wrap Up

Background

- In its 20 years of experience, the NASA IV&V Program has found a lot of defects in all phases of a software development lifecycle (Requirements, Design, Code, and Test)
- The IV&V Program documents each defect in a Technical Issue Memorandum (TIM)
- TIMs are submitted to the development project for disposition
- TIMs are archived in repositories by IV&V Project
- The NASA IV&V Program uses metadata about TIMs to build metrics that help ensure it is providing a valuable service
  - How early IV&V gets involved during development
  - Phase Containment of issues found by IV&V
  - Ratio of issues accepted by the development project
  - Top most frequent types of issues discovered by IV&V

A Fitting Analogy?

- Mine Earth for coal
- We can use coal for energy
- Energy makes our lives more efficient

- Mine TIMs for patterns
- We can learn from and apply patterns
- Knowledge gained from mining TIMS will make the IV&V Program more efficient

What could we learn from the Past?

- Most frequent issues found at every phase of the software development lifecycle
- Most frequent recommended actions for resolving a specific type of issue
- Most frequent potential impacts to a mission system capability adverted by IV&V
- New categorizes of TIMs
- Rules for characterizing a TIM
- Distributions of TIMs

How could we wield the Knowledge?

- Share most common issues found, along with their most popular recommended actions, with the software engineering community to help prevent future occurrences
- Aid an analyst in characterizing a new issue to gain efficiency in effort it takes to write a TIM to IV&V stakeholders
- Inform scope of work decisions for new IV&V Projects
- Aid an auditor in auditing the quality of TIM writing
- Refine issue categorization
- Reclassify existing TIMs

Challenges

- Over the 20 years, a TIM has not been consistently categorized
- Older TIMs have been archived
- TIMs are stored per project and are not collated across projects
  - Any collation of TIMs is done manually
- There are a lot of TIMs that have been created throughout the history of the NASA IV&V Program
  - Collating TIMs manually is time intensive due to the volume
  - Personnel of the NASA IV&V Program Already busy with current work load

What is Needed

- A solution that automatically collates a user-specified set of TIMs
- A solution that assists a user in building predictors that automatically learn from a user-specified set of TIMs
- A solution that automatically applies predictors on a user-specified set of TIMs
- A solution that can be easily modified to use on other collections of defects or other types of data
- A solution that any analyst could use
- A solution that does not require a lot of time to use

Available Technology

- Text Mining
  - Process the text documents (e.g. Tokenize, remove stop words, make case consistent, stem)
  - Transform the text data into vectors (e.g. TF-IDF)
- Data Mining
  - Measures two vectors for similarity
- Machine Learning
  - Builds prediction models by using vectors and their labels to infer rules

TIM Similarity Analysis Scenario

- A set of 600+ TIMs all categorized as an incorrect code issue type is fed into the system for automatic refinement into categories
- The system takes roughly a minute to automatically separate the 600+ TIMs into 30 clusters
- An analyst is able to view the TIMs assigned to each cluster and determine which TIMs do not belong to that cluster.

TIM Classification Scenario

- A user would like to determine which code issues were found by a static code analyzer tool and which were found by manual inspection
- The user uses the system to train a machine learner to classify issues
- A potentially different user feeds a large collection of unclassified TIMs into the system
- Within 5 minutes the system automatically classifies each TIM according to the model it had learned
- An analyst is able to view the classification of each TIM and determine if the TIM was appropriately classified.

Challenges with Technology

- Selection of mining algorithms
- Selection of machine learning algorithms
- Selection of Text Processing Libraries, Data Mining Engines, Machine Learner
- Accuracy of automatic categorization
- Appropriateness of automatic issue classification
- Issues often contain text that would throw off automatic categorization schemes (e.g. code snippets)
  - Issues would have to be preprocessed
- Technology complexity
  - The solution would have to be designed such that its users do not need knowledge of mining techniques

IV&V Exposure to Technology

- Automated Issue Report Similarity Analysis
  - Bojan Cukic, Sean Banerjee
  - Presented at NASA IV&V, November 2012
  - Presented various ways to measure defect similarity
- Automated Severity Assessment of Software Defect Reports
  - Tim Menzies and Andrian Marcus
  - Presented at ICSM 2008
  - Predicting issue severity based on issue description
- Learning better IV&V practices
  - Tim Menzies, Markland Benson, Ken Costello, Christina Moats, Melissa Northey, Julian Richardson
  - Journal publication, 2007
  - Looked at predicting frequency of high severity defects based on SILAP rationale

Potential Next Steps

- Research on best
  - Text mining algorithms
  - Machine learning algorithms
- Trade Studies on
  - Text Processing Libraries
  - Data Mining Engines
  - Machine Learning Algorithms
- Description and Evaluation of possible architectures for a software solution
- Reach out to other organizations within the agency that have starting using similar technology
- Investigate partnerships with Academia

Importance of writing good TIMs

- Impact statements describe what can happen to the system capability and the impact to the mission or safety
- Recommended Actions are sufficiently detailed without prescribing how to change the affected artifact(s)

References

- T. Menzies, A Marcus; "**Automated Severity Assessment of Software Defect Reports**", 24th IEEE International Conference on Software Maintenance (ICSM) 2008
- D. Port, A Nikora, J. Hihn, L. Huang; "**Experiences with Text Mining Large Collections of Unstructured Systems Development Artifacts at JPL**", 33rd International Conference on Software Engineering (ICSE) 2011