GSAW 2020 Tutorial F:

Large Scale Mission Software Data Exploitation

Length: Half day

Overview:
Many ground systems operate for long periods of time, and will have large amounts of software. Even with extensive pre-launch testing, their software will have defects and vulnerabilities that need to be fixed, and further defects and vulnerabilities that emerge as the software evolves. Studies of long-lived software have found that the later the defects and vulnerabilities are found and fixed, the more expensive will be the fix. This phenomenon is now known as technical debt, as the increased cost of making the fix is similar to paying interest on the cost of the fix. Further, delays in living with defective software will have serious impacts on the ground systems' operational capabilities. Thus, investments in methods, processes, and tools for performing large-scale data analytics of their ground system software, or other software of interest will have large returns on investment. A recent study by the Consortium for Software Quality estimated that the cost of technical debt of software worldwide is roughly $2.8 trillion.

This tutorial will summarize current research and operational results from performing large-scale data analysis on large-scale software technical debt, using one of the leading technical debt analyzers, USC's Software Quality Understanding by Analysis of Abundant Data (SQUAAD) system as an example.

SQUAAD is a comprehensive framework including a cloud-based automated infrastructure accompanied by a data analytics and visualization toolset. SQUAAD has been documented in multiple recent research publications empowering their empirical studies and is used by a major governmental entity and a major commercial corporation. Our approach to conduct large-scale replicable empirical studies on software evolution has been to capitalize on cloud services to analyze full maintainability and technical debt commit histories of large families of open-source software systems available through GitHub. SQUAAD automatically:

1. Retrieves a subject system's metadata (e.g., number of contributors) as well as its commit history from GitHub.
2. Distributes hundreds of revisions (i.e., official releases and/or revisions created by commits) on multiple cloud instances.
3. Compiles each revision and runs static/dynamic programming analysis techniques.
4. Collects and parses the artifacts generated by programming analysis techniques to extract quality attributes.
5. Runs various statistical analysis on software quality evolution.We recently delivered advanced tool assessments tutorials to front line acquisition engineers of a major governmental entity. This led to an in-depth analysis of the quality aspects of an open source software complex for decisions regarding quality, safety, and security "sniffs" and "taints" to assess an acquisition program of an unmanned system.

The outline for this tutorial is as follows:

- Introduction
  - Applying large scale data analytics to Ground System software.
  - Mining Software Repository.
  - Importance of Commit Level Software Evolution Analysis.
- Software Quality Metrics and Their Inference
  - Basic, Code Quality, and Security
  - Technical Debt

- Commit-Impact Analysis
  - Targeting a Software Module.
  - Understanding the Impact of Every Developer.
- SQUAAD Interactive Demo
  - Open Source Case Study.
  - Private Cloud for DoD Affiliated Organizations.
- Discussions and Conclusions

Instructors: Barry Boehm and Pooyan Behnamghader, University of Southern California, Center for Systems and Software Engineering

Biographies:

Dr. Barry Boehm is a USC Distinguished Professor and the TRW Professor in the USC Computer Sciences, Industrial and Systems Engineering, and Astronautics Departments. He is also the Research Council Director of the DoD-Stevens-USC Systems Engineering Research Center, and the founding Director of the USC Center for Systems and Software Engineering. He was director of DARPA-ISTO 1989-92, at TRW 1973-89, at Rand Corporation 1959-73, and at General Dynamics 1955-59. His contributions include the COCOMO family of cost models, the Spiral family of process models, and the Theory W (win-win) approach for creating and evolving successful systems. He is a Fellow of the primary professional societies in computing (ACM), aerospace (AIAA), electronics (IEEE), systems engineering (INCOSE), and Lean methods (LSS), and a member of the U.S. National Academy of Engineering.

Dr. Pooyan Behnamghader received his Ph.D. in computer science from the University of Southern California. He currently works at the Center for Systems and Software Engineering under the supervision of Dr. Barry Boehm. Pooyan received his BS (with honors) from the University of Tehran, Iran. He is a recipient of USC Provost's Ph.D. Fellowship and is ranked third in the 16th Iran National Scientific Olympiad for University Students in Computer Engineering. His research focuses on mining software repositories and software quality trade-space. More information is available on his homepage: http://behnamghader.net/.

Description of Intended Students and Prerequisites:

Persons interested in scalable data analytics systems to cost-effectively improve their ground systems software's reliability, availability, maintainability, security, and life cycle efficiency.

What can Attendees Expect to Learn:

Methods, processes, and tools for performing large-scale data analytics of their ground system software, or other software of interest.