Journal of Materials Science and Nanomaterials

Extended Abstract Open Access

Towards A Systematic Approach to Actionable Software Engineering Analytics: Applications of Data Science & Machine Learning

Muthu Ramachandran

Leeds Beckett University, UK

Software Engineering Analytics & Applications

Traditional software engineering has established well-proven software development practices, techniques, and methods that we have enjoyed over fifty years or so. Besides, SDLC has also produced an enormous amount of datasets that have been deposited in various popular platforms such as PROMISE (2020). However, there is a complete paradigm shift to service computing since early 2000 and the evolution of cloud computing which has forced us to deliver software products as a service on-the-fly. Service computing, Cloud computing, DevOps Software Engineering, IoT, and integration with Data Science Applications have revolutionized the way we develop software as a service. Besides, it remains challenging to understand how to apply Software Engineering Analytics (SEA) and repositories to new product development and cloud-native software services. This paper presents a systematic approach to Software Engineering Analytics which used data science and machine learning tools to learn from the past and make decisions, actions precisely. This paper also highlights some of the software engineering analytics applications and challenges such as reuse of requirements with process mining, reuse of test cases with bug predictive modelling, and predictive modelling for software process improvement cloud services.

Big data analytics, visualisation, and predictions have been useful and very popular research and applications in recent years. However, in the context, the application of the big data practices to Software Engineering Analytics, there are some key questions need to be addressed:

• How do we apply to software engineering analytics?

• How do we collect and access Software Engineering experience data?

• How do we apply them to decision making for business as well as the SE practices: process, methods, and technology?

• How do we apply predictive modelling to on-the-fly cloud-native services?

• How do we capture software problems in real-time and to actioning remotely (DevOps)?

• How do we apply data science and machine learning across the software engineering life cycle, cloud software engineering life cycle, and DevOps Software Engineering pipeline?

• How do we systematically merge Software Engineering Analytics across the software development life cycle and DevOps Software Engineering life cycle which requires real-time action?

• What are the cost estimation techniques for SEA?

Software engineering analytics have emerged as part of growing software repositories such as PROMISE, Bug Prediction Datasets, etc. and have recommend real-time analytics (Yang, et al. 2018; Menzies and Zimmermann 2013; Ramachandran 2019). They have also reported that as of late 2012, web searches show that Mozilla Firefox had 800,000 bug reports, and platforms such as Sourceforge.net and GitHub hosted 324,000 and 11.2 million projects, respectively. The PROMISE repository of software engineering data has grown to more than 100 projects and is just one of more than a dozen open source repositories that are readily available to industrial practitioners and researchers.

Furthermore, fifty years of software engineering research and practices have produced more than sixty software development methodologies, fifty static analysis tools, forty software design methods, thirty-seven benchmark organisations and standards committees/board, twenty-five software size estimation and complexity analysis metrics, around twenty software project management methods and tools, more than twenty-two software testing methods and tools (Erdogmus, Medvidovic, and Paulisch 2018). Furthermore astonishingly, there are at least 3000 programming languages that have been developed and used in software developments, even though only 100 were frequently used. There are at least new programming languages that have been discovered and announced every 2 weeks, and new tools have been developed for more than one in each month. Also, there are new methodologies are discovered for at least every 10 months. Win current advancements in DevOps and DevSecOps, there are hundreds of cloud-naïve tolls that have discovered and developed almost monthly if not weekly. However, still there is no silver bullet on the software crisis, complexity and software productivity remains unsolved (Ramachandran 2019).

Besides, Ozkaya (2020) recommends current software engineering skills needed to change and demands the inclusion of data science and machine learning experiences in addition to software engineering skills. Therefore, SEA will revolutionize software and cloud computing industries soon in a much different way towards achieving improved quality, productivity, and cost efficiency for sustaining software workforce.

Furthermore, we need to adopt a comprehensive machine learning tools after cleaning and assessing datasets to conduct predictive modelling to identify actionable outcomes in terms of software artifacts: reuse of requirements, reuse of design, reuse of testing, reuse of quality attributes, and reuse of processes and patterns. Besides, we also need accuracy metrics to be applied to the outcome of machine learning predictive models. Therefore, in our SEA projects, we have adopted Azure ML as it provides many ML algorithms as a service such as two-class boosted decision tree which provides an excellent accuracy, two-class neural network, two-class averaged perception, and two-class support vector machine which seems to provide an excellent fast training time.

However, current research studies on actionable software engineering analytics lacks providing a systematic approach to analytics, decision making, and making actions autonomically with cloud-native services. Therefore, this paper proposes a systematic approach which is shown in Figure 1.

Journal of Materials Science and Nanomaterials

Extended Abstract Open Access

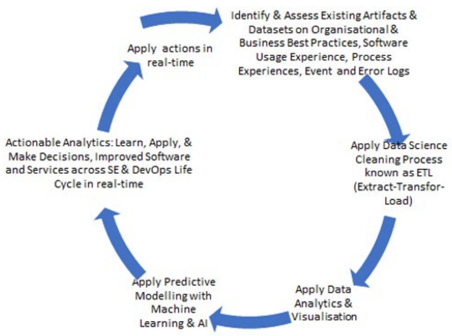

Figure 1 Software Engineering Analytics Framework (SEAF) for SDLC

According to SEAF, the process firstly starts with identifying and assessing existing datasets on various aspects of any software businesses, product history, event and error logs, user experiences, etc. Secondly to apply data cleaning techniques known as ETL (Extract, Transform, and Load) which can be done using existing data analytics tools such as Tableau, SAS, and Mathematics. Thirdly, to apply visual analytics and conduct queries to extract past experiences and best practices which then can be used for making decision as well as embedding then into software artifacts such as reuse of requirements, reuse of design, reuse of testing and bug tracking for build-in trust, security, and privacy of current cloud-native applications. Fourthly, to apply predictive modelling techniques to identify expected best practices, and software artifacts across the SDLC., The final two phases are to apply actionable analytics which can be in the form of fault-tolerant services and composing new services in real-time.

Software Engineering Analytics applications (Ramachandran and Mahmood 2020) include reuse of requirements using process mining techniques, use of bug prediction models to extract best practices, and to reuse of requirements, design, and testing efforts. This is one of the log-term research areas of software engineering analytics and other applications include autonomically collecting key practices metrics for Software Process Improvement (SPI) for maturity models such as CMMI which often takes years to collect software development and product quality data manually.

REFERENCES:

Yang, Y. et al (2018) Actionable Analytics for Software Engineering, Actionable Analytics, Guest editors Introduction to Special Issue on Actionable Analytics for SE, IEEE Software, Jan/Feb 2018

Menzies, T and Zimmermann, T (2013) Software Analytics: So What?," IEEE Software, vol. 30, no. 4, 2013

Ozkaya, I (2020) What Should a Software Engineer Know?, IEEE Software, Jan/Feb 2020

Ramachandran, M., & Mahmood, Z. (2020). Software engineering in the era of cloud computing. [electronic resource]: Springer.

Ramachandran, M (2019) Big Data Software Engineering vs Software Engineering for Data Science, 5th International Conference on Data Science, Belgrade, Serbia, https://datasciconference.com/, https://twitter.com/datasciconf/status/1167388467946577921, 18-21 November 2019, Belgrade, Hyatt Regency

PROMISE (2020) http://promise.site.uottawa.ca/SERepository/, accessed on 18th June.

Erdogmus, H., Medvidovic, N., and Paulisch, F (2018) 50 Years of Software Engineering, SI, IEEE Software, September/October 2018