Kashvi: A Framework for Software Process Intelligence

Ashish Sureka
IIIT-Delhi, India
ashish@iiitd.ac.in

Atul Kumar
Siemens Research, India
kumar.atul@siemens.com

Girish Maskeri Rama
Infosys Labs, India
Girish_Rama@infosys.com

ABSTRACT

Software Process Intelligence (SPI) is an emerging and evolving discipline involving mining and analysis of software processes. This is modeled on the lines of Business Process Intelligence (BPI), but with the focus on software processes and its applicability in software systems. Process mining consists of mining event log and process trace data for the purpose of process discovery (run-time process model), process verification or compliance checking (comparison between design-time and run-time process model), process enhancement and recommendation. Software Process Mining or Intelligence is a new and emerging discipline which falls at the intersection of Software Process & Mining, and Software & Process Mining. Software Process Mining is integral to discovering and verifying the processes in a software system.

Software Process Mining is a three word phrase which can be viewed from two perspectives: Software + Process Mining and Software Process + Mining. Software development and evolution involves usage of several workflow management and information systems and tools such as Issue Tracking Systems (ITS), Version Control Systems (VCS), Peer Code Review Systems (PCR) and Continuous Integration Tools (CIT). Such information systems log data consisting of events, activities, time-stamp, user or actor and context specific data. Such events or trace data generated by information systems used during software construction (as part of the software development process) contains valuable information which can be mined for gaining useful insights and actionable information. In this paper, we present *Kashvi*: A Framework for Software Process Intelligence

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining

Keywords

Automated Software Engineering, Business Process Intelligence (BPI), Mining Software Repositories, Process Mining, Software Process Intelligence

The 20th International Conference on Management of Data (COMAD),
17th-19th Dec 2014 at Hyderabad, India.

1. PROCESS MINING

Process mining is an area at the intersection of business process intelligence and data mining consisting of mining event logs from process aware information systems for the purpose of process discovery, process performance analysis, conformance verification, process improvement and organizational analysis. The approaches and algorithms within process mining enables information extraction from event logs or traces generated as a result of execution of a business process [7][8]. An audit trails of a workflow management system within a health-care organization (Hospital Information Management System) can be used to discover models describing processes and organizations. Similarly, the transaction logs of an enterprise resource planning system within a manufacturing unit can be used to discover models describing processes which can be used for process conformance and verification [7][8]. The event logs consists of several events. Each event in the event-log refers to an activity which is a well-defined step within the business process. Each event also refers to a case or trace (i.e., a process instance). Each event can have a performer also referred to as originator (the actor executing or initiating the activity) and events have a timestamp. The events in the event-logs are totally ordered [7][8].

ProM[1] (an abbreviation for Process Mining framework) is a Free and Open Source tool as well as framework for process mining algorithms. ProM provides a usable and scalable platform to process analysts and developers of the process mining algorithms. The architecture of ProM is such that it is easy to extend using plug-ins. ProM consists of several types of plug-ins. Mining plug-ins which implement mining algorithm to construct a Petri-Net based on an event log. Import and Export plug-ins, Analysis plug-ins and conversion plug-ins (which implement conversions between different data formats, e.g., from EPCs to Petri-Nets)

2. MINING SOFTWARE REPOSITORIES

Large and complex software projects use defect tracking systems for managing the workflow of bug reporting, archiving, triaging and tracking. Version control or source code control systems are used to manage changes to project files and documents. Peer code review systems are used to manage peer review of source code before committing the

[1] http://www.processmining.org/prom/start

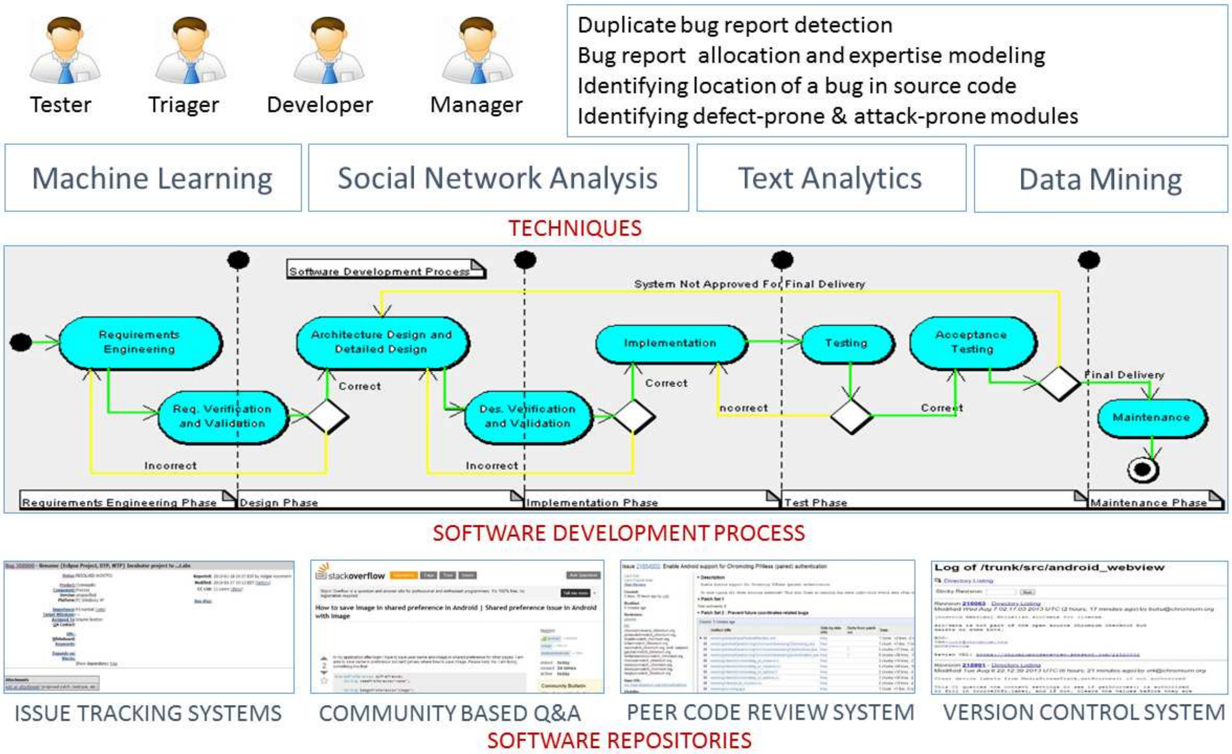
Tester
Triager
Developer
Manager
Duplicate bug report detection
Bug report allocation and expertise modeling
Identifying location of a bug in source code
Identifying defect-prone & attack-prone modules
Machine Learning
Social Network Analysis
Text Analytics
Data Mining
TECHNIQUES
Software Development Process
System Not Approved For Final Delivery
Requirements Engineering
Req. Verification and Validation
Correct
Incorrect
Architecture Design and Detailed Design
Des. Verification and Validation
Correct
Incorrect
Implementation
Testing
Incorrect
Correct
Acceptance Testing
Final Delivery
Maintenance
Requirements Engineering Phase
Design Phase
Implementation Phase
Test Phase
Maintenance Phase
SOFTWARE DEVELOPMENT PROCESS
stackoverflow
How to save image in shared preference in Android | Shared preference issue in Android with image
Log of /trunk/src/android_webview
ISSUE TRACKING SYSTEMS
COMMUNITY BASED Q&A
PEER CODE REVIEW SYSTEM
VERSION CONTROL SYSTEM
SOFTWARE REPOSITORIES

Figure 1: Kashvi: A Framework for Software Process Intelligence. Figure showing the Software Repositories, Data getting generated during Construction of Software, Mining Techniques, Practitioners and Problems Encountered by Practitioners

source code to identify defects though inspection. Community based Q&A websites for programmers and online forums are widely used by developers for asking questions and sharing knowledge. Bug databases, version archives, source code repository, peer code review system, community based Q&A websites, mailing lists and online forums for programmers are software repositories containing large volumes of valuable structured data and unstructured data (free-form text) entered by developers during the software development process. For example, a bug report typically contains information describing the problem, application environment, steps to reproduce and stack trace. A source control system contains information regarding the files that were revised, the changes that were made, developer who made the change, developer comments and time-stamp.

These repositories have been primarily serving the purpose of archiving information or recording keeping. Mining Software Repositories (MSR) researchers have investigated social network analysis, data mining, machine learning and information retrieval based approaches to analyze software repositories to uncover interesting patterns and knowledge which can be used to support developers in the process of software maintenance. The work on Mining Software Repositories is based on the premise that historical data present in software repositories can be mined to derive actionable information resulting in increased productivity and effectiveness of developers [1][9]. Researchers have also conducted field studies and survey of practitioners to understand problems encountered by them and developed mining software repositories based solutions to address the problems encountered by developers and project teams [1][9]. Some of the general themes[2] within MSR are: analysis of software ecosystems and mining of repositories across multiple projects, models for social and development processes that occur in large software projects, prediction of future software qualities via analysis of software repositories, models of software project evolution based on historical repository data, characterization, classification, and prediction of software defects based on analysis of software repositories, techniques to model reliability and defect occurrences, search-driven software development, including search techniques to assist developers in finding suitable components and code fragments for reuse, and software search engines, analysis of change patterns and trends to assist in future development and Visualization techniques and models of mined data [1][9].

3. SOFTWARE PROCESS INTELLIGENCE

Software Process Intelligence (SPI) is an emerging and evolving discipline involving mining and analysis of software processes. This is modeled on the lines of application of Business Intelligence techniques to business processes (Business Process Intelligence (BPI)), but with the focus on software processes and its applicability in software engineering and information technology systems. Software Process Mining or falls at the intersection of Software Process & Mining, and Software & Process Mining. It is a three word phrase which can be viewed from two perspectives: Software + Process Mining and Software Process + Mining. Software

[2] http://2015.msrconf.org/

development and evolution involves usage of several workflow management and information systems and tools such as Issue Tracking Systems (ITS), Version Control Systems (VCS), Peer Code Review Systems (PCR) and Continuous Integration Tools (CIT). Such information systems log data consisting of events, activities, time-stamp, user or actor and context specific data. Such events or trace data generated by information systems used during software construction (as part of the software development process) contains valuable information which can be mined for gaining useful insights and actionable information [5][6].

Figure 1 illustrates the broad framework for Software Process Intelligence. As shown in Figure, the framework consists of software repositories (version control system, issue tracking system, peer code review system, community based Q&A websites, source code repositories and developer mailing lists) containing data generated as part of constructing a software. Figure 1 shows the complete software development process: requirements engineering, design, implementation, test and maintenance. Software Process Intelligence consists of applying machine learning, information retrieval, social network analysis, text analytics and data mining based techniques on the software engineering data to extract actionable information aimed at solving problems encountered by practitioners. Figure 1 shows the practitioners (tester, triager, developer, project manager, quality assurance manager, requirements engineer) and some of the technical problems (defect prediction, identifying fault-prone entities, bug localization, automatic bug triaging, bug report allocation and expertise modeling)

Software Process Intelligence has diverse applications and is an area that has recently attracted several researcher's attention due to availability of vast data generated during software development. Some of the business applications of process mining software repositories are: uncovering runtime process model, discovering process inefficiencies and inconsistencies, observing project key indicators and computing correlation between product and process metrics, extracting general visual process patterns for effort estimation and analyzing problem resolution activities [2][3][5][6]. Some of the themes within Software Process Intelligence are: Big-Data and scalability issues in software process intelligence, Integration of agile development methods and process mining, Metrics for software process intelligence, Predictive analysis using process mining results, Privacy and confidentiality aspects in software process intelligence, Process mining for software process assessment and improvement, Program workflow mining, Relationship between effect of software process intelligence and organizational performance, Software process intelligence tool support, Software process intelligence in small and medium scale enterprises, Software quality and use of software process intelligence, Techniques to monitor software processes, Visualization in software processes, Visualization of software process mining and/or conformance results.

Mittal et al. present an approach for mining the process data (process mining) from software repositories archiving data generated as a result of constructing software by student teams in an educational setting [4]. They present an application of mining three software repositories: team wiki (used during requirement engineering), version control system (development and maintenance) and issue tracking system (corrective and adaptive maintenance) in the context of an undergraduate Software Engineering course [4]. Gupta et al. present an application of process mining three software repositories (ITS, PCR and VCS) from control flow and organizational perspective for effective process management [3]. They discover runtime process model for bug resolution process spanning three repositories using process mining tool, Disco, and conduct process performance and efficiency analysis. They identify bottlenecks, define and detect basic and composite anti-patterns. In addition to control flow analysis, they mine event log to perform organizational analysis and discover metrics such as handover of work, subcontracting, joint cases and joint activities [3]. Gupta et al. apply business process mining tools and techniques to analyze the event log data (bug report history) generated by an issue tracking system with the objective of discovering runtime process maps, inefficiencies and inconsistencies. They conduct a case-study on data extracted from Bugzilla issue tracking system of the popular open-source Firefox browser project [2].

4. REFERENCES

[1] Msr 2014: Proceedings of the 11th working conference on mining software repositories. 2014.

[2] M. Gupta and A. Sureka. Nirikshan: Mining bug report history for discovering process maps, inefficiencies and inconsistencies. In *Proceedings of the 7th India Software Engineering Conference*, ISEC '14, pages 1:1–1:10, 2014.

[3] M. Gupta, A. Sureka, and S. Padmanabhuni. Process mining multiple repositories for software defect resolution from control and organizational perspective. In *Proceedings of the 11th Working Conference on Mining Software Repositories*, MSR 2014, pages 122–131, 2014.

[4] M. Mittal and A. Sureka. Process mining software repositories from student projects in an undergraduate software engineering course. In *Companion Proceedings of the 36th International Conference on Software Engineering*, ICSE Companion 2014, pages 344–353, 2014.

[5] W. Poncin, A. Serebrenik, and M. van den Brand. Process mining software repositories. In *Software Maintenance and Reengineering (CSMR), 2011 15th European Conference on*, pages 5–14, March 2011.

[6] V. Rubin, C. W. Günther, W. M. P. Van Der Aalst, E. Kindler, B. F. Van Dongen, and W. Schäfer. Process mining framework for software processes. In *Proceedings of the 2007 International Conference on Software Process*, ICSP'07, pages 169–181, 2007.

[7] W. van der Aalst, T. Weijters, and L. Maruster. Workflow mining: Discovering process models from event logs. *IEEE Trans. on Knowl. and Data Eng.*, 16(9):1128–1142, Sept. 2004.

[8] W. M. P. van der Aalst. *Process Mining: Discovery, Conformance and Enhancement of Business Processes.* Springer Publishing Company, Incorporated, 1st edition, 2011.

[9] T. Zimmermann, M. D. Penta, and S. Kim. Proceedings of the 10th working conference on mining software repositories, msr '13, san francisco, ca, usa, may 18-19, 2013. 2013.