Nirikshan: Process Mining Software Repositories to Identify Inefficiencies, Imperfections, and Enhance Existing Process Capabilities

Monika Gupta
monikag@iiitd.ac.in

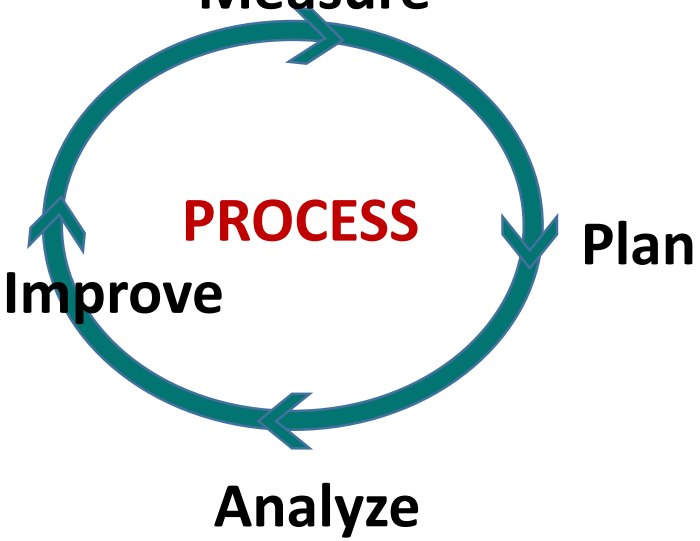
PhD Advisor: Dr. Ashish Sureka
Industry Mentor: Dr. Srinivas Padmanabhuni

Indraprastha Institute of Information Technology
New Delhi, India

Presentation Outline

- Research Motivation
- Research Aim
- Related Work
- Related Methodology and Proposed Contributions
  - Data Source
  - Technique and Contributions
  - Evaluation
- Preliminary Results

Research Motivation



Process Analyst

Process Mining

Software Repositories

"If one cannot measure it,
one cannot improve it"

Research Motivation

Process Mining:

- Extract knowledge from event logs recorded by an information system [1].
- Event logs (e.g. transaction logs) with four fields:

CaseID	Event	Timestamp	Resource
---	---	---	---

- Tools and framework for process mining:
  - ProM [1] (Open Source)
  - Disco [2] (Commercial)

[1] van der Aalst, Wil MP, et al. "Business process mining: An industrial application." *Information Systems* 32.5 (2007): 713-732.

1. http://www.promtools.org/prom6/
2. http://www.fluxicon.com/disco/

Research Motivation

Software Repositories:

- Artifacts generated by the tools during software evolution and archived for future reference.
- Rich data available.
- Uncover interesting and actionable information for process improvement.

For example :
Issue Tracking System (ITS) ,
Version Control System,
Code Review etc.

Research Aim

Novel applications of process mining on software repositories for:

- Runtime Process Map Discovery
- Performance Analysis
- Conformance verification
- User behavior pattern investigation
- Enhancement of process mining capabilities of existing tools

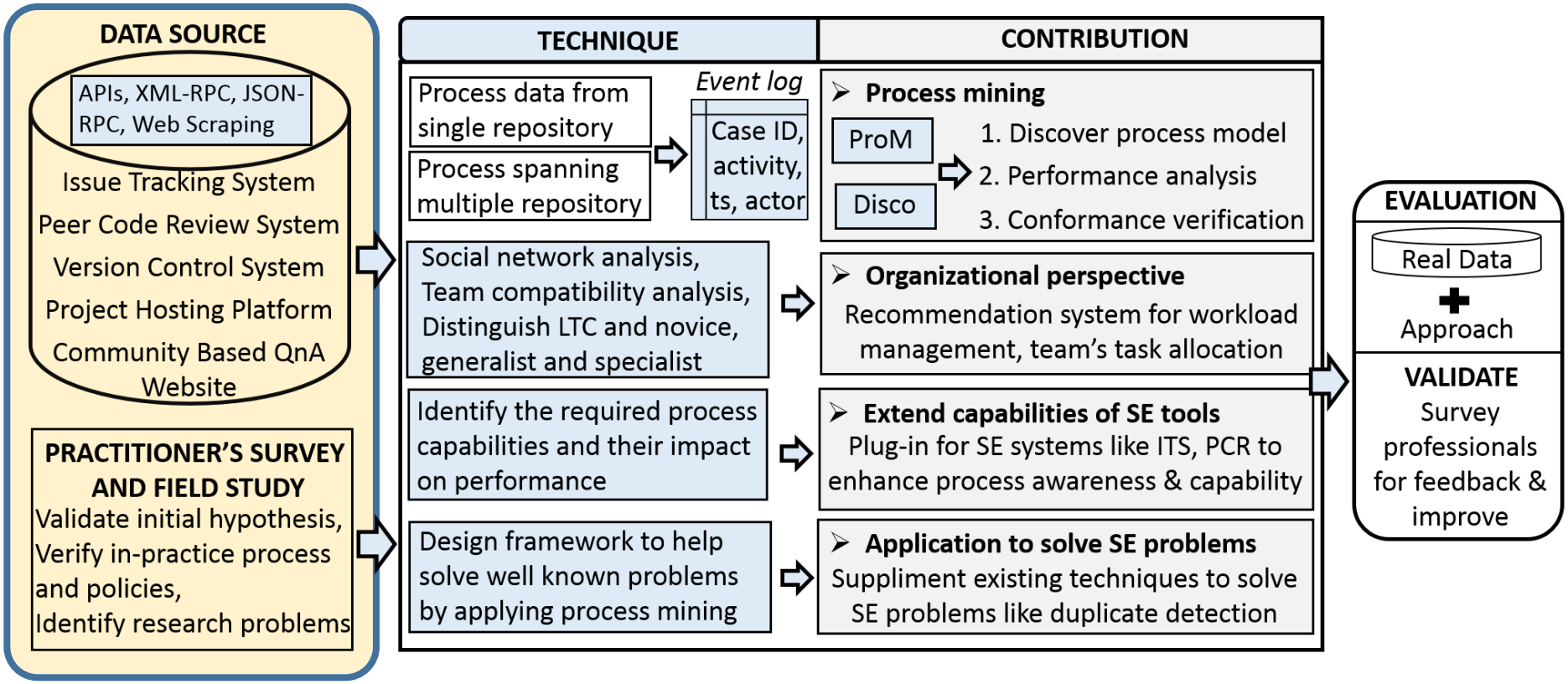
Related Work

Author	Year	Repository	Objectives
Rubin *et al.*	2007	Subversion logs of the ArgoUML project. (OSS)	Linear Temporal Logic (LTL) Checking (Conformance Analysis), Social Network Discovery, Performance Analysis, Petri Net Discovery
Akman *et al.*	2009	Software Configuration Management of industry project.	Analyzed and compared the effectiveness of four process discovery algorithms on software process. Analyzed discrepancies between real time and design time process.
Knab *et al.*	2010	ITS of EUREKA project SERIOUS (CSS).	Interactive approach to visualize effort estimation and process lifecycle patterns in ITS to detect outliers, flaws and interesting properties.
Poncin *et al.*	2011	aMSN and GCC bug repositories, mail archives, SVM	Combined different repositories for analysis using a prototype, FRASR. Role classification and Bug life cycle construction using ProM.
Sunindyo *et al.*	2012	Red Hat Linux ITS	Framework for collecting and analyzing data from bug reporting system, conformance checking to improve process quality.

Technical Challenges

- Mining from multiple perspectives
- Gathering data from heterogeneous sources
- Data incompatibility with process mining tools
- Mining hidden tasks
- Missing clear design process and goals

Research Methodology



NIRIKSHAN (Sanskrit word which means 'to investigate'):
Research framework showing proposed research approach for the research contributions followed by evaluation.

Research Methodology

PRACTITIONER'S SURVEY AND FIELD STUDY

Validate initial hypothesis

Verify in-practice process and policies

Identify research problems

Research Questions that usually process analysts have and can be answered by process mining

DATA SOURCE

Issue Tracking System like Bugzilla, JIRA, Mantis

Peer Code Review System like Gerrit, Rietveld

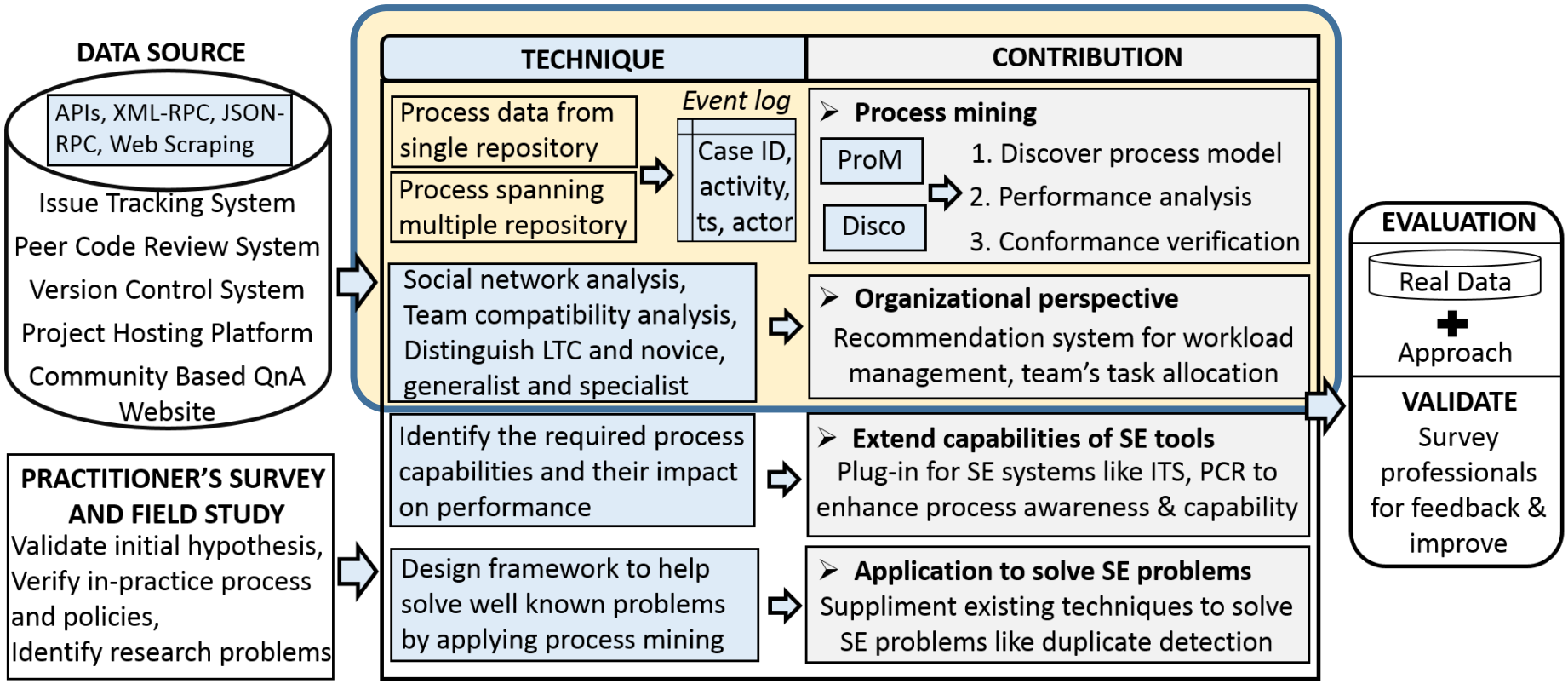
Version Control System like SVN and Mercurial

Project Hosting Platform like Sorceforge, Git

APIs, XML-RPC, JSON-RPC, Web Scraping

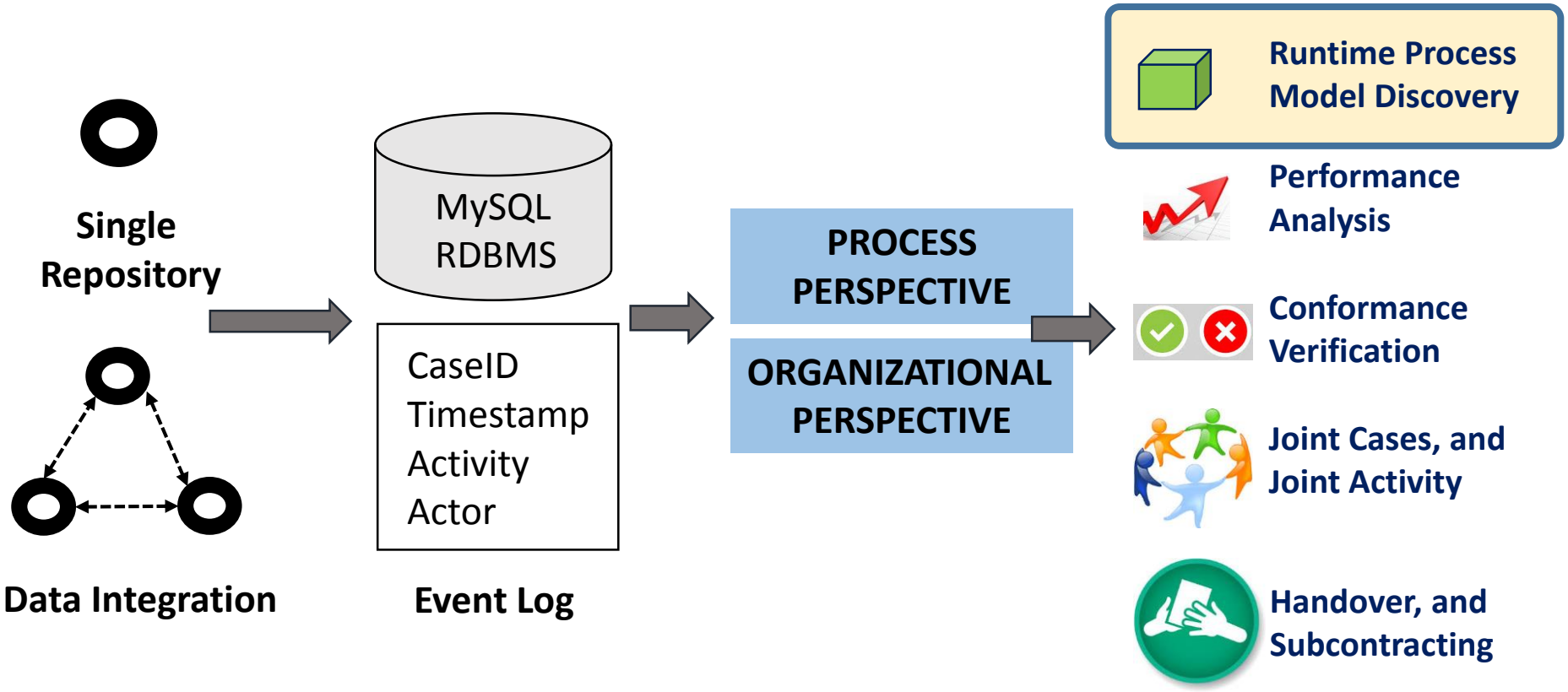
Event Log Data

Research Methodology

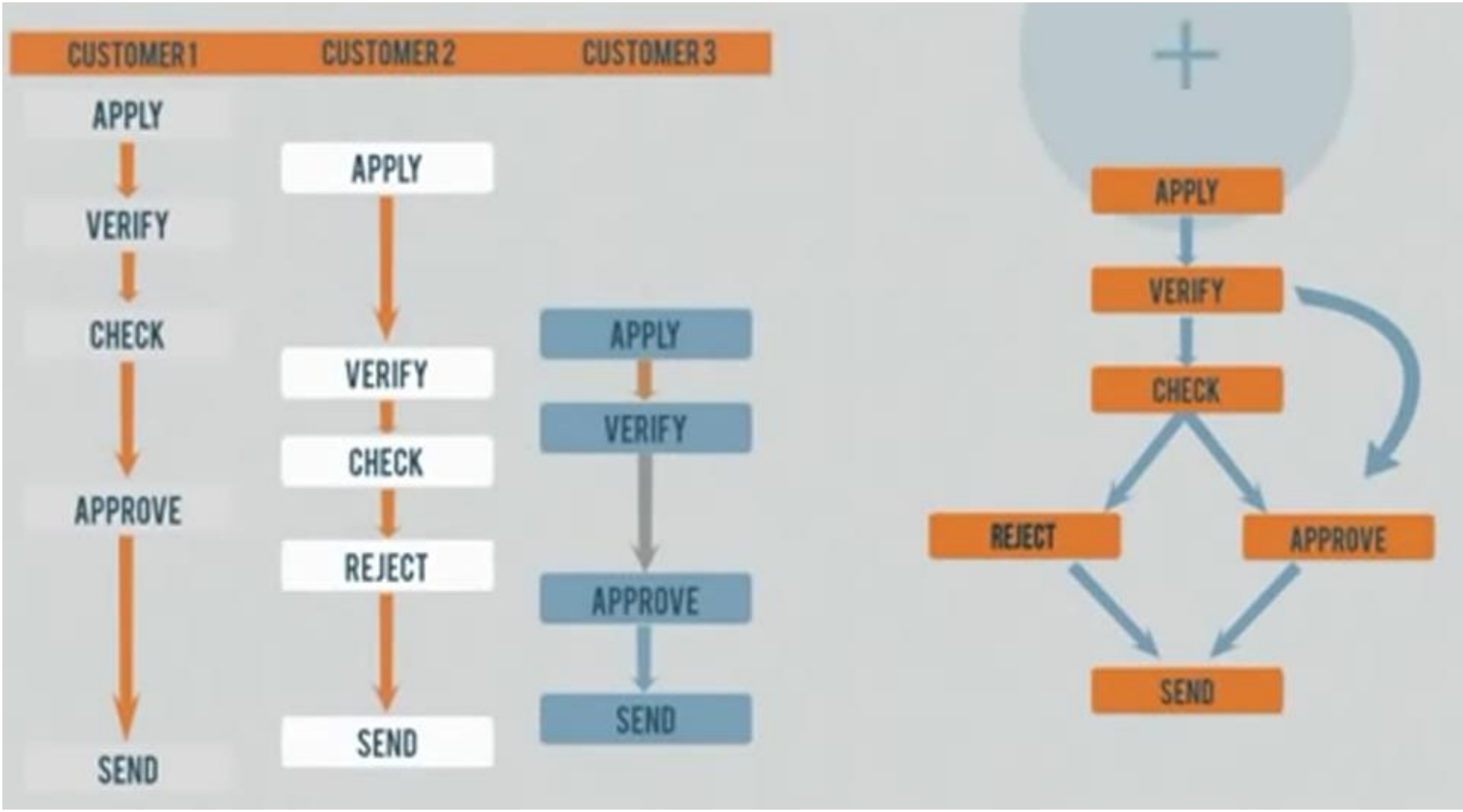


NIRIKSHAN (Sanskrit word which means 'to investigate'):
Research framework showing proposed research approach for the research contributions followed by evaluation.

Research Methodology

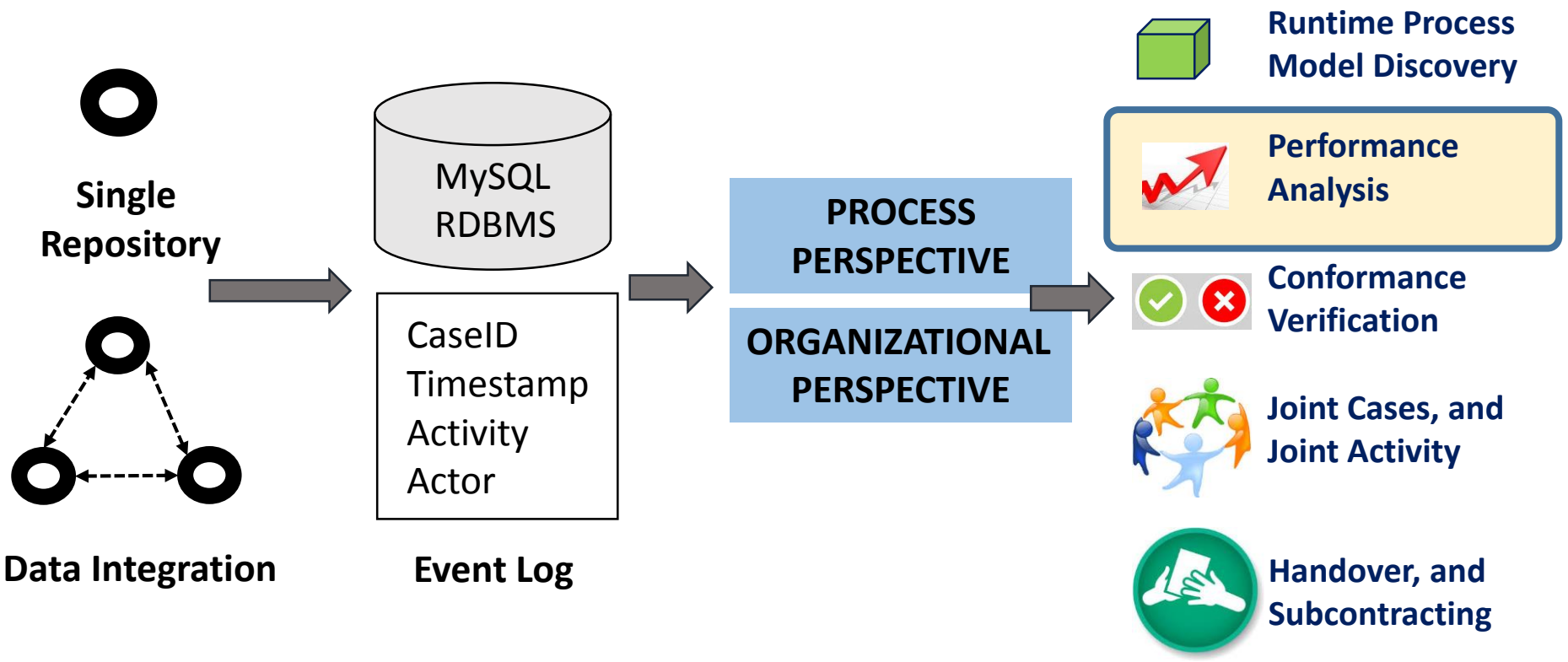


Research Methodology



Research Methodology

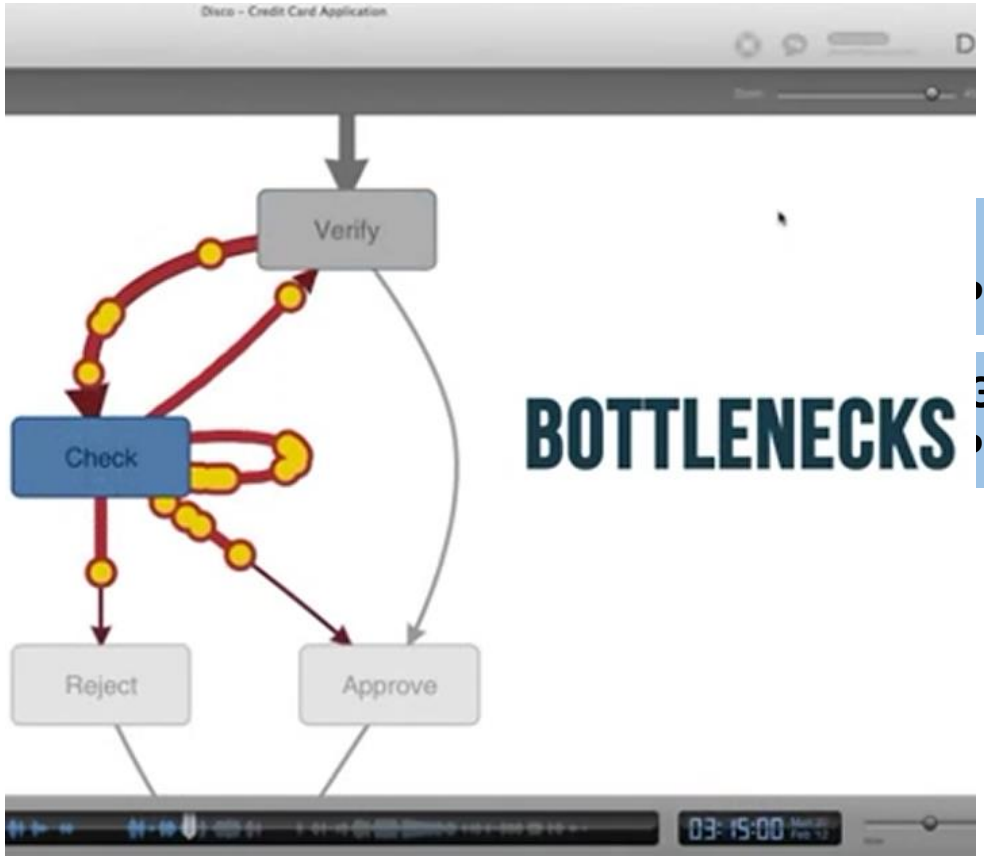
Technique And Contribution

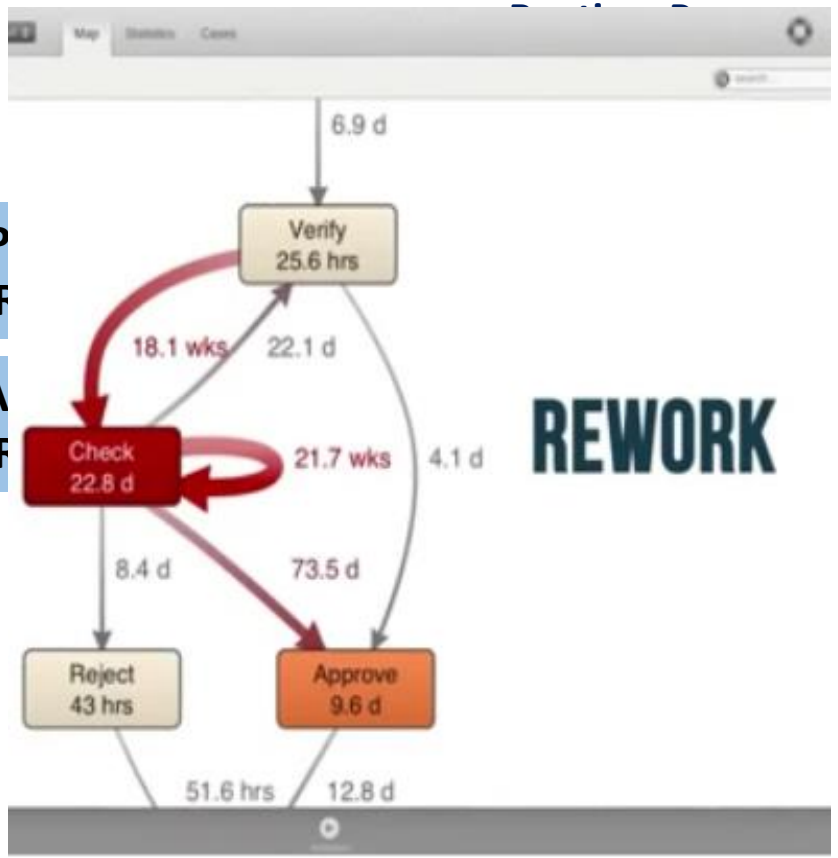


[2] Eid-Sabbagh et al. "Business process architecture: use and correctness." *Business Process Management*. Springer Berlin Heidelberg, 2012.

Research Methodology

Technique And Contribution

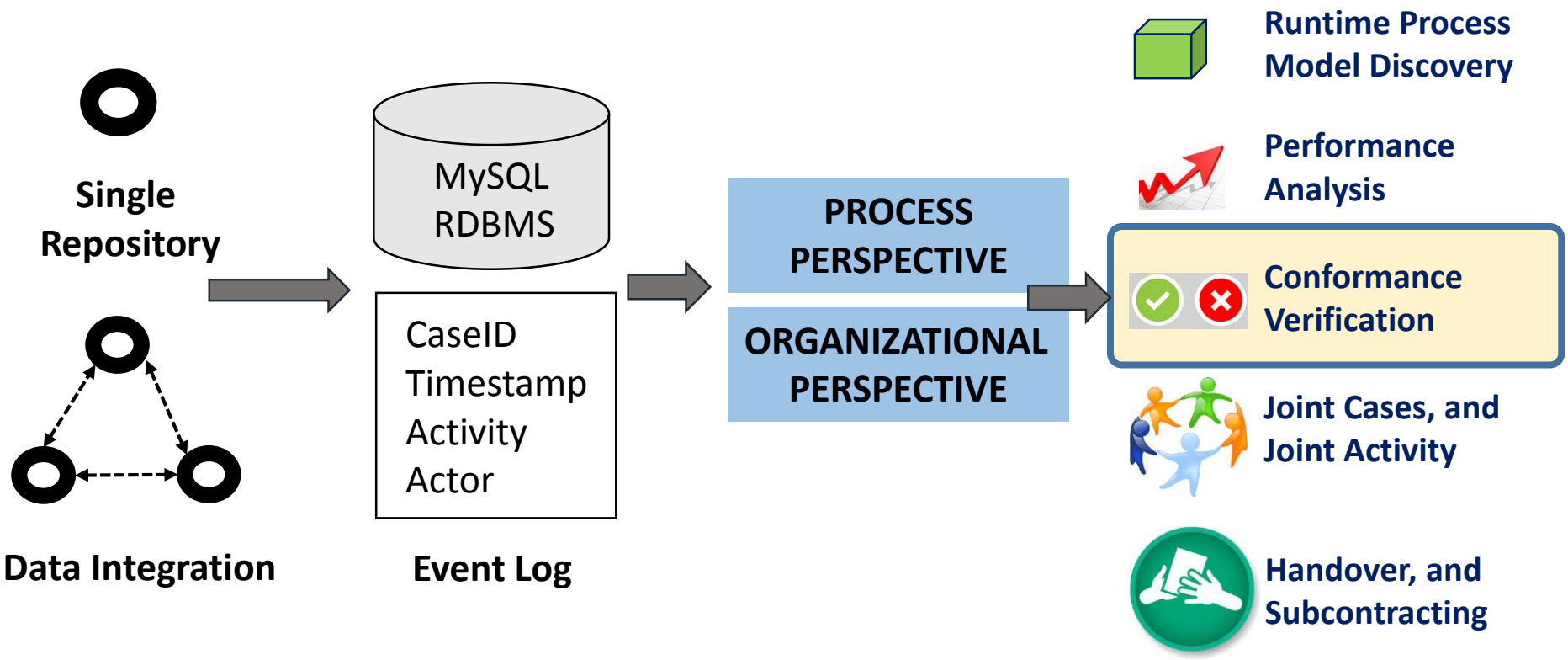




[2] Eid-Sabbagh et al. "Business process architecture: use and correctness." *Business Process Management*. Springer Berlin Heidelberg, 2012.

Research Methodology

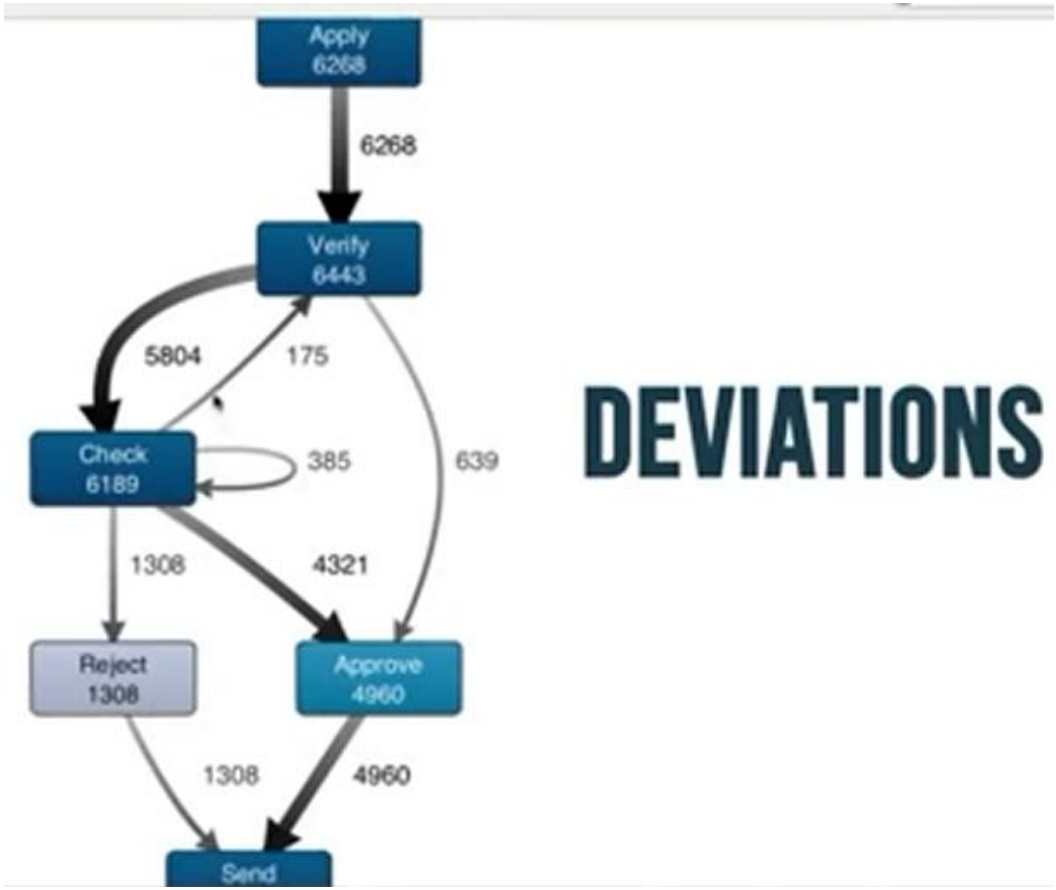
Technique And Contribution



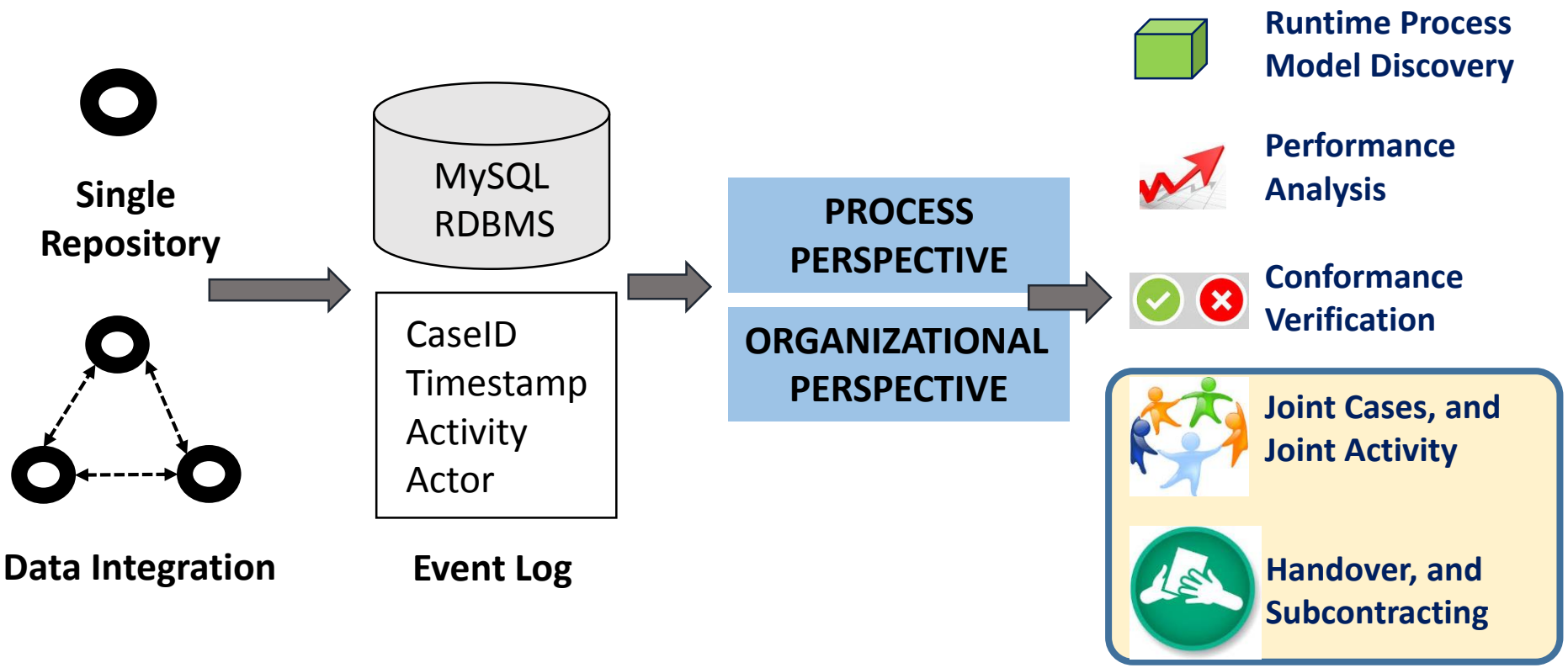
Conformance Verification

"Do we do what was agreed upon?"

Design Process Model v/s Discovered Runtime Process Model

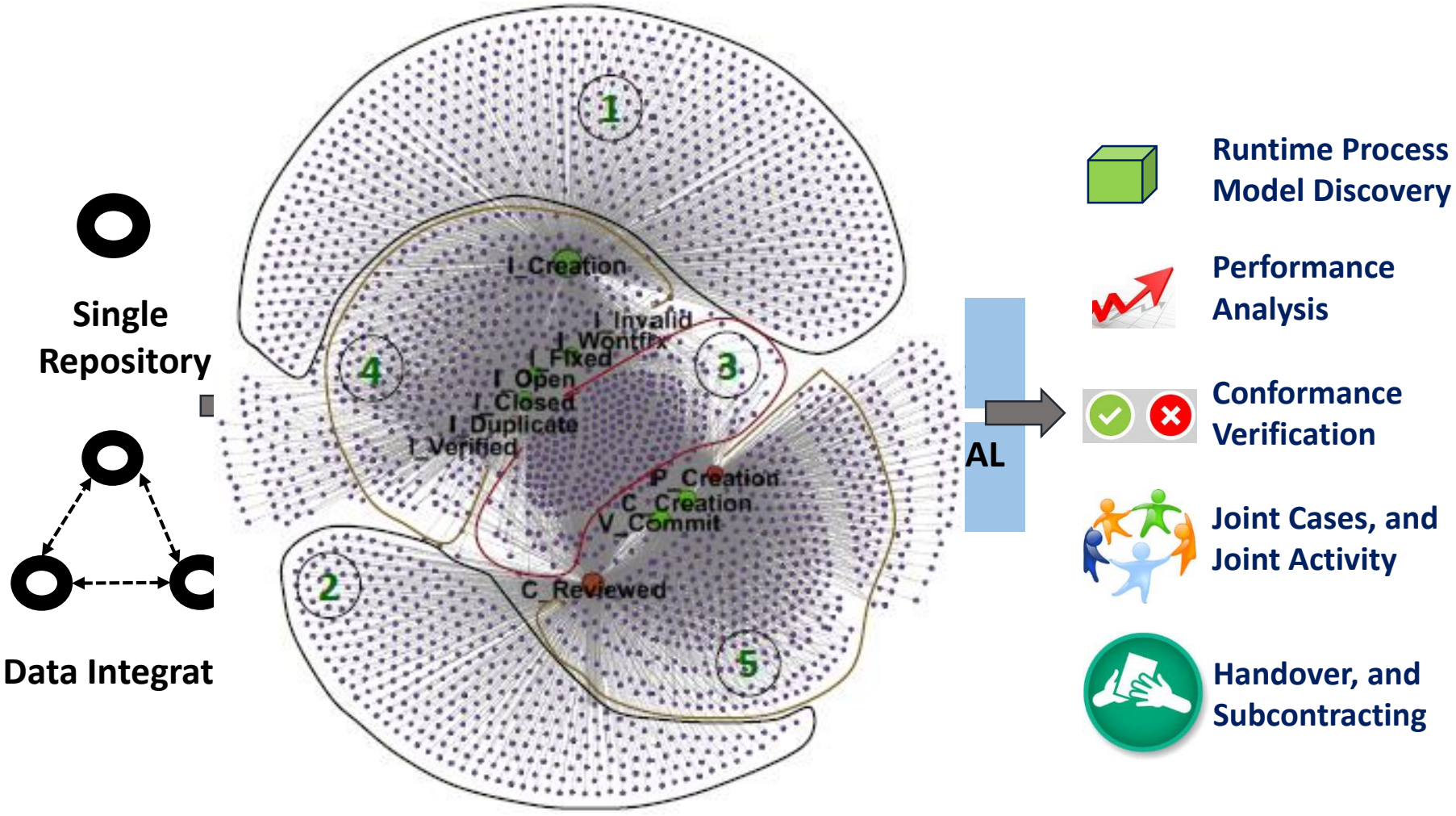


Research Methodology



[3] Van Der Aalst, Wil MP, Hajo A. Reijers, and Minseok Song. "Discovering social networks from event logs.", *CSCW* 2005, 549-593.

Research Methodology



[3] Van Der Aalst, Wil MP, Hajo A. Reijers, and Minseok Song. "Discovering social networks from event logs.", *CSCW* 2005, 549-593.

Preliminary Results

Process Mining Single Software Repository: Issue Tracking System [4]

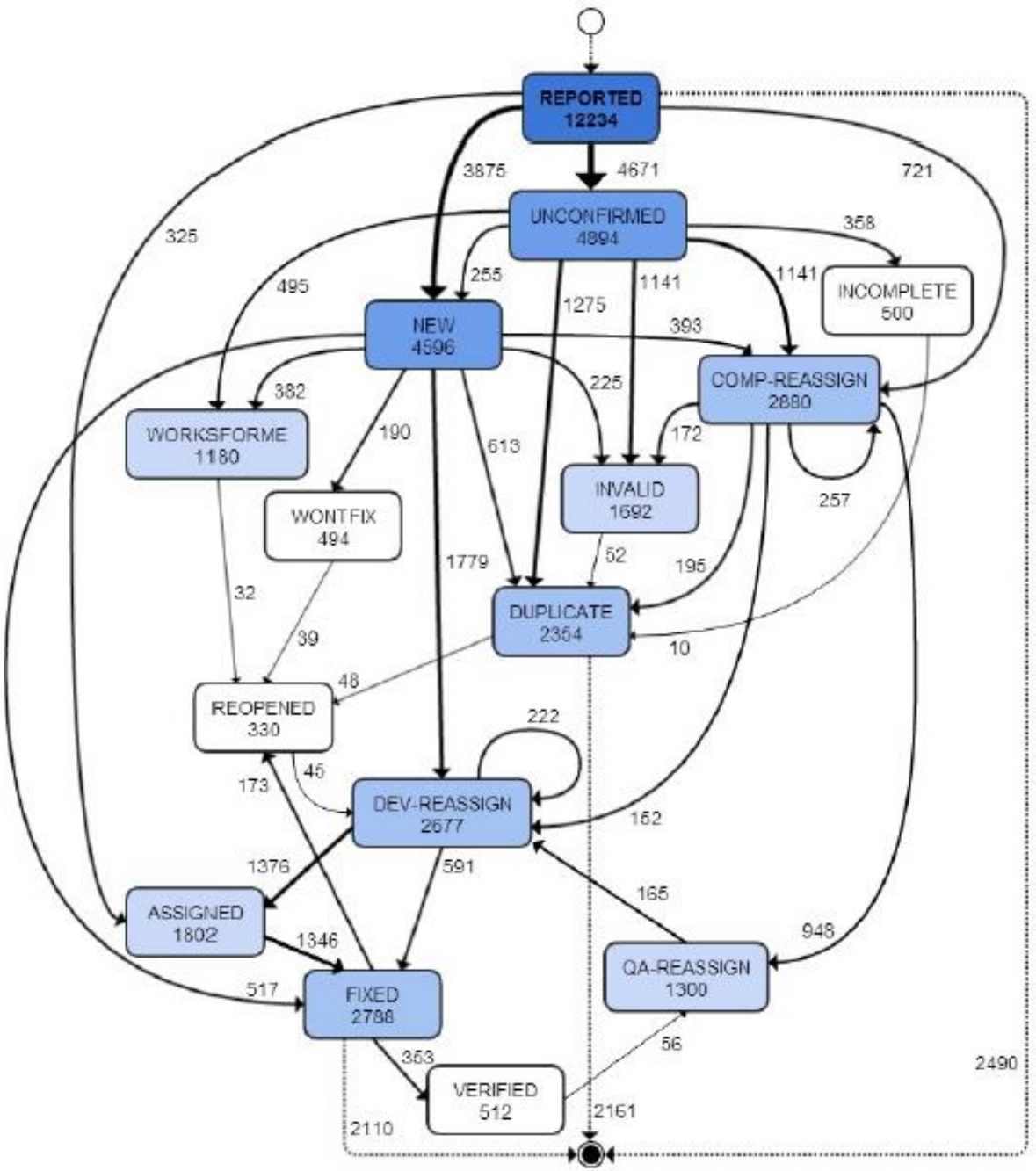
- Event log data of ITS analyzed.
- Case study on open source Firefox browser and Core project.
- Discover process map, self-loops, back-forth, issue reopen.
- Algorithm for conformance verification.

Insights:

- 15 activities in event log
- Component and developer reassignment frequent
- 3-4 events in lifecycle, few with >7 events
- 80% of the cases covered with 2% unique traces
- Total unique traces: Core-1164, Firefox-622

[4] Gupta, M., & Sureka, A. Nirikshan: Mining bug report history for discovering process maps, inefficiencies and inconsistencies. In *seventh india software engineering conference (ISEC)*.

lts



Process Map for Firefox with labels as absolute frequency of transition and activity

enth india software engineering conference (ISEC).

Preliminary Results

Bottleneck:

- *Worksforme, Wontfix* state

Reopened:

- *Worksforme, Wontfix, Fixed*

Conformance:

- 0.86 for Core, 0.91 for Firefox
- Cause of inconsistency:

  Reported ⟶ *Assigned*

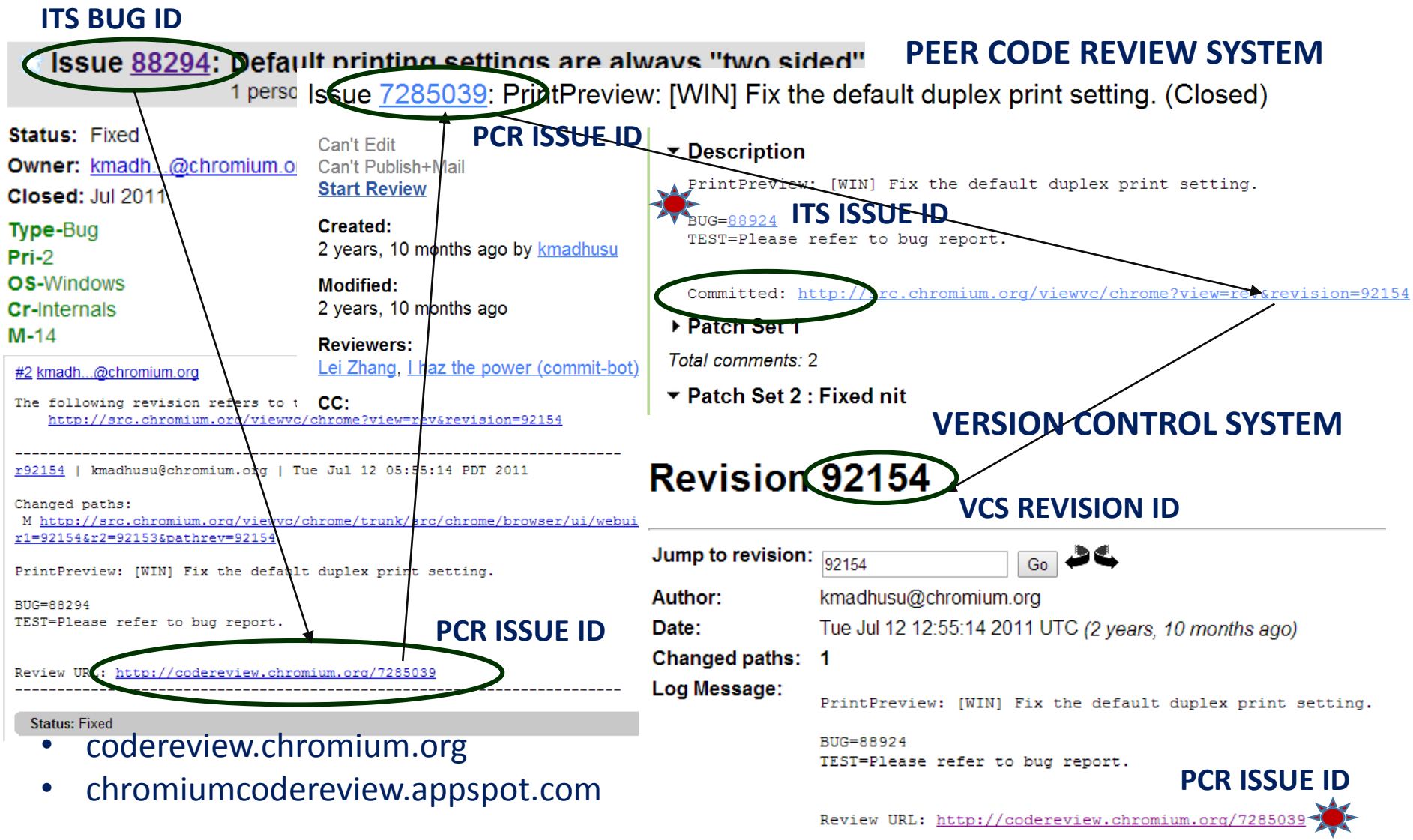
Preliminary Results

Process Mining Multiple Software Repositories: Issue Tracking System, Peer Code Review System and Version Control System [5]

- Issue resolution process starting from issue reporting in ITS, followed by patch submission to PCR for review and finally committed to VCS.
- Case study on open source Google Chromium project.
- *Discover process map, anti-patterns, bottlenecks, organizational metrics.*

[5] Gupta, Monika, Ashish Sureka, and Srinivas Padmanabhuni. "Process mining multiple repositories for software defect resolution from control and organizational perspective." *Proceedings of the 11th Working Conference on Mining Software Repositories*. ACM, 2014.

Preliminary Results

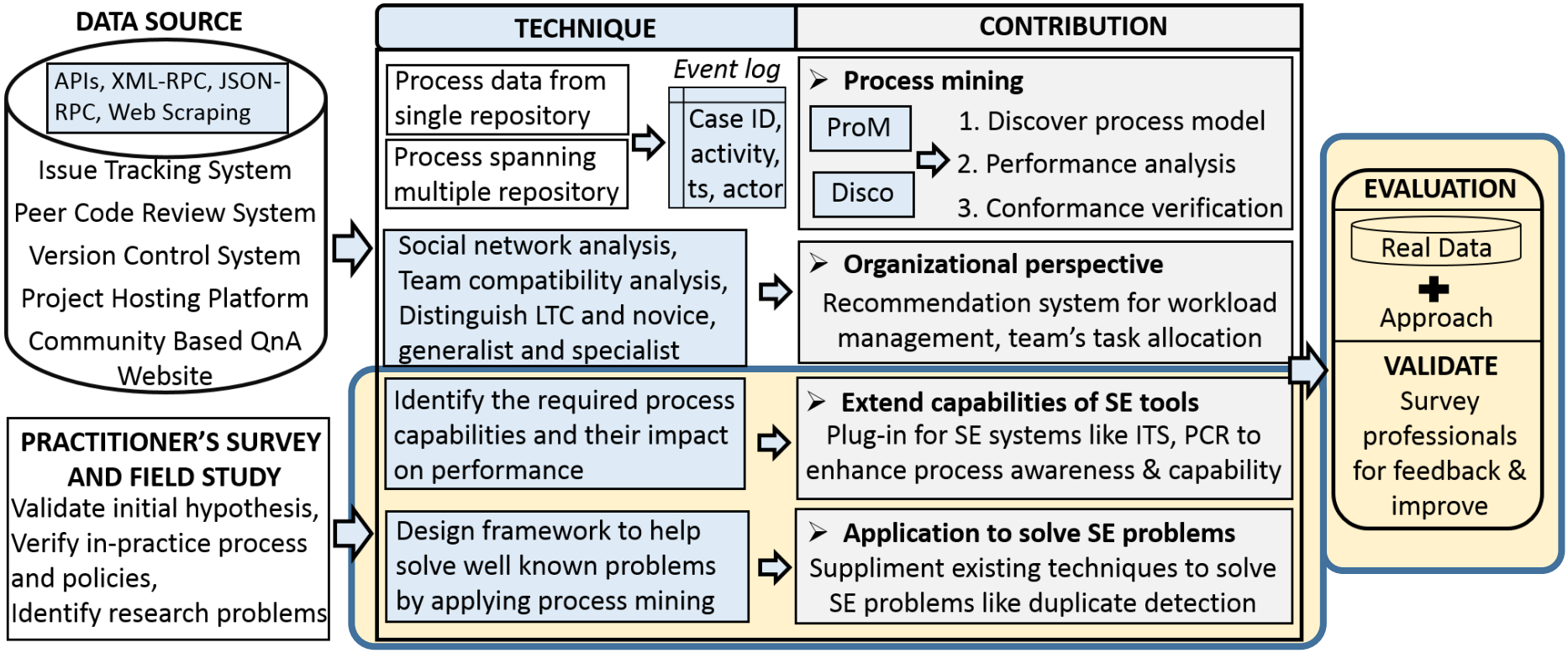


- codereview.chromium.org
- chromiumcodereview.appspot.com

Preliminary Results

- Resolution process efficient with high chances of issues getting *Fixed*
- Basic and Composite *anti-patterns* like loops, and information flow detected
- *Bottlenecks* are identified such as control transfer between ITS and PCR
- More social performers are more *active*
- Joint activities helps to identify *generalists and specialists*
- Same performer performs *multiple subsequent activities*

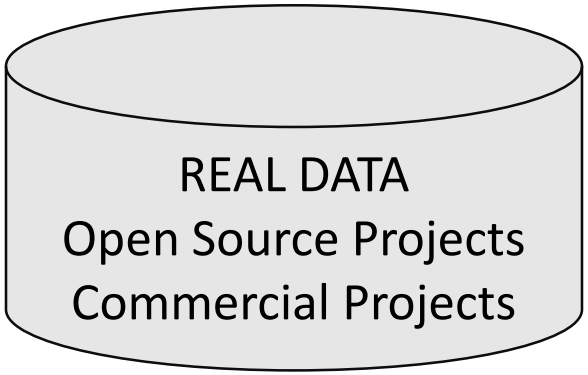
Research Methodology



NIRIKSHAN (Sanskrit word which means 'to investigate'):
Research framework showing proposed research approach for the research contributions followed by evaluation.

Research Methodology

EVALUATION



Process Mining Approach

VALIDATION:
Survey
professionals

Feedback

Improve

Usefulness

- Visualize the process reality for better improvement
- Streamline the process
- Reduce flow time
- Manage changing workloads
- Efficient task allocation
- Better process maintainability, capability, reliability, efficiency and stability [6][7]

6. Patrick Knab, Martin Pinzger, and Harald C. Gall. **Visual patterns in Issue Tracking System**. ICSP 2010, LNCS 6195, pp. 222-233,2010
7. Ekkart Kindler, Vladimir Rubin, Wilhelm Schafer. **Activity mining for Discovering Software Process Models.** In proceeding of: Software Engineering 2006, Fachtagung des GI-Fachbereichs Softwaretechnik, 28.-31.3.2006 in Leipzig.

References

1. Rubin, Vladimir, et al. "Process mining framework for software processes."*Software Process Dynamics and Agility*. Springer Berlin Heidelberg, 2007. 169-181.
2. Akman, Burcu, and O. Demirors. "Applicability of Process Discovery Algorithms for Software Organizations." *Software Engineering and Advanced Applications, 2009. SEAA'09. 35th Euromicro Conference on*. IEEE, 2009.
3. Patrick Knab, Martin Pinzger, and Harald C. Gall. “Visual patterns in Issue Tracking System”. ICSP 2010, LNCS 6195, pp. 222-233,2010
4. Wouter Poncin, Alexander Serebrenik, Mark van den Brand. “Process Mining Software Repositories”. CSMR 2011
5. Sunindyo, Wikan, et al. "Improving Open Source Software Process Quality Based on Defect Data Mining." *Software Quality. Process Automation in Software Development*. Springer Berlin Heidelberg, 2012. 84-102.
6. Ekkart Kindler, Vladimir Rubin, Wilhelm Schafer. “Activity mining for Discovering Software Process Models”. In proceeding of: Software Engineering 2006, Fachtagung des GI-Fachbereichs Softwaretechnik, 28.-31.3.2006 in Leipzig.
7. Günther, Christian W., and Wil MP Van Der Aalst. "Fuzzy mining–adaptive process simplification based on multi-perspective metrics." *Business Process Management*. Springer Berlin Heidelberg, 2007. 328-343.
8. Shihab, Emad, et al. "Predicting re-opened bugs: A case study on the eclipse project." *Reverse Engineering (WCRE), 2010 17th Working Conference on*. IEEE, 2010.
9. Thomas Zimmermann, Nachiappan Nagappan, Philip J. Guo, Brendan Murphy. ”Characterizing and Predicting Which Bugs Get Reopened”. In Proceedings of the 34th International Conference on Software Engineering (ICSE 2012 SEIP Track), Zurich, Switzerland, June 2012.
10. C. A. Halverson, J. B. Ellis, C. Danis, and W. A. Kellogg**. “**Designing task visualizations to support the coordination of work in software development”**.** In Proceedings of the 2006, 20th anniversary conference on Computer supported cooperative work (CSCW 2006), pages 39–48, New York, NY, USA, 2006. ACM Press.

THANK YOU!