Introduction to Software Analytics

Dongmei Zhang

Software Analytics Group

Microsoft Research

November 26, 2014

Outline

- Overview of Software Analytics
- Selected projects
- Experience sharing on Software Analytics in practice

New Era…Software itself is changing…

Software

Services

How people use software is changing…

Individual

Isolated

Not much content generation

Social

Collaborative

Huge amount of artifacts generated anywhere anytime

How software is built & operated is changing…

Code centric	Data pervasive
In-lab testing	Debugging in the large
Experience & gut-feeling	Informed decision making
Centralized development	Distributed development
Long product cycle	Continuous release
…	…

Software Analytics

Software analytics is to enable software practitioners to perform data exploration and analysis in order to obtain insightful and actionable information for data-driven tasks around software and services.

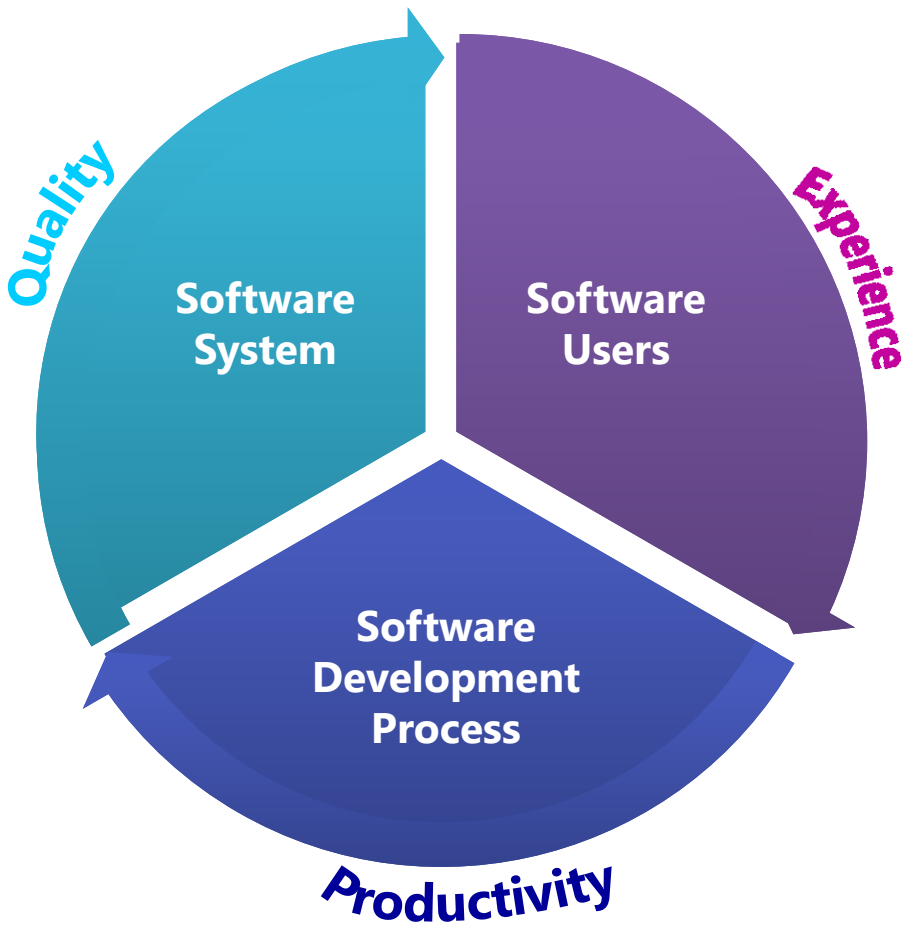
Software Analytics

Software analytics is to enable *software practitioners* to perform data exploration and analysis in order to obtain *insightful and actionable information* for *data-driven tasks* around software and services.

Five dimensions

Research Topics	Target Audience	Output
Technology Pillars	Connection to Practice	

Research topics



- Covering different areas of software domain
- Throughout entire development cycle
- Enabling practitioners to obtain insights

Data sources

Runtime traces

Program logs

System events

Perf counters

…

Usage log

User surveys

Online forum posts

Blog & Twitter

…

Source code

Bug history

Check-in history

Test cases

…

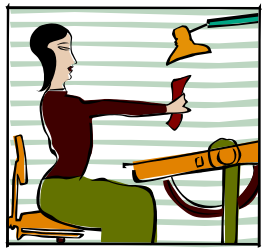
Target audience – software practitioners

Program Manager

Developer

Management personnel

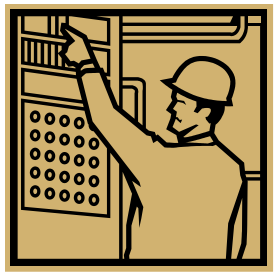
Designer



Tester

Support engineer

Operation engineer



Usability engineer

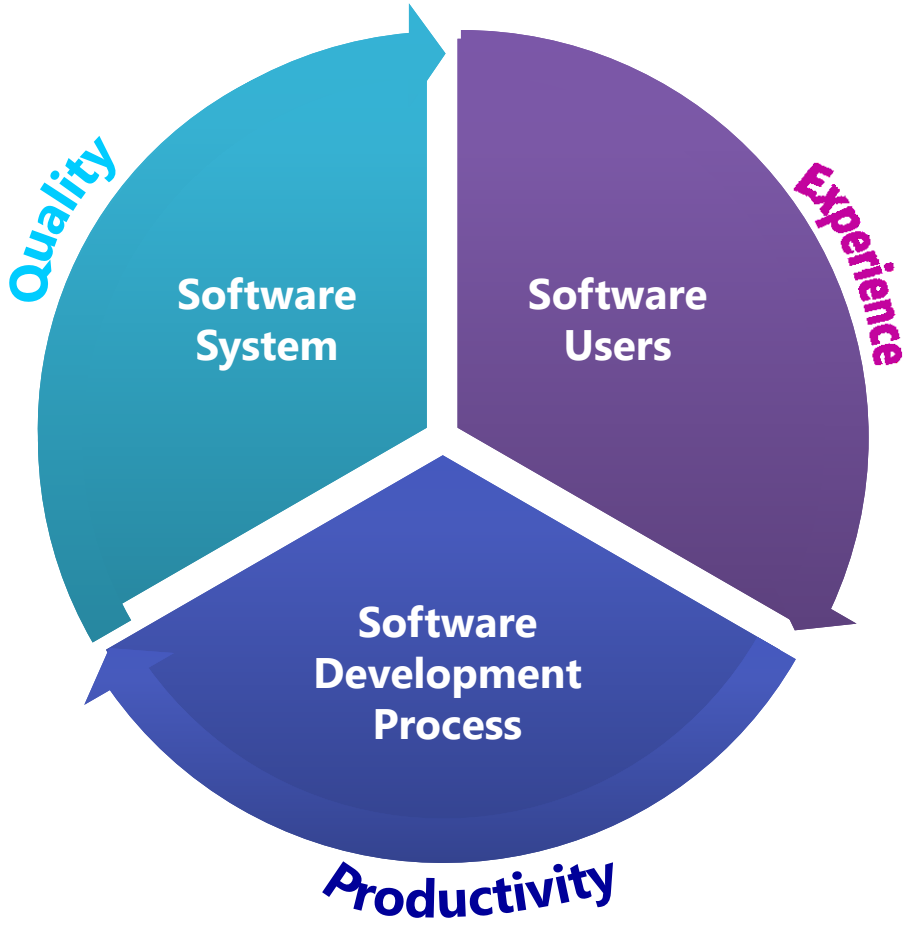
Output – insightful information

- Conveys meaningful and useful understanding or knowledge towards completing the target task
- Not easily attainable via directly investigating raw data without aid of analytics technologies
- Examples
  - It is easy to count the number of re-opened bugs, but how to find out the primary reasons for these re-opened bugs?
  - When the availability of an online service drops below a threshold, how to localize the problem?

Output – actionable information

- Enables software practitioners to come up with concrete solutions towards completing the target task
- Examples
  - Why bugs were re-opened?
    - A list of bug groups each with the same reason of re-opening
  - Why availability of online services dropped?
    - A list of problematic areas with associated confidence values
  - Which part of my code should be refactored?
    - A list of cloned code snippets easily explored from different perspectives

Research topics and technology pillars



Vertical

Horizontal

Information Visualization

Data Analysis Algorithms

Large-scale Computing

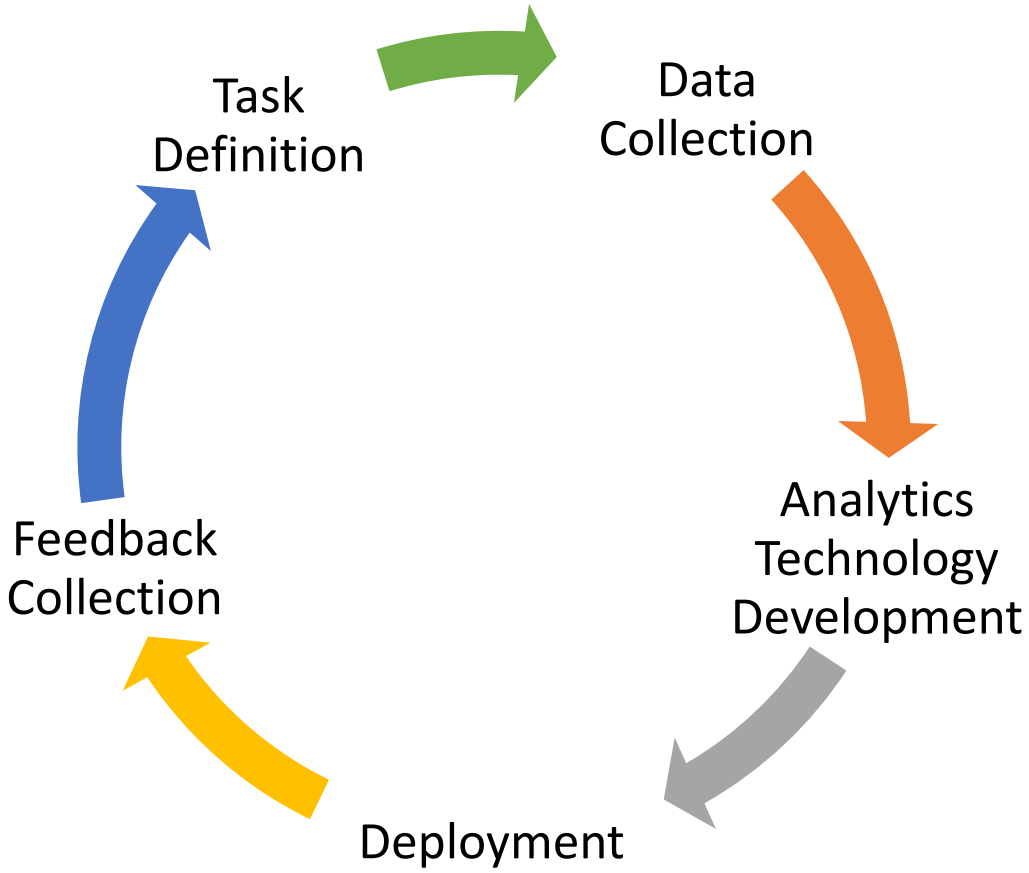
Connection to practice

- Software Analytics is naturally tied with software development practice

- Getting real

Real Data	Real Problems	Real Users	Real Tools

Approach


Task
Definition
Data
Collection
Analytics
Technology
Development
Deployment
Feedback
Collection

Various related efforts…

- Mining Software Repositories (MSR)
- Software Intelligence
- Software Development Analytics

http://www.msrconf.org/
A. E. Hassan and T. Xie. Software intelligence: Future of mining software engineering data. In Proc. FSE/SDP Workshop on Future of Software Engineering Research (FoSER 2010), pages 161–166, 2010.
R. P. Buse and T. Zimmermann. Analytics for software development. In Proc. FSE/SDP Workshop on Future of Software Engineering Research (FoSER 2010), pages 77–80, 2010.

Outline

- Overview of Software Analytics
- Selected projects
- Experience sharing on Software Analytics in practice

Selected projects

XIAO Scalable code clone analysis

StackMine – Performance debugging in the large via mining millions of stack traces

Service Analysis Studio: Incident management for online services

XIAO

Scalable code clone analysis

Yingnong Dang, Dongmei Zhang, Song Ge, Chengyun Chu, Yingjun Qiu, Tao Xie, XIAO: Tuning Code Clones at Hands of Engineers in Practice, in *Proceedings of Annual Computer Security Applications Conference 2012, (ACSAC 2012)*, Orlando, Florida, USA, December, 2012.

Code clone research

- Tons of papers published in the past decade

- 8 years of International Workshop on Software Clones (IWSC) since 2006

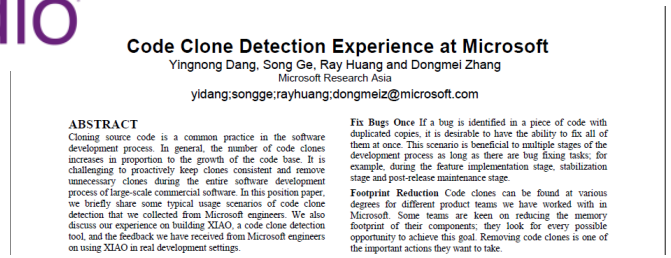
Source: http://www.dagstuhl.de/12071

- Dagstuhl Seminar
  - Software Clone Management towards Industrial Application (2012)
  - Duplication, Redundancy, and Similarity in Software (2006)

XIAO: Code clone analysis

- Motivation
  - Copy-and-paste is a common developer behavior
  - A real tool widely adopted internally and externally
- XIAO enables code clone analysis in the following way
  - High tunability
  - High scalability
  - High compatibility
  - High explorability

Visual Studio



Code Clone Detection Experience at Microsoft

Yingnong Dang, Song Ge, Ray Huang and Dongmei Zhang
Microsoft Research Asia
yidang;songge;rayhuang;dongmeiz@microsoft.com

ABSTRACT
Cloning source code is a common practice in the software development process. In general, the number of code clones increases in proportion to the growth of the code base. It is challenging to proactively keep clones consistent and remove unnecessary clones during the entire software development process of large-scale commercial software. In this position paper, we briefly share some typical usage scenarios of code clone detection that we collected from Microsoft engineers. We also discuss our experience on building XIAO, a code clone detection tool, and the feedback we have received from Microsoft engineers on using XIAO in real development settings.

Fix Bugs Once If a bug is identified in a piece of code with duplicated copies, it is desirable to have the ability to fix all of them at once. This scenario is beneficial to multiple stages of the development process as long as there are bug fixing tasks; for example, during the feature implementation stage, stabilization stage and post-release maintenance stage.

Footprint Reduction Code clones can be found at various degrees for different product teams we have worked with in Microsoft. Some teams are keen on reducing the memory footprint of their components; they look for every possible opportunity to achieve this goal. Removing code clones is one of the important actions they want to take.

[IWSC'11 Dang et.al.]

High tunability – what you tune is what you get

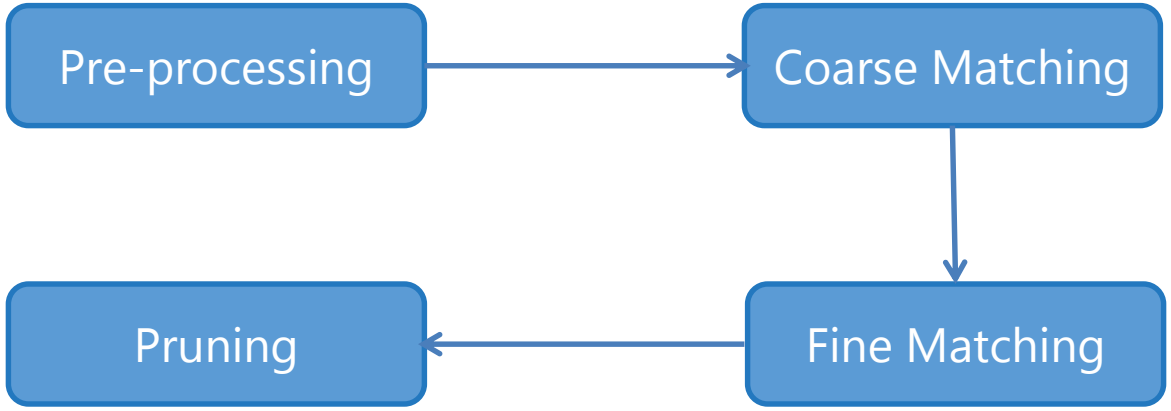
- Intuitive similarity metric
  - Effective control of the degree of syntactical differences between two code snippets
- Tunable at fine granularity
  - Statement similarity
  - % of inserted/deleted/modified statements
  - Balance between code structure and disordered statements

```
for (i = 0; i < n; i ++) {
      a ++;
      b ++;
      c = foo(a, b);
      d = bar(a, b, c);
      e = a + c; }
```

```
for (i = 0; i < n; i ++) {
      c = foo(a, b);
      a ++;
      b ++;
      d = bar(a, b, c);
      e = a + d;
      e ++; }
```

High scalability

- Four-step analysis process



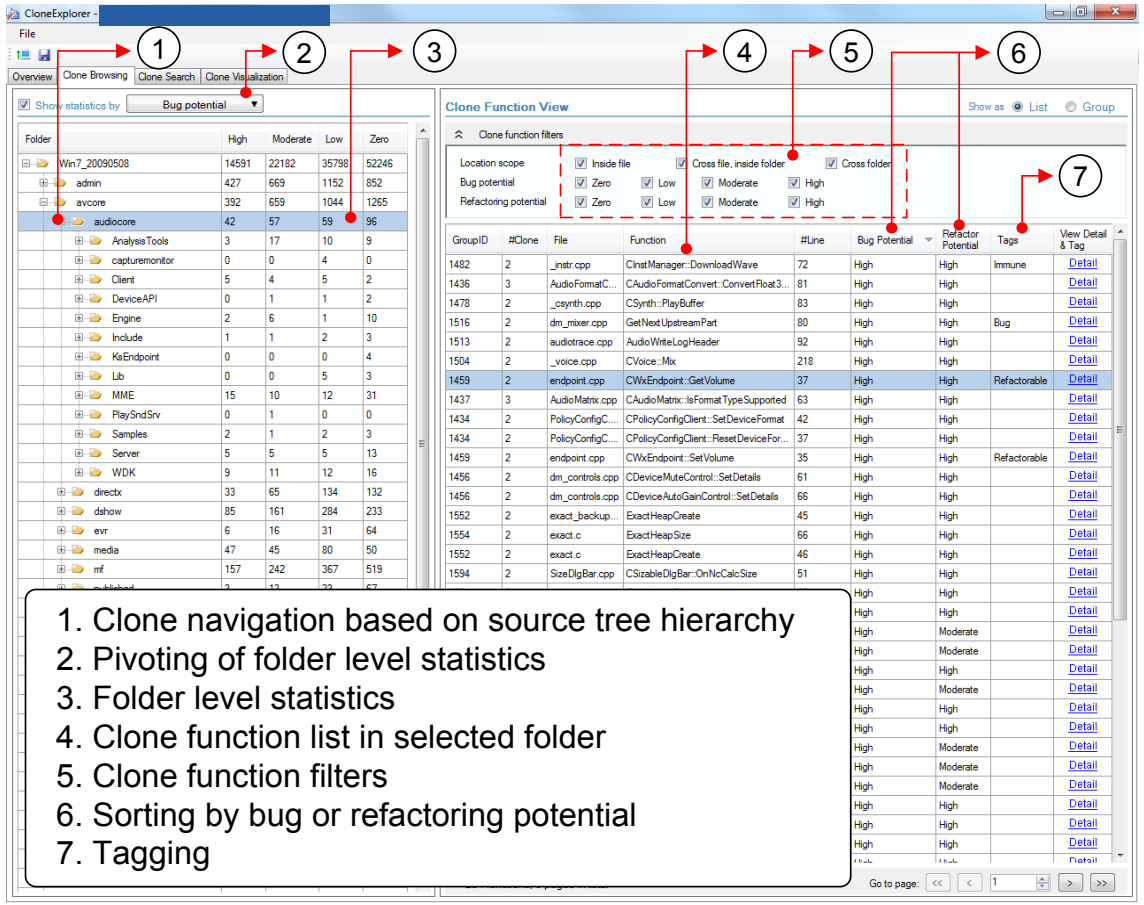
- Easily parallelizable based on source code partition

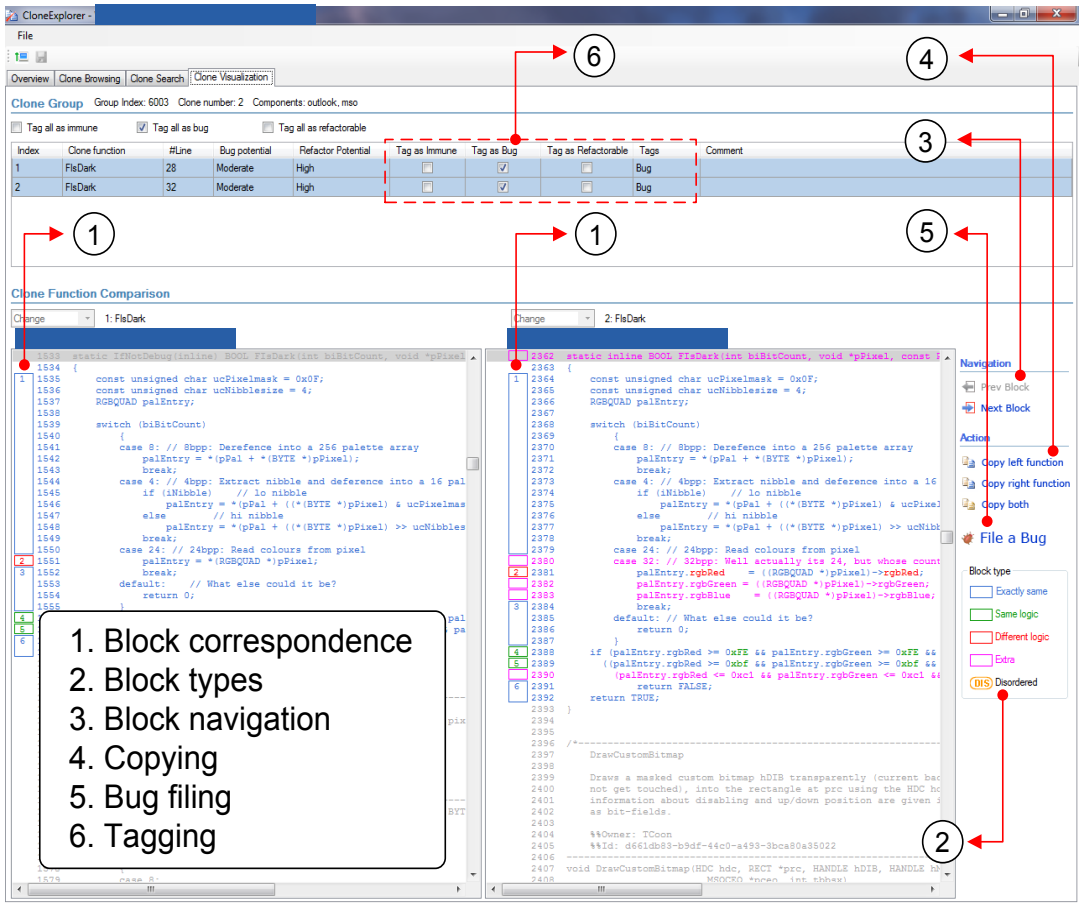
High compatibility

- Compiler independent
- Light-weight built-in parsers for C/C++ and C#
- Open architecture for plug-in parsers to support different languages

- Easy adoption by product teams
  - Different build environment
  - Almost zero cost for trial

High explorability





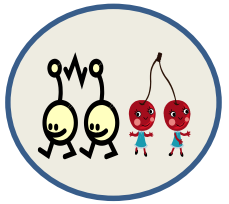
Scenarios and solutions

Quality gates at milestones
- Architecture refactoring
- Code clone clean up
- Bug fixing

Post-release maintenance
- Security bug investigation
- Bug investigation for sustained engineering

Development and testing
- Checking for similar issues before check-in
- Reference info for code review
- Supporting tool for bug triage

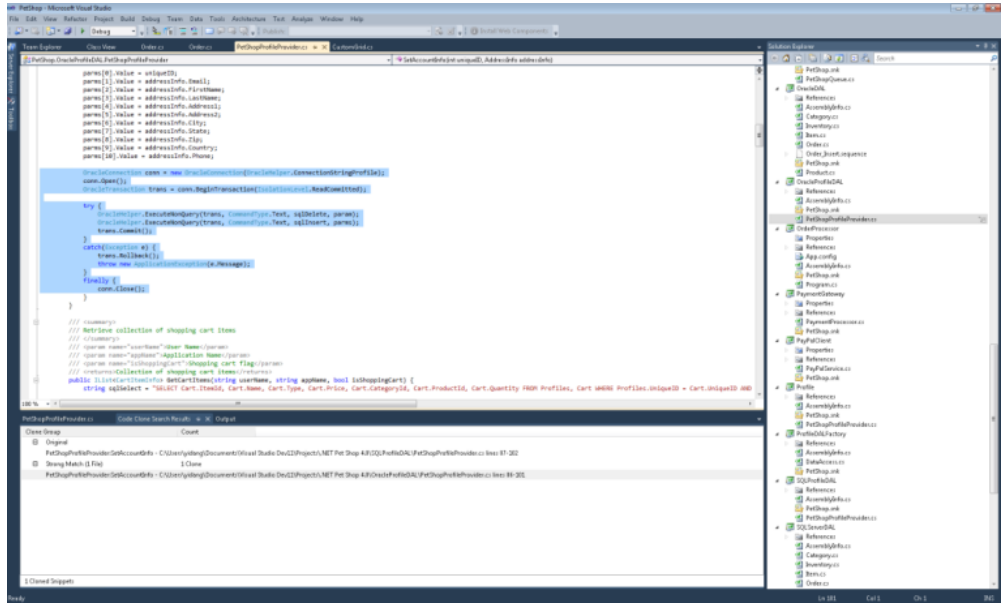
Online code clone search



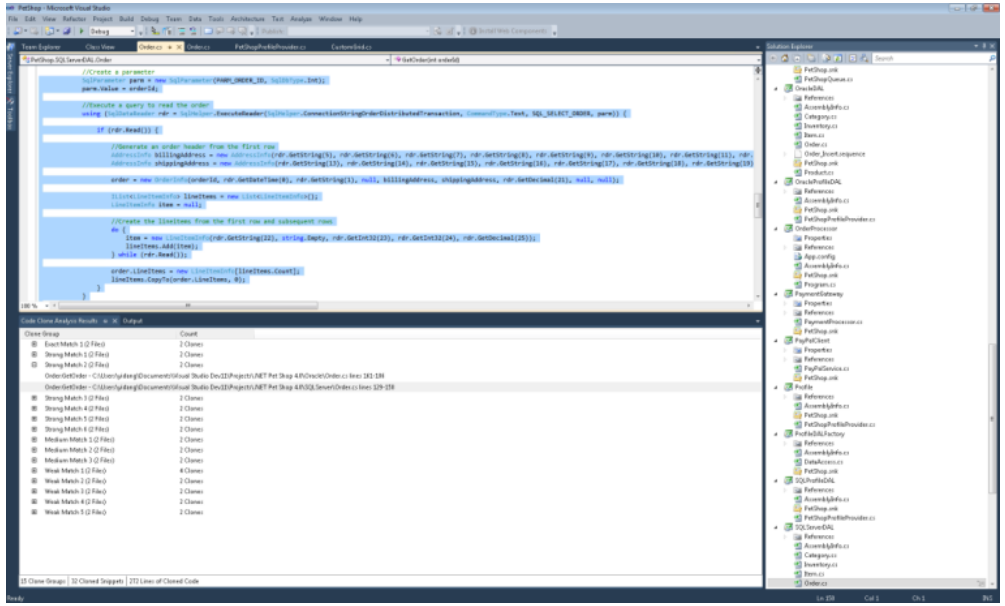
Offline code clone analysis

Benefiting developer community

Available in Visual Studio 2012



Searching similar snippets for fixing bug once



Finding refactoring opportunity

More secure Microsoft products

Code Clone Search service integrated into workflow of Microsoft Security Response Center

Over hundreds of million lines of code indexed across multiple products

Real security issues proactively identified and addressed

Example – MS security bulletin MS12-034

Combined Security Update for Microsoft Office, Windows, .NET Framework, and Silverlight, published: Tuesday, May 08, 2012

3 publicly disclosed vulnerabilities and seven privately reported involved. Specifically, one is exploited by the Duqu malware to execute arbitrary code when a user opened a malicious Office document

Insufficient bounds check within the font parsing subsystem of win32k.sys

Cloned copy in gdiplus.dll, ogl.dll (office), Silver Light, Windows Journal viewer

Microsoft Technet Blog about this bulletin

"However, we wanted to be sure to address the vulnerable code wherever it appeared across the Microsoft code base. To that end, *we have been working with Microsoft Research to develop a "Cloned Code Detection" system that we can run for every MSRC case to find any instance of the vulnerable code in any shipping product*. This system is the one that found several of the copies of CVE-2011-3402 that we are now addressing with MS12-034."

Three years of effort

Prototype development

- Problem formulation
- Algorithm research
- Prototype development

Early adoption

- Algorithm improvement
- System / UX improvement

Tech transfer

- System integration
- Process integration

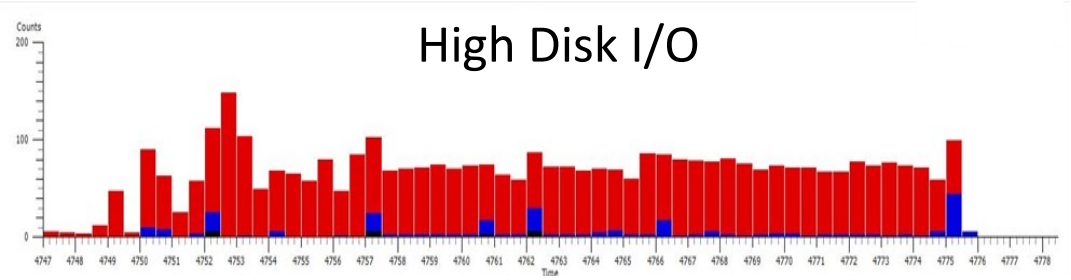
StackMine

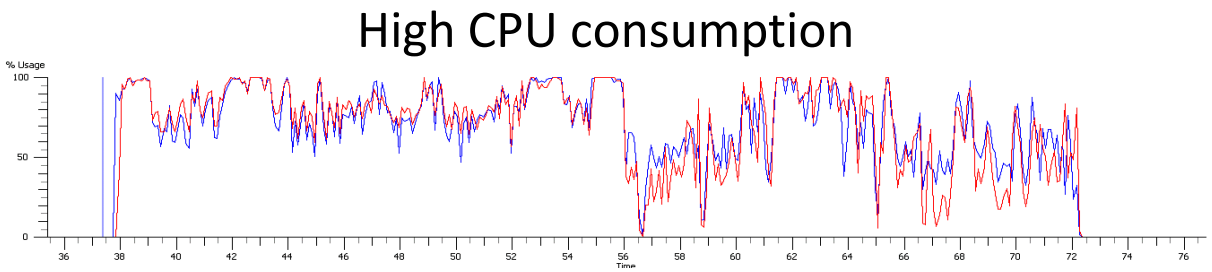
Performance debugging in the large via mining millions of stack traces

Shi Han, Yingnong Dang, Song Ge, Dongmei Zhang, and Tao Xie, Performance Debugging in the Large via Mining Millions of Stack Traces, in *Proceedings of the 34th International Conference on Software Engineering (ICSE 2012)*, Zurich, Switzerland, June 2012.

Performance issues in the real world

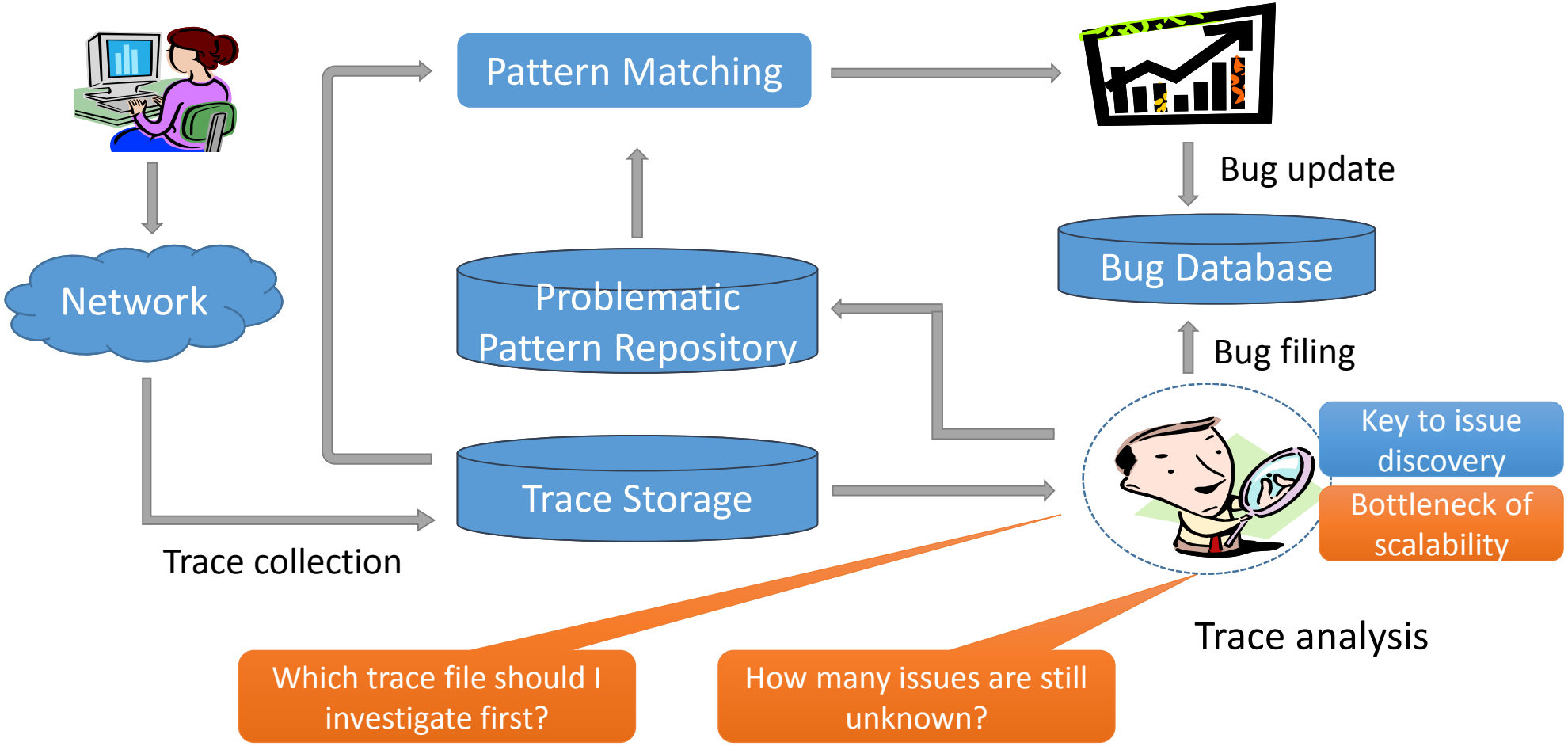
- One of top user complaints
- Impacting large number of users every day
- High impact on usability and productivity





As modern ***software systems*** tend to get more and more ***complex***, given ***limited*** time and resource ***before*** software ***release***, ***development-site*** testing and debugging become more and more ***insufficient*** to ensure satisfactory software performance.

Performance debugging in the large



Problem definition

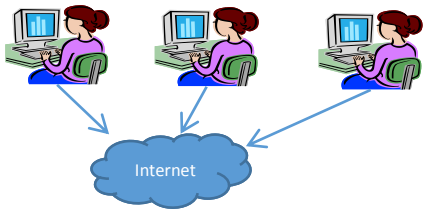
Given operating system traces collected from tens of thousands (potentially millions) of users, how to help domain experts identify the program execution patterns that cause the ***most impactful*** underlying performance problems with ***limited time and resource***?

Goal

Systematic analysis of OS trace sets that enables

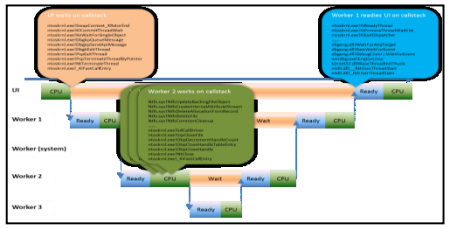
- Efficient handling of large-scale trace sets
- Automatic discovery of new program execution patterns
- Effective prioritization of performance investigation

Challenges



Large-scale trace data

- TBs of trace files and increasing
- Millions of events in single trace stream



Highly complex analysis

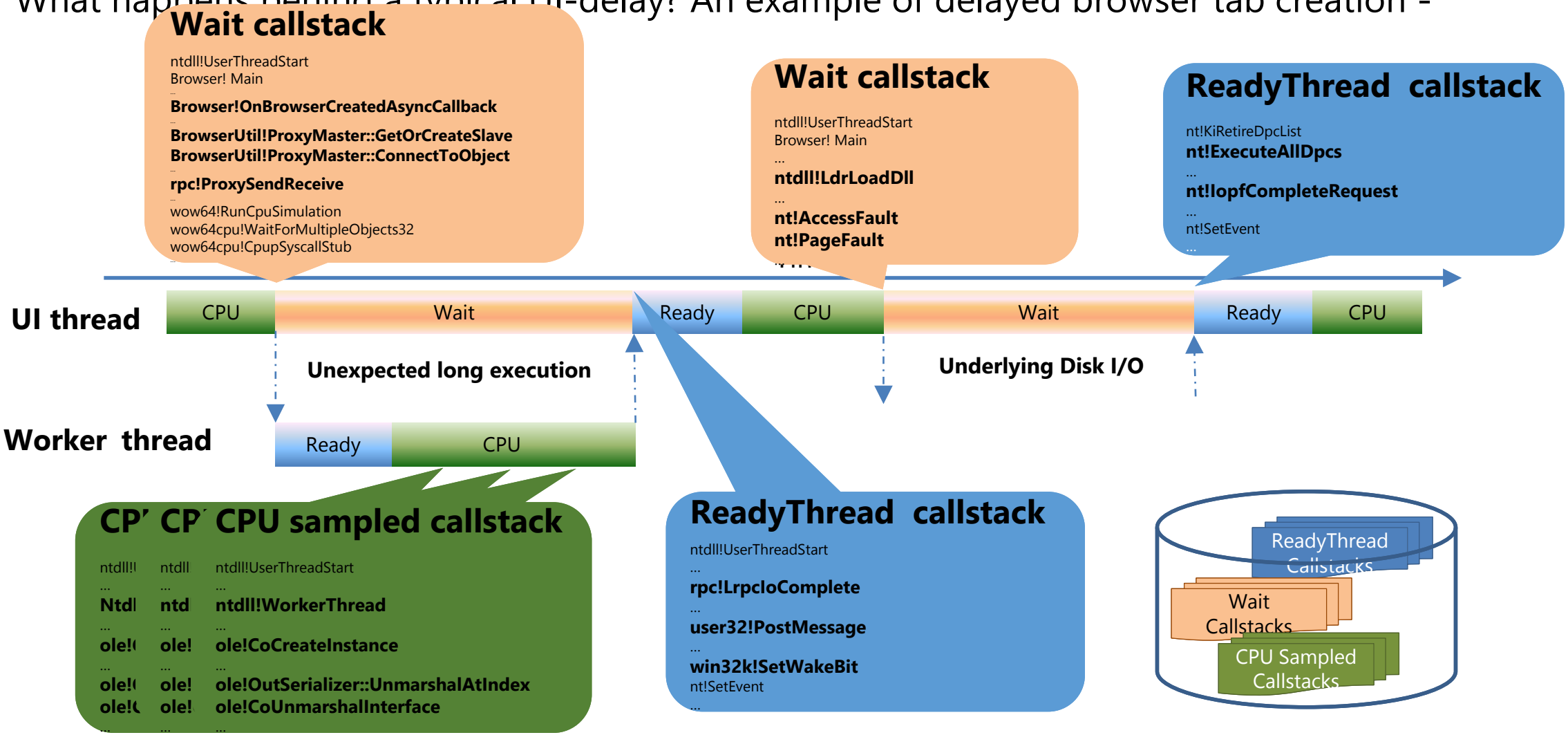
- Numerous program runtime combinations triggering performance problems
- Multi-layer runtime components from application to kernel intertwined

Combination of expertise

- Generic machine learning tools without domain knowledge guidance do not work well

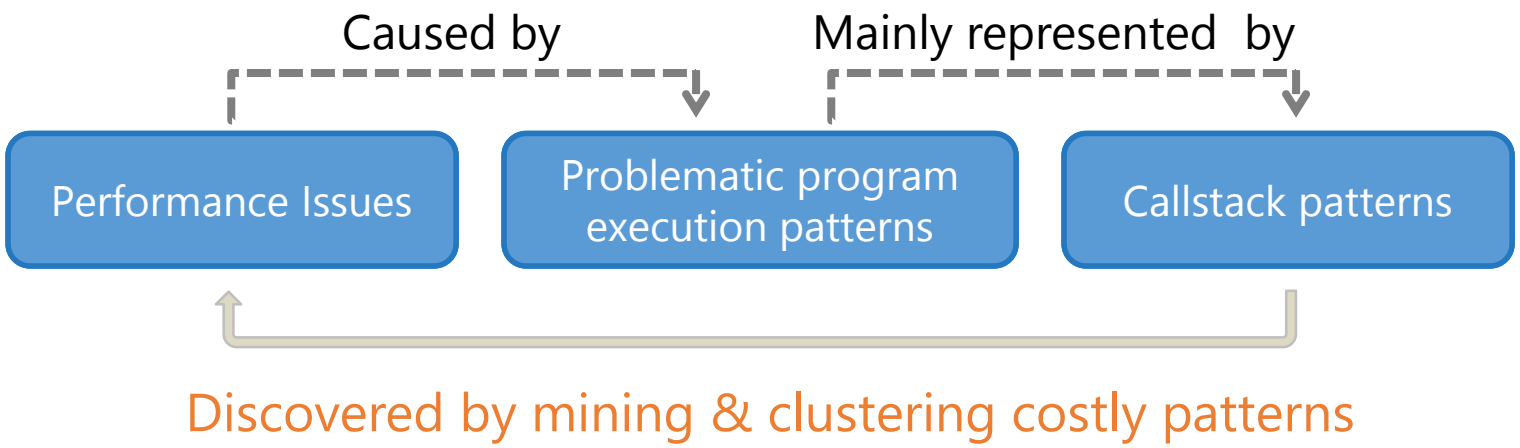
Intuition

What happens behind a typical UI-delay? An example of delayed browser tab creation -



Approach

Formulate as a callstack mining and clustering problem

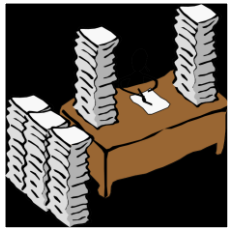


Technical highlights

- Machine learning for system domain
  - Formulate the discovery of problematic execution patterns as callstack mining and clustering
  - Systematic mechanism to incorporate domain knowledge

- Interactive performance analysis system
  - Parallel mining infrastructure based on HPC+MPI
  - Visualization aided interactive exploration

Impact



"We believe that the MSRA tool is highly valuable and much more efficient for mass trace (100+ traces) analysis. For 1000 traces, we believe the tool saves us 4-6 weeks of time to create new signatures, which is quite a significant productivity boost."

Highly effective new issue discovery on Windows mini-hang

Continuous impact on future Windows versions

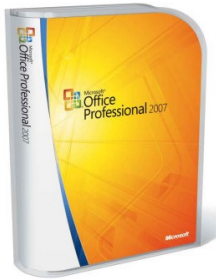
Service Analysis Studio

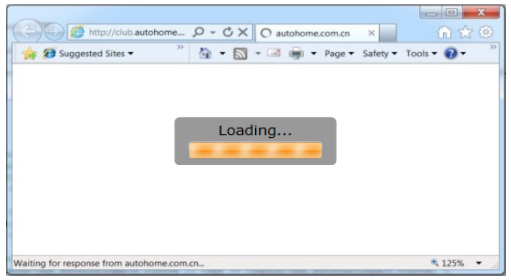
Incident management for online services

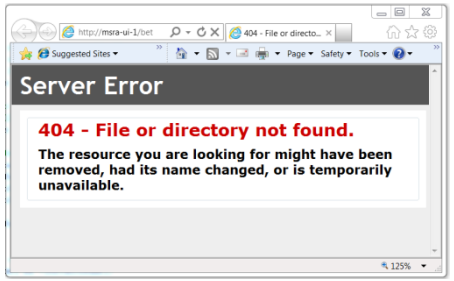
Jian-Guang Lou, Qingwei Lin, Rui Ding, Qiang Fu, Dongmei Zhang and Tao Xie, Software Analytics for Incident Management of Online Services: An Experience Report, in *Proceedings of the 28th IEEE/ACM International Conference on Automated Software Engineering (ASE 2013),* Experience papers, Palo Alto, California, November 2013.

Motivation

- Online services are increasingly popular and important
- High service quality is the key
- Incident management is a critical task to ensure service quality







Incident management: workflow

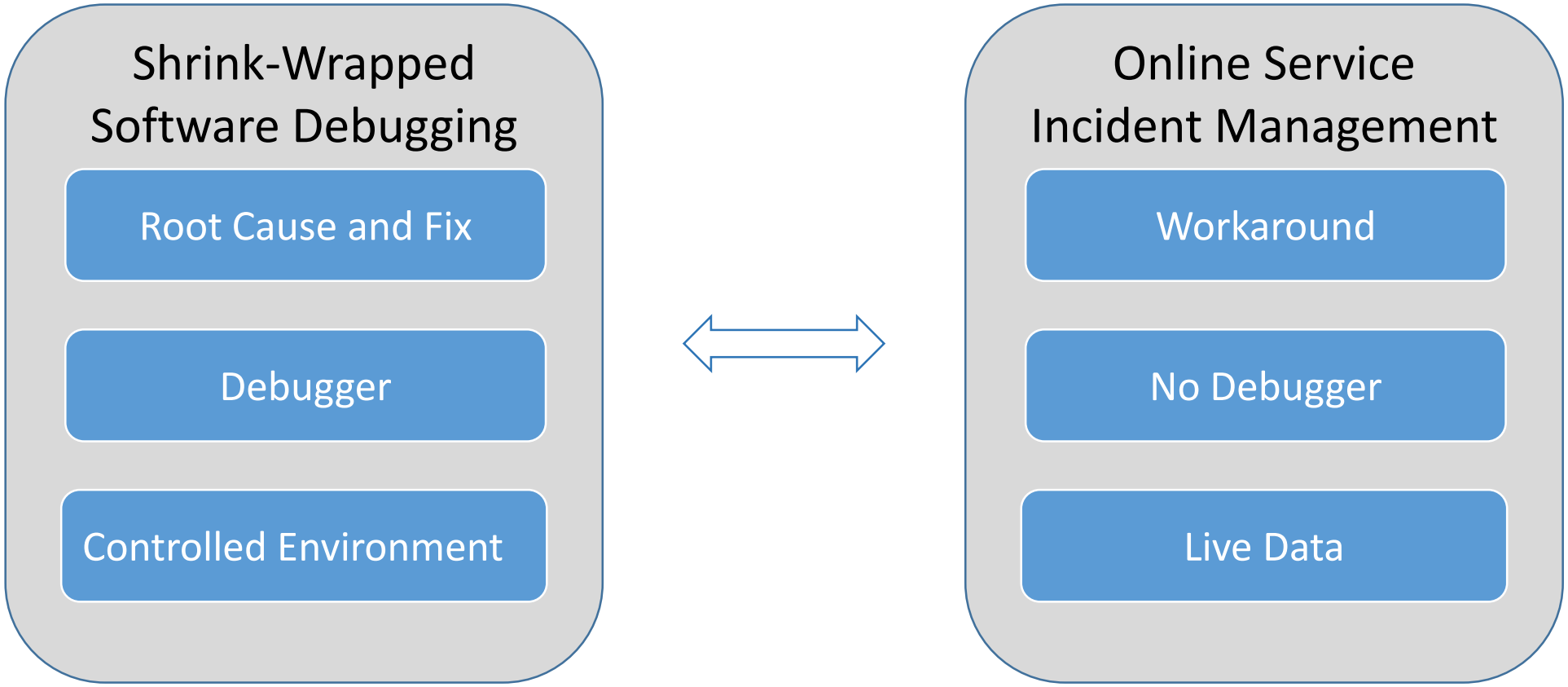
Alert On-Call Engineers (OCEs)

Investigate the problem

Restore the service

Fix root cause via postmortem analysis

Incident management: characteristics



Incident management: challenges

Large-volume and noisy data

Highly complex problem space

Knowledge scattered and not well organized

Few people with knowledge of entire system

Data sources

Name	Description	Examples
Key Performance Indicators (KPI)	Measurements indicating the major quality perspectives of an online service	Request failure rate, average request latency, etc.
Performance counters and system events	Measurements and events indicating the status of the underlying system and applications	CPU, disk queue length, I/O, request workload, SQL-related metrics, and application-specific metrics, etc.
User requests	Information on user requests	Request return status, processing time, consumed resources, etc.
Transaction logs	Generated during execution, recording system runtime behaviors when processing requests	Timestamp, request ID, thread ID, event ID, and detailed text message, etc.
Incident repository	Historical records of service incidents	Incident description, investigation details, restoration solution, etc.

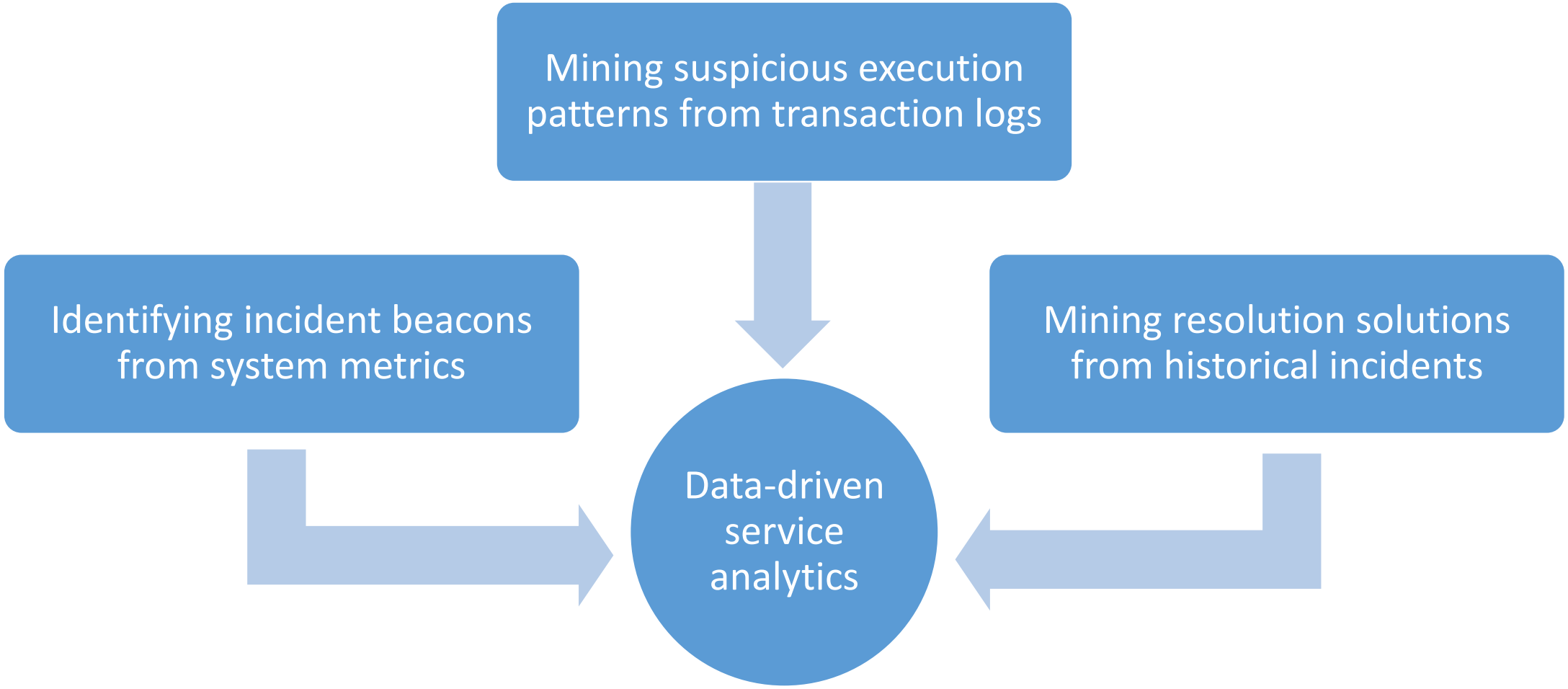
Service Analysis Studio (SAS)

- Goal

  Given an incident in an online service, effectively helping service engineers reduce Mean Time To Restore (MTTR).

- Design principals
  - Automating data analysis
  - Handling heterogeneous data sources
  - Accumulating knowledge
  - Supporting human-in-the-loop (HITL)

Data analysis techniques



Impact

Deployment

- SAS deployed to worldwide datacenters of Service X in June 2011
- Five more updates since first deployment

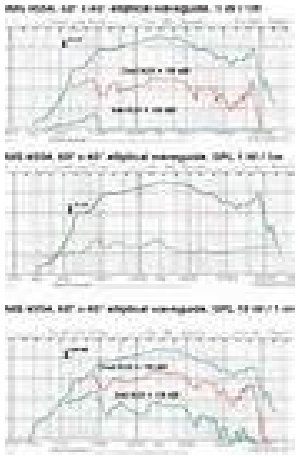
Usage

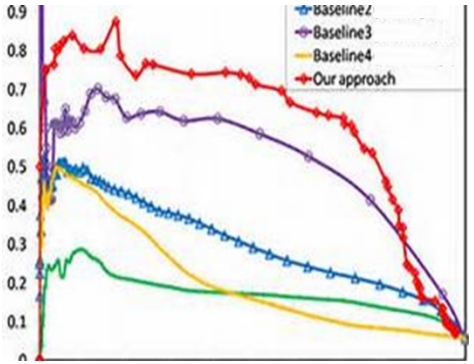
- Heavily used by On-Call Engineers of Service X for about 2 years
- Helped successfully diagnose ~76% of service incidents

Lessons learned

- Understanding and solving real problems
- Understanding data and system
- Handling data issues
- Making SAS highly usable
- Achieving high availability and performance
- Delivering step-by-step

Understanding and solving real problems



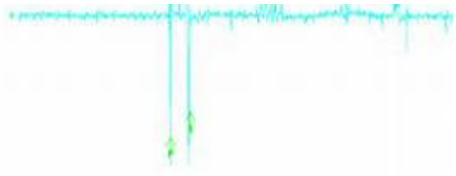


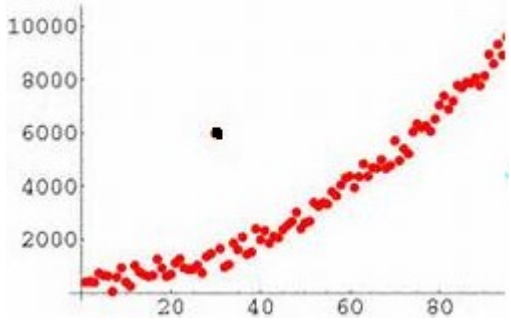
- Working side-by-side with On-Call Engineers
- Targeting at reducing MTTR
- Focusing on addressing challenges in real-world scenarios

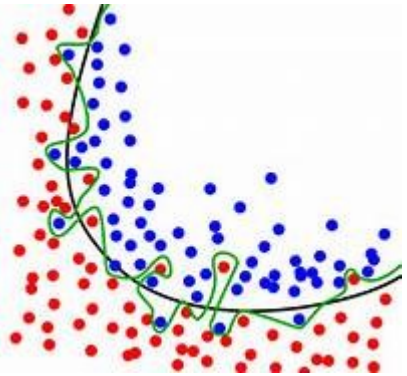
Understanding data and system

Techniques

Practical Problems







Handling data issues

Data issues	Approach	Experience
(1) Missing/duplicated (2) Buggy (3) Disordered	(1) Preprocessing (2) Designing robust algorithms	Data preprocessing cannot be perfect. Robust algorithms are in great need.

Making SAS highly usable

There is an internal server error related issue.

Datacenter: DC1

Start time: 9/4/2012 3:48:00 AM End time: 9/4/2012 3:58:00 AM

Impact:

Influenced requests	1000
Influenced end users	100

Diagnosis:

This issue is a problem of "Credential loss". The source of the issue mainly locates at Front End Server—"FE001".

Here are similar previous occurrences of the issue:

- Incident ID 91236: 3/14/2012 10:49:00 AM (see detail)
- Incident ID 91271: 7/26/2012 14:25:00 AM (see detail)

See also:

Malfunctioned Frontend Servers	973 of 1000 failed requests related to FE001.
Malfunctioned SQL Servers	No malfunctioned SQL servers detected.
Suspicious Metrics	No highly correlated metrics found.
Suspicious Execution Patterns	1 major pattern in the logs covers 973 of 1000 failed requests.

Suggested actions based on similar past incident (ID 91236):

Reset the IIS service on the front end server FE001.

Easy to navigate

Understandable

Actionable

Achieving high availability and performance

- SAS is also a service
  - To serve On-Call Engineers at any time with high performance
  - Critical to reducing MTTR of services
- Auto recovery
  - Continuously monitored
  - Check-point mechanism adopted
- Backend service + On-demand analysis

Delivering step-by-step

- Demonstrating value and building trust
  - Deployment in production has cost and risk
  - In-house → dogfood → one datacenter → worldwide datacenters
- Getting timely feedback
  - Requirements may not be clear early on and requirements may change
  - Gaining troubleshooting experiences from On-Call Engineers
  - Understanding how SAS was used
  - Identifying direction of improvement

Outline

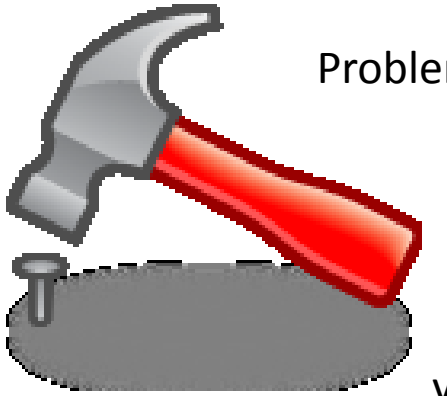
- Overview of Software Analytics
- Selected projects
- Experience sharing on Software Analytics in Practice

Analytics is the means to the end

Interesting results

vs.

Actionable results



Problem hunting

vs.

Problem driven

Beyond the “usual” mining

- Mining vs. matching
- Automatic vs. interactive
- Researchers vs. practitioners

Keys to making real impact

- Engagement of practitioners
  - Solving their problem
  - Timing
  - Champions in product teams
  - Culture
- Walking the last mile
  - Targeting at real scenarios
  - "It works" is not enough
  - Trying out tool has cost
  - Getting engineering support
- Combination of expertise
  - Research capabilities
  - Visualization & design
  - Engineering skills to build systems
  - Communication

Summary

Software Analytics

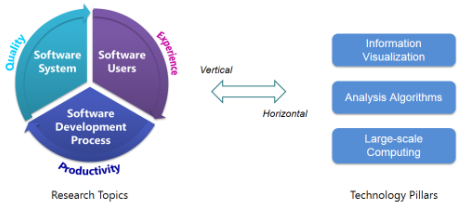
Software analytics is to enable *software practitioners* to perform data exploration and analysis in order to obtain *insightful and actionable* information for *data-driven tasks* around software and services.

NASAC 2013 9

Five dimensions

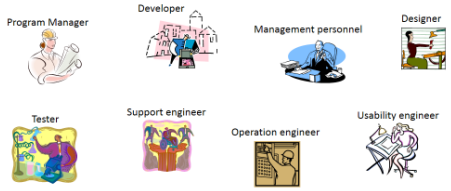
FSE 2014 Tutorial 10

Research topics and technology pillars



NASAC 2013 23

Target audience – software practitioners



NASAC 2013 17

Output – insightful information

- Meaningful and useful understanding or knowledge towards completing target tasks

Output – actionable information

- Enabling software practitioners to come up with concrete solutions towards completing target tasks
- Examples
  - How to reduce regression (re-opened bugs) in development?
    Groups of re-opened bugs each with the same reason of re-opening
  - How to recover an online service when an incident occurs?
    Problematic areas identified with reasons and confidence values
  - How to decide which part of my code should be refactored?
    Detected cloned code snippets easily explored from different perspectives

NASAC 2013 21

Connection to practice

- Software Analytics is naturally tied with software development practice
- Getting real

FSE 2014 Tutorial 17

Together let us walk the exciting journey to make great impact!

Q & A

http://research.microsoft.com/groups/sa/