

# A Survey of Tools for Software Operation Knowledge Acquisition

*Baldur Kristjánsson*  
*Henk van der Schuur*

Technical Report 2009-028  
November 2009

Department of Information and Computing Sciences  
Utrecht University, Utrecht, The Netherlands  
[www.cs.uu.nl](http://www.cs.uu.nl)

ISSN: 0924-3275

Department of Information and Computing Sciences  
Utrecht University  
P.O. Box 80.089  
3508 TB Utrecht  
The Netherlands

# A SURVEY OF TOOLS FOR SOFTWARE OPERATION KNOWLEDGE ACQUISITION

Baldur Kristjánsson, B.Sc.

Department of Information and Computing Sciences, Utrecht University  
Padualaan 14, 3584 CH Utrecht, The Netherlands  
bkristja@cs.uu.nl

Henk van der Schuur, M.Sc.

Department of Information and Computing Sciences, Utrecht University  
Padualaan 14, 3584 CH Utrecht, The Netherlands  
hw.schuur@cs.uu.nl

***Abstract.** Software Operation Knowledge (SOK) is knowledge of in-the-field performance, quality and usage of software, and knowledge of end-user experience and feedback. This report describes the process of screening, evaluating, selecting 20 industrially validated tools available for the acquisition of such knowledge from an initial list of 78 leads. After selection, a high-level survey of the features, technical characteristics and market status of each tool was made. It is concluded that there are clear industry leaders in all five market segments identified (crash reporters, feedback gathering, monitoring, profiling and tracing), and that no tool covers the acquisition of all types of SOK, although building such a tool is feasible.*

***Keywords:** Software operation knowledge, software engineering, software development, software tracing, software profiling, software quality, application monitoring, user feedback.*

## 1 INTRODUCTION

Software vendors have a large interest in gaining knowledge about the performance, quality and usage of software in the field at end-users, as well as the users' opinions on the software. However, systematic practice of gathering, analyzing and acting on such knowledge is still limited, due to a multitude of different approaches and tools available, and an unclear picture about how to use this knowledge in a consistent and effective manner. It can be said that software vendors are generally unaware of how their software is performing in the field (Van der Schuur et al., 2008)<sup>1</sup>.

Van der Schuur et al. (2010)<sup>2</sup> attempt to clarify the picture by defining and formulating the concept *Software Operation Knowledge (SOK)*:

Knowledge of in-the-field performance, quality and usage of software, and knowledge of end-user experience and end-user feedback.

They define four types of SOK: Performance, quality, usage and feedback. In addition, five SOK lifecycle processes (identification, acquisition, integration, presentation and utilization) are described.

This report focuses on the acquisition of all types of SOK. According to the SOK framework, acquisition consists of steps:

1. Translating end-user behavior into operation data based on acquisition criteria.
2. Transferring the data from the field to the vendor.
3. Relevant information is extracted from all the data gathered.

Tools that offer one or more of these functionalities qualify as *SOK acquisition tools*.

The remainder of this report is a documentation of the principal work: identifying a wide array of SOK acquisition tools and performing a thorough survey of a selection of those tools. Chapter 2 describes the approach, how leads were generated, evaluated and eventually selected for the final reviews. Chapter 3 presents the final list of tools along with explanations. Chapter 4 presents categorized reviews of the tools that were selected. Each review contains a summary overview, a list of key features (including integration features and mining capabilities), SOK acquisition characteristics (source object, output data type and acquisition techniques) that were available, and market status (industrial validation). Finally, in Chapter 5, the main conclusions are summarized as well as suggestions for further work.

---

<sup>1</sup> H. van der Schuur, S. Jansen, and S. Brinkkemper. *Becoming Responsive to Service Usage and Performance Changes by Applying Service Feedback Metrics to Software Maintenance*. In ASE'08: Proceedings of the 23rd IEEE/ACM International Conference on Automated Software Engineering – Workshops, pages 53–62, 2008.

<sup>2</sup> H. van der Schuur, S. Jansen, and S. Brinkkemper. *A Reference Framework for Utilization of Software Operation Knowledge*. 2010. To appear.

## 2 SELECTION AND REVIEWING PROCESS

This section describes the process of selecting the tools reviewed, from lead generation to final review. In figure 1, the process is shown graphically, and in the following sections, each step is described in further detail.

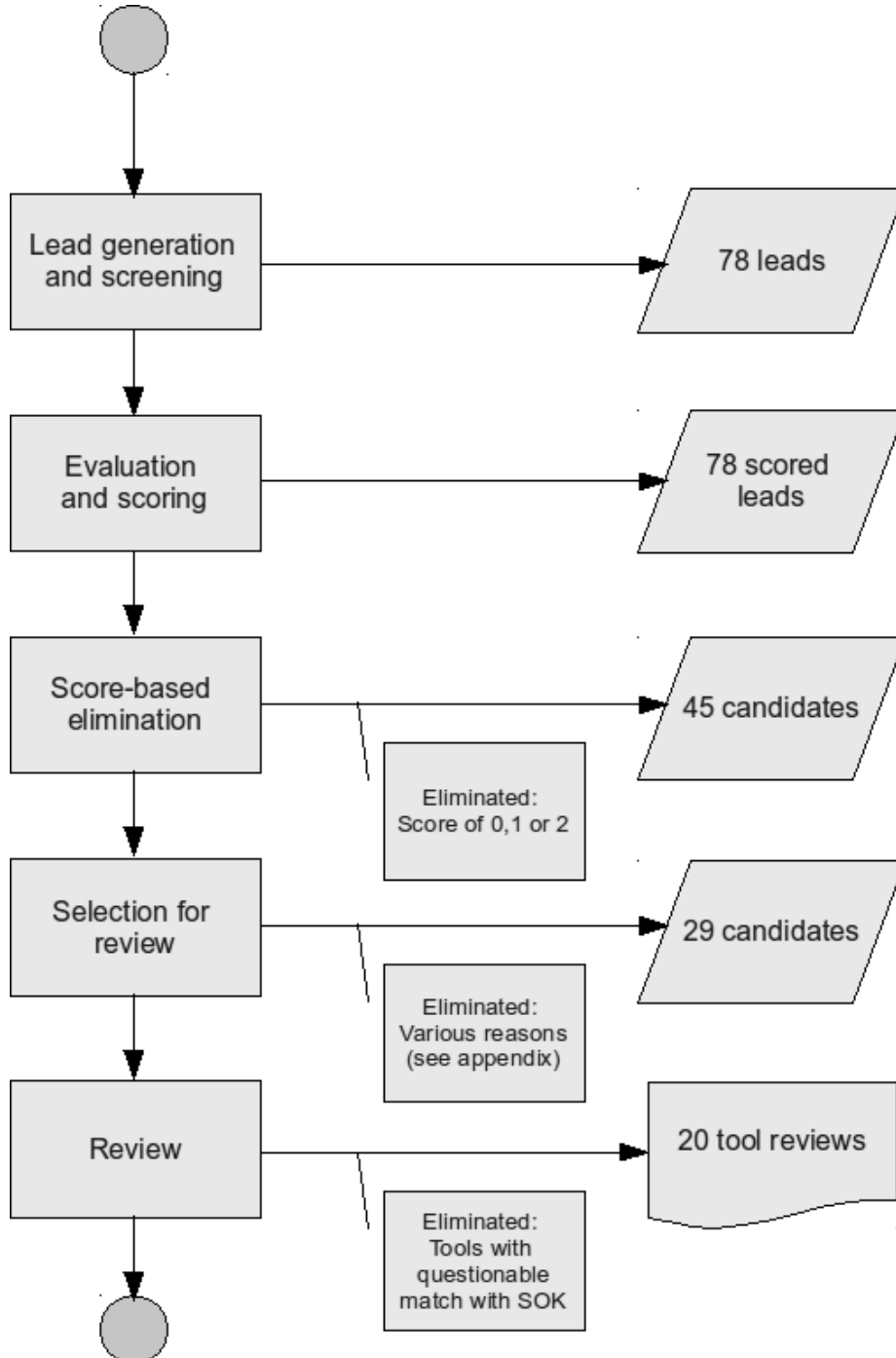


Figure 1 Selection and reviewing process

## **2.1 LEAD GENERATION AND SCREENING**

The main means of generating leads is Internet search. The following tool categories have been identified in preparation for the assignment:

- Software profiling
- Measurement services
- Crash reporting
- Logging
- Tracing
- Web analytics
- User feedback
- User surveys
- Social media monitoring

This reveals several leads, as well as lists of tools (e.g. Wikipedia pages and open source tool listings) that generate even more leads. Other important sources are download sites and social media. During the generation of leads, each tool is screened and not included if it fulfils one of these criteria:

- No own web page (project site as Sourceforge.net qualifies as own web page).
- No recent stable release (2007 or later qualifies as recent stable release).
- Experimental or only used in academic circles.
- Too specialized platform (e.g. only targeted at a narrow part of a development platform).
- Limited UI capabilities compared to competitors.

This approach has generated a total of 78 leads, which are all included in Appendix A.

## 2.2 EVALUATION AND SCORING

Each lead that is generated is evaluated and scored with regard to five criteria. Each tool is shortly analyzed to verify its match with the SOK framework and derive a score. Below, in Table 1, the indicators are listed as well as the requirements for each possible score.

Indicator	Description	0	1	2	3
<b>SOK scope</b>	How many types of SOK does the tool seem to acquire?	None or limited scope of one type	1 type	2 types	3 or 4 types
<b>Runtime platforms</b>	What operating system does the tool run on?	Minor Unix-like or obsolete Windows	Only Windows or only Unix-like OS (incl. Mac)	Cross-platform	
<b>Dev. platforms</b>	How many major development platforms does the tool support?	Library-specific tool	One major platform	Two or more major platforms	
<b>Support</b>	Is there active vendor or community support for the product?	No support or dead community	Currently supported		
<b>Popularity</b>	What are the indications of industrial validation?	No indication of use at all	Some indication of use	User praise, positive reviews, widespread use, awarded	

**Table 1 Scoring criteria**

The popularity indicator requires more explanation. Some indication of use refers to the tool being mentioned on a forum or several download sites listing the tool. Widespread use refers to web searches for the tool revealing obviously more results than most other tools – a typical industrially validated tool generates 30-60,000 hits but 300-600,000 hits would indicate widespread use.

The final score for each tool is the product of all the indicators. Thus the lowest possible score was 0 and the highest possible score was 24. The scores are shown in Appendix A, and the indicators are also part of the final list (Table 2 in chapter 3).

## 2.3 SCORE-BASED ELIMINATION

Some tools were then eliminated on basis of score – it was decided (with three exceptions) that tools with a score of 0, 1 or 2 did not have a place in the list of final candidates. Compared to other tools, they were simply too limited in scope, unsupported or not used in the field. After this step, 33 tools had been eliminated, leaving 45 tools left for further review.

## 2.4 SELECTION FOR REVIEW

Other tools (score 4-24) were evaluated somewhat arbitrarily but industrial validation and scope of features were favored above other indicators. The higher ranked tools needed less scrutiny to get selected. After this step, 16 more tools had been eliminated, leaving a list of 29 final candidates.

Reasons for inclusion of the 3 tools that were included in the former step, and the exclusion of the 15 tools not selected in the latter step, are stated in Appendix A.

## **2.5 REVIEW**

Finally, the tools selected were reviewed in detail with regard to features, SOK acquisition characteristics and market status.

The last tools eliminated were eliminated during the review process itself when aspects were discovered in the review that led to the conclusion that they were not really a match to the definition of a SOK acquisition tool. This is especially true for profilers, which are not typical SOK acquisition tools since most of them do not gather persistent information and require a developer to follow the execution “live”. Again, reasons for elimination are shown in Appendix A.



### 3 FINAL LIST OF TOOLS

Table 2 shows the final list of tools that were selected for review.

Tool	SOK type	Tool type	Vendor	Runtime platform	Dev platform	Popularity	License	Latest stable release
Breakpad	Q	Crash reporter	OS team	Cross-platform	Cross-platform	2	Open Source	No official release
Bugsplat	Q,F	Crash reporter	Bugsplat	Cross-platform	Cross-platform	2	SaaS	N/A (online service)
EurekaLog	Q,F	Crash reporter	EurekaLab	Windows	Borland/.NET	2	Commercial	Borland: 6.0.21 (Sep 2009) .NET: 6.5.3 (Nov 2009)
SmartCrashReports	Q	Crash reporter	Unsanity	Mac OS X	N/A	2	Freeware	1.5 (August 2008)
Winqual	Q	Crash reporter	Microsoft	Windows	Native Windows	2	Freeware	N/A (online service)
GetSatisfaction	F	Feedback tool	GetSatisfaction	Cross-platform	Cross-platform	2	SaaS	2.0 (July 2009)
UserVoice	F	Feedback tool	UserVoice	Cross-platform	Cross-platform	2	SaaS	Dropped beta (Dec 2008)
AlertSite	P,Q	Monitoring	AlertSite	Cross-platform	Web dev.	2	SaaS	N/A (online service)
Glassbox	P,Q	Monitoring	OS team	Cross-platform	Java	2	Open Source	2.0 (Sept 2008)
GlobalWatch	P,Q	Monitoring	WebMetrics	Cross-platform	Web dev.	2	SaaS	N/A (online service)
Precise	P,Q,U	Monitoring	Precise	Cross-platform	.NET/Java	2	Commercial	8.5 (March 2009)
AQTime	P	Profiler	AutomatedQA	Windows	.NET/Borland	2	Commercial	6.21 (June 2009)
JProfiler	P	Profiler	EJ Tech	Cross-platform	Java	2	Commercial	5.2.4 (Sept 2009)
YourKit	P	Profiler	YourKit	Cross-platform	Java/.NET	2	Commercial	.NET: 4.5 (Sept 2009) Java: 8.0 (Feb 2009)
Apache Logging	Q,U	Tracing	Apache	Cross-platform	C++/Java/.NET	2	Open Source	log4cxx: 1.0 (Sep 2007) log4j: 1.2.15 (Sep 2007) log4net: 1.2.10 (Sep 2006)
CodeSite	Q,U	Tracing	Raize	Windows	Delphi/.NET	1	Commercial	4.5 (Sept 2009)
iTech Logging	Q,U	Tracing	iTech	Windows	Cross-platform	1	Commercial	2.6 (Jul 2009)
JAMon API	P,Q,U	Tracing	OS team	Cross-platform	Java	2	Open Source	2.7 (Sept 2007)
LucidLog.Net	P,Q,U	Tracing	Devriver	Windows	.NET	2	Commercial	1.0 (Apr 2007)
SmartInspect	P,Q,U	Tracing	Gurock	Cross-platform	.NET/Java/Delphi	2	Commercial	2.0 (Feb 2007)

Table 2 – Final list of tools

Following is an explanation of each column of Table 2.

## **TOOL**

Refers to the official name of the given tool.

## **SOK TYPE**

The type of software operation knowledge gathered according to the SOK framework<sup>3</sup>.

- **Performance (P)** (e.g. resource usage or response time) can be measured using performance tests (e.g. load tests, stress tests, endurance tests) but knowledge about software performance in the field has to be acquired from the point of deployment using tools built for that purpose.
- **Quality (Q)** can be measured in terms of error metrics such as exceptions, crashes and time between failures. Other possible metrics can be those related to the usability of the solution - end-user productivity and effectiveness.
- **Usage (U)** knowledge refers to data about how a software solution is used by its end users, such as what (components, functions, interfaces) is used and how often.
- **End-user feedback (F)** refers to how end-users value their experience of a software solution.

## **TOOL TYPE**

The principal type of tool expressed as a well-known concept. Five types of tools are identified:

- **Crash reporters.** Crash reporters report crash data to the vendor or a third party, sometimes along with a problem description from the end user.
- **Feedback tools.** Any applications that encourage and gather feedback from the end-user, and present it in a useful way. The tools covered here are popular Web 2.0 “crowdsourcing” sites. Social media monitoring and survey tools were considered too generic to fit into the listing, although they have proved useful to gather user feedback.
- **Application monitoring tools.** Any tools that gather useful information about the execution of an application without the developer having to explicitly insert logging code. This category also covers web site monitoring (with meaningful application specific data-gathering).
- **Profilers.** Dynamic analysis tools for gathering information about a program’s behavior, mostly related to performance and problems in the code such as bottlenecks and memory leaks.
- **Tracing (and logging) tools.** Tools and libraries that record information about a program’s execution according to criteria defined by the developer.

## **VENDOR**

Refers to the vendor responsible for developing and supporting the tool. The term “OS team” refers to the open source team, if applicable.

## **RUNTIME PLATFORM**

Refers to the operating system where the tool is executed. In most cases, this is the same platform as the target application. Exceptions are web application monitors and feedback tools, which are web-based as such, but target applications can be written on various platforms, and are thus classified as cross-platform. Tools that run both on Windows and Unix-like operating systems (Mac OS X, Linux, FreeBSD, and Solaris) are classified as cross-platform.

---

<sup>3</sup> H. van der Schuur, S. Jansen, and S. Brinkkemper. *A Reference Framework for Utilization of Software Operation Knowledge*. 2010. To appear.

**DEVELOPMENT PLATFORM**

The development platform the tool is targeted at.

**LICENSE**

Refers to what kind of license is associated with the use of the tool. Commercial license can also refer to “shareware” (free trial and then buy a license key) since most commercial desktop software is licensed in such a way. Additionally, there are open source tools, freeware (but closed source), and SaaS (Software as a Service, pay per use of web application).

**LATEST STABLE RELEASE**

Shows the version number of the latest official stable release, and the month of the release.

## 4 TOOL DESCRIPTIONS

Following are reviews of the 20 tools that made it to the final list, grouped by tool type.

### 4.1 CRASH REPORTERS

#### 4.1.1 BREAKPAD

<http://code.google.com/p/google-breakpad/>

##### Overview

Google Breakpad is an "open source multi-platform crash reporting system". More specifically, it is licensed under the BSD license and available for Windows 32-bit, Mac OS X and Linux<sup>4</sup>.

##### Key Features

When a crash happens in an application, Google Breakpad allows it to gather and send crash information to a server. Other than that, Breakpad is designed as a very basic and narrowly scoped library with regard to functionality.

##### SOK acquisition characteristics

Google Breakpad consists of two components - a client component that consists of a handler and sender, and a server component that receives the crash information. The handler is an exception handler that generates a "minidump" which is then sent from the client application via HTTP POST. The server component, called Socorro, reads the dumps and turns them into useful debugging information.

##### Market status

Due to Google's endorsement, Breakpad is well known, although there is not much visible support or documentation. Breakpad is in production use in Mozilla Firefox<sup>5</sup>, and there are plans to replacing the current crash reporting component of GNOME with Breakpad<sup>6</sup>.

---

4 <https://blueprints.launchpad.net/ember/+spec/ember-breakpad>

5 <http://benjamin.smedbergs.us/blog/tag/breakpad/>

6 [http://www.opensolaris.org/os/community/edu/os\\_contest/Google\\_Breakpad](http://www.opensolaris.org/os/community/edu/os_contest/Google_Breakpad)

### 4.1.2 BUGSPLAT



<http://www.bugsplatsoftware.com/>

#### **Overview**

Bugsplat is a cross-platform (Windows, .NET, Java, Mac OS X) crash reporting library licensed as a SaaS solution. According to its vendor (of the same name) it "provides a comprehensive set of tools that enable you to track end user crashes in your own products."

#### **Key features**

- Detailed crash report sent to a central server, accessible via web interface.
- User is given the option to upload crash information along with additional remarks.
- More information can be added to the report, such as log files and system information.
- Branding (whitelabelling) of crash reporting dialog.
- Can be integrated with the vendor's defect tracking software.
- Online statistical analysis of aggregate crash report data.

#### **SOK acquisition characteristics**

Bugsplat is implemented as a library which is woven into application code. When a crash occurs, a detailed crash report is generated with stack trace including source code line numbers and function names, and sent via HTTP to the server.

#### **Market status**

Bugsplat Software offers online developer support and documentation. They state that they have received over 3.2 million reports from vendors, which indicates widespread use. It also gets widespread mention on blogs and electronic magazines. According to the company's "competitive analysis", Bugsplat is far superior to its main competitors, Winqual and Google Breakpad.

### 4.1.3 EUREKALOG



<http://www.eurekalog.com/>

#### **Overview**

EurekaLog is a crash reporter supporting the development of Windows applications (Borland environments and .NET). It is described by the vendor as a "complete bug resolution tool [with] the power to catch every exception and memory leak, directly on the end user PC".

#### **Key features**

- Generates a detailed log of the call stack that raised the exception (including line number) and displays it on-screen.
- Creates logs when application freezes or enters an infinite loop.
- Option to send it back to the vendor via e-mail or HTTP message to bug tracking tool (using built-in interfaces with popular tools). EurekaLog Viewer on the vendor end if not integrated.
- Attach screenshot and description to message.
- Debug of third-party DLLs.
- Filtering of exceptions, so only crashes that are "relevant" are reported.

#### **SOK acquisition characteristics**

EurekaLog adds instrumentation code at compile time within the Borland or Visual Studio IDE, enabling the developer to add the functionality with a simple rebuild. It can submit crash logs via HTTP, HTTPS, FTP and in XML format if it's desired.

#### **Market status**

The vendor displays the names and logos of 55 customers on the home page, the best known being Siemens AG. EurekaLog got the 2004 Reader's Awards as the best debugging tool in the Delphi Informant Magazine. On About.com, the reviewer, Zarko Gajic, states that EurekaLog is "a most complete bug resolving solution for Delphi and C++ Builder<sup>7</sup>".

---

<sup>7</sup> <http://delphi.about.com/od/productreviews/ss/eurekalog.htm>

#### 4.1.4 SMART CRASH REPORTS



<http://smartercrashreports.com/>

##### **Overview**

Smart Crash Reports by Unsanity is a free-of-charge crash reporting tool that builds upon Apple's Crash Reporter and allows third party developers to register their products in order to receive crash reports via the Crash Reporter client.

##### **Key features**

- User submits crash reports to vendor via the already familiar Apple Crash Reporter interface.
- Reports be submitted to Unsanity, where developers can access and search crash logs via a web interface.
- User comments can be added to the reports.
- Supports six localizations in addition to English.

##### **SOK acquisition characteristics**

Smart Crash Reports operates on the InputManager mechanism that is built into Mac OS X. The developer adds keys to the Info.plist file and from then on crash reports are sent to the vendor, either via e-mail or via a CGI script set up on the vendor's web server.

##### **Market status**

In addition to their own products, Unsanity lists 110 Mac applications, including the Opera web browser and the Yahoo messenger, which make use of Smart Crash Reports on their homepage. Many bloggers mention Smart Crash reports as a useful tool for Mac developers.

#### **4.1.5 WINQUAL (WINDOWS ERROR REPORTING)**

<http://www.microsoft.com/whdc/winlogo/maintain/StartWER.aspx>

##### **Overview**

Winqual is a free service from Microsoft which utilizes Windows Error Reporting to submit crash information to Microsoft and allows vendors (software vendors, hardware vendors and system manufacturers) to access their crash logs through a web interface.

##### **Key features**

- Captures both kernel-mode crashes and user-mode crashes.
- Submits minidumps (stack traces, registers) for every thread to the Winqual server.
- Allows vendor (only with digital certificate) to access crash logs on the Winqual server.
- A response feature to provide solutions to end-users.

##### **SOK acquisition characteristics**

The vendor establishes a Winqual account using a VeriSign digital certificate. No changes need to be done to source code, since Windows detects application crashes and acts accordingly by sending the crash log to Microsoft. After that, crash logs can be accessed based on mapping of “buckets” (all instances of a specific error associated with application, hardware or OS profile) to the organization.

##### **Market status**

Winqual is well known and Microsoft states that over 1.900 ISVs have signed up with Winqual<sup>8</sup>. Some of them complain about the need to use digital certificates<sup>9</sup>.

---

<sup>8</sup> <http://msdn.microsoft.com/en-us/isv/bb190483.aspx>

<sup>9</sup> <http://www.idevelopsoftware.com/2008/03/winqual-registration-head-aches/>



## 4.2 FEEDBACK TOOLS

### 4.2.1 GETSATISFACTION



<http://www.getsatisfaction.com>

#### Overview

GetSatisfaction is a Web 2.0 feedback application offering so-called customer support communities, “a place where customers and companies connect around the products and services they use.”

#### Key features

- Each vendor gets a community page that looks in some respects like a forum. At the top it is indicated whether employees are “there to help”, and a company logo can be set.
- Each post/topic is focused around a specific goal. A goal can be a question, an idea, a problem or praise. The posts can then be followed up by other customers or employees. Ideas and problems map to feature requests and bug reports in the case of software. It is marked whether the topic is resolved/answered.
- Vendors can make GetSatisfaction their support hub, since it integrates with other services like email, Zendesk and Twitter.
- A feature for paying customers is “feedback analytics”, statistical overviews of customer feedback.
- The interface can be customized further. There are widgets that can be embedded into the company website and some API exposure for those who want to integrate their own systems with GetSatisfaction.

#### SOK acquisition characteristics

GetSatisfaction is delivered as an independent web application, HTML/JavaScript widgets and an API (technical platform is not specified). Feedback mechanisms could thus be woven into application code in order to send feedback data to GetSatisfaction from the application itself. All data is stored at Get Satisfaction’s own database.

#### Market status

GetSatisfaction has over 25,000 support communities. Among customers are Apple, Microsoft, Mozilla and Nike. Its main competitors, however, are UserVoice and SuggestionBox. The former is included in this report, but the latter was omitted due to limited usefulness in order to gather defect reports and feature requests. GetSatisfaction is an inspiration or a relatively new and small service called SiteRemark, which was so similar that it was omitted.

## 4.2.2 USERVOICE



<http://www.uservoice.com/>

### Overview

UserVoice is a Web 2.0 feedback service targeted at companies developing their products (not necessarily software). It provides the vendor with a community where users “share ideas and vote up the best”, and the vendor “responds, implements and repeats!”

### Key features

- Feedback forums where their users can post ideas and suggestions, and other users can then comment and vote on the idea.
- Each user gets 10 votes and can cast up to 3 votes on an idea. Votes can be changed afterwards and are returned when an idea has been responded to, “closed”.
- Forums have a customizable look and feel, and the vendor's logo (whitelabeling).
- Feedback can be gathered by embedding HTML and JavaScript code in the vendor's web site (widget).
- Search functionality that searches for existing ideas whenever a new idea is posted, to prevent duplicates.
- Single sign-in feature for users.
- Analytics and export to Excel.

### SOK acquisition characteristics

UserVoice is delivered as a standalone web application and HTML/JavaScript widgets. All data is stored at UserVoice's own database.

### Market status

UserVoice does not display the number of communities, but some well known companies (Sun Microsystems, Myspace.com, and SonyBMG Entertainment) use it. In marketing material it is stated that the real difference lies in UserVoice's voting system. The company UserVoice Inc., which was a startup in 2007 and has only 9 employees, received recently funding of \$800,000<sup>10</sup>.

---

<sup>10</sup> <http://www.techcrunch.com/2009/05/18/uservoice-raises-funding-white-labels-user-feedback-facilitator/>

## 4.3 APPLICATION MONITORING

### 4.3.1 ALERTSITE



<http://www.alertsite.com/>

#### Overview

AlertSite is a suite of online tools to measure and monitor the performance, usage and availability of web sites in the field. The company AlertSite describes itself as “a leading provider of Web performance management solutions”.

#### Key features

- Measures web sites from 40 different locations by testing them regularly according to customer-specified criteria.
- The business transaction monitor (DejaClick), also described as “inside the browser” monitoring, automatically builds tests scripts (sort of operational profiles) by following users' click streams. Scripts can be customized by the customer to model replay behavior.
- DejaCapture records snapshots of errors when they occur.
- InSite, a configuration of AlertSite to gather data internally.
- Correlation of the above data with Google Analytics.
- Email and phone alerts in case of underperformance.
- Detailed and customizable reports and graphs.

#### SOK acquisition characteristics

HTTP requests are sent from multiple geographical locations and data gathered about the response. For transaction monitoring, a proprietary technology called TrueUser is used, but the source object here is the application front-end and its behavior. Additionally, the technology monitors real users to create operational profiles, adjusts for dynamic features and records interactions with Flash content.

#### Market status

AlertSite is an established company and has gotten much publicity over the years. Among well-known customers are Honda, Exostar and AT&T.

### 4.3.2 GLASSBOX



<http://www.glassbox.com/>

#### Overview

Glassbox is an open source application monitoring solution for server-side Java applications running in a container. It is described as “a troubleshooting agent for Java applications that automatically diagnoses common problems.”

#### Key features

- Plug-and-play – deploy the .war file to a server directory and add one line to the configuration.
- Monitors code execution in the Java Virtual Machine (JVM) by instrumentation without any changes to source code.
- Glassbox Inspector allows user to monitor JVM activity.
- Glassbox Troubleshooter cross-references execution data with a knowledge base of known problems, highlighting them in plain English, thus allowing technical staff to bypass the process of manually analyzing logs, triaging and (mis)communicating about the problem.
- API to interface with network monitoring applications (e.g. SiteScope), incorporating monitoring of application-specific operations.

#### SOK acquisition characteristics

Glassbox uses aspect-oriented programming to weave instrumentation code into the JVM at class load time. A diagnostic layer analyses the raw monitor logs and cross-references the logs with a knowledge base of known failures.

#### Market status

Glassbox receives good reputation on StackOverflow<sup>11</sup> and has been featured on Google Tech Talk<sup>12</sup>. It is also well distributed on download web sites. It has over 7,000 downloads on Sourceforge.net<sup>13</sup>.

---

11 <http://stackoverflow.com/questions/1686711/what-are-some-interesting-free-open-source-dynamic-analysis-tools-for-java>

12 <http://www.bestechvideos.com/2006/11/22/glassbox-automated-troubleshooting>

13 <http://sourceforge.net/projects/glassbox/files/>

### 4.3.3 GLOBALWATCH



[http://www.webmetrics.com/products/global\\_watch.html](http://www.webmetrics.com/products/global_watch.html)

#### Overview

GlobalWatch is a platform that provides performance and availability analysis for web sites. Its vendor, WebMetrics, describes it as “the most flexible website monitoring services platform enabling online businesses to solve immediate performance management needs.”

#### Key features

- Measures web applications by testing them regularly from over 100 major cities<sup>14</sup>.
- Monitors customer-defined transactions by simulating them on regular intervals and recording performance data. Transactions are defined with the GlobalWatch Script Recorder.
- Web site monitoring service reports on average response time and uptime per client location.
- API Monitor that monitors performance of SOAP and REST web services. Series can be created that simulate a flow of requests.
- Streaming Monitoring Service that monitors connect time, buffer time and throughput of media streams.
- Web Ecosystem Management monitors partner web sites as well.
- Internal Agent that is set up on one or more infrastructure components, data is then submitted to the monitoring server.
- Email and phone alerts when the above are underperforming.
- Customizable reports and graphs.

#### SOK acquisition characteristics

HTTP requests are sent from multiple geographical locations and data gathered about the response. Transaction monitoring is done by means of a Ruby engine driving an Internet Explorer 7 browser, which interacts with the web site on DOM level, meaning that the source object is the web site front-end.

#### Market status

WebMetrics was founded in 1999 and acquired by Neustar in 2008, and regularly receives publicity in the press. They display known customers (e.g. London Business School, Forever 21) on their web site as well as customer testimonials. It is recommended by current users on StackOverflow and mentioned as a high-end tool.

---

<sup>14</sup> <http://www.reuters.com/article/pressRelease/idUS180745+22-Jun-2009+PRN20090622>

#### 4.3.4 PRECISE

<http://www.precise.com/>



##### Overview

Precise is a commercial integrated transaction performance management (TPM) solution. It is cross-platform, available for .NET, J2EE, database systems as well as various ERP systems such as SAP and Oracle. It provides "shared insight for business and IT into performance problems and accelerated resolutions."

##### Key features

- Gathers various performance data (e.g. CPU load, HTTP response time, memory usage, frequency, execution time) and associates with transactions (method calls, database commands).
- Correlates activity across servers.
- Build role-specific dashboards and choose what data is relevant.
- Manage SLA compliance.
- Uses a "common transaction model" to associate performance indicators with transactions and facilitate drill-down to locate performance bottlenecks.
- Performance Warehouse where trends can be viewed in order to possibly prevent future problems.
- Leak Seeker to locate memory leaks.

##### SOK acquisition characteristics

Precise uses in-memory byte code instrumentation to collect method invocation counts, response time, CPU time, lock wait time and SQL statements in real time, in addition to other useful data. It then correlates the data into tree structures where the branches are call paths to methods<sup>15</sup>.

##### Market status

Precise was previously owned by Symantec but in 2008, Precise took over all development and support of Symantec's own application performance management solution, and Symantec sold off the company<sup>16</sup>. Precise claims to be running its software at half of the Fortune 100 companies. Well-known Dutch companies running Precise are Eneco and Ziggo.

---

<sup>15</sup> <http://my.precise.com/asset.aspx?oid=2ff1a2dc-97a9-494b-89d3-e1f4b699ab72>

<sup>16</sup> [http://www.symantec.com/about/news/release/article.jsp?prid=20080117\\_01](http://www.symantec.com/about/news/release/article.jsp?prid=20080117_01)

## 4.4 PROFILERS

### 4.4.1 AQTIME



<http://www.automatedqa.com/products/aqtime/>

#### Overview

AQTime is a commercial profiler targeted at Windows developers (native and .NET). Its vendor, AutomatedQA, describes it as a “TOTAL solution for professional Windows and .NET developers that want to eliminate guesswork during development and deliver rock solid programs when the project is complete.”

#### Key features

- Profiling of CPU time, memory, call graphs, resource allocations and more, displayed graphically.
- Supports both managed (.NET) and unmanaged (Native Windows) code, and both 32-bit and 64-bit Windows.
- Supports profiling of scripts.
- Applications can be profiled in their entirety or partially, and application events can turn profiling on and off.
- Integrated code coverage analysis.
- Integrates into major IDEs for Windows as well as Microsoft Team System.
- Integrates with other software engineering tools from AutomatedQA (automated testing and build tools).

#### SOK acquisition characteristics

AQTime uses instrumentation to profile the target code, what kind of instrumentation is inserted and when remains unclear. Profiling data can only be viewed “online”, that is, during and right after the code has been running. AQTime can run on production binaries but with a severe performance penalty.

#### Market status

AQTime won six different awards during 2003-2006, both from the .NET and Delphi community. It gets mixed reviews at StackOverflow, some are happy, some are not. A profiling guru blogs about it and says that it is a “useful tool” but also mentions a big performance penalty (5-10x) for using it<sup>17</sup>.

---

<sup>17</sup> <http://blog.vlad1.com/2008/07/20/aqtime/>

## 4.4.2 JPROFILER



<http://www.ej-technologies.com/products/jprofiler/overview.html>

### Overview

JProfiler is a commercially licensed profiler for Java developers. Its vendor, ej-technologies, describes it as an “all-in-one” profiler with an intuitive GUI.

### Key features

- Real-time profiling of CPU time (including call graphs and hotspots), memory, monitor (waiting and blocking events), threads and more in a GUI environment.
- Profiling of local and remote sessions.
- Offline profiling – applications can be distributed to runtime (e.g. production) environments with triggers that start and stop profiling, and snapshots can then be viewed in the GUI.
- A “heap walker” to locate memory leaks.
- Telemetry views.
- Supports integration with all popular Java IDEs, 32- and 64-bit Java, and J2EE application servers.
- Extensibility and an API.

### SOK acquisition characteristics

No technical details are given on the JProfiler website about the profiling technique JProfiler uses. To start profiling, the developer inserts a “session” into an XML settings file, and then JProfiler starts profiling that binary.

### Market status

JProfiler got the Best Java Profiling / Testing Tool award from the Java Development Journal in 2003, 2005 and 2007. The vendor displays an impressive portfolio of customers: Apple, Accenture and Allianz, just to name those starting with an A.



### 4.4.3 YOURKIT



<http://www.yourkit.com/>

#### **Overview**

YourKit is a commercially licensed profiler available for Java (multi-platform) and .NET (Windows) from a company bearing the same name. They affirm they are “the standard-setter in the evolution of profiling tools”.

#### **Key features**

- The features listed here apply to the Java profiler, which is the more mature configuration.
- Standalone profiling UI with typical features – CPU times (including hot spots and call graphs) and memory use.
- High-level profiling, including HTTP requests and SQL statements.
- Profiles any application, anywhere (including production environments, remote profiling possible), in any configuration with low overhead.
- Deadlock detection.
- Telemetry views.
- Comprehensive object view (including “dominators” that consume the most resources).
- Results can be exported as HTML or CSV.
- Integrates with all major Java IDEs, J2EE application servers and supports J2EE versions 5 and 6, as well as 64-bit Java.
- Profiler API – integrate the profiler programmatically.

#### **SOK acquisition characteristics**

YourKit allows tracing via both byte-code instrumentation and sampling (querying stacks of running threads).

#### **Market status**

YourKit Java Profiler has received the Java Developer's Journal Editors' Choice Award. It gets prominent praise on developer forums - “great”, “wins hands down”, “very stable”<sup>18</sup>. YourKit LLC claims that their product is used by the big players in the IT industry – SAP, IBM, HP and SonyEricsson, to name a few.

---

<sup>18</sup> <http://stackoverflow.com/questions/163722/which-java-profiler-is-better-jprofiler-or-yourkit>

## 4.5 TRACING AND LOGGING TOOLS

### 4.5.1 APACHE LOGGING SERVICES



<http://logging.apache.org/>

#### Overview

The Apache Logging Services (log4J, log4net and log4cxx) are extensible opensource logging libraries for Java, .NET and C++, respectively.

#### Key features<sup>19</sup>

- The developer invokes a *Logger* class in the target code to generate a log message at one of five levels (fatal, error, warning, info or debug). The levels are either enabled or disabled.
- Various targets can be chosen for outputting the log message – a text file, database, system logs, e-mail etc. This is called choosing an *Appender*.
- A *Layout* for the log message can also be chosen (included simple, pattern, HTML or XML).
- Logging behavior can be set at runtime using a configuration file.
- Chainsaw, a Java-based companion application, can be used to view logs from Log4J only.

#### SOK acquisition characteristics

Like with other loggers, Apache Logging Services are invoked by inserting logging statements into code. Output formats and targets are extremely flexible due to the open source model and structure of the library.

#### Market status

Use of the Apache Logging Services is widespread, especially in the case of log4j. Its open and extensible model is so popular that there have been around 15-20 different ports (how many depends on the definition of “port” or “copycat implementation”, not only the official Apache ports for .NET and C++ but also for almost any industrially validated development platform<sup>20</sup>).

---

<sup>19</sup> [http://www.allappforum.com/log4j/log4j\\_features.htm](http://www.allappforum.com/log4j/log4j_features.htm)

<sup>20</sup> <http://en.wikipedia.org/wiki/Log4j>

## 4.5.2 CODESITE

# CODESITE

<http://www.raize.com/DevTools/CodeSite/>

### Overview

CodeSite is a logging and tracing library targeted at the Windows platform (native Windows, Borland and .NET). The whole system “gives developers deeper insight into how their code is executing, which enables them to locate problems more quickly and ensure their application is running correctly”, according to the vendor, Raize.

### Key features

- Logging classes that generate log messages, from simple string messages to screenshots, stack traces and object collections.
- A separate dispatcher service which allows either live logging or file-based logging, and both locally and remotely.
- Viewer applications with inspectors and tools to view the log messages.
- A message organizer to group, view and analyze messages from a specific category, application, thread or component.

### SOK acquisition characteristics

The CodeSite VCL components are included in the target code and then invoked via instrumentation to send various types of log messages to the log file (or log viewer), which is in a proprietary format (.clf). The dispatcher running on the same machine can submit messages locally or to a remote machine using TCP, UDP or HTTP.

### Market status

CodeSite is mostly recognized as a library for Borland environments rather than .NET. It was easy to find praise for it, both on the Raize website and forums. CodeSite states that a third party market has developed around the product and has partnerships with some vendors of software engineering tools, such as AutomatedQA.

### 4.5.3 ITECH LOGGING



<http://www.itech-software.com/english/>

#### Overview

iTech Logging is a tracing and logging library targeted at native Windows, .NET and Borland developers. It is described by its vendor, iTech, as “a sophisticated 2nd generation solution for application specific protocolling, program monitoring and execution analysis.”

#### Key features

- Generates log messages with meta-information, user-defined object and textual description.
- Log messages are sent to one or more “output channels”, which are configured to write their messages in text, HTML or XML, output templates being customizable by developers.
- Channels are configured to output to either a log file or a log server, which is a Windows service that aggregates messages.
- Another application, ITConfigManager, allows output to be partially or fully deactivated at runtime.
- The viewer application, ITLogBook, can filter and sort messages, monitor logging live and also view remotely log files on a customer workstation.

#### SOK acquisition characteristics

Similar to other loggers, developers write code that uses the iTech library (ITLogLib) to record log messages, either to a file or to a Windows service. The library can be configured online at runtime using the ITConfigManager. The ITLogBook application is standalone and allows local or remote access.

#### Market status

iTech Logging has been sold in ten countries and present two client project references on their home page. There is no recent mention of the product on forums or in magazines.

#### 4.5.4 JAMON

<http://jamonapi.sourceforge.net/>



##### **Overview**

JAMon (Java Application Monitor) is an open source, cross-platform tracing API for Java, primarily developed for monitoring J2EE applications. Its principal claim is being "a free, simple, high performance, thread safe, Java API that allows developers to easily monitor production applications."

##### **Key features**

- Gathering of aggregate descriptive performance statistics in order to allow for the location of performance bottlenecks.
- Track any numeric variables, statistics and details on exceptions and database queries.
- Application server monitoring.
- Monitoring of popular support libraries (Log4J, JDBC, EJB) without code changes.

##### **SOK acquisition characteristics**

Data is acquired by simply referring to the library and inserting start() and stop () statements into blocks of Java code, such as method calls or JSP pages, which makes it ideal to use an aspect weaver to insert those statements. JAMon then gathers summary statistics such as hits, execution times (total, average, minimum, maximum, standard deviation), and simultaneous application requests. It includes a clickable web user interface to view the reports. However, the data is only persisted in the JVM, so if it is shut down, the data is deleted.

##### **Market status**

JAMon is well documented and gets 600 downloads a month on average according to its main contributor. It is recommended by Dutch technical consultancy company Atos Origin as "a simple tool that offers various possibilities for monitoring of applications, with negligible performance penalty". It seems to be widely used, judging by mentions on blogs and forums.

#### 4.5.5 LUCIDLOG.NET



<http://www.devriver.com/logging/>

##### **Overview**

LucidLog.Net is a commercial logging package for .NET developers. It is described by its vendor, DevRiver, as “a tool for tracing, debugging, monitoring and troubleshooting.”

##### **Key features**

- Developer inserts logging code that logs messages at various levels (Message, Warning, Error, Exception and Fatal). Logs stack traces, method timings and any .NET object with public properties.
- Logging can be configured in a configuration file.
- Writes to different targets (log file, log viewer, system event log, and message box). Messages can be sent remotely via TCP/IP (e.g. from end-user to developer) to the log viewer.
- Log viewer with different views, drilldown on objects, hierarchic logs (for method calls), filtering and grouping, navigation and search.
- Special support for tracing multithreaded applications.
- CSV and HTML export.

##### **SOK acquisition characteristics**

Logging statements are inserted into the code by the developer. Messages are written to a plain text file or in a proprietary format to the log viewer.

##### **Market status**

LucidLog.Net is very well distributed on download sites, but it was difficult to find any reviews or mentions on blogs or magazines. It has to be assumed that it's not very much used by developers, although it seems like a stable and feature-rich product.

#### 4.5.6 SMARTINSPECT



<http://www.gurock.com/smartinspect/>

##### **Overview**

SmartInspect is a commercial logging tool which comes in configurations for .NET, Java and Delphi. Its vendor, Gurock Software, claims that “hundreds of organizations around the world choose SmartInspect to build rock-solid software applications every day.”

##### **Key features**

- By inserting logging code, developers can log and trace exception traces, messages, objects, files and database records.
- Settings (e.g. log targets and log levels) can be configured and reconfigured without changing source code.
- Tracing of system resources, such as variable data, session data and thread information.
- Remote logging and monitoring via TCP/IP or Named Pipes. Logs from multiple sources can be merged into one log at the vendor site.
- A configuration builder application to build configuration files in a user-friendly way.
- A powerful log viewer, SmartInspect console. Apart from the normal list view showing log messages and toolboxes that show more detailed information about entries, it has many features:
  - A *Watches Graph* that visualizes how variable values change with time.
  - A *Process Flow* toolbox that gives overview of threads and processes.
  - A *Viewer* toolbox that helps to visualize different types of traces (text, XML, binary, datasets, and objects).

##### **SOK acquisition characteristics**

Log messages are generated by inserting logging statements into source code. Log files can be text or binary, unencrypted or encrypted. Logging can also be done to memory, and the log is only dumped to a file if the application crashes. Logging via network can be synchronous or asynchronous.

##### **Market status**

SmartInspect has gotten very good reviews from magazines (e.g. devCity.net, ASP.NET pro, Bitwise magazine), it gets good references on StackOverflow and it is used by large companies like Siemens and Thomson Reuters.

## 5 CONCLUSION AND DISCUSSION

This survey identifies and describes 20 industrially validated tools for acquiring software operation knowledge. In addition to using objective measures (SOK types supported, platforms and popularity) the tools can be compared using the detailed information from the reviews – features, adaptability and market status (beyond the notion “popular”). By comparing tools on this basis as well as total score, these conclusions can be drawn on what tools stand out

- **Crash reporting:** In the crash reporting category, Bugsplat supports the most platforms, is richer in features than the free solutions (Breakpad, Smart Crash Reports and Winqual) and is the only one to offer statistical analysis of aggregate crash data.
- **Feedback tools:** Eventually, only standalone web applications were surveyed for generating end-user feedback. No software vendor explicitly offers end-user feedback functionality that can be woven into application code. Out of the two tools that were surveyed, GetSatisfaction offered a more flexible architecture – its API feature actually allows developers to gather feedback, weave it into application code and submit it to GetSatisfaction.
- **Application monitoring solutions:** These types of solutions combine some of the best characteristics of profilers and tracing tools without requiring much work on behalf of the developer. All the tools in the application monitoring category score high as relevant SOK acquisition tools. AlertSite, GlobalWatch and Precise all gather three types of SOK data, and Glassbox has an impressive troubleshooting mechanism where log message patterns are correlated with a knowledge base of known archetypes of problems. However, out of those four, Precise supports the widest array of platforms, has the best market status and has integration, correlation and analytics features that are unmatched by any of the other tools surveyed.
- **Profilers:** In the profiling category, relatively few candidates made it to the final list due to the fact that profilers that do not support remote or offline profiling cannot be considered SOK acquisition tools, since they are not used in the field except under rare circumstances. YourKit is a popular, mature and feature-rich cross-platform profiler.
- **Tracing and logging tools:** The Apache tools (log4j, log4net and log4cxx) are by far the most popular due to their open source license and flexibility when it comes to output targets and formats. However, SmartInspect is a very powerful cross-platform tool which gathers three types of SOK data and presents it in a helpful way.

An important finding is the absence of a tool that acquires all types of SOK. To develop such a tool, one would have to combine the functionality of an application monitoring tool or tracing tool (with profiling features such as the most powerful ones have) with that of a crash reporter with user messages and/or feedback dialogs to gather end-user feedback samples. One might also look to the end-user feedback mechanism of some of the software update tools on the market, but in that case such feedback is limited to the deployment experience<sup>21</sup>.

There are some limitations to this work. First, generating leads using Internet search and browsing lists and download sites will never reveal all possible tools on the market, i.e., the survey is not complete. This limitation could be overcome by involving more than one researcher independently in

---

<sup>21</sup> Jansen, S., Ballintijn, G., & Brinkkemper, S. (2005). A Process Model and Typology for Software Product Updaters. In *Proceedings of the Ninth European Conference on Software Maintenance and Reengineering* (pp. 265-274). IEEE Computer Society.



lead generation. Second, other researchers may have come up with different scoring criteria, leading to different results from the evaluation. The scoring criteria might be derived in to a more extent by looking at similar models and validated by interviewing experts in software engineering. Third, the reviews only consist of information publicly available on the Internet.

We have two specific suggestions for possible future work related to this survey. First, a systematic comparison of the features of tools which are similar in structure – crash reporters and logging tools, and a presentation of such a comparison in a feature matrix. This would effectively lead to the discovery of “standard” and “value-added” feature sets that would provide input into the work of defining an ideal SOK acquisition tool and evaluating these tools with relation to the definition of SOK and its sub-concepts. This type of work has been successfully applied to similar work on software product update tools<sup>22</sup>. Second, evaluate the tools with the best reviews and highest scores, in order to verify the claims made by the vendors and gain better insight into their capabilities. This, along with questioning vendors regarding any uncertain aspects of the tools would effectively mitigate the third limitation above, the fact that only publicly available information was used to build the tool summaries.

---

<sup>22</sup> Jansen, S., Ballintijn, G., & Brinkkemper, S. (2005). A Process Model and Typology for Software Product Updaters. In *Proceedings of the Ninth European Conference on Software Maintenance and Reengineering* (pp. 265-274). IEEE Computer Society.

## APPENDIX A – LEADS, SCORES AND ELIMINATION

Tool	SOK	Scope	Runtime	Dev	Support	Popularity	Final score	Conclusion	Reason for exclusion or inclusion
Foglight	P,Q,U	3	2	2	1	2	24	Eliminated (review)	Impressive but does only integration and presentation, not acquisition.
Precise	P,Q,U	3	2	2	1	2	24	Final list	
SmartInspect	P,Q,U	3	2	2	1	2	24	Final list	
AlertSite	P,Q	2	2	2	1	2	16	Final list	
Apache Logging	Q,U	2	2	2	1	2	16	Final list	
Bugsplat	Q,F	2	2	2	1	2	16	Final list	
WebMetrics GlobalWatch	P,Q	2	2	2	1	2	16	Final list	
AMD CodeAnalyst	P	1	2	2	1	2	8	Eliminated (review)	Only for AMD processors, limited usefulness for Intel processors.
Gomez Web Perf Mgmt	P,U	2	1	2	1	2	8	Eliminated (review)	Collects no application-specific data, only operational data.
Rational Purify	Q	1	2	2	1	2	8	Eliminated (review)	No performance profiling, and not deployed in the field.
SiteRemark	F	1	2	2	1	2	8	Eliminated (review)	Same features as GetSatisfaction but less mature and popular.
Breakpad	Q	1	2	2	1	2	8	Final list	
EurekaLog	Q,F	2	1	2	1	2	8	Final list	
GetSatisfaction	F	1	2	2	1	2	8	Final list	
Glassbox	P,Q	2	2	1	1	2	8	Final list	
Google Breakpad	Q	1	2	2	1	2	8	Final list	
JAMon API	P,Q,U	2	2	1	1	2	8	Final list	
UserVoice	F	1	2	2	1	2	8	Final list	
YourKit	P	1	2	2	1	2	8	Final list	
DevPartner	P	1	2	2	1	2	8	Eliminated (selection)	Not deployed in the field.
Logback(-Audit)	Q,U	2	2	1	1	2	8	Eliminated (selection)	No convincing argument that it's better than Apache which is more validated.
Nimsoft Monitoring	P,U	1	2	2	1	2	8	Eliminated (selection)	Primarily a network availability monitoring solution.
VTune	P	1	2	2	1	2	8	Eliminated (selection)	Not deployed in the field.
LucidLog.Net	P,Q,U	3	1	1	1	2	6	Final list	

Tool	SOK	Scope	Runtime	Dev	Support	Popularity	Final score	Conclusion	Reason for exclusion or inclusion
DTrace	P	1	1	2	1	2	4	Eliminated (review)	No Linux or Mac OS X port, targeted at whole production infrastructure.
JRat	P	1	2	1	1	2	4	Eliminated (review)	Not deployed in the field.
AQTime	P	1	1	2	1	2	4	Final list	
CodeSite	Q,U	2	1	2	1	1	4	Final list	
iTech Logging	Q,U	2	1	2	1	1	4	Final list	
JProfiler	P	1	2	1	1	2	4	Final list	
Acumem	P	1	1	2	1	2	4	Eliminated (selection)	Only supports Win64, and only low-level languages. No indication of in-the-field use.
Appspy	P,U	2	2	1	1	1	4	Eliminated (selection)	Seems like a weaker version of the web app monitoring tools.
CFC Beta Testing System	F,U	1	2	2	1	1	4	Eliminated (selection)	Standalone and web-based, and only targeted at beta testing.
Coyote Tracing for .NET	Q,U	2	1	1	1	2	4	Eliminated (selection)	A GUI monitor on top of a similar solution as Apache.
DBG	P	1	2	1	1	2	4	Eliminated (selection)	Not deployed in the field.
Eclipse TPTP	P	1	2	1	1	2	4	Eliminated (selection)	Not deployed in the field - part of IDE.
FeVote	F	1	2	2	1	1	4	Eliminated (selection)	Only for ideas / feature requests.
Insure++	Q	1	2	1	1	2	4	Eliminated (selection)	Static analysis (not used in the field).
JProbe	P	1	2	1	1	2	4	Eliminated (selection)	Not deployed in the field.
Log4Cpp	Q,U	2	2	1	1	1	4	Eliminated (selection)	Not nearly as popular as the Apache version.
MS Enterprise Library	Q	2	1	1	1	2	4	Eliminated (selection)	Not mature or popular, same scope as for Apache which is industry leader.
NetBeans Profiler	P	1	2	1	1	2	4	Eliminated (selection)	Not deployed in the field - part of IDE.
Exception Reporter	Q,F	1	1	1	1	2	2	Eliminated (review)	Least popular out of the four Windows crash reporters.
Zoom	P	1	1	1	1	2	2	Eliminated (review)	Limited platform (only Linux) and does not seem to be deployed in the field.

Tool	SOK	Scope	Runtime	Dev	Support	Popularity	Final score	Conclusion	Reason for exclusion or inclusion
Smart Crash Reports	Q	1	1	1	1	2	2	Final list	Unique tool, popular for Mac applications.
Ajax View	P,U	2	1	1	1	1	2	Eliminated (scoring)	
Ants Performance Profiler	P	1	1	1	1	2	2	Eliminated (scoring)	
BugTrap	Q	1	1	2	1	1	2	Eliminated (scoring)	
CrashRpt	Q	1	1	1	1	2	2	Eliminated (scoring)	
dotTrace	P	1	1	1	1	2	2	Eliminated (scoring)	
FeedbackReporter	Q,F	2	1	1	1	1	2	Eliminated (scoring)	
GlowCode	P	1	1	1	1	2	2	Eliminated (scoring)	
JIP	P	1	2	1	1	1	2	Eliminated (scoring)	
JMeasurement	U	1	2	1	1	1	2	Eliminated (scoring)	
LTPProf	P	1	1	2	1	1	2	Eliminated (scoring)	
OKTECH Profiler	P	1	2	1	1	1	2	Eliminated (scoring)	
Performance Validator	P	1	1	2	1	1	2	Eliminated (scoring)	
Profiler4J	P	1	2	1	1	1	2	Eliminated (scoring)	
Shark	P	1	1	1	1	2	2	Eliminated (scoring)	
smartassembly	Q	1	1	1	1	2	2	Eliminated (scoring)	
VSTS Profiler	P	1	1	1	1	2	2	Eliminated (scoring)	
ProfileSharp	P	1	1	1	1	1	1	Eliminated (scoring)	
SamplingProfiler	P	1	1	1	1	1	1	Eliminated (scoring)	
SlimTune	P	1	1	1	1	1	1	Eliminated (scoring)	
Valgrind	P	1	1	1	1	1	1	Eliminated (scoring)	
Featurelist.org	F	0	2	2	1	1	0	Eliminated (scoring)	
GetSnappyNow	P	0	0	0	0	0	0	Eliminated (scoring)	
InfraRED	P	1	2	1	0	1	0	Eliminated (scoring)	
inTrace	Q,U	2	1	0	1	1	0	Eliminated (scoring)	
JMeter	P	0	2	1	1	1	0	Eliminated (scoring)	
NSpring	Q,U	2	2	1	0	1	0	Eliminated (scoring)	
Optimizelt	P+	1	1	1	0	1	0	Eliminated (scoring)	
Parallel Studio	Q	0	2	1	1	2	0	Eliminated (scoring)	
Statement Tracer	Q,U	2	2	0	1	1	0	Eliminated (scoring)	
SuggestionBox	F	0	2	2	1	2	0	Eliminated (scoring)	
Thread Profiler	P	0	1	1	1	1	0	Eliminated (scoring)	
VBA Code Profiler	P	1	1	0	1	1	0	Eliminated (scoring)	
Web Performance Monitor	P	0	1	1	1	1	0	Eliminated (scoring)	