

2nd Annual Exception Analytics Survey: 2011

Development recognition of the importance of exception analytics has increased 50% year over year and the percentage of companies within this group who are “at risk” because they do not have access to this vital information has doubled.

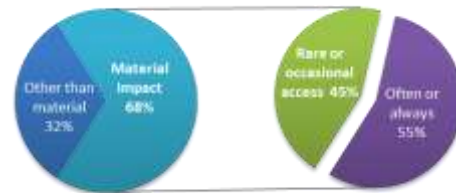
Abstract

During the week of October 10, 2011, PreEmptive Solutions conducted its second annual survey to capture best practices in the use of software exception analytics in improving software quality and customer experience. The study incorporates over 300 responses from 222 companies across 56 industries and 40 countries.

The resulting data reveals an exploding “practice gap” where development organizations’ practices are out of alignment with their stated principles and priorities. This gap was identified in the 2010 survey but has widened dramatically. Specifically, the percent of developers that recognize the materiality of exception analytics has increased by 50% year over year and the percentage of those companies who only rarely or occasionally have access to this self-described “material” data has doubled.

2010 Respondents

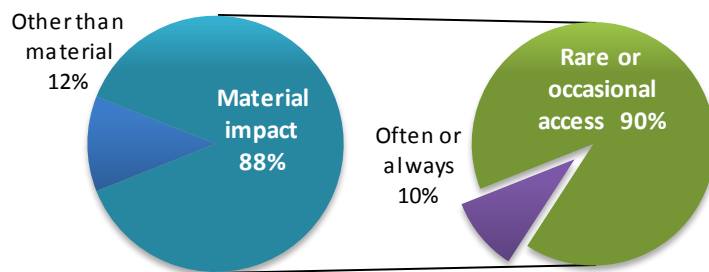
In 2010, of the 68% of respondents who indicated that exception analytics offered material improvements to software quality and user experience, 45% “rarely” or only “occasionally” have access to this “material development resource.”



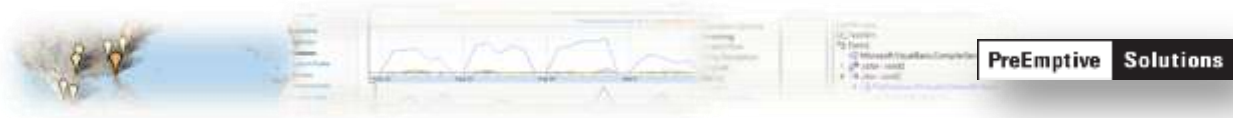
2010 Exception Analytics "Practice Gap"

2011 Respondents

In 2011, of the **88% of respondents** who indicated that exception analytics offered material improvements to software quality and user experience, **90% “rarely” or only “occasionally”** have access to this “material development resource.”



2011 Exception Analytics "Practice Gap"



The proliferation of software platforms, surfaces, and distributed computing architectures is the primary force behind the growing chasm between actual and best practices.

Motivation: exceptions are the rule

Identifying and remediating defects “in the wild” is an inescapable *and expensive* part of the software development process. The damage that a failed production application causes is real, time to repair is typically long, and the skills and roles required can be extensive.

Errors reported by users are limited to descriptions of external symptoms, whereas systems and processes that automate the aggregation and analysis of exception reports (that may also include user comments) can, in principle, efficiently and consistently capture and manage detailed runtime data across large populations of users. In one well-documented case, bugs reported to Microsoft through their Windows Error Reporting (WER) system were “4.5 to 5.1 times more likely to be fixed than a bug reported directly by a human¹.” Conversely, without such an infrastructure and process, diagnostic information becomes scarce, is often limited in scope, inefficiently delivered, and expensive to manage.

The proliferation of software platforms, surfaces, and distributed computing architectures has only served to exacerbate these challenges at a time when software quality and customer experience has never been more critical.

This survey was designed to assess the perceived value of exception analytics and to measure the extent to which exception analytics has been incorporated into application development lifecycle and support practices.

2011 addendum

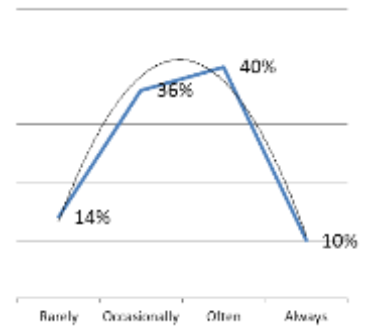
While the coming wave of new platforms, technologies and business drivers were anticipated in 2010, very few would have forecasted the speed and impact that cloud and mobile computing would have in twelve short months. As the following data clearly shows, the dramatic expansion of the gap between best and actual development practices with regards to exception analytics is being driven by an increasingly diverse and distributed computing environment.

¹ Debugging in the (Very) Large: Ten Years of Implementation and Experience, Kirk Glerum, et al, Microsoft Corporation, 22nd ACM Symposium on Operating Systems Principles, October 11-14, 2009.

Practice: Effective exception analytics management and response

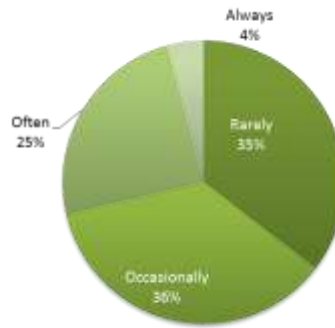
The survey distinguished between exception analytics stemming from thrown, caught and unhandled exceptions.

The frequency of caught and thrown exception data collection falls into a near perfect bell curve distribution. Whereas unhandled exception data collection frequency tells a very different story.



Caught and thrown exception access

Exception reporting is not yet a common practice and standardized and automated responses are rare.

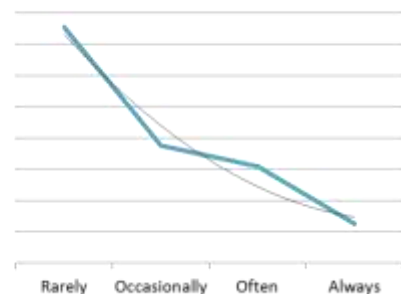
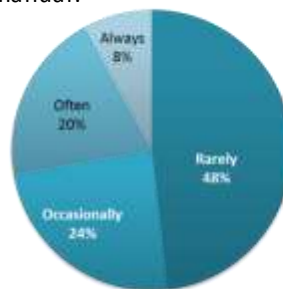


Unhandled exception access

71% of respondents indicated that they either rarely or only occasionally have access to exception analytics stemming from unhandled exceptions (only 4% "always" have access to unhandled exception analytics).

Incident response is rarely standardized or automated

To the extent that organizations do have access to production exception reporting data, the associated incident response processes are either ad hoc or manual.



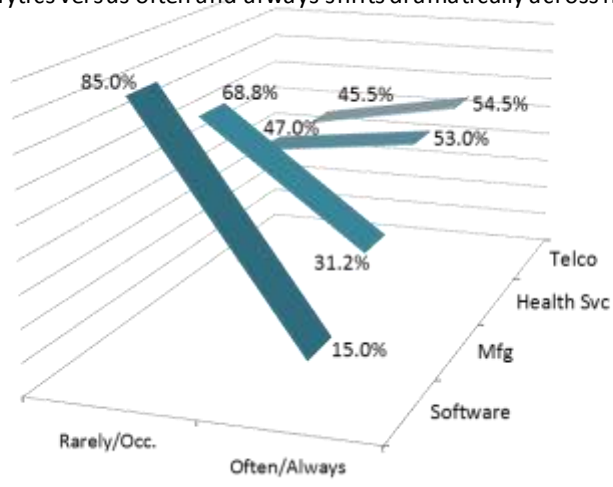
Automated generation of work items from exception "incidents"

72% of respondents either rarely or only occasionally benefit from automated production incident response systems.

There are industry-specific common practices for exception analytics and incident response demonstrating both unique requirements and differing degrees of “operational maturity.”

Unhandled exception analytics practices by industry

The split between rare and occasional access to unhandled exception analytics versus often and always shifts dramatically across industries.



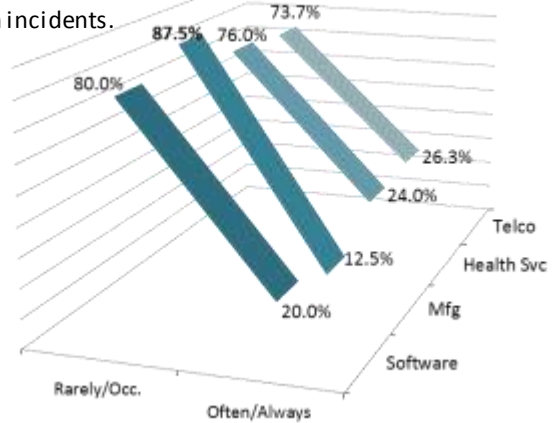
Access to unhandled exception data

Only 15% of ISV respondents (software vendors) and 31% of manufacturing firms indicated that they either “often” or “always” had access to unhandled exception analytics. Conversely, the majority of Health Service organizations and Telecommunications respondents indicated that they had such access (53% and 55% respectively).

Is exception monitoring and automated response simply not important enough to implement on a consistent basis or does this represent an “unmet material requirement?” (Operational gap)

Exception incident response automation and prioritization is immature

However, the segmentation of practice by industry does not seem to have emerged at this time for the automated prioritization and response to production incidents.



Automated response and prioritization of production incidents

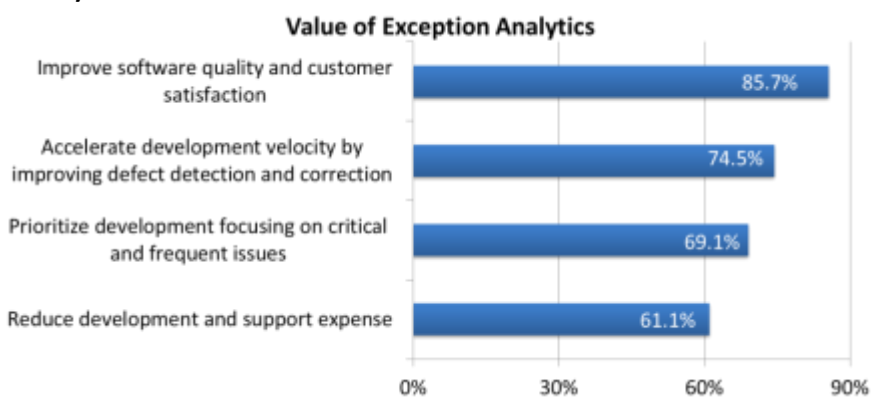
This begs the question, “is exception analytics and response simply not important enough to implement on a consistent basis or is this an “unmet material requirement” (operational gap)?

Principles and Priorities: Exception analytics materiality

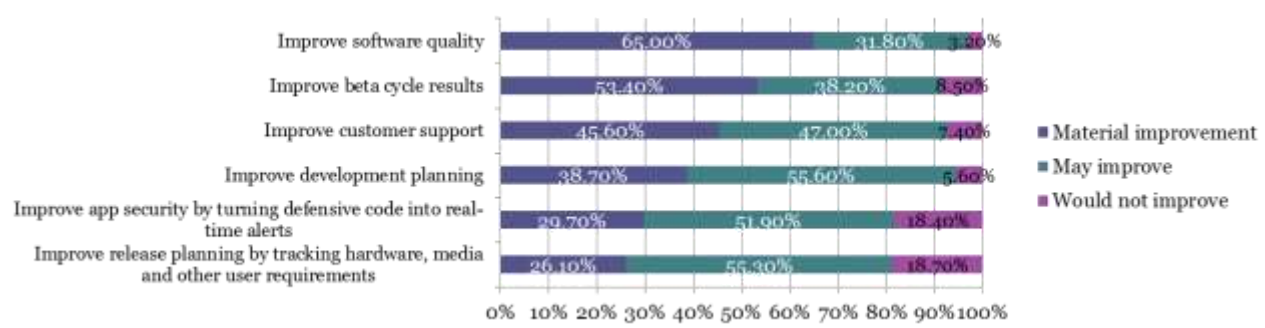
Principles

Effective exception analytics “materially improves” software quality, customer satisfaction, developer efficiency and focus.”

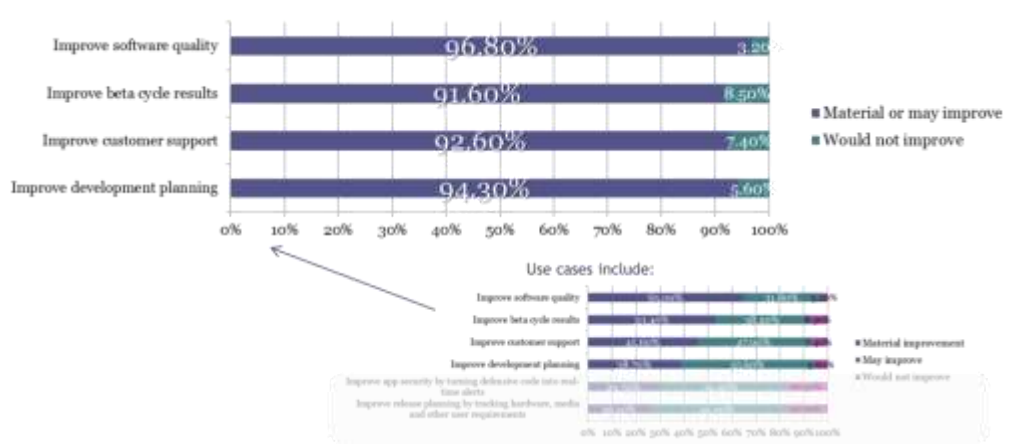
99.7% of respondents identified one or more of the following exception analytics benefits:



Further, respondents indicated the following specific use case scenarios where application analytics may add material value to the development process.



When Material and potential for material improvement rankings are combined, more than 9 out of 10 of all respondents agreed that the top four use cases hold the promise of material development advantage.





Development risk is proportionate to untapped development gains and opportunities.

Respondents who are at risk know that they are at risk.

The larger the user base, the greater the development risk.

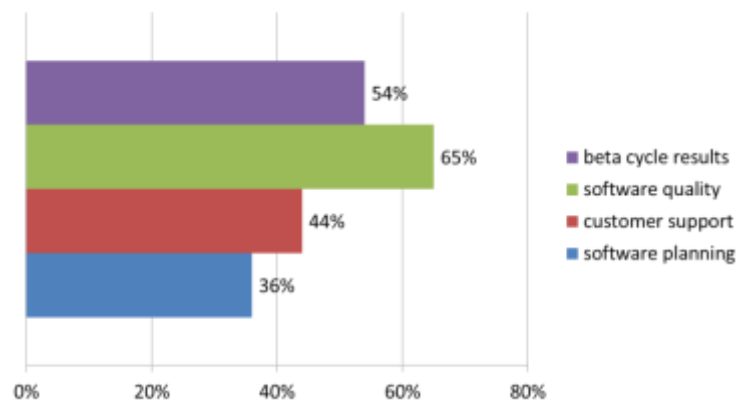
The Practice Gap: Material risks stemming from misaligned exception analytics policies

Inefficient development practices put both development organizations and their users at risk. Clearly, organizations that identify exception analytics as adding material value but do not actually leverage that resource are, by their own reckoning, at risk.

Organizations at risk

79% of respondents explicitly identified exception analytics as offering material improvements AND, at the same time, admitted to having little or no access to exception reports and associated forensic data. *These organizations, by their own admission, are "at risk."*

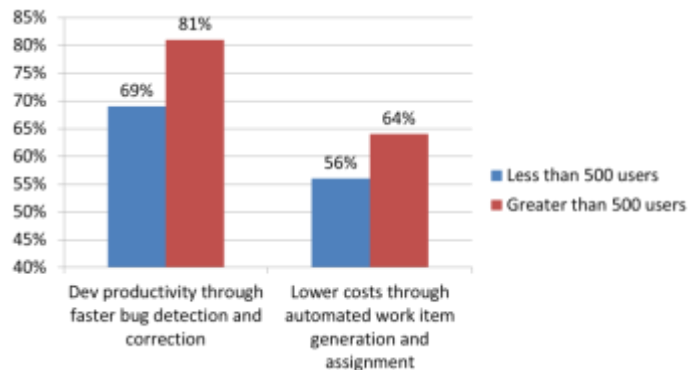
Focusing only on the 79% of development organizations who self-identified as being at risk, they identified the following four scenarios where access to exception analytics would add material value. This is untapped potential value or, conversely, this represents present-day development and operational risk.



Use case scenarios identified by development organizations "at risk"

The larger the user-base the great the risk

In general, as installations and users increase the greater the need (and the potential risk) for exception analytics. When looking at the relative value that respondents attached to developer productivity and cost reduction, there is a clear difference between those with less than 500 users and those with over 500.





*A lack of best practices
and access to
underlying technology
is at the heart of
today's "practice gap."*

Conclusions

The survey results strongly suggest the following conclusions:

- Automated exception analytics are widely recognized as adding material value to software quality and customer experience.
- A significant segment of the development community has little or no access to this valuable resource.
- Unhandled exception reporting and analytics is especially valuable and poorly managed.
- Organizations that are at risk know that they are at risk.

These trends and requirements have become significantly more critical and urgent over the past twelve months. This is due, in large part, to the expanding suite of deployment platforms and devices and the increasing penalties for commercial failure.

Questions left unanswered

WHY? What is impeding these "at risk" organizations from gathering and organizing exception reports and analytics when they recognize their importance?

WHO? While this report focused primarily on the sub-segment of "at risk" organizations that explicitly identified a gap between their principles and their practices, many of the other respondent organizations may also be at risk if they have not institutionalized best practices in the areas listed here.

What can your development team do?

Establish an exception analytics policy: ensure that the exception analytics policy is explicitly scoped, implemented, and tested. Do not accept a de facto policy born out of passive neglect or omission.

Automate and institutionalize: implement a technology and process framework that bridges the development and support environments with the end-users' runtime environment.

Contact PreEmptive Solutions: for more information on how PreEmptive Solutions is helping organizations implement and automate software exception analytics and incident response solutions, visit www.preemptive.com or email sales@preemptive.com and request a personalized demonstration.