Ben Kepes

Righting the Wrongs

Error Reporting as a critical part of Enterprise IT

RAYGUN.IO

Introduction

As the world becomes ever more competitive, organizations are seeking ways to differentiate. Delivering data and applications to internal and external stakeholders becomes a central part of what organizations do. Increasingly, organizations of every type will start to think of applications as a critical component to their continued growth and success.

As organizations make this change to being driven by technology, large portions of their operations will need to begin to look and act like technology companies. Organizations need to consider what it means to become a technology company and arm themselves with the tools that allow them to operate like Google, Yahoo or Netflix.

Organizations need to consider what it means to become a technology company and arm themselves with the tools that allow them to operate like Google, Yahoo or Netflix.

The obvious difference between these organizations and other ones is that these "webscale" vendors have the resource to build out their own teams to fulfil all the different functional areas a technology company has.

The challenge for smaller organizations is to get this same level of quality, on a smaller resource pool. This is where developer tools come in, and we see an increasing number of these tools designed to be acquired and used as a service.

Ensuring visibility over software performance generally, and application errors specifically, is one of these areas, and it is an important part of ensuring consistent performance for application users.

In this report, we will look at why error tracking is an important area for the development teams within an enterprise to check off.

Modern approaches towards building, deploying and running software

Organizations are increasingly forced to rethink what they do in the face of both internal and external changes. There are a number of mega-trends that are fundamentally changing the environment within which organizations operate. Economic changes, the rise of the millennial generation, ubiquitous internet connectivity, the demand for remote working and the rise of disruptive players in every sector all create a "perfect storm" for organizations.

We contend that every industry, sector and organization is sitting on top of a burning platform, and needs to fundamentally reinvent what it does. Tinkering around the edges simply won't suffice anymore, and a total rethink of the status quo is necessary.

Fortunately, however, alongside the fundamental challenges that organizations face there is the increasing quality, flexibility and ease of creating software. Whereas software development was, in the past, the bastion of only the largest of organizations, we have seen a total democratization in recent years.

These changes, the rise of readily accessible development, alongside the drive for innovation and agility, has changed the world of software development. New languages, a plethora of different modular development tools and cloud platforms all make the creation of software easier than ever before.

But while software development is ever-easier, there is a somewhat counter-intuitive need to revisit the way software organizations work in order to best leverage these tools. While modular developer tools

have made customizing an IT stack far easier than ever before, this added complexity introduces some new challenges to organizations in terms of gaining visibility over all those different parts.

We are seeing the breaking down of the traditional, monolithic and generally static approaches towards building software. In their place we see distributed applications and a myriad of different tools being used to build, deploy and manage the variety of applications in existence.

The way software is created and run is different from ever before, this is hugely enabling but means that organizations need to rethink the tools and approaches they use towards their software.

Disruption in action – software as the catalyst for change

All too often we, as consumers, use new services with abandon but don't take the time to really consider their existence within the context of what went before. Services like Netflix, Airbnb and Uber are good examples of this – we use these services regularly, but without taking a minute to think about what they mean for the economy.

We contend that all three of these examples, and many more besides them, are examples of using software to disrupt an existing industry. Uber disrupts traditional taxi companies, Airbnb impacts upon the hotel industry and Netflix and its ilk crush the traditional movie distribution models.

All three of these companies share one thing in common – they look to software to be the catalyst with which they disrupt the existing player. Uber owns no cars, Airbnb has no hotel inventory of its own and you don't see Netflix stores on every street corner – rather they look to software as the distribution mechanism for what they do.

Uber is often held up as the exemplar of this model and, in the context of this paper about a specific developer tool, it is perhaps the best one. Uber is, at its essence, an aggregation of a number of different aspects. Of course primarily it is about aggregating passenger demand with driver availability. But from a software perspective it is something very interesting. Uber makes use of a number of discrete software offerings – a payment service, a mapping solution, mobile application platforms and the like are all glued together to create the Uber applications.

It cannot be overstated just how much of a revolution this is. In the past, if we look at the organizations that created the biggest impacts on society, they were organizations with the ability to make massive capital investments. Look to the railroad companies or shipping lines as examples. Today however it is

by having a smart idea, leveraging a range of different services and rapidly iterating on a concept that organizations can prove disruptive to incumbents.

For all of these organizations, software is the enabler for transformation. We contend that, in this regard, they are in no way outliers but rather good examples of the way commerce will happen in the future. By leveraging technology, organizations can either displace incumbent players or avoid disruption themselves. If of course, the software is delivered to end users reliably, without them running into problems.

Critical factors in ensuring effective applications

Uber and its ilk have shown us the power that a software-enabled enterprise can bring to the world. As more organizations move to this model, we need to consider what enterprise IT organizations should think about when making their application decisions. In a world where consumers are fickle, and competition is rife, application efficacy becomes ever more important.

Mobile Isn't Everything

Organizations would be misguided, however, to consider their application needs purely in the context of mobile. While mobile is certainly a huge and rapidly growing area, the general internet is still massive and growing similarly rapidly.

Organizations need to consider their application's performance across both mobile, and the web – application monitoring tools need to cover both delivery modalities.

It's no use being mobile only, or only relevant to the Apple ecosystem – software teams need tools that embrace the totality of their needs.

The Future Is Diverse

As we have pointed out previously, traditional IT organizations would generally be built around one or two monolithic technology stacks. This approach extended to programming languages where organizations would decide on a couple of languages to cover all of their needs.

The burgeoning number of different languages, along with the specific use cases that different languages fulfil results in a huge increase in the number of languages an organization uses. Hence there is a need for their monitoring solutions to cover a broad range of languages.

Multi-language support, and a commitment to extending support to new and emerging platforms is an important part of monitoring solutions.

Why is error tracking important?

One of the components that the development and operations teams within an organization need to think about is error tracking. While error tracking is but one of the multitudes of functional areas to be covered off, it is an important one, since software errors directly impact upon end users.

When thinking about the tools to be used within an organization, we contend that the problem areas that have a direct impact on customer experience should be prioritized.

Software errors not only damage the brand through reduced customer experience, but they also have a significant flow-on effect in terms of wasted developer time. As such, having a good handle over the totality of the errors generated by an organization's software spread is important.

Within large webscale teams, there is sufficient resource to dedicate specific teams to the error tracking function. Google, Yahoo, and the other webscale vendors have the scale to build out around the clock error-tracking teams as part of their broader monitoring resource.

It should be obvious however, that only the largest of organizations can justify building out such a specific team, and this is where individual developer tools come in. By utilizing a third party error-tracking solution, organizations and their developer and operations teams can have the kind of visibility into their errors that were formerly only available to the largest organizations.

These sort of developer tools, available on-demand and as a service, are democratizing the development and operations roles and are helping even small organizations deliver software of the robustness that we see from webscale vendors.

Errors Cost Money

Software errors cost money to fix. With development time and resource being

constrained, it is important that monitoring solutions give enough context, and in a timely manner, to help development teams identify, fix and deploy bug fixes.

This “need for speed” is especially so in larger organizations where the prevailing threat they face is disruption by a smaller, more agile, team. Tools, therefore, that help larger organizations mimic the agility of smaller ones are increasingly of value.

As organizations look to marketing as their primary way of attracting customers, the corresponding spend on marketing initiatives is huge. These initiatives often include software. To invest hugely in a marketing initiative only to have the customer leave the sales funnel due to a software fault is problematic. Thus effective monitoring, and the ability to respond to the insights identified, help to justify the investment in marketing campaigns and help to ensure a return on that investment.

Error tracking as an enabler of agility

In recent years, there has been much attention given to DevOps, a term used to cover the trend towards an increasingly close relationship between developers and operations teams. DevOps allows software teams to be more agile and iterative and allows the organization to innovate more readily with software.

In order to more fully move to a DevOps model, developers need to have better visibility over the production behaviors of their applications. No longer is it viable to simply pass code over to operations teams and forget about it.

Developers need mechanisms and tools whereby they can be constantly aware of the issues around their software – this need is increasingly complex as organizations move to distributed software stacks and multi-tier applications.

Developers fundamentally want to create performant applications, but need visibility over the operational characteristics of those applications to do so. This is where developer tools that tie together operations and developers come in – error tracking is one example of that in action.

Summary

As businesses increasingly look to software to deliver competitive differentiation, organizations are having to think about how they can build, run and manage software with as much agility as possible.

Organizations need to do this within the context of an increasingly complex and heterogeneous IT footprint.

While application monitoring is just one part of a much larger enterprise IT stack, it is of high importance given that it directly impacts upon end-users experience using software.

Rather than simply a tool to "ensure the lights stay on", application monitoring is a critical step in the organizational value chain – as important as any other part of the supply chain.

It is for this reason that organizations should think deeply about their approach to application monitoring, and choose a solution that covers as broad a footprint as possible to ensure room for future growth. This applies to the error tracking space and organizations should consider error tracking as a key requirement for their developer and operations teams.

About the author

Ben Kepes

Ben Kepes (Diversity Limited) is a technology evangelist, an entrepreneur, a commentator and a business adviser. Ben covers the convergence of technology, mobile, ubiquity and agility, all enabled by the Cloud. His areas of interest extend to aviation technology, enterprise software, software integration, financial/accounting software, platforms and infrastructure as well as articulating technology simply for everyday users.

He is a globally recognized subject matter expert with an extensive following across multiple channels.

Ben currently writes for Forbes and his own blog. His commentary has been published in ReadWriteWeb, GigaOm, The Guardian, and a wide variety of publications – both print and online. Often included in lists of the most influential technology thinkers globally, Ben is also an active member of the Clouderati, a global group of Cloud thought leaders and is in demand as a speaker at conferences and events all around the world.

As organizations react to the demands of more flexible working environments, the impacts of the economic downturn and the existence of multiple form-factor devices and ubiquitous connectivity, Cloud Computing stands alone as the technology paradigm that enables the convergence of those trends. Ben's insight into these factors has helped organizations large and small, buy-side and sell-side, to navigate a challenging path from the old paradigm to the new one.

Ben is passionate about technology as an enabler and enjoys exploring that theme in various settings.

About Raygun

Raygun Inc is the market leader in error tracking and crash reporting tools for software developers. Raygun's powerful yet easy-to-use error tracking solutions transform how software teams of all sizes work – and work together – to drive dramatically lower software error resolution times and associated costs. The company's proven technology, comprehensive services and expert tools are helping more than 1,000 companies around the world to turn software errors from a costly process to a revenue driver.

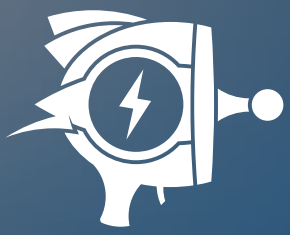
For more information, visit **www.raygun.io**, or subscribe to Raygun's blog at **www.raygun.io/blog**

U.S.A Office

Raygun Inc
C/O, 1510 Page Mill Rd
Suite 110
Palo Alto, CA 94304

NZ Office

Raygun
175A Cuba Street
Te Aro
Wellington 6011



Raygun helps you build better software

Get started for free at **raygun.io**

RGBK-201505-1