Community Research Report

in association with

and

Optimizing the software supplier and customer relationship

Insights and lessons from the sharp-end

Freeform Dynamics, 2018

Management Summary

Few would argue that software is now an integral part of the business landscape, but just how critical is it, and what are the consequences of it not running reliably? These questions and others were addressed in a recent study by Freeform Dynamics in which feedback was gathered from the IT vendor and enterprise customer communities. The results shed light on the importance of software reliability, as well as identifying strategies and tactics for minimizing software-related downtime. The study also provides insights into how suppliers and customers often need to work together more effectively in this area.

KEY FINDINGS

Applications you can rely on are critical in today's digital world

Software reliability and availability are key to the successful running of a modern business. This view emerged almost unanimously (97% agreement) when 195 software makers and 159 mainstream enterprises provided feedback during a recent study.

Poor software reliability translates to a tangible business risk

Over 95% of both software makers and enterprises regard the disruptive potential of a software failure and/or data corruption as significant (rating of 4 or 5 on a 1-5 scale). Furthermore, over 80% highlight the way in which even intermittent problems can disrupt the running of a business. Such views carry weight as all respondents had experienced software failures over the past 12 months.

When things go wrong, the big enemies are delay and uncertainty

Over 9 out of 10 enterprise respondents (93%) say that business stakeholder and user satisfaction are undermined when software issues '*drag on, while no one can explain the cause*'. Conversely, when issues are '*diagnosed and fixed swiftly and efficiently*', the majority (84%) say the impact on business satisfaction is actually positive.

Despite the diagnostic imperative, it's hard to troubleshoot production systems

Over 90% of enterprise and supplier respondents (94% and 92% respectively) agree that troubleshooting software issues in a complex production environment can be hard. The majority highlight that this is because it's not just about the software, but the conditions under which it is running when the problem occurs. Recreating runtime conditions can be particularly difficult.

Too many issues are left undiagnosed, then they later come back to bite

The majority (91%) of enterprises say they have experienced production failures in the last year because of previously seen but so far undiagnosed problems. And it doesn't help that risks often originate from the supplier's environment, with 1 in 5 software makers (82%) reporting defects associated with undiagnosed test failures causing production problems within their customer base.

Advanced diagnostic solutions can help in both development and production

Continuous integration and continuous testing provide a foundation for software quality. Systems monitoring and log analytics are then the starting point for troubleshooting production systems. Recently, however, 'record and replay' solutions – akin to aircraft black-box flight recorders – have emerged to help speed and streamline diagnostics in complex environments. While knowledge and awareness are limited, two thirds of respondents acknowledge a role for this kind of technology.

But it takes two to tango - people and relationships are important too

Beyond technology and process, our study revealed that customers and suppliers often need to understand each other more and develop better working relationships. A partnering rather than transactional mindset is required, and both parties must make the effort. When relationships break down, modern, open architectures which lower barriers to switching makes divorce more likely. Read on for more insights into this, along with a deeper dive into other key findings.

Introduction

About this Document

The insights presented in this document are derived from a research study designed and executed by Freeform Dynamics. Intelligence was gathered via an online survey from 354 respondents drawn from a mix of software companies and mainstream enterprise organisations. The study was sponsored by Undo.

Trucks were filling the loading bays, and the loading yard, and the road outside, and the police had stepped in to redirect the traffic. Why? A crashed ERP system. The 'PANIC' button was pressed, and the software vendor sent out two consultants at great expense to troubleshoot. The problem was fixed, but the incident cost the business tens of thousands of dollars.

This is one of many stories we heard from respondents in a recent research study looking at software reliability in production environments. That particular failure, while costly, painful and embarrassing for the business, did lead to at least one good outcome. The person involved ended up getting a promotion because of the way they handled the situation. Had they not acted so quickly to identify the nature of the problem and the need to leverage the supplier's resources, things could have been a whole lot worse. It clearly wasn't good that the crash occurred, but software failures are inevitable in even the best of systems. It's what happens then that really matters. It's an important aspect of overall software reliability that is often overlooked.

A lot of the other tales we heard reinforced this principle. Examples ranged from emergency calls failing in a telecom network due to a software fault, leading to lives put at risk, through discount calculation errors in a hotel booking system resulting in significant reputational damage and financial loss, to intermittent but persistent software problems disrupting employee productivity and satisfaction. The number and diversity of incidents reported by the 354 study participants are far too great to cover in detail here. Time and again, however, the importance of responding effectively to failures came through, as did the consequences of not doing so.

Software failures are inevitable, even in the best of systems. It's what happens when they occur that really matters.

In the remainder of this paper we'll be exploring some of the practicalities in this area, particularly looking at how well software makers (55% of our sample) and their customers (45% of our sample) work together when things go wrong. Our aim is to highlight some of the key challenges, but as importantly to provide ideas on how things might be improved.

You probably don't need telling, but…

Stop press! Breaking news … software is now critical to the majority of businesses. OK, so we jest, but when you have your head down dealing with the cut-and-thrust of software engineering, systems support, or whatever other aspect of software delivery you work on, it's easy to lose sight of how much of what you do really matters (Figure 1).

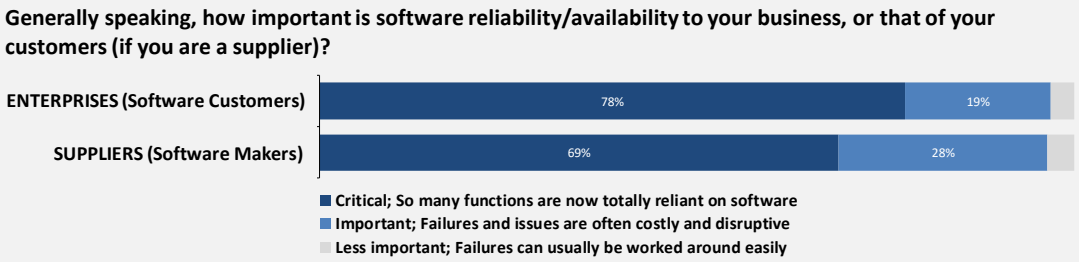


Figure 1 Software is now critical to the majority of businesses

While those working in a customer environment are on average a little more likely to appreciate the criticality of software, the general consensus is clear.

As we have heard anecdotally already, it follows from the above that when software failures occur, the disruptive potential is significant, and this doesn't just apply to major incidents. If lack of software reliability means that even less critical issues persist, the business can still suffer disruption. Again, suppliers and customers largely agree on this (Figure 2).

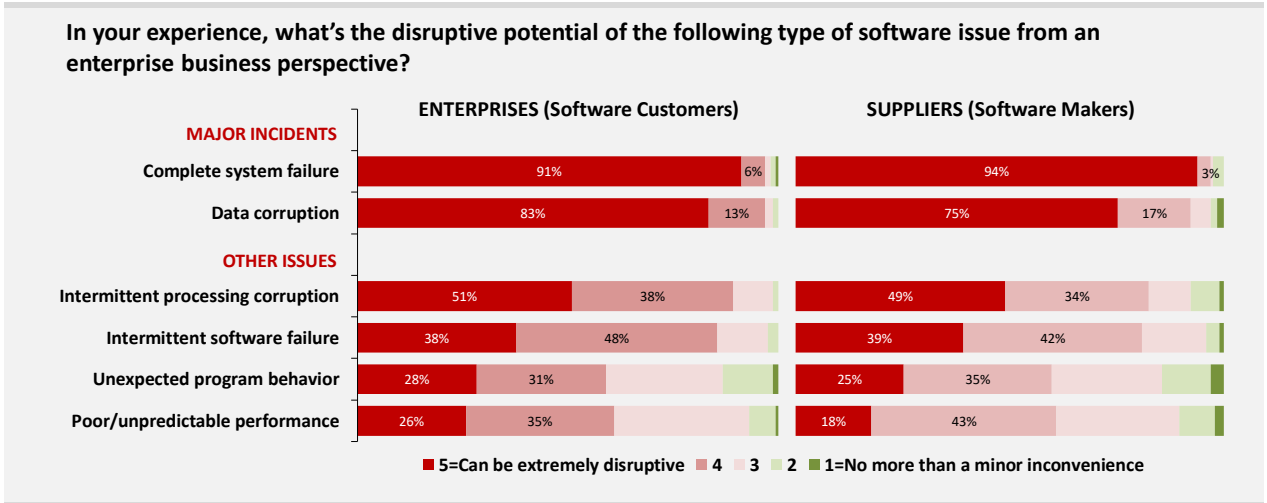


Figure 2 It's not just major incidents that can cause significant disruption

And when we said that software failures are inevitable, this too is confirmed by the data. Every respondent reported seeing at least one category of issue over the past 12 months – whether directly in the case of enterprises, or across their customer base in the case of suppliers. Furthermore, as the spread of issues illustrates, the majority have clearly had to deal with problems in multiple categories (Figure 3).

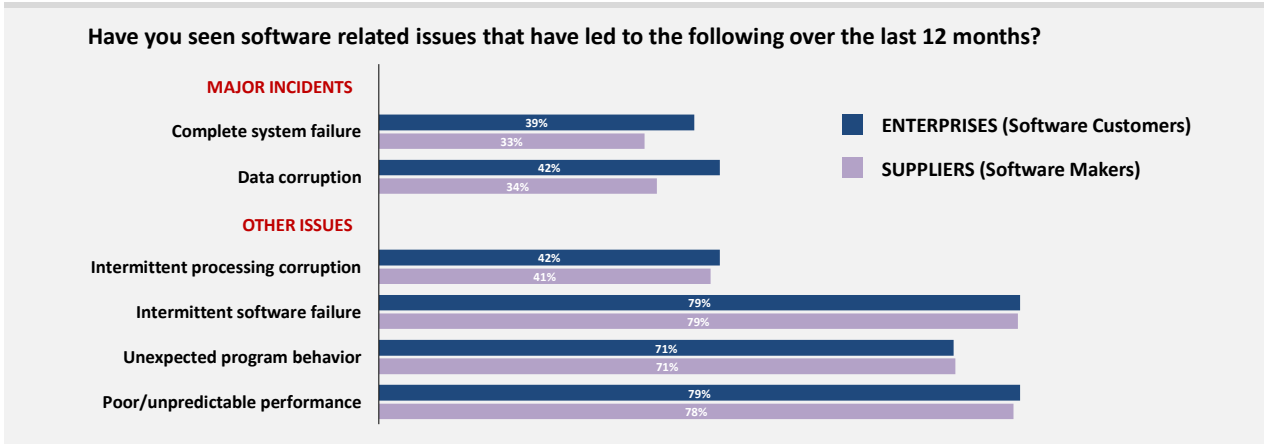


Figure 3 Failures are inevitable, even with the best software

Having covered the basics of why our discussion matters, let's drill a little deeper into the response part of the equation.

The stakeholder and user perspective

The big enemies of stakeholder and user satisfaction are delay and uncertainty.

The premise of this paper is that internal IT and application teams must work together with software suppliers in order to deliver a good service to the business, and particularly to respond effectively to software-related incidents. When considering performance in this context, the impact of response behaviors and outcomes on stakeholder and user satisfaction provides a convenient empirical indicator.

The study tells us that the big enemies here are delay and uncertainty, closely followed by perceptions of IT not being in control of suppliers, and users being forced into disruptive and inconvenient workarounds. What some refer to as 'death by a thousand cuts' is up there as a prominent software reliability related problem too, i.e. users' patience being worn down by a constant stream of irritating minor issues (Figure 4).

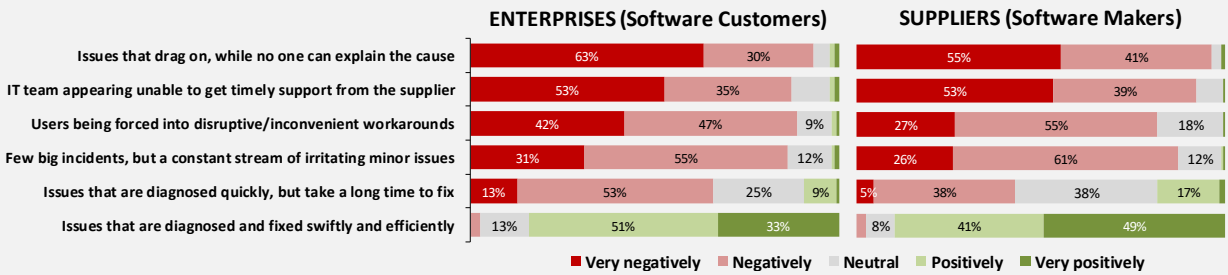


Figure 4 Stakeholder and user satisfaction are important measures of success

While enterprises and suppliers again largely agree, a notable difference between the two is in relation to workarounds. This reflects the fact, for example, that what might seem simple and straightforward from an engineer's perspective to get a particular operation functional again, can translate to a big burden on a user if they are performing that operation tens or hundreds of times per day.

The impact on business stakeholders and users is actually net positive when issues are diagnosed and fixed swiftly and efficiently.

Another difference between the two camps can be seen in relation to issues that are diagnosed quickly, but take a long time to fix. Fast diagnosis significantly lessens the chances of a negative impact, but the data suggests that this isn't enough. From the customer's perspective, pinning down the cause of the problem is seen as just a stepping stone to an ultimate resolution, not a significant achievement in its own right. Closing a case with an escalation to engineering or an email with a link to a workaround might look good in terms of support team metrics, but it doesn't mean the customer is happy.

Turning to the bottom part of the above chart, indications are that the impact is actually net positive when issues are diagnosed and fixed swiftly and efficiently. This further reinforces our thesis that, within reason, it's not software faults per se that are the cause of customer woes, but failure to deal with the issues quickly and to the customer's satisfaction. And from the supplier perspective, this isn't just about avoiding negative results on customer surveys and preventing negative net promoter scores, it's a hard-core business issue as the next set of results illustrates.

Supplier switching / customer churn

Pulling together some of the issues we have already discussed with a range of what we might call 'relationship management' factors, it's clear that customers aren't afraid to switch if software reliability is lacking and suppliers persistently fail to respond appropriately (Figure 5).

If the following occur repetitively, how likely are they to influence a decision to switch to an alternative supplier/solution [Enterprises] OR contribute to a customer decision to switch to a competitor [Suppliers]?

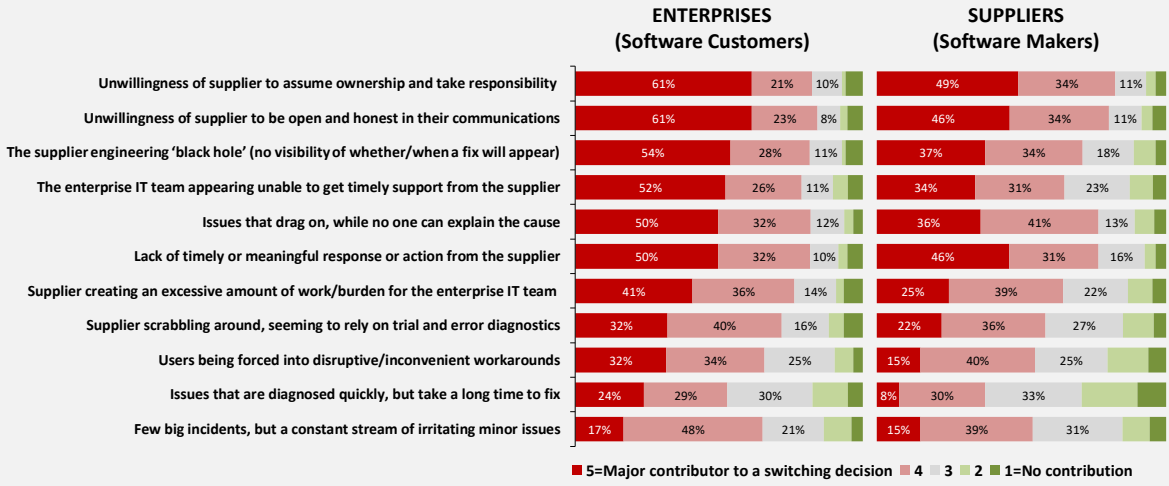


Figure 5 Repetitive bad experiences will precipitate a switch in supplier

For the avoidance of doubt, we are talking about repetitive behavior here rather than one off incidents. If you consider it as a points-system of sorts, the more red you see against a transgression on the above chart, the more the customer will be driven towards a switching decision if you continually fail as a supplier in that area. Repeated problems in multiple areas will obviously accelerate things towards such a decision.

Switching decisions are often down to simple economics.

During the study, we asked enterprise respondents for relevant examples of switching decisions, and again participants were very forthcoming. Here are some representative quotes from their responses illustrating that beyond the disruption factor, it's sometimes a case of simple arithmetic:

> "We switched when continued issues meant our own IT costs to maintain and support the software exceeded what we were paying the supplier."

> "Instability and data corruption issues eventually required us eating sunk costs and buying a different solution."

Easier switching and substitution means delivering good service is more critical than ever.

> "Although issues all had work arounds, they were constant and repetitive, and required a lot of extra IT support."

> "In short, we're happy to move vendors if needed to get good service."

That last comment is pertinent when you consider the ongoing trend towards open architectures and API-enabled integration. Easier switching and substitution means delivering good service is more critical than ever.

It takes two to tango

During our research at Freeform Dynamics, we don't just hear stories from customers about suppliers, we are also privy to the other perspective.

Customers can sometimes become their own worst enemy.

On the record, software companies clearly shout success stories to the world when they have worked together with a customer to achieve something exciting. Off the record, however, even though they typically won't name names, suppliers also discuss with us lessons learned from when things haven't gone quite so well. Tales of the 'customer from hell' sometimes come up in this context.

If you have enough of these conversations, or like most members of the Freeform Dynamics analyst team have worked on both sides of the fence, the old cliché, "it takes two to tango", often springs to mind. Sometimes, for example, a relatively minor software problem can end up as a major crisis mostly as a result of miscommunication or mishandling within the customer camp. When customers don't acknowledge internal limitations, withhold key information, fail to take responsibility for their actions, and so on, they can become their own worst enemy (Figure 6).

Switching the spotlight from the software supplier to the customer, how often do software issues occur or persist as a result of the following?

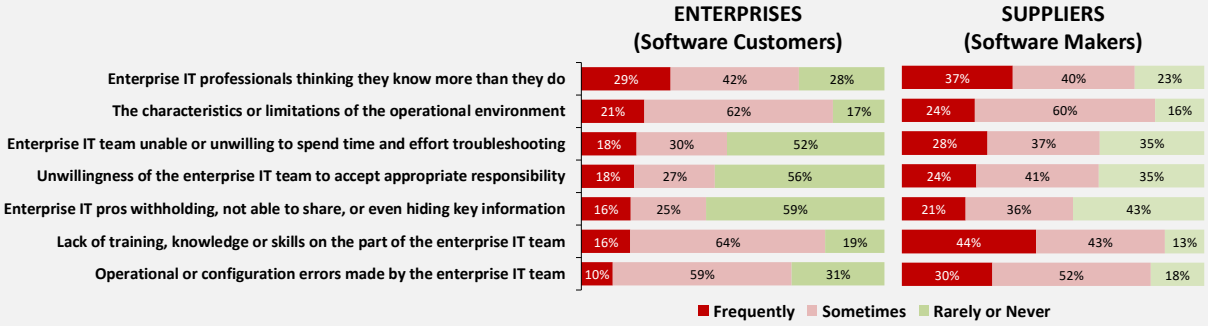


Figure 6 Customer behavior has a big impact on outcomes too

Here are some relevant comments from the supplier participants in our study:

Anecdotal feedback shines a spotlight on diagnostic processes and the importance of modern solutions to underpin them.

> "I'd say around 40% of the cases we get are problems with the human(s) involved, either not understanding the tool they have bought or the environment they are using it in, or never having looked at the manual."

> "Typical, would be cases where infrastructure teams don't talk to each other - so things like firewall ports not being opened, or having timeouts that cut-in overnight or over a weekend, are what's led to the problem."

> "A frequent early question is 'what has changed in your environment recently?' This question sometimes needs multiple repetitions, as the person may be unaware, may assume major changes are not relevant, or just be too embarrassed to admit what they have done."

Points like these shine a spotlight on the diagnostics processes and the importance of having modern solutions in place to underpin them.

Drill down on diagnostics and resolution

Troubleshooting software issues in a production environment is fraught with challenges at the best of times, especially when problems are intermittent and difficult to reproduce (Figure 7).

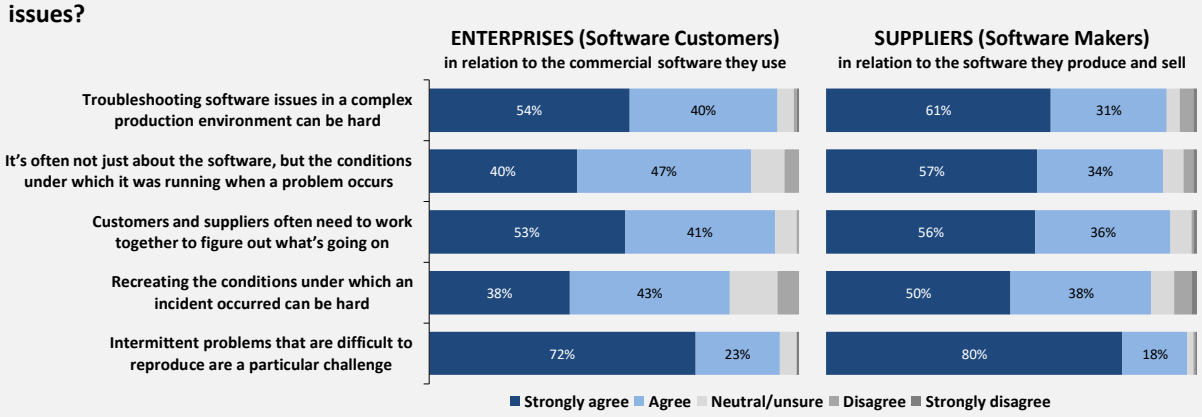


Figure 7 Troubleshooting is hard in a complex production environment

Given this, it's not surprising that incidents are often left unresolved, then come back to bite at a later stage (Figure 8).

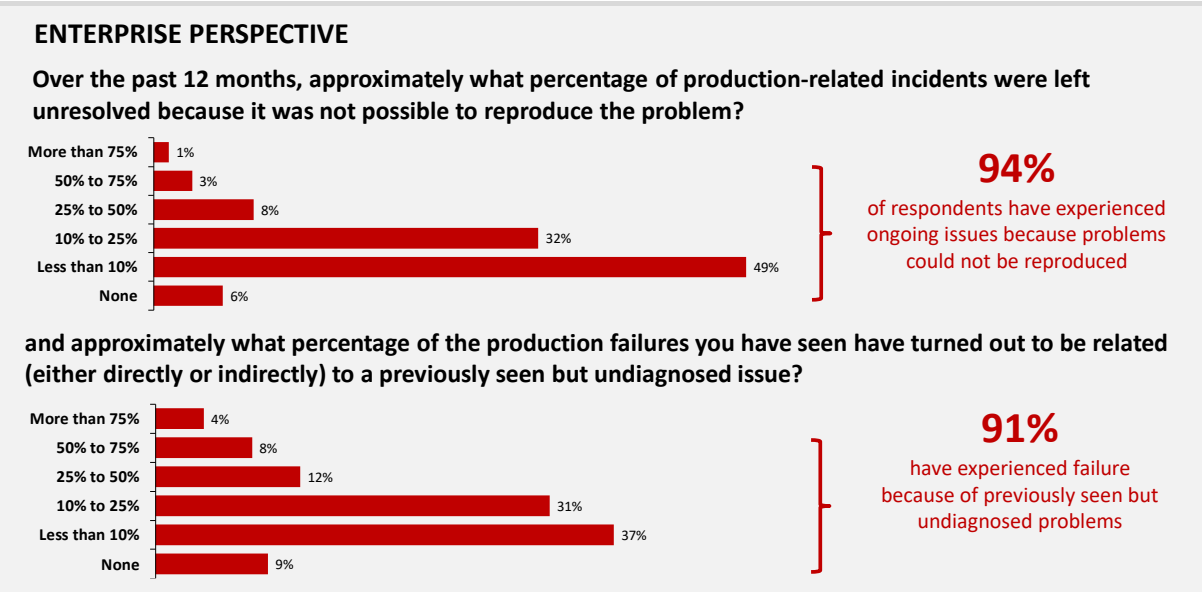
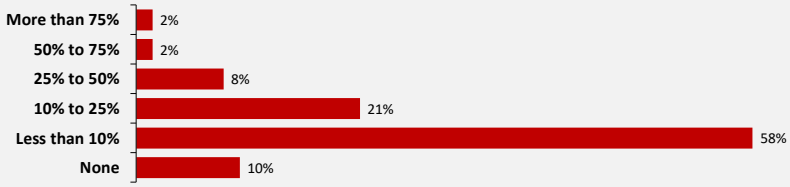


Figure 8 Issues come back to bite if left undiagnosed

But the truth is that the diagnostic challenges too often originate upstream within the software maker's own environment. Test failures are sometimes left unresolved because it was not possible to reproduce the failed test in development; again, undiagnosed failures have a nasty habit of reappearing later (Figure 9).

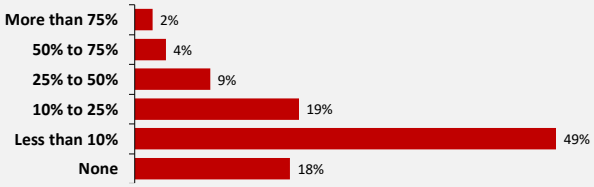
SUPPLIER PERSPECTIVE

Over the past 12 months, approximately what percentage of test failures were left unresolved because it was not possible to reproduce the failed test in development?



90%
of respondents have experienced ongoing issues because problems could not be reproduced

and approximately what percentage of the production failures you have seen have turned out to be related (either directly or indirectly) to a previously seen but undiagnosed test failure?



82%
have experienced failure because of previously seen but undiagnosed problems

Figure 9 Upstream diagnostic challenges are evident in many supplier environments

Given the implied need here, solutions based on record and replay technology have emerged to fill the gap. These complement familiar tooling options in areas such as continuous integration and continuous testing, but at the moment awareness and knowledge is relatively limited (Figure 10).

RECORD AND REPLAY SOLUTIONS

Framing and definition

We'd like to get your views on a specific type of solution that has recently become available - program execution record and replay technology. From a customer perspective, it's like installing the equivalent of an airplane 'black box' flight recorder into your production environment which can be enabled on demand to provide full visibility into what the software really did before it crashed or misbehaved. By catching failures 'in the act', it makes them 100% reproducible and speeds up the diagnostic and issue resolution process. The software engineers responsible for the application get a recording file (a standalone reproducible test case) which they can replay in the development environment - i.e. run through the recording forwards and backwards with a reversible-debugger to inspect the entire program state at any point in its execution history. In doing so, they can obtain all the data they need in one-go to diagnose and fix issues quicker than with traditional debugging approaches.

Are you familiar with 'record and replay' technology as we have described it?

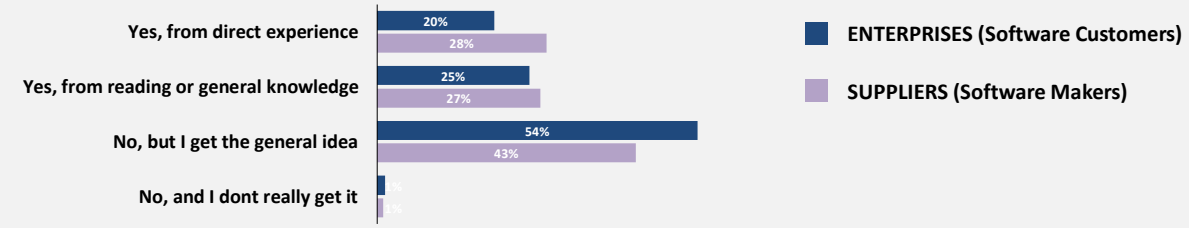


Figure 10 Awareness and knowledge of emerging diagnostic tools is limited

Despite the relative shortage of direct experience, however, most of the respondents in our study acknowledged the potential of record and replay technology. For suppliers it's about fast and effective diagnostics and remediation within development in response to test failures, especially when multiple system and data dependencies are at play. This helps to minimize the

number of defects in a given release. From the customer perspective, as well as enjoying better inherent software reliability, they also benefit from the supplier's ability to troubleshoot problems more quickly and efficiently, and with less distraction, in the complex production environment (Figure 11).

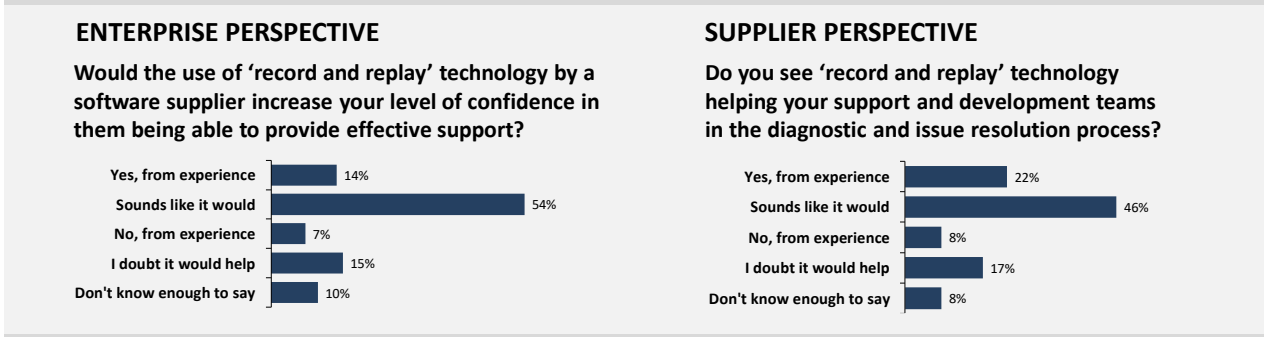


Figure 11 Most acknowledge the potential of 'record and replay' solutions

Modernization is often required equally as much within the software maker environment.

Record and replay technology is clearly just one example of the kind of solutions that can help suppliers in the continual quest to enhance software reliability and deliver a better service to customers. However, it serves to highlight that modernization more broadly is often required as much within the software maker environment as it is in the mainstream enterprise domain. This brings us onto the topic of optimizing things in general.

Pulling it all together

If we stand back and look at some of the key factors that can impact how well customers and suppliers work together on software reliability, the encouraging observation is that there is good agreement on what's important (Figure 12).

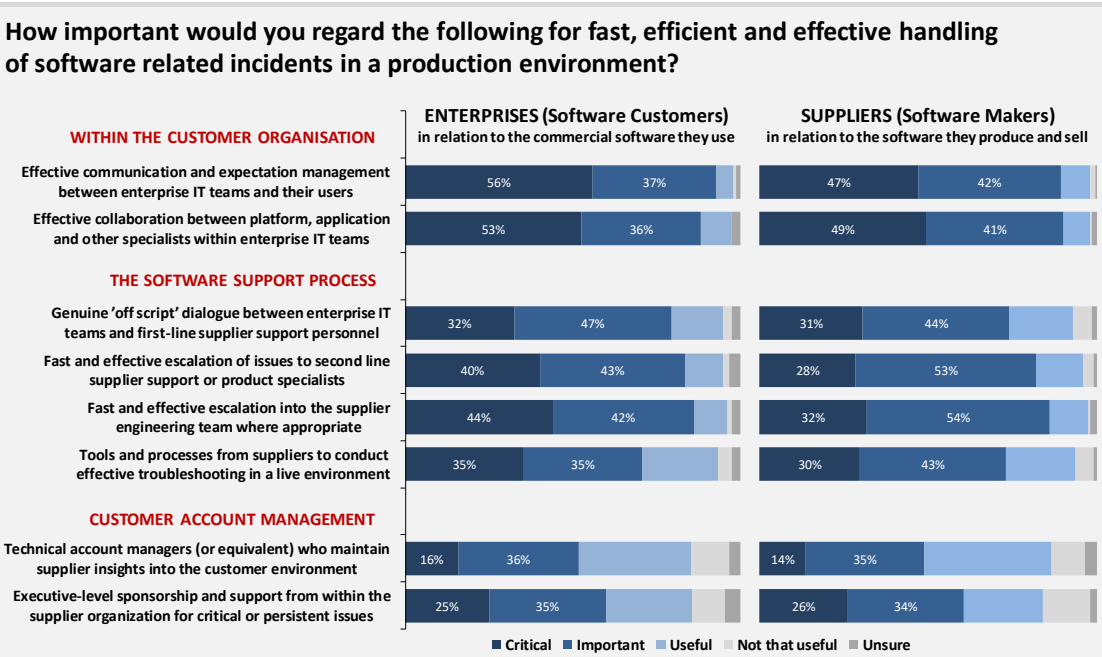


Figure 12 There's general agreement on the main ingredients for success

Many imperatives are about people and communication.

When looking at the above chart, you will note that many of the imperatives are about people, communication and relationship management. Sure, processes and tooling are important, but the key to a successful relationship is everyone talking and working together in the right kind of spirit.

The good news is that many in our research sample understand the requirements and some are doing pretty well in key areas already. However, there's still clearly work to be done by a significant number on both the enterprise and supply side of the equation (Figure 13).

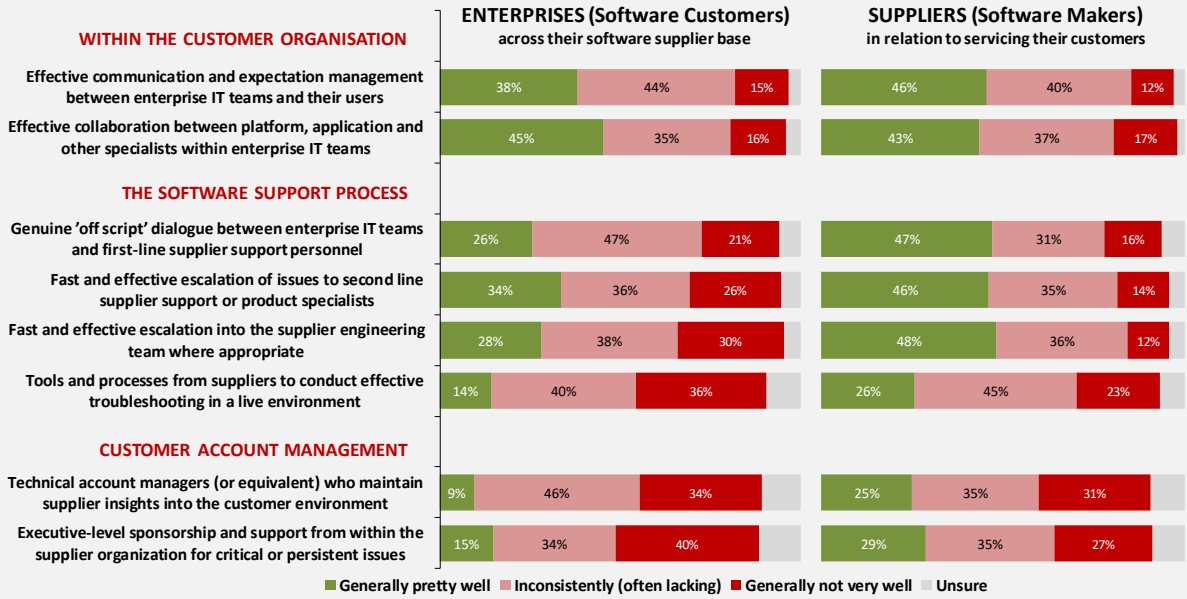


Figure 13 Gaps and shortfalls are evident in a number of key areas

More specific observations we can make from this chart include:

- Suppliers often overestimate the fitness for purpose of their support process. In the middle section of the above chart, you can see significantly less green on the left in the enterprise column, and more red on the right.
- A particular weakness among suppliers is in relation to the tools and processes used to deal with troubleshooting in a live environment. As landscapes become more complex, this issue needs to be prioritized.
- Again, relevant to dealing with increasing complexity, there is a clear shortfall in relation to technical account management. An informed, proactive and preemptive approach to support can have benefits for both parties.
- Lots of software vendor executives tell us as analysts that maintaining superior customer service is a strategic consideration, yet those same executives are too often nowhere to be seen when customers are hurting.

A particular weakness among suppliers is in relation to the tools and processes used to deal with troubleshooting in a live environment.

As every organization, product and scenario is different, you will undoubtedly be able to pick out other findings on this and other charts that are important to you.

Final thoughts

When considering the significance of this research, it's useful to recap the key findings we outlined at the beginning:

- Applications you can rely on are critical in today's digital world
- Poor software reliability translates to a tangible business risk
- When things go wrong, the big enemies are delay and uncertainty
- Despite the diagnostic need, it's hard to troubleshoot production systems
- Too many issues are left undiagnosed, then they later come back to bite
- Emerging diagnostic solutions, such as record and replay, can help
- But it takes two to tango - people and relationships are important too

There's no getting away from the conclusion that software reliability is a huge imperative in today's fast-moving digital business environment, even though delivering on it is hard.

Picking up on this last point, enterprise IT teams are increasingly expected to work as a genuine partner to the business, and this in turn impacts the relationship between software suppliers and their customers. Old transactional approaches to relationship management therefore no longer cut it, especially when more strategic software solutions are involved.

On the supplier side of the equation, effective commercial and technical account management are both key enablers here, as well as paying attention to some of the software reliability and support practicalities we have covered. And, quite frankly, with barriers to switching coming down, if suppliers don't step up and focus more on software reliability, relationship management and service delivery, then escalating levels of customer churn are inevitable.

The enterprise IT team is increasingly expected to work as a genuine partner to the business, so suppliers must assume a partnering role too.

But our study also highlights that customers have a responsibility too. An open, honest and cooperative approach will make sure you get the most from your suppliers. Conversely, even if you don't become your supplier's 'customer from hell', if you put up the barriers, fail to acknowledge the real-world practicalities of supporting software in complex production environments, and don't provide the supplier with the information and help they need, then you'll get the poor service you deserve.

With this in mind, the ultimate test of a good partnership is how the parties behave when problems arise. Do the barriers go up and the defense mechanisms kick in? Do the parties focus on what matters to them and lose sight of what's important to their partner? Or, do they start talking and work shoulder-to-shoulder to achieve a resolution?

The findings of our study suggest that both suppliers and customers can often do better here, and we hope our discussion has provided food for thought on how you might optimize the supplier-customer relationships that are important to your own business.

About Freeform Dynamics

Freeform Dynamics is an IT industry analyst firm. Through our research and insights, we aim to help busy IT and business professionals get up to speed on the latest technology developments, and make better informed investment decisions.

For more information, please visit https://www.freeformdynamics.com.

About The Register

The Register started life as a daily news operation on the web in May 1998. On the first day, 300 readers visited; in November 2013, 9.5 million unique readers visited the site every month, according to an independent audit by ABCe. The Register's blend of breaking news, strong personalities, and its accessible online execution, has made it one of the most popular authorities on the IT industry.

With an international team of journalists and columnists, The Register reports on the IT industry from the inside out – covering everything from enterprise software to chip developments.

About Undo

Undo is the leader in software reliability based on software flight recorder technology.

Undo's Live Recorder solution is used on mission-critical applications to fix bugs faster and resolve customer issues orders of magnitude faster than with traditional debugging methods. Live Recorder allows engineering teams to record and replay software execution in development, test or in production. Live Recorder records the execution of a program as it fails - capturing bugs 'in the act' and turning sporadic or intractable issues into standalone 100% reproducible test cases that offer full visibility into all the factors that led up to a program failure. Engineers can then replay that recording using a Undo's reversible-debugger to step backwards or forwards through the code, see the contents of any memory location at any point in the program's execution history and get to the root cause of the problem - fast. Customers include global players such as SAP, Micro Focus, Mentor Graphics, and Cadence Design Systems.

For more information, please visit https://undo.io/

Terms of Use