WHITEPAPER

ADTMAG
APPLICATION DEVELOPMENT TRENDS

ORGANIZATIONS INCREASINGLY LOOK TO **CONTINUOUS DEPLOYMENT** TO REDUCE FAILURES

IBM

www.ibm.com

Executives and IT managers fear failing to successfully deploy applications across development, testing and production because it can result in poor quality and slow cycle times. Loss of sleep is nothing compared to the potential loss of revenue and reputation as frustrated customers wait for slow and unreliable software delivery processes to deliver new, functional applications.

A NEW APPLICATION DEVELOPMENT TRENDS survey shows "reducing risk of failure" is one of the top of mind priorities for every management level—from CEOs to IT directors. The survey indicates that executives and managers are looking to DevOps practices for ways to simultaneously increase the quality of the release process while reducing deployment times for on-premises systems as well as a variety of cloud implementations. Successful deployments are increasingly important for respondents who are deploying apps to meet the insatiable demands of mobile device users.

JOB TITLES OF 144 SURVEY PARTICANTS INCLUDED:

- CEO, President, Owner
- CIO, CTO, Chief Software Architect, VP of App Dev
- Director of Software Development
- Application Development Manager
- Software Architect/Engineer
- IT Manager/Director

Deployment failure in the mobile/social media era can result in missed opportunities to be first to market, damage to an organization's brand and reputation, and customer dissatisfaction—all of which were identified as major problems in the survey.

Participants in the survey cited "outdated, cumbersome and slow manual processes" as major sources of delays in deploying increasingly complex business critical applications. Company leaders are concerned that manual processes require too many people performing error-prone tasks. To solve this problem, executives and managers indicate they need a solution that automates the application deployment process. This white paper explores the issues raised by this new survey—as well as available remedies for the typical causes of application deployment failure.

THE COSTS OF FAILURE

Failure causes quantifiable dollar losses. It can happen when there are configuration and application errors due to manual scripts and processes. These cause errors in production which lead

NO COMPANY CAN AFFORD **DECREASED CUSTOMER SATISFACTION** OR **DAMAGED BRAND REPUTATION**, OR **PRODUCT LAUNCH DELAYS** WITHOUT SUFFERING **LOST REVENUE**.

to lost time, lost revenue and decreased customer satisfaction. Take for example downtime. Besides being frustrating for IT professionals and end-users alike, money is being lost when an application goes down. A 2014 IDC report [1] on financial losses due to downtime and infrastructure failure found that on average, unplanned application downtime cost Fortune 1000 companies between $1.25 billion and $2.5 billion per year. The average cost of infrastructure failure was $100,000 per hour. Worse yet, mission-critical application failure costs were estimated between $500,000 and $1 million per hour.

Asked to rate the degree of impact that failed application deployments have on their business on a scale of 1 to 6, with 6 being the highest impact, responses starting with the worst impact were:

1. Decreased customer satisfaction → **4.6**
2. Damaged brand reputation → **4.02**
3. Wasted time rolling back and/or fixing the failure → **3.95**
4. Resource allocation to fix the failure → **3.94**
5. Product launch delays to market → **3.64**
6. Loss of revenue → **3.17**

In the ADT survey, more than half (53.8%) of respondents said they were able to recover from a failed deployment into production in less than a day. Another 21.7% said it took them a day or so. However, if failures can cost as much as $100,000 per hour, even relatively quick recoveries are potentially very costly. For the 8.4% of respondents who said it took several days to recover and the 12.6% who said it takes a week or more, the hourly costs of failure could have a significant impact on a company's bottom line.

In the ADT survey, almost one third (32.6%) of all respondents said "reducing risk of a failure" is their IT organization's most important application development delivery goal.

While "loss of revenue" appears to rank last among the impacts of failure, a closer look at the rankings indicates the potential for dollar losses in every one. No company can afford decreased customer satisfaction or damaged brand reputation, or product launch delays without suffering lost revenue. Internally, there are also bound to be human and technical resource costs due to wasted time rolling back and/or fixing the failure.

Whether they occur in application development, deployment or maintenance, failures are costly.

[1] DevOps and the Cost of Downtime: Fortune 1000 Best Practice Metrics Quantified, IDC, Stephen Elliot, 2014

MAINTAINING EFFICIENCY IN THE FACE OF COMPLEXITY

Organizations are moving to DevOps, an essential enterprise capability for the continuous delivery of software-driven innovation that enables organizations to seize market opportunities and reduce time to customer feedback. By automating the process of software delivery and infrastructure changes, this creates an environment where building, testing, and deploying applications meets the demand for rapid, frequent, and reliable releases.

While the promise of DevOps is greater automation of application development, testing, deployment, and monitoring, the ADT survey indicates IT professionals are dealing with issues arising from increasingly complex environments (including on-premises, cloud, mobile, etc.), which make it harder to get an application from concept to market. Complexity of applications was the number one bottleneck to faster application deployments, as seen in the answer to this survey question:

WHAT IS YOUR NUMBER ONE IMPEDIMENT TO FASTER APPLICATION DEPLOYMENTS?

- Complexity of applications → **34.4%**
- Outdated/cumbersome/slow manual processes → **25%**
- Lack of automation → **19.5%**
- Too many people involved → **18.8%**
- Other → **2.3%**

Coping with complexity isn't made any easier when a project is divided among multiple teams working at different speeds, especially when some teams are using inefficient processes.

But the problem isn't always too many people. The ADT survey indicates it can also be a problem if you don't have enough people with specific deployment skills. Some respondents indicated their problem was having too few or perhaps only one engineer capable of handling application deployment. Some organizations are facing complexity without the expertise needed to understand and deal with it. Depending on one expert in a transient work culture is risky.

Even though respondents to the survey indicated they were mostly deploying applications on-premises and beginning to move to private, public and hybrid clouds, working with these multiple and often complex environments is not easy as seen in the answers to the question on challenges when deploying applications.

Complexity of heterogeneous environments was the top challenge when deploying applications that touch both legacy and new systems. This reflects issues of multi-speed IT where Systems of Engagement (new front-end cloud and mobile systems) and Systems of Record (core business systems) have different teams, tooling, and processes. Yet the business has needs requiring integration and collaboration between the two.

Respondents also identified managing work across multiple teams (51.7%) and deploying across on-premises, cloud or hybrid (43.7%) as challenges.

WHAT ARE SOME OF THE CHALLENGES WHEN DEPLOYING APPLICATIONS THAT TOUCH BOTH LEGACY AND NEW SYSTEMS?

- Mix of application & system architectures → **70.1%**
- Managing work across multiple teams and executing at different speeds / tempo → **51.7%**
- Deploying across on-premises, cloud or hybrid environments → **43.7%**
- Server configurations & high availability → **40.2%**
- Number of servers (database, application, web) → **28.7%**

Note: Percentages reflect respondents having more than one choice on this question.

While automating deployment processes would help organizations to cope with all the complexity, a little more than a quarter (25.2%) of respondents indicated they had automated or even semi-automated processes, and almost a fifth (19.5%) cited lack of automation as a stumbling block.

WHAT IS DELAYING DEPLOYMENT?

It should not be a surprise that more than one-third of respondents (34.5%) said testing was the biggest source of delay when deploying an app into production. When asked: What is the biggest source of delay when deploying an application into production? Testing and development were cited by more than half (50.7%) of respondents as the biggest source of delay. Adding in the 12.7% citing change management (getting into production is very hard), and 8.5% blaming environment provisioning/configuration, plus 7.7% bogged down by deployment, all those parts of the software development lifecycle account for almost 80% (79.6%) of the delays in getting an application deployed.

Looking beyond the usual suspects (testing and development) at the larger deployment picture including change management (12.7%), reliance on the knowledge of a few (or one) deployment engineers (12%), environmental provisioning/

WHAT IS THE BIGGEST SOURCE OF DELAY WHEN DEPLOYING AN APPLICATION INTO PRODUCTION?

- Testing → **34.5%**
- Development → **16.2%**
- Change management (getting into production is very hard) → **12.7%**
- Reliance on the knowledge of a few (or one) deployment engineers to deploy applications into production → **12%**
- Environment provisioning/configuration → **8.5%**
- Deployment → **7.7%**
- Maintenance of manual deployment scripts → **4.2%**
- Other → **2.8%**
- Visibility and traceability within deployments → **1.4%**

configuration (8.5%), deployment in general (7.7%) and maintenance of manual deployment scripts (4.2%), the survey results indicate that almost half (45.1%) of respondents are seeing problems with deployment processes.

WHAT CAN BE DONE ABOUT QUALITY? SHIFT LEFT!

Software architects and technology analysts have been talking about incorporating testing earlier in the software development lifecycle to find bugs sooner since at least the client/server era began more than 20 years ago. The results of the ADT survey, showing that more than one-third of respondents still view testing as the culprit in delaying deployments, would suggest that this perennial problem has not been solved to everyone's satisfaction.

Shift Left is a solution advocated in a blog by Darrell R. Schrag (@devopsdrs), an experienced software lifecycle consultant focused on implementing continuous delivery of software applications. (In the text box at the bottom of this page is an excerpt explaining the term and how it might be a breakthrough in software quality management).

SHIFT LEFT

THE TERM SHIFT LEFT REFERS TO a practice in software development where teams focus on quality, work on problem prevention instead of detection, and begin testing earlier than ever before. The goal is to increase quality, shorten long test cycles and reduce the possibility of unpleasant surprises at the end of the development cycle—or still worse, in production.

Shifting left requires two key DevOps practices: continuous testing and continuous deployment. Continuous testing involves automating tests and running those tests as early and often as possible, along with service virtualization to mimic unavailable systems. Continuous deployment automates the provisioning and deployment of new builds, enabling continuous testing to happen quickly and efficiently.

The first and most obvious way to shift operations left is to work side-by-side with development in creating the deployment and testing processes. Failures observed in production are often not seen earlier in the lifecycle. Many times these failures can be directly attributed to differences in deployment procedures. Development may create its own deployment procedures that are very different from those used by operations for production. Sometimes production procedures are much more manual and may even use different tooling. Operations and development need to take ownership in building standard deployment procedures using tools like IBM UrbanCode Deploy. The deployment process is then practiced potentially hundreds of times in test environments before reaching production. You then have much more confidence that the production deployment will be successful. *—Darrell R. Schrag*

UrbanCode Deploy helped Fidelity International reduce the time required for software releases by 99 percent, from 2 - 3 days to just 1 - 2 hours. The company also achieved cost avoidance of more than USD 2.3 million per year.

"Our application release process has been fully automated with IBM UrbanCode Deploy. Applications that took days to release now take just an hour"

– Head of technology architecture and engineering, Fidelity International

Find Out More

COST AND BENEFITS OF AUTOMATED SOLUTIONS

Automated solutions can help ease many deployment headaches and reduce delays and failures.

For example, IBM UrbanCode provides:

- Automated, consistent deployments and rollbacks of applications
- Orchestration of changes across servers, tiers and components
- Configuration and security differences across environments
- Visibility into what is deployed where and who changed what
- Integration with middleware, provisioning, service virtualization, and test automation
- Full stack configuration and deployment for on-premises and public, private and hybrid cloud providers
- Documented deployment processes that can be reused

The ADT survey indicates many organizations are still working with only partially automated deployment processes, which makes them more prone to failure.

The positive news is that a substantial majority of respondents (77.7%) are enjoying success by using some form of automation for deployments.

Asked: **How would you rate your organization's proficiency with application deployments?** 20.8% responded that their deployments were effective and automated, In addition, 56.9% said their deployments were effective although not fully automated yet.

HOW ARE YOU CURRENTLY MANAGING YOUR APPLICATION DEPLOYMENTS?

- Manually using spreadsheets and documents → **11.1%**
- Writing deployment scripts → **15.6%**
- Automation tool(s) (automated or semi- automated) → **25.2%**
- A combination of one or more of the above → **48.1%**

However, when asked in the survey what was the obstacle to automation, respondents indicated that the biggest stumbling block was the cost of the tools. When asked:

What was/is your biggest obstacle to purchasing an automated application deployment solution?

Cost of the product / securing budget and approvals was the most frequently checked box with 41%.

That concern is always a legitimate one but it needs to be balanced by considering the costs of failure, which were discussed at the beginning of the paper. There are major financial consequences to having:

- Dissatisfied customers
- Damage to the company's brand
- Delays in deploying mission critical applications to support the business
- Disastrous expenses dealing with downtime

UrbanCode Deploy also helped Amica reduce software deployment times by up to 98%.

"With the UrbanCode Deploy software, we can reject a bad build and leave the environment alone until the issue has been fixed. And then it's just a click of a button to push the deployment when we're ready," said Greg Calderiso, information systems officer in Amica's corporate information systems department.

Find Out More

To help sort out the cost/benefits of an automated solution, IBM provides an advanced ROI calculator.

Using the calculator you will be able to quantify the ROI for an automated solution, which may be key to securing budget approvals. The calculator will show how:

- Automating manual processes affects your bottom line
- Using pre-built integrations vs. writing scripts helps speed deployment
- App deployment automation reduces errors in production and pre-production

MORE ABOUT URBANCODE DEPLOY

IBM UrbanCode Deploy automates manual, error-prone processes, enabling IT organizations to reduce production failures. It is the only solution robust enough to decrease risk of failed deployments to increasingly complex hybrid cloud environments. Find out more by watching this video.

APPENDIX A: IBM URBANCODE APPLICATION

Development and Deployment Survey 2016

Conducted by Application Development Trends. View here.