White Paper

Application Delivery Management

Study: Agile and DevOps Reduces Volume, Cost, and Impact of Production Defects

It has long been believed that defects found in production are more costly to fix than defects detected earlier in the process. Organizations using Agile or DevOps observe that these methodologies are helping to reduce the impact of defects escaping into production. However, defects continue to appear in production, and even low-impact defects can have a cumulative detrimental effect on a brand. The results in this study by The Harris Poll appear to back up this belief.

MICRO FOCUS

Table of Contents

A recent survey conducted for Micro Focus by The Harris Poll explored how organizations track the cost of defects, the breakdown of the cost of a defect, and whether incremental software development methodologies such as Agile and DevOps help to reduce the cost of a defect when compared to Waterfall software development.

Introduction

Organizations using Agile or DevOps to develop and deliver software observe that these methodologies are helping to reduce the impact of defects escaping into production. However, defects continue to appear in production, and even low-impact defects can have a cumulative detrimental effect on a brand.

A recent survey conducted for Micro Focus® by The Harris Poll explored how organizations track the cost of defects, the breakdown of the cost of a defect, and whether incremental software development methodologies such as Agile and DevOps help to reduce the cost of a defect when compared to Waterfall software development. For the purposes of the survey, the term 'defect' was defined to be 'incorrect, unexpected, or unintended behavior identified in the software'.

The main conclusions of the research include:

- Agile or DevOps methods help reduce the impact of defects getting into production workloads.
- A single low-impact defect may not have a great impact on the brand, but there is a greater corrosive effect from multiple such defects. A cumulative effect, if you will. This suggests that the point-of-view that it is only worth fixing defects that result in outages or data loss should be re-evaluated.
- Tracking the cost of defects is widely reported, with the top goal to uncover systemic problems leading to defects.
- Security risks are contributing the most to the cost of defects, but all factors tested are important contributors.
- There is no clear salient challenge in managing or mitigating defects, and challenges are similar regardless of development methodology.
- When we look at the nature and distribution of defects, we see more similarity overall than dissimilarity between followers of Waterfall vs. Agile/DevOps methods.
  - There is a slight reduction in the volume of high-impact defects under the Agile/DevOps method—potentially a result of uncovering problems earlier in the process.
  - And, while there is agreement that Agile/DevOps reduces the impact of production defects, when they rate the effect of defects, these effects are weighted to the production stage under both Waterfall and Agile/DevOps methods—Agile may have less chance of the defect escaping to production, but production is where there is the greatest cost.
  - There is an increased emphasis on data recovery under the Agile/DevOps approach.

The survey was conducted online within the United States between August 23–September 5, 2019 among 204 US adults 18+ working in organizations with 5,000 employees or more operating across a variety of sectors.

Overall, the findings of the survey support the oft-repeated claim that Agile and DevOps reduces the number of high-impact defects escaping into production when compared with Waterfall. The rest of this paper looks at the findings of the survey in detail and offers recommendations that reduce both the impact and abundance of production defects.

As expected, the vast majority of respondents (84%) agree that that low-impact defects typically have a low-effect on brand reputation.

Defect Assessment and Tracking

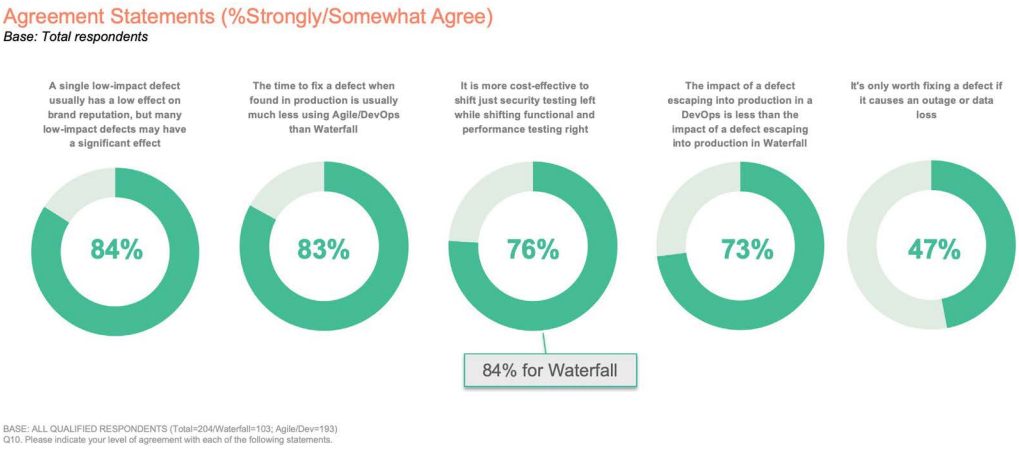


Figure 1. Agreement with statements on defects

For the purposes of the survey, the impact of production defects was defined as follows:

- **Low-impact:** Little or no impact on users or revenue.
- **Medium-impact:** Impacts some users or business activities from working effectively or efficiently.
- **High-impact:** Has widespread impact on users or impacts key business activities or revenue is being lost.

As expected, the vast majority of respondents (84%) agree that that low-impact defects typically have a low-effect on brand reputation *(see Figure 1)*. However, many low-impact defects can combine to have a significant effect. At the same time, 47% of respondents agree that it's only worth fixing a defect if it causes an outage, or data loss.

Recommendation: Fix known low-impact defects before you release the software. The fewer low-impact defects you release into production, the less likely you are to damage your brand if further low-impact defects are subsequently discovered in production.

Almost three-quarters (73%) agree that DevOps defects in production have less of an impact than defects escaping into production in Waterfall, and 83% agree that it takes less time to fix a defect in production in an Agile or DevOps environment than Waterfall.

Almost three-quarters (73%) agree that Agile/DevOps defects in production have less of an impact than defects escaping into production in Waterfall, and 83% agree that it takes less time to fix a defect in production in an Agile or DevOps environment than Waterfall. Taken together, these results indicate that Agile or DevOps is preferable than Waterfall, when appropriate to the situation.

Recommendation: If you've not yet done so, move to Agile or DevOps. Not only will defects found in production have a lower impact than Waterfall, you'll be able to fix them quicker and reduce the effect on your customers and your brand.

We also asked whether shifting testing left, i.e., performing more testing earlier in the lifecycle, has a noticeable impact on cost. Given limited resources, teams cannot test everything in the short cycles available and are often required to optimize their testing focus. Some teams choose to consciously release software into production before it is fully tested, and essentially test it in production. The survey sought to determine whether there is a distinction between security testing and other types of testing.

Due to the fear typically associated with a potential security breach, we asked if it would be more cost effective to delay some functional and performance testing typically performed during development to production instead, while moving security testing left, towards development. 76% of Agile/DevOps respondents agreed, while 84% of Waterfall respondents agreed. This discrepancy can be explained by recalling the previous finding that the impact of Agile/DevOps defects, and the time to fix them, are both less than in Waterfall. Thus, in a Waterfall environment where the turnaround for a defect to be repaired is considerably longer than in Agile/DevOps, it is better to test security earlier if resources or time are limited. In conclusion, all testing should be shifted left, and the software continuously tested and monitored in production as well.

Recommendation: Ideally, shift all testing left. But if resources are limited, shift security testing left first.

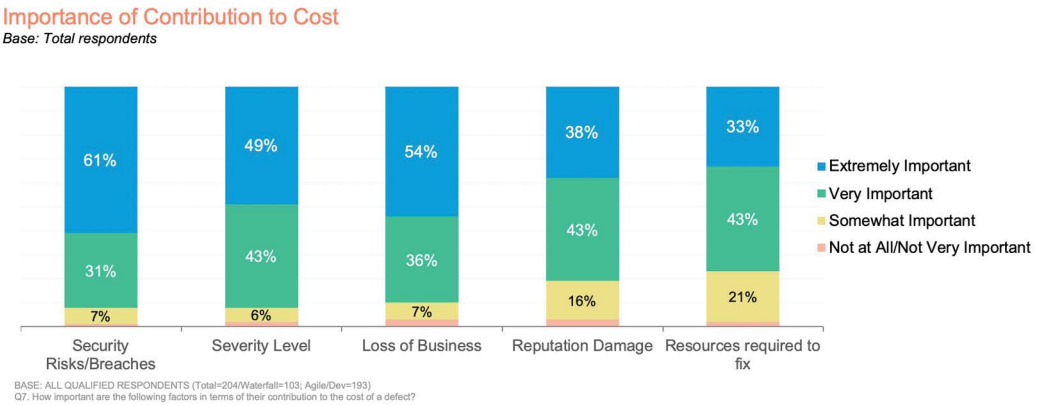


Figure 2. Contribution to the cost of a defect

Having mentioned costs, we asked the respondents what contributes to the cost of a defect *(see Figure 2 on the previous page)*. The respondents rated, and considered, the following factors:

- Security risks/breaches
- Severity level
- Loss of business
- Damage to reputation
- Resources required to fix

Each of the factors contribute to the cost, but security has the highest effect, with 61% considering it extremely important.

We asked the respondents if they assess the cost of fixing it first. The survey found that in a Waterfall environment, 52% always assess the cost, 25% assess the costs if the defect is severe enough, and 17% if the size and scope of the defect is large enough.

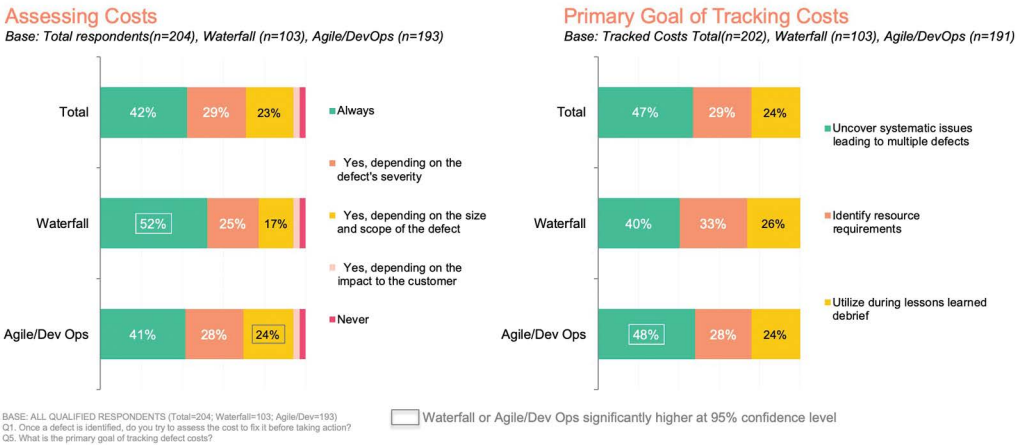


Figure 3. Assessing and tracking defect cost

Once a defect has been found, a decision must be made whether to fix it. We asked the respondents if they assess the cost of fixing it first *(see Figure 3)*. The survey found that in a Waterfall environment, 52% always assess the cost, 25% assess the costs if the defect is severe enough, and 17% if the size and scope of the defect is large enough. In Agile/DevOps, the numbers are 41%, 28%, and 24%, respectively. While the actual percentages differ, both development methodologies consider severity to be a more common motivation for assessing costs than size and scope. Very few respondents (3%, in both methodologies) consider the impact to the customer to be a reason to assess costs.

The reason for tracking costs was the same for both methodologies, and reveals that over 70% of respondents desire to improve processes, not just decide tactical resources: almost half are trying to uncover systematic issues leading to multiple defects, and a quarter look at the cost factor during 'lessons learned' debriefs. Around a third want to identify the resources required to fix the defect.

There is almost no difference between the two development methods, with 38% performing systematic defect tracking, 31% tracking as part of ongoing development costs, and 17% performing limited tracking of most defects.

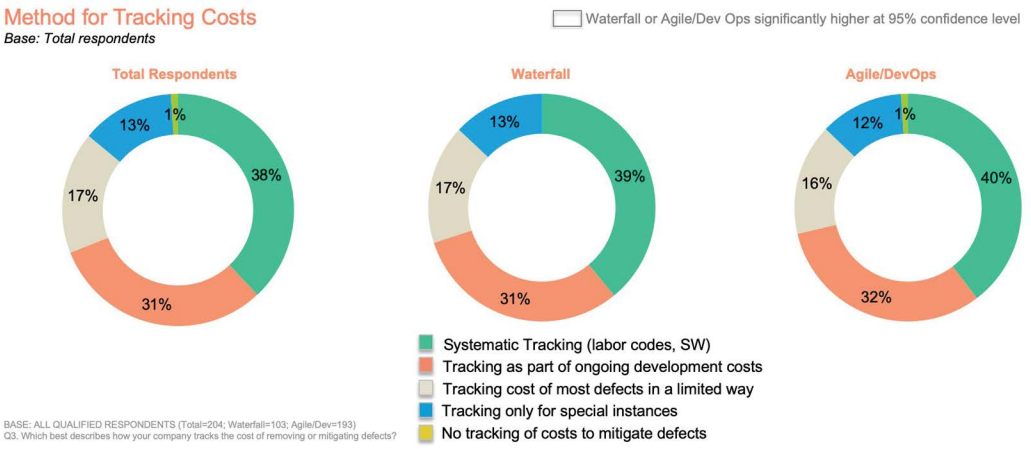


Figure 4. How costs are tracked

Regardless of the development method, most respondents keep track of a defect's cost once it is discovered. There is almost no difference between the two development methods, with 38% performing systematic defect tracking, 31% tracking as part of ongoing development costs, and 17% performing limited tracking of most defects *(see Figure 4)*. Only 1% of respondents do not track defect costs at all, and 13% track only in special instances.

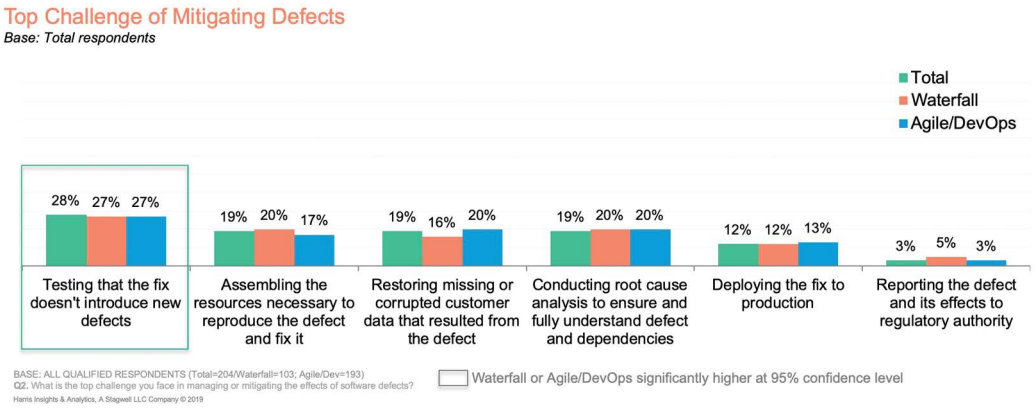


Figure 5. Challenges of managing or mitigating defect effects

Once the decision has been made to fix a defect, there are several challenges that must be overcome in order to fix it. The main challenge *(see Figure 5)* is whether Waterfall or Agile/DevOps is ensuring that the fix does not introduce new defects. The next challenges are approximately equal in importance between: assembling the resources needed to reproduce and fix the defect, restoring customer data affected by the defect, and conducting root-cause analysis to fully understand the defect and its dependencies. Only 12% report they have challenges with deploying the fix to production.

Distribution and Impact of Defects

The respondents were asked to estimate the distribution of production defects found in their product across low-, medium- and high-impact. The results indicate that in Waterfall, production defects are evenly distributed.

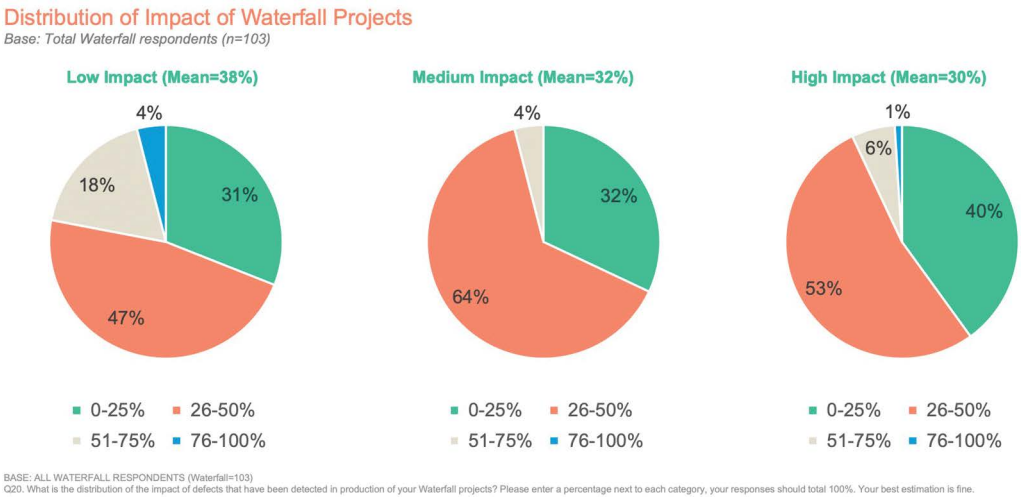


Figure 6. Distribution of production defects (Waterfall)

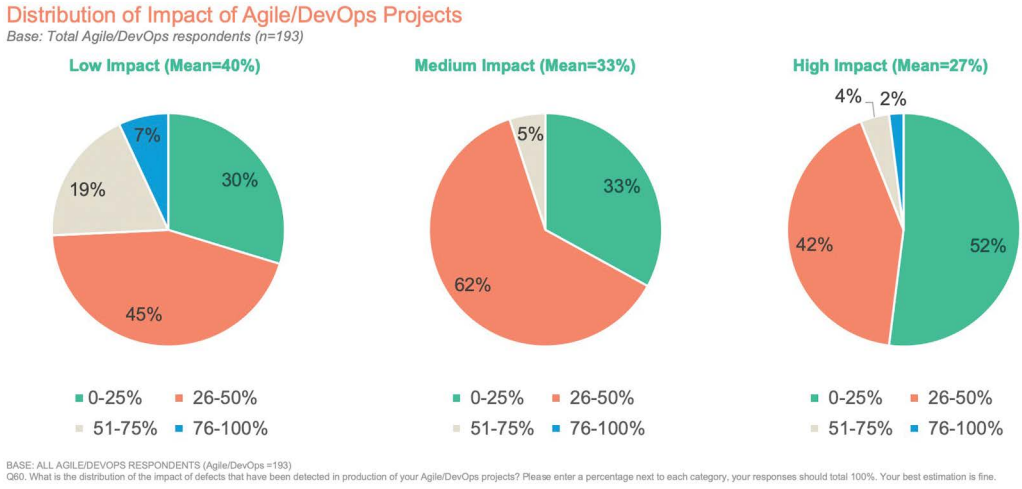


Figure 7. Distribution of production defects (Agile/DevOps)

It is an unfortunate reality that defects escape into production, whether in a Waterfall environment or in an Agile/DevOps environment. The respondents were asked to estimate the distribution of production defects found in their product across low-, medium- and high-impact. The results indicate that in Waterfall, production defects are evenly distributed (see Figure 6). But in Agile/DevOps *(see Figure 7)*, there tends to be fewer high-impact defects found in production, but generally more low-impact defects. The distribution of medium-impact defects is relatively unchanged.

The respondents were asked to estimate what elements influence these business impacts negatively the most for three primary defect categories. These are namely: Functional, Performance, and Security, for both Waterfall and Agile/DevOps projects.

An obvious question is whether Agile/DevOps reduces the number of defects found in production, in comparison with Waterfall. However, the abundance of defects that a product has depends greatly on the nature, size and scope of the product, and would be difficult to compare without developing the same product multiple times using a different methodology, which does not typically happen in business.

Effect on Different Types of Defect

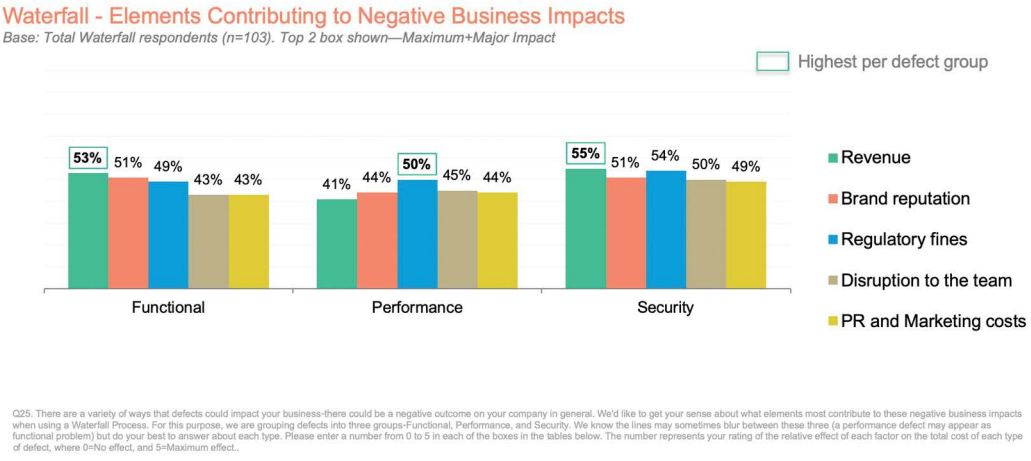


Figure 8. Elements contributing to impact on business (Waterfall)

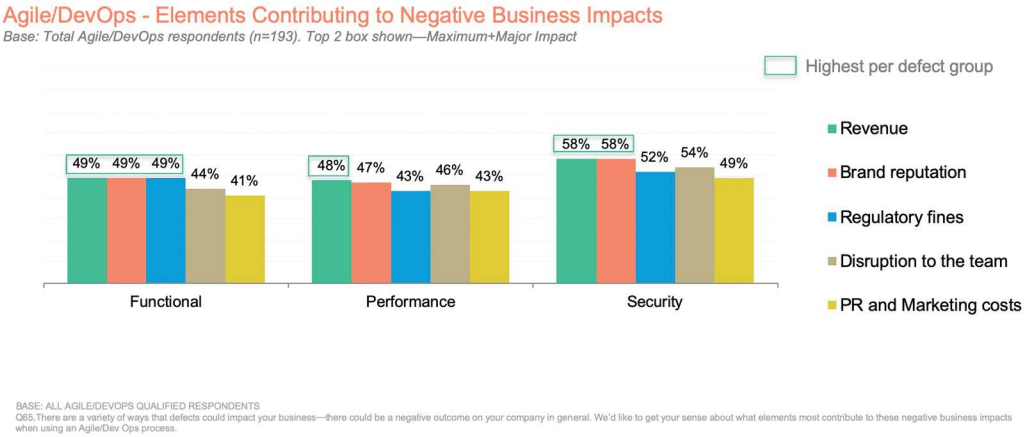


Figure 9. Elements contributing to impact on business (Agile/DevOps)

There are a variety of ways that defects can impact a business and can contribute to negative outcomes on the organization. The respondents were asked to estimate what elements influence these business impacts negatively the most for three primary defect categories. These are namely: Functional, Performance,

and Security, for both Waterfall and Agile/DevOps projects. For this research, the elements identified as having a significant effect on the business were:

- Revenue
- Brand reputation
- Regulatory fines
- Disruption to the team
- PR and Marketing costs

In Waterfall *(see Figure 8 on the previous page)*, the respondents indicate that functional defects primarily impact revenue, while performance defects lead to regulatory fines more than anything else. Security defects impact both revenue and regulatory fines with almost equal weight.

In Agile/DevOps *(see Figure 9 on the previous page)*, the salient factors for functional defects are revenue, brand reputation and regulatory fines (all with approximately equal weight). For both performance and security defects, fines tend to have a lower impact, with the greatest impact being revenue and brand reputation.

Overall, security defects are perceived to have the highest impact on negative business outcomes, whether Waterfall or Agile/DevOps.

Overall, security defects are perceived to have the highest impact on negative business outcomes, whether Waterfall or Agile/DevOps.

Recommendation: Ensure that development processes are designed to uncover security issues at the earliest development stage possible to minimize the number of security risks reaching later stages of development.

Impact of Defects Discovered in Different Stages of the Lifecycle

It is widely believed that defects found earlier in the development process cost less to fix. Given that each defect is unique and has a variety of factors influencing its actual cost, it is not possible to give a specific cost value. However, it is possible to identify the relative impact aspects of the defect lifecycle have depending on when it is detected within development. The respondents were asked to consider the same three types of defect as before—functional, performance, and security. They were subsequently asked to estimate the relative effect of the following factors:

- Defect reporting
- Defect reproduction
- Root-cause analysis
- Fixing the defect
- Testing the defect

While iterative methods have much shorter cycles (of small increments of functionality), these stages are similar enough for meaningful comparisons to be made.

- Deploying a fix
- Data recovery

Due to the nature of the different development methodologies, there is not a one-to-one correspondence for each development stage. For Waterfall, we identified the following phases:

- Plan
- Design
- Development
- Testing
- Release
- Production

For Agile and DevOps, the phases were:

- Backlog refinement
- Work-item design
- Work-item development
- Continuous Integration (CI) build
- CI Test
- Deployment
- Production

However, while iterative methods have much shorter cycles (of small increments of functionality), these stages are similar enough for meaningful comparisons to be made.

Functional Defects

Highest percentage per row

Bold - Highest percentage per Stage

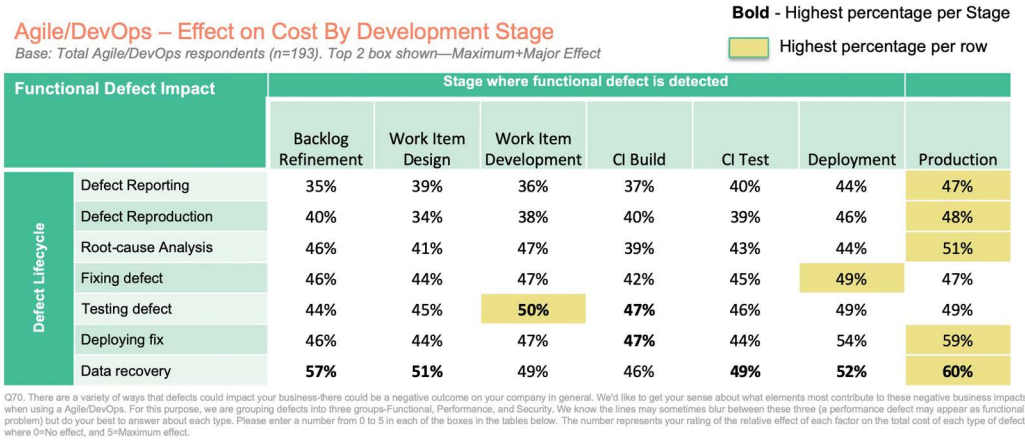
Waterfall – Effect on Cost By Development Stage

Base: Total Waterfall respondents (n=103). Top 2 box shown—Maximum+Major Effect

Functional Defect Impact		Stage where functional defect is detected					
		Plan	Design	Development	Testing	Release	Production
Defect Lifecycle	Defect Reporting	38%	37%	43%	46%	52%	51%
	Defect Reproduction	41%	40%	43%	50%	46%	52%
	Root-cause Analysis	46%	48%	46%	45%	50%	**62%**
	Fixing defect	46%	45%	45%	50%	50%	55%
	Testing defect	49%	49%	**53%**	**57%**	47%	58%
	Deploying fix	**53%**	45%	43%	53%	**55%**	61%
	Data recovery	50%	**51%**	44%	54%	46%	51%

Q30. We would like to understand your perception of the impact of a functional defect based on where the impact occurs in the Waterfall process. Assume that the defect is a relatively high impact defect, one that has widespread impact on users, business activities or revenue. Please enter a number from 0 to 5 in each of the boxes in the tables below. The number represents your rating of the relative effect of each factor on the total cost of each type of high-impact defect, where 0=No effect, and 5=Maximum effect.

Figure 10. Impact of functional defects (Waterfall)



Agile/DevOps – Effect on Cost By Development Stage

Base: Total Agile/DevOps respondents (n=193). Top 2 box shown—Maximum+Major Effect

Bold - Highest percentage per Stage

Highest percentage per row

Functional Defect Impact		Stage where functional defect is detected						
		Backlog Refinement	Work Item Design	Work Item Development	CI Build	CI Test	Deployment	Production
Defect Lifecycle	Defect Reporting	35%	39%	36%	37%	40%	44%	47%
	Defect Reproduction	40%	34%	38%	40%	39%	46%	48%
	Root-cause Analysis	46%	41%	47%	39%	43%	44%	51%
	Fixing defect	46%	44%	47%	42%	45%	49%	47%
	Testing defect	44%	45%	**50%**	**47%**	46%	49%	49%
	Deploying fix	46%	44%	47%	**47%**	44%	54%	59%
	Data recovery	**57%**	**51%**	49%	46%	**49%**	**52%**	**60%**

Q70. There are a variety of ways that defects could impact your business-there could be a negative outcome on your company in general. We'd like to get your sense about what elements most contribute to these negative business impacts when using a Agile/DevOps. For this purpose, we are grouping defects into three groups-Functional, Performance, and Security. We know the lines may sometimes blur between these three (a performance defect may appear as functional problem) but do your best to answer about each type. Please enter a number from 0 to 5 in each of the boxes in the tables below. The number represents your rating of the relative effect of each factor on the total cost of each type of defect, where 0=No effect, and 5=Maximum effect.

Figure 11. Impact of functional defects (Agile/DevOps)

Unsurprisingly, the impact of a functional defect is generally highest in the production phase of a Waterfall project, with root-cause analysis and deploying the fix contributing the greatest.

Unsurprisingly, the impact of a functional defect is generally highest in the production phase of a Waterfall project, with root-cause analysis and deploying the fix contributing the greatest *(see Figure 10 on the previous page)*.

When a defect is detected during the planning stage, deploying the fix appears to have the highest impact. This repeats for all types (functional, performance, security) of Waterfall defect. Given that in a Waterfall project, the planning stage typically does not involve code, it is difficult to understand exactly what the respondents considered to be deployed, and why it has such a high-impact. However, a defect discovered during planning could potentially affect the entire design, with 'deploying the fix' meaning conducting a new requirements-gathering exercise, which is costly.

For Agile/DevOps projects, the impact is also highest in production, but the overall impact is less *(see Figure 11)*. The highest impact found in production is due to data recovery. In Waterfall though, data recovery in production is perceived to have a lower effect than during the testing phase.

During the Agile backlog refinement stage, data recovery has a high-impact on the cost of the defect. At first glance this might seem unusual, since as with Waterfall, the backlog refinement stage does not involve code or data. However, in Agile, backlog refinement happens all the time, and a defect in a backlog item, such as an incorrect assumption in a user story, might necessitate an update to the data in order to accommodate the backlog item.

Regardless of which stage defects are discovered, cost impacts are lower earlier in the lifecycle of a functional defect, with lower impact percentages appearing towards the top of the table and growing overall as the defect is fixed and deployed, and data is recovered. This pattern is repeated for both performance and security defects and is almost identical for Waterfall and Agile/DevOps projects.

Performance defects follow a similar pattern to functional defects.

Performance Defects

Waterfall – Effect on Cost By Development Stage

Base: Total Waterfall respondents (n=103). Top 2 box shown—Maximum+Major Effect

Highest percentage per row

Bold - Highest percentage per Stage

Performance Defect Impact		Stage where performance defect is detected					
		Plan	Design	Development	Testing	Release	Production
Defect Lifecycle	Defect Reporting	37%	35%	44%	47%	53%	52%
	Defect Reproduction	39%	40%	43%	49%	47%	52%
	Root-cause Analysis	48%	50%	46%	43%	50%	**62%**
	Fixing defect	48%	46%	48%	46%	50%	55%
	Testing defect	48%	47%	**53%**	**56%**	49%	56%
	Deploying fix	**53%**	45%	43%	52%	**55%**	61%
	Data recovery	52%	**52%**	45%	53%	45%	51%

Q35. We would like to understand your perception of the impact of a performance defect based on where the impact occurs in the Waterfall process. Assume that the defect is a relatively high impact defect, one that has widespread impact on users, business activities or revenue. Please enter a number from 0 to 5 in each of the boxes in the tables below. The number represents your rating of the relative effect of each factor on the total cost of each type of high-impact defect, where 0=No effect, and 5=Maximum effect.

Figure 12. Impact of performance defects (Waterfall)

Agile/DevOps – Effect on Cost By Development Stage

Base: Total Agile/DevOps respondents (n=193). Top 2 box shown—Maximum+Major Effect

Bold - Highest percentage per Stage

Highest percentage per row

Performance Defect Impact		Stage where performance defect is detected						
		Backlog Refinement	Work Item Design	Work Item Development	CI Build	CI Test	Deployment	Production
Defect Lifecycle	Defect Reporting	33%	39%	37%	36%	41%	44%	46%
	Defect Reproduction	41%	36%	40%	39%	40%	45%	48%
	Root-cause Analysis	46%	42%	48%	40%	44%	45%	51%
	Fixing defect	49%	44%	48%	44%	46%	48%	50%
	Testing defect	42%	46%	49%	47%	46%	50%	51%
	Deploying fix	44%	47%	47%	49%	45%	**53%**	**60%**
	Data recovery	**59%**	**54%**	**51%**	**50%**	**50%**	**53%**	**60%**

Q75. We would like to understand your perception of the impact of a performance defect based on where the impact occurs in the Waterfall process. Assume that the defect is a relatively high impact defect, one that has widespread impact on users, business activities or revenue. Please enter a number from 0 to 5 in each of the boxes in the tables below. The number represents your rating of the relative effect of each factor on the total cost of each type of high-impact defect, where 0=No effect, and 5=Maximum effect.

Figure 13. Impact of performance defects (Agile/DevOps)

Performance defects follow a similar pattern to functional defects. In Waterfall *(see Figure 12)*, the highest impact is also during production, again with root-cause analysis providing the highest impact. Testing a fix for a performance defect has equal impact on cost whether the defect is detected during testing or in production.

In Agile/DevOps *(see Figure 13)*, the data is very clear, and backs up the accepted wisdom that finding a defect in production has the highest impact across every activity in the defect lifecycle, and data recovery incurs the highest cost, regardless of when the defect is discovered.

Security Defects

Waterfall project security defects discovered during testing are slightly more costly to reproduce than in production.

Highest percentage per row

Bold - Highest percentage per Stage

Waterfall – Effect on Cost By Development Stage

Base: Total Waterfall respondents (n=103). Top 2 box shown—Maximum+Major Effect

Security Defect Impact		Stage where security defect is detected					
		Plan	Design	Development	Testing	Release	Production
Defect Lifecycle	Defect Reporting	39%	37%	43%	46%	52%	49%
	Defect Reproduction	43%	41%	47%	51%	45%	50%
	Root-cause Analysis	43%	48%	44%	45%	49%	**62%**
	Fixing defect	46%	41%	43%	49%	47%	54%
	Testing defect	49%	44%	**52%**	**54%**	48%	53%
	Deploying fix	**51%**	45%	44%	52%	**56%**	58%
	Data recovery	48%	**50%**	48%	53%	48%	54%

Q40. We would like to understand your perception of the impact of a security defect based on where the impact occurs in the Waterfall process. Assume that the defect is a relatively high impact defect, one that has widespread impact on users, business activities or revenue. Please enter a number from 0 to 5 in each of the boxes in the tables below. The number represents your rating of the relative effect of each factor on the total cost of each type of high-impact defect, where 0=No effect, and 5=Maximum effect.

Figure 14. Impact of security defects (Waterfall)

Bold - Highest percentage per Stage

Highest percentage per row

Agile/DevOps – Effect on Cost By Development Stage

Base: Total Agile/DevOps respondents (n=193). Top 2 box shown—Maximum+Major Effect

Security Defect Impact		Stage where security defect is detected						
		Backlog Refinement	Work Item Design	Work Item Development	CI Build	CI Test	Deployment	Production
Defect Lifecycle	Defect Reporting	37%	41%	38%	39%	42%	45%	46%
	Defect Reproduction	41%	34%	40%	40%	41%	45%	50%
	Root-cause Analysis	46%	41%	49%	39%	41%	45%	51%
	Fixing defect	49%	43%	48%	41%	47%	49%	49%
	Testing defect	42%	45%	49%	47%	47%	50%	49%
	Deploying fix	47%	46%	47%	46%	43%	54%	59%
	Data recovery	**61%**	**51%**	**51%**	**48%**	**53%**	**53%**	**61%**

Q80. We would like to understand your perception of the impact of a security defect based on where the impact occurs in the Agile/DevOps process. Assume that the defect is a relatively high impact defect, one that has widespread impact on users, business activities or revenue. Please enter a number from 0 to 5 in each of the boxes in the tables below. The number represents your rating of the relative effect of each factor on the total cost of each type of high-impact defect, where 0=No effect, and 5=Maximum effect.

Figure 15. Impact of security defects (Agile/DevOps)

Waterfall project security defects *(see Figure 14)* discovered during testing are slightly more costly to reproduce than in production. This is most probably due to the differences in ecosystem, as vulnerabilities are not just a factor of the software's code but depend on the entire deployment environment.

In Agile/DevOps *(see Figure 15)*, security defects follow a similar pattern to performance defects, although testing the defects appears to incur a higher cost when the defect is found during deployment. However, the numbers are so close that this is unlikely to have any significance.

This paper looked at both Waterfall and Agile/DevOps and found that the latter results in lower impacting defects detected in production. We also observed that while a single low-impact defect does not have much effect on the brand, multiple low-impact defects can combine to have a more significant, cumulative, effect.

Conclusions

It has long been believed that defects found in production are more costly to fix than defects detected earlier in the process. The results in this study appear to back up this belief through two primary observations:

1. Defects (regardless of type) incur a greater cost the closer they are found to production and
2. The activities involved in a defect's lifecycle become more costly as the defect moves from detection, through fixing, and out into production.

In general, each of the tables have lower percentage values towards the top-left of the table, and the numbers increase towards the bottom-right.

This paper looked at both Waterfall and Agile/DevOps and found that the latter results in lower impacting defects detected in production. We also observed that while a single low-impact defect does not have much effect on the brand, multiple low-impact defects can combine to have a more significant, cumulative, effect. Many respondents reported that they track the cost of defects, with their main goal to uncover systemic problems that lead to further or future defects.

While the advantages of an Agile/DevOps approach over Waterfall are widely understood, we found that the nature and distribution of defects follows a similar pattern, regardless of the methodology. However, we noticed that there is a slight reduction in the abundance of high-impact defects in Agile/DevOps. While Agile/DevOps might reduce the chance of a defect escaping to production, it is in production where the cost is greatest.

Summary of Recommendations

Based on the findings in this study, we would recommend that organizations:

- **Fix known low-impact defects before releasing software.** The fewer low-impact defects you release into production, the less likely you are to damage your brand if further low-impact defects are subsequently discovered in production.
- **Adopt Agile or DevOps at the earliest opportunity.** Not only will defects found in production have a lower impact than using Waterfall, you will be able to fix them quicker and reduce the effect on your customers and your brand.
- **Shift all testing left. But if resources are limited, shift security testing left first.** Where possible, all testing should be shifted left, and the software continuously tested, and monitored, in production as well. But when that is not possible, prioritize security testing earlier in development.
- **Prioritize security testing overall.** Ensure that development processes are designed to uncover security issues at the earliest development stage possible to minimize the number of security risks reaching later stages of development.

About the Study

This survey was conducted online within the United States by The Harris Poll in September 2019 on behalf of Micro Focus among 204 US adults 18+ who met the following criteria *(see Figures 16 and 17 on the following page)*:

- Employed full-time (97%) or part time (3%)
- Work for companies with 5,000 employees or more
- Have direct involvement with Software Development Process and release
- Able to speak to impact of defects on development
- Familiar and involved with project using Waterfall, Agile, Scrum, Kanban or similar, DevOps, or both
- Have title of Research and Development Manager, Technologist, Application Development or Product Manager (Note: Titles of CIO, CTO, CISO or similar were excluded)
- Work in Financial, Manufacturing/Industrial or Consumer Product industries

Data were not weighted and are only representative of those who completed the survey.

All sample surveys and polls, whether or not they use probability sampling, are subject to multiple sources of error which are most often not possible to quantify or estimate, including sampling error, coverage error, error associated with nonresponse, error associated with question wording and response options, and post-survey weighting and adjustments. Therefore, The Harris Poll avoids the words "margin of error" as they are misleading. All that can be calculated are different possible sampling errors with different probabilities for pure, unweighted, random samples with 100% response rates. These are only theoretical because no published polls come close to this ideal.

Respondents for this survey were selected among panel members who have agreed to participate in surveys. Because the sample is based on those who agreed to be invited to participate, no estimates of theoretical sampling error can be calculated.

Respondents for this survey were selected among panel members who have agreed to participate in surveys. Because the sample is based on those who agreed to be invited to participate, no estimates of theoretical sampling error can be calculated.

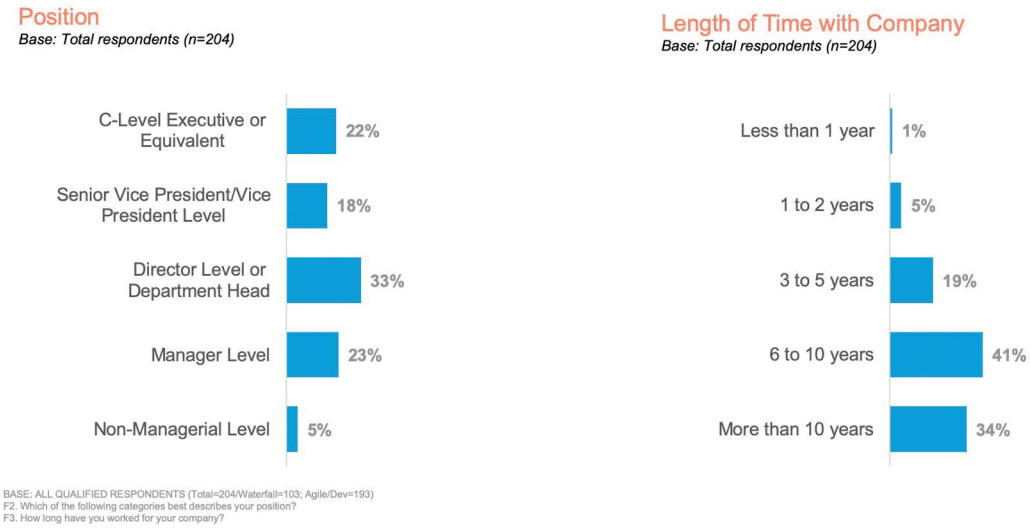


Figure 16. Respondents' positions in organization

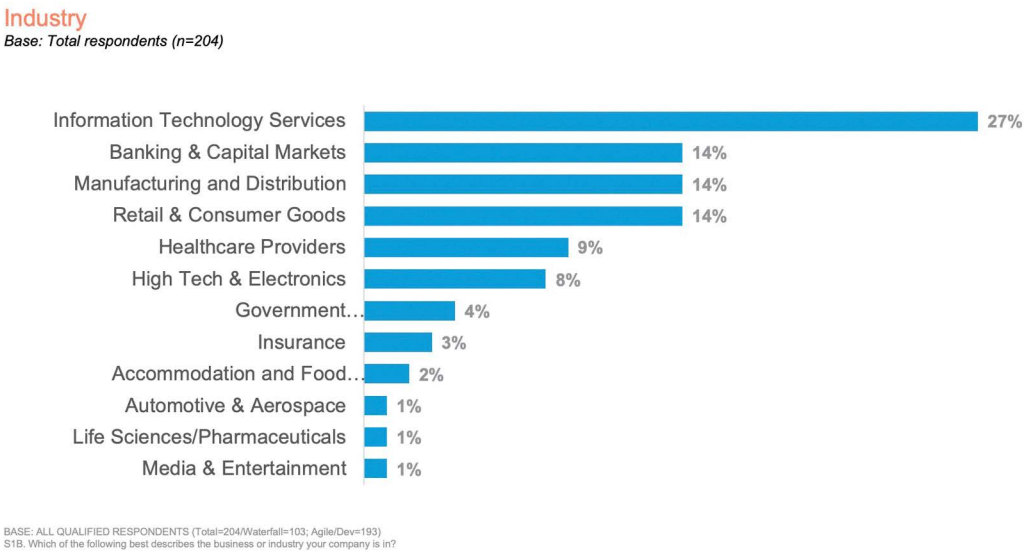


Figure 17. Respondents' industries

Contact us at:
www.microfocus.com

Like what you read? Share it.