



Hewlett Packard
Enterprise

DevOps Mindset More Important Than Formality

Research White Paper

DevOps mindset more important than formality

Executive summary

Crisply defining DevOps has proven elusive, even for supposed experts. Some of this relates to disagreement around substantive considerations, and some of it is just semantics. But the bottom line is that it's challenging to get an accurate answer to the question "are you doing DevOps"? Here we report the results of a statistical cluster analysis that reveals two segments who have embraced DevOps practices in different ways, one more formally, and one less formally - to the point that they don't even know they're doing DevOps. The informal segment is the most successful, suggesting those organizations who approach DevOps as an ongoing journey, a series of pilots and experiments, are the ones we should be following. The mindset is more important than the formality.

ABOUT THIS RESEARCH

We interviewed 403 Development and IT Professionals using a 15 minute online survey.

Profile of companies:

- 500+ employees in company
- All verticals except ISVs and Education

Participant's primary role in organization:

- Dev Team (n=100)
- IT Operations (n=103)
- Test (n=100)
- Project Mgmt/Enterprise Project Management Office (EPMO) (n=100)

Key topic areas:

- Stage of DevOps at organization
- DevOps approaches used at org
- Development practices used at org
- Success metric ratings for six key areas for a focal application worked on
- Project outcome metrics for focal app

Cluster analysis

Given the challenges around defining DevOps, and getting a simple answer to the question of whether someone is "doing DevOps", we decided to run a statistical *cluster analysis* on data from a study focused on Agile Development and DevOps practices. A cluster analysis is an exploratory technique that can reveal if there are underlying segments in your data that include participants who behave similarly to each other, and different from those in other segments. The box to the right shows the questions used as inputs to our analysis, including stated DevOps stage, development methods used at the org, and three sets of questions related to development and DevOps methods, practices, and activities.

This research was sponsored by Hewlett Packard Enterprise and conducted by YouGov



CLUSTER ANALYSIS INPUTS

- Stated DevOps stage (Researching/Evaluating, Piloting, Partial, Widespread; s8)
- Development Method (Combo of Methods/Other, Agile Only, Hybrid Only; s7 recoded)
- What specific DevOps principles and approaches have you implemented or are you planning to implement in the next 12 months? (D4_)
- D4r1: Acquired or built automation tools for end to end use across development, testing, staging, and production environments
- D4r2: Created formal processes that link across Dev and IT Ops domains (e.g., value stream mapping, including deployment as part of an overall process map)
- D4r3: Created joint metrics used by Dev and IT Ops
- D4r4: Used better communication tools for Dev and IT Ops (e.g., ChatOps, Video conferencing, Messaging)
- D4r5: Created joint shared dashboard used by Dev and IT Ops that provides visibility into pipeline (D4r5)
- To what extent do each of the following statements reflect development practices in your company? (D8_)
- D8r1: Shift to a product or service focus from a one-off project focus
- D8r2: Our teams are seeing more engineers taking a hybrid Dev-Test role
- D8r3: The ratio of testers to coders is decreasing (i.e. the proportion of Testers employed is smaller than before)
- D8r4: Coders are taking more responsibility for their own testing than
- D8r5: Automated testing is increasing
- D8r6: Trunk development (as opposed to multiple release branches)
- D8r7: Have testing centers of excellence on release trains that provide end-to-end testing of integrated software
- D8r8: Developers and testers use the same tools for development and testing
- D8r9: Testing is done in production environment
- D8r10: Use of containerization (e.g.Docker) in test
- D8r11: Use of containerization (e.g.Docker) in production
- D8r12: Teams Build, Run and Own their applications/ products (No more throw it over the wall)
- D8r13: Use of Scaled Agile Framework (SAFe)
- D8r14: Security is more important than speed
- Three "Continuous" questions... Please indicate the extent to which your organization is using each of the below approaches (D9_)
- D9r1: Continuous Integration (Automatically building and running unit tests periodically or even after every commit and reporting the results to the developers)
- D9r2: Continuous Delivery (Software development discipline where you build software in such a way that the software can be released to production at any time. Involves continuously integrating the software done by the development team, building executables, and running automated tests on those executables in increasingly production-like environments to ensure the software will work in production)
- D9r3: Continuous Deployment (The practice of AUTOMATICALLY releasing and installing every good code build to production and end users. Continuous deployment is the next step of continuous delivery: Every change that passes the automated tests is deployed to production automatically)

Given there is no single "right" cluster solution, a number of analyses were run that involved variation in the exact inputs used, as well as the number of segments involved. We landed on a solution with four segments that we called DevOps Majority, DevOps Laggards, Formal DevOps Leaders, and Informal DevOps Leaders, based on the

characteristics of each segment. The graph to the right shows the distribution of the segments, with the DevOps Majority including about half the sample, the DevOps Laggards accounting for another quarter, and the two DevOps Leaders segments accounting for 10% each. We start by examining the pattern of results on the variables used as inputs to the analysis, and follow that with additional profiling of the segments by variables not used as inputs.

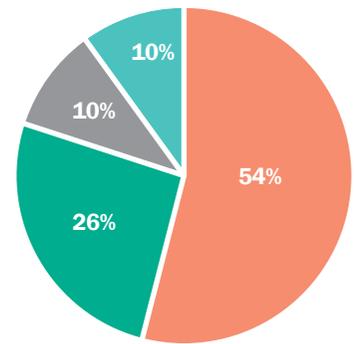
Profiling the segments

The graph to the right depicts the segments by what the participant said when asked about the extent to which their organization has implemented DevOps. Perhaps the most interesting finding is that the segment we dubbed Informal DevOps Leaders says they aren't doing DevOps, and that they are still researching/evaluating approaches. The Formal DevOps Leaders look the way you might expect, with most saying they have widespread implementations, with the rest saying they have partial implementations. The reason why the segment of organizations called the Informal DevOps Leaders, who say they aren't doing DevOps, are actually leaders in DevOps, will become clear by the end of this paper.

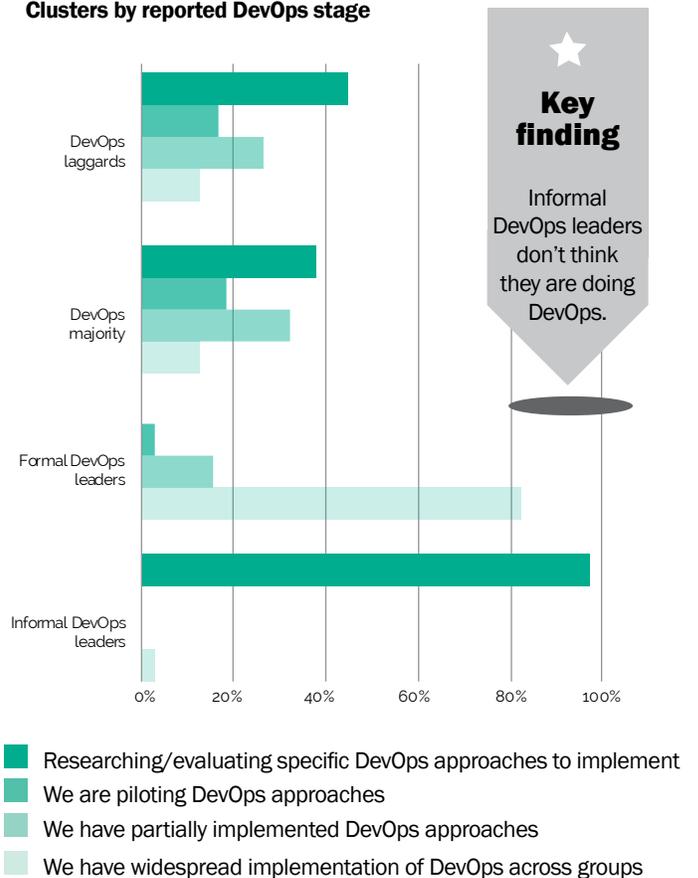
Below are the segments sliced by the development method(s) used at their organization, including Agile Only (who only use Agile), Hybrid Only (who only use Hybrid, or sometimes called WaterScrumFall), and Combo of Methods/Other (which mostly consists of organizations who use a mix of approaches). As can be seen, the Laggards use all types, the Majority and Formal DevOps segments use all types, but a large percentage are Combo, and the Informal DevOps segment almost exclusively uses Agile-Only.

Size of segments

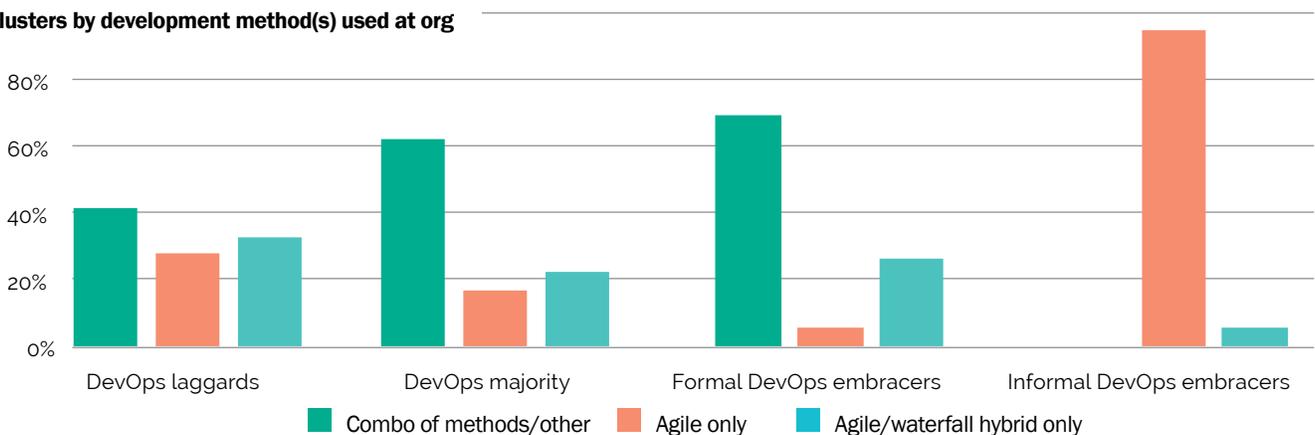
- Informal DevOps leaders
- Formal DevOps leaders
- DevOps laggards
- DevOps majority



Clusters by reported DevOps stage



Clusters by development method(s) used at org



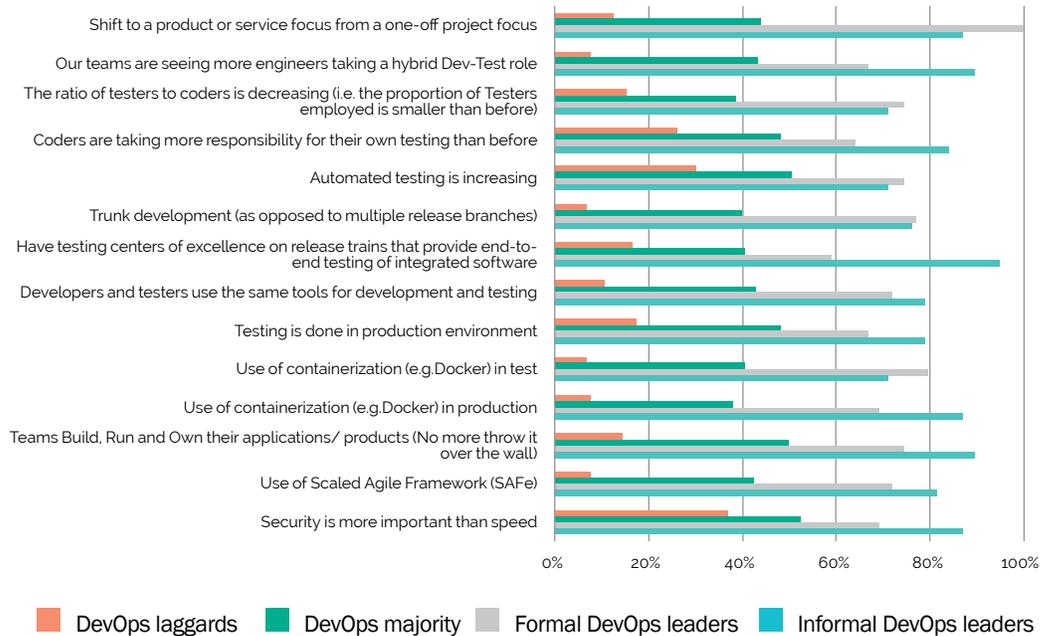
Below we show the segments sliced by the rest of the variables that were used as inputs to the analysis. The Development Practices questions involved a 4-pt rating scale, with the top-box percentages corresponding to the response “definitely reflects development practices at my org” shown below. As can be seen the Formal and Informal segments are highest across the board, followed by the Majority, and then the

★

Key finding

Informal DevOps Leaders have the highest usage, followed by Formal DevOps Leaders.

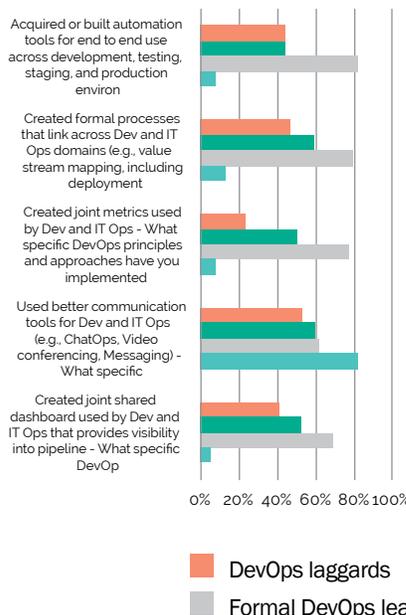
Development practices used at org



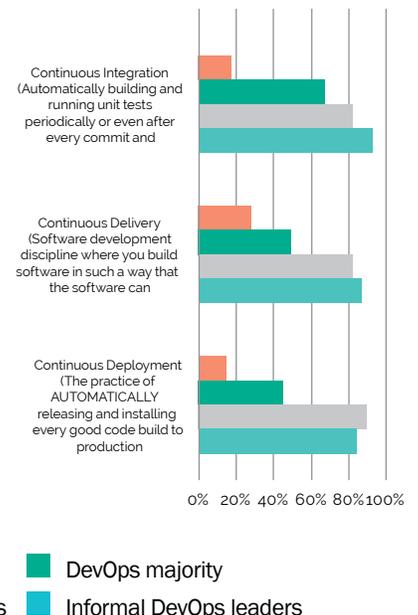
Laggards. The Informal Leader segment is higher than the Formal Leader segment on most practices, with the exception of *shift to a product or service focus from a one-off project focus*, and *use of containerization in test*, where the Formal Leader segment is higher.

The below graph depicts the segments sliced by the other two sets of questions used as inputs, including some DevOps specific approaches, and three questions on continuous integration, delivery, and deployment. For both questions the participant just had to indicate if their org used the approach or not. The pattern for the continuous questions sort of mirrors what we saw above, with the Formal and Informal segments reporting higher usage than the Majority, who in turn reported higher usage than the Laggards. The pattern for the DevOps principles and approaches is a bit different. The most striking pattern is the low reported usage of all of the practices by the Informal Leaders group, with the exception of using better communication tools

DevOps principles & approaches



“Continuous” approaches

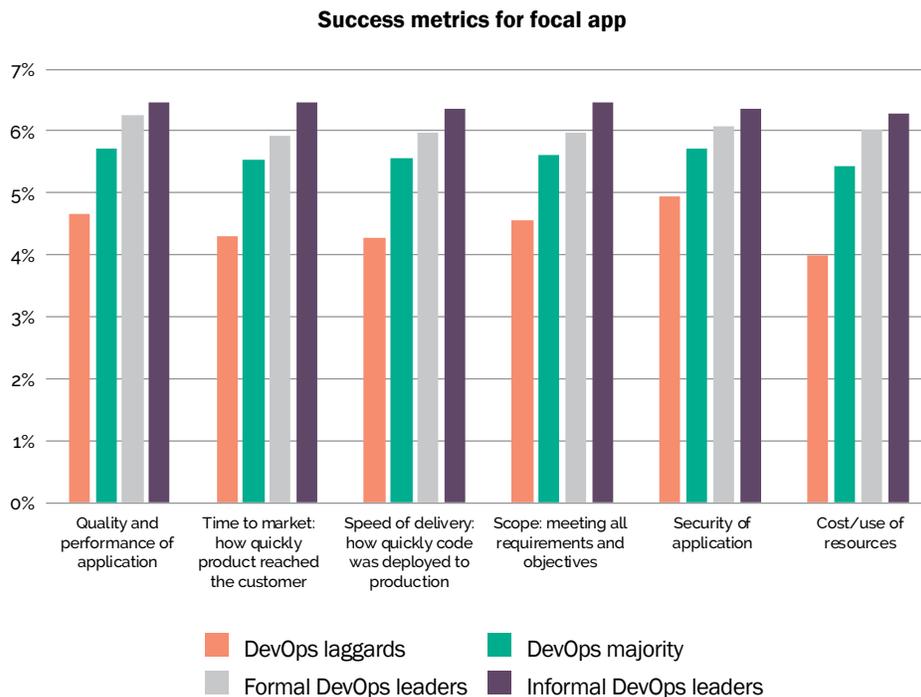


for Dev and IT Ops. That includes no end to end automation tools, no formal processes that link across Dev and IT Ops, no joint metrics used by Dev and IT Ops, and no shared dashboard. It is the pattern of results on this set of questions that earned this segment the *informal* designation.

We also profiled the segments by some other considerations that were not used as inputs to the segmentation. Perhaps the most interesting results related to three considerations, including *company size*, *public cloud usage*, and the *geographic distribution of the IT department*. Most of the interesting findings relate the Informal DevOps Leaders segment. First, almost none of the organizations in the very largest company size band (i.e., 10,000+ employees) were in the Informal DevOps Leaders segment. Second, the Informal DevOps Leaders leverage the public cloud much more than any of the other segments. And third, most of the organizations in the Informal DevOps Leaders segment are distributed across the US, as opposed to being more centrally located in one office.

How successful are the various segments?

Now we examine how successful organizations in each segment are on development projects. Participants were asked to rate their experience on the most important development project they worked on in the past 12 months, including a number of questions related to the success and outcome of the project. The metrics cover a range of considerations typically used to gauge the success of development projects. The question was: “For the focal application, please rate the degree to which the following goals were met”; 7pt scale, *Fell Short of Goals to Exceeded Goals*. As can be seen, success systematically increases across the segments, with the Laggards the least successful, followed by the Majority, then the Formal DevOps Leaders, with the Informal DevOps leaders having the most success.



★

Key finding

Informal DevOps Leaders have the most success, followed by Formal DevOps Leaders, based on six distinct success metric ratings.

Participants were asked several additional questions about the focal application to better characterize the pace and quality of the development project. On the next page we show the segments broken out by those questions. The first graph show *release frequency*, and *percent changes requiring remediation*. The patterns are complex. First, both the DevOps Leaders segments release the application more frequently. While the Formal DevOps Leaders are the only segment to achieve on-demand releases, a reasonable percentage of projects in this segment are released much less frequently.

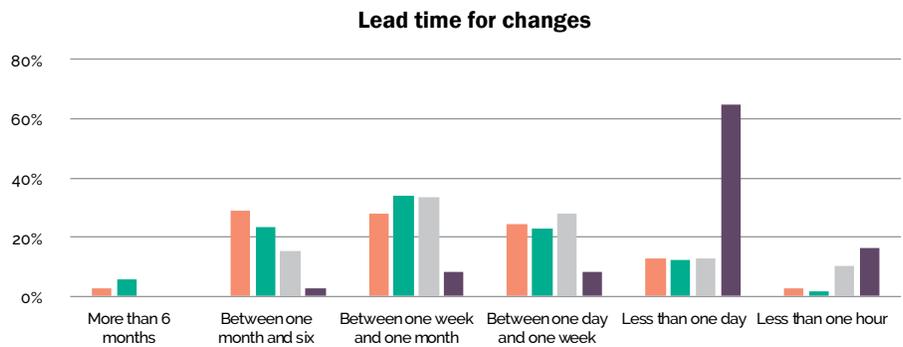
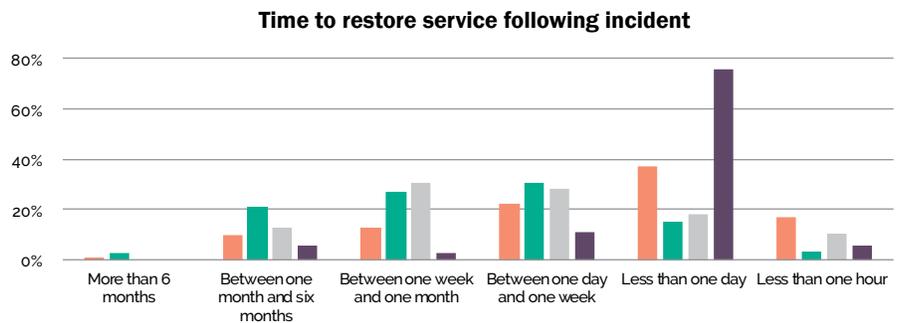
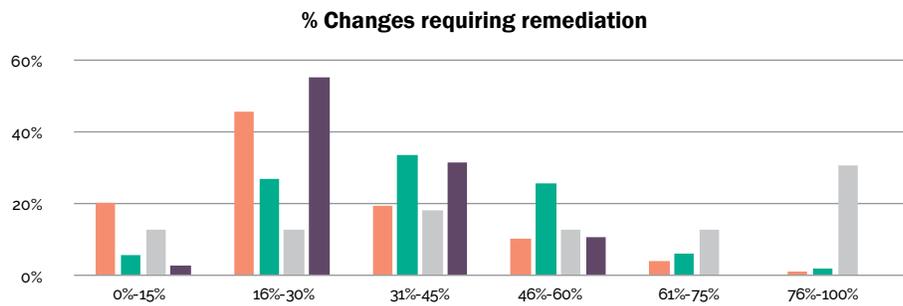
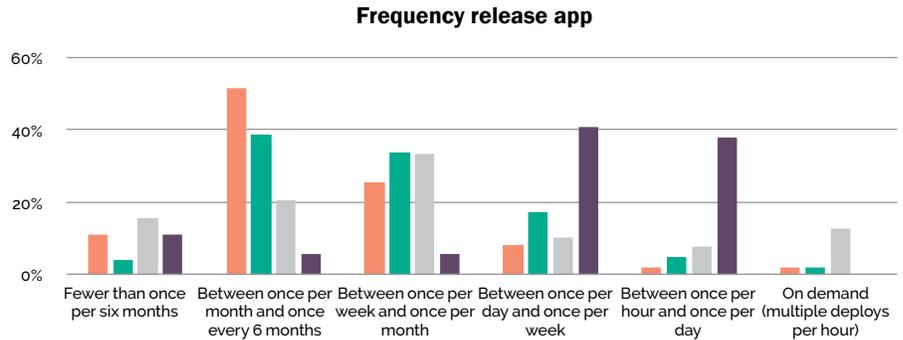
The pattern for the Informal DevOps Leaders is a bit different, with little on-demand, but high levels of daily to weekly releases. The pattern for remediation is a bit surprising on the surface, with the Laggards showing the lowest levels of remediation. But this segment probably includes the most conservative of organizations, so perhaps it's not surprising that the mistakes are kept to a minimum. Besides that trend, perhaps the most surprising data point is that roughly a third of those in the Formal DevOps Leaders segment cite remediation rates of 76-100%. Given they are the only segment pulling off on-demand releases, it appears that may come with a price.

Below are the results of two additional questions related to speed, including *time to restore service*, and *lead time for changes*. For the time to restore service question the most striking trend is the high level of *less than one day* for the Informal DevOps Leaders segment. The other trend worth noting is the relatively high level of *less than one hour* for the Laggards. Given this represents less than one fifth of that one

★
Key finding

Informal DevOps leaders deliver faster with higher quality compared to Formal DevOps Leaders, who don't fare much better than the DevOps Majority.

segment, these may represent those organizations that take a very conservative approach and who have systems in place to ensure no downtime. The most interesting finding for lead time for changes is for the Informal DevOps Leaders, with a large percentage of their projects involving *less than one day* to *less than one hour* lead times.



Conclusion

Using development and DevOps activities to segment organizations reveals two different approaches to DevOps, one formal, and one informal, with the informal approach leading to more success. It appears that adopting a DevOps mindset, and seeing it as a journey and a series of pilots and experiments, is more fruitful than focusing on the formalization of common practices.

HPE ALM Octane

HPE ALM Octane is an Application Lifecycle Management (ALM) software offering for Agile and DevOps environments, designed to enable software development and testing teams to harness the proven benefits of DevOps and Agile development to deliver software with speed, quality and scale. HPE ALM Octane provides insights to developers and testers, helping them deliver applications quickly, without sacrificing quality or end-user experience.

[Start your free HPE ALM Octane trial today.](#)