https://pmakhakhei.github.io/

от кода до прода за 1 час

https://pmakhakhei.github.io/

Павел Махахей

Software Testing

Team Leader

- 2008 года в ЕПАМ
- занимался автоматизацией тестирования на .NET, Java, JavaScript
- консультант в центре компетенций по тестированию

ProQuality

Agenda

https://pmakhakhei.github.io/

WHY CONTINUOUS TESTING?

SYRVEY INTRODUCTION

SUMMARY ON CONTINUOUS TESTING MATURITY

KEY TECHNOLOGY LANDSPACE

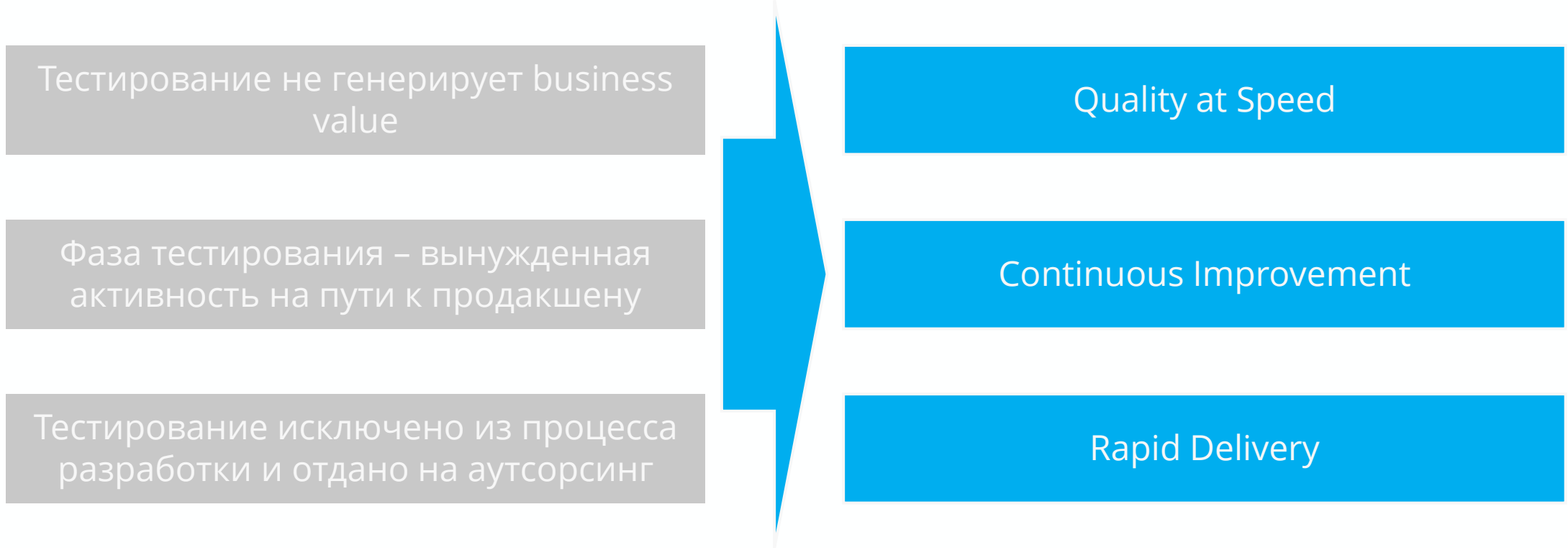
TEST PROCESS MATURITY

CONTINUOUS DEPLOYMENT & DELIVERY

TEST AUTOMATION LANDSCAPE

https://pmakhakhei.github.io/

Трансформация тестирования



https://pmakhakhei.github.io/

Survey Introduction

Overview

585
RESPONDENTS

377
PROJECTS

167
CUSTOMERS

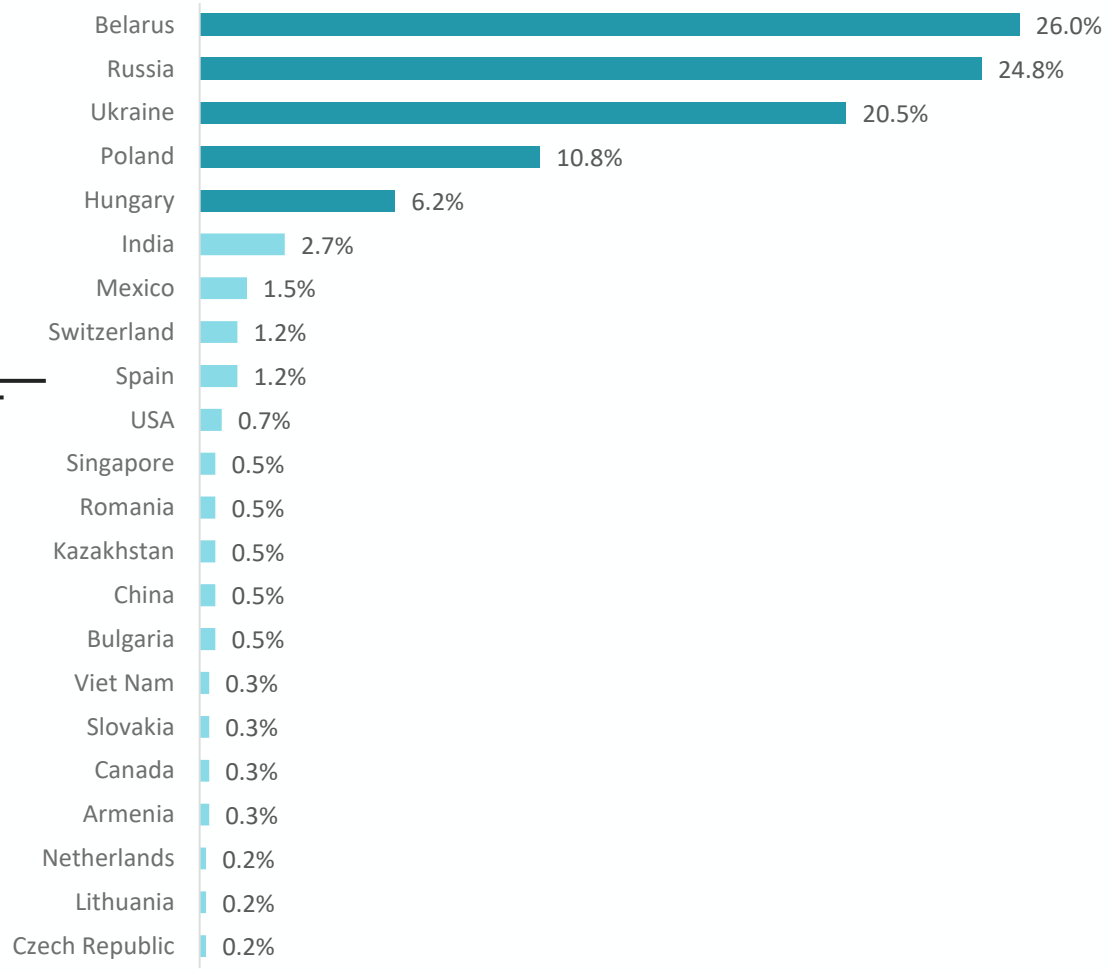
57
REGIONS

https://pmakhakhei.github.io/

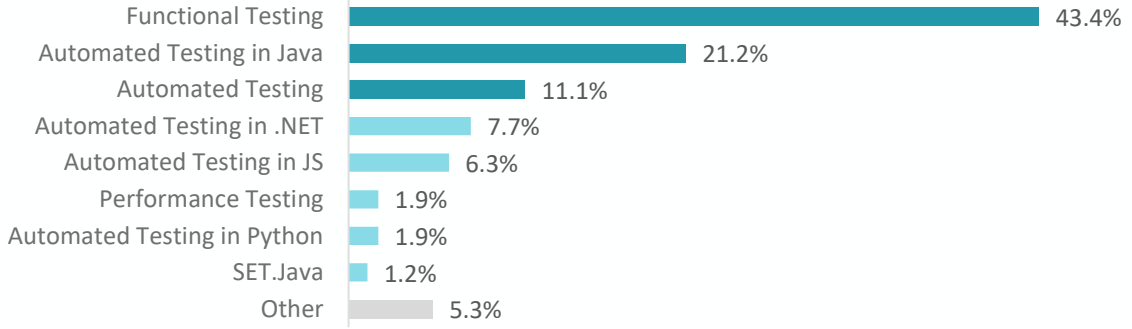
Survey Introduction

Respondents Summary

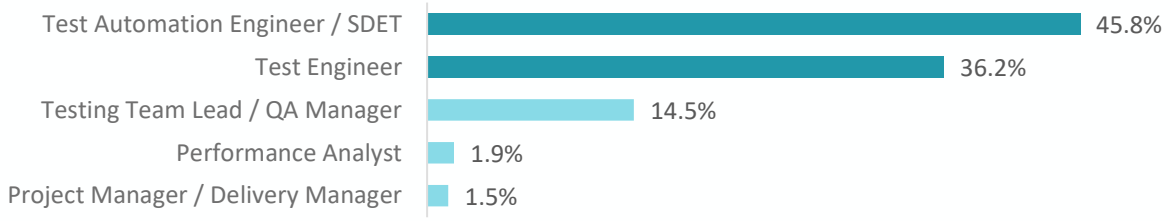
By Location



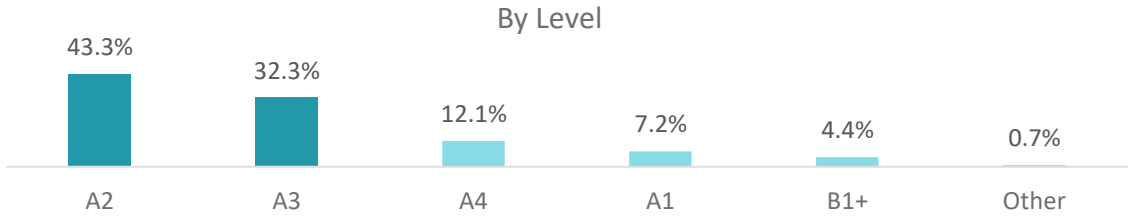
By Primary Skill



By Project Role



By Level



https://pmakhakhei.github.io/

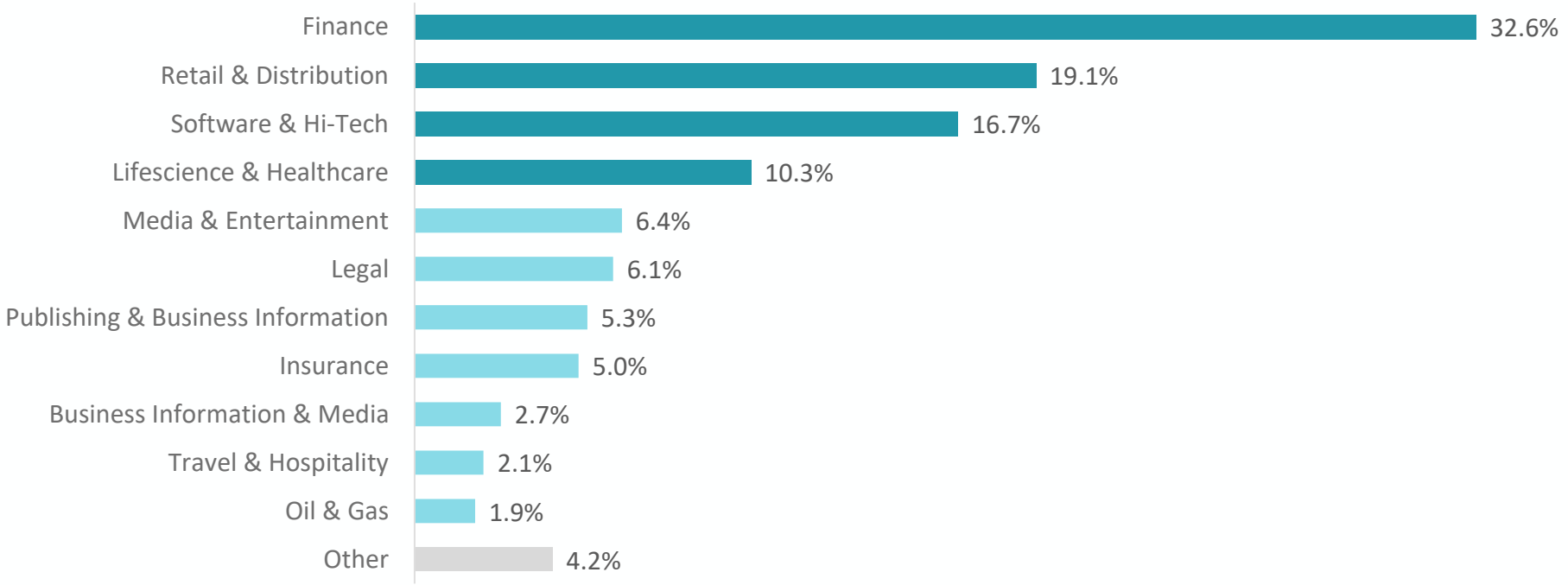
Survey Introduction

Projects and Customers

Total Distinct Count

377	167	17
PROJECTS	CUSTOMERS	DOMAINS

Represented Domains by Project Distinct Count



Summary on Continuous Testing Maturity

https://pmakhakhei.github.io/

Project Segments

UNICORNS: 2%

Top performers which can release features to Pre-Production or Production instance within **1 hour** and typically have high Unit test coverage **>80%,** high test automation coverage **>80%**, heavy focus on non-UI tests **>50%** and fast feedback on test results **< 1 hour**. Serverless architecture is distinctive for this segment.

HIGH PERFORMERS: 6%

Projects which can release features to Pre-Production or Production instance within **1 day** are typically represented by Cloud-Native applications with Microservices-based architecture. They are mainly characterized by mature test pyramid with **> 80%** API tests and high level of test automation maturity.

TRANSITIONING PROJECTS: 11%

This segment is represented by projects which can deliver to Pre-Production or Production instance within **1 week.** Analysis shows that test automation level is close to high performers with **> 90% pass ratio** however the differentiator is much longer duration of automated suite: **around 8 hours** and higher level of GUI tests **> 50%**.

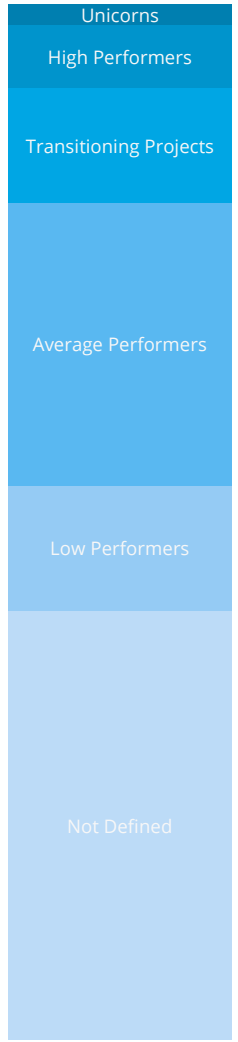
AVERAGE PERFORMERS: 27%

Projects which deliver new functionality within **1 month** and mostly having purely manual testing activities. Typically, tightly-coupled architecture is used often based on legacy technologies. If test automation exists, the maturity is not high with low level of automation coverage, large number of tests and disbalanced test pyramid.

LOWER PERFORMERS: 12%

Projects with full validation cycle for new features **longer than 1 month** and typically having low functional automation coverage **< 50%**, low Unit test coverage **< 50%**, unstable tests with pass ratio **< 50%,** high number of functional tests **> 5000** and test run time **much longer than 8 hours**.

Projects Segments



Key Insights

https://pmakhakhei.github.io/

Higher Performers

More than a third of projects deliver new features within 1 week

The percentage of projects adopting mature Test Automation and DevOps practices and capable to deliver features at least weekly is **38.2%** and the percentage is roughly stable YOY. Analysis also shows **9.8%** projects can deliver changes daily and **2.65%** - hourly.

13% projects have Continuous Deployment up to Pre-Production

Analysis of distinct projects shows **13%** projects have fully automated CI/CD pipeline and can deploy up to staging or pre-production environment and **4.5%** can deploy automatically straight to production instance after all defined quality gates passed.

Non-Functional quality gates are influencers on Continuous Testing maturity

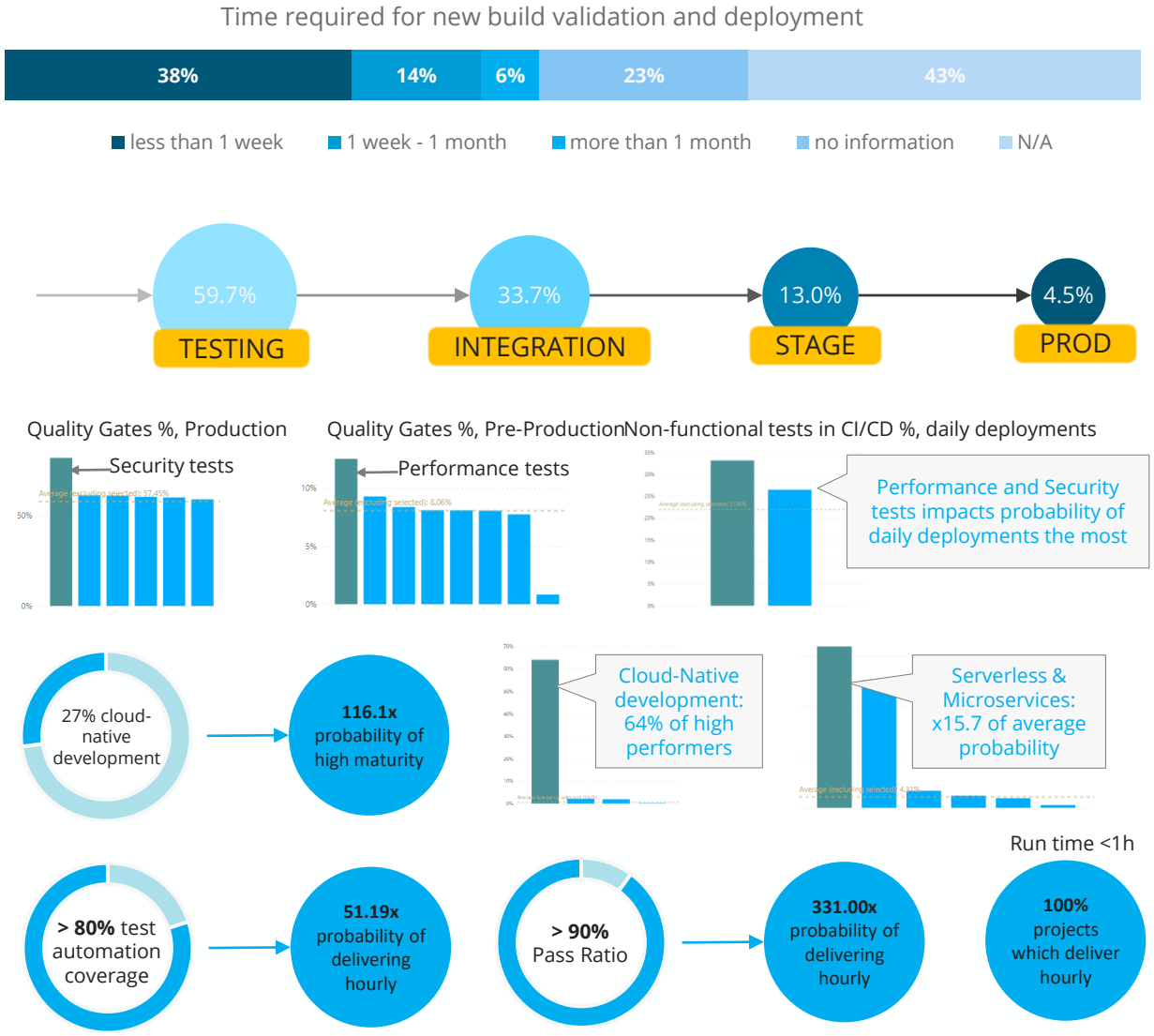
While functional tests serve as quality gates on many projects, analysis shows non-functional quality gates impact continuous deployment to Pre-Production and Production instances the most. When **security and performance tests** are quality gates in pipeline, the chance of continuous delivery increases by **1.4 and 1.57 times**.

Cloud technologies highly facilitate Continuous Testing adoption

When Cloud-Native technologies are used, the probability of project maturity is **116x** more likely to be High Performer. **Serverless and Microservices** architecture have comparable influence on high level of continuous testing maturity.

Automated run duration is crucial factor for high maturity projects

Analysis shows that the key differentiator for projects with comparable test automation maturity capable of delivering features within **1 day** is test automation run duration which is less than **1 hour**. Other typical factors are high test automation coverage and high average pass ratio of automated tests.



ProQuality

Key Insights

https://pmakhakhei.github.io/

Lower Performers

23% of projects have full validation cycle for new features 1+ month long

Projects which can deliver features within several months typically have increased number of functional tests in comparison with others and immature or inefficient test automation practices. Typical segment representing such projects show they have more than **5000** automated tests though low automation coverage **< 50%**.

Ineffective test automation impacts delivery worse than pure manual activities

Analysis shows that no automation adopted on the project increases probability of delayed releases (1 month or more) less than unbalanced test pyramid, low pass ratio (<50%) and long running tests which increased risk of late delivery by 7, 53 and 4 times correspondingly.

Outdated technologies for automation may influence speed of delivery

As an example, when Visual Basic language was used for implementation of automated tests, the risk of having low maturity in Continuous Testing increased by 21 times. This might be caused by implicit factors related to execution environment or run time.

Legacy architecture and lack of Cloud technologies is typical for low performers

When Cloud technologies are not used, the probability of delayed delivery is increased by more than 6 times. Most often the architecture of system under test for low performing segment is Service-Oriented architecture, N-Layered architecture or Monolithic.

Large number of tests with low pass ratio is key impediment for fast feedback

Key factors which have impact on low continuous testing maturity include high number of tests **> 5000** and low pass ratio **< 50%**. Analysis shows that projects with ineffective test automation which requires increased effort for maintaining, execution and analysis in parallel with manual testing activities slows down delivery the most.

What influences the probability of being Low Performer the most

when...	...the likelihood of project being low performer increased by
functional test automation level **< 50%**	76.37x
pass ratio for automation runs **< 50%**	53.22x
programming language for automation is **Visual Basic**	21.04x
Unit test coverage **< 50%**	18.53x
number of functional automated tests **> 5000**	13.00x
Cloud technologies **are not used**	6.59x

Technology Landscape

https://pmakhakhei.github.io/

Architecture & Cloud

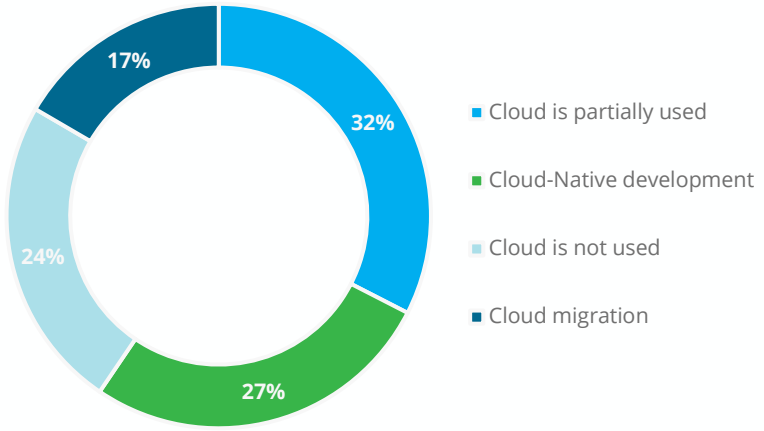
Serverless & Microservices architecture is a trend

Analysis of technologies and architecture styles used across projects shows the trend for increasing adoption of loosely coupled architecture: Microservices and Serverless architecture styles prevail over others and percentage of projects using them increased from 35% in 2020 to 43% in 2021 while percentage of Monolithic architecture decreased from 18% to 10%.

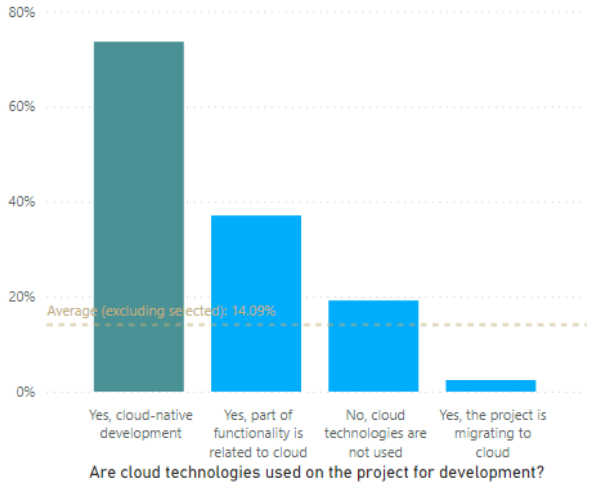
Cloud adoption has greatest influence on architecture style

The probability of using Service-Oriented, Microservices or Serverless architecture increases the most when Cloud-native development is used or at least part of functionality is related to Cloud. From other side, when the project was migrating to Cloud, the probability of architecture being Monolithic increased by 6.31 times.

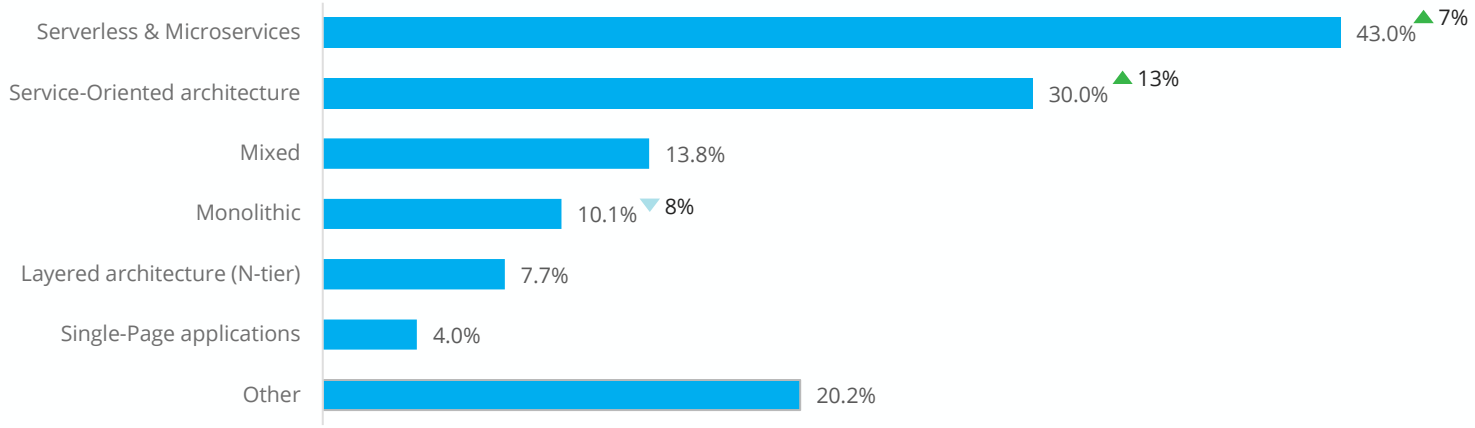
Usage of Cloud Technologies



Serverless or Microservices and Cloud



Architecture Style Used on Project



Test Process Maturity

https://pmakhakhei.github.io/

Summary by Projects

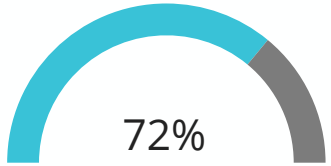
Non-functional tests have lower automation level

Average level of integration in CI/CD for non-functional tests is **36%** while it is **52%** for functional tests. Though level of automation for performance tests is 10% higher than in 2020. Part of projects integrating security tests in pipeline remains at 9% level and still low while analysis shows both performance and security tests have crucial impact on capabilities of continuous delivery.

Functional test automation level remains stable

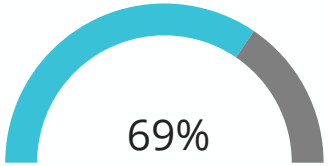
Level of integration of functional tests into pipeline and automation health remains relatively stable with highest percentage for BVT, Smoke and Regression tests and close to average numbers in 2020 with minimal fluctuations. Analysis shows that low automation level often depends on architecture style with lower automation level for Monolithic applications. Traditionally, automation coverage is lower for IoT and Telecom domains.

Automation level by projects %



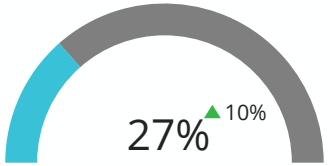
72%

have **smoke test** in pipeline



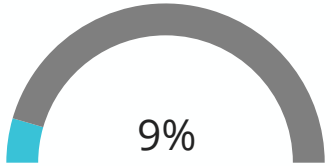
69%

have **regression** tests in pipeline



27% ▲10%

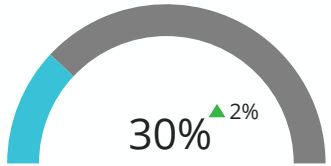
have **performance** tests in pipeline



9%

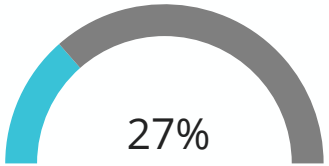
have **security** tests in pipeline

Automation health by projects %



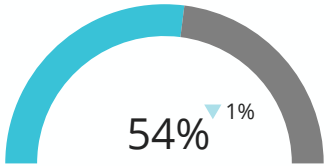
30% ▲2%

have Unit test coverage **> 80%**



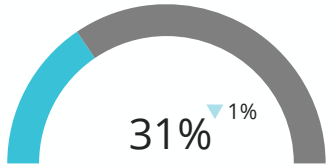
27%

have functional test coverage **> 80%**



54% ▼1%

have pass ratio **> 90%**



31% ▼1%

have **> 80%** tests on non-UI level

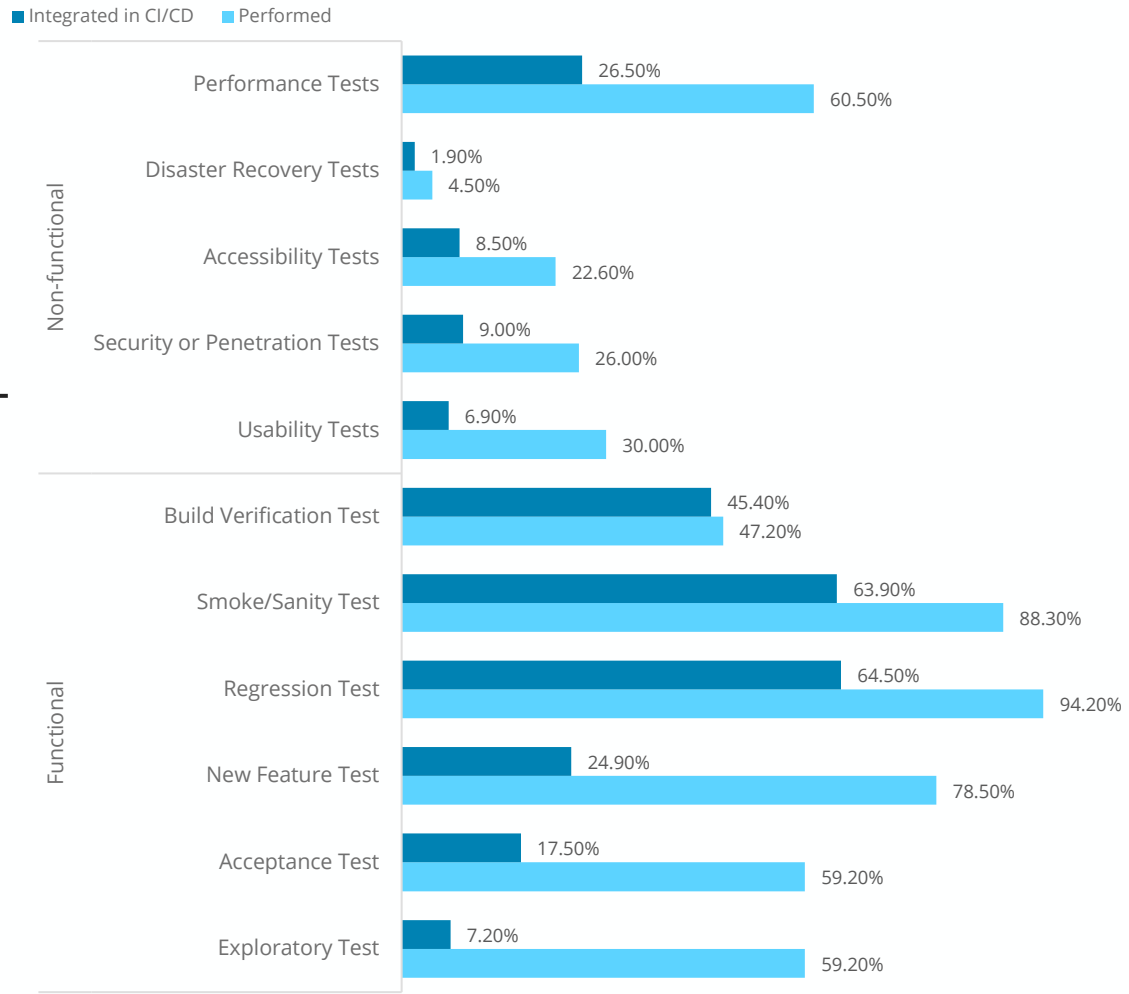
ProQuality

Test Process Maturity

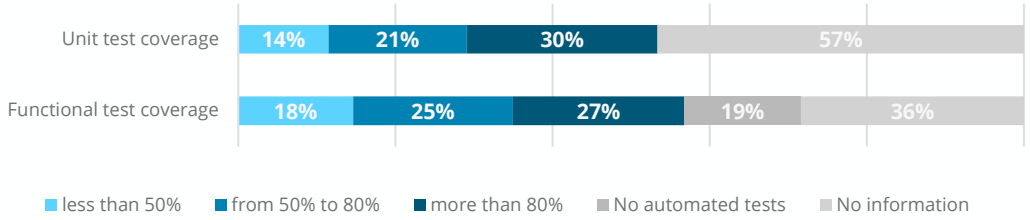
https://pmakhakhei.github.io/

Automation Level

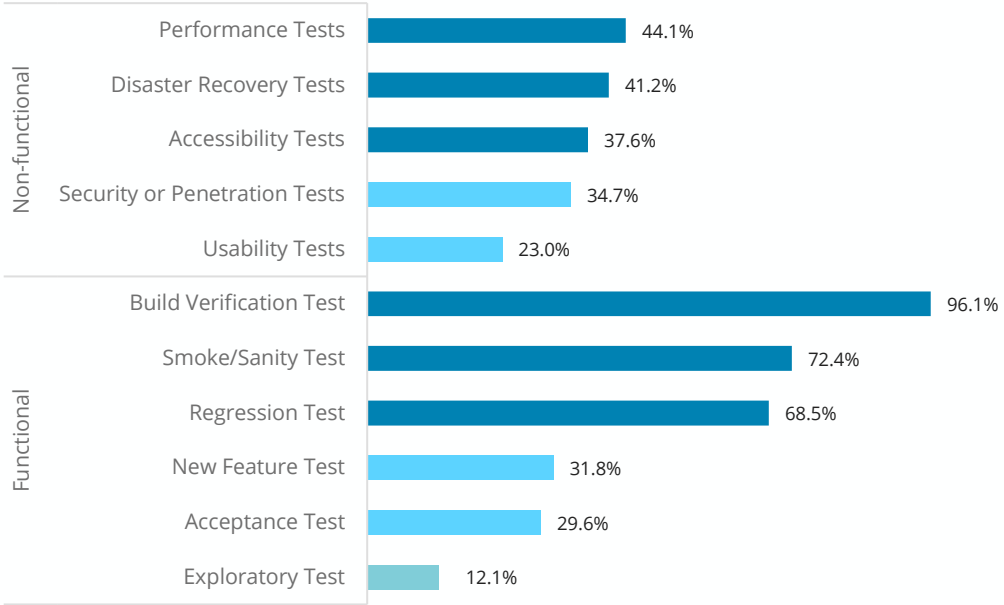
Test Types Executed by Projects %



Distribution of Projects by Automation Coverage



Level of Integration into Pipeline by Projects %



Test Process

https://pmakhakhei.github.io/

Automation Health

Pass Ratio and Run Time are direct influencers on delivery

Typical segment with high delivery speed includes projects with pass ratio > 90% and execution time < 1 hour. When average Pass Ratio was > 90% the probability of delivery within 1 hour increased by 282 times. And vice versa: the probability of late delivery increased the most when Pass Ratio was < 50%.

Test pyramid impacts automation stability the most

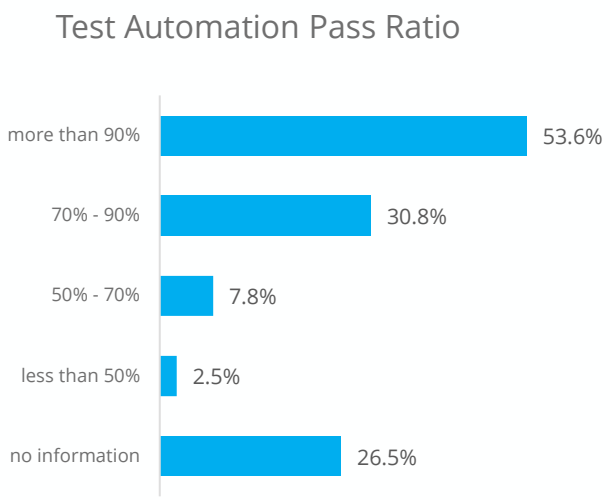
Percentage of non-UI tests had the greatest influence on average Pass Ratio. When percentage of non-UI tests was **< 50%** the probability of low Pass Ratio **< 50%** increased by **156** times. While the probability of Pass Ratio being **> 90%** increased the most when more than a half of tests were implemented on API level.

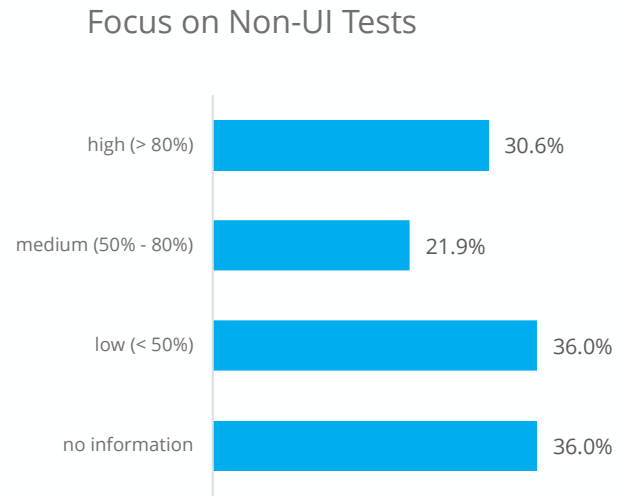
Duration of automated run is not in direct ratio to test count

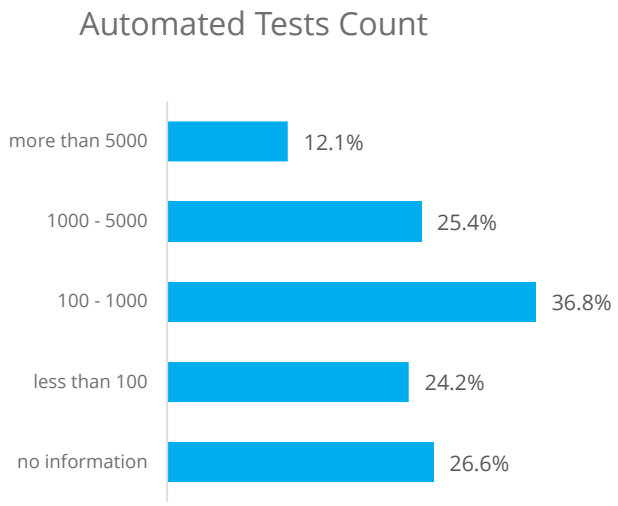
Though test run time correlates with the number of tests, it is not in direct proportion to it. When percentage of non-UI tests was higher **80%** the probability of run time to be **< 1 hour** increased by 5 times. Time required for test execution can significantly decrease depending on test levels and test execution platform used.

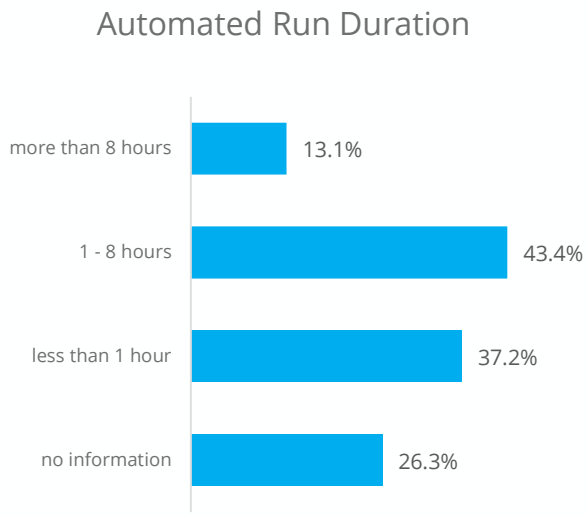
Pass Ratio and run duration are also interdependent

Analysis shows that the longer tests are running the more probability of lower Pass Ratio. Highest stability of automated tests is typical for project with large number of tests though fast execution: < 1 hour.





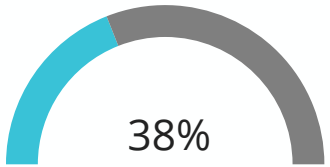




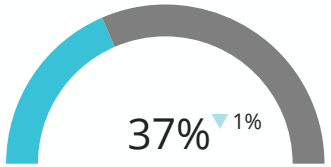
Continuous Deployment & Delivery

https://pmakhakhei.github.io/

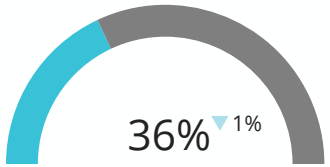
Summary by Projects



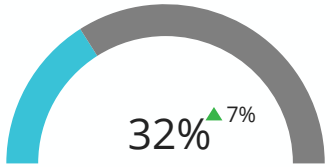
can promote new feature **within 1 week**



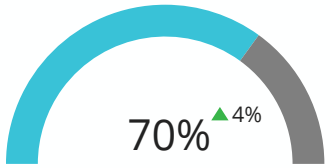
have **static code analysis** as QG in pipeline



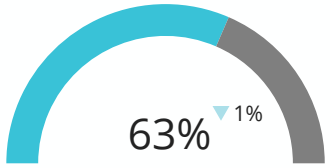
have **smoke test** as QG in pipeline



have **regression** tests as QG in pipeline

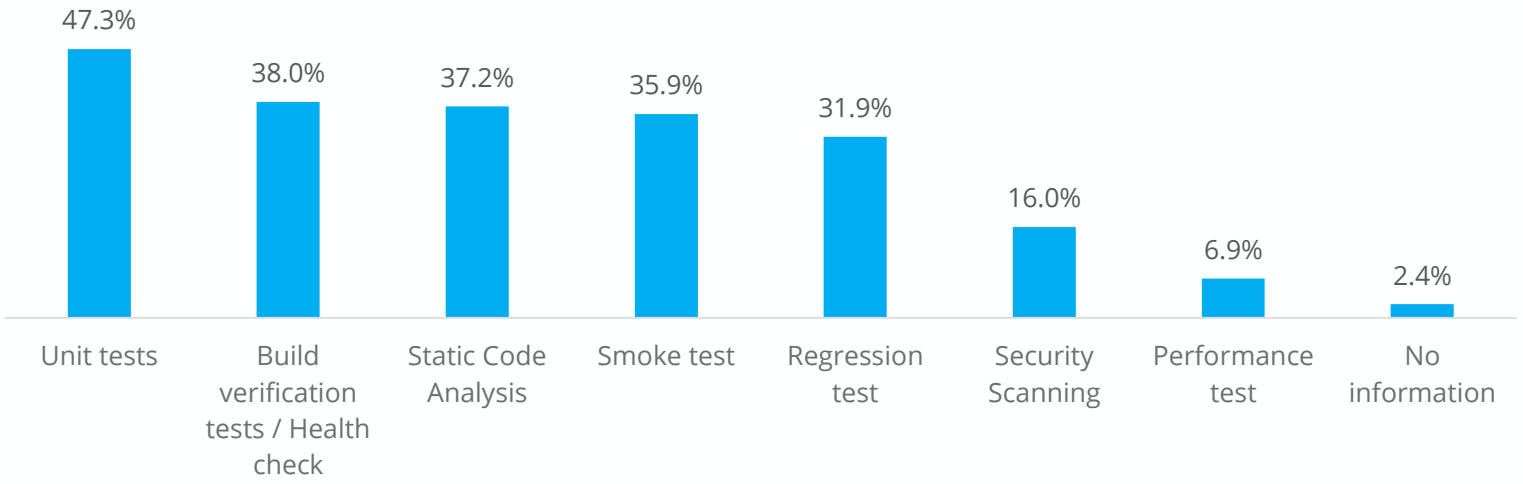


have automated process of **deployment**

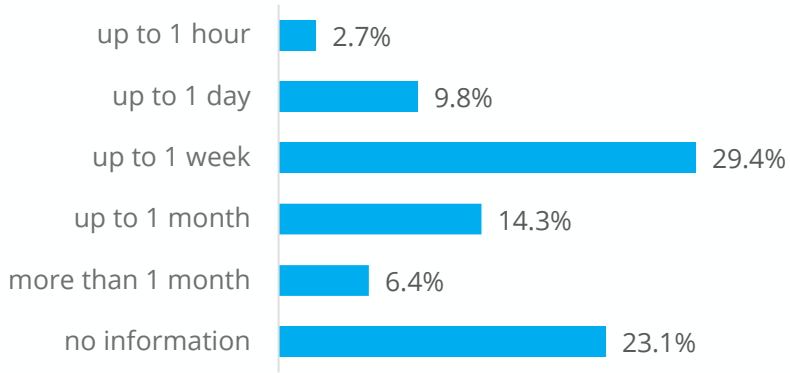


have centralized **reporting** system

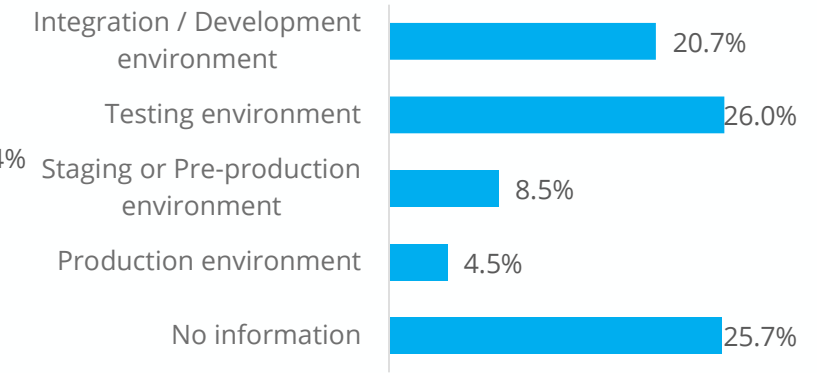
Quality Gates Applied on Projects



Full Validation Cycle of Build



Ultimate Environment



ProQuality

Test Automation Landscape

https://pmakhakhei.github.io/

Languages and Tools

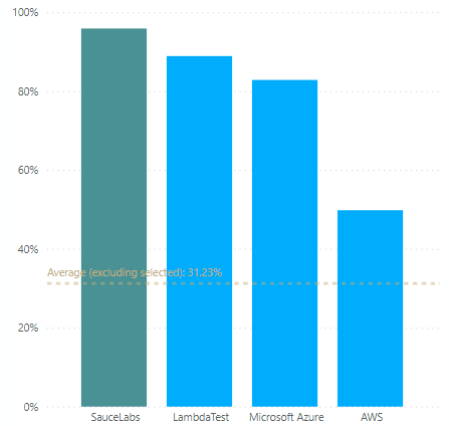
Selenium is still most popular tool but not influencer on maturity

The share of Selenium is still very high: **> 60%** of distinct projects reported they use it for UI automation. However, Selenium is not differentiator for project with highest continuous testing maturity. **Cypress**, **Webdriver.IO** and **Playwright** are relatively new players on the market but are the tools of choice for higher performers.

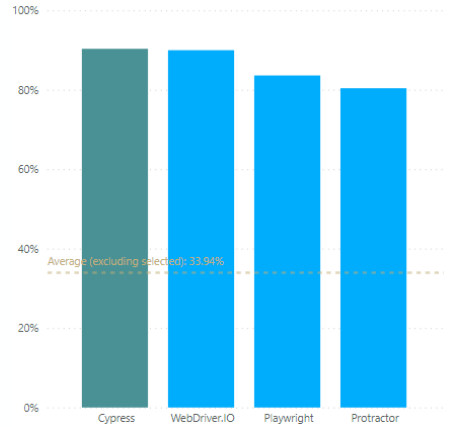
Cloud execution platforms are differentiators for top performers

Sauce Labs, **LambdaTest**, **Microsoft Azure**, **AWS** are used with higher probability for top projects capable of delivery within 1 hour. While high performers tend to use **Kubernetes**, **Docker**, **AWS** as platform for continuous test execution.

Execution platforms for high performers



Automation tools for high performers



Test Automation Landscape

https://pmakhakhei.github.io/

Programming Languages

JavaScript breaks into 2nd place

JavaScript solely hits the second place in popularity after Java with **6%** growth comparing to 2020-year results. Together with TypeScript, the percentage of engineers using one of language or both is close to **35%**.

Python and C# popularity grows moderately

C# for test automation is used on approximately **19%** of distinct projects and this is **3%** higher than in 2020. Python increased its share from **8%** in 2020 to **9.5%** in 2021.

Niche languages popularity remains unchanged

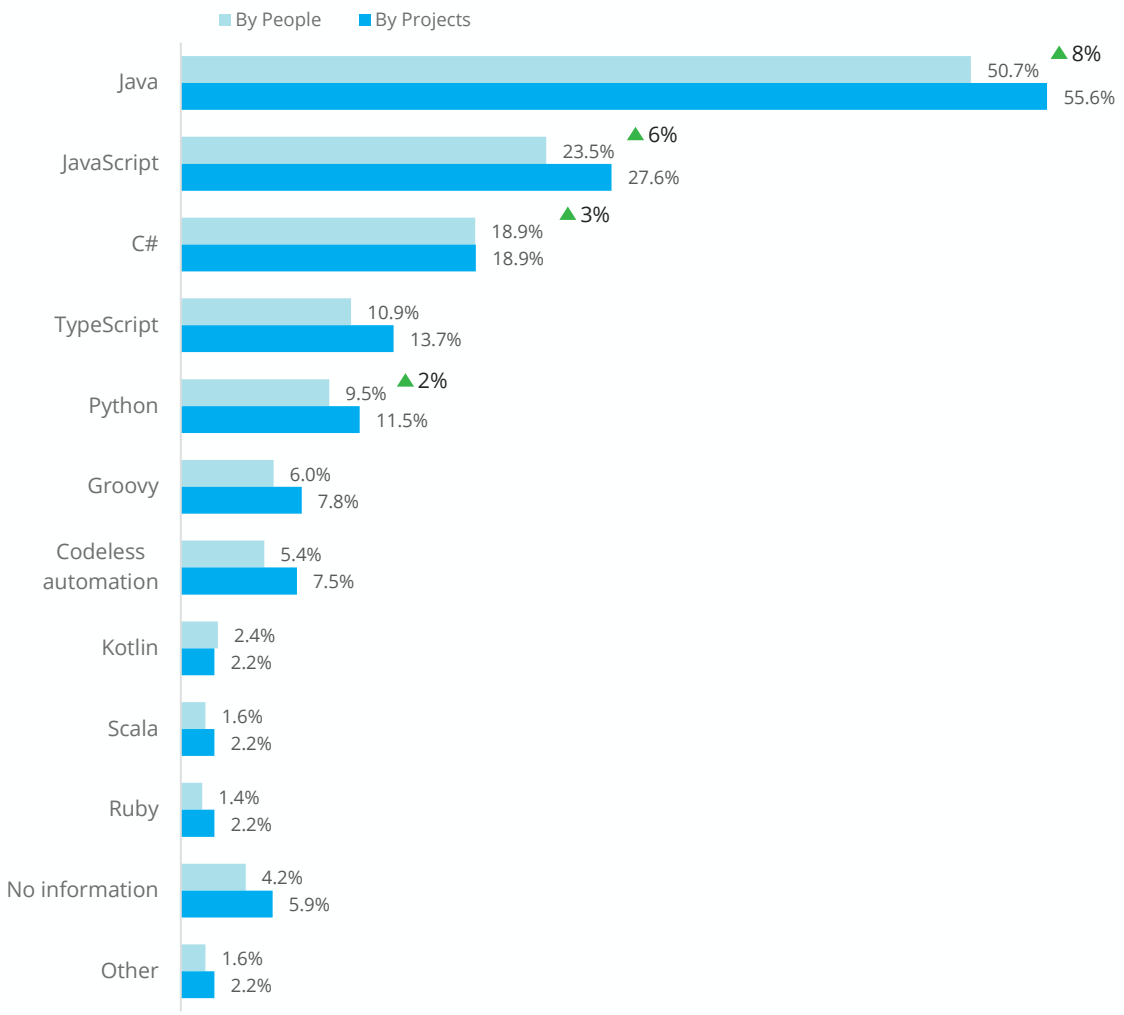
Kotlin, **Scala**, **Ruby**, **Golang** all together have less than 10%. Their popularity did not change from 2020 with only minor fluctuations within only tenths of a percent.

Codeless automation steadily grows year to year

There is no rapid change in Codeless automation level but the percentage of scriptless automation has stable increase from 2019 to 2021. Almost **8%** of project claim they use codeless automation and around **5%** of all respondents mention they do not use any programming language to develop scripts.

High Performers focus on niche program languages

While **C#** and **JavaScript** are languages of choice for unicorn projects, the hallmark of high performers projects is focusing on **Scala**, **Kotlin, Ruby or Python.**



ProQuality

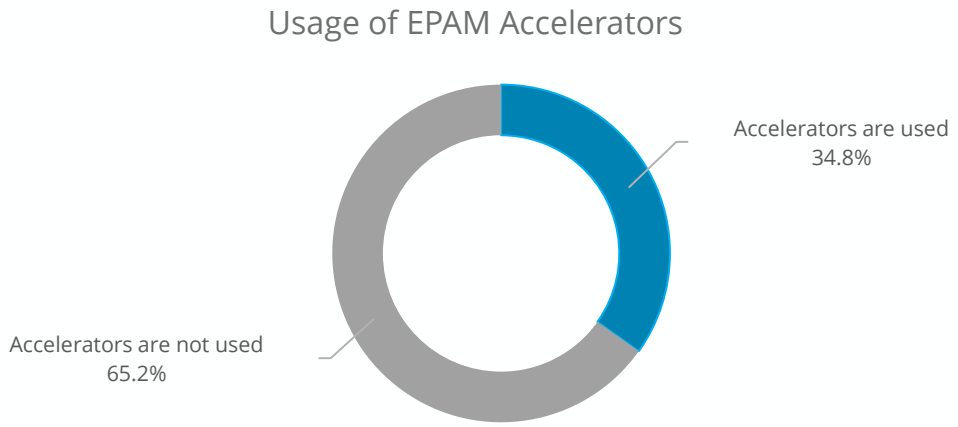
Test Automation Landscape

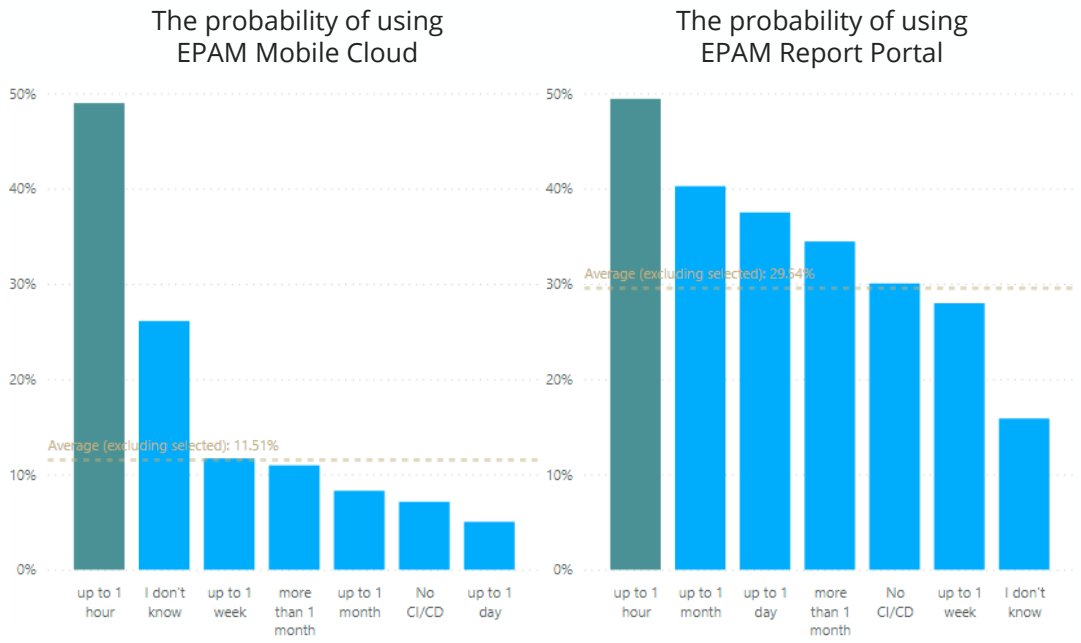
https://pmakhakhei.github.io/

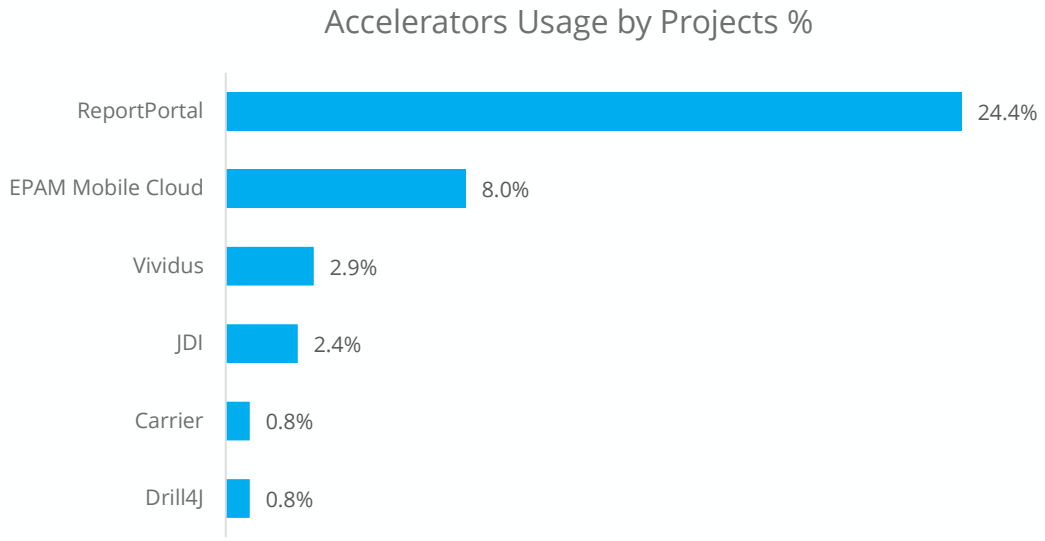
EPAM Accelerators

EPAM Mobile Cloud and Report Portal are most adopted accelerators

Together they are used on more than a third of all projects while Report Portal has higher adoption: **24%** vs **8%** for Mobile Cloud. However, the share of EPAM Mobile Cloud across unicorn and high performing projects is higher than share of Report Portal or other accelerators. When projects are capable to deliver within 1 hour, the probability of using either EPAM Mobile Cloud or Report Portal increase by 4.2x and 1.7x of average.





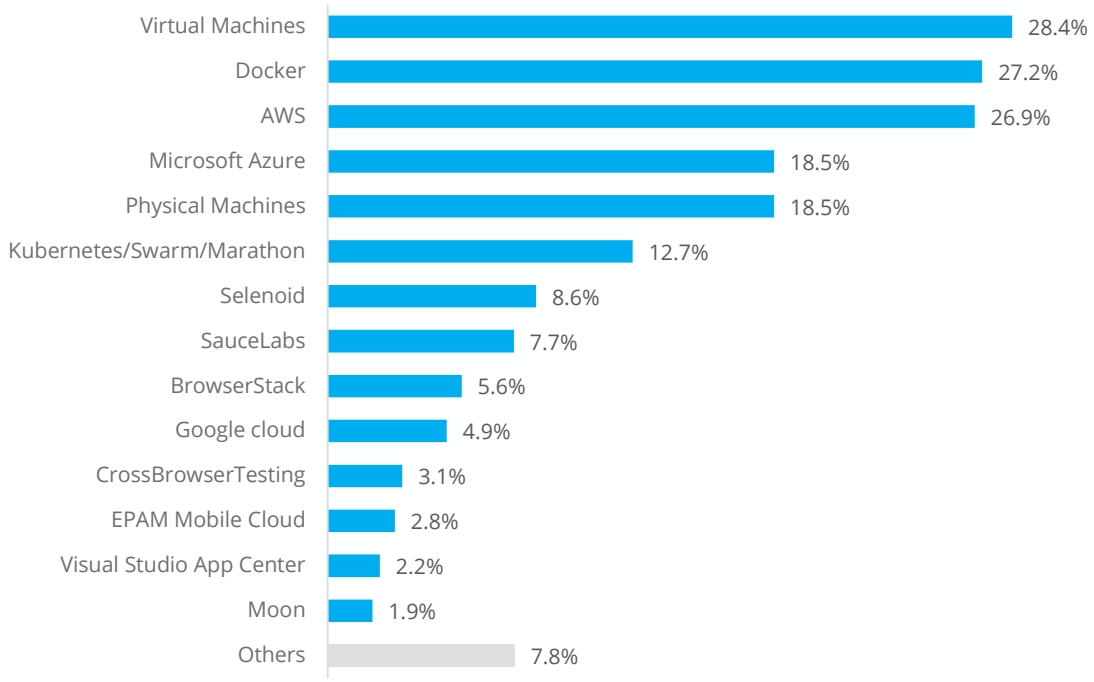


Test Automation Landscape

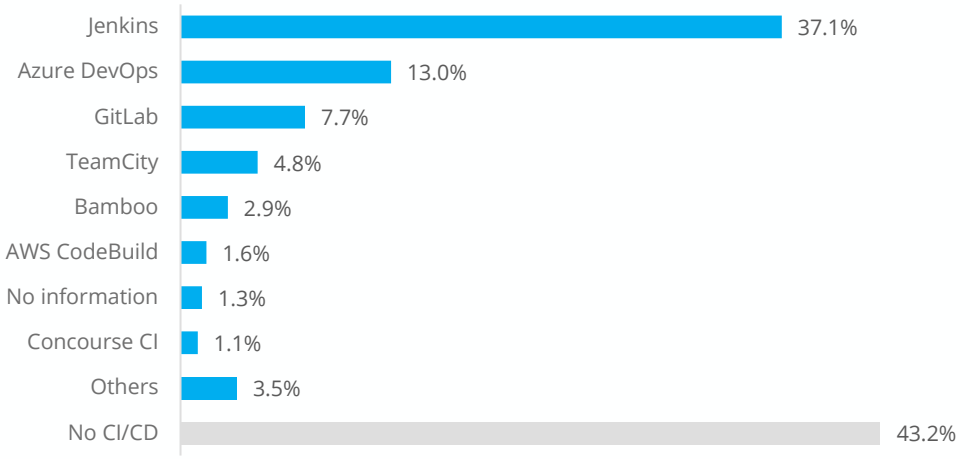
https://pmakhakhei.github.io/

Execution & Reporting

Test Execution Platforms



CI/CD tools used



Reporting Tools Used