



DevOps Ecosystem: Application Testing

Q1 2021

Kyle Howard-Johnson

Move Fast and **Don't** Break Things



DevOps was created to do two things: 1) accelerate time-to-value from when a developer starts building an application to when it is pushed into production (*Dev*) and 2) ensure the application operates at scale (*Ops*)

- DevOps tools and best-practices support these goals of accelerating time-to-value and operating at scale; however, as the market evolves and tools proliferate, so do the challenges associated with them:
 - Speed of Innovation: as opposed to rolling out quarterly or monthly updates, teams are now pushing out app updates daily (if not more frequently), creating more opportunities for something to go wrong
 - Infrastructure and Deployment Complexity: enabling technologies continue to grow rapidly, increasing the complexity of environments and deployment options for apps
 - Scale and Distribution: more users are consuming software in more places across more devices than ever before
- **Application testing** is just one piece of the broader DevOps ecosystem but one that faces an uphill battle due to the challenges mentioned above
 - As the speed of innovation accelerates, it correspondingly leaves less time for testing, exacerbated further by the increasing complexity of the underlying infrastructure and the diverse and rapidly scaling user base
 - While time-to-value and innovation feel like they have been prioritized above all else, it takes just one massive service outage (like those at [Slack](#) and [Zoom](#)) to demonstrate the importance of moving fast without breaking things

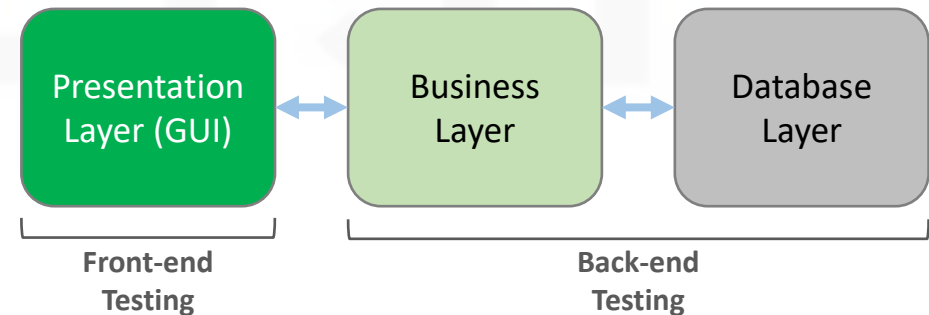
Testing remains the biggest bottleneck to getting new functionality in the hands of users with the desired velocity and quality

Back to the Basics



Application testing is typically carried out on the Dev side of DevOps and allows teams to identify errors during the development process (ideally as early as possible) and fix them before production

- There are two distinct approaches to application testing: manual testing and automated testing ^[1]
 - Manual testing is often used in the initial phases of development for testing specific features
 - Automated testing tends to be used once an app is closer to completion, facilitating scaled testing for usability, functionality and performance
- Software applications are categorized in three ways: web, mobile and desktop applications, with most apps published today coming in one of the first two categories but all with distinct challenges
- Application testing is conducted in two phases: front-end testing and back-end testing^[2], with a litany of test types falling into each of these buckets
 - Front-end testing includes unit testing, UI testing, compatibility testing, etc.
 - Back-end testing includes functional testing, structural testing, API testing, etc.



There is an ever-expanding set of test vectors that teams need to address before pushing code into production and a shrinking window in which they can test

Sources: [1] BrowserStack, [2] TestCraft

Tailwinds Driving Growth

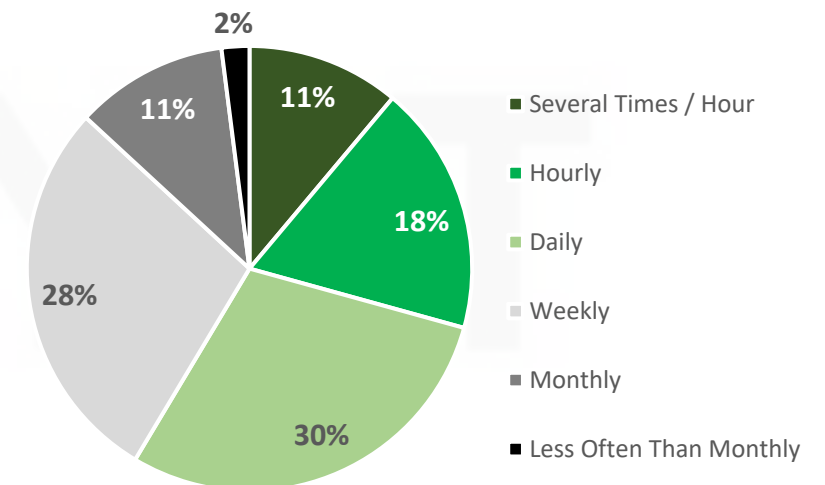


Historically, application testing has been personnel-driven, requiring human capital resources – teams are increasingly looking for ways to automate / outsource basic testing requirements, freeing up high-value resources to focus on driving incremental value

Between the DevOps software tools market and the broader tech ecosystem, specific trends we have observed driving growth in the application testing market are as follows:

- CI / CD reduces time to production, shortening the testing window
- Device / infrastructure growth increases deployment complexity while also expanding the number of test vectors
- Security awareness moves security testing earlier in the development process, alongside more traditional visual / functional testing practices
- Shortage of skilled engineers / developers creates an even greater need for automation in application testing
- Shift to remote work necessitates that quality assurance (QA) engineers can collaborate across testing environments

Ideal Deployment Frequency^[3]



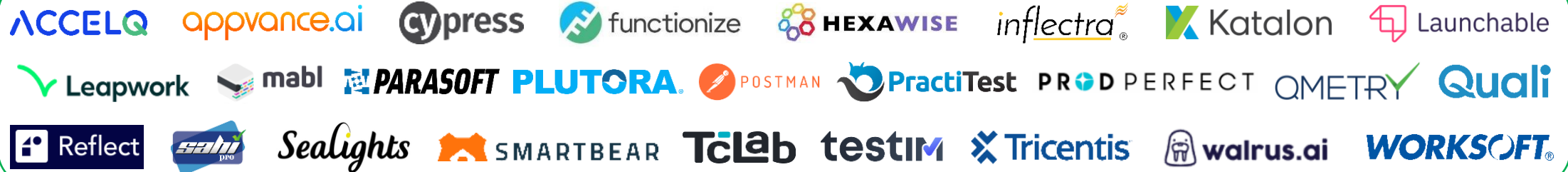
Testing platforms are capitalizing on these challenges, providing greater leverage to QA teams by requiring fewer people to manage testing across a greater number of apps, but the testing market remains fragmented with a variety of functionally-specific solutions

Sources: [3] Sogeti

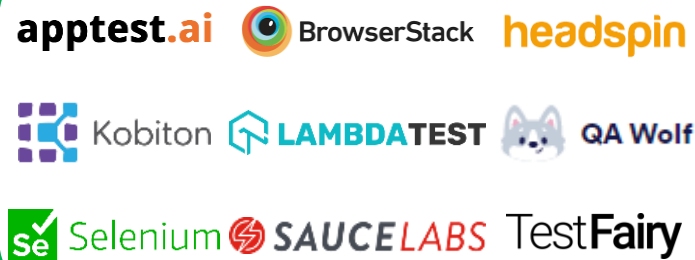
Application Testing Landscape



Test Case Management



Browser / Device



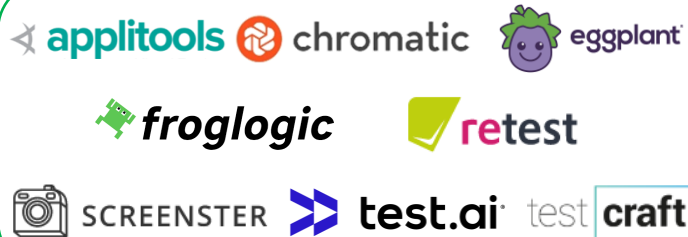
Performance



Code Review



Functional / UI



Crowdtesting



Security



Sources: Company websites

Future State of Testing

Internal Slide









For the reasons listed on slide 4, we anticipate the following trends to perpetuate in the near and medium term:

- 1) App testing will become increasingly automated
- 2) App testing will move earlier in the development lifecycle (shift left)
- 3) Security testing will become increasingly integrated with the development process (DevSecOps)
- 4) App testing will continue to require some level of manual input

We believe a successful app testing platform will:

- 1) Have a high level of automation
- 2) Provide leverage to existing testing resources (tangible ROI)
- 3) Integrate with applicable dev tools in project management, development and CI/CD

Opportunities Catalyst is actively exploring that fit into our thesis around the future state of testing:

Company	Description	Last Touchpoint
 Functional / UI	Automated visual testing tools for web and mobile apps	Jan-21: engaged William Blair for growth round
 Browser / Device	Mobile app testing platform that accelerates testing	Sep-20: intro call; evaluating M&A throughout 2021
 Security	Mobile app security testing platform	Aug-20: held call w/ CEO to discuss DevOps tools piece
 Code Review	Error-tracking platform to identify bugs earlier in development	Aug-20: intro call; raising round in 2021
 Test Case Management	Software quality governance platform for developers	Feb-21: check-in call w/ CEO; raising CAT round next year
 Crowdtesting	Testing community and platform for mobile app testing	Dec-20: quarterly call w/ CEO

DevOps Ecosystem Update

Internal Slide

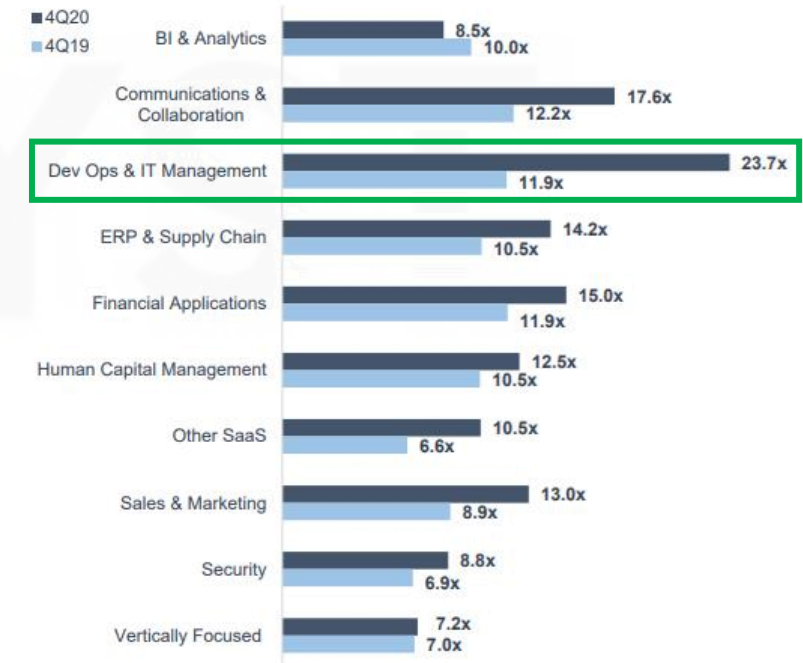


After publishing the *Evolution of the Software Development Lifecycle and DevOps Tools* piece, I began more systematically engaging with companies in that ecosystem, from which I learned the following:

- The DevOps tools ecosystem is incredibly hot right now with many high-profile VC and growth equity investors making big bets early and often, underwriting massive outcomes supported by frothy public market multiples (see chart)
 - This interest has led to greater competition and caused firms to invest earlier in companies' lifecycles, which means companies are raising 'growth rounds' before they reach our threshold
- The developer community has meaningful influence in terms of which products gain traction (something we have been exploring at EDB), feeding into the [Product Led Growth](#) movement that has been coined by OpenView
 - This virality has fundamentally changed the growth trajectory and economics associated with acquiring / retaining customers for many of these B2B companies, causing them to look more like B2C companies
- **App testing is one area that appears to have slightly less interest / hype than the rest of the ecosystem, which could represent an opportunity for Catalyst**

SEG: Product Category Performance

Median EV/TTM Revenue Multiple





CATALYST

INVESTORS

Please send any inquiries to kyle@catalyst.com