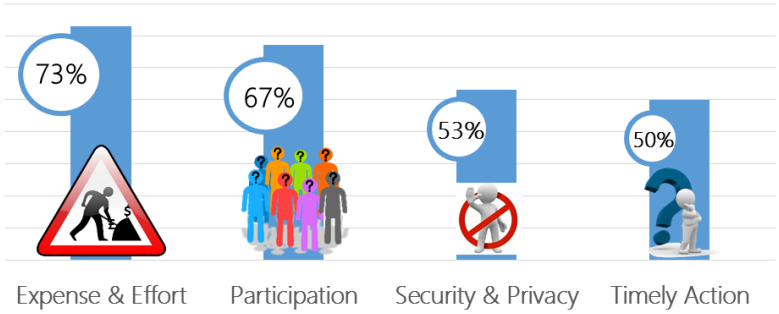
Developers speak: better βετα = better software

Development organizations reveal the value and the challenges of effective beta software program management.

In a survey distributed to over 10,000 professional developers, 8 out of 10 respondents reported that visibility into user activity and application quality through an organized beta program would significantly improve development productivity and increase the value of their applications. Yet, only 39% of those developers who recognized this value reported that their development teams consistently run beta programs. This report reconciles this contradiction by examining the risks and obstacles to effective beta program management identified by these same developers.

Risks and obstacles to effective beta program management

When asked to identify and rank obstacles that required significant preparation or represented material risk to an effective beta program, developers who value, *but do not regularly implement*, beta programs identified four.



Risks and Obstacles to Effective Beta Program Management

- Expenses and required resources were either too high or not allocated.
- Sufficient participation levels could not be guaranteed to assure meaningful results.
- Security, privacy, and compliance obligations were too onerous.
- Organizations could not act quickly enough or did not know what steps to take to make meaningful improvements even after beta feedback had been collected.

Beta blocked

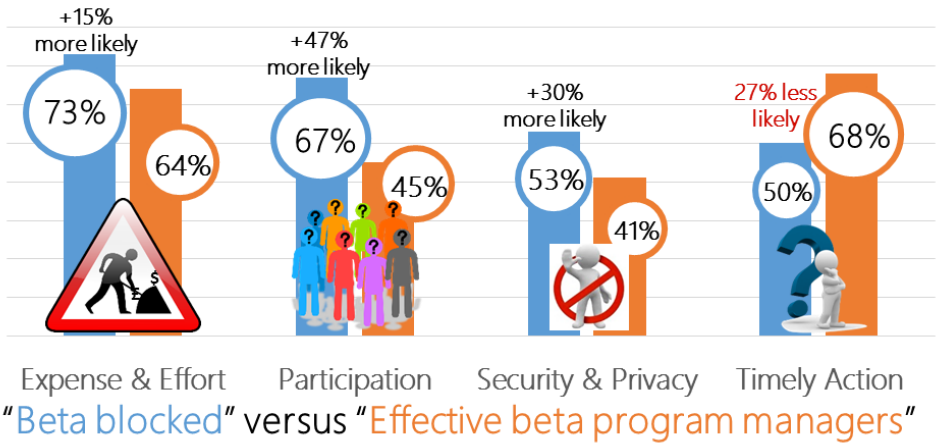
87% of developers who did not consistently employ beta programs while acknowledging the intrinsic value of those programs cited one or more of these four risk categories as the root cause of why they were effectively "beta blocked."

Overcoming barriers to effective program management

Is it possible for this group of *beta blocked* developers to work through what they perceive as insurmountable obstacles? Or, is there something about these development teams or their applications that make their risk profile or burden exceptionally high?

Not surprisingly, *beta blocked* developers were 300% more likely to identify one or more risks as being so significant that it "prevented any serious implementation of a beta program." However, those responses accounted for only 18% of the "beta blocked" developers. The remaining 82% ranked their "blockers" as either "an obstacle that must be anticipated and planned for" or, at most, "a significant obstacle that prevents all but the most extreme business and use cases."

Might these obstacles be overcome with a combination of new tooling and/or skills?



Comparing the risk and obstacle responses of those developers who are able to consistently employ beta programs (Effective Beta Program Management) with those that cannot (Beta Blocked) suggest that all of these factors play a role – and, as such, there is a significant opportunity to expand effective beta program management.

Expense & Effort: Beta blocked respondents were only 15% more likely to cite expense and effort as a risk to be managed and planned for than their Effective beta counterparts. What this suggests is that, while resource and expense management is always a requirement, the Effective beta teams are able to justify their investments while Beta blocked developers cannot.

Tools to reduce cost of beta programs and/or better evidence of the return on those investments are what these "beta blocked" developers are missing.

Participation: Beta blocked respondents were 47% more likely to cite an inability to collect feedback from a statistically significant user sample than their counterparts. There is a correlation between the application architecture and the likelihood that a development team will identify adequate participation as a significant risk to their beta program. Developers targeting thick PC

clients, native mobile clients, and cloud services were between 20-25% more likely to fall into the "beta blocked" category due to participation concerns than their counterparts.

Tools that can safely and efficiently monitor applications installed on 3rd party devices and/or running without a "presentation layer" are what these "beta blocked" developers are missing.

Security & Privacy: Beta blocked respondents were 30% more likely to cite burdensome security and privacy obligations than their counterparts. There was no statistically significant correlation between this risk and development platforms, number of targeted users, or number of targeted organizations. This survey did not track industry and therefore, without supporting data, we have to assume *that a combination of tools and policy (a kind of skill) would be required by these "beta blocked" developers*.

Timely action: This risk category stands alone from the other three in that it is ranked 27% more frequently by organizations that are running effective beta programs. In fact, Timely Action was the second most common risk category cited by "Effective beta" respondents and it was ranked at the very bottom of the list by Beta blocked respondents. This suggests that Beta blocked respondents, while generally aware of the challenges of efficiently collecting and distributing data, do not have a full appreciation for the workflow and processes required to actually effect change as a result.

In order to ensure the greatest likelihood of success, tools that can deliver usage and quality metrics quickly, to the proper roles, and in formats that can be readily consumed and acted upon are what these beta blocked developers will need.

Conclusions

The value of effective beta program management is broadly recognized across the application development community; regardless of architecture, development team size, or preferred technology stacks.

While a minority of development teams may truly be blocked from implementing a beta program, the overwhelming majority of development organizations should be incorporating beta program management into their development process. In spite of this, the majority of development teams are not.

Risks and obstacles are not confined to development teams that cannot sustain effective beta campaigns; they impact and impede development organizations that have managed to implement these programs in spite of these very same challenges.

Armed with the right tools and skills, most development organizations should be able to take advantage of the benefits of beta release programs to improve quality, usability, and – ultimately – application value.

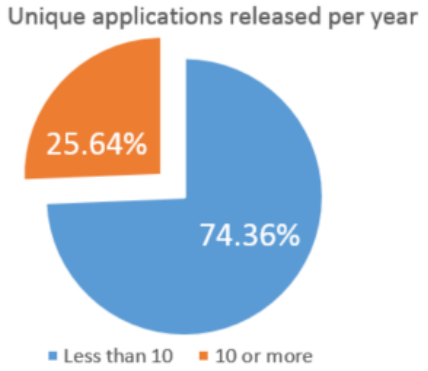
Appendix

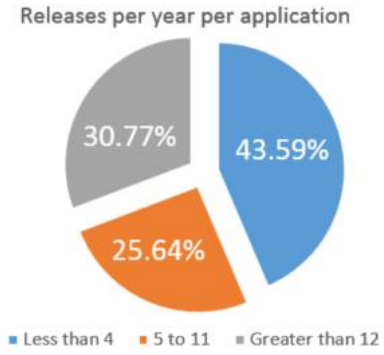
About this survey

The survey was distributed to over 10,000 professional developers during the week of September 13, 2015. Respondents' diversity of development scale, velocity, adoption, and technology are summarized below.

Velocity and output

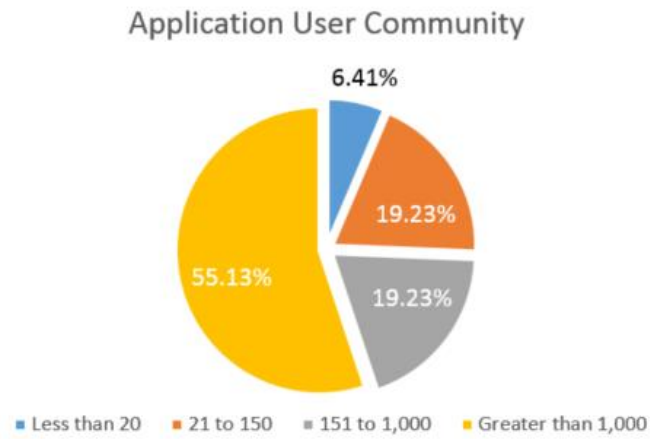
Number of distinct applications and releases of applications: how many applications were respondents producing per year and (as an average) how many releases per app per year do they release?

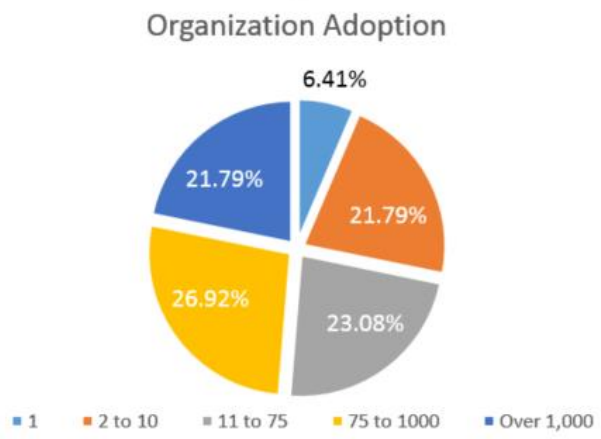




Adoption by user and organization count

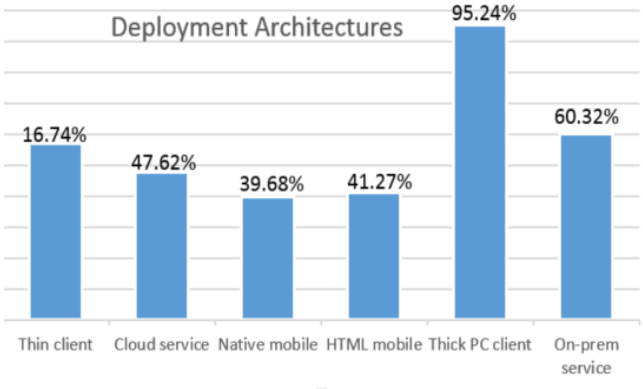
How many unique users are supported across all released applications and how many individual organizations do these users work within?

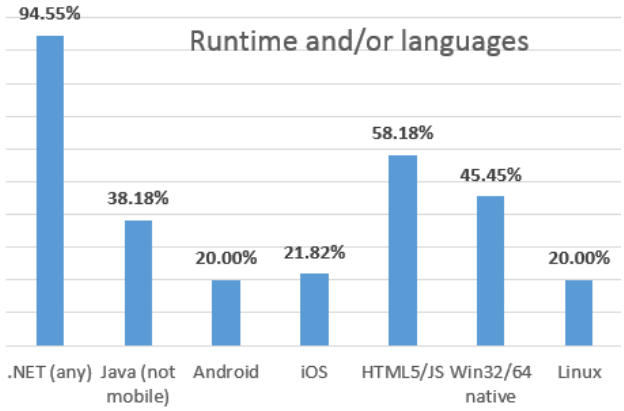




Platforms, languages, and architectures *(multiple responses permitted)*

Respondents were asked to identify the platforms, languages, and architectures where they currently had applications in production.





For more information

Visit www.preemptive.com/betterBeta to understand how PreEmptive Analytics

- Reduces the cost and effort of effective beta program management,
- Helps ensure maximum participation improving the quality of each beta release cycle, and
- Delivers usage and quality results in real-time to the stakeholders that need it most,
- While ensuring that privacy, security, and IP protection requirements are met.

About the author

Sebastian Holst, Chief Strategy Officer at PreEmptive Solutions, is responsible for product strategy and the development of best practices in application risk management and application analytics deployments. In addition to his work building development software, Sebastian has also been active in computing and industry standards bodies as a member of the W3C Advisory Committee, the MMA committees on Privacy and on Analytics, IDEAlliance board, co-founder the Compliance Consortium, and most recently, as a member of ACT, The App Association, where he testified before the US Congressional Committee on the Judiciary on the role of application analytics and Intellectual Property protection in fostering technology innovation.