

Shri Ramdeobaba College of Engineering and Management Nagpur

Name- Harsh Agrawal CSE SEM-6 Roll no-43 Section-B

Subject→ Software Engineering.

Under the guidance of Henna Agrawal Ma'am.

Teacher Assessment (Content Writing)

Topic→ Alpha Beta and Acceptance Testing.

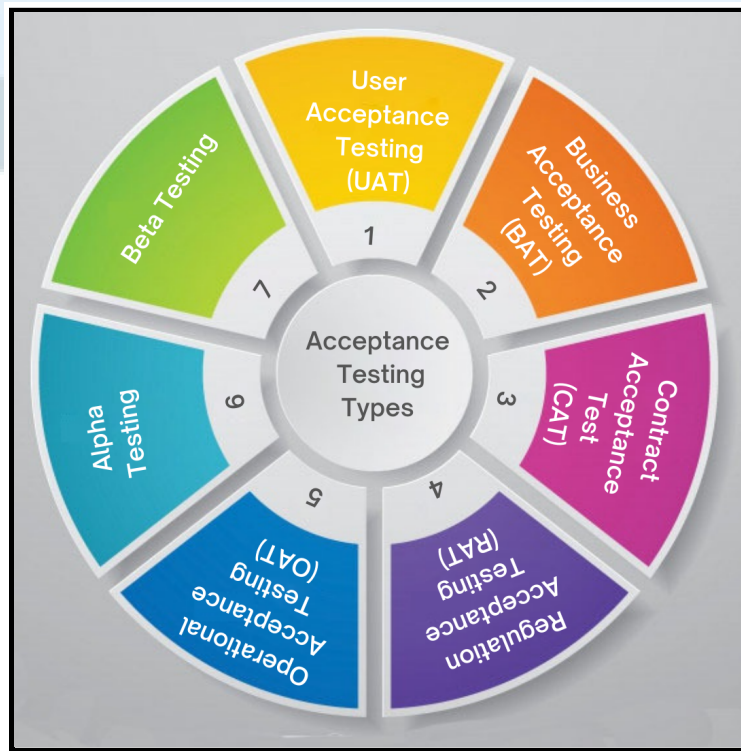
- **Alpha Testing** : Alpha testing refers to the system testing carried out by the test team within the development organization.
- **Beta Testing** : Beta testing is the system testing performed by a selected group of friendly customers.
- **Acceptance Testing** : Acceptance testing is the system testing performed by the customer to determine whether to accept or reject the delivery of the system.

Acceptance Testing is the final level of software testing. The main aim of this testing is to determine the working process of the system by satisfying the required specifications and it is acceptable for delivery. It is also known as End-User Testing. It also works under the **Black Box Testing Method**.

Acceptance Testing is a method of software testing where a system is tested for acceptability. The major aim of this test is to evaluate the compliance of the system with the business requirements and assess whether it is acceptable for delivery or not.

Use of Acceptance Testing:

- To find the defects missed during the functional testing phase.
- How well the product is developed.
- A product is what the customers actually need.
- Feedbacks help in improving the product performance and user experience.
- Minimize or eliminate the issues arising from the production.



Alpha testing is the final stage of testing performed by your QA team to check that your application is ready for release outside your company. The testing is coordinated in-house, structured and is usually done by your own test team. However, sometimes it involves real users (especially if the software is being created for a 3rd party). Alpha testing is predominantly about ensuring bug-free functionality.

Beta testing involves releasing the software to a limited number of real users. They are free to use it as they want. In other words, this testing is unstructured. However, the users are encouraged to give feedback about how the application performs. Usually, you will also monitor how the backend is performing during these tests. Beta testing is more focussed on performance and scalability.

Why Every Company Should Invest in Alpha and Beta Testing? (Industrial use of Alpha Beta Testing)

The software testing industry is known for its jargon — from black-box testing and defects to mutation testing and Gherkin. But two important terms in the software delivery lifecycle are alpha testing and beta testing. Beta testing is the best-known, even for those with a less-technical background.

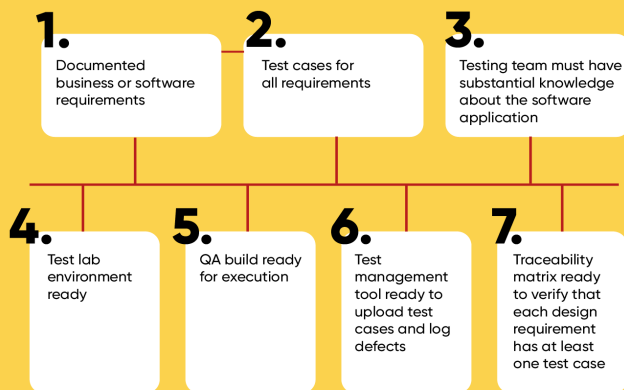
Major brands like **Apple and Google** have made this term popular, by running large-scale beta test programs to give users a sneak peek at their new products. But, for most companies, why take the risk of releasing a product in beta, revealing the latest features to their direct competition? The answer is because you need to be sure you are releasing software that is actually useful and usable.

The Need for Acceptance Testing→

- In most stages of the software testing cycle, QA test engineers perform functional tests to eliminate bugs and defects, ready for release. This ensures the software doesn't crash and the UI behaves as expected. But, in order to verify the software delivers a positive experience, developers turn to two types of acceptance testing: alpha and beta testing.
- For some companies, it doesn't stop there. **Gmail**, for example, was in beta for five years. This approach of long-running betas that are available to anyone allows companies to avoid potential litigation or negative press that may arise from any bugs. Plus it gives companies the right to change or update the software whenever necessary.
- Together, alpha and beta testing ensure software is stable, works at scale, and performs as designed and as users expected. Companies that avoid acceptance testing risk delivering negative customer experiences. But if companies invest in intelligent test tools that leverage machine learning to assist in the creation, running, and analysis of their tests, they can accelerate their product roadmap, giving them a competitive edge and providing the ultimate customer experience.

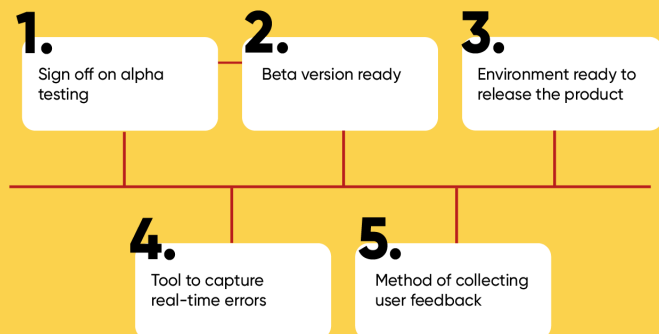
CRITERIA TO START

ALPHA TESTING?



CRITERIA TO START

BETA TESTING?



An Industrial Case Study on Speeding Up User Acceptance Testing by Mining Execution Logs

- Software reliability is defined as the probability of failure-free operation for a period of time, under certain conditions. To determine whether the reliability of an application satisfies the reliability requirements, User Acceptance Testing is performed at deployment sites.
- To support the wide variation in configurations and usage patterns, User Acceptance Testing has become a crucial step in large deployments of mission-critical applications. However, verifying the long-term reliability of an application requires lengthy on-site engagements and dedicated

use of costly lab setups. In this paper, we propose a technique to reduce the time and cost needed for User Acceptance Testing.

- We use a repository of execution logs from related deployments and prior tests of the application to mine reliability estimates. We then customize these estimates by mining logs generated from a limited-time User Acceptance Test (i.e., one day of testing) instead of from traditionally longer tests (e.g., one week of testing).
- Deployers of applications can use such customized estimates to determine whether an application satisfies their reliability requirements. Through a case study on a large-scale enterprise application, we show that our reliability estimate lies within 4% of the reliability estimate derived from the longer User Acceptance Tests.

Beta testing of a Mobile Application: A case study (Introduction)

- Beta testing is the last stage of testing, and normally involves sending the product for beta testing and real-world exposure outside the company. Beta testing is often preceded by a round of testing called alpha testing. Beta testing can be considered a form of external user acceptance testing.
- Software in the beta phase will generally have many more bugs in it than completed applications, as well as speed and performance issues that may still cause crashes or data loss. Beta testing is the first opportunity to get real feedback from target customers.
- The launch of a mobile application is especially crucial because it is the single biggest opportunity to get an application discovered in the mobile markets. In this article, the beta testing of mobile applications is presented. Our aim is to identify the optimal number of testers, who can reveal the majority of errors and mistakes during beta testing.
- The findings were obtained through the case study research method. Copyright © by the paper's authors. Copying permitted only for private and academic purposes.

Complete Case Study -Alpha Beta and Acceptance Testing of Mobile Applications.

The focus of this study is the beta testing of mobile applications. Beta testing can be considered to be a form of external user acceptance testing. Software in the beta phase will generally have many more bugs in it than completed applications, as well as speed and performance issues that may cause crashes or data loss. Beta testing is the first opportunity to get real feedback from target customers, because the beta testing team is independent from the development team. Pre-beta testing should be noted too. **Pre-beta** testing is the phase between the alpha and beta testing phase, although there is no exact definition of this term.

Case study research is conducted by iteration over a set of phases. In the design phase, the objectives are decided on and the case is defined. Data collection is first planned with respect to data collection technique and data sources. Methods for data collection include interviews, observation and the use of archival data. During the analysis phase, insights are both generated and analyzed, e.g. through the coding of evidence from the findings to the original data.

How many testers are needed to reveal the majority of defects of a mobile application during beta testing?

Performing beta testing→

- Testers were performing automated and also manual testing on real devices and in different environments. The main goal of every group of testers was to identify as many defects as possible to ensure a standalone application. To ensure this condition, some essential resources were needed, the most important being human resources. How many people were actually needed?
- The best results come from testing with only 5 users/testers. The number of defects (X) found at testing with n users is:

$$X = N (1 - (1 - L)^n)$$

- where N is the total number of detected problems in the design and L is the proportion of defects discovered while a single user is testing. The typical value of L is 31%, averaged across a large number of projects. Plotting the curve for L = 31% gives the following results: The most striking truth is that zero users give zero insights (Equation 1). As soon as data from a single test user is collected, the insights dramatically rise and almost a third of all defects are found. The difference between zero and even a little bit of data is astounding.
- The second user discovers some of the same things as the first user, so there is some overlap. People are definitely different, so there will also be something new that the second user does that was not observed by the first user.
- In order to facilitate data analysis, we will concentrate on the last week before the last existing version of our mobile application was published. The discovered defects were classified according to whether they were a contextual or technical problem. All testers together found 39 defects (29 errors, 10 deficiencies and inconsistencies). According to the classification, 62% of all faults were technical problems while the rest were contextual. Duplicates were not included. To validate our theory, we first focused on the first 5 testers.

10 Cardinal Steps
to Beta Test
Your Mobile App



Table 1: Number of defects of first five testers

	Number of defects	Number of new defects
Tester A	17	17
Tester B	12	5
Tester C	10	3
Tester D	7	2
Tester E	6	1

The rest of the testers from F to J found a minor number of defects (*Table 2*). But more than the number of defects are significant new defects. *Table 2* shows that the rest of the testers from F to J found only one new defect.

Table 2: Number of defects of second five testers

	Number of defects	Number of new defects
Tester F	5	1
Tester G	4	0
Tester H	3	0
Tester I	1	0
Tester J	1	0

Research confirmed numbers and facts from which we can easily answer the research question stated above. Five testers can reveal the majority of defects, while adding an additional tester does not add a crucial performance boost. Although a sixth tester can enhance the ratio of detected defects, this difference is not crucial. With the current project we revealed that the optimal number of testers when performing beta testing is five. With such a number of testers, work efficiency and the optimal consumption of resources can be ensured.

Beta testing in the mobile domain is a relatively undiscovered area. Since mobile applications are different from traditional ones, they require different and specialized new techniques for testing. Beta testing is one of the phases where mobile applications are verified.

CONCLUSION

We can conclude that a group for beta testing is essential in a stand-alone project that is waiting to be published. Another important perspective is also the need for iterations. The described process needs to be performed all over again, in order to gain effectiveness. This is especially important if the functionalities change or if the customer does not know exactly what to offer to the user. That is why the specifications for software are very important documents for customers, developers and especially for testers, who ensure quality. Our case study can confirm that the optimal number of testers is roughly five. Adding more testers to a group does not ensure additional quality.

Which beta testing tools are best for your software?

Testing tools	Supported Platform	Supported Testing Type	Tool Usability	Integration With 3rd Party Tools	Pricing
Centercode.com	<ul style="list-style-type: none"> - Cloud-based SaaS - Desktop - Tablet - Mobile devices 	<ul style="list-style-type: none"> - Beta Test 	<ul style="list-style-type: none"> - One-time integration - Easy to deploy - Real-time management 	<ul style="list-style-type: none"> - Defect tracker integration: - JIRA - Bugzilla 	<ul style="list-style-type: none"> - Standard Edition: \$499/month - Impact Edition: \$699/month - Enterprise Edition: \$999/month
UserSnap.com	<ul style="list-style-type: none"> - Web (All browsers) - Mobile (iOS, Android) 	<ul style="list-style-type: none"> - Alpha Test - Beta Test 	<ul style="list-style-type: none"> - Easy installation - Easy to use 	<ul style="list-style-type: none"> - Strong integration with 3rd party tools: - Bitbucket - GitHub - JIRA - MS Visual Studio - PivotalTracker - Trello - Kabanize 	<ul style="list-style-type: none"> - Startup: 3 projects/10 users - \$79/month - Company: 10 projects/15 users - \$149/month
UserTesting.com	<ul style="list-style-type: none"> - Desktop - Tablet - Mobile devices 	<ul style="list-style-type: none"> - Beta Test - Usability Test 	<ul style="list-style-type: none"> - Fast feedback - Allows real-time team monitoring - Multi-platform support - Live intercepts of your website visitors 	N/A	<ul style="list-style-type: none"> - Basic: + \$49/video - + \$99/video after first 10 videos - Pro: Call for a quote
TryMyUi.com	<ul style="list-style-type: none"> - Web (All browsers) - Mobile (iOS, Android) 	<ul style="list-style-type: none"> - Beta Test - Usability Test 	<ul style="list-style-type: none"> - Simple to set up - Simple to interpret - Suitable at all stages of development process 	<ul style="list-style-type: none"> - Integrates with usability testing tools: - UX diagnostics - UXCrowd 	<ul style="list-style-type: none"> - Personal: Pay per test: \$35/credit - Team: \$299/month - Agency/Enterprise: Call for a quote

References →(Articles and Research Papers)

1. <https://www.govinfo.gov/content/pkg/GOVPUB-C13-af8b002264009fb032292575187b7acd/pdf/GOVPUB-C13-af8b002264009fb032292575187b7acd.pdf>
2. <http://ceur-ws.org/Vol-1053/sqamia2013paper4.pdf>
3. <http://www.onestoptesting.com/testing-interview-questions/manual-testing/details/-what-is-the-difference-between-alpha-beta-and-acceptance-testing-574.asp>
4. <https://www.devopsdigest.com/why-every-company-should-invest-in-alpha-and-beta-testing>
5. <https://www.govinfo.gov/content/pkg/GOVPUB-C13-af8b002264009fb032292575187b7acd/pdf/GOVPUB-C13-af8b002264009fb032292575187b7acd.pdf>

