Leveraging the Power of the Crowd for Software Testing

Niklas Leicht, Ivo Blohm, and Jan Marco Leimeister,
University of St. Gallen

// To profit from crowdtesting, companies can use three approaches: engage an external crowd of Internet users, engage their employees, or engage their customers. Researchers' experiences with these approaches have led to guidelines for establishing crowdtesting. //

TODAY, MANY IT DEPARTMENTS face an increasingly dynamic environment, shorter product lifecycles, and cost pressure. The rapid development of new IT-enabled business models, a fast-growing hardware market, and that market's segmentation—smartphones, tablets, wearable technologies, or the Internet of Things—are making software testing increasingly complex. Given that increased complexity, traditional manual testing is becoming less applicable—both economically and practicably.[1]

Runkeeper, a company offering a fitness app used by more than 45 million runners, faced just such a situation. Because Runkeeper is a small organization with few personnel, ensuring compatibility across devices, proper localization, and functionality seemed almost impossible. (For example, the company had to take into account several generations of iPhones with various iOS versions; a plethora of Android phones with different hardware setups, screen sizes, and customized Android versions; and smart watches and other fitness equipment communicating with the app. Also, the app is available in more than 10 languages.) Functionality testing made things even more complicated: to test a running app, you need runners. They need to go for a run and ensure that the app works in different locations with varying conditions such as signal strength.

Seeking help in this matter, Runkeeper used the services of Applause (www.applause.com).[2] Applause had acquired a crowd of more than 200,000 skilled testing experts across the world. Out of this group, Runkeeper selected trained testers (on the basis of testing skill, language, device, and so on), who went for a run, put the app to the acid test on their own devices, and ultimately helped make Runkeeper one of the most popular fitness apps.

Employing on-demand testers and testing, Runkeeper (together with Applause) used the principle of crowdsourcing, which has become popular in software engineering.[3,4] In crowdsourcing, groups of individuals follow a flexible open call and self-select to undertake tasks proposed by other individuals or organizations.[5] In crowdsourced software testing, or *crowdtesting*, diverse pools of people test software in real environments using their own devices.[6]

For years, we researched both companies that adopted crowdtesting and intermediary companies that enabled crowdtesting. We determined that there are three valuable crowdtesting approaches:

- companies use an intermediary's on-demand services to have an anonymous, external crowd perform testing (for example, for testing a public customer application),

 0740-7459/17/$33.00 © 2017 IEEE

TABLE 1

Crowdtesting application areas.

Testing type	Crowd members	Examples*
Functional and verification	Test experts and trained crowdtesters	1, 6, 9, 10
Nonfunctional	Test experts	10 (performance testing)
Validation (user acceptance)	Users	6, 11, 12
Usability	Users and the target audience	13, 14

** The numbers refer to the reference list at the end of the article.*

- companies have a crowd of their own employees perform testing (for example, for testing an internal application), and
- companies use a branded (proprietary) platform for testing by a crowd of their customers (for example, for testing and codeveloping new features of an application).

Here, we present case studies illustrating each approach and offer guidelines for establishing crowdtesting in companies.

What Is Crowdtesting?

Crowdtesting is a dynamic testing scenario in which a crowd is concerned mostly with output from given specific inputs because they don't know or see the source code. This form of black-box testing covers both functional aspects and nonfunctional aspects such as performance, reliability, and security.

Currently, crowdtesting usually involves verification testing and validation (specifically, user acceptance) testing. Verification testing aims to eliminate defects that cause error states in the software (functional black-box testing). In validation testing, the user runs a test to determine whether the system meets his or her needs.[7] Another important part of software testing deals with usability—"the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use"[8]—which is also often part of user tests. Table 1 sums up crowdtesting application areas.

For a comparison of crowdtesting and beta testing, see the sidebar.

The Case Studies

In each study, we conducted interviews with company managers and testers (for details, see the Web Extra at extras.computer.org/extra/mso2017020062s1.pdf). Table 2 describes the case details; we changed the company names to ensure privacy.

Alpha Corp.—Using an External Crowd

Alpha Corp. is a large industrial enterprise with subsidiaries worldwide. It mainly sells plumbing hardware. Because the company performs relatively little software development and testing, it occasionally experienced development situations exceeding its peak capacity owing to multiple software releases and staffing limitations.

So, Alpha Corp. used crowdtesting for its public mobile app, from which customers can obtain information about Alpha Corp. products and order spare parts. The company focused only on functional testing. The main goal was to guarantee functionality in separate versions of the app for 12 countries, including eight languages. The testing occurred from Friday evening to Monday morning. So, the weekend could serve to take pressure off the testing department.

A diverse crowd. The company selected a diverse crowd: 22 testers who covered all 12 countries and a range of device and OS combinations. Thus, the company could conduct a broad compatibility test reaching beyond its boundaries.

Comparing in-house and crowdsourced results. The testers had no expertise in Alpha Corp.'s business and products. Because of the intermediary's bug-hunt reward scheme, most bugs were submitted during the first 36 hours after the test started. To assess the testing quality, Alpha Corp. had its in-house testers test the app so that it could compare the results.

The crowd submitted 49 bugs; Alpha Corp. accepted 19 of them. Six of the 19 bugs were previously unknown to Alpha Corp. and eventually fixed. The company already knew about seven of the other 19, and the last six weren't further

WHY CROWDTESTING ISN'T BETA TESTING

The International Software Testing Qualifications Board defines beta testing as

operational testing by potential and/or existing users/customers … to determine whether or not a component or system satisfies the user/customer needs and fits within the business processes. Beta testing is often employed as a form of external acceptance testing for commercial off-the-shelf software in order to acquire feedback from the market.[1]

Accordingly, crowdtesting (in which a diverse pool of people test software in real environments using their own devices) and beta testing both use external resources to test software under real-world conditions. However, there are four substantial differences.

INCREASED SCOPE

Crowdtesting's strength is its applicability early during development. With crowdtesting, you can acquire testers without addressing the general public and can, for instance, evaluate software mock-ups or designs, perform regression testing, perform even nonfunctional testing such as performance testing, and conduct verification and quality assurance tests right before the release. In contrast, beta testing functions as the final quality gate before software release.

TASK-BASED TESTING

Whereas beta testing usually has a strong explorative focus ("Use the software and report bugs"), crowdtesting tasks are much more specific. Testers are asked to go on a user journey and test certain use cases or even conduct traditional test-case-based testing. So, submissions have less noise—that is, ill-advised or irrelevant feedback.

TRAINING AND INCENTIVES

In beta testing, testers usually don't receive a concrete reward. In crowdtesting, testers frequently receive a monetary reward (often based on the number of bugs they catch). Sometimes, the payment is on a first-come, first-served basis, such that testers have a strong incentive to be the first to report a bug. So, the individual tester is motivated to learn and improve his or her testing skills, which will also lead to faster, better feedback.[2]

TASK AND TESTER MATCHING

In beta testing, the call for participation is rather open. With crowdtesting, companies can select testers from a large pool of testers on the basis of a variety of factors and characteristics (for example, testing experience, language, and devices, but also age, gender, and so on). So, the selection provides an efficient way to match tasks to testers (who aren't necessarily customers or clients), ensuring that only suitable people test the software.

References

1. *Standard Glossary of Terms Used in Software Testing*, ver. 3.1, Int'l Software Testing Qualifications Board; www.istqb.org/downloads/category/20-istqb-glossary.html.
2. S. Zogaj, U. Bretschneider, and J.M. Leimeister, "Managing Crowdsourced Software Testing: A Case Study Based Insight on the Challenges of a Crowdsourcing Intermediary," *J. Business Economics*, vol. 84, no. 3, 2014, pp. 375–405.

considered owing to their low severity. The crowdtesters found six bugs that the in-house testing didn't find. On the flip side, the in-house testing found five bugs that the crowd didn't. However, these results indicate that even expert applications can be functionally tested by a crowd, which performed comparably to the in-house testing team.

Unexpected helpful feedback. Although Alpha Corp. hadn't specifically asked or compensated the crowdtesters for user experience feedback, almost half of them gave such feedback. This was of great additional value to the company, which reprioritized the development roadmap:

> *We knew [about] the usability issue before, but for us it had a medium severity, at most. When we saw that more than half of the crowdtesters submitted that very same usability issue, even though it was out of scope, we knew it was really problematic. The result is that we changed the severity and will fix it. (test manager, Alpha Corp.)*

Beta Enterprises—Using Employees

Beta Enterprises is an insurance company with more than 3,000 employees who all use a company-wide intranet. The company planned to

TABLE 2

Details of the three case studies.

Category	Criteria	Company		
		Alpha Corp.	Beta Enterprises	Gamma Bank
Company details	Industry	Industrial production	Insurance	Banking
	Development paradigm	Agile	Agile	Waterfall
	No. of employees	>10,000	3,000	1,500
	No. of employees in the testing department	<10	<100	<50
Application details	Application type	Public mobile app	Intranet	Mobile-banking app
	Application maturity	Medium	Low	High
	Release to be tested (minor or major)	Minor (only a few bug fixes)	Major (completely new software)	Minor (only a few bug fixes)
Test details	Test focus	Functionality	User experience and user acceptance	User experience and functionality
	No. of test iterations	1	2	1
	Crowd members	External test experts	Employees	Customers
	No. of crowdtesters	22	118	38

replace the intranet with a customized standard software solution. It chose crowdtesting for user acceptance testing and as an additional quality gate before releasing the software. Because the test manager was interested in user feedback, the crowd comprised company employees.

Through crowdtesting, the project team hoped to increase the new solution's acceptance by integrating the users into the development process and giving them a voice to provide feedback and thus support IT change management. To coordinate and manage the crowdtesting, the company licensed an intermediary's IT platform as software as a service.

Involving the employees. The project broadcasted a call for participation on its intranet's news page. The call addressed mainly the intrinsic motivation for participating. The overall claim was, "Let's team up and create a better intranet for all of us."

The response was impressive—almost eight percent of the personnel registered to participate, even though the only incentives were three gift certificates for a dinner. On the basis of the participants' functions and locations, they were split into groups for the different tests.

Timely feedback and much work. The employees were enthusiastic about the testing but unfortunately weren't always skilled in testing and using such an IT platform. So, they had many questions about these matters. To overcome those problems, Beta Enterprises created a simple, systematic guide for using the platform.

More than two-thirds of the participants' reports were suggestions to enhance the site's user experience. Overall, the test manager had to go through more than 200 reports, provide timely feedback to the testers, and decide what to do about the issues at hand.

The test timing's importance. Beta Enterprises conducted the crowdtests in two sprints. For the first sprint, the company created basic use cases (for example, "Edit and complete your profile" or "Use the search function to find a colleague") to ensure that the employees tested all the parts of the software.

In the second sprint, which was the last one before the planned release, the participants tested the intranet in an explorative manner ("Do what you would naturally do in your job"), just like in traditional beta testing. Unfortunately, the

software wasn't entirely ready for release. So, the employees reported a vast number of bugs. Hence, the timing of such crowdtests is crucial not only to reduce noisy submissions but also for another reason:

> *If you release the software too early, you get a lot of submissions, but more importantly it can damage your reputation because people are worried about the quality of the end product. However, if you release the software too late, you have only little chance to implement changes and bug fixes. (test manager, Beta Enterprises)*

Identifying training needs. As we investigated the participants' reports, it became evident that the new solution had some problems—multiple testers had reported several issues. Beta Enterprises used these reports to identify training needs for the employees:

> *We noticed that people struggled with the profile section of the page. Because it is standard software, we are not able to change this, so we have to train people. Every employee got an accompanying document with the release. (project manager, Beta Enterprises)*

Gamma Bank—Using Customers

Gamma Bank is a retail bank that employs approximately 1,500 people. Because the bank is rather small, most of its software products, including its mobile-banking app, are standard commercial solutions.

Gamma Bank is aware of its customer applications' high importance. Customers expect them to be flawless and available 24/7 on any given device. This is noticeably causing more and more trouble for the testing department owing to the increasing demand for compatibility testing. Furthermore, the management intends to push the company "closer to its customers," resulting in customer-centered software and products. So, the bank set up a company-branded crowdtesting platform to build and manage a customer crowd.

Overcoming test data and infrastructure issues. Legal regulations and security policies impose high burdens on banks' testing departments. Gamma Bank employs a mature test infrastructure that copes with these requirements. However, its in-house testing relies partly on test data from customers (for example, real-world bank accounts), which the bank can't make available to the crowd.

So, Gamma Bank asks customers to write test and experience reports after doing their usual banking business. The bank deploys the market-ready app as a "preproduction" version—customers log on with their real accounts and use the app naturally in the production environment. The test reports are then created on a crowdtesting platform that's hosted on the premises. Because this approach ensures that the test data remains in the bank, Gamma Bank effectively has overcome the strict security requirements of the highly regulated banking sector.

Obtaining real feedback. At first sight, this testing might look like traditional beta testing. However, Gamma Bank can now conduct different tests (design validation, functionality, and so on) with specific tasks (for example, use-case-based testing) with a highly specifiable and selectable customer crowd without having to address the public. Only the few selected testers will see the test object and related tasks.

For the testing we report here, the crowdtesters covered almost twice as many device-and-OS combinations than Gamma Bank covers internally. Although the release version to be tested included only minor bug fixes, participants still reported a few bugs. However, the project team felt that the greatest benefit was that it obtained customer feedback without exposing the app to the general public.

Many possibilities for the platform. The positive experiences strengthened the bank's course of action. Gamma Bank is expanding crowdtesting to other applications by developing an operational model for crowdtesting. A service center for crowdtesting bundles competencies and offers crowdtesting to other projects or departments in the company. Through this measure, Gamma Bank can generate economies of scale while keeping customers active with recurring or new tasks and offering them the chance to contribute their skills and opinions.

Key Findings

All three companies successfully used crowdtesting for front-end applications with a group of diverse users or a specific user group with many users. All the companies felt that the results were of high quality and effectively complemented in-house testing.

In particular, we learned these lessons:

- Crowdtesting with an external crowd can overcome testing departments' capacity limits when the time pressure is great or resources are scarce.
- Crowdtesting with employees helps companies conduct

TABLE 3

Crowdtesting approaches and main findings.

Approach	Main findings
External crowdtesting	Is especially suited for functional testing
	Can be set up quickly and is useful under high time pressure
	Helps overcome testing departments' capacity limits
Crowdtesting with employees	Promotes high motivation and interest among employees
	Fosters acceptance of new applications as part of IT change management
	Enables easy identification of training needs
Crowdtesting with customers	Obtains feedback from real customers without exposing the software to the general public
	Offers a new channel for interaction with customers
	Is especially suited for usability testing or user experience feedback combined with functional testing

traditional user acceptance testing while letting employees actively participate in the change processes, fostering the new application's acceptance.
- Crowdtesting with customers creates a new customer interaction channel. With this approach, companies can test software applications and systematically involve customers in user-centered design.

Table 3 summarizes the key findings for each approach.

Getting Started with Crowdtesting

On the basis of our experiences with different companies, intermediaries, and crowdtesting approaches, we now provide five guidelines for chief information officers and other organizational leaders on how to implement crowdtesting.

Define Software Quality More Broadly

Expect crowdtesting to provide many benefits. Today, software quality is a matter of not only functionality but also the user experience. Crowdtesting can greatly increase software quality, but for companies to leverage its full potential, multiple stakeholders must work together. (For example, the testing department can work on functional issues, while the marketing or product management department can deal with the user experience.)

Define Goals and Choose an Appropriate Approach

Depending on the software and its maturity, companies can apply crowdtesting to target numerous testing goals. Define a clear goal and pick one of the three crowdtesting approaches. For instance, for functional testing, crowdtesting serves as system integration testing or as final quality assurance testing before the release (similarly to beta testing).

Start a Small Pilot Project

Initially, companies should conduct a pilot project within a smaller, noncritical project environment to ensure that the involved stakeholders can focus on crowdtesting to gain as much experience as possible. Promote the topic to the project team so that the people involved are motivated and convinced of crowdtesting's potential. Select the test object carefully. A smaller test object or test scope will prevent the pilot project from losing focus or getting caught up in other areas of conflict, respectively. Whenever possible, simultaneously test the application with both your current in-house testing approach and the crowd, which will let you draw meaningful comparisons.

Scale Up Crowdtesting in the Organization

The first step in scaling up crowdtesting is expanding the pilot application's test scope. At this point, your testing team will already have experience with crowdtesting and can experiment without high setup costs. Aim to develop a crowdtesting process for regular operations for a class of applications (for example, mobile apps). Furthermore, transfer the knowledge to other similar applications (for example,

ABOUT THE AUTHORS

NIKLAS LEICHT is a research associate at the University of St. Gallen's Institute of Information Management and manages several industry-funded projects at the university's Competence Center Crowdsourcing. He's also pursuing a PhD in business innovation at the university. His research interests are digital business, digital transformation, crowdtesting and crowdsourcing, and the Internet economy. Leicht received a master's in management from the University of Erlangen-Nuremberg. Contact him at niklas.leicht@unisg.ch.

IVO BLOHM is an assistant professor of data science and management at the University of St. Gallen's Institute of Information Management, and he heads the university's Competence Center Crowdsourcing. His research interests are data science and big data, crowdsourcing and crowdfunding, and the Internet economy. Blohm received a doctorate in information systems research from Technische Universität München. Contact him at ivo.blohm@unisg.ch.

JAN MARCO LEIMEISTER is a full professor and the director of the University of St. Gallen's Institute of Information Management. He's also a full professor of business information systems and the director of the University of Kassel's Research Center for Information System Design. His research covers digital business, digital transformation, service engineering and service management, crowdsourcing, digital work, collaboration engineering, and IT innovation management. Leimeister received a doctorate in information systems from the University of Hohenheim and a postdoctoral lecture qualification (habilitation) from Technische Universität München. Contact him at janmarco.leimeister@unisg.ch.

expand from a public mobile app to all public front-end applications) to profit from the pilot project without large adjustments.

Establish and Promote an Internal Service

After your company has had experiences with crowdtesting, expand the crowdtesting's scope to the entire company by establishing an internal crowdtesting service to fully exploit crowdtesting's potential. This internal service should offer "crowdtesting as a service" to other project teams (for example, other testing projects) or departments (for example, marketing, product management, or user experience design). By bundling knowledge and competencies, project and test managers don't have to build up crowdtesting expertise from scratch because the service center can act as a consultant or operate in an executive role by managing the crowdtesting.

Crowdtesting offers several possibilities for coping with today's software-testing challenges. Software quality is no longer determined by functional accuracy; user experience, localization, and many other dimensions are becoming even more important. Crowdtesting effectively combines testing in these many dimensions to help create better software. It should be considered a new tool in the testing toolbox that goes way beyond traditional beta testing.

References

1. E. Dolstra, R. Vliegendhart, and J. Pouwelse, "Crowdsourcing GUI Tests," *Proc. 6th Int'l Conf. Software Testing, Verification and Validation* (ICST 13), 2013, pp. 332–341.
2. *Case Study: RunKeeper*, Applause, 2016; go.applause.com/rs/539-CKP-074/images/RunKeeper-Case-Study-Applause.pdf.
3. T.D. LaToza and A. van der Hoek, "Crowdsourcing in Software Engineering: Models, Motivations, and Challenges," *IEEE Software*, vol. 33, no. 1, 2016, pp. 74–80.
4. K.-J. Stol and B. Fitzgerald, "Two's Company, Three's a Crowd: A Case Study of Crowdsourcing Software Development," *Proc. 2014 Int'l Conf. Software Eng.* (ICSE 14), 2014, pp. 187–198.
5. I. Blohm, J.M. Leimeister, and H. Krcmar, "Crowdsourcing: How to Benefit from (Too) Many Great

Ideas," *MIS Q. Executive*, vol. 4, no. 12, 2013, pp. 199–211.
6. N. Leicht et al., "When Is Crowdsourcing Advantageous? The Case of Crowdsourced Software Testing," *Proc. 2016 European Conf. Information Systems* (ECIS 16), 2016; aisel.aisnet.org/ecis2016_rp/60.
7. R.D. Stutzke, *Estimating Software-Intensive Systems: Projects, Products, and Processes*, Pearson Education, 2005.
8. *Ergonomics Requirements for Office Work with Visual Display Terminals (VDTs)*, ISO 9241-11, Int'l Org. for Standardization, 1998.
9. M.V. Mäntylä and J. Itkonen, "More Testers—the Effect of Crowd Size and Time Restriction in Software Testing," *Information and Software Technology*, vol. 55, no. 6, 2013, pp. 986–1003.
10. Z. Chen and B. Luo, "Quasi-crowdsourcing Testing for Educational Projects," *Proc. 2014 Int'l Conf. Software Eng.* (ICSE 14), 2014, pp. 272–275.
11. S. Amini et al., *Towards Scalable Evaluation of Mobile Applications through Crowdsourcing and Automation*, tech. report CMU-CyLab-12-006, CyLab, Carnegie Mellon Univ., 2012.
12. Y.-H. Tung and S.-S. Tseng, "A Novel Approach to Collaborative Testing in a Crowdsourcing Environment," *J. Systems and Software*, vol. 86, no. 8, 2013, pp. 2143–2153.
13. T. Hoßfeld et al., "Best Practices for QoE Crowdtesting: QoE Assessment with Crowdsourcing," *IEEE Trans. Multimedia*, vol. 16, no. 2, 2014, pp. 541–558.
14. D. Liu et al., "Crowdsourcing for Usability Testing," *Proc. Am. Soc. for Information Science and Technology*, vol. 49, no. 1, 2012; onlinelibrary.wiley.com/doi/10.1002/meet.14504901100/pdf.