



RITE+Krug: A Combination of Usability Test Methods for Agile Design

Jennifer (Jen) McGinn

Sr. Principal Usability Engineer
Oracle Corporation
10 Van De Graaff Drive
Burlington, MA
USA
jen.mcginn@oracle.com

Ana Ramírez Chang

Sr. Member of Technical Staff
Oracle Corporation
500 Eldorado Boulevard
Broomfield, CO
USA
ana.ramirez.chang@oracle.com

Abstract

As user experience professionals, we often face objections that there is not enough time or resources to conduct usability testing during development. With the proliferation of Agile methods being used by development teams to compress the software lifecycle, the focus on time becomes even more critical. To meet the challenges of Agile development, we combined aspects of two discount usability methods: the Rapid Iterative Test and Evaluation method (RITE) and the approach to usability testing taken by Steve Krug. In this paper, we describe why we combined the methods and which elements we incorporated from each. The impact of using this new RITE+Krug combination has been remarkable. Test sessions are getting between 12 and 15 observers on a regular basis. The development, design, product management, and usability teams are engaged and actively participating in the usability testing process. Each test takes only a few hours time from the stakeholders' schedules, because the testing and debriefing can be conducted and concluded in one to two days. As a result, we can conduct a RITE+Krug test every two weeks and can get feedback on more aspects of the product than ever before.

Keywords

user research, usability testing, rapid iterative test and evaluation method (RITE), agile development, discount methods

Introduction

In the fall of 2011, a traditional usability test was conducted on a system under development. Ten users were recruited and run through the test protocol. The sessions were sparsely attended by stakeholders, and the report took a few weeks to complete and present. During and after the presentation of the results, the development and project management teams disagreed not only with the issues identified but also with the interpretation of the results. There were questions of transparency and trust by the development side, and there was disbelief from the user experience (UX) side that the development team really wanted feedback on the system. Not an uncommon story among usability practitioners.

To overcome these challenges, we tried two things. First, in the winter of 2011, a practitioner, who didn't take part in the earlier test, conducted another traditional usability test of the system to establish a baseline to see which of the issues from the first test remained and which of the issues had been fixed in the months that had passed. Second, we proposed changes to the user feedback process, which we believed would better align with the Agile design and development practices that the rest of the team was already following. It is this second approach that we describe in the remainder of this paper.

Our purpose for writing this paper is two-fold. First, we have seen little detailed discussion in the literature of successful user research methods in Agile development environments. Second, we wanted to share with other user experience practitioners what we found to have worked well, and continues to work well.

The RITE Method

Central to the RITE method is the notion that as few as one participant can complete a usability test session; problems are identified and fixed, and then another participant completes the same tasks with the updated system (Medlock, Wixon, McGee, & Welsh, 2005; Medlock, Wixon, Terrano, Romero, & Fulton, 2002). After that session, the system under test may be modified again to fix problems observed in that second session, and the team continues to run participants and modify the system until they are satisfied that the biggest usability problems have been identified, been fixed, and that the fixes have been validated (Medlock et al., 2005; Medlock et al., 2002). The RITE method addresses some of the most common reasons that usability issues go unfixed (Medlock et al., 2005; Medlock et al., 2002). Some of these issues we encountered with our own project teams: usability issues are not believed by stakeholder teams, fixing problems takes time and resources, and usability feedback arrives after decisions have been made (Douglass & Hylton, 2010).

While both of us (the authors) embraced the RITE method from the beginning, there were elements not well suited to our Agile development environment. First, if you need to validate the fixes as the method requires, it could take a large number of participants (Medlock et al., 2002; Medlock et al., 2005,). Second, due to the large number of participants and the time in between sessions to make changes to the system, the RITE method can take up to two weeks to complete (Medlock et al., 2005). Third, if you continue testing until fixes have been validated, it's hard to estimate when the testing will end.

The Krug Method

Central to the usability test method developed by Steve Krug (2006, 2010; referred to here as the Krug method) are the notions that some usability testing is better than none and that "testing one user early in the project is better than testing 50 near the end" (Krug, 2006, p.134).

Another tenet of the Krug method is that the testing be quick, so that stakeholders don't have to commit days or weeks of time. The Krug method can be conducted and completed in a single day by using only three or four participants and replacing a formal report with a 1-hour debrief meeting shortly after the last participant has concluded their session (Krug, 2010).

As a result, instead of taking a week or two to produce and deliver a report, the practitioner leads a discussion on the observations in the test and asks stakeholders to document the changes that they'll make. And then the debrief meeting concludes (Krug, 2010). The

practitioner spends no more than 30 minutes documenting the changes that will be made, sends it out, and it's over. Not only can the usability practitioner move on to the next research study, but the design and/or development team can make changes immediately without having to wait weeks for a formal report as can be the case with a traditional study.

Our RITE+Krug Combination

Because the RITE method hadn't been used before with this team, there were challenges and opportunities. The potential challenge would be to educate development and product management stakeholders about what the new method would entail and to get their buy-in for providing the necessary resources to make it work. We were fortunate that this team had no prior experience with the RITE method, so we could adapt it to include elements of the Krug method to better meet their needs and goals as an Agile team.

The next two sections discuss how we gained support for the RITE+Krug combination of methods. Then "The Method" section describes how we adapted, and currently use, the RITE and Krug methods for our usability testing.

Goals

We needed a method that was transparent and engendered trust with our stakeholders because that was a key issue with the first round of traditional usability testing. Other goals and constraints that this method needed to accommodate were to

- work within the time and resource constraints of an Agile development environment, so we could get results in days rather than weeks;
- uncover usability issues with proposed designs before they were implemented by development;
- allow us to make changes to the designs between participants;
- allow us to make changes to designs after usability testing had concluded, but before development had begun coding; and
- get a high level of buy-in from stakeholders for the changes that needed to be made.

Stakeholder Buy-In

The RITE method requires that stakeholders must observe the test sessions and participate in the discussions about what to fix and how to fix it (Medlock et al., 2005; Medlock et al., 2002). Krug *suggests* that stakeholders should observe at least one session to be allowed to attend the debrief meeting (Krug, 2010). We went a bit further and said that we would *require* attendance by key stakeholders at each session as well as at the debrief meeting. Because we were asking for management to commit the time of key people from product management, development, and design, our stakeholders had to approve and agree to the level of participation we were requiring.

We held a meeting with our stakeholders, including the VP of development, the design manager, and the product manager, to propose the new combination of methods. To be honest, we just called it RITE testing so we could focus on getting their buy-in and not get stuck debating the details of the method. Our goals were to get these stakeholders' approval to move forward with the new "method" and to obtain a list of individuals who would be invited to each usability test session and debrief meeting. The conversation went surprisingly well. We described the method (see "The Method" section) and what we needed from the team. In the meeting, we proposed that we would give them as much or as little control as they wanted with task development, script review, recruitment requirements, and solution definition. In return, we needed key stakeholders to attend all sessions. If they couldn't attend, they would need to send a proxy with decision-making authority. One way we got buy-in on the process was by noting that their presence in the sessions also gave them the ability to ask the participants questions. The VP of development approved the proposal, and the development and product management representatives were identified by the stakeholders.

The Method

Our RITE+Krug combination contains elements of both RITE and Krug methods, with a few adaptations of our own. For example, neither RITE nor Krug require testing in a lab environment. While our combination of RITE+Krug doesn't either, we conduct our sessions in a traditional usability test lab, we record the sessions, and we use a think-aloud protocol as participants complete the tasks.

Like the Krug method, we began by testing three to four participants in a single day and held the debrief session an hour or two after the final participant. Both the RITE and Krug methods rely on research that small numbers of participants can uncover the most common problems for a given set of tasks, participants, and evaluators (Krug, 2006; Krug, 2010; Medlock et al., 2005; Medlock et al., 2002).

Like the RITE method, we made changes to the designs between participants if we found a usability problem. Changing the design after only one participant can be risky because the one participant could be an outlier; however, only design issues that the team thinks are not unique to the one participant are changed. Also like the RITE method, we require key stakeholders to attend each session (Medlock et al., 2005; Medlock et al., 2002). Any other observers are welcome, but we need at least one person from each department—development, product management, and design—to observe in real time.

Like the Krug method, rather than spending a week or two creating a long written report, the findings take more lightweight forms. As soon as possible after the final session (usually within 30 minutes), the user researcher emails a list of observations to the stakeholders. Everyone has the opportunity to ensure that any additional input or feedback of theirs is incorporated before the debrief meeting, where we go over the observations and determine what actions to take as a result (see the "Debrief Meeting and Results" section). A smaller set of people are invited to the debrief meeting than to the sessions, but we require that the attendees observe all of the test sessions.

Session Details

For each participant session, we set up a conference call and a web conference, and then send out a meeting invitation. Then after the meeting we post a link to a recording of each participant's session. While not a requirement of our method, a huge side benefit of using tools to accommodate a team that is distributed around the globe is inclusion. We can recruit participants globally, not just from the geography where the researcher is located.

The use of low-fidelity wireframes in a slide deck and a distributed conference also allows for a wide audience of stakeholders and team members to observe the sessions. We only require three stakeholders to observe the session, but word has spread and we've recently had more than a dozen observers in each of the test sessions—something we've never achieved in local tests. We believe the large number of observers increases the chance that a recommended change will be agreed upon and implemented.

Participant Recruiting and Scheduling

While Krug (2006) promotes the notion of recruiting loosely and grading on a curve, we do not. We use the same rigor in recruiting participants for the RITE+Krug tests as we do for any other usability activity. The participants must meet the recruiting criteria for experience, job role, and product usage to participate in the tests. However, to increase recruiting efficiency, we do allow participants to test another part of the product in a subsequent RITE+Krug test if the new part of the system was not included in their first session and if that earlier interaction doesn't create a bias for what we are testing in the subsequent session. As a result, we do not use team members to emulate users like some Agile teams (Sy, 2008).

Because our team is distributed across time zones, we schedule sessions to try to accommodate those differences. Inspired by Krug, we initially scheduled three 1-hour sessions with 30 minutes in between and the debrief meeting 1 hour after the third participant. That was convenient, but produced two problems. First, 30 minutes between sessions was often not enough time for the designer to create a solution for a problem and update the prototype. Second, if one participant had to reschedule, then we needed to run the session the following day and move the debrief meeting, which wreaked havoc with stakeholder schedules. As a result, we tend to recruit four participants (Krug, 2006; Krug, 2010; Nielsen, 2000). Three

participants complete the test on day one, the fourth participant completes the test on day two, and then we hold the debrief meeting on day two with the stakeholders. This two-day format allows us to schedule at least an hour between participants to make design changes and to avoid rescheduling the debrief meeting if one participant can't make the session. If we lose one participant, we still have our minimum of three or have time to reschedule a participant from the first day to the second.

Prototypes

We use both working code as well as low-fidelity drawings in our studies, depending on the state of the feature we want to study. The Agile methodology recommends short sprints where a feature is implemented over a series of sprints, depending on where it is added to the backlog (Schwaber & Beedle, 2002). If a feature has been implemented and refinement of the feature is coming up in the backlog, we test the current implementation of the feature to inform the refinement step.

When the team is working on a new feature, we test low-fidelity drawings in PowerPoint, essentially paper prototypes as the RITE method advocates. PowerPoint is great for testing low-fidelity or high-fidelity designs, it allows us to include navigational hot-spots, and it works far better than paper over a web conference. It's also easy to change between test sessions and requires no development skills whatsoever, so no work is required by the development team to make changes to the designs.

Debrief Meeting and Results

The debrief meeting happens as soon as possible after the final session concludes—usually an hour later. At the meeting, we walk through the observations and discuss changes that need to be made. After the meeting, the user researcher updates the wiki page for the test with the list of observations and agreed-upon actions that will be taken, and then sends out a link to the rest of the extended team.

Because a RITE+Krug test takes less recruiting effort and only two days to conduct and report, we can run several RITE+Krug tests in the time it normally takes to run a single, traditional test. As a result, we can test more new features earlier in the design cycle and can make more changes faster, because the code is either scheduled to be changed or hasn't been written yet. Then later, once the features have been fully developed, we still have time to run a full baseline test so we can validate fixes.

How This Works with Our Agile Development Process

Our development, design, and product management teams use a modified Agile development methodology (Schwaber & Beedle, 2002). We have 4-month iterations made up of six 3-week sprints. Four of the sprints are designated for feature implementation and two are designated for bug fixes and stability. Prior to using the RITE+Krug combination, the user research process and results had been divorced from the Agile processes, which resulted in the findings coming too late to be acted on. Because of this issue, we integrated the user research with the rest of the process, as illustrated in Table 1.

The team consists of developers, product managers, user experience designers, visual designers, quality assurance engineers, and a user researcher. The user experience designers and product managers work closely with the development team during the feature sprints—answering questions, giving feedback on progress, and fine tuning the feature as it is implemented. The bug fix sprints give the developers time to focus on product stability.

Meanwhile the product managers, user experience designers, visual designers, and the user researcher work on preparing the small set of features that will be implemented in the next iteration (see Table 1). This work includes feature selection, design, user testing, and redesigns. The whole team (including developers) gives

feedback on the feature specification and design before it is ready to be implemented. Like others (McInerney, 2005; Nielsen, 2008; Sy, 2007), our design team stays an iteration ahead of the development team. Like Patton (2008, August) recommends, we iterate the UI before it ever reaches development, thereby turning what is traditionally a validation process into a design process (2008, May).

Table 1. Schedule of Design and Development Activities.

	Product Management and Design Teams	Development team
	<i>Iteration N</i>	<i>Iteration N-1</i>
Sprint 1		Feature development
Sprint 2	Feature selection	Feature development
Sprint 3	Specifications	Feature development
Sprint 4	Wireframes	Feature development
Sprint 5	Create prototypes for testing, observe RITE+Krug study & update wireframes	Bug fix
Sprint 6	Get visual designs & deliver materials to development	Bug fix
	<i>Repeat for Iteration N+1</i>	<i>Repeat for Iteration N</i>

Conclusions

The weaknesses of the RITE+Krug combination are similar to the weaknesses of the methods from which it derives. There are only a few participants, and the changes to the designs will need to be validated in a subsequent test.

But the strengths are numerous. Our RITE+Krug combination has produced better acceptance for usability test results than any other user research method we've ever used. We believe that central to that success is the participation in all of the sessions by key stakeholders from design, development, and product management.

RITE+Krug allows for testing, debriefing, and making initial changes to be completed in fewer than two days. This method requires only a few hours of time from stakeholders and allows us to conduct tests every two to three weeks, which aligns nicely with an Agile development process. That efficiency comes as a result of recruiting and running small numbers of participants, employing light-weight reporting mechanisms, and involving stakeholders. As a result, we can get feedback on, and improve, more aspects of the product than before, while still working within the time constraints of our Agile development environment.

While the RITE+Krug combination requires the buy-in of management and our stakeholders to give up time to attend the sessions, we come together as a group of equals in the debrief meeting. We discuss what we all saw in the sessions, sometimes debating what the next step might be, but taking those steps together as an integrated team. As a result, neither the designer nor the user researcher owns the research but shares that experience. So as the developers and product managers move forward, they are the voice of user experience. They are the advocates of what they saw, of what worked well, and what needs to be changed.

It's this last point that we think is the greatest value of RITE+Krug—we haven't talked about the number of problems found nor the number of bugs fixed because these are not our measurements of success. Instead, the attendance of the stakeholders, their enthusiasm, their commitment to make changes as a result of the RITE+Krug sessions, and their willingness to seek out user experience in future iterations and projects are the key rewards. We believe those impacts are more valuable than counting problems or bugs.

Tips for Usability Practitioners

We have anecdotal evidence that what we've done isn't that different from what many other Agile teams are doing; however, again based on those conversations, we are achieving better results, so here are the bits that we recommend you work into your process:

- Hold a kick-off meeting with the key stakeholders to describe the RITE+Krug process and to get their buy-in before implementing this process. In our case, we included the VP of development who had the authority to ask his team members to attend the sessions.
- Set expectations with the team in the kick-off meeting that the testing will be iterative and will be conducted on prototypes or parts of the system slated for redesign.
- Get agreement that at least one stakeholder from design, development, and product management will attend every usability test session and the debrief session. This person needs to be the individual who is responsible for the product, who will own decisions that come out of the debrief meeting, and who has the authority to perform or delegate the tasks that are agreed to.
- Keep a running list of observations that you amend after each participant so you can send out your observations and recommendations within 30 minutes of the conclusion of the last participant.
- Ask attendees of the stakeholder meeting to review your list of observations prior to the meeting and to reply to the group if they have additions or corrections.
- Keep the debrief meeting as short as possible—no more than 60 minutes.
- Walk through the observations in the debrief meeting, and ask what action should be taken as a result. Sometimes the answer isn't obvious. What's critical is that everyone in the meeting leaves with a common view of which problems need to be solved.
- Document the observations and changes immediately after the meeting—even if it's a list of bullet points on a team wiki—and send the link to everyone who was in the debrief meeting.

Acknowledgements

We thank all of our reviewers who provided valuable feedback prior to submission, including Steve Krug, Joe Dumas, Diana DeMarco Brown, Chauncey Wilson, Robert L. Thomas, Mike Ryan, Erika Webb, Kristyn Greenwood, Andrea Evans, Jayson Webb, and Ilona Posner. We thank our management teams who supported the writing of this paper. And we thank the development and product management teams with whom we get to work.

References

- Douglass, R., & Hylton, K. (2010). Get it RITE. *User Experience*, 9(1), 12-13.
- Krug, S. (2006). *Don't make me think: A common sense approach to the web usability* (2nd ed.). Thousand Oaks, CA, USA: New Riders Publishing.
- Krug, S. (2010). *Rocket surgery made easy: The do-it-yourself guide to finding and fixing usability problems*. Thousand Oaks, CA, USA: New Riders Publishing.
- McInerney, P., & Maurer, F. (2005, November). UCD in Agile projects: Dream team or odd couple? *interactions*, 12(6), 19-23.
- Medlock, M.C., Wixon, D., McGee, M., & Welsh, D. (2005). The rapid iterative test and evaluation method: Better products in less Time. In Bias R., and Mayhew, D.J. (Eds.), *Cost-justifying usability: An update for the Internet age* (489-517). San Francisco, CA, USA: Morgan Kaufmann.
- Medlock, M.C., Wixon, D., Terrano, M., Romero, R., & Fulton, B. (2002). Using the RITE method to improve products: A definition and a case study. Presented at the Usability Professionals Association, Orlando Florida.

- Nielsen, J. (2008, November 17). Agile Development Projects and Usability. In *Jakob Nielsen's Alertbox*. <http://www.useit.com/alertbox/agile-methods.html>.
- Nielsen, J. (2000, March 19). Why you only need to test with 5 users. *NN/g Nielsen Norman Group; Jakob Nielsen's Alertbox*. Retrieved from <http://www.useit.com/alertbox/20000319.html>.
- Patton, J. (2008, May). Getting software RITE. *IEEE Software* 25(3), 20-21.
- Patton, J. (2008, August 5). 12 best practices for UX in an Agile environment – Part2. *User Interface Engineering*. Retrieved from http://www.uie.com/articles/best_practices_part2/.
- Schwaber, K., & Beedle, M. (2002). *Agile software development with Scrum*. Upper Saddle River, NJ, USA: Prentice Hall PTR.
- Sy, D. (May 2007). Adapting usability investigations for Agile user-centered design. *Journal of Usability Studies* 2(3), 112-132.

About the Authors



Jen McGinn

Ms. McGinn is a senior principal usability engineer at Oracle, in Middleware User Experience, where she is encouraged to try new things. She is fascinated by people and feels lucky to work in a role where she can help align the functions of software with peoples' needs and expectations.



Ana Ramírez Chang

Dr. Ramírez Chang is a senior user experience designer at Oracle on the Oracle Social Network team and has a PhD from UC Berkeley. She enjoys working closely with a small team, iterating on a design, and seeing it come to life when users successfully complete a task.