

# Human Centric User Acceptance Testing

Rebecca Long

rebecca.long1@engie.com

## Abstract

Software is built by people to solve problems and make lives easier for other people. As quality professionals, we understand the importance of ensuring our software is properly and fully tested before getting shipped. Often our testing efforts require a User Acceptance Test (UAT) step to get final sign-off before a release. This step, unfortunately, can be messy due to communication barriers that may exist between the engineering team and the users which can lead to misses, frustration and other strong emotions for all involved. Overcoming these barriers requires strong up-front organization, clear expectations, and keeping the human aspect of this testing step in focus. I will share with you some best practices to keep our UAT sessions human centric, setting you up for success with better communication and greater trust between teams.

## Biography

Rebecca Long is the Quality Assurance Manager at Engie Impact in Spokane, Washington. She has 15 years' experience in software engineering focused on quality assurance and DevOps and holds undergraduate and master's degrees in computer science. Rebecca has been a leader in the local tech community for most of the last decade running the QA user group SpoQuality and in 2018 launching the non-profit Future Ada to support and advocate for diversity and inclusion in STEAM (science, technology, engineering, art, and mathematics). She lives with two cats, who are the true masters of the household.

*Copyright Rebecca Long 2020*

# 1 Introduction

Software development goes through many phases to get from inception into users' hands. Testing throughout each phase is critical for producing a high-quality software product. Each testing phase has a different focus and different purpose. Many software projects include a User Acceptance Testing (UAT) phase before releasing the software into a production system. UAT is defined as testing *“from the perspective of the users and other stakeholders for whom [the software] has been built or acquired”* with the goal of managing risk, building confidence, assessing business processes are correct, and confirming that the software is ready for release (Hambling and Goethem, 2013).

The UA testers are often not engineers and are instead a representative of your end-user base. This base may be the general public, a specific industry or group, or internal business consumers. These testers should be treated as your customer when it comes to all feedback provided back to the engineering team. This feedback can give insight into problems needing to be fixed and gaps that need to be addressed. Setting up a positive environment where this feedback can be quickly received, logged, tracked and actioned on is important.

## 2 Human Elements of UAT

Given humans are involved in the entire software development process, considering the human-element is equally important to paying attention to technical details when building and testing applications. Unlike technology, humans come with a lot of characteristics like emotions, lives, families, baggage, hopes and dreams plus different communication styles, backgrounds, educations, and experiences. All of which play into the success of everyday interactions as well as activities like UAT. These human-elements should be kept upfront during UAT sessions to avoid pitfalls in communication, minimizing frustrations and maximizing the effectiveness of the process.

### 2.1 Communication and Trust

Trust is a human-element that has a tight relationship with different levels of communication. Stephen Covey diagrams this relationship in his book, *“The 7 Habits of Highly Effective People”* (Figure 1) that the lower the trust the lower the cooperation level (Covey, 1998).

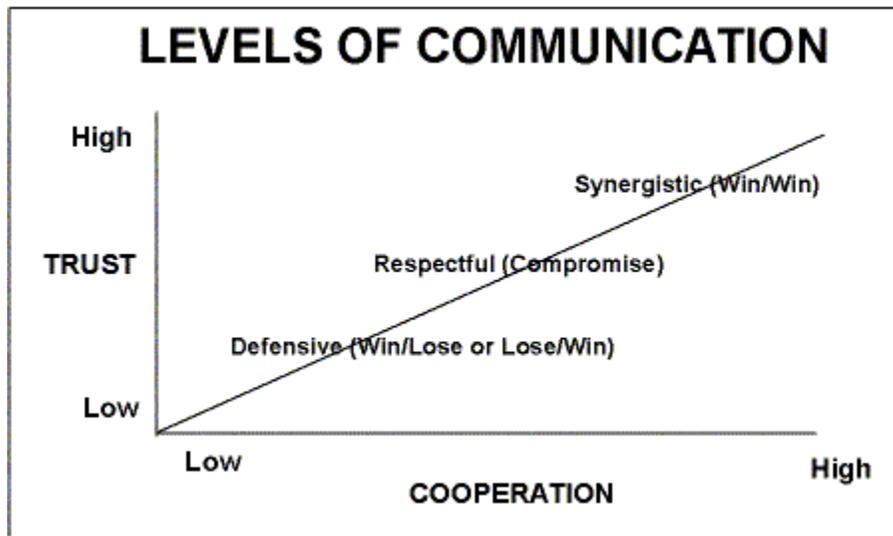


Figure 1: Levels of Communication, Stephen Covey, 7 Habits of Highly Effective People]

It states in “The 7 Habits of Highly Effective People” that *“the lowest level of communication coming out of low-trust situations would be characterized by defensiveness, protectiveness, and often legalistic language”* and leads to only Win/Lose or Lose/Lose situations causing even further reason to be defensive (Covey, 1998). Operating from this place will not create a positive space for an effective UAT session. We want to focus on building trust between members within our engineering team as well as with our UA test team to allow us to reach that high-trust/high-cooperation level of communication.

There are several behaviors that we can work on and be mindful of to build up our “Trust Accounts” (Covey, 2006). Trust Accounts, like bank accounts, can have deposits when behaving in ways that build trust and withdrawals when behaving in ways that break trust. These can look different to different people and often a withdrawal is going to be larger than a deposit. We want to maximize our deposits between our software team and our UAT group as well as minimize withdrawals as much as possible to quickly build up and maintain trust. In “The Speed of Trust” there are 13 behaviors called out as trust building (Covey, 2006):

Straight Talk	Demonstrate Respect
Create Transparency	Right Wrongs
Show Loyalty	Deliver Results
Get Better	Confront Reality
Clarify Expectations	Practice Accountability
Listen First	Keep Commitments
Extend Trust	

Table 1: Trust Building Behaviors, Stephen M. R. Covey, Speed of Trust

We can instill these behaviors into our interactions during UAT by being open and honest with the reality of our software and processes, recognizing gaps and actively working to improve them, owning up to mistakes, stick to commitments, being transparent if things need to shift, and extending trust to others involved in this process by assuming everyone else is doing these same things.

Getting to a high-trust environment will also allow individuals to avoid blame and shame sessions when reviewing feedback provided by UA team. It will also help avoid judgement if confusion arises. High-trust environments allow information to flow freely, mistakes are tolerated as learning opportunities, issues raised can be dealt with directly, and collaboration and innovation thrive (Covey, 2006). This is where we want to operate in and be mindful as we run any User Acceptance Test session.

## 2.2 Inclusive Environment

Inclusivity is another human-element to take into consideration to create a positive space for effective communication during a UAT. This means your UAT spaces and communication channels should all be welcoming so everyone involved feels they belong, and their feedback is valued. There are many ways to build an inclusive space including using inclusive language, avoiding inappropriate language or humor, defining terms that may not be known by all involved, and answering questions without judgement. Inclusion expert Jennifer Brown writes that people feel included *“when our coworkers affirm that (1) we belong, (2) we matter, (3) what we do matters, and (4) “they” hear what we say”* (Brown, 2017). Creating this inclusive space will maximize engagement and successful communication during UAT.

Part of an inclusive environment is having the engineering team take an empathetic leadership role when running a UAT session. Empathy is about being able to step into someone else’s shoes and see and feel what they are experiencing. Merriam-Webster Dictionary defines empathy as *“the action of understanding, being aware of, being sensitive to, and vicariously experiencing the feelings, thoughts, and experience of another of either the past or present without having the feelings, thoughts, and experience fully communicated in an objectively explicit manner”* (Merriam-Webster).

Author and TED Talk speaker, Simon Sinek, is quoted as saying *“My success hinges entirely on the people I work with—the people who enlist themselves to join me in my vision. And it’s my responsibility to see that they’re working at their best capacity”* (Levitt, 2017). This is the mindset the engineering team

should take -- doing everything they can to ensure UA team is able to test and provides feedback at the best of their ability. Having empathy on their experience as testers includes withholding judgement for “mistakes” or “user error”, checking your biases about users (and testers and anyone else) at the door, and not getting frustrated when problems are reported. All this feedback is valid and what your users are likely to experience. We need to have empathy for these experiences, take them seriously and ensure they can provide these experiences back to the engineering team.

## 3 Best Practices

There are some standard best practices that should be used for a successful UAT session while also considering the human-element explained here.

### 3.1 Clear Expectations

The first step is always getting a commitment from leadership on doing a UAT session. Then you can work with your software team to define the purpose, goals and timeline needs for your UAT. Before scheduling anything with UA testers, you should have answers to the following questions:

- What environment will be used for testing?
- Who is responsible for setting up the test environment?
- What feedback are we looking for out of this test session?
- Will there be formal test scenarios? If so, who will create them?
- What are the test data requirements?
- What kind of data set will be used? Production replica or generated data or something else?
- Where will UAT results be documented?
- How will problems be communicated, vetted, logged and prioritized?
- What is the timeline for running this testing session?

Once these questions are answered, communicate them to everyone involved in the UAT. Clearly and transparently define roles and goals for those involved. It’s important to explicitly set the appropriate expectations around the session up front to keep everyone on the same page. People will come into the UAT session with their own implicit expectations which they will use to judge things by. Not clearly setting expectations around process, roles, responsibilities and goals at the start leaves people to their original expectations and *“if they feel like their basic expectations have been violated, the reserve of trust is diminished”* (Covey, 1998).

Creating and maintaining a high-trust environment for these test sessions is important to maximizing their success. As “The 7 Habits of Highly Effective People” calls out, *“when expectations are not clear and shared, people begin to become emotionally involved and simple misunderstandings become compounded, turning into personality clashes and communication breakdowns”* (Covey, 1998). All those communication breakdowns lead to an inefficient, stressful and frustrating UAT experience for all involved thus lowering the value produced by this important testing phase.

### 3.2 Feedback Loops

Feedback loops of different types are important in any testing phase. These loops *“have different costs and response times and return different types of information”* all focused on getting feedback around quality to the engineering team as fast as possible (Cummings-John and Owais, 2019). According to Hubspot:

*“A feedback loop is a process in which the outputs of a system are circled back and used as inputs. In business, this refers to the process of using customer or employee feedback (the outputs of a service or product), to create a better product or workplace”* (Hubspot Blog, 2018).

We often see feedback loops in software development through our continuous integration systems and automated testing processes. Many of our development and test tools can quickly provide automated feedback to the development team to be actioned on as needed in a timely manner. These are highly valuable and necessary for teams to utilize.

Building in intentional human-centric feedback loops will aid in a fast flow of communication back and forth with the development team during the UAT phase. Human-centric feedback loops for UAT should include structured and clear communication channels. Planned meetings such as a kickoff and close out session along with daily touchpoints provide consistent known channels for communicating with the overall UAT team on feedback and get concerns addressed. These meetings can be in a scheduled room, over a conference call, and/or via a designated group chat for all UAT participants to be in for support and discussion. A UAT coordinator should be designated to facilitate all the meetings and formal discussions. Having a coordinator also makes it easy for anyone to reach out for immediate assistance or updates on status.

All feedback loops and communication channels should be inclusive and safe for participants. The human-element must always be considered for fostering trust and collaboration. The UAT coordinator is responsible for ensuring there is no blaming when feedback is reported back to the development team. The UAT coordinator is also responsible for ensuring there is no judgement when working with UA team or when questions arise. Everyone participating should be doing this as well, but the UAT coordinator needs to set the tone and call out bad behavior, keeping everyone focused on the shared goal of producing a high-quality product.

Every team and every UAT is different in what they need and what the preferred feedback loops are. We should be working to continuously improve these processes and learn from what works for our groups and what doesn't. At the end or following UAT, capture lessons learned and actively work to incorporate improvements in future UAT sessions. This is one of the core trust behaviors, "Get Better":

*"Get Better is based on the principles of continuous improvement, learning, and change. It is what the Japanese call kaizen, and it builds enormous trust. ... When people see you as a learning, growing, renewing person—or your organization as a learning, growing, renewing organization—they develop confidence in your ability to succeed in a rapidly changing environment, enabling you to build high-trust relationships and move with incredible speed"* (Covey, 2016).

### 3.3 Documentation

In a similar vein, having clear documentation up-front and throughout the UAT process is critical to keep everyone on the same page and maintain a human-centered focus. Even within an Agile Development environment, creating good documentation upfront on direction and instruction for how UAT is to be run will save you time and frustration. This initial documentation should include answers to the questions you had at the start of UAT: the purpose, goals, roles, responsibilities, scope, assumptions, constraints, environment information, risks, etc.

Take advantage of checklists to keep track of vital setup and teardown procedures for UAT. Checklists *"remind us of the minimum necessary steps and make them explicit. They not only offer the possibility of verification but also instill a kind of discipline of higher performance"* (Gawande, 2009). You can find a variety of UAT checklist examples to help get you started in *User Acceptance Testing: A Step-By-Step Guide* (Hambling, and Goethem, 2013). Create ones for your unique needs and modify it as needed with each run to continuously improve.

The scope of what is under test and what is not, which parts need UAT focus and feedback, need to be documented and provided to the UAT team before starting. This information can be sent to the team to review ahead of time and/or reviewed together at the UAT kickoff. Scope could be items to test from a high-level view or feature focused on what's ready for test. Test scripts of specific test scenario to run should be provided and ready at the start of the kickoff for UA testers to use.

UAT should also produce artifacts (e.g. bug reports) which are clearly documented for easy troubleshooting and prioritization. Templates should be provided for UAT team to use for filling out their reports. UAT testers should work with a QA representative to assist in quickly vetting problems and getting them properly logged.

Avoid making this documentation part of UAT tedious or inconvenient as it can “[discourage] ... recording of found bugs” (Weinberg, 2008). Additionally, it will be important to stay committed to a blameless environment when issues are discovered and reported back to the development team. Be mindful of tone in written reports and how problems are presented to the UAT group. There is “no benefit ... gained by adopting a blaming or patronizing tone” and it only leads to eroding trust (Kaner and Bach, 2001). All issues are opportunities for the quality of the product to improve which is everyone’s shared goal.

At the end of UAT, make sure to clearly and explicitly document the outcome. What passes? What fails? What gaps were discovered? What items are critical to be addressed before getting a release sign-off? Put front and center in the top-level UAT documentation a summary of everything with the overall outcome stated and any release recommendation to go with it.

## 4 Success Template

Considering everything discussed here so far -- creating a human focused and inclusive environment – here is a baseline template to get you started on setting up a successful human-centric UAT.

### 4.1 Preparation Work

1. Understand your testing needs from users
  - a. What questions are you asking from them?
  - b. What feedback are you looking for?
2. Get leadership commitment
3. Create documentation for test session
  - a. Checklists
  - b. Test scripts
  - c. Defect reporting process & template
4. Setup UAT environment for testing
  - a. Confirm correct builds / releases are deployed and configured properly
  - b. Run a build verification test on environment to validate it works
5. Identify & document test data requirements
  - a. What can / should be used?
  - b. What cannot / shouldn’t be used?
  - c. Does any of it need to be generated special for this test session?
  - d. Is data already loaded in the system which can be used?
6. Identify UAT team
  - a. Are they actual users of the system?
  - b. Are they representatives of actual users of the system?
  - c. Do they already have access to the system?
  - d. Do user accounts / access need to be set up?

### 4.2 User Acceptance Testing

1. Kick Off
  - a. Include UAT team & key engineering support resources
  - b. Explain testing plan
  - c. Walk through documentation and test data requirements
  - d. Discuss feedback loops & communication ground rules
  - e. Identify & share facilitator / key engineering point-of-contact
  - f. Provide any known issues to UAT team

- g. Be thankful for everyone's time in helping with the test session
- 2. Testing Session
  - a. Stay in constant contact with UAT team
  - b. Answer questions and address problems quickly when brought up
  - c. Be grateful for issues raised & take note to avoid them in the future
  - d. Report defects in issue tracking system (confirmed or not)
- 3. Daily touchpoint
  - a. Include UAT team & key engineering resources
  - b. No-blame and no-judgement zone
  - c. Discuss status and defect found
  - d. Get a pulse on the emotional impact of the test session on users
  - e. Clear any roadblocks blocking the UAT team
  - f. Express gratitude in meetings and thank everyone in attendance
- 4. Close Out
  - a. Collect any final feedback from UAT team
  - b. Confirm everything was properly documented
  - c. Document lessons learned
  - d. Get sign-off or recommendation from UAT team on go/no-go for release
  - e. Thank everyone for their help

## 5 Conclusion

The human-element is always the most complicated to account for and manage. However, considering human factors when designing a User Acceptance Test session can prove extremely fruitful resulting in minimal frustration and stress and maximum effective communication and collaboration. The template and suggestions in this paper will help get you setup on a path for success. Every team and individual are different so you will need to take that into account, adapting as you go. Just like with Agile Development, working with humans is an iterative process where you want to fail fast, acknowledge and learn from your mistakes, and continuously improve.

## References

- Brown, Jennifer. Inclusion: Diversity, the New Workplace & the Will to Change. Purpose Driven Publishing, 2017.
- Covey, Stephen M. R. The Speed of Trust. New York, NY: Free Press, 2006.
- Covey, Stephen R. The 7 Habits of Highly Effective People. Provo, UT: Franklin Covey, 1998.
- Cummings-John, Ronald, and Owais Peer. Leading Quality: How Great Leaders Deliver High-Quality Software and Accelerate Growth. Great Britain: ROI Press, 2019.
- “Empathy.” Merriam-Webster. Merriam-Webster. Accessed September 7, 2020. <https://www.merriam-webster.com/dictionary/empathy>.
- Forsey, Caroline. “The Definition of Negative and Positive Feedback Loops in 200 Words or Less.” HubSpot Blog, July 6, 2018. <https://blog.hubspot.com/marketing/feedback-loop>.
- Gawande, Atul. The Checklist Manifesto. New York, NY: Picador, 2009.
- Hambling, Brian, and Pauline Van Goethem. User Acceptance Testing: a Step-by-Step Guide. Swindon, UK: BCS, 2013.
- Kaner, Cem, and James Bach. Lessons Learned in Software Testing. New York, NY: Wiley, 2001.
- Levitt, Shelley. “Why the Empathetic Leader Is the Best Leader.” SUCCESS, March 15, 2017. <https://www.success.com/why-the-empathetic-leader-is-the-best-leader/>.
- Weinberg, Gerald M. Perfect Software: and Other Illusions about Testing. New York, NY: Dorset House, 2008.