



# The Essential Guide to Integrating User Experience and Usability Testing in Agile

## Authors:

Dean T. Barker, Sr. Director of User Experience at Optum

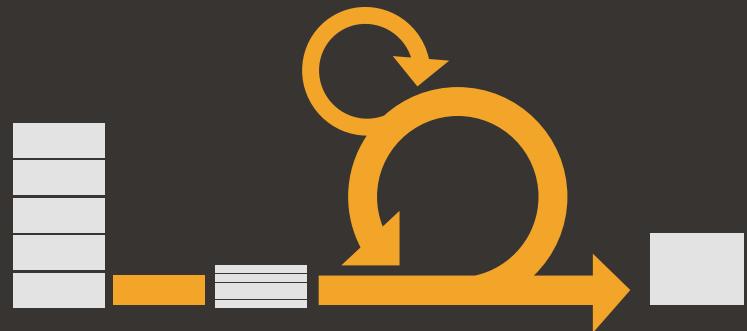
Jon Innes, 'The Agile UX GURU' at UX Innovation LLC

Jeff Sauro, 'The UX Measurement MASTER' at Measuring Usability

Eunice Chang, Senior Principal User Researcher at Autodesk Inc.

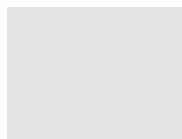
Olivia Williamson, Principal User Experience Designer at Autodesk Inc.

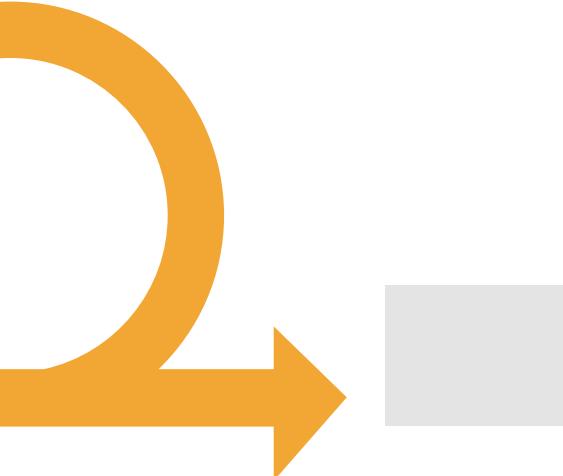
Edited & compiled by Toni Allen, User Research Manager at UserZoom Inc.



# Contents

About this eBook .....	4
About the Authors .....	5
<b>1.0 Intro .....</b>	<b>6</b>
1.1 Evolution of Agile + UX .....	7
<b>2.0 What is Agile?.....</b>	<b>9</b>
2.1 Scrum .....	10
2.2 Lean .....	11
2.2.1 Lean UX .....	12
<b>3.0 Agile development cycles &amp; the challenge of UX .....</b>	<b>14</b>
3.1 Scrum roles – where does UX fit?.....	15
3.2 Scrum processes & UX .....	16
3.3 Integrating UX Design with Scrum .....	17
3.4 Release Planning & User Research .....	19
3.5 SUMMARY: Dean Barker's Seven Keys to Success .....	22
<b>4.0 Dispelling UX myths about usability testing in an Agile Design Process .....</b>	<b>23</b>
4.1 Myth #1: Agile and user testing timelines are incompatible .....	24





4.2 Myth #2: You need working code to test with users.....	24
4.2.1 The 3x3 Way: Test Multiple Concepts Early.....	25
4.2.2 The RITE Way: Refine One Concept Faster .....	26
4.2.3 The A/B Way: Refine Details At End .....	26
4.3 Myth #3: It's hard to scale the usability testing process .....	27
4.4 Myth #4: Automating usability testing is not possible .....	27
<b>5.0 What is Remote Automated User Testing?.....</b>	<b>28</b>
5.1 Tips for testing in Agile.....	31
5.2 Tips for using Remote Automated Testing tools .....	32
5.3 Keep UX principles at the forefront .....	33
<b>6.0 Agile Metrics: An Example from Jeff Sauro .....</b>	<b>34</b>
<b>7.0 CASE STUDY: Formative Usability Testing in Agile: Piloting New Techniques at Autodesk .....</b>	<b>39</b>
7.1 Customer Councils .....	41
7.2 Testing the Design: Are we building the right thing?.....	42
7.3 Testing the Product: Does it work as expected?.....	45
7.4 Best practices based on the AutoCAD Customer Council .....	46
7.5 Summary: Customer Councils – What success looks like .....	48

# About this eBook

This eBook is designed to help you understand **how to integrate UX in an Agile world, and how to master user testing in an Agile design process.** Whether you are a novice or an expert in Agile, it is full of valuable information for everyone who is in UX or has an interest in UX. In addition, you will read a case study how Autodesk Inc. was able to pilot new techniques to integrate formative usability testing into their AutoCAD product development process.

# About the Authors

**Toni Allen** is a User Research Manager and Pre-Sales Consultant at UserZoom. She specializes in remote unmoderated usability testing. She was formerly a Sr. Product Manager at HP where she contributed to many Agile projects and user experience research studies to develop software for HP consumer PCs.

**Dean Barker** leads the UX group for Optum, the commercial software arm of United Health Group. Barker has a long history with Agile design and development and will share many valuable insights and observations based on his knowledge and experience.

**Jon Innes** is known as the 'The Agile UX GURU' and is the founder of UX Innovation LLC. He is a user experience specialist, and an expert in user-centered design and user research. He has held roles as a manager/director of multi-disciplinary UX organizations. He has extensive

experience in building teams in companies new to UX, and helping low performing groups achieve greater success. Jon is an internationally known speaker on the topic of user experience strategy.

**Jeff Sauro** has been dubbed 'The UX Measurement MASTER'. Jeff is the founding principal of MeasuringU (measuringu.com), a company providing statistics and usability consulting to Fortune 1000 companies. He is the author of over 20 journal articles and 5 books including his latest, Customer Analytics for Dummies.

From Autodesk Inc., **Eunice Chang**, Senior Principal User Researcher and **Olivia Williamson**, Principal User Experience Designer will share how they have been successful integrating formative usability testing and collecting on-going user feedback in an Agile environment.

# 1.0 Intro

Why should we think of Agile in the context of UX? Agile is what is happening in the software/web development industry, and this is the time for user experience practitioners to embrace it.

Dean Barker, Sr. Director of User Experience at Optum, believes that **user experience is about a concern for productivity and satisfaction**, whether from the standpoint of a consumer or business user. **Concern about efficient software development is what drove the Agile movement and its inception.** Note the common themes around productivity and efficiency, which is a great place to start when discussing Agile and UX together.

## Agile is about 4 things:

1. Valuing individuals and interactions over processes and tools
2. Creating working software and valuing that over comprehensive documentation
3. Valuing customer collaboration over contract negotiation
4. Responding to change over following a plan

Those four guiding principles have led the Agile movement over the years and the spirit of agility in a software development context. Agile begins with a philosophy. When people say, “We do Agile development” they are not talking about methodology. Agile, per se, is not a software development method. It is a philosophy that goes back to a document called Agile Manifesto (see agilemanifesto.org).

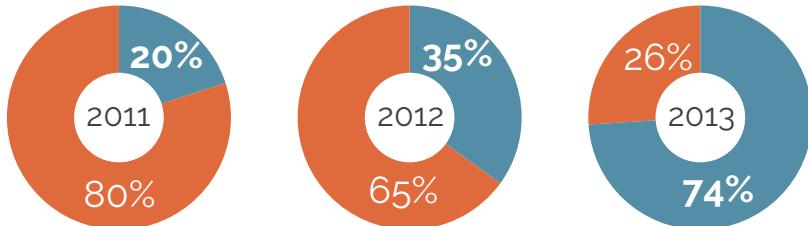
If you are skeptical about usability testing in an Agile environment, Jon Innes, ‘The Agile UX GURU’ at UX Innovation LLC, will help dispel **4 UX myths about usability testing in an Agile Design Process.**

In addition, Jeff Sauro, ‘The UX Measurement MASTER’ at Measuring Usability, will describe **examples of Agile metrics you can capture through remote user testing.**

## 1.1 Evolution of Agile + UX

UX'ers need to learn to navigate through the waters of the Agile world. In 2008, 20% of development projects could be classified as Agile. By 2010 it was 35% and by 2013 it was 74%. You can easily see the trend.

**Percentage of software development projects classified as Agile:**



The good news for UX is that it's now an "Experience" economy. The 2010 Forrester Report, *How To Prepare For The Era Of Experience* describes the evolution of the U.S. economy from the Industrial Evolution which was based on production, to post World War II which was based on distribution. In effect the winners and losers in business were solely based on their ability to produce goods and distribute them. In the distribution era, the winners in efficiency were Wal-Mart from a brick-and-mortar standpoint and Amazon from an online standpoint. But today, with production and distribution normalized, it is an experience-driven economy. Today, the differentiator that business needs is the quality of experience. Most people think of Apple, for example, as the current poster child for the experience economy.

### It's an Experience Economy

#### Production

- Industrial Revolution
- Efficient production of goods
- Produced cheaply

#### Distribution

- Post World War II
- More efficient delivery methods
- Modern shipping methods
- Wal-Mart as classic distribution efficiency
- Amazon as leading edge poster child

#### Experience

- Can't succeed by simply building product
- Or relying on distribution efficiency
- Competitors can do the same
- Now it is product experience that will create value
- And establish power in the marketplace

Forrester Report October 2010  
*How to Prepare For The Era Of Experience*

Whatever value you're providing - intelligence, workflow, connectivity or expertise - if you make software, all of that value flows through the user interface. So, if you are in the software business, you're necessarily in the UX business. This is good news for everyone in UX.

## Software Value Flows Through UX



Customers respond to good UX by being more willing to purchase, being less willing to do business with your competitors, and being more willing to recommend you, which is critical in an era that uses the Net Promoter Score as a key business metric.

# 2.0 What is Agile?

First of all, it's important to acknowledge that in the UX community, there is an amount of miscommunication and misunderstanding about Agile. Companies that adopt a small part of Agile and a small part of UX processes don't necessarily do it all the same or do it well. **Jon Innes spent time studying Agile and was certified as a Scrum product owner because he was interested in learning about why Agile works and doesn't work in some cases. He shares his thoughts and expertise throughout the next few chapters.**

Originally, Agile was oriented around development. This lead to UX'ers wondering "What about us?" Agile is not about holding those daily meetings (it's a big part of it); it's not just about having user stories (a great benefit for UX). It is also about defining tests for code before writing the code itself. A lot of people don't do it, but it is a key point when you scale and make Agile work well. And it is about working in short iterations to create incremental improvements. And that's something that sounds great to UX; you have the opportunity to make incremental improvements to your designs. The most common variations (brands or types of Agile) are Scrum and XP. They all have a mindset that is best described as "lean". When we think about how to integrate UX and testing within software development, we're going to think about how to be lean and fit into the Agile model. As UX becomes more mainstream, it is necessary for UX practitioners to understand Agile and how to integrate UX work in an Agile environment.

## What is Agile?

Agile is a collection of best practices for developing software including:

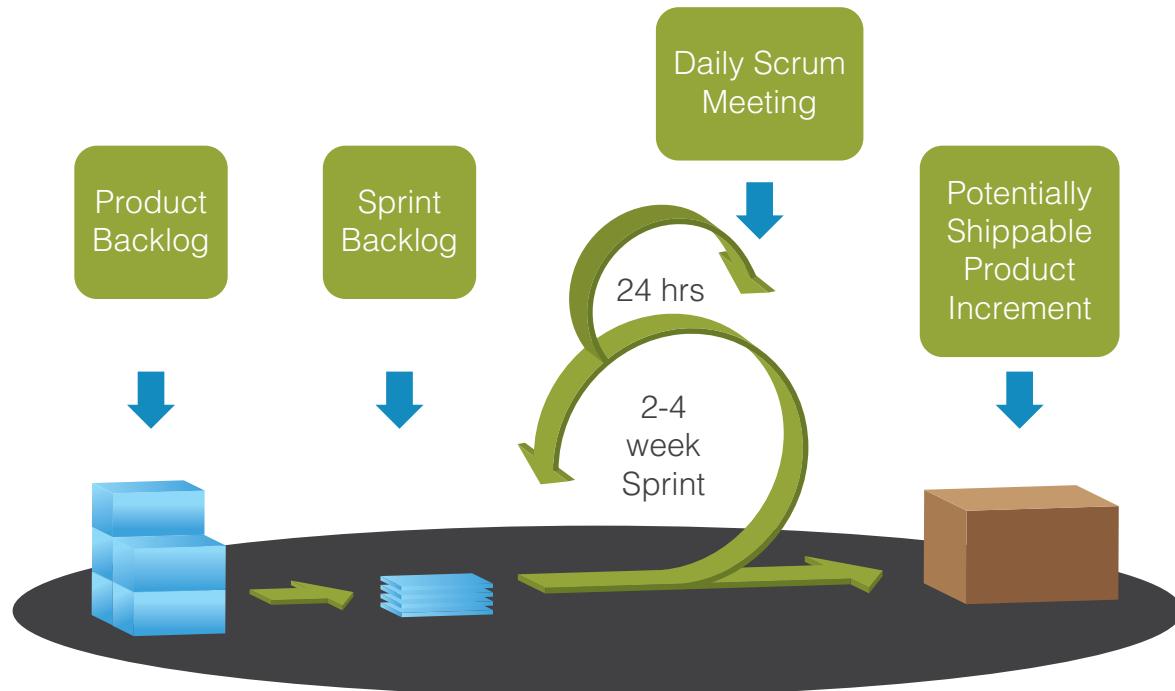
- Holding regular status meetings in a standup format
- Requirements in the form of user stories or use cases
- Defining tests for code before writing the code itself
- Working in short iterations to create incremental improvements

Common variations include Scrum & XP and all have a mindset that is best described as "lean".

## 2.1 Scrum

Let's explore the variation of Agile called Scrum. Scrum is the dominant Agile software development framework that's currently in use in the industry. With Scrum there is a "backlog" which is a collection of requirements in the form of user stories in very short sentence format. These are akin to a to-do list. At any given time, the team pulls user stories off the backlog and starts planning future sprints. The idea is to build a product in 2-4 week increments. Sprints are usually 2-4 weeks in length. Every day the team talks about progress made on the sprint. At the end of the 2-4 weeks, the goal is to have a potentially shippable product increment - a stand-alone product that has been designed, tested and is functional.

### What is Scrum?



## 2.2 Lean

Behind all Agile methods is the concept of “lean”. If you trace the history of this concept, it goes back to initial guides and practices at Toyota for improving manufacturing. Lean is an interesting concept. It is based on the western interpretation of Taiichi Ohno’s set of values and practices for removing three types of inefficiencies:

- 無駄 = muda wasted outputs
- 斑 = mura unnecessary inconsistencies
- 無理 = muri wasted efforts



If you are a user researcher and you’ve worked in the field a long time, you’re probably familiar with all these. A wasted output is the big report you’ve written but nobody reads. The unnecessary inconsistency could be the team not talking to each other about the UI. Perhaps they built things very differently and it didn’t need to be different, and it confused the user. Finally the muri concept - you spent all that time writing the report; stayed up late documenting the inconsistency in a report that doesn’t get read. You experienced a lot of things muri is trying to solve for. The terms “Lean Startup” and “Lean UX” are based on these values and practices to avoid inefficiencies.

## 2.2.1 Lean UX

What is Lean UX and what does it mean to us?

Lean UX is a fairly new concept and many people have questions about it. It is a response to the overall trend of using the Agile method and applying lean principles to user experience work. The key concepts are:

- Breaking deliverables down to fit into Agile timelines so they are “just in time”
- Reducing wasted effort associated with internal deliverables
- Reducing the variability in UX deliverables that create waste
- Collaborating more closely with non-UX team members
- Getting user feedback earlier and on a more regular basis

The overall ideas are to break down deliverables to smaller chunks so the receiving team or person will not wait for a long time, and they can be efficient. Then, aim to reduce or get out of the deliverable business, minimize internal deliverables in particular, and instead emphasize on talking to people more and collaborating with non-UX team members. Try to eliminate the throw-it-over-the-wall mentality. Fortunately, Lean Startups and Lean UX concepts easily lend themselves to getting user feedback earlier and on a more frequent basis. It's all about eliminating waste. Think about those big formal reports that take a lot of time to produce.

**Studies have shown that user research teams frequently find big problems after it's too late.**

Consider the balance of time and effort across various types of testing. For example, A/B tests can easily be run with thousands of users but perhaps to test something very minor. Could other larger problems have higher priority? Retesting repeatedly in a manual way might be automated with QA testing tools. Fixing problems after launch is wasteful, but testing prototypes early and often with remote testing and even automated testing is likely to be more efficient approach.

## What's changed?



Product development cycles are getting faster



Businesses are now more UX savvy



In the US 79% of the population is now online



Remote usability testing solutions are commonplace and are:

- More capable
- Less costly
- Easier to use

For more on Lean UX see:

<http://www.slideshare.net/balancedteam/02-sat-janice-lean-ux-landscape>

<http://joshuaseiden.com/blog/2011/09/what-makes-it-lean/>

<http://uxdesign.smashingmagazine.com/2011/03/07/lean-ux-getting-out-of-the-deliverables-business/>

# 3.0 Agile development cycles & the challenge of UX

The difference between Agile and a waterfall or classical approach is the notion of a development cycle. In this example of an Agile development calendar, on the 3rd of the month sprint planning would start, which is the first event that happens in a sprint. Then there are perhaps 8-12 days for development, ending with a sprint review and retrospective. It's a very compressed schedule, and it creates certain challenges with design.

## Sample Agile Development Cycle

Sun	Mon	Tues	Wed	Thurs	Fri	Sat
	1	2	3	4	5	6
7	8	9	10	11	12	13
	*Story Time	Development				
14	15	16	17	18	19	20
		Sprint Review & Retro / *Prioritize Backlog	Sprint Planning	Development		
21	22	23	24	25	26	27
		*Story Time	Development			... (Repeat)

\* For next Sprint    <http://coachingagile.blogspot.com/2010/05/when-do-we-start-when-should-we-finish.html>

**The Agile framework does not explicitly outline how design happens; it is indeed a framework not a refined process or methodology.** There is no place built in for design or user research, so we have to create it. In addition, only 2 weeks of development and testing should raise a red flag for UX'ers, especially if it involves risks with the UI design. And, it is tough to recruit users and complete a search study in that timeframe. However, keep in mind that sprints are intended to be ongoing, not two weeks then stop. It is a process and processes can be designed, measured and improved. Remember to balance the bigger picture as you move across each individual sprint.

# 3.1 Scrum roles – where does UX fit?

The main process elements of Scrum are described below.

## Scrum Process Elements

### Three Roles

- Product owner
- ScrumMaster
- Team

### Four Ceremonies

- Sprint planning
- Daily scrum
- Sprint reviews
- Sprint retrospectives

### Three Artifacts

- Product backlog
- Sprint backlog
- Product increment

There are three roles in Scrum: Product Owner, Scrum Master or Team Member (anyone that isn't a Product Owner or Scrum Master, such as developers, designers, or QA). The Scrum role of Team Member is not only generic, but also is an out-of-the-box principle where roles are supposed to be swappable, meaning a developer ought to be able to do documentation or testing.

However, UX was probably not considered in this model. A designer usually should not be coding, and a programmer or business analyst generally should not be designing as long as specialized resources are available. And then there are user researchers who also have specific skills and should ideally try to stay agnostic. The challenge is to decide how you will bring UX resources such as "Team Members" in the Scrum process, and broadly incorporate the UX function as well.

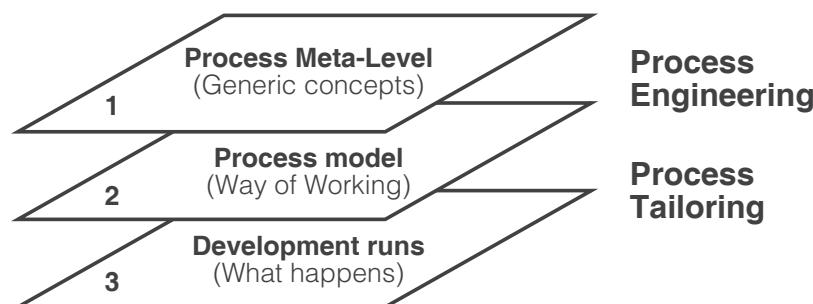
**Dean Barker recommends at least one designer on the Scrum team full-time.** If you do not have a dedicated designer, how will you handle those duties? For other roles, you'll need to plan for full-time or part-time allocation. Also consider if you have experienced UX professionals or if you will use cross-trained specialists such as business analysts. A good rule of thumb is a 1:4 ratio of designers to developers. Also be sure to allocate project budget to user research and usability evaluation, ideally 10% of budget is a good target if possible.

Barker also recommends adding UI developers as another specialization. The UI developer specializes in front-end development and can partner closely with designers and other UX team members to drive to desired user experience outcomes.

## 3.2 Scrum processes & UX

Let's turn now from resources to processes that will help those resources be successful. The key is to look at best practices and process improvements to prescribe the best ways of working for your team.

Here are some general concepts to be aware of, in particular the three levels of Process Engineering. Generic concepts, which like the Scrum model itself, you can't take at face value. You have to refine these processes in some way in order to make a prescriptive process model, or in other words, a way of working that a team agrees to. So for user experience design, you are engineering processes that work specifically for UX. Once processes play out, the actual result is the development run. You may want to consider becoming more familiar with literature from IEEE and Colette Rolland and leverage it as needed.



IEEE STANDARD 1074-1997 - IEEE Standard for Developing Software Life Cycle Processes  
Colette Rolland, [http://en.m.wikipedia.org/wiki/Meta-process\\_modeling](http://en.m.wikipedia.org/wiki/Meta-process_modeling)

UX design processes and best practices need to integrate and map with the software development life cycle. It helps to think about UX work in terms of inputs and outputs. In Agile, **user stories** represent the documentation of a requirement, which is the key input. You have to decide what type of UX output you will create. A prototype? Wireframes? Working code? All three or two of those? These decisions will have an impact on process.

Analysis and architecture also don't have a clear place in a Scrum process. Things like user profiles and personas, tasks, workflows, and high-level design have to be tailored into the process. There is a movement in the Agile world called Agile Modeling ([agilemodeling.com](http://agilemodeling.com)) that provides leverage to do these types of things in a quick, not necessarily formal way. **For example, if you draw a site map or navigation model on a white board, you can simply take a picture of it and post it for shared team access. You don't necessarily take the time for formal documentation and you do it in a matter of hours instead of days or weeks.** That's the spirit of Agile modeling and it's a key example of successfully injecting analysis into your process.

## 3.3 Integrating UX Design with Scrum

Barker has observed four ways of integrating user experience design with Scrum. They are described here. You may see others based on your environment.

### **Big Upfront Design**

- Design resources operate in advance of development resources before a release
- Requires Product Owner collaboration
- Designs may be used for estimating and reducing risk
- Works for prototyping and proof-of concept work
- Works well with Kanban, Lean concepts
- No longer commonly used

### **Just In Time**

- Design done within a Sprint
- Typically requires “stubs” - predefined & commonly used chunks of design or development
- Requires a lot of collaboration; usually a fire drill
- Iterative with parallel efforts and predicted throw away work
- May be difficult to find a window for usability testing
- Easier to track in Scrum
- Works well with mature standards and pattern libraries
- Most common

### **Design Spikes**

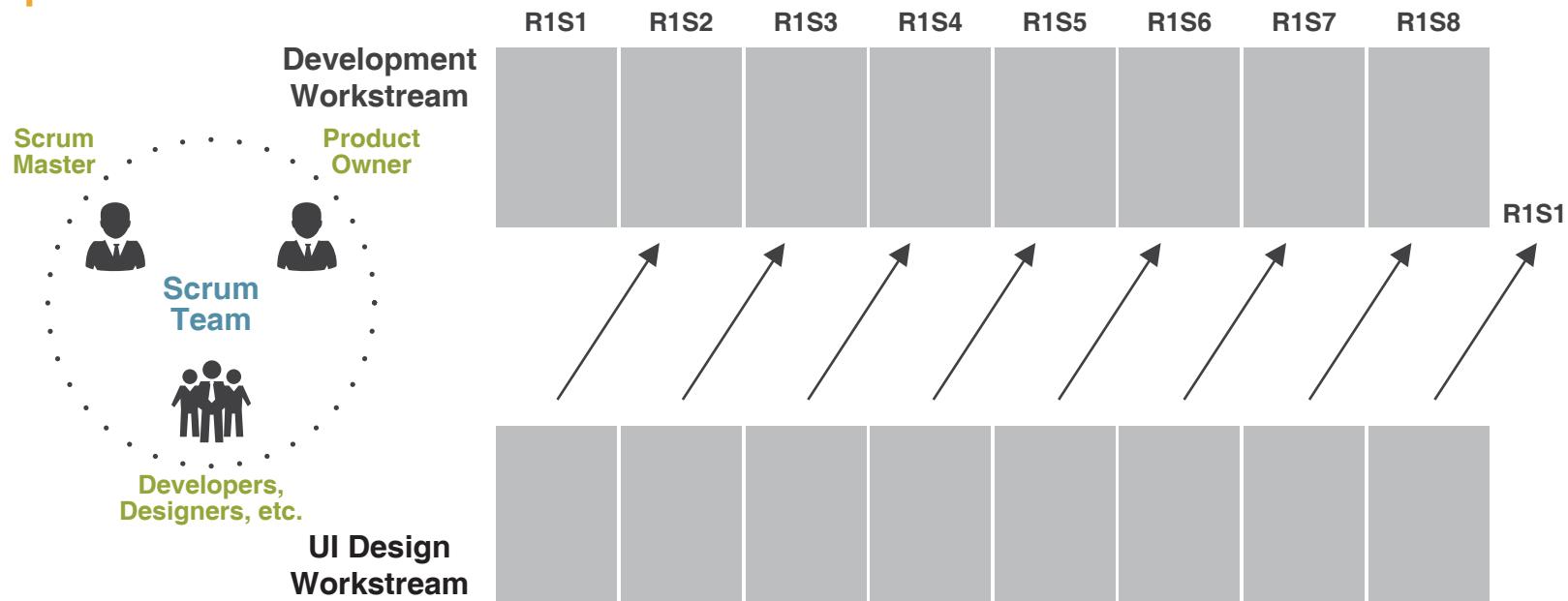
- Not developing a shippable product increment
- Interrupt a normal Sprint - “take a Sprint off”
- Typically used by architects to address technical exploration
- An exception and done in rare instances

## Sprint Pairs

- Design works a Sprint ahead of development
- Scope is traded off for designing within time box
- Requires extra coordination to track Sprints and handle coordination between design & development
- Sometimes described as “Scrumerfall” – it sounds more like waterfall
- Respects functional dependencies- coding is not happening on top of design
- Seems to have most traction in industry

The diagram below illustrates how UI design workstreams precede development workstreams.

## Sprint Pairs



If you were to consider using 2 approaches, use Just In Time OR Sprint Pairs, then add Design Spikes (if needed) or Big Upfront Design (if schedule allows).

## 3.4 Release Planning & User Research

A release planning event is an opportunity to determine which user stories will be incorporated into which releases. It's a good time to choose the methods for integrating UX and create a UX release plan overlays the overall plan. It also allows a good opportunity to add user research.

UX release plans should include the following:

- A “Sprint-focused” UI design work plan (critical path)
- Add Phase 0 work as appropriate and feasible
- Include iteration within release
- Define UX metrics
- Develop usability issues backlog

In terms of research, you would develop your research plan and highlight any upfront work that will be needed at some point within the plan.

Below is an example of a robust research program where there was time for Barker and his team to do some upfront research.

### Example Research Program



As a result, they were able to **start establishing UX metrics early**. They utilized UserZoom for remote usability testing which allowed them to measure and quantify the following:

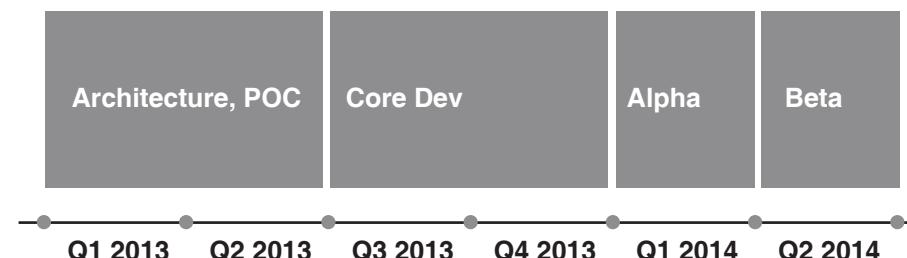
- Task completion rates
- Time on task
- Errors per user task
- Requests for help
- Click count
- SUS surveys
- Perceived ease of use
- Satisfaction
- Brand salience
- % of users impacted



Section 3.0 of this eBook contains much more on the topic of usability testing in Agile!

Release planning helps you ensure UX planning is covered. For example, if there is an opportunity for Phase 0 (before Sprint 1, day 1) the chart below shows key value-add activities you may want to perform. Release planning can help ensure that table stakes are covered – ensuring that a designer, developer, and UI developer will be together in the Sprints as part of the Scrum team. Once Sprints are underway, the chart shows important UX activities for the team to incorporate.

## Value Add UX Activities



### Phase 0 UX Activities

- User research
- Baseline metrics
- Groom usability backlog
- Ideation
- Concept testing
- UI Architecture

### Sprint-Driven UX Activities

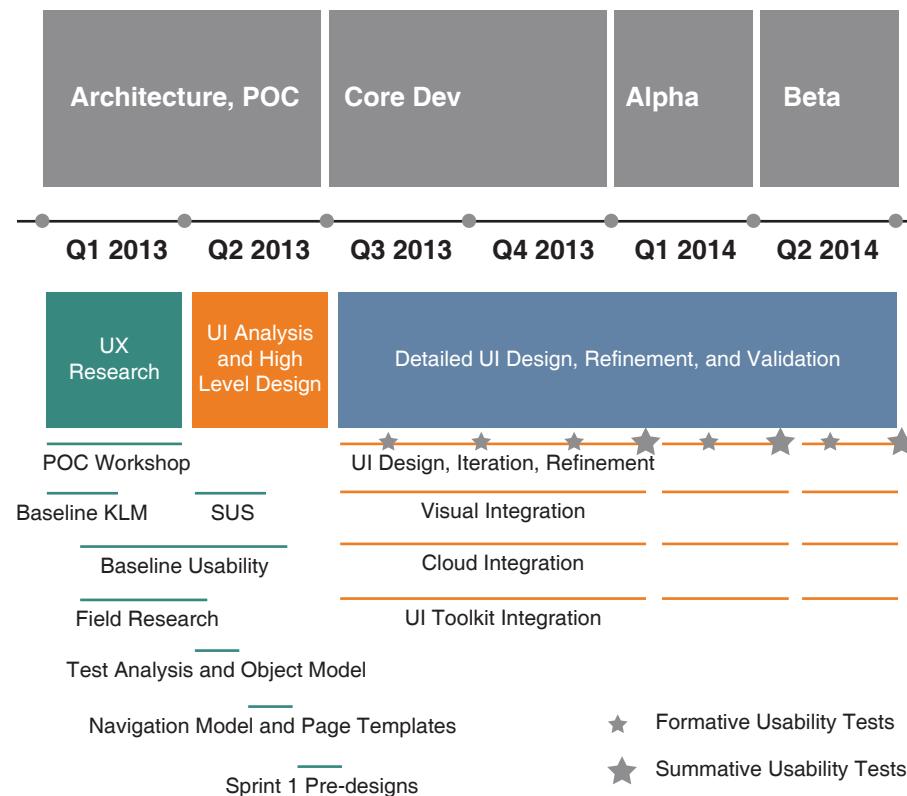
- Formative/Summative U-Tests
- Remote U-Tests
- Iterations
- Styleguides
- Design Integration
- Validation metrics

A **usability backlog** is an on-going list of issues discovered during usability evaluation and testing. The backlog needs to be actively managed so defects are fixed and improvements are made.

Defects in the usability backlog should be:

- Rated by severity
- Analyzed by theme
- Ranked for business value
- Prioritized for remediation per release

Finally, when putting all the pieces together, a UX Release Roadmap should look something like this:



A good roadmap includes a phase for research, a phase for analysis and design, then detailed work. Map out in advance where formative and summative usability tests will occur. End with a validation. The timelines, of course, can be adjusted for shorter or longer projects. In a shorter project, you may only have a few weeks to do all the work you need to do. You'll need to adjust or reduce the amount of design iterations, usability testing or other non-critical aspects for your project.

## 3.5 SUMMARY: Dean Barker's Seven Keys to Success



1. Assign UX designers to Scrum teams
2. Assign UI developers to Scrum teams and have them report to UX
3. Define a tailored process for UX with team input
4. Include Sprint/Phase 0 for UX research or other preliminary UX work
5. Influence and encourage iteration
6. Create a UX metrics program to drive improvements and prove value
7. Actively manage a usability defects backlog

# 4.0 Dispelling UX myths about usability testing in an Agile Design Process

**4 UX common myths about usability testing in an Agile Design Process** have been outlined by Jon Innes. Perhaps you have heard these before!

1. Agile and user testing timelines are incompatible
2. You need working code to test with users
3. It's hard to scale the usability testing process
4. Automating usability testing is not possible

Innes helps dispel these myths and provides more useful background on Agile. In order to dispel these myths, we need to acknowledge that the landscape of software design and development, technology, and the business world itself is changing and evolving. We need to adapt to change.

## 4.1 Myth #1: Agile and user testing timelines are incompatible

There so much going on in today's market. With today's technology we can do things that we could never do just 10 or 15 years ago. With social media it is very easy for customers to hear about alternative or competitive product offerings. With e-commerce, the cost for consumers to switch to an alternative product is smaller than ever. These factors make it more critical that your customers stay satisfied with your product. Product/software development processes have evolved to using Scrum and XP, which means software can be delivered on a much more frequent basis, as often as weekly or daily in order to keep pace with the market.

Over the past 10 years, many excellent researchers have complained about how hard it is to run traditional usability test with an Agile project. **Teams were accustomed to established processes for conducting usability studies, which might typically take about 4-6 weeks from start to results. However, given the timeframes for most Agile projects, this model no longer fits.** You'll soon see how changing the techniques and tools you use for testing will allow you to test frequently and effectively within Agile.

## 4.2 Myth #2: You need working code to test with users

There are many ways you can test concepts, prototypes, wireframes early and often in order to validate the user experience. Yes, eventually you can and will test working code, but waiting is guaranteed to compromise the quality of your user experience. Here are a few techniques to consider that are very common for UX in an Agile project.

## 4.2.1 The 3x3 Way: Test Multiple Concepts Early

The concept called 3x3 was originally developed by Carol Righi. 3x3 is a way of coming up with a conceptual design based on testing early low-fidelity alternative prototypes to help select the best design. It can save you rework later by ensuring you are starting down the right path. It's ideally suited for new projects or major site redesigns. It can be extended to greater numbers of design alternatives and iterations as needed. An important key to this type of formative study is recruiting the right participants - those who are unbiased yet fit the desired profile.

3x3 focuses on prototypes versus working code. It allows you to refine various prototype iterations and breed the best result. Once you have a solid single concept, you will then apply it through a traditional Agile or standard usability processes. You end up debugging problems and iterating on that one concept.

- **3 concepts, 3 pages deep**
- **3 iterations of design & test**
- **3 or more users per iteration**
- **Test low fidelity first, then high**
- **Improve concept each iteration**
- **Combine best elements into 1 concept & iterate**



## 4.2.2 The RITE Way: Refine One Concept Faster

Another technique you'll hear about in the Agile community is the concept of RITE testing (rapid iterative test and evaluation). RITE tests have been around for a while, and are very popular at Microsoft's game division and at Oracle.

With RITE, you run many small tests, and instead of waiting until all tests are finished to fix problems. You fix as you go and continue testing. Keep in mind it's purpose is to rapidly refine an existing design. While it can be adapted, it's best suited for refining a single design.

**RITE is a collaborative, fast-paced effort with your team. You test with a few users at a time to identify real, repeatable problems, without spending time to write a large report. Then work with your team to refine and fix issues.** Then repeat the cycle, testing again over several days, fixing other problems as they are discovered. RITE is typically used with working code or functional prototypes. Though RITE is well-suited to fit within short sprint cycles of just a week or two, the on-going schedule of testing & refinement can be very demanding on a researcher and the team, so scaling over the long term can be challenging.

## 4.2.3 The A/B Way: Refine Details At End

Finally another technique, which is very familiar on the Internet, is A/B testing. It has its pros and cons, which will be discussed again later. Usually with A/B testing, you build something, launch it on your site live, and collect data from large samples of site visitors to show where users are clicking. It is useful to show which site features or assets perform better with users, however, it is difficult to understand exactly what is going on. Why did people click or not click certain links, images, etc.? A/B testing data doesn't provide insights or details about why people do things or what their mindsets might be. On the plus side, A/B testing doesn't take time & effort for recruiting because you are tracking users who are already on your site.

## 4.3 Myth #3: It's hard to scale the usability testing process

Those who have tried traditional lab testing with Agile will know that it does not fit well into Agile schedules. Even remote tests can take up to a couple of weeks, especially if you need to find users that fit an exact profile you require. Test planning can take time and you may not have a stable UI until later sprints. It's also a challenge if the team is making UI changes throughout the sprints.

**What can we do?** All of the approaches discussed previously have benefits, and can work in an Agile timeframe. As you've seen, it's possible to perform testing inside of a week if you have a process to handle on-going recruiting. RITE allows you to fix problems as you go with smaller samples. A/B tests with large samples can be run quickly and are beneficial to measure user behavior on a live site as you are closing in on a final release. Methods or activities that require longer lead times can be used alongside the Agile timeframe (e.g. at Sprint 0). For example, you can handle recruiting. Or use the 3x3 Way to get early feedback to make sure that when you start a Sprint, you're running in the right direction. An important aside: Sample sizes are often misunderstood in the UX community, but it's important to realize that both small and large samples can be suitable for certain purposes and testing schedules.

## 4.4 Myth #4: Automating usability testing is not possible

With current technology you can effectively complete your testing through online means. Read on to learn more!

# 5.0 What is Remote Automated User Testing?

Many people think that remote automated testing refers to site analytics or surveys. They immediately assume it's A/B testing or other testing designed for advertising, such as measuring impressions or time on a site. Instead it is primarily **task-oriented testing**. It is not limited to the types of users that visit your site today. You can test with people who have never been on your site, or have never heard of it. Remote testing can be used on prototypes, which is key to early phases or when you are generating initial ideas or concepts.

With remote testing, **no moderators are required** – you set up the study online, wait for participants to complete the test, and watch the results come in which usually happens very quickly. Remote testing combines many of the benefits of lab testing with the scale of A/B testing so you can have a high degree of confidence in the results.

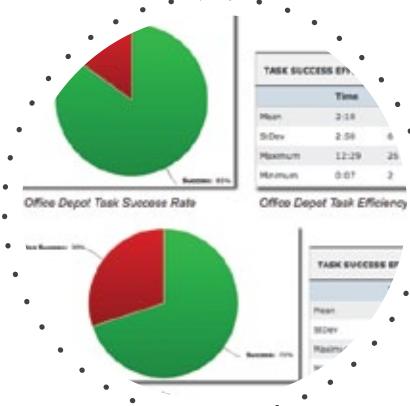
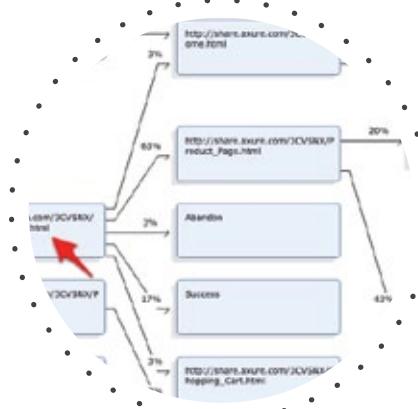
Remote automated testing is designed to test tasks and **time-on-task can also be measured**. From the point of view of web analytics, the longer a user is on a page or site the better – this is a logical benefit from the standpoint of general user engagement and advertising. But from a user experience standpoint, **if a user is taking a long time to accomplish a task, it's generally not a good thing**.



In a traditional in-person usability test, you're usually watching what users are doing, step-by-step, page-by-page. Having this information captured in the form of click path data can be very valuable. **Clickstreams** are collected in some remote testing tools such as UserZoom.

UserZoom also allows you to **track task effectiveness** by establishing success or failure metrics based on where users clicked or based on self-reporting by participants.

**Heatmaps** show where users click on a page. They prevent the need to closely watch each session and manually track or log a user's actions. Heatmaps provide powerful graphical representations to include in study reporting, or to quickly share with your team.



**Structured surveys** are available in many remote testing tools so you can include proven tools like System Usability Scale (SUS). Many tools also include live **recordings of each user session**, including recording of what occurs on the user's screen, audio recording of the participant speaking their thoughts out loud, and even video recording to capture a user's facial expressions.

As you may have already surmised, with remote testing you can **easily present multiple sites, prototypes or concepts to participants**. This can include iterations of your own site or designs, or testing one or more competitor sites against your site. With competitor testing, you can understand how your experience compares to your competitors, and you can benchmark user experiences to track how your site is performing compared to your competitors over time.

Remote automated testing allows you to **repeat tests quickly and frequently as needed**. You can test big things as well as small things. You can utilize large samples or smaller samples for more qualitative insights akin to traditional lab usability testing. Using prototyping tools like Axure in combination with remote automated testing tools is a powerful combination that can reduce both testing AND development costs.

## **Summary of the advantages of Remote Automated Testing**

- Conduct large scale studies before having working code
- Easily test & benchmark competitive sites against your site
- Compare functionality from past designs with minimal effort – iterate tests quickly and easily
- Gather large sample sizes like A/B testing and actionable insights comparable to traditional usability data
- Gather data from users who do not currently visit your site
- Reduce both testing costs and associated development

# 5.1 Tips for testing in Agile

## **1. Have personas agreed to by your team & develop recruiting strategy before sprinting**

Your team should have a clear, consistent view of who your target user is. You also need an appropriate recruiting strategy, so you can gather those people in a timely manner for your testing. Consider customer lists, people who are engaged with your blog, e-mail newsletters or other marketing, or outside panel(s) that you may use. Some remote automated testing tools also offer their own panels or the ability to access the largest panels in the US and around the world. Determine where you will recruit early on (such as Sprint 0). Don't wait until you're about to run the test. You may even be able to recruit users in advance and ask them to wait for further instructions once your tests are ready. If you have a very specific or unique persona, you will need to allow even more time to recruit.

## **2. Define hypotheses & target metrics early, and track usability metrics in a public place**

Define hypotheses and target metrics for your user stories early (again, ideally in Sprint 0) on with your team and product owner. This is what Lean Startup looks like. Don't figure out metrics as you're running a study. For example, we want users to do "X" 80% of the time, and that they can do "X" 80% of the time in under 5 minutes. Agile works best when information is posted publically, so you want to have metrics and other study information posted on walls, or shared on a server for a remotely distributed team.

## **3. Test early design concepts not just final details**

Test design concepts as early as possible. Agile's sprint structure is well-suited to allow you to test iteratively, early and often, even with just sketches of UI designs.

## **4. Automate user testing for stories as you go & retest - waiting until the end is the waterfall way**

Automated user testing for stories may seem like a new concept, but it's a really important one. Using an automated tool to run studies makes it easy to repeat studies with different design iterations and makes it easier to compare results. Remember - waiting to test until the end of your design/ development cycle is the waterfall way! Working your testing into sprints as often as possible is the Agile way!

## 5.2 Tips for using Remote Automated Testing tools

### **1. Always run small pilots before large-scale tests**

When using a remote automated testing tool, always run a pilot test first with one user or a small handful of users, similar to a “dry-run” test you may perform in the lab before the real participants complete the test. A big plus of automated testing is the ability to run studies quickly. But if you hastily pull the trigger, you could get bad data, waste time and money, and sacrifice your participants. Run an early pilot of your test and learn from the results. Make changes as needed to clarify the test flow, optimize the data collected, or make corrections to the site or assets used in the test. As a bonus, you will get an early glimpse into the results!

### **2. Allow at least a few days for user testing**

No matter how you recruit participants or what remote automated testing tool you use, don’t underestimate the time needed to get results from your test. While most automated tools are capable of collecting same-day responses from participants, be sure to take into account the amount of time you’ll need to coordinate with your team. For example, others may want to review your test script. Or engineers or designers may need to prepare the site, prototype or other assets you are testing. Typically you will need at least 1-2 days to setup the test and run a pilot, 1-2 days to launch the test and receive results, and perhaps more time to review and analyze the results. Actual time will vary of course, depending on the nature and complexity of your study.

### **3. Combine intercept recruiting with panels to minimize costs while reaching both new and existing users**

Intercept testing involves testing with people who are already going to your site, and inviting them to participate in a study during their site visit. Often you can eliminate compensating these users, or just provide an incentive from the site itself such as a promo code. Participants who are part of a panel or those who are specially recruited based on their profile are generally compensated for their testing efforts. By combining these recruitment methods, you could save costs and expand your learnings based on a more diverse pool of participants.

#### **4. Keep to a small number of tasks to minimize drop out rates**

Don't expect remote participants to spend 1-2 hours on a given test, as they would in a lab. 10-20 minutes is a reasonable length for a remote participant. 30 minutes is a good maximum, even if your tool allows for longer tests. Beyond this, participants are known to become fatigued or distracted by their own environment, for example, they'll want to get a cup of coffee or answer the phone calls they are missing.

#### **5. Use content analysis techniques to analyze open-ended survey data**

If you have open-ended questions in your survey, consider content analysis tools such as text clouds or word clouds that find common keywords across responses to help you find common themes across written responses. Some tools such as UserZoom have this capability built in to their analysis capabilities.

## 5.3 Keep UX principles at the forefront

In all of this, be sure to keep the actual user experience as a priority in Agile. Many core UX principles stay the same in Agile. Think about testing as a learning opportunity. Continually ask what is being learned and identify users' problems that you're trying to solve. Measure against specific metrics when possible. The definition of "done" can only be determined by users. It doesn't matter if your team thinks the product is done, or how great the product owner thinks the end result is - if you have data that says users can't use it or don't like it, as a UX team member you must continue to underscore user experience problems.

### **Questions to ask your team**

- Have you validated the user stories and personas?
- Did each iteration incorporate user feedback?
  - Did we learn anything about user's needs or whether or not they could use the product?
  - Would they recommend the product to a friend?
  - Did we make the product measurably better?

# 6.0 Agile Metrics: An Example from Jeff Sauro

In this section, Jeff Sauro provides some great examples of how to gather real UX metrics on very short timeframes with Agile.

Consider the example where a designer has generated some wireframes or design comps and you want to quickly find out which are preferred by users. The challenge is to take these static assets, and create tasks or scenarios around them that generate real metrics to help you make decisions.

Jeff suggests a “**Trick Test**” which is a combination of a tree test and click test.

Start by creating simple .jpeg files from Photoshop or other tool a designer may have used. Then create a realistic user task and scenario and consider the high level business objective – what do you want users to do. Let’s take an example of downloading drivers for printers from a major manufacturer’s website, and demonstrate how to obtain a series of metrics that show how well the task is accomplished.

One useful metric is **findability**— you want to know where users would click first in order to accomplish their task and how long it takes them to locate it. After they have chosen the area(s) they would click on a static image, you can ask how confident they were that they would be able to accomplish their task, and how difficult they thought the task was. As well, you should overtly ask which design concept do you actually prefer? These should all be closed-ended questions, usually in the form of ratings scales or single-choice / radio button responses.

## The Metrics:

- **Time to locate (task)**
- **Confidence**
- **Task Difficulty**
- **Preference**

Below are three comparable designs that came out of Photoshop around the example of locating and downloading a printer driver.

## 3 Design Comps: 2 Navigation Comps



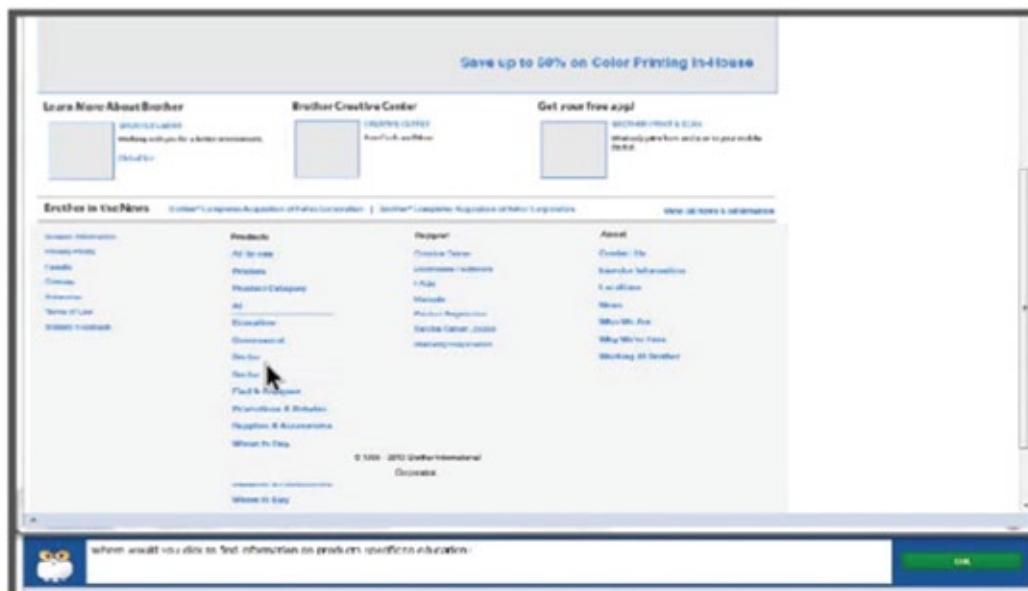
We want to know:

- Which design better supports find ability when participants attempt tasks?
- Which design takes less time?
- Which design creates a higher confidence in users' ability to complete tasks?
- Is there a general/overall preference?

In this example, 30 participants were recruited for an online study. To make it clickable, simple CSS & short HTML was used to create hotspots on the navigation elements. So the image looks like a webpage, and when users mouse over the hotspot, the navigation will appear. You may need a professional to help code, but it is a good idea to add interactivity to your designs.

You can use services like UserZoom and YouEye.com to set up the tests for users to complete online. Both services can recruit participants online, and record their screen, audio and video. You'll actually be able to see users natural behavior as they locate and click the elements in the design.

In this example, UserZoom is administering one of the tasks. UserZoom can even automatically randomize the order of tasks.

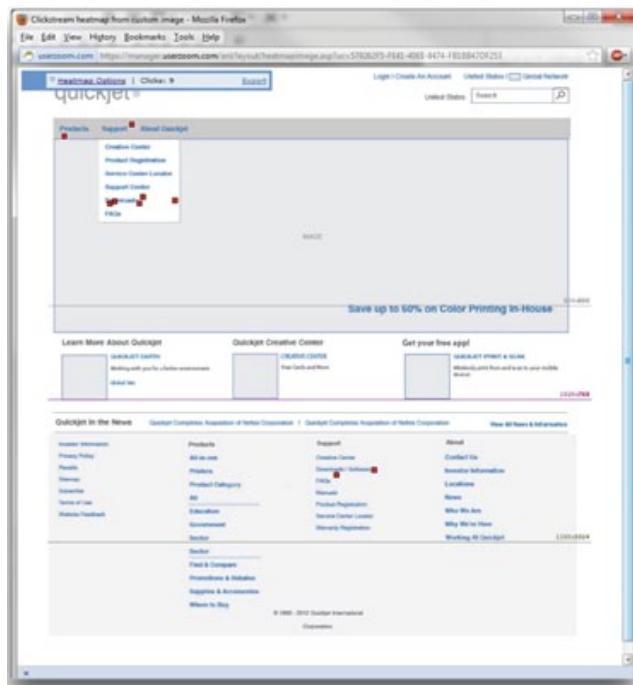


Metrics are collected after each task. Users would rate how confident they were with the tasks, which is compelling and useful quantitative data. But often one or two video clips of users really struggling with a design comp is all that's needed to convince a designer or stakeholder that a design is not going to work or needs modifications.

UserZoom can also produce click maps. This example shows where users clicked on one design comp. If you use confidence intervals, you could make inferences about where users would be likely to click in a real-world scenario.

## Click Map from UserZoom

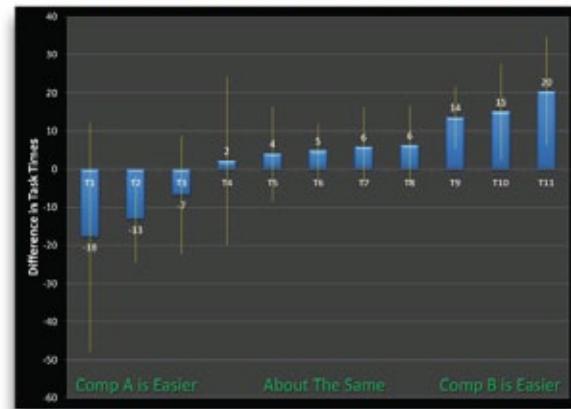
- 73% click on top navigation
- 90% confidence interval (50% - 93%)



Less automated solutions can also be used with Agile. In the example below, moderated sessions were held over Go To Meeting where users attempted the same tasks on 2 comps. Time-on-task was recorded as users were asked to find information on the designs. In this case, with just 13 users there were statistical differences. Design B performed statistically faster.

## Which Design / Tasks Are Faster? (13 Users)

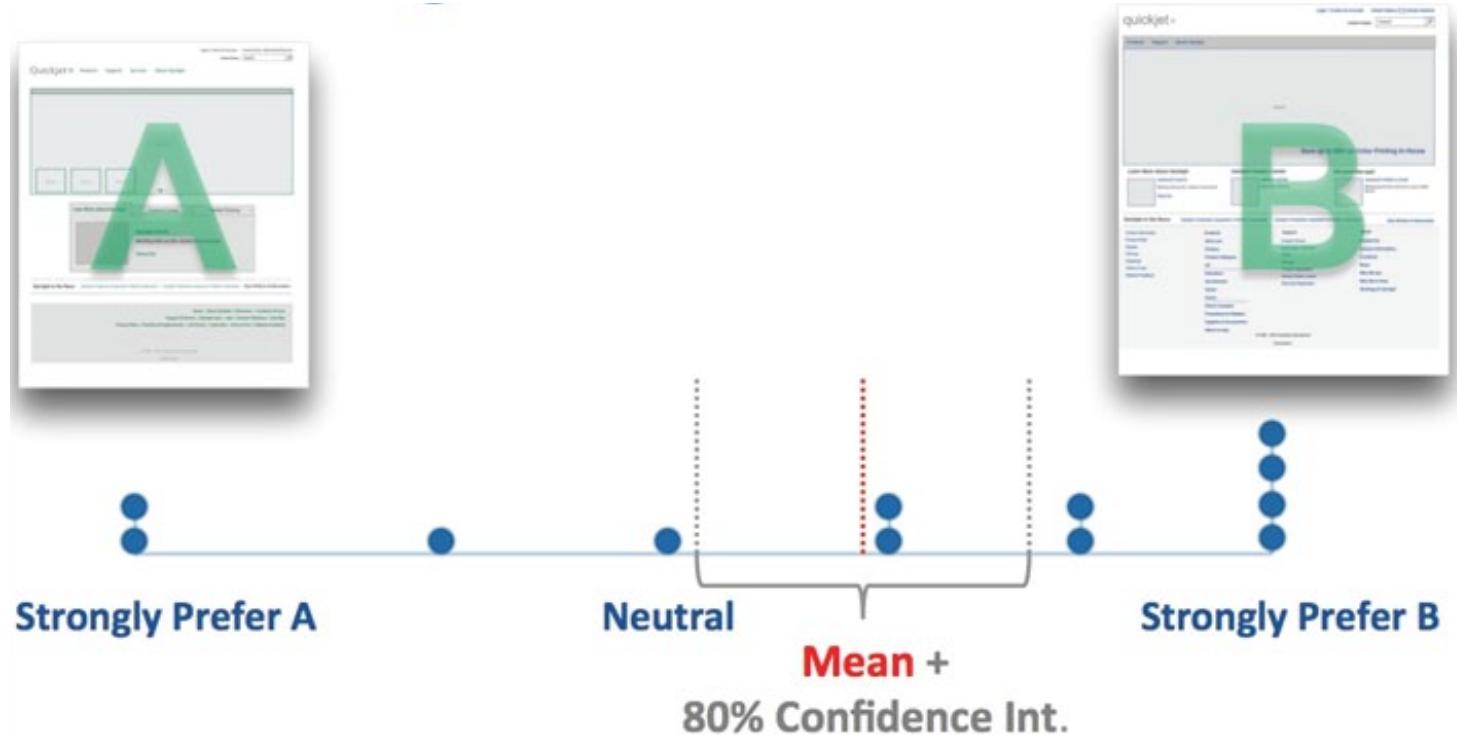
### Which Design / Tasks Are Faster? (13 Users)



Easier and faster in this case is referring to time. Even when data shows that users' time spent on a task is about the same across designs, it can help calm concerns that new designs won't be as difficult to adopt as anticipated. Early evidence can provide confidence that your team is moving in the right direction and you can more quickly shift focus to test other tasks.

In terms of the design preference data, the chart below shows that even with a smaller sample, there are reasonably tight confidence intervals to state that Design B is preferred by participants.

## Which Design is Preferred?



To summarize, this example shows that in less than two weeks it is possible to gather user feedback and data to select the best design comp and navigation. From there, your team can confidently move forward to create a stellar user experience.

# 7.0 CASE STUDY: Formative Usability Testing in Agile: Piloting New Techniques at Autodesk



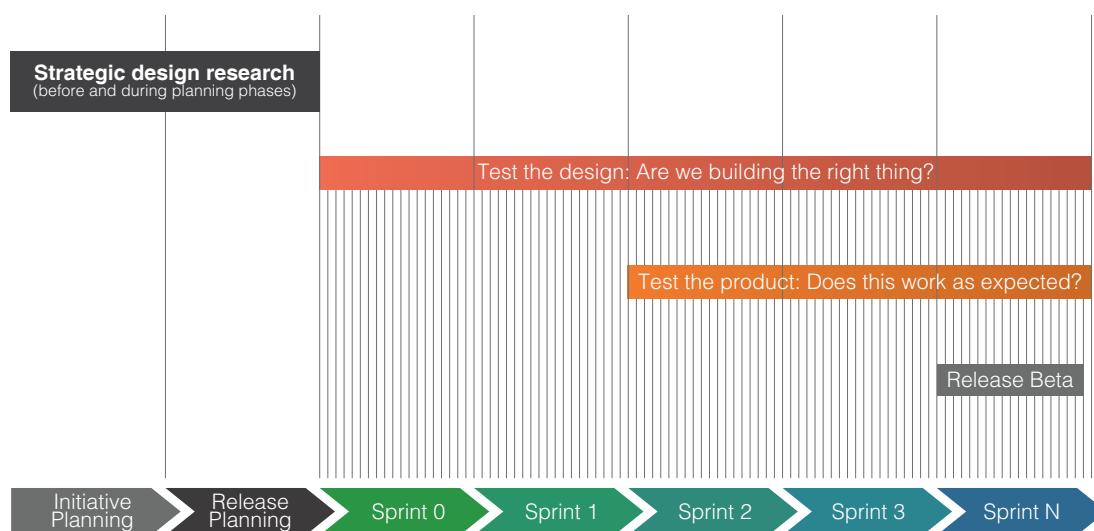
## About Autodesk and the impact of Agile

Autodesk recently moved from a waterfall approach to Agile. Eunice Chang and Olivia Williamson from Autodesk were willing to share their experiences for a recent webinar, and their case study has been adapted for this eBook. Their User Research Team adapted to Agile by forming a feature-specific customer council that was leveraged at different stages of the Agile development process. The council was leveraged both at the start of the Sprint, when testing the design before implementation, and also at the end of the Sprint, when testing features that were already implemented. As you read further, you will learn about best practices for how to recruit such a council, how to get the most out of a council, and what success might look like.

Autodesk is a multinational software company, focusing on 2D and 3D design software. Designers use AutoDesk software for a variety of purposes including architecture, construction of buildings and bridges, manufacturing and entertainment. One of Autodesk's flagship products is AutoCAD, an application for computer-aided design (CAD) and drafting. AutoCAD has generally been released on an annual cycle. AutoCAD has desktop, web and mobile AutoCAD applications. The customer council was developed to gather user feedback about features for the desktop application.

The introduction of Agile in the AutoCAD development process represented a big change for the organization because the annual release cycle trained the team to work sequentially from planning and development, to testing, then to deployment. One of the big challenges with this approach is that the bulk of user testing occurred at the completion of development, and by that time it was often too late to make significant changes to features. By validating the quality of releases on a per-Sprint rather than yearly basis, Autodesk's research team was able to restructure the customer feedback process to be more responsive and flexible to customer needs and feedback. The team provided customers with a version of AutoCAD that they could use with their own projects, so Autodesk could receive feedback earlier and in context with users' own environments and drawings. Autodesk was able to make quick corrections during each Sprint based on customer feedback.

The Agile process starts at the initiative and release planning stages, where the AutoCAD team identified three or four new features that they wanted to roll out in the next version. In addition to strategic research at this phase, the team started conducting iterative concept and usability testing. Before each Sprint the team wanted understand whether they were building the right thing. At the end of each Sprint they wanted to test the working product to see if it functioned as expected.



The AutoCAD team defined the **minimum viable solution (MVS)** as “the smallest usable and sellable set of functionality that provides value to the user”. To get to MVS they used **story-mapping** to understand and prioritize the user needs for each feature area. As an example, the MVS for a particular new cloud-sharing feature in AutoCAD might allow users to create and attach a drawing, but not edit it.

# 7.1 Customer Councils

In order to determine MVS, the AutoCAD team set experience goals called the **minimum viable experience (MVX)**. Feature-specific customer councils were developed to get user feedback on the MVX. They used online community tools to post announcements, feature videos, tasks, satisfaction surveys and more. The customer council acts as a mini beta for each feature area. Council members were recruited based on their specific needs and interest. They committed to testing new builds every 1-2 Sprints and completing satisfaction surveys. Each council had about 50-75 users to start, and they were not compensated for their participation unless they participated in one-on-one feedback sessions outside of their regular council commitment. AutoCAD decided not to compensate council members because they wanted to emphasize the collaborative relationship and partnership they endeavored to establish with their customer base.

Welcome to the AutoCAD Connected Desktop Council

**Customer Council**  
Connected Desktop



**Hi Melissa!**

Thanks for joining us, and welcome to the AutoCAD Connected Desktop Customer Council. We're excited to have you partner with us on making the **Design Feed** a fun collaborative feature for you and people that you work with.

**Meet the Autodesk Point Cloud Team**

**Introductions: Meet the Autodesk Team**

**Einar Bel** Einar is the entrepreneurial executive with extensive complex projects and businesses in an international environment. Autodesk Labs and former founder and CEO of Alice Lab, an int'l software to speed up reality capture pipeline for creating high resolution photogrammetric data acquisition. Interests lie in building individuals and making them commercially successful.

**Mauro Bourassa** Mauro is the Engineering senior manager. He has been with Autodesk for 24 years, working in product development. He joined the AutoCAD Reality Capture team out of BIMD and QA team members, which was responsible for enhancements in AutoCAD and the new graphic system extension for all AutoCAD series, in its off office time you can find him sailing the SF Bay and Pacific coast.

**Maren Foster** Maren is a User Experience Architect with Autodesk. He has worked on projects including AutoCAD, 3ds Max, he now leads the design efforts for Autodesk's Reality Capture.

**Working with K2**

Please rate how strongly you agree or disagree with each of the following statements:

I was easy to do everything I was asked with the K2 application (e.g., create the new project, navigate point clouds, change visibility of point cloud data and display style).

1. Strongly Disagree	2	3	4	5	6	7. Strongly Agree
----------------------	---	---	---	---	---	-------------------

I would imagine that most people would learn to use the K2 application very quickly.

1. Strongly Disagree	2	3	4	5	6	7. Strongly Agree
----------------------	---	---	---	---	---	-------------------

Using the K2 application will enable me to be more productive.

1. Strongly Disagree	2	3	4	5	6	7. Strongly Agree
----------------------	---	---	---	---	---	-------------------

**Working with K2**

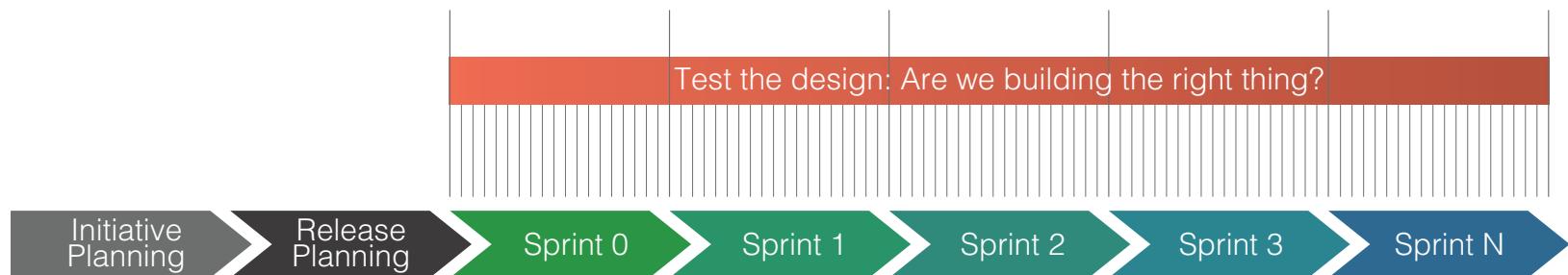
1. Complete Import scan files to create Point Cloud Project	<a href="#">Scan Details</a>	<a href="#">Next Step</a>	<a href="#">Results</a>
2. Complete Navigating Point Clouds	<a href="#">Scan Details</a>	<a href="#">Next Step</a>	<a href="#">Results</a>
3. Not Started □	Changing Point Cloud Display Style	<a href="#">Scan Details</a>	
4. Not Started □	Use your own scan data	<a href="#">Scan Details</a>	
5. Not Started □	Complete the Task Feedback Survey	<a href="#">Scan Details</a>	

Most of AutoCAD's research is done remotely. They recruited users from around the world for their customer councils. The example shown here is for the Connected Desktop Customer Council. In order to develop trust and a collaborative relationship with users, they posted pictures and bios of the AutoCAD product/UX team and also encouraged council participants to send photos of themselves.

## 7.2 Testing the Design: Are we building the right thing?

The AutoCAD team became keenly interested in whether what they were designing was truly the right thing for customers. They began looking for customer feedback as early as possible. Within their development cycle, they added 1 or 2 Sprints before each design was scheduled to be implemented, and used the opportunity to have conversations with customers. They showed customers raw concepts or sketches, even if functioning prototypes were not yet available. They aimed to receive early feedback and continued with an iterative feedback cycle throughout each Sprint as feature designs evolved. They started recruiting customer council members during release planning & early Sprints and continued to build up council membership. Early council participants were invited

to have informal one-on-one interviews with the AutoCAD team. The AutoCAD team would gather feedback from council members using a variety of tools such as remote user testing, remote conferencing and screen sharing, card sorting, concept sketches, and prototypes. Tools such as UserZoom provided interesting and useful insights. These early conversations allowed the team to refine designs to address usability issues, update or add new user stories or scenarios for later Sprints, and collect additional details that can help with future feature planning.



<b>When</b>	<b>How</b>	<b>Outcomes</b>
<ul style="list-style-type: none"> <li>• One or two sprints before design is scheduled to be implemented</li> </ul>	<ul style="list-style-type: none"> <li>• One-on-one conversations</li> <li>• Remote user testing <ul style="list-style-type: none"> <li>• Card sorting</li> <li>• Concept sketches</li> <li>• Prototypes</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Refine designs to address usability issues</li> <li>• Update or add new user stories and scenarios</li> <li>• Add additional detail to future feature planning</li> </ul>

These interviews were approached as interactive dialogs - a less formal approach than traditional usability testing, with the goal of encouraging ongoing customer engagement. Ideally council members would start to see changes in the product and realize that their contributions are making a difference. These conversations would be scheduled throughout the Sprints, and topic lists were maintained that helped the team prioritize and balance both strategic and immediate needs. The team might seek specific usability feedback on a prototype or might explore the viability of a new workflow.

They found that a sample of about 5 to 7 people gave them well-rounded feedback for a particular topic, even with varied conversations with each person.

The AutoCAD team would initiate the conversations by better understanding users' current environment and needs:

- How are they currently using with the software?
- Where are gaps in their experience?
- What additional needs could be met?

As an example, the AutoCAD team found that users had a need to export images in addition to exporting data for building materials. Users were spending a good deal of unnecessary time copying and pasting pictures into the spreadsheet. This proved to be a great example of a gap in the experience.

The team would solicit feedback on alternate approaches to feature design & implementation with questions such as:

- Tell us the pros and cons given your workload, environment, & needs what are the pros & cons to each approach?
- What concerns do you have?

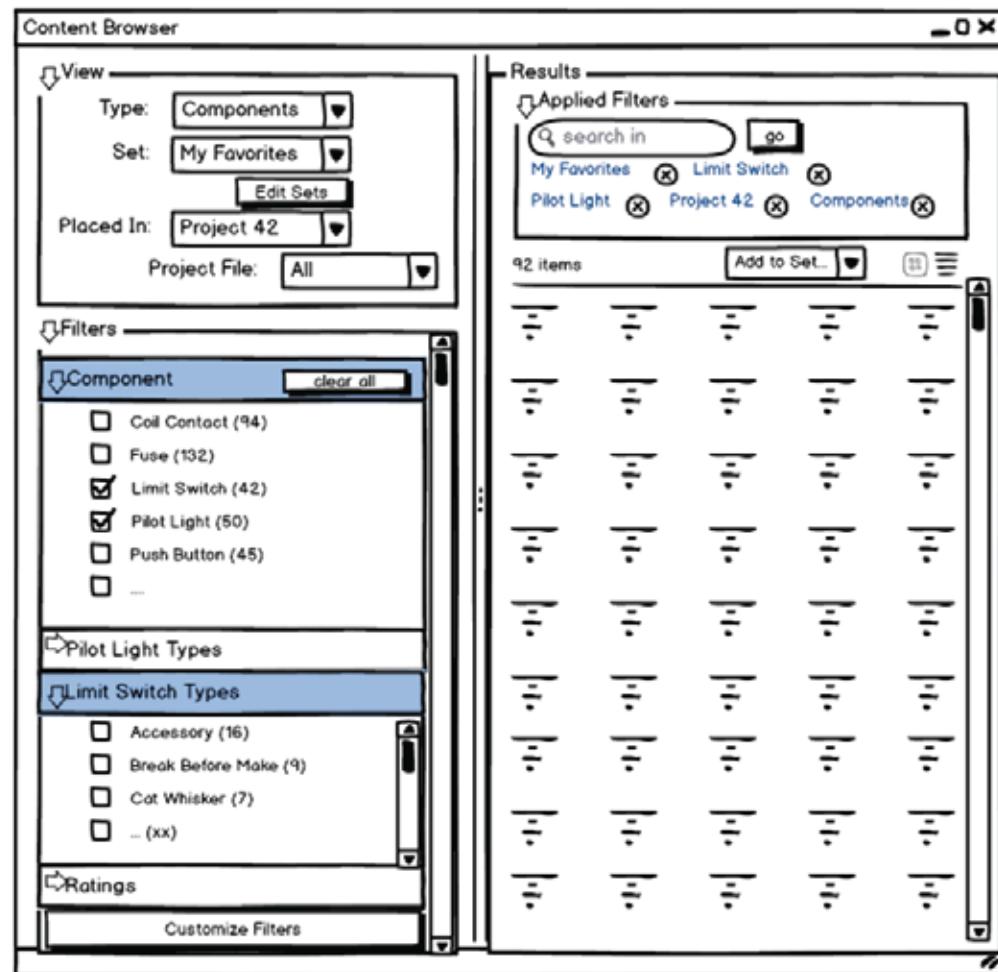
The team also presented more detailed concepts, sketches or other visuals and asked questions such as:

- How might you use this?
- How might this be relevant in your work?

These early needs-finding exercises helped with feature prioritization and validated real designs that were moving into implementation.

One example was complex searching and filtering capabilities for content. This was considered a highly important and frequently used element of the application and the AutoCAD team wanted to get the interaction right.

A sample sketch is included here:



The AutoCAD team found it valuable to relate directly to real-world customers. They were able to point to specific examples of feedback from customers, which emphasized how the software experience personally impacts customers in their day-to-day work.

### **Summary: Testing the design: What the Autodesk team gained over time**

- **Immediate:** Specific input that affects current design decisions
- **Ongoing:** More detail on the customer's environment and application needs
- **Long-term:** Working relationships with customers
  - Get customers invested in product success
  - A deeper understanding of the customer's world

## 7.3 Testing the Product: Does it work as expected?

At the end of each Sprint, the AutoCAD team wanted to test what they've built and learn if the product meets users' expectations.

To validate the product, Customer Council participants tested AutoCAD features on their own computers with their own data, for something like a 1-2 week beta test. Participants are asked to try out specified functionality then complete surveys and engage in discussion forums about their experience. The AutoCAD team measured responses against pre-determined metrics and validated the MVX. If the product didn't meet users' satisfaction targets, the team would re-work in a subsequent Sprint cycle then reevaluate with the Customer Council.

Response from the Council was active and quick – often within an hour of being notified of new functionality to test, users were engaged in testing and offering ideas for improvement via the online forum.

# 7.4 Best practices based on the AutoCAD Customer Council

## 1. Recruiting for Customer Councils

- a. Recruit broadly. Consider the following potential participants:
  - i. People who have participated in usability testing in the past
  - ii. People who attend product conferences (e.g. Autodesk University conference)
  - iii. People who are members of user groups
- b. Send a general invitation e-mail such as: "We have some interesting upcoming customer feedback opportunities - would you be interested in participating?"
- c. For those who respond to the invitation, survey them to gather details about:
  - i. Who they are & what kind of work they were doing
  - ii. What amount of time would they devote
  - iii. What kinds of things would they be interested in
- d. Create Customer Council groups based on product MVS's, or other feature/functionality groupings
- e. Invite potential participants based on their interest or experience levels for each MVS.
- f. Expect invitation response rates to vary, depending on users' interest or prior experience with features.  
The AutoCAD team saw response rates of 7-15%.

## 2. Planning & managing Customer Councils

- a. Decide upon tools that will be used to communicate & collaborate with the Councils and collect feedback. Consider online tools & methods to facilitate the following app downloads (if needed), tasks/scenarios, surveys, comment-posting, online meetings, & any necessary hardware or other equipment
- b. Engage new Customer Council members immediately by asking members to introduce themselves and post information about the kind of work they do
- c. Post bios of your product/UX team

- d. Start discussions right away to gather needs & expectations even before Council members start providing feedback on release builds
- e. Set expectations about the amount of time Council members should devote to testing, for example, 8-10 hours per week for the next two weeks
- f. Set expectations for one-on-one conversations
- g Decide upon compensation, if any, for Customer Council participation. Consider testing duties as well as one-on-one conversations.

### **3. Planning your testing cycles**

- a. Clearly identify task flow(s) that Council Members should follow
- b. Aim to create scenarios that allow Council Members to use the product in context with their actual day-to-day work
- c. Introduce the release via a webinar or conference and set users' expectations about the capabilities of the build. Conduct a demo and emphasize what users expect to be able to do or not do based on the current state of the release. Make these sessions interactive and allow Council Members to ask questions.
- d. Think of creative ways for Council Members to engage with each other through comparing their experiences, or perhaps by setting up contests or competitions around creative ways they used the application while testing.
- e. Track task completion closely. Design tasks so users can easily track their progress through the task workflow.
- f. Continually ask for feedback. Don't wait until the end of the testing cycle. It's better to receive in-the-moment feedback on each task, and get an early sense for how the build is performing for users. Ask users to report any problems they encountered and submit suggestions they may have. Respond quickly and make critical adjustments if you can, to avoid wasting users' time on show-stopper issues.

#### **4. Maximize the opportunity for quality feedback from Customer Councils**

- a. Look beyond traditional usability testing
- b. Use the opportunity to collect feedback not just on current working designs or workflows, but also future concepts and a strategic understanding of customers' needs and goals
- c. Be clear with the Councils about what they can contribute and how they are beneficial to the product development process. Keep communication precise and clear.
- d. Keep Councils engaged on a variety of levels. Keep asking questions between cycles. Use the opportunity to schedule one-on-one conversations or generate additional online discussions.
- e. Don't over-burden your Councils or your team... be realistic about number of releases you'll distribute to Councils and the amount of time they should devote to testing and providing feedback. Strike a balance between the amount of time your Councils can reasonably devote and the amount of feedback that is absolutely necessary for your team.

### **7.5 Summary: Customer Councils – What success looks like**

1. Successful Customer Councils are those where **participants understand their role, are actively engaged, and (at some point) can actually see their feedback enacted within a product**. Ideally they should evaluate your product releases within the context of their day-to-day work and apply real scenarios.
2. Bringing feedback from Customer Councils to your internal stakeholders will **emphasize the importance and seriousness of fixing design and usability issues** throughout the Agile development process.
3. Your Minimum Viable Experience (MVX) is being validated through real customer usage and feedback from your Customer Councils. When you establish MVX target metrics, you can ensure that **new/modified features are included in a final release only when they meet MVX targets**.

Autodesk is continuing their Customer Council program they hope your team can be successful as well by learning from their experiences!



# Further Reading

This eBook is based upon the following webinars:

## **How to Integrate UX into an Agile Environment**

<http://bit.ly/1i7V1L4>

Dean T. Barker, Sr. Director of User Experience at Optum

## **How to Master UX Testing in an Agile Design Process**

<http://bit.ly/1PDompW>

Jon Innes, 'The Agile UX GURU' at UX Innovation LLC

Jeff Sauro, 'The UX Measurement MASTER' at Measuring Usability

## **Formative Usability Testing in Agile: Piloting new techniques at Autodesk**

<http://bit.ly/1KqPMkx>

Eunice Chang, Olivia Williamson at Autodesk



**userzoom**  
THE UX INSIGHTS COMPANY

[www.userzoom.com](http://www.userzoom.com)

#### San Jose, CA (USA)

10 Almaden Blvd #250,  
San Jose, CA 95113  
Phone: 866-599-1550

#### Denver, CO (USA)

1630 Welton Street #500C  
Denver, CO 80202

#### Madrid (Spain)

C/ O'Donnell 12, 2<sup>a</sup> planta,  
28009, Madrid  
Phone: +(34) 91 111 48 82

#### Barcelona (Spain)

Av. Diagonal, 618 3<sup>o</sup>D  
E08021 Barcelona  
Phone: +34 93 414 7554

#### London (UK)

Thomas House,  
84 Eccleston Square  
London, SW1VPX  
Phone: 0203 948 1360

#### Manchester (UK)

4-6 Bank Square Wilmslow,  
Cheshire SK9 1AN  
Phone: + 44 (0) 1625 525 650

#### München (DE)

Kronstädterstrasse 4,  
81677 München  
Phone: +49 (0)89 45 35 35 25

