10 Product Management Research Pitfalls

by Lorraine Chapman
“Unfortunately, user research is rife with pitfalls.”
Today’s product managers know it’s their job to be in tune with the market and with real-world users. More and more, they turn to user research techniques like customer interviews, focus groups, analytics, personas and prototyping to discover what customers want next from their products. Managers from companies like Apple, SalesForce.com, and SAP, who incorporate user research into their development cycles have consistently released software that sets a new bar in the industry, so won’t these techniques benefit everyone?

Unfortunately, user research is rife with pitfalls. Often what customers say they want and what they really need are very different. Surveys, focus groups, and analytics can be misleading. And even if you do glean the correct insights from your research, incorporating the results into the next product release can be even more of a challenge.

In this paper, we identify ten of the most common errors that product managers and business analysts make when gathering user research and customer feedback, and explain how to avoid them.
Mistake #1

But We Asked Users What They Want!

HAVE YOU SEEN THE EPISODE OF THE SIMPSONS where Homer’s long-lost brother, who owns a car company, instructs his engineers to design a car that average people want by getting input from Homer — the typical middle American customer? Homer demands extra large cup-holders, tail fins, a bubble dome, and horns that play La Cucaracha. The car is an expensive flop, and his brother’s company goes bankrupt.

In the end, customers aren’t product designers. They will happily give you ideas for your product, but they love to jump straight to the solution as they see it without understanding interactions or the context in which people use the software.

Customers might even come up with a wish list of features or changes such as, “This button should be blue” and “I want to print with one click.” But that doesn’t give you insight into why they need that feature, or why it might (or might not) help other customers.

When a customer asks for a feature, your first job is to understand the underlying problem — Why did they request that particular feature or fix? By getting out of the solution space, and staying in the problem space, you understand the motivations and context behind the feedback. (We’ll talk more about that later.)

Here’s an example. Imagine you are creating an application for border security. It scans passports and displays a warning if the traveler has a criminal record. Your customer, the owner of the new system, wants to be sure that the customs agent sees the warnings, so they tell you to make the screen blink red whenever the passport scan brings up a suspicious person. Now imagine a land border crossing at midnight, where the border guard sits in a glass booth. It’s dark, and the person crossing comes up as having a criminal record. The
screen flashes red. It reflects off the glass, lighting up the whole booth — and is visible to the person in the car outside. They know they’ve been flagged. If the traveler is armed, you have the potential for a deadly situation.

Maybe lives don’t rest on your application, but blindly following user feedback without understanding the problem or context of use can have consequences for your software, too.
A LOT OF PEOPLE RECOGNIZE that simply talking to customers and doing what they say isn’t enough. But many of these people conclude that talking to real-world users is therefore not a good use of time, usually quoting the Henry Ford adage, “If I had asked people what they wanted, they would have said faster horses.”

To quote a famous saying right back, “Don’t throw the baby out with the bathwater.”

Talking to end-users is necessary to gain true customer insight, but it’s not enough by itself. You need to put the ideas you get from talking to customers into a wider context. You do this using data from other research techniques, particularly observational research — watching users use the product in their real-world environment and understanding their unspoken needs.

For example, when we do a field study and task analysis, we often find workflow issues with the product that cause a lot of frustration and time wasted. The customer didn’t tell us these issues during the interview, but these issues usually present a greater opportunity for product improvement than the customer’s wish list of features.

Additionally, product and marketing executives are investigating trends such as web analytics and mining social media data related to their products. These should be used in addition to (but not as a substitute for) direct and observational research techniques. For examples of some of these methodologies, see Macadamian’s research technique quick reference as well as a practical application in Overhaul a UI Design Without Upsetting Users.

Often, what customers say they want and what they actually need differ significantly. The solution is not to give up on research, but to triangulate the data using research techniques that have a demonstrated track record.
**Mistake #3**

We’ve Gathered Tons of Data

**WITH THE RISE OF ANALYTICS,** this research trap has been growing. Tools like Google Analytics, Flurry, and Preemptive Mobile give you more data than you know what to do with, but that’s not inherently a good thing.

If you ask enough questions, and gather enough data, meaningless patterns will emerge.

Simple example: if you ask enough people what month they were born in, and a hundred other research questions, you will “discover” things like people born in November drink a disproportionate amount of Budweiser compared to everyone else, or that 86% of people with July birthdays like driving foreign cars over North American cars.

You haven’t actually discovered anything except a statistical phenomenon known as **clustering.** In a large set of random data, you get clusters of the same type of information. Roll a pair of dice 100,000 times and you’ll get strings of snake eyes or 12’s and other unlikely sequences. It doesn’t reveal a strange meaningful pattern. There are just a lot more clustered patterns than non-clustered possibilities out there.

To find out if the trend has meaning, you would have to run a separate follow-up experiment. Automatically assuming that a cluster of data has meaning — and making design decisions based on that — can lead you astray.

For example, a client once asked us why their website was so popular on Tuesdays, as shown by their demographic data. It didn’t make sense, so we asked them to redo the test. In the second test, the pattern disappeared.

When gathering web or mobile analytics, survey data, or any kind of quantitative measure, your biggest asset is knowing the right questions to ask. Start with a theory that you want to prove or disprove, or use analytics and surveys to steer more detailed user research. For example, you could use survey data to get an overall preference for a feature, then use more qualitative methods like user interviews to understand that preference in context and detail.
CLIENTS SOMETIMES ASK FOR MORE DATA to prove a data point is “statistically significant,” or believe they have a meaningful result because something is “statistically significant.”

Research methods such as surveys demand a certain amount of rigor to prove statistical significance. However, when it comes to testing how users actually interact with the system, statistical significance is far from a sufficient criterion for meaningful or proper research on its own.

On the other hand, you can’t draw conclusions from interviewing only two or three users. So how many users do you need to speak with or observe? It depends on the range of user groups your product is targeted to, the scope of product interactions you want to observe, how the results will be used, and how many rounds of research you conduct.

If you want to test a software application for usability and opportunities to improve the design, focus on one or two primary user groups, and five to six users each time will detect most usability issues.

Jakob Nielsen has shown that approximately five to six users will likely detect 80% of usability problems for a specific use of a product. Keep in mind this “formula only holds for comparable users who will be using the product in fairly similar ways.” (Jakob Neilson, Useit.com. 2000, 03, 19. Why You Only Need to Test with 5 Users)

Laura Faulkner tested Nielsen’s theory on a web-based employee timesheet (Faulkner, L. Beyond the Five User Assumption: Benefits of Increased Sample Sizes in Usability Testing. Behavior Research Methods, Instruments and Computers (Volume 35(3) 379-383)). She ran tests with a group of 65 users, then selected random groups of five participants and compared the percentage of issues each group of five detected compared to issues detected by all participants. Nielsen’s 80% rule held, but variability meant some groups of five
identified as few as 55% of the issues. By increasing the number of users to ten, groups found an average of 95% of issues.

Ultimately, the number of users you need to draw on depends on what type of information you want to get from your users. For a product benchmark study, you’ll need to run with a much greater number of users than a usability test, and you have to make sure you test across all user groups. For customer interviews, the number can vary significantly.

Many important research goals can be satisfied with a limited amount of research from a limited number of participants. We encourage clients to seek advice from a User Experience professional with a background in experimental practices to help determine the extent you need for your product goals.
Mistake #5

Jumping into the Solution Space Too Quickly

GETTING GOOD USER FEEDBACK IS A SCIENCE. But there’s one thing that you can do to improve the quality of feedback you’re getting from customers right now. Stay in what we call the “problem space” longer.

What that means is to take time, when talking to real-world users, to think about and discuss the different facets of a problem, rather than trying to solve it as quickly as possible.

Although the theory sounds obvious, the practice is remarkably hard. Both interviewers and interviewees naturally start brainstorming solutions right away.

Product managers who ask the questions should remember Steven Covey’s 5th habit of highly effective people, “Seek first to understand, then to be understood.” Sure, you’re an expert in the product and you’ve probably heard what the user has to say hundreds of times before. You may have even thought it yourself. But if you automatically tune out or assume you understand what the user is saying, you may not discover the true root cause of the problem.

As the interviewer, you should also get the user to stay in the problem space longer. Each time they begin to express their opinions in the form of a solution (“Just add another feature that works like this…”), your job should be to ask why. A product executive we spoke to recently referred to this process as “peeling the onion.”

Designing a formal interview protocol in advance helps tremendously. These interview protocols often ask the same questions in different ways, to keep the user reflecting on the problem until you get to the root cause.

This is also why organizations increasingly bring in outside help from UX research specialists. Professionals experienced in interview techniques explore problems before solutions the same way a clinical psychologist trains to interview patients. UX researchers don’t have ties to any particular solution, unlike...
a manager or developer who might — even unconsciously — want one way forward over another.

The same principle applies internally. The next time the sales team comes to you saying that customers absolutely need a mobile version of the product by Q2, stop and ask why. Set up customer interviews and get to the core of the problem so you have a full understanding of the issue. Observe users in their natural habitat. You may even find insights and untapped opportunities that lead to innovations for your next product.
Mistake #6

Confusing Focus Groups and Usability Testing

**FOCUS GROUPS ARE BEST USED IN THE EARLY STAGES** of developing user requirements to provide rich information on the opinions and attitudes of the target audience. They are suited to an exploratory situation — getting a feel for the range of opinions, understanding the reasons underlying preferences, developing a basic list of user requirements — information that provides direction in the early stages of development.

Focus groups are not a good source of behavioral data. What people say they will do and what they actually do are two different things.

Our intentions and actions often vary because we can’t predict contextual factors that may alter our behavior in a given situation. We aren’t even fully aware of all our behaviors. Some everyday actions are so commonplace that we can’t remember them. For instance, if you ask someone what features they use on their telephone and how often, and then observe them using the phone, the results are usually staggeringly different.

The more complex the behavior pattern you ask people to predict or recall, the worse people are at predicting or recalling. So how do you get around this problem? That’s where observational techniques like user testing come in.

User testing provides behavioral information by assessing users’ performance on pre-determined tasks critical to the successful use of an application or website. Testing collects performance measures such as time to complete task, number of errors, and success rate, along with ease of use ratings for a series of tasks within a typical usage scenario. Using this information, you can identify the roadblocks and lesser obstacles to the successful use of the application or website.
FOCUS GROUPS | USER TESTING
--- | ---
METHOD | Group discussion with 6-8 participants | Individual testing of 6-8 participants
WHEN TO USE | Pre-development, early in program | Throughout development (concept, detailed design, verification)
KEY MOTIVATION | Find out what the customers want | Find out how well end users can use the functionality the product provides
KEY FOCUS | Find out what motivates potential customers to buy the product | Find out what will make the product easy for end users to learn and use
OUTPUT | Opinions, attitudes, preferences | Behavioral measures and preferences based on usage, ease of use ratings

The bottom line is that you use focus groups in a development program to help define user requirements. Once development has begun, you can employ user testing at set milestones — as early as a wireframe design — to test that the product being designed and built matches the original expectations from the focus group.

Both research methods play a useful role in supporting design decisions in any development project. To get the most value for your research budget, it’s important to know which method to use and for what purpose.
Mistake #7

Misuse of A/B Testing

Sometimes design questions end up in an internal debate between proposed solutions. The lead architect is convinced the features should look or function one way, and the product manager has a different theory.

A situation like this can be a good candidate for A/B testing, a fast form of user research where you launch a few designs to different user groups, perform some testing, and compare the results.

A/B testing measures which of several designs produces the most conversions, fewest clicks, fastest time, most intense emotional response, or whatever metric you decide to measure. Online testing tools like verifyapp.com take advantage of crowdsourcing to make this process even more convenient.

But you need to be aware of the drawbacks of A/B testing. First, you have to rely on your best guess as to the real reason why Design A performed better than Design B. You don’t get any feedback on whether or not the user “gets” the system. (This makes it hard to stay in the problem space.)

Also, you can inadvertently commit yourself to a non-optimal solution. Incremental A/B testing finds the solution that, relative to other presented solutions, produced the best results. However, because you always test one solution against the others, you run the risk of getting stuck with the best you’ve got, not the best possible solution.

A/B testing should be used as a quick cheat, a complement to other forms of research like the contextual interviews, concept walkthroughs, and usability testing that we described earlier.
Mistake #8

Not Structuring or Prioritizing the Insights

YOU’VE JUST COME BACK FROM A RUSH of customer visits across the nation. You have pages and pages of notes, and now you need to figure out how to make sense of it all — for yourself, and for the rest of the development organization. What’s more, you need to balance all of those ideas with your own internal stakeholders’ requests.

Giving the raw data to the development team is a recipe for disaster, so paring down the information into well-defined requirements usually falls on the product manager or business analyst’s shoulders.

Structuring this data and these requirements using formal methods is the key. We recommend applying, at a minimum, the following four stage pattern:

Identify User Groups – Identifying who will use the product informs feature and design decisions. These can be illustrated through the use of User Personas, which often take the form of fictional characters representing the different user types that have similar needs, goals and behaviors when using a product.

Identify Tasks – Identifying the operations that user groups want to perform lets you determine what features to prioritize and helps the UI designer determine the overall information architecture.

Clarifying the Context of Use – Knowing the circumstances, the common physical and organizational environments, under which users will use the product and features will yield the information you need to make it easy for users to accomplish tasks quickly.

Developing Usage Scenarios – Organizing requirements in the form of common scenarios for the use of the application is a clear and intuitive way of illustrating how features will be used beyond just presenting a set of requirements. It is even more powerful when you combine these user scenarios with the user personas, painting a real-life picture of users and their tasks.

As a general rule, if you can organize and classify around user types, context and usage scenarios, and tasks, you’re on the right path.
Mistake #9

Product Requirements Should Be More Than Just Words

WE’VE ALL READ BULLET POINT LISTS of requirements that start with, “The product shall…”

In On Reqs and Specs: The Roles and Behaviors for Effective Product Definition, Steve Johnson and John Milburn make the case for abandoning this document format. At a minimum, they suggest writing requirements in the form, “[Persona] has [problem] with [frequency].” Johnson and Milburn say this “forces product managers to explore the problem, not the solution, and helps the design team understand the context of the problem.”

We agree. Traditional requirements documents tend to lose the voice of the customer and often don’t illustrate the context explored in the original requirement-gathering sessions.

Even with clearer requirements, we still find that there can be a vast difference between the product manager’s intent, the designer’s interpretation, and the final product created by the development team.

A picture is worth a thousand words in product requirements. We highly recommend product managers work with a designer at the requirements stage to illustrate the concepts, so that every team member has a shared vision of how the application will look and behave.

These illustrations can take the form of low-fidelity wireframes of usage scenarios like the ones we discussed in the previous section.

This isn’t a substitute for design specifications. But full specs take time to develop. During that time, the development team is either idle or starts development, maybe going down the wrong path.
Illustrating usage scenarios in the initial requirements document provides a comprehensive way of getting a common visual understanding of the product direction fast.

For more details on how Product Managers, UX, and Software Development teams can work together on requirements, specifications, and iterations throughout the cycle, see our paper *How To Get Amazing Software Out The Door Fast.*
USER RESEARCH IS A TRUE SCIENCE, not a matter of opinion and interpretation. If the person doing research doesn’t have the right background, you could end up with the wrong conclusions.

To get a sense of whether your product teams have the right competencies, here are a few things we like to see in the people gathering real-world data from customers:

• Do they have a human-computer interface background, or a general background in experimental practices? Do they mention things like getting user feedback, doing user research, and defining personas?

• Look for a progression of titles like “user researcher,” “User Experience designer,” or “product manager” (though the last title can mean many different things such as a marketing, sales, or technical background, depending on the organization).

• Beware of too much emphasis on technical skill. If the candidate spent a lot of time as a developer or learning programming languages, it’s possible they haven’t had enough time to develop user research skills. At some companies, developers work directly with clients to gather requirements. Sometimes this works, but it doesn’t necessarily mean the individual has the formal training to consistently identify customer needs and synthesize them into a meaningful action plan.

• It’s a plus if the candidate worked at a company that takes product management/User Experience design seriously — Apple, Yahoo, or Microsoft, for instance. It indicates they’ve probably had great mentors and lots of experience working in a disciplined research process.

In talking with candidates, use open-ended questions to really understand how they think through gathering requirements, talking to users, and planning
designs. Ask them to take you through, in detail, how they have done this in the past. Does it match all that we have described above?

One of the most successful ways we've seen research gathered, interpreted, and disseminated to organizations is by pairing a product manager with a part-time UX researcher. The product manager is an expert on the product and on the market. The UX researcher helps with research strategy and execution, and pairs the insights with the product manager's expertise, creating a full set of requirements. The UX researcher can also help bridge the gap between product manager and designer to illustrate the requirements via wireframes.

This process is detailed in our paper User Experience Design — Helping Product Managers Sleep at Night.

**Does User Research Make Better Software?**

Focusing on user research to create better software might be trendy right now due to the success of companies who have changed the software landscape. But simply interviewing customers doesn't automatically translate into more compelling products. It isn't enough to ask your customers what they want, then implement the solutions they ask for. Rushing to conclusions without understanding the full context of how people use your software might even take your product in the wrong direction.

The good news is that if you can avoid the pitfalls we've described, you can collect the right data, interpret it with scientific rigor, and provide it to development teams in a way they can use. It is the successful combination of these activities that will take your software to the next level with customers.
Contact Us

For questions or comments about this white paper, or for more information on a consultation, please contact:

Didier Thizy
Director, Marketing
didier@macadamian.com
+ 1 877-779-6336 x136

About Macadamian

Macadamian is a global UI design and software innovation studio that provides a complete range of highest quality usability, design, and software engineering services to industry leaders across North America. Our experience, and proven ability to work seamlessly with product management executives and software teams is why companies turn to Macadamian to develop product strategies, design compelling user experiences, and build quality software.

Whether you’re a small start-up or a corporate giant, we can help you transform ideas into market-ready features or products that will stand out from your competition and delight customers.

Additional information can be found at www.macadamian.com.