



By [Heidi Adkisson](#)

For years I'd heard references to the Winchester House—a mansion built over many years by an eccentric heiress without any master build plan. It is well-known for its oddities such as doors and stairs that go nowhere and windows overlooking other rooms. A couple of years ago on a Blink project I wound up staying less than a mile away from the famed mansion. Of course I had to go!

Why the interest in an old mansion? In part because I think it serves as an apt metaphor to the situation a number of our clients find themselves in. They have built a software system over time, gradually, adding features often in direct response to users' requests. This goes on for many years until the system is a labyrinth of features and functions that are difficult for users to navigate. In fact, sometimes people in the organization who developed the system don't even know every nook and cranny. Often built on aging technology, the system eventually becomes too costly to manage. The situation usually forces a complete re-architecting of the solution.

The cautions against such a "radical redesign" are well-founded, but the reality is that in some situations it is the best option—at least from a technical and user experience standpoint. The problem is these systems often have a loyal embedded base of users whose business processes may revolve around the system's quirks and limitations. It is the devil they know—and regardless of how improved the new system is—the change will be traumatic.

We are often asked for advice on rolling out a radically redesigned system—and getting existing users on-board. Assuming this is a web-based solution, you could, in theory, just "flip the switch" to replace the old system with the new system. Your users login one day and everything has changed. Your call center is flooded with angry and confused users, including your most loyal and long-standing customers.

At the other end of the spectrum is fully supporting both versions indefinitely. This is an approach exemplified by Basecamp. When they recently released Version 3—a complete re-write—they emphasized their [commitment to supporting all previous versions](#) "until the end of the Internet."

Most situations, however, call for an approach between these two extremes.

To start with, we recommend, where possible, engaging some of your most important or loyal customers in providing early feedback on the new design. We always advise some form of early prototype to test the new design with real users. Visiting your high-profile users with a prototype gives visibility to the upcoming changes, helps users feel involved, and of course helps ensure the design meets users' needs in the most effective ways possible.

In most situations, however, it's not possible to visit every user for feedback. For a broader audience, you can keep them apprised of the work in progress. For our clients, we suggest they highlight they've hired a user experience firm—and explain how the user-centered approach is guiding development efforts.

A beta program is also an important component to mitigate risk with a radical redesign. Normally, a smaller set of interested, motivated beta users will yield the best feedback. A beta program will identify technical and user experience problems under real-world conditions.

Planning the transition

In terms of planning the release, you can of course start your new customers on the new version. For your existing customers, you'll need to map out a transition strategy.

The specifics of the transition strategy are often driven by the type of system. For example, with some workflow systems both the new and old systems can run in parallel, with new workflow items being processed by the new system and older items by the old system. Eventually, the old items "age off" and the transition to the new system is complete. This has the advantage of not having to convert existing data to the new system. It also makes for a more gradual transition as the new system becomes increasingly more active.

It is also possible to run both systems in complete parallel - meaning data and processes in the old system are mirrored for some period of time in the new system. This has the advantage of minimizing mostly technical risks associated with the new system. If something goes terribly awry in the new system, users can revert to using the old system. However, the costs associated with running parallel systems are usually prohibitive.

More commonly, users will need to plan for a system "cutover" - a no-turning-back transition from the old system to the new one. In a sense this is the "flip the switch" approach, however, you can mitigate the pain associated with this by both preparing users for the transition and allowing them a grace period to make the transition with timing that is best for them. However, this gradual sunsetting of the old version comes with the risk that some customers will be laggards. You could of course force the transition at the end of the grace period, but we find that it usually makes better business sense to work with customers on the timing. This usually means extending the grace period, so that is a factor to consider and plan for.

In a perfect world, there would be no need for the radical redesign. Changes would be rolled out gradually and small amounts of refactoring would ensure that new features are appropriately integrated into the user experience. The reality is that our clients can find themselves having to negotiate through a radical redesign. With appropriate planning, this can be done successfully, from both the perspective of the client's business and their customer base.