blink

By [Kelly Franznick](#)

We have a few methods at Blink to pencil in future functionality, but still optimize your users' initial experience.

During an interaction design project, our wireframes are typically reviewed by clients both for what can be done and what should be done. One of our greatest challenges is keeping track of functionality that our clients know is crucial for a good user experience (the "ideal" design), but simply cannot be implemented yet. Future functionality that is removed from the wireframes can easily be lost in an old version of a document, and the issue may be revisited all over again in the next phase.

We have a few methods we utilize at Blink to pencil in future functionality within our wireframes, and help you keep your "eye on the prize." However, in applying these techniques, it is important to conduct user testing to understand your customers' initial experience with your product, and also validate design decisions made for the final product.

The approach we prefer for a design project is to create wireframes that reflect the ideal state, but visually label the features or content that would only be introduced in future deployments.

When we first review and analyze a system, our first impressions are the most important to capture before we become 'tainted' by familiarity. Designing for the ideal state of the system works best at this stage, since the ideas start flowing immediately as we work through realistic scenarios, site flows and high level wireframes that illustrate the best user experience.

The risk with restricting design solutions at this point is that the lines can begin to blur between whether things were changed to improve the user experience or to meet technical or resource requirements. Once an interim solution is launched, a long-term vision that has not been well-documented can be lost as new issues or immediate requirements overshadow it. Our approach

allows us to deliver a roadmap for the best user experience, and then pull back from there to a short-term solution.

For projects that allow only enough time for designing the short-term solution, another approach we use is to create detailed wireframes for the nearest launch, but document future-phase items. With this approach, wireframe page annotations can include future elements for that specific page, future pages may be indicated in the flow map, or future scenarios or page templates may be saved in a separate section. Features, content or interface elements that are recognized as necessary improvements but are out of scope are documented. While this documentation can take some extra time, the payoff is great when it comes time to design for the next release.

When it comes to testing the prototype, our clients often ask us, "Should we test the ideal functionality or only the functionality that will be deployed?" A first deployment that has usability issues can have a negative impact on a company's user base. Since those first impressions are so important, we have some specific guidance for testing.

- Test what you will deploy in the near-term. We recommend testing what will actually be deployed in the near-term to reveal where user errors occur, and any major usability issues that can be resolved easily prior to that deployment.
- Learn from that deployment. Find out if goals are met, if other issues arise, or if new business requirements change those goals. Confirm that solutions in the "final" design address any remaining issues, or if other refinements still need to be made.
- Make refinements to the "final" design based on what you learn. Modify the "final" wireframes accordingly, and create any additional detailed wireframe pages that are necessary to include all the functionality for the upcoming deployment.
- Test the "final" design prior to deployment. Yes, test again. Test a prototype of what will be released in the next deployment, whether it is an incremental step or the "final" deployment. Validate the design decisions that were made, and make any final refinements needed. If this is the "final" design, then after deployment, re-assess that design from a user-experience perspective, asking, "How can it delight our users even more?" Testing again after deployment can help you learn how your users' expectations evolve as they continue to use your product.

We find that our "now-and-later" approaches help capture ideal solutions while grounding wireframes in the kind reality that clients need for upcoming launches.