TAKING AIM

The Power of User Experience Goals

BY JOHN SCHRAG

A user experience goal is a choice made by your product team about what kind of experience you want your users to have with your product or service. You use these choices to *measure* and *direct* the design of your product. Goals let us know when our tasks are complete, so that we can move on to something else. They stop us from obsessing over the wrong details and help us direct our energies to what is important. Goals tell us what to measure, and what can be ignored.

Goals need to have teeth. If you test your design and find your goals are not met, are you willing to change your design? If you aren't willing to change the design, then you didn't have goals—you just had a wish list.

What Makes a Good Goal?

Good goals are written down and shared. When people on your team don't have explicit goals, they will automatically work towards implicit ones—their own unstated assumptions.

Good goals have measurable or observable success criteria. There is no point in having a goal that you can't validate. You have no way of knowing if decisions you make are helping you achieve the goal or making the situation worse.

Good goals are specific. Specific goals have numbers, limits, context, and conditions. Notice the difference between "Feature X will be findable and usable," and "Within one minute of being given task Y, novice test participants will be able to locate Feature X and recognize that it is the feature they need to complete the task. They will be able to complete the task within three additional minutes using their own data without having to consult the online help."

Good goals are realistic and achievable. If you set your goals impossibly high, they

MICHEL ZAYADINE

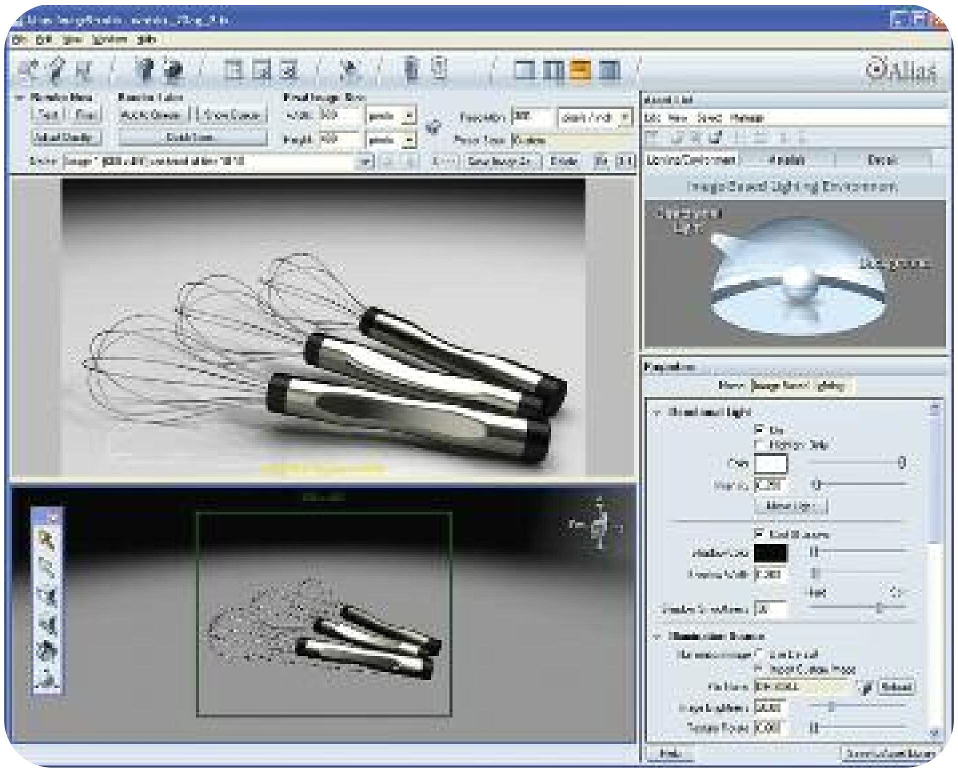
won't help you in your day-to-day work. Goals should help you figure out where to focus your efforts and what things are good enough to leave alone.

If you are given unrealistic goals by management, you can often turn them into realistic goals by asking how badly a goal would have to fail before management would be willing to delay shipment of the product.

Good goals are practical. Goals should be related to the required uses of your product or service in the context of its use. This flows from your user research and requirements gathering. For example, if your hardware product will be used outside in a cold climate, you'll need goals relating to the screen not freezing up or slowing down under typical temperatures, and the interface being operable by people wearing thick gloves or mittens.

Good goals do not depend on any particular UI design. Goals are about users achieving things they need to achieve and having the experience you want them to have. You should be able to completely change your UI design approach and still have the same goals.

Good goals support the business. You should have goals at many levels: business goals, product goals, release goals, and feature goals. Each level of goals should flow from and support the goals above it. For example, we'd all like our designs to be easy-to-learn, easy-to-use, and efficient, but if your users will all receive days of intense training before using the software, then ease-of-learning is much less important than efficiency.


A 3D rendering

Goals are about users achieving things they need to achieve and having the experience you want them to have.

Case Study: Alias ImageStudio

(This case study describes work done at Alias, which has since been acquired by Autodesk.)

In 3D computer graphics, creating a photorealistic rendering has traditionally been a difficult and technical task. It required good software, artistic talent, and a fairly deep understanding of arcane terminology, tricks, and quirks of different software and rendering algorithms.

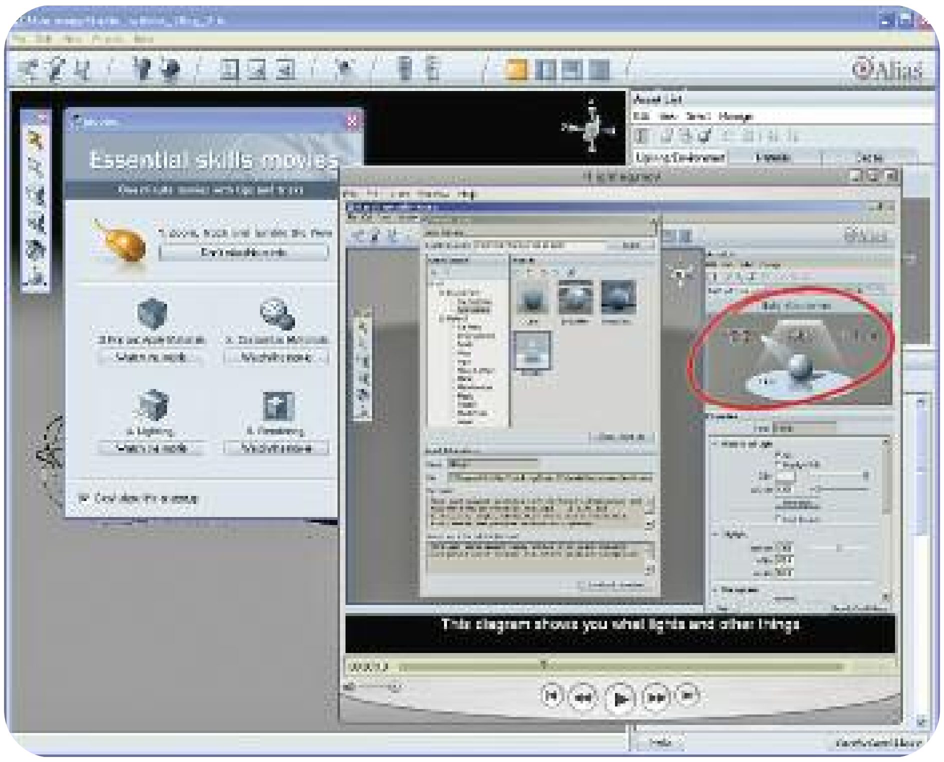
My company had, for many years, produced a line of software used primarily by industrial designers and car designers that allowed them to create and render computer models of their designs. We decided to create a new program around the goal of "easy-to-use rendering." Users would be able to take 3D models from any software and "easily" produce photorealistic renderings with it.

However, while "easy-to-use rendering" has a nice sound to it, it is not a useful goal. We needed a goal that we could use for planning, executing, and verifying a UI design. Just asking management for clarification was not a useful strategy. Instead, the UX team reviewed the original reasons for beginning the project, and asked questions like, "What would you have to see to be convinced that we had an easy-to-use rendering program?" Phrasing the question this way produced more insightful answers, such as, "Someone would be able to render on their first day," or, "Someone could finish five renderings in a day."

Based on this, we proposed our own set of goals for the application and developed them with the whole team until we had buy-in. The main learnability goal, for example, was this: "Any industrial or automotive designer with computer modeling experience will be able to produce a good rendering within their first hour of use."

In usability tests to examine this goal, we asked participants to try to make a reasonably good rendering of a model using our prototype software. After that, we gave them no more guidance, unless they became completely stuck. If a rendering was not produced within an hour, we knew the goal had failed. If a participant produced one or more renderings, we would ask what they thought of the quality of the rendering. We opted for this open-ended question because it provided rich information on the aspects of quality that were considered important to our participants.

We used this goal to prioritize the fixes the development team would make, focusing first on those problems that blocked users from completing their first good rendering quickly. However, as development neared completion, it became clear that there were some problems that we could not easily fix by tweaking the UI.



Would users find and watch the movie?

We needed a way to give our users the information they required to be successful, even though they didn't know they needed it.

Typically, these problems were brought on by the fact that that the users needed to know something (more information was required), but they didn't know they needed it. For example, there was a tool that could dramatically decrease the time it would take a user to do a common task, but the task could also be done more slowly with standard tools the user knew from other programs. Similarly, participants would frequently miss simple, direct workflows because they had already internalized the complex and arcane workflows of other rendering software.

We needed a way to give our users the information they required to be successful, even though they didn't know they needed it. From years of usability testing, we knew that most industrial designers avoid online help and documentation if at all possible, and almost never look at a manual before starting to use software.

Instead, we decided to create "Learning Movies," a series of one-minute clips containing missing concepts identified by our earlier testing. (For more on Learning Movies, see Miller and Sy's "Using Movies to Make Complex Software More Approachable," presented at UPA 2003). To ensure our movies were made well, we set the following usability goals for them:

1. Users would choose to watch the movies within the first fifteen minutes of using the software.
2. Users would retain the concepts illustrated in the movies and be able to apply them.

Types of User Experience Goals

While not a comprehensive list of user experience goals, you can use this list to help you consider different aspects of the user experience that you may want to have goals for.

Discoverability: **How quickly do users need to find a feature? How many false finds are allowed? Is it okay if they have to look at the help to find it?**

Learnability: **How quickly do users need to figure out how to use a feature? Under what training conditions? What kinds of tasks do they need to complete to convince you they've learned it? How many errors are allowed? Is it okay if they use the online help?**

Error avoidance: **What kinds of errors should users never encounter? Or, how many times is it reasonable for them to make an error during a particular task?**

Error recovery: **How quickly do users need to recognize they are in an error state? How quickly do they have to be able to get out of it? How much work is it reasonable for them to lose getting out of the error state?**

Emotional response: **How should users feel about the experience of using your product? How *shouldn't* they feel?**

Task completion: **What tasks should users be able to complete? In how much time and with what size of data? What's the greatest number of errors they should reasonably make in completing the task?**

Preference: **When comparing different products or interfaces, should users strongly prefer one to another? Under what circumstances, which types of users, and for what tasks?**

Retention: **After learning the system, how much of the training should users retain? After how much time of non-use?**

Efficiency: **How quickly can users process a fixed amount of work? Of what scope? What file sizes? Under what circumstances (interruptions, noise, factory floor, etc.)?**

Engagement: **Do users need to be able to get into, and stay in, "flow" while using your software?**

Responsiveness: **Does the task require an instantaneous response, a real-time "feel," or is a UI lag acceptable? What types of feedback are required? Does the task need to be interruptible?**

"WHERE'S WALDEN?"

3. Users would be able to find and get back to the movies if they so desired.

Measuring the success of the first goal was easy—we just needed a stopwatch. The second goal was trickier; we decided that we would measure this goal by observation. We had a list of the concepts in each movie, and the user was deemed to have retained a concept if:

a. They applied the concept (used the software correctly for that problem), or
b. When requiring the concept, they did not recall what it was, but did recall that it was in one of the movies and went to find it there.

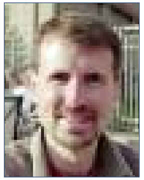
The test protocol was similar to the earlier testing. When the program started up, it offered the user a choice of movies to watch. If participants chose not to watch any of the movies in the first fifteen minutes, we would count that as a failure of the first goal. Then we would intervene and ask them to find and watch the movies before continuing so that we could collect data on goals two and three. If the participant did watch the movies at the start but never went back to them, at the end of the test we would ask them to locate the movies again so that we had data on goal three.

Our first problem was getting participants to watch the movies at all. Participants assumed the movies would be "marketing crap" rather than something useful. We redesigned the movies' dialog, changing the language and presentation. This improved the watching rate.

Our second problem was retention. We had tried to keep the movies short and to the point, but even, so some content was getting lost among all the information. Reflecting back on the first experience goals, we ruthlessly pared the movies down to contain only those concepts that related directly to the problems we had observed blocking users from creating their first "good" rendering. This simplification significantly increased retention of the remaining concepts.

When we finally had six sequential test runs where all the goals were met, we knew we were done and were able to lock down the UI with confidence. The software was not perfect, and there were still some usability problems, but our goals helped the whole team achieve a specific, measurable success. UX

ABOUT THE AUTHOR



John Schrag has been designing user experiences for the last twenty years, most of which was spent at Alias, acquired by Autodesk in 2006. Specializing in interface design for artistic users with open-ended tasks, John has helped create 3D graphics tools for graphic artists, typographers, industrial designers, automotive stylists, animators, filmmakers, and architects.