



### #63 Curse of knowledge

We need to understand how users relate to the removed functionality. At the same time, our personal opinion does not matter. We should double-check with the team any assumptions about how users will interpret the removal of functionality. We need the same knowledge for proper communication about the changes ([#22 Framing effect](#)).

### #46 Functional fixedness

We should be cautious when removing the elements that are part of some workflow in the system.

### #83 Loss aversion

We should expect disappointment disproportionate to the degree of use of the removed functionality. Because of this, it is advisable to add something along with the removal of the functionality to "soften" the discomfort.

### #88 Endowment effect

Even if the user has used a certain product element once every six months, we should understand that this element is his property (for him). Removing such an element without explanation or referring to "usage statistics" can be perceived sharply negatively. Users may call our actions "dishonest" and "unfair" because we took them without their participation ([#47 Just-world fallacy](#), [#91 Reactance](#)).

### #86 Zero-risk bias

In explaining our decision, we may appeal to the desire to eliminate a specific risk.

### #3 Illusory truth effect

If we anticipate a sharply negative reaction to removing a certain functionality, we can take action to mitigate this reaction. So, a few months before the action, we can highlight the imperfection of the component that we are going to remove. The idea is that on the day of removal of the functionality, users can refer to some "constantly appearing" materials associated with the component that has been removed. Thus, it will be easier for them to rationalize what has happened.



## #5 Context effect

In some cases, we can first move a component to another part of the product, so that it falls out of context. Then, analyze users' reactions, and only after that remove the component as "obviously interfering."

## #16 Self-reference effect

We need to be very careful when working with product components in which people "see themselves." For example, it may seem to us that the "Premium" prefix next to the users' nicknames is unnecessary and does not carry any semantic meaning. However, due to the effect of self-reference, if we remove this prefix, users can begin to leave without telling anything. Any element of the product that feeds the user's ego is critical.

## #19 Conservatism (belief revision)

If our users are conservative due to their ideology, age, or something else, we should keep the number of changes to a minimum.

## #21 Distinction bias

Sometimes users find it difficult to assess the impact of changes. Especially if these changes affect several different components of the product. It might be a good idea to visualize the changes in one image in a "Before - After" format.

## #27 Post-purchase rationalization

If we remove a component that has been actively used in our marketing or post-purchase materials, users may have the feeling that they were "cheated."

## #74 Dunning-Kruger effect

In some cases, users may dislike the changes without even understanding their meaning. This is especially true in the B2B sector, where it can often be convenient for the user to hide his incompetence in front of his management, referring to "extreme changes" in the product.