#46 Functional fixedness
Any changes to the most popular workflows will cause a sharp reaction. Therefore, we should gradually prepare users for the changes so that it does not come as a surprise (**#58 Normality bias**).

#79 Hyperbolic discounting
If our competitors launch campaigns with "instant bonuses" at the right moment, we risk losing some of our users. This is one of the reasons why you sometimes should not talk about the dates of key releases.

#80 Appeal to novelty
If we explain the need for updates by referring to the "novelty" factor, many may not like it, and they will leave. Even if the desire for change came directly from the users, they may still be unhappy.

#5 Context effect
If our changes are fully consistent with the product's context, then it would be easy to break them up into small parts and announce them in stages. In this case, we will minimize the discomfort of changes for users.

#19 Conservatism (belief revision)
If the product changes relate to its ideological concepts or used terminology, we have to make sure that the changes correlate with users' beliefs. The simplest example: if a group of open-source software developers starts selling their software licenses it will go against their audience's values.

#21 Distinction bias
Users may want to see the difference in "before" and "after" views. If such transparency is appropriate in our product's logic, we can visually show the upcoming changes in the form of a table "before" - "after". If we see risks in displaying information in this form, we can show changes in different areas, making comparison difficult.

#28 Selective perception

We may publish several articles related to the upcoming changes in order to analyze the reaction of our audience. Based on the reaction, we may find out what exactly worries our users the most. We can use this information to create mechanisms to mitigate negative reactions to future changes.

#73 Hard-easy effect

We need to create extremely simple instructions for using the new functionality. Anything that we can simplify in the interface should be simplified. If the product's specificity allows, we can create two visual representations of the functionality (Standard and Advanced view). This may be a temporary solution until users are familiar with the basic concepts.

#88 Endowment effect

Even those parts of the product that users use the least, they consider as their "property." Unreasonable changes in these components can provoke a sharp reaction, which may seem not proportional to the component's popularity (**#17 Negativity bias**).

#68 Pro-innovation bias

We all intuitively understand the risks of significant changes. Usually, if we decide on them, the future of the product is reduced to a zero-sum game, where we either get everything or nothing. In such cases, our responsibility is to check our bold assumptions with our teammates to minimize the likelihood of any error. So, we can distort our assumptions due to a misunderstanding of the "world" of our audience (**#63 Curse of knowledge**, **#72 Consensus bias**), our arrogance (**#69 Overconfidence effect**, **#67 Planning fallacy**), or our blind faith, like "Through our efforts we deserve success!" (**#47 Just-world fallacy**).