



#10 Omission bias

Do not ignore requests.

#73 Hard-easy effect, #72 Consensus bias

We can explain as transparently as possible all the difficulties that we will face if we follow the users' requests.

#76 Illusion of control

We can modify user requirements and bring several similar options to a general vote. At the same time, each of the options must be feasible and consistent with our business goals. The voting itself can be held without specifying the absolute number of participants (only percentages). Then, we can appeal to the majority and implement those that won the voting. At the same time, we emphasize that we do everything following the wishes of users (**#50 Bandwagon effect**).

#46 Functional fixedness, #58 Normality bias

We can show users how the changes they requested may modify the workflow of the familiar components of the product. The idea is to show users how much they underestimate the convenience of current practices.

#79 Hyperbolic discounting

We can distract users from their requirements with a series of short-term bonus offers/rewards.

#5 Context effect

If what the users suggest falls out of the product's context, we should frankly explain it in as much detail as possible. Of course, users are not required to understand the importance of product integrity and uniformity of context, but we should at least show our concerns.

#21 Distinction bias

Users may want to see the difference in "before" and "after" views. Based on our goals we have to think if it is feasible for us to show the changes so explicit, or not.

#67 Planning fallacy

Sometimes it may be enough for us to show the time it will take to develop solutions requested by users. If we are talking about several months required for development, then we can explain that the requested features probably will not seem as useful by the time the update is launched.