

#55 Какие компоненты продукта наиболее чувствительны к изменениям?

Development

Post-SignUp



UX Core Guide by Wolf Alexanyan
<https://uxcg.io>



#91 Реактивное сопротивление

Если наши изменения продукта ограничивают какие-либо пользовательские возможности — это высокочувствительные изменения.

#88 Эффект владения

Если наши изменения затрагивают компонент системы, который пользователи считали «своей собственностью» - это высокочувствительные изменения. При этом не имеет значения как часто пользователи использовали компонент (**#83 Неприятие потерь**).

#84 Эффект IKEA

Любой компонент, который пользователи «сделали сами» посредством использования своей креативности, своего времени и энергии является высокочувствительным к изменениям.

#1 Эвристика доступности

Любой компонент, который является «свежим» в памяти пользователя становится чувствительным к изменениям.

#50 Присоединение к большинству

Любой компонент, который используется большинством пользователей проекта является чувствительным к изменениям.

#5 Эффект контекста

В любом продукте есть основной набор компонентов задающих «контекст» всего продукта. Подобные компоненты являются высокочувствительными к изменениям.

#19 Консерватизм (пересмотр системы верований)

Любые изменения затрагивающие морально-этические или идеологические ценности нашей аудитории являются критически чувствительными.

#27 Рационализация после покупки

Компоненты, использованные чтобы помочь пользователю рационализировать его решения являются чувствительными к изменениям.

#46 Функциональная закрепленность

Любые изменения в привычном процессе (workflow-e) в продукте являются высокочувствительными.

#58 Смещение нормальности

Если наши изменения привносят неопределенность в будущее компонента/продукта, это вызовет негативную пользовательскую реакцию.

#74 Эффект Даннинга-Крюгера

Это прозвучит очевидно, но наиболее чувствительными всегда являются те компоненты, которые пользователь считает наиболее важными для себя. Я написал это потому что часто, особенно в B2B сфере, пользовательская некомпетентность приводит к тому, что компания-разработчик не может менять устаревшие/сырые части системы из-за их иллюзорной «важности» для клиента.