

Інтерком про продакт менеджмент

Переклад книги [Intercom on Product Management](#)



Деякі корисні думки про труднощі створення програмних продуктів від команди Інтерком. Наш продукт допомагає компаніям бачити, хто користується їхнім застосунком, і легко спілкуватися з користувачами через email або прямо в застосунку.

<https://www.intercom.com/>

Ми також регулярно ділимося в нашому блозі думками про продакт-менеджмент, дизайн, стартапи та бізнес.

<https://www.intercom.com/blog/>

Ілюстрації обкладинки та в розділах: Intercom Brand Studio

Діаграма звичок у главі 1: Нір Еяль, автор книги "На гачку"

Блог: NirAndFar.com

© 2017 Intercom Inc.

Ми позбавимо вас юридичних дисклеймерів. Але будь ласка, якщо ділитиметеся книжкою чи її частинами, згадуйте авторів і залишайте відповідні посилання.

Переклад на українську [SEOBAG](#)

Зміст

[Вступ](#)

[Розділ 1. Аналіз поточного стану вашого продукту](#)

[Глава 2. Коли говорити "ні" новому функціоналу](#)

[Розділ 3. Які функції додавати в продукт](#)

[Глава 4. Що робити, щоб функцією користувалися](#)

Вступ

В Інтерком ми віримо, що в кожному стартапі основний фокус має бути на видатному продукті. Продакт-менеджер - роль, у якій є елементи інженерії, маркетингу, досліджень та управління проєктами, і така роль є ключовою у створенні видатних продуктів. Вона вимагає безперервної уваги до цілісності продукту, об'єднуючи під єдиним баченням те, як продукт проєктується, створюється, називається, брендується і просувається на ринку.

Ця дисципліна все ще молода і слабо визначена через постійно мінливу природу розробки програмних продуктів. Багато хто вважає, що це навіть не окрема дисципліна в наші дні. У світі стартапів іноді стають продакт-менеджерами не усвідомлено, а тому що комусь треба виконувати цю роль.

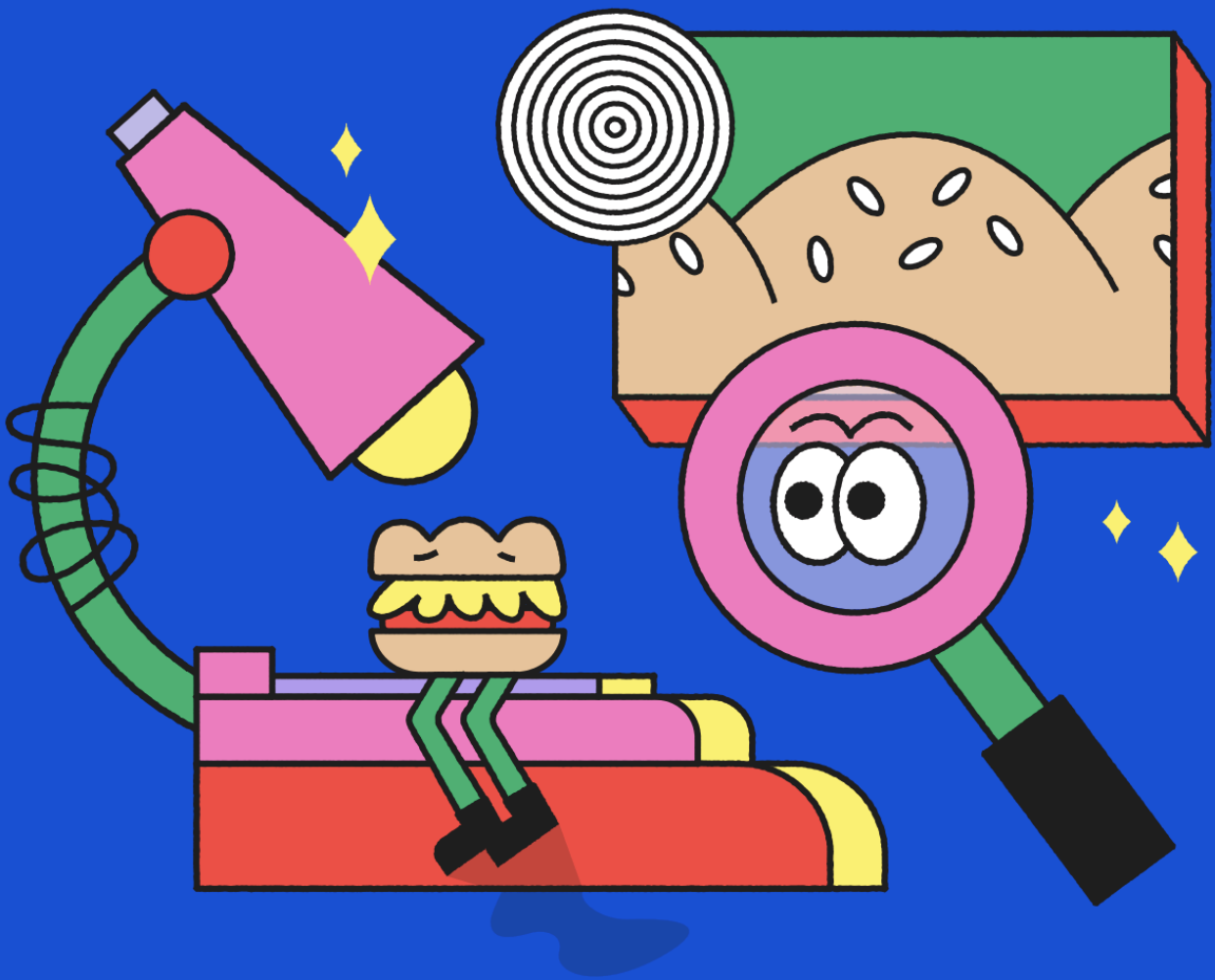
Незалежно від того, яким шляхом ви прийшли, або яка ваша поточна роль, якщо ви працюєте над продуктом - ми створили цю книжку для вас.

Ми пишемо в нашому блозі "[Всередині Інтерком](#)" про уроки та досвід розробки продукту понад шість років. Ця невелика книжка пов'язує найбільш релевантні статті з блогу про розробку та управління продуктом. Ми сподіваємося, вона допоможе прискорити ваш розвиток у цій галузі. Постійні читачі блогу помітять, що ми додали кілька розділів, щоб пов'язати ідеї різних статей.

Ми почнемо з оцінки поточного стану вашого продукту. Подивимося, які частини використовуються і які потребують поліпшення. У другому розділі ви дізнаєтеся про важкий вибір, який виникає під час додавання нового функціоналу, і чому "ні" - це найважливіше слово в словнику продакт-менеджера. Третій розділ містить чек-лист, який має пройти кожна нова функція; у цьому ж розділі обговоримо, як випускати новий функціонал. На завершення ви дізнаєтеся, як допомогти користувачам почати

використовувати нові функції. Тому що кінцева мета випуску нового коду, щоб він використовувався і був корисним.

Поїхали!



Розділ 1. Аналіз поточного стану вашого продукту

Почнемо спочатку. Ви продакт-менеджер? Засновник стартапу? СЕО? СТО? Хоч би як називалася ваша посада, припустимо, ви ухвалюєте рішення про ваш продукт. Приблизно такі: який функціонал розробляти, коли це зробити, як допомогти користувачам почати користуватися ним. У нас у всіх великі надії та мрії про наші продукти і як вони житимуть, коли потраплять у великий жорстокий світ. У вас, можливо, блокноти повні ідей, як зробити наступний Uber, Slack або Facebook. Але перш ніж зробити наступний крок на шляху до великого, потрібно підбити підсумки і подивитися, що відбувається з вашим продуктом зараз.

Що насправді користувачі роблять із вашим продуктом?

Чи всі користувачі використовують усі доступні функції? Звичайно, ні. Давайте розберемося з цим.

Коли ви плануєте розвиток продукту, і на що ваша команда витратить час, корисно запитати себе "скільки людей використовують кожну з функцій нашого продукту?". Це основи продакт-менеджменту і, можливо, потребуватиме певного часу, роблячи запити до бази, аналізу статистики або кілька секунд, щоб дізнатися це в Інтеркомі. Простий спосіб візуально уявити використання функцій - це побудувати таблицю у двох вимірах, скільки людей використовують функцію, і як часто.

У вас вийде щось схоже на таку таблицю.

Весь час	Календар	Час відстеження	Редактор Docs	★ Повідомлення
Часто	△ Чат-рум		Особливість X	Завантаження
Іноді				Особливість Y
Рідко				
	Кілька користувачів	Багато користувачів	Більшість користувачів	Усі користувачі

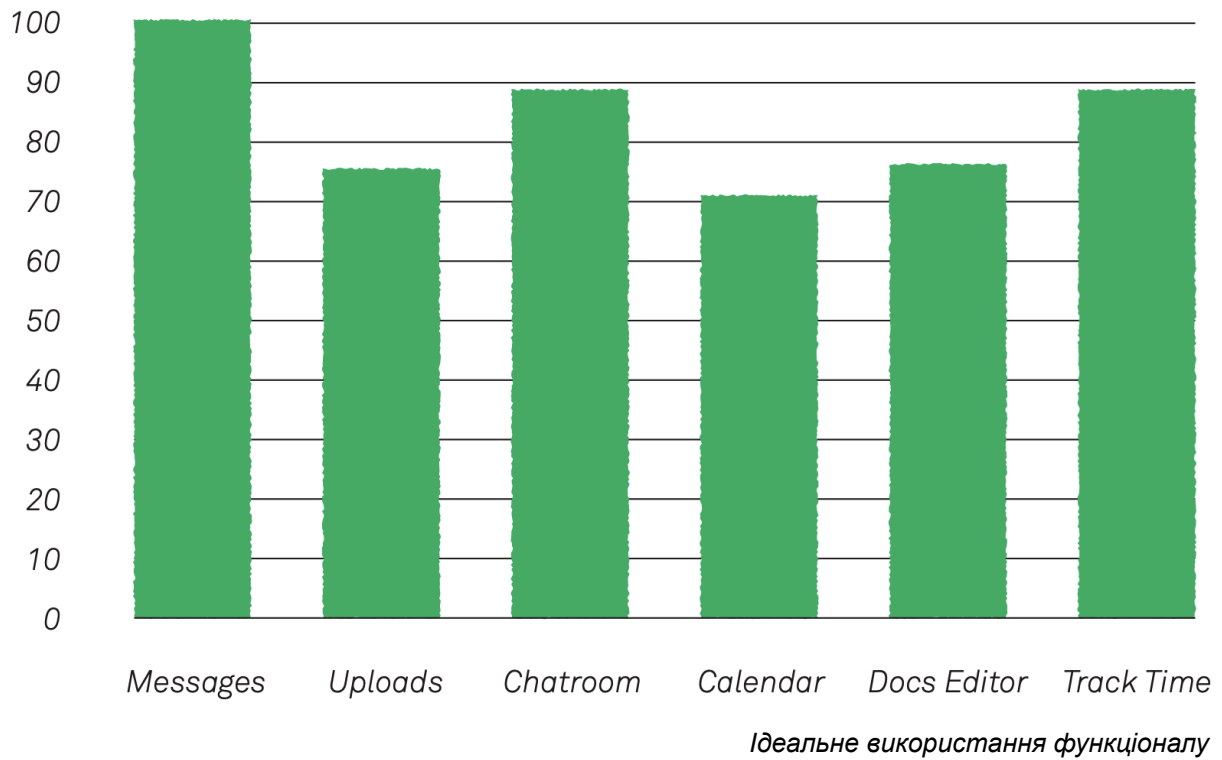
Типовий аудит функціоналу

Основна цінність вашого продукту в правому верхньому кутку, там де зірка, тому що це той функціонал, яким користуються.

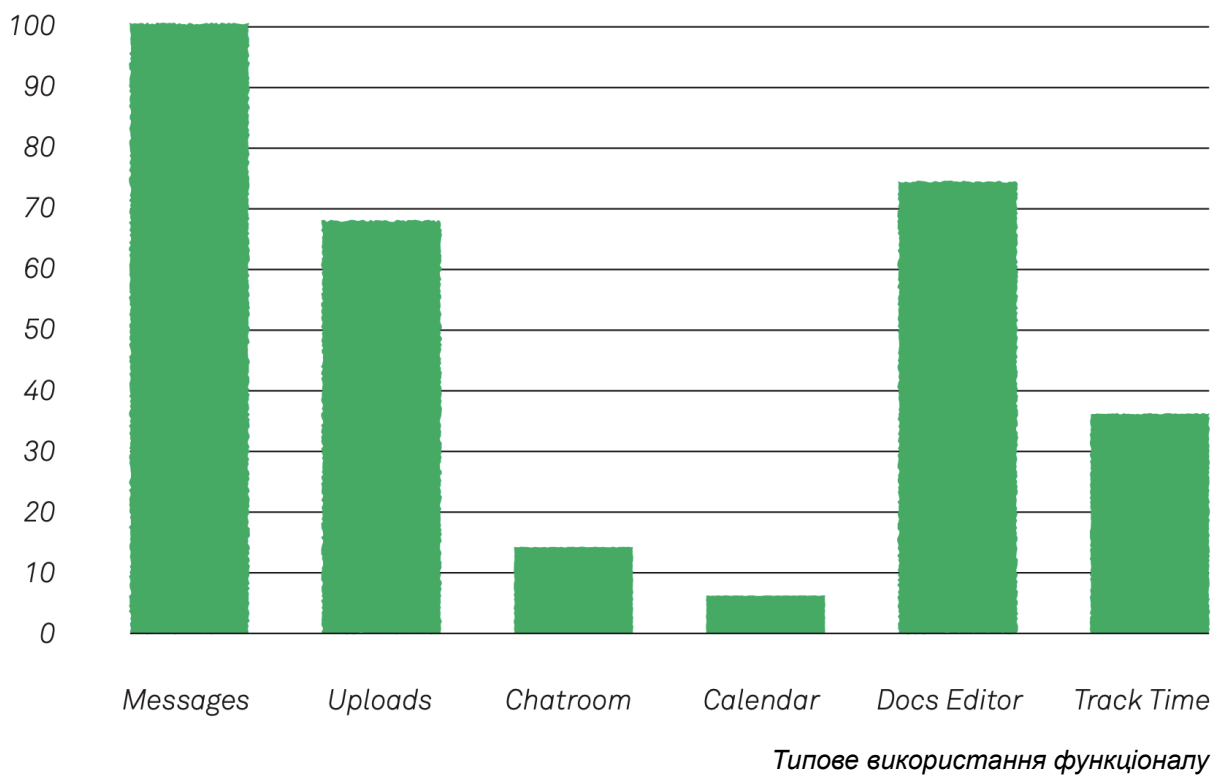
Примітка: Виключіть адміністративні функції, як-от створення облікового запису, скидання пароля тощо з цієї таблиці. Також виключіть функціонал, призначений тільки для деяких користувачів, наприклад, корпоративних користувачів. Такий функціонал потрібно аналізувати окремо.

Якщо у вас є функції в лівому верхньому кутку, це означає, що мало користувачів прийняли їх. Іншими словами, мало користувачів залежать від них, а більшість вкрай рідко використовують їх.

Ще один варіант аналізу: побачити який відсоток ваших клієнтів прийняли кожен функцію. Ви можете представити це у вигляді простої діаграми. На діаграмі нижче показано ідеальний продукт. Той самий, про який ми думаємо, коли малюємо його в Photoshop. Усі мої користувачі будуть використовувати всі функції, вірно?



Але як кожен продакт-менеджер виявляє пізніше, що сувора реальність має такий вигляд:



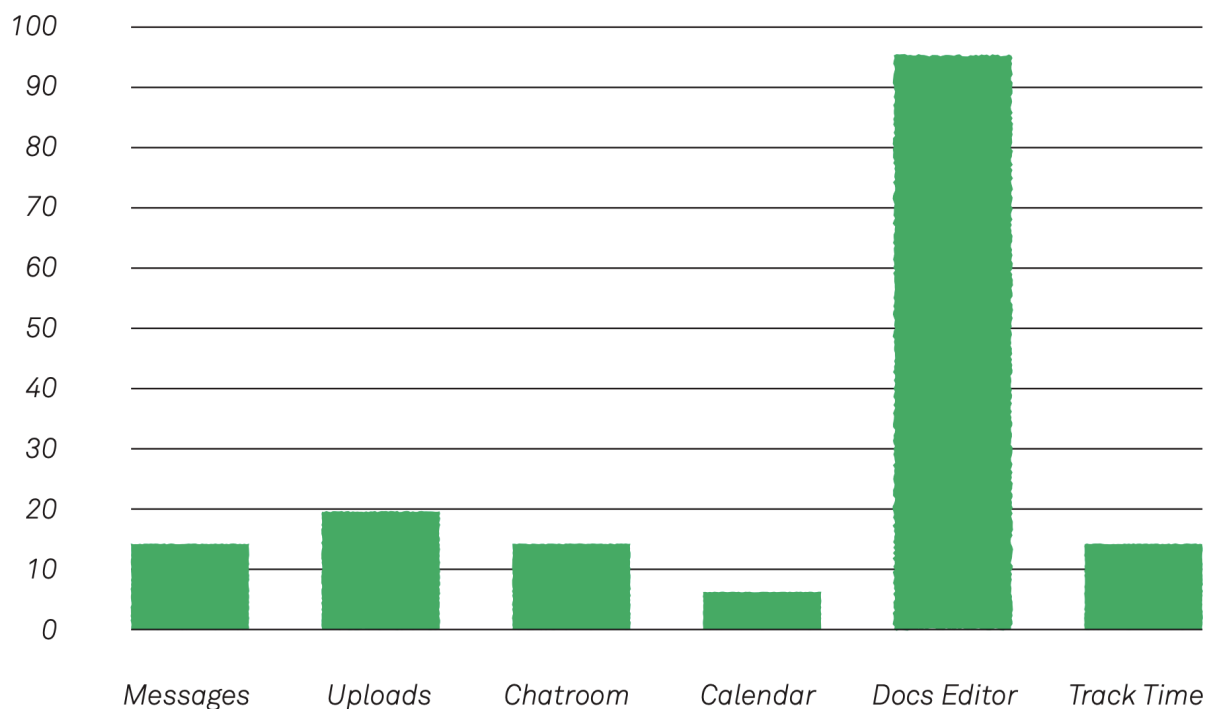
Як ви прийшли до такого? Адже ви створили продукт, у якому є чудовий месенджер, редагування файлів і документів. Ці ключові функції виділені на вашому сайті, що продає, чітко пояснені скріншотами, є у вашому інтерактивному керівництві, і кожен новий користувач любив їх.

Але успіх - поганий учитель. Ви були впевнені, що не припуститеся помилок і зможете створити набагато більше корисних функцій.

Тому ви додали чат, але він не пішов у користувачів через невдалий реліз. Потім ви додали календар, і він пішов ще гірше. Ніхто не створив більше однієї події, і про нього досі ні слова на вашому сайті, що продає. Ви додали функцію обліку часу, яка стала досить популярною, але лише у деяких користувачів.

Це ваш продукт. Ви той, хто може його виправити.

Примітка про ризик підривних інновацій



Розподіл, що вказує на ризик стати жертвою підривної інновації

Якщо ви спостерігаєте такий розподіл використання функцій, це означає, що у вас чудовий продукт для одного вузького випадку використання. А всіма іншими функціями продукту ніхто не користується. Уявіть Hangouts всередині Google+, як приклад, або додатковий примітивний редактор документів (типу *BUT*) всередині Microsoft Word.

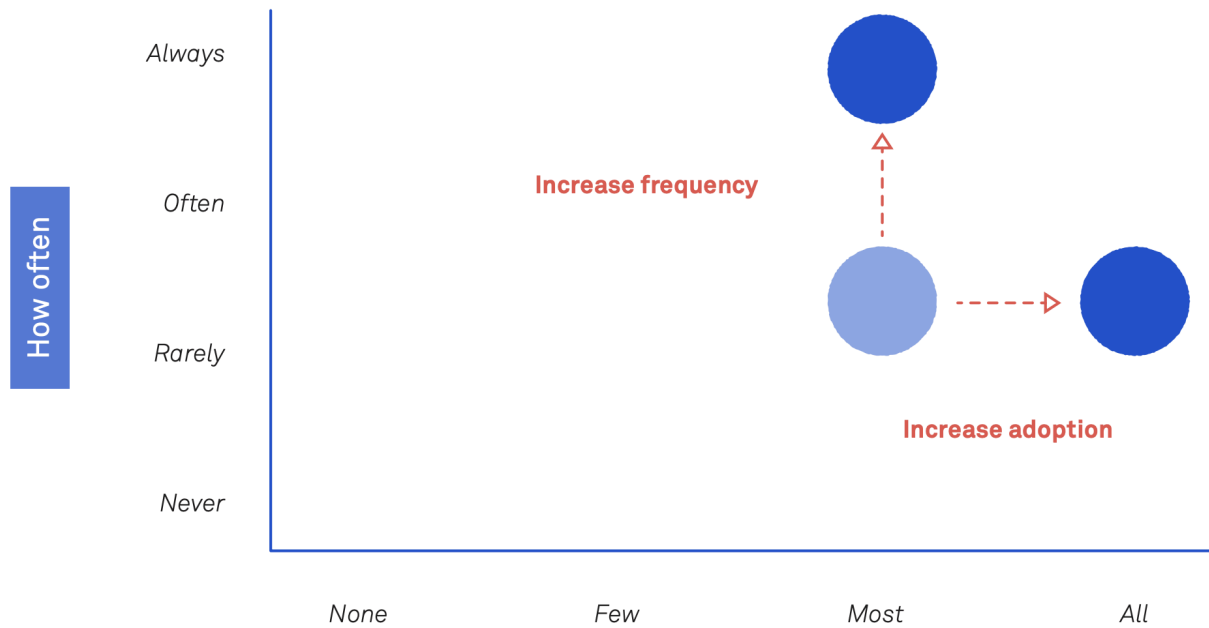
За такого розподілу у вас великий ризик стати жертвою [підривної інновації](#). Хтось може створити простий продукт із фокусом на цій єдиній функції, яка буде кращою в чомусь одному (дешевше, швидше, зі спільною роботою, зручніше, з підтримкою мобільних пристроїв тощо). І ви не зможете конкурувати, бо підтримуєте всі інші непотрібні функції.

Що робити далі з результатами аудиту продукту?

Для кожної функції, яку мало використовують, у вас є вибір із чотирьох варіантів:

- ☐ Видалити: визнати поразку і почати видалення з продукту.
- ☐ Збільшити рівень прийняття: зробити так, щоб більше людей користувалися нею.
- ☐ Збільшити частоту використання: зробити так, щоб люди користувалися нею частіше.
- ☐ Покращити: зробити її значно кращою, для тих, хто вже користується нею.

Це можна представити у вигляді діаграми:



Як поліпшити функціонал продукту

Кайдзен це філософія безперервного поліпшення. Усі онлайн компанії в пошуку продукту або ринку слідуєть Кайдзен тією чи іншою мірою, навіть якщо вони не знають про це.

Реліз нового коду не означає, що ви поліпшили щось. Аналогічно, ви можете зробити незаперечні поліпшення, але при цьому не отримати відгуку або схвалення від користувачів. Усе залежить від того, які поліпшення ви робите.

Два найпопулярніші способи поліпшити продукт - додати нові функції або поліпшити наявні. Спочатку ми розглянемо поліпшення наявного функціоналу. Потім перейдемо до створення нових функцій у наступному розділі.

Поліпшення наявного функціоналу

Поліпшити наявні функції можна такими способами:

- ❑ Зробити їх ще кращими (навмисне поліпшення).
- ❑ Змінити їх так, щоб ними користувалися частіше (збільшення частоти використання).
- ❑ Змінити так, щоб більше людей користувалися ними (збільшення прийняття).

1. Навмисне поліпшення

Це той випадок, коли ви знаєте чому клієнти користуються наявною функцією і чому вона їм подобається. Навмисне поліпшення передбачає зробити функціонал кращим саме для наявних користувачів. Наприклад, зробити функціонал швидшим, зручнішим або поліпшити зовнішній вигляд.

Використовуйте навмисне поліпшення, коли всі ваші клієнти використовують функцію і вона їм подобається. І ви бачите можливість зробити її значно кориснішою.

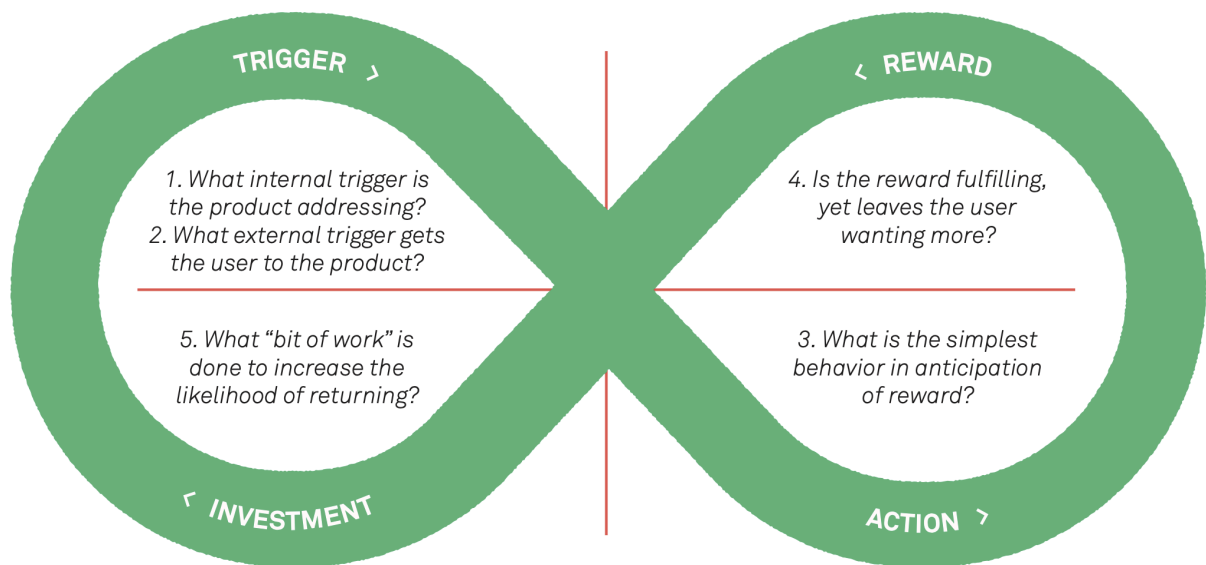
Варто зауважити, що навмисне поліпшення часто використовуваної функції - це одночасно великий ризик і великий виграш. Уявіть поліпшення редактора в платформі ведення блогів. Зробили добре, і кожен користувач виграє від поліпшення. Зробили погано, і ви зіпсували роботу всіх користувачів. Великий ризик, велика нагорода.

2. Збільшення частоти використання

Це поліпшення з надією, що клієнти користуватимуться функцією частіше. Додавання пунктів у стрічці активності або більше налаштувань у пошуку означає, що користувачі читають стрічку частіше або використовують пошук для більшого кола завдань. Такий тип поліпшення може зробити функцію, якою користувалися раз на тиждень, такою, що використовується щодня.

Функція підтвердження навичок у LinkedIn - хороший приклад такого поліпшення. Вони додали можливість перехресного підтвердження навичок ваших друзів в один клік. Ви можете підтвердити навички чотирьох друзів, і випадковим чином отримати підтвердження ваших навичок від інших людей. У результаті, це створює більше підтверджень, більше заходів у додаток і ще більше дій в один клік. Замкнуте коло.

Принцип, який вони використовували, описав Нір Еяль, автор книги На гачку. Він каже, що звичка формується з повторюваних дій, що складаються з чотирьох ключових елементів...



Принцип формування звички

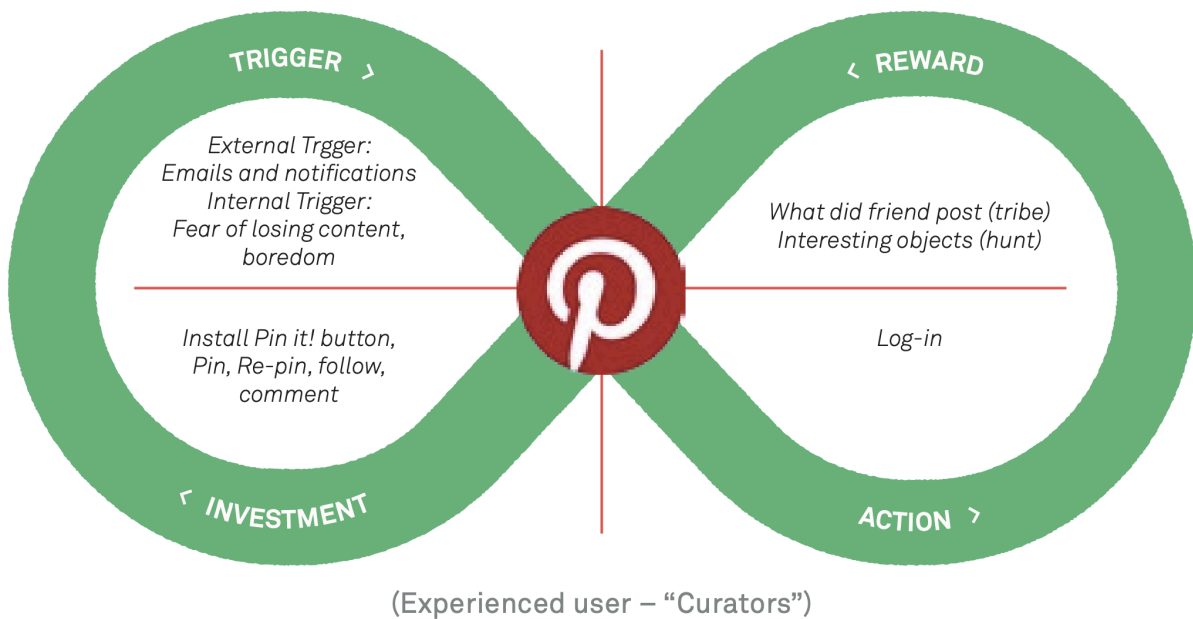
Тригер: причина, через яку користувач заходить у застосунок (отримав лист про те, що його навички хтось підтвердив).

Дія: яку користувач здійснює в очікуванні нагороди (переглядає стрічку, шукає тощо).

Нагорода: яку отримує користувач, від своїх дій (бачить красиву дошку Pinterest).

Інвестиція: дія, яка призведе до більшої кількості тригерів (підписка, лайк, встановлення зв'язку з користувачами тощо).

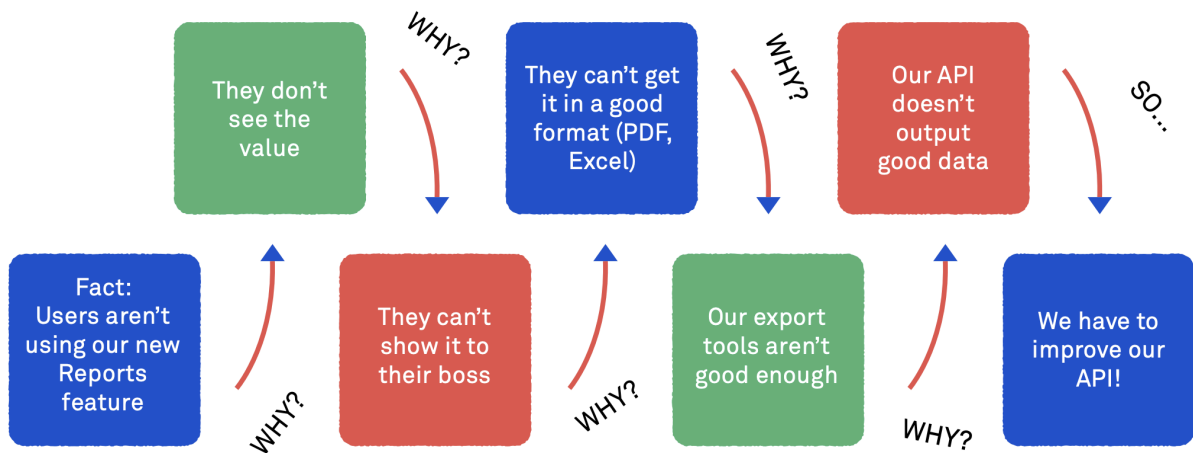
Ось як це виглядає на прикладі Pinterest



Формування звички в Pinterest

Застосовуйте збільшення частоти використання: для функціоналу, який **рідко** використовують **більшість** клієнтів. Якщо ви вірите, що часте використання буде корисним для них. Має сенс оцінити, яка користь для вашого бізнесу від збільшення частоти використання цієї функції. Наприклад, якщо Basecamp збільшить частоту, з якою користувачі створюють проекти, вони природним чином збільшать прибуток, тому що їхня ціна залежить від цього. При цьому є багато можливостей зробити функцію привабливою без користі для бізнесу, а в деяких випадках, навіть погіршити цим основне використання продукту. Наприклад, показники підтверджень навичок LinkedIn, ймовірно, стали кращими. Але треба поставити запитання, чи не отримано це ціною довіри до них?

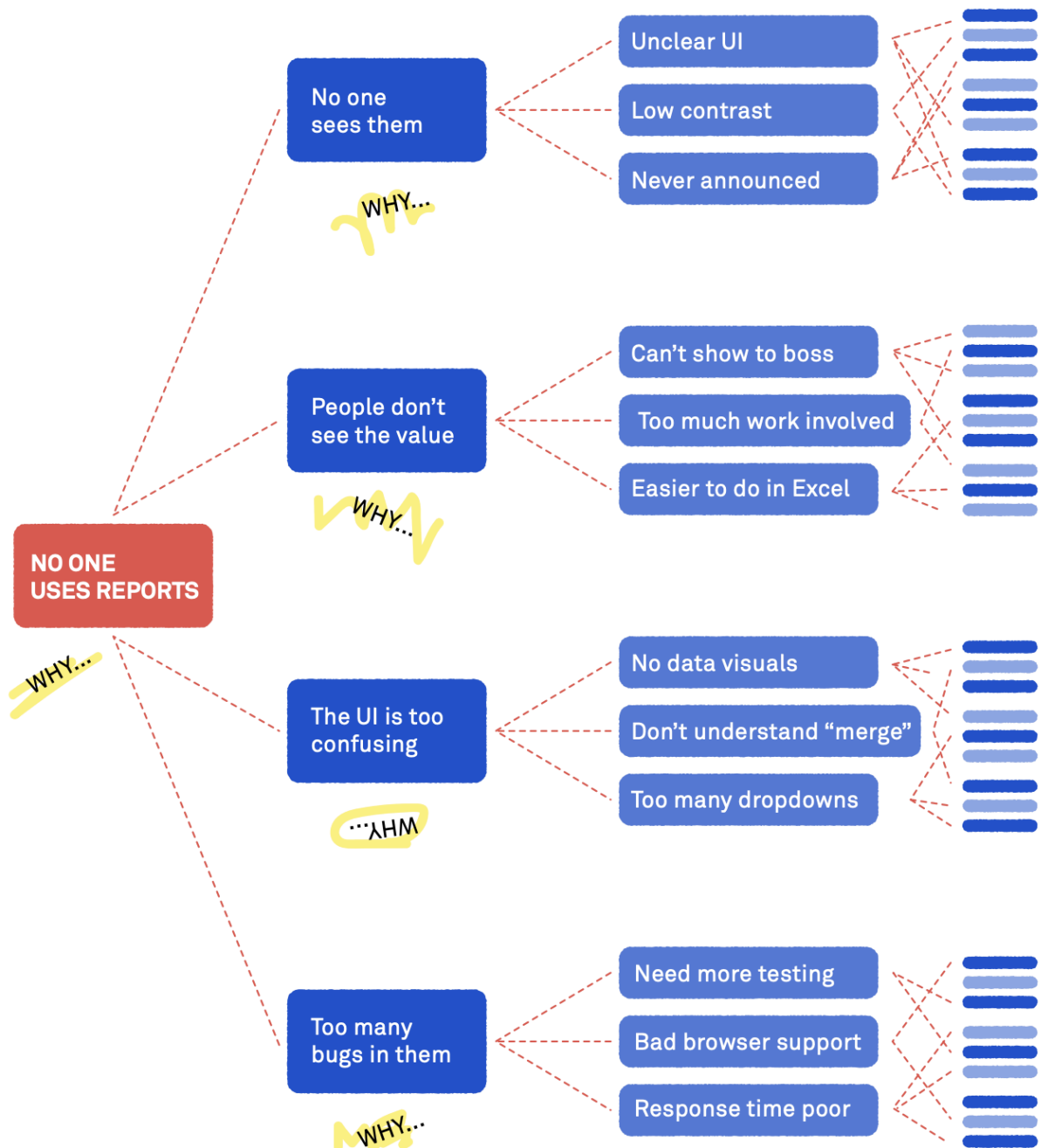
3. Поліпшення прийняття функціоналу



Поліпшення прийняття призначене для тих, хто не користується функцією. Щоб зробити її корисною для більшої кількості користувачів, знайдіть і виправте проблеми, які заважають їм користуватися. Тут добре працює техніка [П'ять чому](#).

Припустимо, ваші клієнти не користуються звітами. Чому? Вони не бачать користі в цих звітах. Чому? Вони не можуть показати ці звіти начальнику. Чому? Вони не можуть вивести у звітах необхідні дані. Чому? Через поганий експорт даних. Чому? Наш API не видає потрібні дані.

Якщо ви поставите запитання достатню кількість разів, ви знайдете корінь проблеми.



Не обмежуйтеся розмовою з одним клієнтом. Швидше за все, причини складніші, ніж описано вище. Ви виявите повторювані причини. Потім, ви зможете виправити ключові проблеми, які не дають клієнтам користуватися всіма доступними функціями.

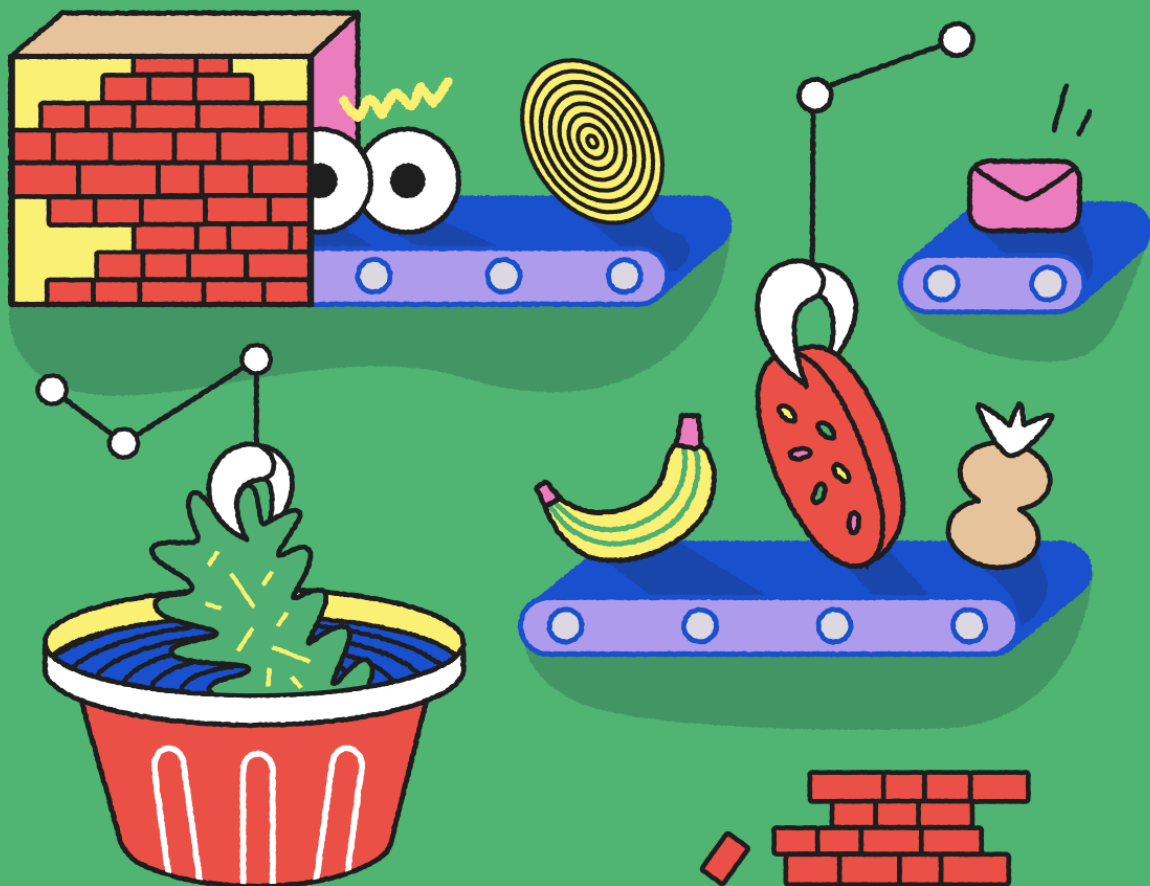
Використовуйте поліпшення ухвалення, якщо є важлива функція, якою не користуються багато користувачів і ви бачите способи, щоб їм було легше почати користуватися.

Коли плануєте покращення прийняття, також розглядайте зміни поза кодом. Іноді проблема не в тому, як функцію спроектовано або зроблено, а в тому, як її подають. Наприклад, іноді користувачі мають знати чому і як використовувати звіти. У цьому випадку поліпшення маркетингу продукту і комунікацій з клієнтами можуть вирішити проблему без зміни продукту.

Безперервне поліпшення

На ранніх стадіях стартапи мають перевагу перед існуючими компаніями. Стартапи швидше адаптуються до змін ринку. Тому що в них ще небагато технічних боргів, немає старого функціоналу, немає проблем зі зворотною сумісністю, немає дорогих клієнтів, що уповільнюють їхній розвиток. Але іноді ця швидкість і гнучкість може спричинити зміну напрямку як у безголових курок, замість того, щоб сфокусуватися на поліпшенні корисних функцій їхнього продукту.

Продакт менеджер у стартапах стикається з двома основними складнощами. Перше, знайти поліпшення, корисні для бізнесу та клієнтів. Друге, переконалися, що ці поліпшення не загубилися в документації, а насправді реалізувати їх. Тому що для онлайн стартапів найважливіше - випускати зміни, інакше ти мертвий.



Глава 2. Коли говорити "ні" новому функціоналу

Існує обмежена кількість поліпшень, які можна зробити в продукті або функціоналі. У якийсь момент вам доведеться вирішувати, які нові функції мають сенс для продукту, і як ви подаватимете їх користувачам.

Коли Facebook створив стрічку новин, це викликало бурю протесту серед ранніх і найбільш лояльних користувачів. Але чи мав би Facebook таке зростання, якби він не проігнорував протест і не показав би цінність нового функціоналу?

У цьому розділі ви дізнаєтеся про створення дорожньої карти продукту і формулу, що показує, де краще докласти зусиль. І чому "ні" це найважливіше слово у вашому словнику, яке ви маєте говорити часто.

Додавання нових функцій

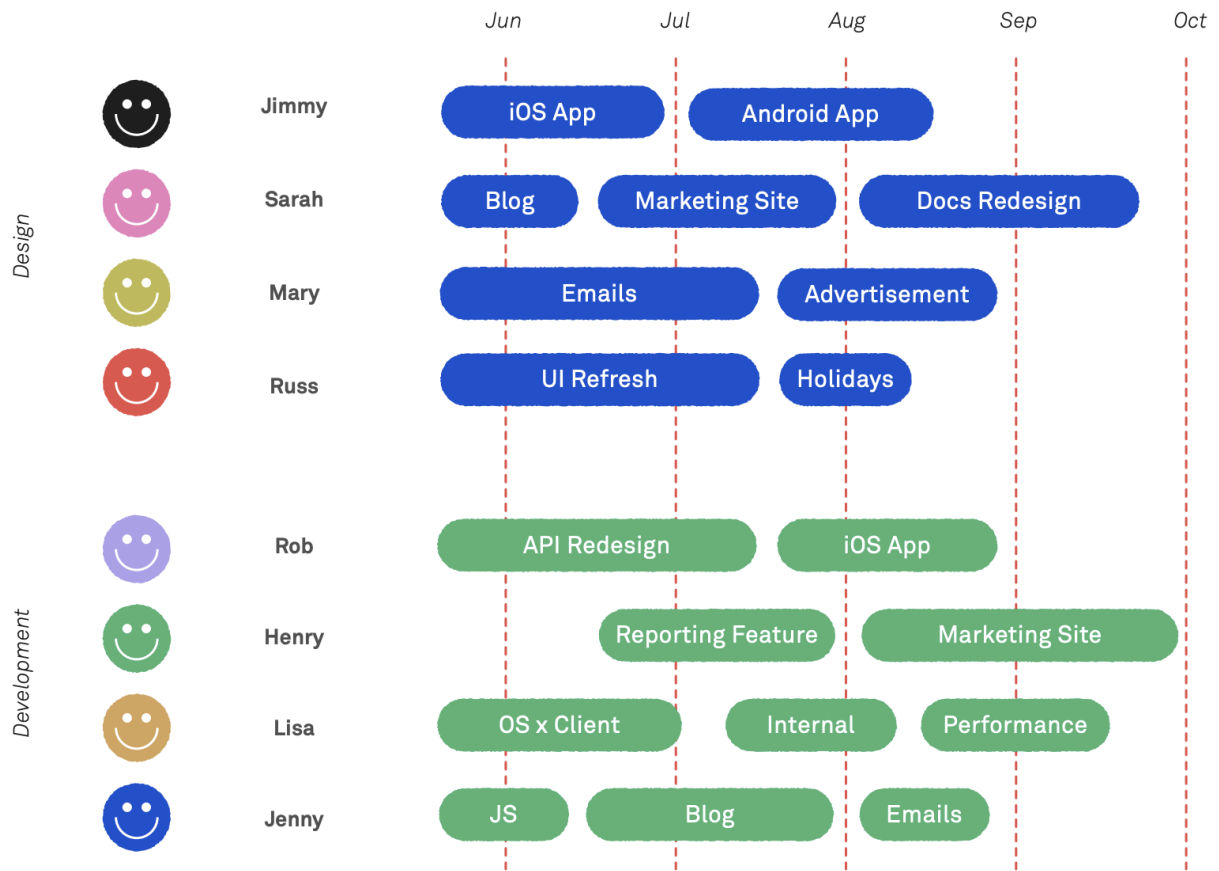
Нові функції розширюють можливості продукту, часто створюють сплеск інтересу, висвітлюються в деяких ЗМІ. Ці фанфари приваблюють нових клієнтів і з'являються нові варіанти використання продукту. Зазвичай, нові функції - єдині поліпшення, про які коли-небудь почують не-клієнти.

Але нові функції містять у собі ризики. Ви маєте бути впевнені, що вони корисні. Бо з ними, як із дитиною, вам доведеться підтримувати їх, що б не трапилося.

Запитайте ваших клієнтів "Хотіли б ви Календар/Трекер часу/Діаграми Ганта?" і вони дадуть відповідь "Так". Тому що це одностороння пропозиція "щось без нічого натомість". Клієнти не робили вибір між конкуруючими пріоритетами. Це призводить до того, що клієнти кажуть "хочу" для функцій, які насправді їм не потрібні.

Запитайте клієнтів "Хотіли б ви, щоб продукт був швидшим або, щоб ми додали функціонал з мітками?" і ви почуєте іншу відповідь. Кожен цінує швидкість роботи. Тому під час планування нових функцій важливо розуміти компроміси.

Де слабкі місця вашого продукту і де це має значення?



Приклад дорожньої карти продукту

Якщо ви фокусуєтеся тільки на нових функціях, ваш продукт буде кілометри завширшки і кілька сантиметрів у глибину. А якщо ви фокусуєтеся тільки на виправленнях, у вас не буде інновацій, і ваш продукт застаріє. Дійсно, складні рішення.

Коли вам потрібно поліпшити продукт, корисно запитати, "Де наші слабкі місця і де це має значення?"

Під час поліпшення продукту, ви фокусуєтеся на найважливіших частинах продукту, робота яких не влаштовує клієнтів. Обидві умови необхідні, інакше

ви працюватимете над частинами, які нікому не потрібні, або над тими, які вже більш ніж хороші.

У своїй відомій статті Створення інновації з відгуків клієнтів, Ентоні Ульвік пропонує алгоритм пошуку можливостей, з практичними кроками створення дорожньої карти, врахування важливості та задоволення потреб.

Де знаходяться можливості?

	Imp.	Sat.	Opp.		Imp.	Sat.	Opp.
<i>Be aware of any problems with any of my projects</i>	9	7	11	<i>Keep our clients aware of what we're working on</i>	10	7	13
<i>Find what my team have been working on recently</i>	6	8	6	<i>Review and individual's contributions to a project</i>	5	8	5
<i>Understand how accurate our project estimates are</i>	4	1	7	<i>Record project decisions in a clear formal manner</i>	9	5	13
<i>Find contact details for a client quickly</i>	8	2	14				

$$\text{OPPORTUNITY} = \text{IMPORTANCE} + (\text{IMPORTANCE} - \text{SATISFACTION})$$

Алгоритм пошуку можливостей Ульвіка чітко визначає недоліки в продукті. Це досягається тим, що користувачі ранжують завдання, які їм потрібно робити, за **важливістю** і наскільки вони задоволені **наявним способом** робити ці завдання. У результаті можливості визначаються формулою:

$$\text{важливість} + (\text{важливість} - \text{наявний спосіб})$$

Ця формула часто виявляє можливості, які не були помічені. Зазвичай найбільші можливості знаходяться в тих частинах, про які продакт-менеджер думає: "готово", "без помилок", "досить добре" тощо. Інакше кажучи, невелике поліпшення у важливому завданні майже завжди є більшою можливістю, ніж велике поліпшення в допоміжному завданні.

Це той випадок, коли правило 80/20 може не спрацювати. Думка, що 20% функцій становлять 80% цінності продукту, може, й правильна. Але це також означає, що ви пропонуєте посередню якість клієнтам у найбільш важливих для них завданнях. Це спостереження зробив Марк Цукерберг, коли розповідав про історію Facebook...

"

Не можна дотримуватися принципу 80/20 у всьому. Мають бути деякі речі, в яких ви найкращі. Речі, в яких ви йдете значно далі за інших, піднімаючи планку на новий рівень і створюючи найкращий продукт, який тільки може бути.

Марк Цукерберг
Засновник Facebook

Не існує єдиного правильного способу пріоритизувати функції в дорожній карті. Але зате є багато неправильних. Якщо є можливості в наявних галузях продукту, і ваша дорожня карта ігнорує їх на користь нових функцій, ваш продукт незабаром може стати таким, що робить усе підряд.

Розберіться з проблемами в продукті, або це зробить хто-небудь із конкурентів.

Чому ви не додаєте нові функції?

Продакт-менеджери і засновники в цій ролі повинні вміти говорити "ні". Не "може бути" чи "пізніше". А тільки одне слово "ні".

Створення продукту це не означає додавання нескінченної кількості корисних функцій, що відповідають тематиці. А створення злагодженого функціоналу з чітко визначеними параметрами.

Коли ваш продукт стає популярним, у вас з'являється море хороших ідей для нового функціоналу. Це ідеї від клієнтів, колег і від вас. Оскільки ідеї хороші, у вас буде багато причин сказати їм "так". Нижче 12 аргументів, які зазвичай наводять, щоб включити функції в продукт:

1. Але статистика має гарний вигляд

"Ми ввімкнули цю функцію для маленької групи клієнтів і ступінь прийняття просто зашкалює". Зазвичай такий підхід страждає від вибіркового аналізу. Продукти - складні системи. Те, що виглядає як збільшення прийняття функції, насправді переміщення цифр з місця на місце.

Вам здається, що ваша функція ярликів - це успіх. Але чи помітили ви, що функція тегів раптом стала непопулярною? Навіть якщо статистика показує інтерес і збільшення прийняття функції, все одно потрібно поставити запитання, чи відповідає цей функціонал продукту? Додайте Тетріс у ваш продукт і ви скоріше за все побачите сплеск використання цієї функції. Але чи став продукт кращим?

2. це займе всього кілька хвилин

Обсяг робіт ніколи не повинен бути причиною додати функцію. Можливо, це причина додати в дорожню карту як легку перемогу, але це рішення про дорожню карту, а не про продукт. Багато поганих ідей можуть бути створені швидко. Не дайте себе обдурити. Не існує маленьких змін. Навіть невеликі зміни додають приховану складність, яку не врахували в оцінці "але це всього 5 хвилин".

3. але ми втратимо клієнта

Це різновид шантажу. Жоден клієнт не може бути важливішим за хороший продукт. Шлях до продукту, зробленого на замовлення, заставлений табличками "зроблено тільки для цього клієнта". Він веде до ідеального продукту, але лише для одного клієнта. І, незважаючи на те, що вони тепер залежать від вас, ви робите те, що вони скажуть. Створення додаткового функціоналу для одного клієнта завжди йде на шкоду багатьом іншим.

4. Але ми можемо зробити це опціональним

Це веде до смерті від великої кількості налаштувань. Роблячи функцію опціональною, безумовно, зменшує складність основного інтерфейсу, але ця складність продовжує просочуватися у всіх інших місцях. Видима ціна за це - складний інтерфейс із безліччю умовних елементів і тоннами налаштувань. Прихована ціна - кожна опціональна функція послаблює ідею вашого продукту. Ви стаєте "трекером часу, який може розсилати інвойси і зводити баланс платежів, але без звітів".

5. Але сусід мого двоюрідного брата сказав...

Це ухвалення рішень на основі байок. Таких випадків повно в споживчих продуктах і в сервісних компаніях, які ніяк не можуть вирішити, що вони все-таки роблять. Адже так легко на основі поодиноких випадків проігнорувати роки досвіду, досліджень, статистики.

Говорячи "в компанії мого брата вони використовують Google Analytics і розширені сегменти" дуже просто переконати використовувати розширені сегменти, ігноруючи питання:

- ☐ що робить ваш продукт?
- ☐ чи є компанія вашого брата гарним прикладом наших клієнтів?
- ☐ чи справді вони використовують, чи просто говорять?
- ☐ чи дійсно розширені сегменти - це гарне рішення для наших клієнтів?

6. Але у нас зараз нічого не заплановано

Диявол підкидає роботу для незайнятих команд, і виходять дійсно жажливі функції. Якщо хтось бачить інженера, який сидить без діла, тут же дає йому створення нової функції, щоб не байдикував. Ухвалюються поспішні рішення і робиться дизайн нашвидкуруч в ім'я того, щоб не втрачати час.

Це поганий шлях "покращення" продукту. Замість того, щоб додавати технічний борг, є можливість зменшити наявний борг. Хто працював на кухні знає: "є час тинятися, значить є час прибратися". Вільний час краще використовувати для виправлення помилок, поліпшення тестів, рефакторингу тощо, а не руйнувати ідею продукту за допомогою гасла "щоб команда була зайнята".

7. Але нам дозволили працювати над чим ми захочемо

Цей аргумент виходить із корпоративної культури. Є багато великих компаній, які обіцяють інженерам створювати те, що вони захочуть і включати це в продукт. Зазвичай ця обіцянка має один із таких наслідків:

1) Це брехня, щоб залучити інженерів. Люди швидко помічають це і кидають спроби додати своє в продукт. Ви не можете імітувати культуру компанії.

2. це правда. У результаті виходить продукт, повний незавершених ідей.

Є різниця між тим, щоб надихати інженерів створювати внутрішні речі (гарний варіант), і між тим, щоб дозволяти додавати функції в продукт, минаючи продакт менеджмент (поганий варіант).

8. Але 713 000 людей хочуть цю функцію

Завжди знайдеться хтось, хто використовуватиме сирі числа, щоб переконати вас. Будь-який продукт, яким уже користуються, може зробити емоційну заяву на кшталт такої: "Ви можете заповнити парк Долорес людьми, які просять інтеграцію з Excel". Ви справді хочете сказати "ні" всім цим людям? Так, ви повинні це зробити. Інакше більшість ваших користувачів постраждають. Питання не в тому, чи можемо ми заповнити парк Долорес людьми, яким потрібен цей функціонал. А в тому, чи є ця функція цінною в рамках бачення нашого продукту для всіх наших користувачів.

9. Але наші конкуренти вже зробили це

Це не означає, що це гарна ідея. Можливо вони пробують щось. Це може бути жахлива ідея. Це може бути щось, що вони планують видалити з продукту. Помилка припускати, що ваші конкуренти значно розумніші за вас. Одержимість функціями ваших конкурентів веде до створення "вчорашніх технологій завтра".

10. Але якщо ми не зробимо, це зробить хтось інший

Це не означає, що це має бути у вашому продукті. Якщо хтось зробить це, чи стане клієнтам не потрібен ваш продукт? Чи підуть вони до конкурентів? Говорячи "хтось інший" звучить добре, але нічого не означає. Ми всі так іноді говоримо. Часто ця логіка використовується для розширення продукту, тому що ви не хочете визнати, що продукт повинен мати чіткі межі. Ви боїтеся прокреслити цю лінію.

Наприклад, типове побачення включає кіно, вечерю і таксі додому. Якщо власник кінотеатру постійно турбується, що інші компанії зробили і намагається збільшити цінність своєї пропозиції, він додасть ресторан до кінотеатру, і створить компанію таксі. У результаті всі три пропозиції будуть слабкими. А ресторани почнуть показувати кіно...

11. але начальник дуже хоче ввімкнути цю функцію

Якщо начальник є також продакт-менеджером і в нього є достатньо часу та знань, щоб ухвалити цілісне рішення, тоді це добре. Однак, якщо хтось намагається заробити прихильність керівництва, займаючись аматорськими проектами, до яких схильний його менеджер, тоді це неминуче веде до проблем.

12. але це може бути "те, чого нам не вистачало"

Це класичний спосіб посилатися на невідоме. Створення продукту вимагає складних рішень про те, що включити в нього. Ви можете будувати домисли про те, що будь-яка не створена функція могла б принести успіх продукту. Але це всього-лише домисли, нічого більше. Коли ви боїтеся ухвалити важкі рішення, ви скочуєтеся до подібних домислів, і, як наслідок, додаєте в продукт усе підряд. Зрештою у вас виходить набір функцій, а не продукт.

Чому "ні" це важливо

Річ у тім, що ніхто не тримає погані ідеї в дорожній карті. Визначити й уникнути поганих ідеї нескладно. Але справжні рішення про продукт - непрості. Вони вимагають подивитися на пропозицію і сказати: "Це справді гарна ідея. І я розумію, чому наші клієнти хочуть її. Але ми не будемо робити її. Натомість, ось що ми зробимо."

Не буває маленьких змін

Все ще не вірите, що маленьких змін не буває? Давайте розглянемо наступний приклад.

"Ми хочемо обмежити довжину тексту в коментарі до 140 символів, тому що нам може знадобитися використовувати SMS для відправки на певному етапі. Це маленька зміна, вірно?"

Не вірно!

Не буває маленьких змін, якщо ви налаштовані робити якісний продукт. Звісно, програміст може зробити це за три хвилини. Зрештою, це всього лише одна умова в коді.

Але продакт менеджмент ніколи не буває таким простим. Перш ніж приступити до "маленьких змін", потрібно поставити кілька запитань. Повертаючись до довжини коментаря в нашому прикладі, давайте почнемо з кількох простих запитань.

Що буде, якщо довжина коментаря буде більшою за 140 символів? Чи повинні ми просто обрізати рядок, чи показати користувачеві повідомлення про помилку? Якщо показуємо повідомлення, де воно з'явиться? Що в ньому говориться? Хто напише текст повідомлення? Як ми пояснимо користувачеві, чому ми обмежили довжину коментаря до 140 символів? Який вигляд матиме повідомлення? Чи визначений у нас стиль для подібних повідомлень? Якщо ні, хто підготує дизайн?

Але почекайте, є дещо ще...

Якщо навіть у нас є відповіді на всі ці запитання, залишилося дещо ще. Робити це тільки на стороні сервера не найкращий спосіб обробки помилок. Потрібно зробити це також і на клієнтській стороні. Але якщо ми робитимемо перевірку помилки на клієнтській стороні, виникають нові запитання...

Хто напише зміни в JavaScript? Чи буде JavaScript код показувати ту саму помилку, що й серверний код? Як воно поводитиметься з вимкненим JavaScript? Що зробити, щоб у майбутньому при будь-яких змінах правила 140 символів не забувати змінювати і на серверній, і на клієнтській стороні?

Але ми ще не закінчили. Подивіться на це з точки зору користувача. Вони дивуються, чому є обмеження в 140 через дивну причину, яку вони не розуміють. А тепер ми ще вимагаємо від них вгадувати довжину повідомлення? Повинен бути спосіб кращий ніж вгадувати. Давайте додамо лічильник символів. О, це породжує ще кілька запитань...

Майже завершили...

Хто зробить лічильник символів? Якщо будемо використовувати готовий, знайдений в інтернеті, хто протестує в усіх браузерах, які ми підтримуємо?

Також, де відображатимуться цифри лічильника? Який вигляд матиме лічильник? Звісно, потрібно змінити стиль цифр, коли користувач наближається до нуля, і, безумовно, цифри мають виглядати інакше, коли користувач написав понад 140 символів. А може потрібно не давати вводити більше? Якщо так, що буде, якщо вони вставлять текст із буфера обміну? Чи повинні ми дозволити редагувати, чи попередити їх?

Ми зробили лічильник, відображення помилок, додали перевірку на стороні сервера, і протестували у всіх підтримуваних браузерах. Тепер залишилося написати автоматичні тести і випустити новий функціонал. Припускаючи, що механізм випуску нової версії у вас налагоджений, це зробити нескладно.

Усе це ігнорує факт, що користувачі будуть дивуватися, чому, хтось написав великий коментар деякий час тому, а вони тепер можуть писати лише 140 символів. Очевидно, що ми маємо попередити техпідтримку про цю зміну, оновити документацію, API, додатки для iPhone та Android. Також, потрібно вирішити, що робити зі старими коментарями, урізати їх чи залишити як є?

Не кажучи вже про те, що робити з усіма цими емодзі, які так популярні сьогодні... удачі в надсиланні їх через SMS. Можливо, нам потрібно не дозволяти вводити неприпустимі символи, а це означає нові повідомлення про помилки, додатковий код... список завдань триває.

Коли ви пройшли все це, ваша нова функція буде в продукті, і це лише для лічильника символів. Тепер спробуйте щось складніше. Не існує маленьких змін, якщо ви робите речі як слід. Ось чому, як продакт-менеджер, ви маєте добре розуміти, що варто додати функцію, перш ніж ви кивнете головою і додасте функцію в дорожню карту.

Облік усієї картини

Те, що здається двохвилинною роботою, часто перетворюється на двогодинну роботу, якщо не враховується вся картина. Функціонал, який має вигляд "корисного" всього за кілька хвилин, можна сміливо оцінювати як понад дві години.

Висновок: Обсяг роботи ховається в хвилинах, а не в місяцях. Слідкуйте за хвилинами, і буде порядок із місяцями. Дати згоду на додавання функції оманливо легко. А ось програмування цих функцій уже навряд чи буде легкою справою. Підтримка буде кошмаром. Коли ви прагнете якості, не буває маленьких змін.

Нічого не з'являється безкоштовно

У розробці програм ви часто можете почути, що отримуєте щось безкоштовно. Наприклад: "фреймворк дає це безкоштовно" або "якщо ми перейдемо на провайдера X, то отримаємо Y безкоштовно". Ваші розробники можуть сказати: "ми отже це робимо, тому отримаємо цей додатковий шматочок безкоштовно".

Нічого не з'являється безплатно в розробці продукту.

Це не безплатно, тому що ви сидите і розмовляєте про це, дискутуєте, чи варто робити це. Це витрати часу. Це не безкоштовно перевіряти, що "безкоштовний"

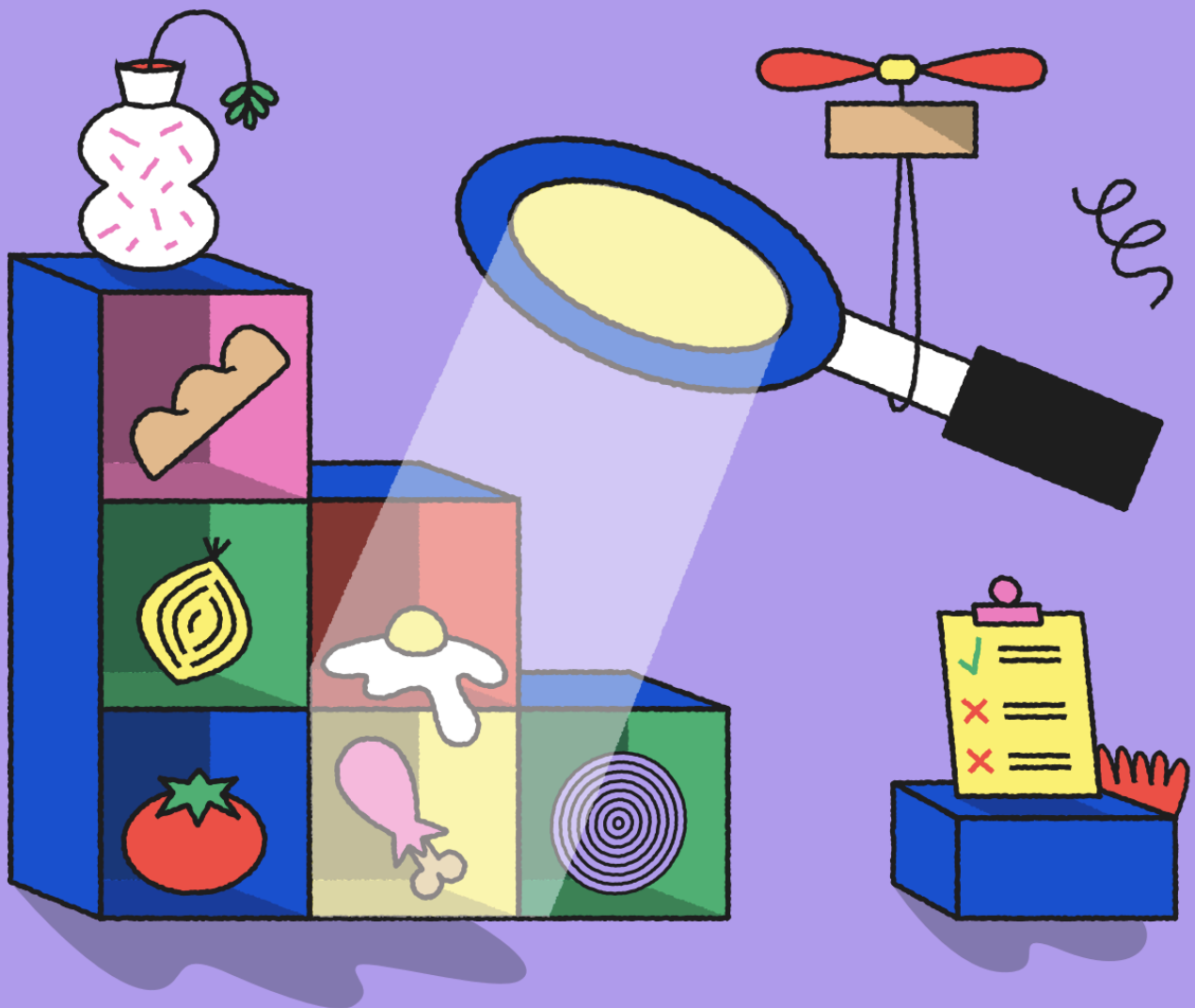
шматок зроблено правильно, і що він працює у всіх сценаріях. Тестування ніколи не було безкоштовним.

Це не безкоштовно обговорювати додавання функції. Якщо ви додасте функцію, вам потрібно повідомити вашій команді, підтримці клієнтів, і самим клієнтам. Ви повинні повідомити клієнтам, інакше це дійсно безглуздо. Комунікації не безкоштовні.

Це не безкоштовно утримувати функцію. Виникають питання від клієнтів до служби підтримки. Вам необхідно оновити документацію, відео, все має бути оновлено.

Навіть видалити функціонал не безкоштовно. Підтримка ніколи не була безкоштовною.

Функціонал можна порівняти з домашніми тваринами. Повно оголошень про безкоштовних тварин. Але ви не берете їх усіх додому з однієї простої причини. Є велика різниця між ціною покупки і вартістю утримання.



Розділ 3. Які функції додавати в продукт

Коли справа стосується стратегії продукту в Інтерком, нас виручає техніка [Робота-яку-треба-зробити](#) (Jobs-to-be-done). Ця техніка допомагає вирішити, що додавати в продукт. За допомогою цього підходу з'ясовуються причини використання продукту клієнтами, замість традиційної техніки сегментування клієнтів за демографічною ознакою.

Люди не купують продукт, тому що вони потрапляють у якусь групу, наприклад, жінки після 30 років, які живуть у приватному будинку, працюють неповний день. Вони купують, щоб вирішити конкретну проблему. Якщо ви розумієте "роботу", для якої клієнти "наймають" ваш продукт, у цьому разі ви зможете сфокусуватися на тому, щоб вони отримали бажаний результат. Функції, які ви додаєте, мають допомагати клієнтам зробити їхню роботу.

Як зробити речі, які потрібні людям

Проблеми, з якими стикаються люди в житті, рідко змінюються з покоління в покоління. Але продукти, які вони використовують для вирішення проблем - змінюються постійно.

Коли ви створюєте новий продукт, ви маєте вірити, що зможете зробити найкраще рішення, яке люди захочуть використовувати, бо воно дає найкращі результати. Чітке розуміння, якого результату хочуть клієнти і як вони досягають його зараз, є основою, щоб досягти успіху в продакт-менеджменті.

Може ваші клієнти хочуть розваг, або проводити більше часу з друзями, або знати, над чим працюють їхні колеги, а може, їм потрібно збільшити зростання їхнього бізнесу. Якщо бажаний результат реальний, вони вже досягають його тим чи іншим способом. Ваше завдання - поліпшити цей спосіб.

Примітка: Якщо ви не можете дізнатися, який спосіб вони використовують зараз, є шанс, що це вигаданий результат ("Було б круто мати всі мої файли з

соціальних ЗМІ в одному місці") або просто бажання ("Звичайно я хочу схуднути").

Фокус на результаті замість категорії, індустрії або типу продукту, дає змогу дізнатися про ваших справжніх конкурентів. Компанія, яка зосереджена на галузі, в якій вона перебуває, а не на результатах, які вона приносить, втрачає зв'язок з реальністю, і незабаром втрачає клієнтів.

Наприклад, газети вірили, що вони в газетній галузі. Як результат, вони не могли зрозуміти, чому нудьгуючі люди дорогою на роботу перестали купувати їх. Вони дивилися по сторонах, намагаючись зрозуміти, хто з їхніх конкурентів вкрав у них клієнтів. Вони експериментували з новими форматами, низькими цінами, помітними заголовками, але не змогли зупинити занепад. Якби вони сфокусувалися на результаті (нудьгуючі люди хочуть розважитися в дорозі в короткі проміжки часу за допомогою міні-статей). Тоді їхні конкуренти (Twitter, Facebook, новинні додатки) не були б такими небезпечними для них.

Розгляньмо кілька галузей, які, як і у випадку людей, що нудьгують у транспорті, залишаються актуальними завдяки сотням технологічних поліпшень.

Люди, зокрема молоді люди та студенти, хотіли передавати записи та повідомлення, без страху, що інші люди побачать їх...

Люди все ще хочуть зробити це, і вони користуються Snapchat.

Люди зберігали фотографії в безпечному місці, такому як коробка з-під взуття під ліжком.

Люди все ще хочуть робити це і користуються Dropbox.

Люди хотіли ставити свої улюблені фотографії на видному місці, наприклад, над каміном, щоб усі могли бачити їх...

Люди все ще хочуть робити це і користуються Facebook.

Люди хотіли збирати вирізки з газет і журналів з ідеями про реновацію будинку або іншими проектами...

Люди все ще хочуть робити це і сьогодні вони користуються Pinterest.

Люди хотіли ділитися відгуками та порадами для інших мандрівників у гостьових книгах...

Люди все ще роблять це і сьогодні вони використовують Foursquare.

Робити роботу краще

Існують сотні прикладів як ті, що показані вище, і в них є закономірність. Щоб люди захотіли щось, необхідно розуміти справжню причину потреби людей або бізнесу і, потім, використовувати технологію для того щоб:

- ☐ спростити кроки досягнення результату,
- ☐ зробити доступним для більшої кількості людей,
- ☐ зробити доступним в інших ситуаціях.

Перший підхід, спрощення кроків досягнення результату, часто використовують стартапи. Взяти потреби, де наявні рішення застаріли або надто складні, і знайти простіші кроки для досягнення того самого результату. Замовлення таксі в місті передбачало дзвінки на кілька номерів, доки не знаходилася компанія з вільною машиною, потім довгий діалог про те, де вас забрати, куди відвезти і о котрій годині вам потрібно бути на місці.

Сьогодні ви натискаєте кнопку і машина з'являється.

Другий підхід зазвичай передбачає зниження витрат часу або нижчу ціну, або зменшення бар'єру використання продукту, щоб більше людей могли користуватися ним, і таким чином, розширити ринок. Чотирнадцять років тому, якщо хтось хотів вести блог онлайн, він мав орендувати Linux-сервер, завантажити на нього архів із програмою ведення блогу, виконати серію дивних команд для розпакування та конфігурації.

Сьогодні це кілька кліків на Medium.

Третій підхід прибирає ситуаційне обмеження. Приймання платежів раніше потребувало масивних комп'ютерів із рулонами термопаперу, надсиленням документів до банку факсом, ISDN лініями і пакетними транзакціями, які виконують ночами.

Сьогодні картки можна приймати навіть за допомогою смартфона.

У Джефа Безоса є знаменитий вислів "зосередьтеся на речах, які не змінюються". Проблеми, з якими стикаються люди і компанії, змінюються не дуже часто. Способи, якими їх вирішують, змінюються майже щороку. Таким чином, створення продукту, який люди будуть використовувати, має починатися з того "що хочуть люди", а не заманливого "те що ми можемо".

Запам'ятайте: Легше зробити речі, які хочуть люди, ніж змусити людей хотіти речі.

Лакмусовий тест для нових функцій

Нижче список запитань "так/ні", на які ви можете швидко відповісти, перш ніж додати нову функцію в дорожню карту.

У попередньому розділі ми розповіли, як стратегія продукту допомагає сказати "ні". Але список причин, через які ми відмовляємо в додаванні функції, не особливо допомагає зрозуміти, чи варто додати функцію. Тому нижче список запитань, на які мають бути відповіді "так".

1. Чи відповідає функція баченню продукту?

У що ви вірите? Що ви знаєте про проблему, про що не знають інші? Як, ви вважаєте, має бути вирішена проблема? Будь-хто може знайти статистику,

запустити фокус групи, прочитати звіти Gartner, вийти з офісу. Але у вас є своє бачення.

Рішення, засновані тільки на баченні, іноді здаються ірраціональними, тому що це складні рішення. Люди розуміють, що їхній трекер помилок не вміє відстежувати час доступності сервера. Ніхто не очікує цього від трекера помилок. Але чи повинен інструмент для ведення проєктів мати звіти? Уже не так очевидно.

Що більше нюансів у рішеннях, то більший опір. Колеги, клієнти, навіть інші продакт-менеджери і засновники компанії чинитимуть опір. Ось кілька прикладів:

- ❑ Apple відмовилася створювати нетбуки в той час, коли нетбуки були найпопулярнішими. Кожен аналітик просив про це.
- ❑ Basecamp відмовилися додавати діаграми Ганта у свій продукт. За це їх прозвали сліпими ідеологами.
- ❑ Розробник Гаррет Дімон відмовляється додавати статус "відкласти" до його трекера помилок - Sifter. Тому що він не вважає, що це правильний спосіб працювати з помилками.

Що ще гірше, ви ніколи не дізнаєтеся, чи були ви праві. Тут не існує правильних або неправильних рішень. Це мистецтво, а не наука. Це лише ви і ваше бачення.

2 Чи буде це актуально через 5 років?

Це важке питання, чи буде ця функція корисною через 5 років? Ви почуватиметеся старим брюзгою. Але додаток, який був популярним у 2013, зник у 2014. Усі ці ефекти ковзання, розчинення і складання, які так всі любили рік тому, виглядають як поганий смак сьогодні. Якщо ви хочете витратити

найкращі роки свого життя на продукт, тримайтеся подалі від модного, зосередьтеся на сенсі.

3. Чи для всіх це корисно?

Не забувайте про пастку "чув часто/недавно". Ви не сумніваєтесь, що речі, які ви чуєте часто або нещодавно, мають бути додані в дорожню карту. Це природна реакція, викликана вашим співпереживанням клієнтам. Це не ввічливо, говорити "ні" людям, і чути ці прохання знову і знову. Тому "чув часто/нещодавно" буває виправданням. "Звісно, ми додамо це, я чув це сьогодні вже двічі", - каже директор із 4800 активними користувачами, для задоволення його 0.0625% користувацької бази.

Небезпека реакції на "чув часто/нещодавно" в тому, що вона дає ілюзію аналізу, відчуття, що ви зробили домашню роботу і це результат раціонального висновку, якого ви дійшли. Насправді ви ухвалили ледаче рішення для задоволення швидкоплинного бажання невеликої групи користувачів, не витративши й секунди на дослідження, скільки насправді користувачів хочуть або потребують цієї функції.

4 Це поліпшення, доповнення або інновація наявного процесу?

Створення повністю нового процесу у вашому продукті має бути рідкісним випадком. Більшість часу, має бути інвестовано в поліпшення, доповнення або інновацію існуючих процесів. І в кожному випадку ви повинні знати, що з перерахованого ви робите.

Якщо ви покращуєте наявне рішення, вашими метриками можуть бути: задоволеність клієнтів, зменшення [часу на використання інструменту](#) або кількості звернень у підтримку.

Якщо ви доповнюєте наявне рішення, метриками можуть бути збільшення частоти використання, тому що тепер воно працює для нових випадків і для нових умов.

Інновація рішення - найскладніше. Ви цілитеся в міфічний "абсолютно новий спосіб", який містить багато ризиків, але дає великий виграш. У цьому випадку можна вимірювати кількість нових клієнтів, хоча вони й будуть отримані неявно, як результат новин або маркетингової кампанії.

Оновлення це завжди захоплююче, але ви можете даремно витратити час. Хороший спосіб визначити чи варто робити, просто запитати: "Чи буде більше людей використовувати це? Чи будуть люди частіше цим користуватися? Якщо ні те ні інше, чи буде це, дійсно, краще для тих, хто вже використовує це?"

5. Чи впливає це на зростання бізнесу?

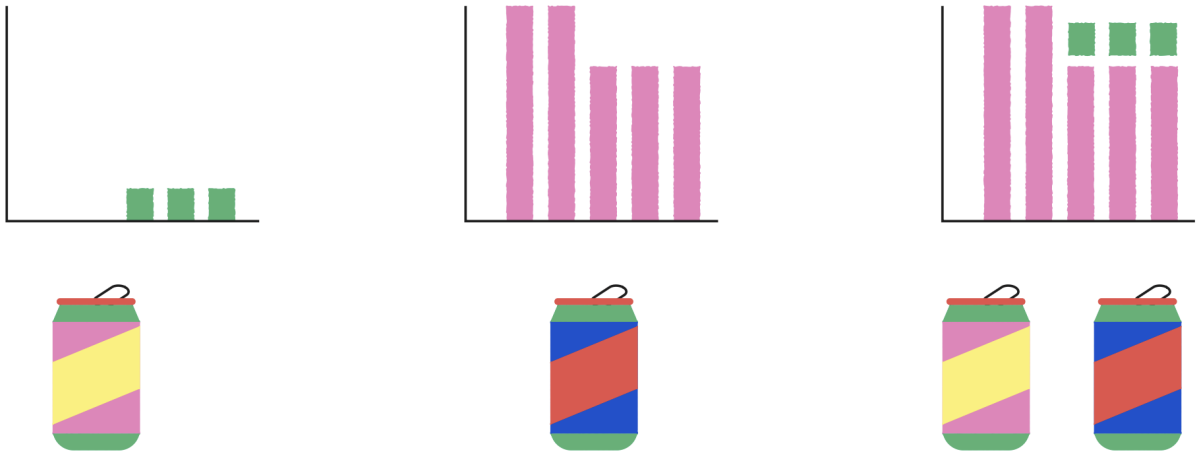
Чи можете ви побачити, який вплив матиме функція на прибуток? Наприклад, у застосунку для управління проектами, шаблони проєктів можуть стати таким випадком. Шаблони можуть допомогти використовувати продукт у незвичних для клієнтів випадках. Це збільшить кількість проєктів, які створюють клієнти, і, своєю чергою, збільшить кількість переходів на дорогі тарифи.

Зверніть увагу, зменшення відходів клієнтів також впливає на зростання бізнесу. Неважливо звідки прийшли долари. Часто функції додаються, щоб утримати клієнтів. Ідея всіх цих випадків у тому, щоб розуміти, як будь-яка робота впливає на зростання бізнесу, врешті-решт, [усі працюють для зростання](#).

6. Чи створить це новий осмислений вид залученості клієнтів?

Більшість метрик ігнорують систему і фокусуються на окремій функції. Додайте нову кнопку в продукт і люди натискатимуть її. Отримали достатньо кліків і ви

можете назвати це збільшенням залученості. Але в цьому немає особливого сенсу. Протилежна метрика може бути "чи зменшилося використання чогось іншого?". Якщо ви додали метрику для однієї сфери продукту, необхідно аналізувати інші сфери, які можуть бути порушені.



Слідкуйте за побічними ефектами

Метафора з реального світу може бути вплив Дієтичної коли з лаймом на продажі Звичайної дієтичної коли

На першому графіку видно, як запуск нової коли вплинув на продажі. Якщо ви продакт менеджер Дієтичної коли з лаймом, ви можете назвати це успіхом. Але якщо ви директор компанії, ви зрозумієте, що ускладнили процес виробництва із закупівлею всіх цих лаймів, соковижималок і рекламою. І при цьому загальний прибуток не збільшився. Не зовсім успіх. Не важливо, які метрики вам показував продакт менеджер.

7. Якщо це добре піде, чи зможемо підтримувати й утримувати це?

Одна з хибних думок про "швидкі перемоги" та "легкі зміни" - це те, що враховуються лише зусилля на створення. Наприклад, хтось здивував вас JavaScript закладками для вашого застосунку, або агенція створила 38 застосунків для Windows Phone у рекордні терміни та за низькою ціною.

Оскільки зусилля були порівняно невеликими, ви вирішуєте випустити це.
Успіх!

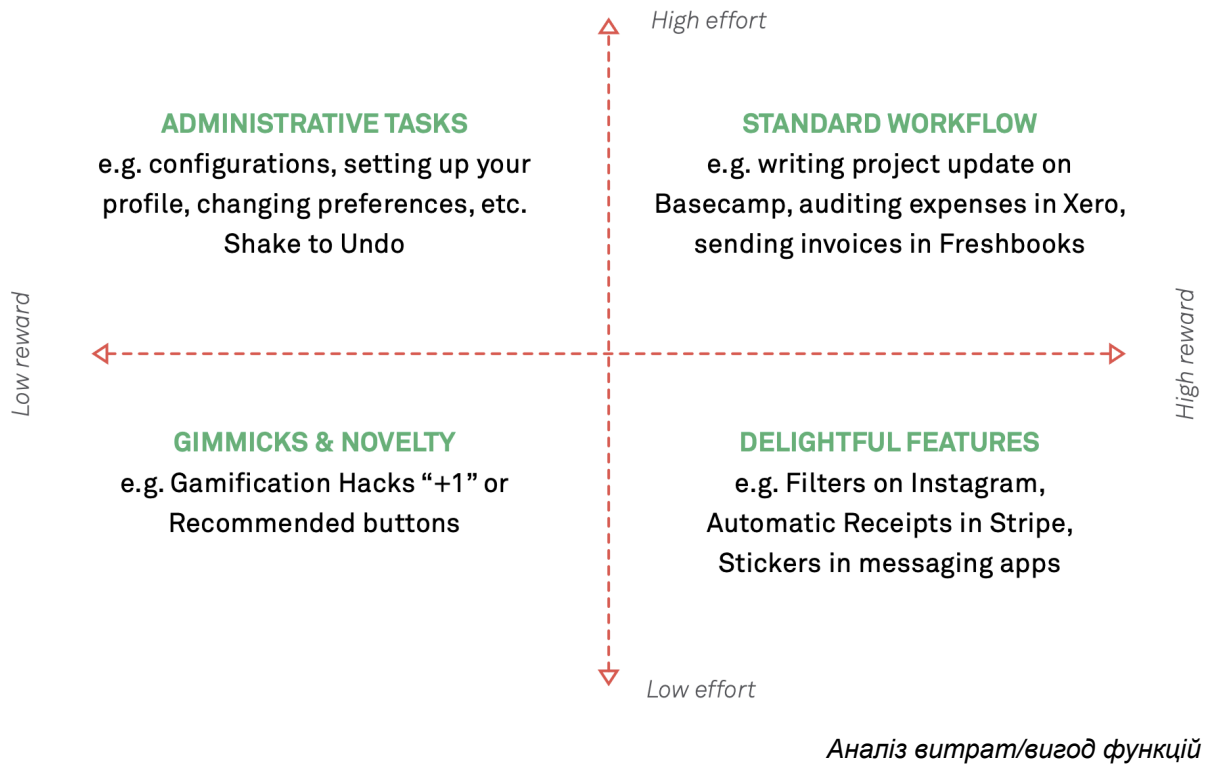
Потім з'являються запити в техпідтримку. Виявляється ніхто з команди не має достатньо досвіду з новим для них фреймворком, який використовували для застосунку. І ви застрягли з поламаним продуктом, і кілька сотень клієнтів скаржаться в підтримку. Звісно, це ніколи не буває таким очевидним. Хороші інженери рідко скажуть, що не знають чогось. Вони розберуться з цим за кілька вечорів і дадуть оцінку, коли виправлять помилку. Але це витрати часу, про які ви не думали, коли випускали продукт.

Інша область, де можна обпектисся - робити пропозиції, які ви не можете потягнути. Якщо у вас є система CMS, і ви пропонуєте безкоштовний дизайн головної сторінки, ви отримаєте більше реєстрацій користувачів. У цьому є сенс на ранній стадії розвитку компанії, коли вам дійсно потрібно спокусити клієнтів спробувати новий продукт. Але це не працює в довгостроковому періоді. Очевидно, що робити речі, які не масштабуються, це чудова стратегія, доти, доки не потрібно масштабуватися.

8. Чи можемо ми спроектувати це так, щоб вигода була більшою за зусилля?

Як [писав мій колега Пол Адамс](#), для будь-якої функції, щоб нею користувалися, цінність, яку отримують, має бути більшою, ніж зусилля на використання. Користувачі розуміли переваги, які може дати Google Plus. Але зусилля на регулярне перетягування безлічі контактів у різні квадрати просто не варті цього. Я відчуваю, що [додатки з розділення чека](#) потрапляють у цю ж категорію. Поділ чека може бути проблемою, але будь-яке рішення, яке було досі, коштує набагато більше, і йдеться не про ціну програми. Мова про час і зусилля.

Проектування продукту - це аналіз витрат/вигод. Наскільки корисно щось порівняно з витратами на використання функції. Ось як це можна уявити...



Важливо знати, над яким квадрантом ви працюєте. І буде мудрим триматися подалі від лівого верхнього квадранта, залишивши його лише для кількох базових функцій.

9. Чи можемо ми зробити це якісно?

Кожен продукт має слабкі сторони, часто в тих галузях, де продакт-менеджер боязко пробує і зазнає поразки. Але ці поразки рідко призводять до видалення функції. У результаті, залишаються "брудні сліди" і заплутана ідея продукту.

Проблема виникає, коли команди зачіпають області, в яких вони слабо розбираються. Часто це трапляється, коли команди виходять за межі "[зроблено для себе](#)", коли вони ще були самі собі клієнтом. У цей момент їхній процес проєктування ламається, і їм потрібен новий.

Наприклад:

- ❑ Команди, які не відстежують час, додають функцію відстеження часу.

- ❑ Люди, які не ведуть календар, проєктують функцію управління календарем.
- ❑ Люди, які не користуються трекером помилок, створюють систему ведення помилок.

Зверніть увагу, нічого з переліченого не є неправильним. Неправильний лише процес проєктування. Його потрібно змінити від "для мене зійде" в бік фокусу на проблемі. Щоб зробити хороший функціонал, необхідно глибоко розуміти завдання. Як у старому прислів'ї: якщо не можете зробити це добре, не варто це робити.

10. Чи можемо ми визначити мінімальний обсяг роботи?

Життєво важливо - почати з [мінімальної реалізації](#) функції. Ранні релізи дають зворотний зв'язок, необхідний для розвитку ідеї. Ознака того, що обсяг робіт обрано неправильно, відсутність специфіки. Наприклад, "щоб працювало для великих компаній". Інша ознака, якщо робота описана з погляду функції, а не результату, який необхідно досягти. Наприклад, "звіти", замість "щоб менеджери могли бачити продуктивність їхньої команди".

Легко погодитися з різними кліше на нарадах. Хтось скаже "це має бути зручніше для великих команд". Кожен киває, і запис про це з'являється на білій дошці. Тільки в специфіці можна визначити обсяг робіт. Наприклад, "потрібно дати можливість групувати колег по відділах" замість "нам потрібна підтримка великих компаній".

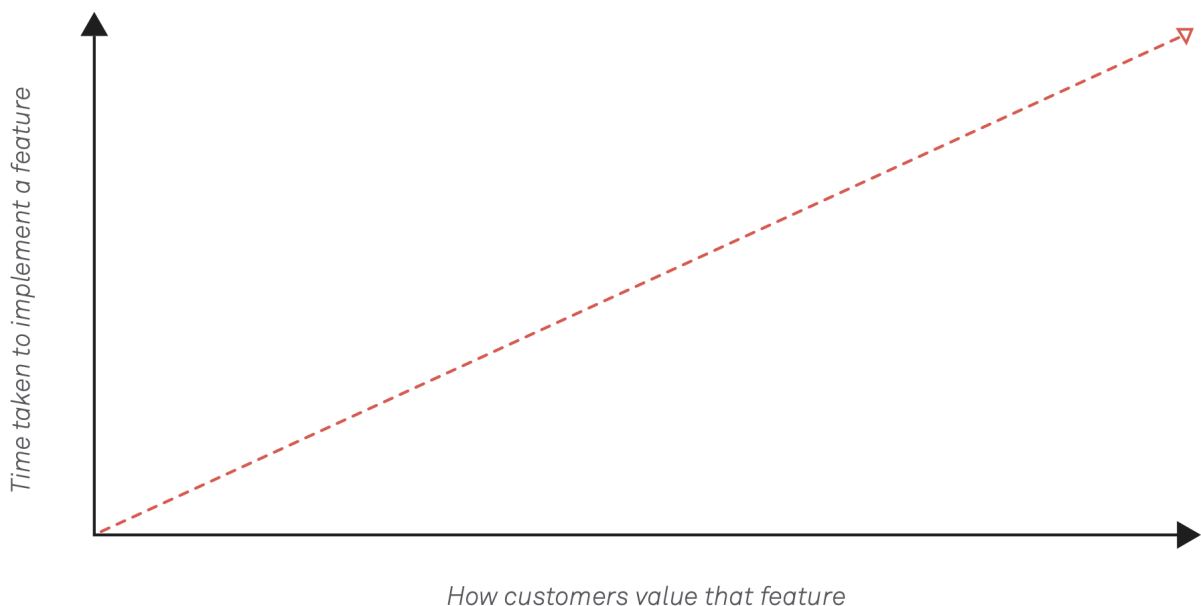
Функції з погано визначеним обсягом робіт не виконують очікувань користувачів, випускаються пізно, якщо взагалі випускаються, і не додають у продукт нічого, крім заплутаності.

Слизька доріжка ігнорування правил

Спокусливо іноді порушувати правила зі списку вище. Можна припустити, що рідкісне порушення правил шкоди не принесе. Можливо, це так у теорії, але це програмне забезпечення. Реальність інша. Кожному хорошому продакт-менеджеру потрібен свій чек-лист, коли говорити "так", а коли "ні". І слідувати йому без винятків.

Продукти роздуваються по одному неправильному рішення за раз. Не існує одного вибору, в результаті якого з'явиться повідомлення: "Ваш продукт тепер занадто складний". Ви ніколи не отримаєте email: "Минулого вівторка, о 14:03 стався підрив вашого продукту". Роздуття, складність і підрив можна побачити тільки в дзеркало заднього виду, тому завжди думайте про цей ризик, кажучи "так".

Функціонал і "заздрість до фізики"

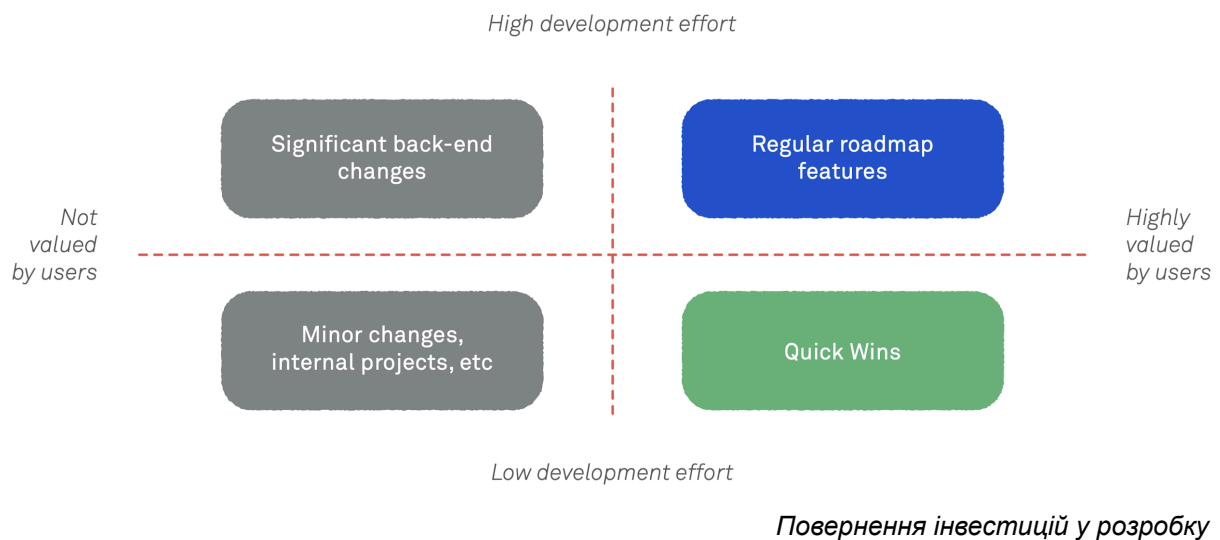


Кожному сподобалося б, якби це було правдою

"Заздрість до фізики" - термін, що використовується для критики теорій у програмному забезпеченні та гуманітарних науках, які намагаються бути більш математично строгими, якою стала свого часу фізика.

У нашому випадку цим терміном можна описати бажання дизайнерів, розробників і продакт-менеджерів створити закони в системі, яка просто їм не підкоряється. Кожному сподобається, якщо повернення інвестицій буде пропорційним витраченому часу. Планування продукту стало б таким легким. Але проблема в тому, що ви можете витратити 5 тижнів, працюючи над "забійною" функцією, щоб потім побачити як її проігнорували користувачі. І в той же самий час, додати один звуковий ефект, що говорить "привіт" або зробити миготливу іконку в браузері, як раптом кожен почне говорити про вас. Це не раціонально, але коли воно було таким?

Замість того, щоб заздрити наукам, у яких немає такої ірраціональності, краще подивимося, як можна це використати. Наступного разу, коли плануватимете роботу на кілька місяців, нанесіть ваші функції на таку діаграму...



Усе, що знаходиться в правому нижньому кутку - особливе. Це функції, де повернення сильно перевищує зусилля. Деякі приклади, які мені доводилося бачити:

- ❑ Змінити текст повідомлень у застосунку, щоб зробити його більш особистим, смішним, точним або корисним.
- ❑ Додати кілька комбінацій клавіш, щоб досвідчені користувачі працювали продуктивніше.
- ❑ Запам'ятовувати останній активний проєкт і відкривати його під час наступного входу користувача.
- ❑ Додати додаткову інформацію та посилання на дії в поштових повідомленнях.

Коли використовувати швидкі перемоги

Клієнти не оцінять все, що ви робите в продукті. Деякі необхідні завдання, як-от бекапи, рефакторинг, оптимізація, поліпшення інфраструктури не дають негайного ефекту. Але це не означає, що їх можна ігнорувати. Секрет у такому плануванні дорожньої карти, щоб не було довгих проміжків часу, коли клієнтам здається, що застосунок не оновлюється.

Це той випадок, коли допоможуть швидкі перемоги. Ви можете розпланувати швидкі перемоги так, щоб покрити проміжки часу, коли довго немає видимих змін. Таким чином, клієнти бачать, що продукт постійно розвивається. А ви в цей час займаєтеся великими функціями або проблемами бекенду. Нічого немає гіршого, якщо користувачі бачать "павутиння", що з'являється в застосунку.

Випуск нових функцій

Релізи або виходять, або ламаються, і їх дуже важко зробити добре. Удвічі важче, якщо ви змінюєте звичні процеси користувачів. Для багатьох B2B додатків - це чиясь робота використовувати додаток. Існують тисячі людей, чия робота використовувати Інтерком. Тому, коли ми плануємо зміни в поведінці, або додаємо нове, ми ставимося до цього не поверхово.

Для цього в нас є процес прогресивних релізів, що складаються з кількох кроків, коли ми включаємо нові функції лише для частини користувачів. Ваш процес може відрізнятися, але для нас цей підхід себе добре зарекомендував.

Тестування командою

Перший крок, випустити реліз для роботи з реальними даними, але зробити його доступним лише для команди, яка створила його. Це дає змогу уникнути випадку, коли занадто багато людей повідомляють про одні й ті самі помилки. Сама команда знає свій реліз, і можуть сказати, чи це помилка, чи просто, щось не готове.

Тестування компанією

Коли команда впевнена, що функція готова, вони роблять її доступною для всієї компанії, яка негайно включає її у свій процес роботи. Ми не робимо спеціальних пояснень або вибачень. Функція випускається так, як вона була б випущена для клієнтів. Це дає змогу побачити, що незрозуміло користувачам. Якщо вже Інтеркомівці не розуміють як користуватися, які шанси, що клієнти зрозуміють?

Обмежений бета реліз

Перший публічний реліз робимо для менш ніж 10% користувачів. Ми підтримуємо групу "Довіrenих тестувальників" для цих цілей. Ми оголошуємо реліз як обмежена бета, і просимо надіслати відгук, коли вони покористувалися функцією. Останнє, дуже важливо запам'ятати. Відгук про те, як працює функція, може бути тільки від людей, які скористалися нею. Не від тих, хто потестував або погрався з нею. Джейк Кнапп у статті "[Реакція важливіша за відгук](#)" навів приклад, коли людина, намагаючись бути корисною, вигадус

поліпшення, хоча її реакція була іншою. Спекуляціям і дружнім стосункам немає місця в розробці продуктів.

На цьому етапі ми перевіряємо:

- ☐ Знахідність - чи знаходять люди функцію?
- ☐ Залучення - чи користуються вони функцією?
- ☐ Прийняття - чи стала функція частиною їхнього процесу?
- ☐ Сценарії - як функція використовується? Які сценарії популярні?
- ☐ Бар'єри - Хто не користується? Чому? Що заважає?

Під час обмеженого бета релізу ви можете перевірити, що користувачі знаходять функцію, відшліфувати маркетингові повідомлення, щоб вони містили справжню цінність очима клієнтів, і прибрати бар'єри використання і прийняття.

Публічний реліз

У цей момент функція доступна всім користувачам, у кого відповідний тарифний план. Тепер саме час подумати, як повідомити їм про реліз. Зазвичай не всі функції доступні для всіх користувачів - працює звичайна дискримінація за ціною, тому щонайменше ви розділяєте повідомлення на "мають" і "не мають" доступ.

Тим, хто має доступ до функції, потрібно продати її. Анонси функцій часто зосереджуються на тому "що ми зробили" замість того "що ви можете зробити за допомогою функції". Якщо хочете, щоб люди користувалися продуктом, надихайте їх, показуючи, що вони можуть досягти.

Тим, кому недоступна функція, потрібно продати її як привід перейти на інший тариф. Це можна зробити за допомогою безкоштовного пробного періоду, наприклад, "Спробуйте преміум тариф протягом одного місяця безкоштовно".

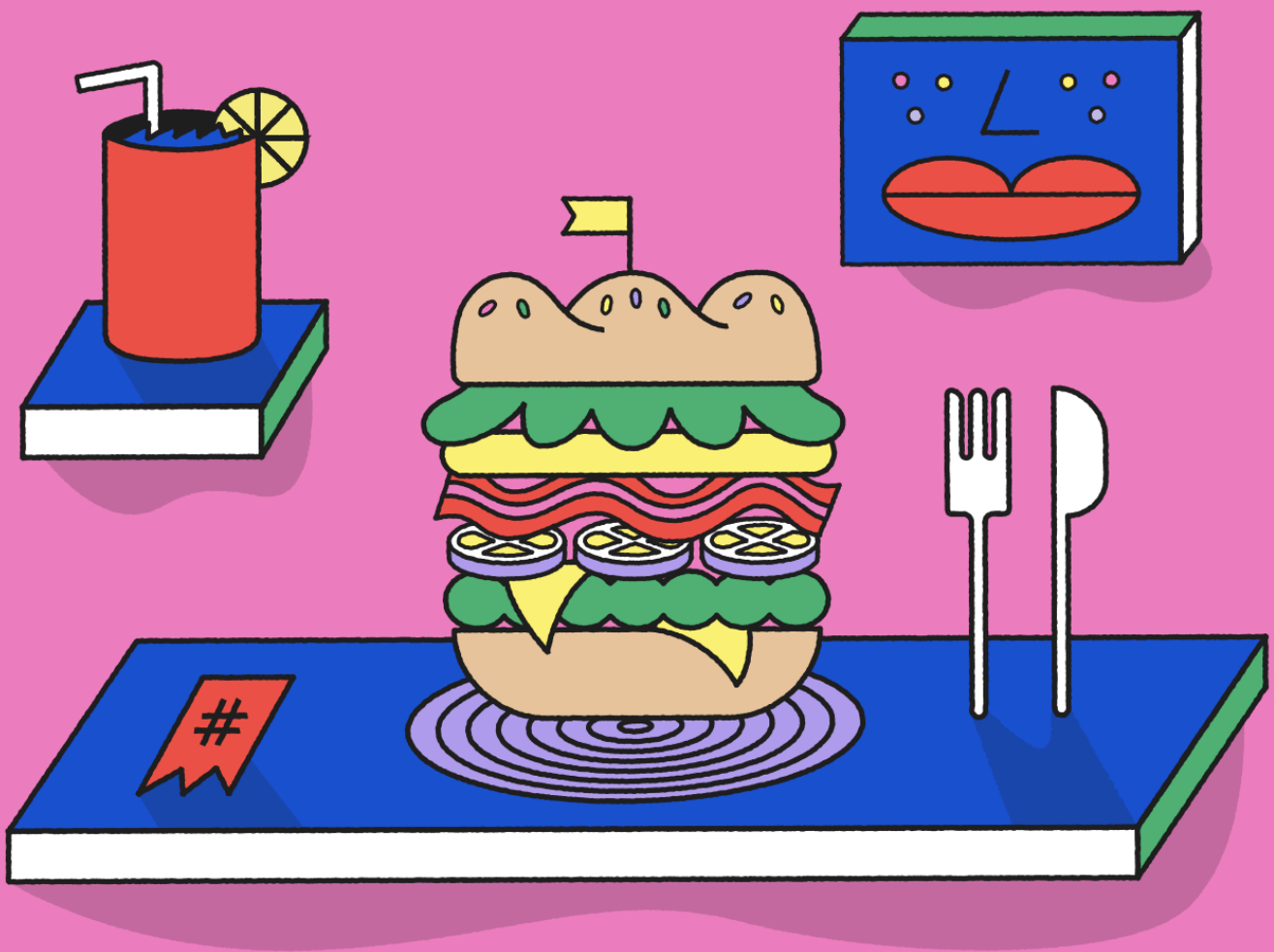
Розклад повідомлення про випуск

Щойно функцію було випущено, легко забути про неї і перейти до задоволення розробки наступної. Але функції без залучення - це насіння для роздутих додатків. Під час випуску кожної значної функції вам необхідний план для тих, хто буде використовувати її, і для тих, хто ні.

Коли функція добре використовується, ви збираєте сценарії використання для майбутніх маркетингових матеріалів. Залежно від ваших планів, ви можете зібрати ще й відгуки.

Коли люди не користуються функцією, ви надихаєте їх, навчаєте, залучаєте. Єдиний прийнятний результат у цьому випадку - вони починають користуватися або кажуть чому не можуть.

Не варто пишатися великим продуктом із безліччю функцій. Що більше площин дотику має продукт, то важливіше ганятися за залученням і вчитися на своїх помилках. Особливо це важливо, коли ви молоді та недосвідчені. Ви можете випускати нову функцію щотижня, але краще спрямувати цю енергію на збільшення використання продукту, а не на зростання продукту.



Глава 4. Що робити, щоб функцією користувалися

Вирішити які функції потрібно додати і, потім, реалізувати їх - лише початок справи. Нові функції це просто програмний код, який збирає віртуальний пил,

поки ними не почали користуватися клієнти. Як ішлося в першому розділі, якщо функцією не користуються - час ухвалити важке рішення: видалити або зробити її кращою. Не прийняли рішення, і ви ризикуєте віддати клієнтів конкуренту, у якого продукт краще сфокусований і працює ефективніше.

Успішний випуск функції вимагає тих самих навичок, що й під час запуску продукту. Різниця в тому, що потрібно врахувати всі питання із сумісністю з попередніми рішеннями і заспокоїти наявних клієнтів. Це не просто.

Більшість нових функцій провалюються. Ви просто не помічаєте цього. У цьому й проблема. "Нові поліпшення" сидять неоцінені, невикористані, і врешті-решт, забуваються. Ласкаво просимо в програмне забезпечення. Покращення даються важко.

Ви витрачаєте місяці на дослідження. Аналізуєте демографію, метрики, етнографію, аналітику, психографію, і щось там ще. Дні, тижні, місяці зустрічаєтеся з клієнтами. Зустрічаєтеся з їхніми батьками, годуєте їхніх собак, годуєте собак їхніх батьків. Ви робите все, щоб зрозуміти, що хочуть клієнти. Не те що вони говорять. Не те що вони думають або просять. А те, що вони дійсно хочуть. І, потім, ви йдете і створюєте це!

І воно все одно провалюється. Це жорстка гра.

Чому новий функціонал зазвичай провалюється

Річ у тім, що функцію, просто, можуть не знайти. Про це часто забувають у поспіху підготовки до релізу. Мова не про викладання коду на сервери, або про галочку в дорожній карті. Мова про те, щоб вашим продуктом користувалися. Тому наступного разу перед випуском нової функції поставте собі такі запитання...

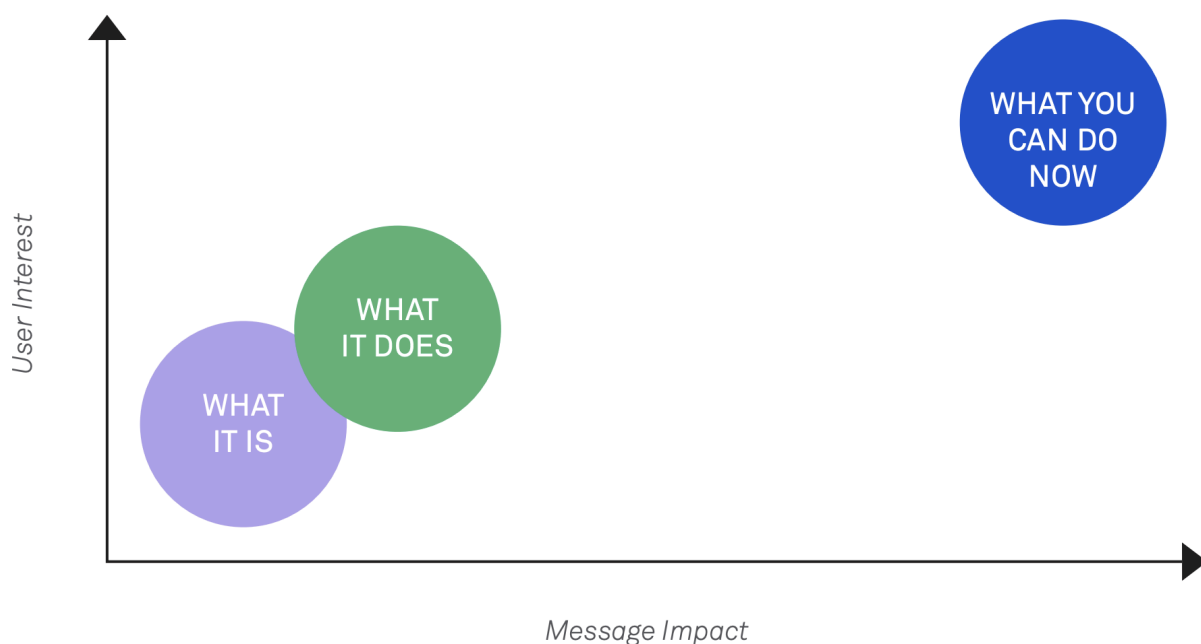
1. Чи кожен може легко знайти і зрозуміти цю функцію?

Цікавий факт: коли Microsoft запитала користувачів, що б вони хотіли додати в Office. Виявилося, що 90% функцій, які попросили користувачі, вже існували. Microsoft припустила, що це була проблема обізнаності, незважаючи на запуск "[стрічки інструментів](#)", яка нібито вирішувала цю проблему. Річ у тім, що стрічка показувала всі функції, а значить жодної. Як я вже сказав, це не просто.

Коли ви розробляєте першу версію продукту, ви прагнете, щоб він був закінченим. Тому ви не плануєте розширення і не залишаєте місця для майбутніх функцій. Коли ви, все ж таки, додаєте нові функції, ви поспіхом звільняєте місце, і користувачам складно знайти їх. Якщо постійно виникають питання, де знаходиться функція, у вас проблеми з виявленням. Як наслідок, випуски нових функцій провалюватимуться, або щонайменше вимагатимуть навчання, щоб знайти і прийняти їх.

2. Ви показуєте користувачам, що ви зробили або, що вони можуть зробити?

Якщо сказати користувачам щось на кшталт: "переписали з нуля", "на базі HTML5", "адаптивний дизайн", для них це не матиме сенсу, крім випадку, коли ви продаєте розробникам. Нікого не цікавить, що ви зробили або як ви це зробили. Ваших клієнтів турбує, що вони можуть зробити.



Вплив анонсів функціоналу

Зосередьтеся у вашому повідомленні на тому, що користувачі можуть досягти завдяки цій функції, і ви отримаєте їхню увагу.

3. Ваші анонси в контексті?

Нові функції, особливо невеликі доповнення, з'являються в продукті без особливого контексту. Ваша мета ніколи не має бути "просто випустити". Мета має бути - "щоб користувалися". Повідомлення поштою - невдалий варіант для таких анонсів, тому що вони часто приходять у невідповідний час і в неправильному контексті. Слушна нагода для анонсу поліпшень - це коли користувач перебуває в застосунку і готовий використовувати поліпшення. Для цього ідеально підходять анонси всередині застосунку, які спрацьовують за певними правилами та подіями.

4. Як завтрашні нові користувачі дізнаються про це?

У міру зростання продукту, він переходить межу, коли його не можна вивчити за один присід. Це не проблема, тому що не кожна функція потрібна відразу. Функції актуальні тільки коли вони вирішують проблеми. Іншими словами, вам не потрібні мітки для файлів до того, як файли завантажені. Вам не потрібне злиття міток до того як у вас буде їх багато. Тому, повідомляти новим користувачам як робити злиття міток - приклад повідомлення, показаного невчасно.

Щоб навчити користувачів як користуватися функцією в потрібний час, є сенс використовувати спрямовані повідомлення. Наприклад, ми розповідаємо про збережені відповіді в Інтерком, після того як було надіслано кілька відповідей. Ми розповідаємо про гарячі клавіші, коли продуктом користувалися тривалий час. Таким чином повідомлення набувають сенсу.

5. Чи плануєте отримати зворотний зв'язок від користувачів і не-користувачів?

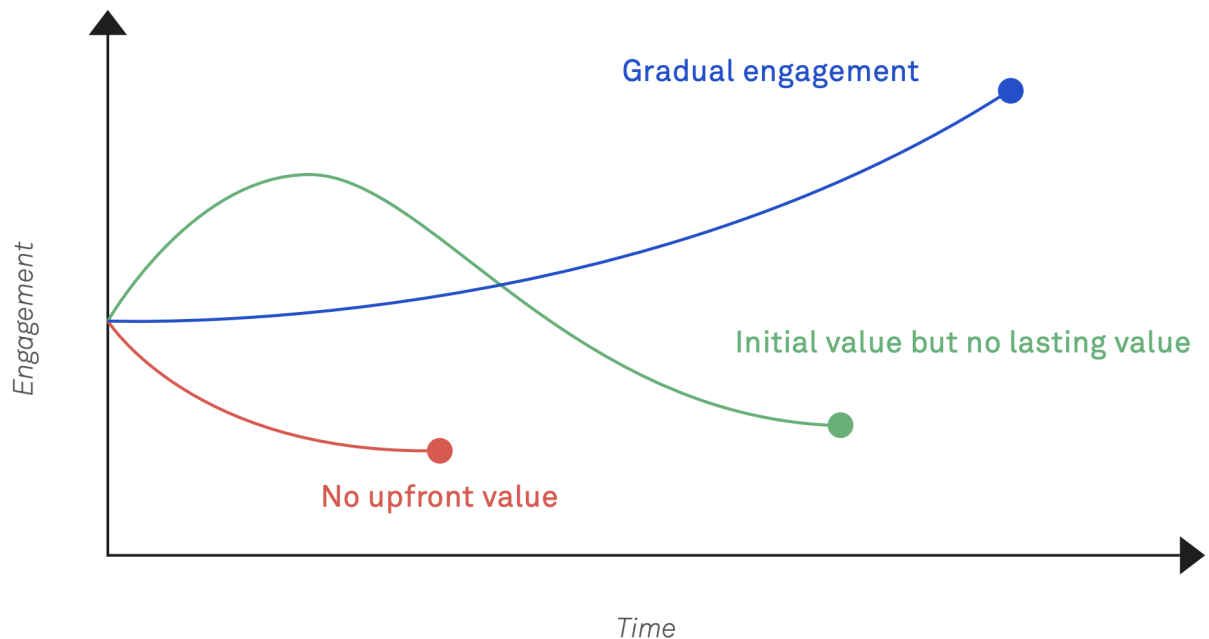
Щойно функцію випущено, вам необхідно отримати зворотний зв'язок від тих, хто користується нею, щоб зрозуміти як, чому і коли нею користуються. І, потім, знайти способи збільшення використання. Ось кілька запитань, які я ставив клієнтам про нову функцію:

- ☐ Коли ви виявили її? Що ви подумали? Чи почали ви використовувати її відразу? Що привернуло вашу увагу?
- ☐ Чи потрібно було вам читати документацію? Чи була вона достатньою?
- ☐ Чи були бар'єри у використанні функції?
- ☐ Чи є час протягом вашого робочого дня, коли ви хотіли б використовувати її, але не можете?
- ☐ Чи говорили ви кому-небудь для чого ви використовуєте її? Що вони сказали?

Однаково важливо, якщо не більше, отримати зворотний зв'язок від тих, хто не використовує функцію, і зрозуміти, що заважає користуватися. Часто, ви виявите перешкоди, які легко усунути, наприклад: "Я весь час забуваю спробувати функцію", "Я не знаю, чи користується функцією будь-хто ще в нашій компанії", "Мені потрібно буде отримати CSV-файл із даними" тощо. Усе це розв'язувані проблеми, щойно ви їх зрозумієте.

Ваш процес розробки повинен допускати, що ви можете зробити щось не так. Так багато дорожніх карт і спринтів ігнорують цей факт. Дизайнери і продакт-менеджери помиляються весь час. І коли це трапляється, у них є вибір: виправити або проігнорувати і перейти до наступної функції. Виберіть другий варіант достатню кількість разів, і ви теж будете розповідати, що 90% прохань від користувачів уже є у вашому застосунку. Сподіваюся, ви любите "стрічки інструментів"!

Збільшення залученості користувачів



Види залученості з плином часу

Визначення "залучений користувач" варіюється від продукту до продукту. Для застосунку списку справ, залучений користувач повинен заходити в застосунок щодня, щоб відзначити виконані завдання. У той же час для програми створення рахунків-фактур, залучений користувач може входити лише раз на місяць. Не існує єдиного точного визначення залучення для всіх продуктів.

На відміну від таких метрик як: перегляди сторінок, кількість відвідувачів, відвідувачі, що повернулися, та конверсії, не існує аналітичного застосунку, який одразу міг би показати те, що вам потрібно. Але не поспішайте ігнорувати залученість.

Google+ заявляв, що у них 170,000,000 користувачів. Це число виглядає вражаюче в новинах. Але при цьому ігнорується реальна проблема. Майже ніхто з цих користувачів не залучений. Це просто люди, які клікнули на посилання "ОК" коли перед ними з'явився Google+. Це нагадує як новинні сайти накручують свої перегляди, використовуючи слайд-шоу або багатосторінкові статті. У якийсь момент ви повинні розібратися кого ви обманюєте.

Залученість лише невелика частина пазла. Її часто ігнорують, хоча тут можна отримати багато швидких перемог. Ось три способи збільшити залученість користувачів.

1. Справити гарне перше враження

Щодня потенційні клієнти бачать ваш інтерфейс уперше. Про це часто забувають, особливо команди, які розробляють дизайн для себе. У першого екрана, який бачить користувач, є три важливі завдання:

- ☐ Пояснити як працює додаток.
- ☐ Надихнути почати користуватися.
- ☐ Дати знати, як отримати допомогу, якщо буде потрібно.

Веб-застосунки, які не роблять нічого з перерахованого вище, отримують те, для чого вони спроектовані - користувачі йдуть і ніколи не повертаються.

Існує багато чудових способів привітати користувача в застосунку та допомогти зорієнтуватися. Заготовки документів, покрокове керівництво, вигадані дані тощо. Як перший крок, ми радимо клієнтам Інтерком надіслати повідомлення з привітанням кожному користувачеві їхнього сайту, який знову зареєструвався. Це добре працює як початок бесіди, що своєю чергою збільшує конверсію.

Просте спілкування з вашими користувачами - чудовий спосіб, щоб вони ставили більше запитань, пробували нові функції та оглядалися. Включаючись у діалог, вони з більшою ймовірністю скажуть щось на кшталт "Як мені відправити email неактивним користувачам" або "Як використовувати мітки?".

У кращому випадку ви отримаєте більше клієнтів, у гіршому - дізнаєтеся чого не вистачає або, що не зрозуміло в застосунку.

2. Поступово розкривайте деталі продукту

Кожен вартісний застосунок містить функції, які не потрібні одразу. Це може включати повідомлення електронною поштою, інтеграції зі сторонніми додатками, гарячі клавіші.

Ці зручності не можна використовувати відразу. Кого хвилює експорт даних до того, як з'явилися дані, або гарячі клавіші до того, як почали користуватися додатком?

Багато власників продуктів прагнуть показати ці зручності за допомогою невчасно надісланих листів, документації або поширених запитань. Жоден із цих підходів не працює досить добре.

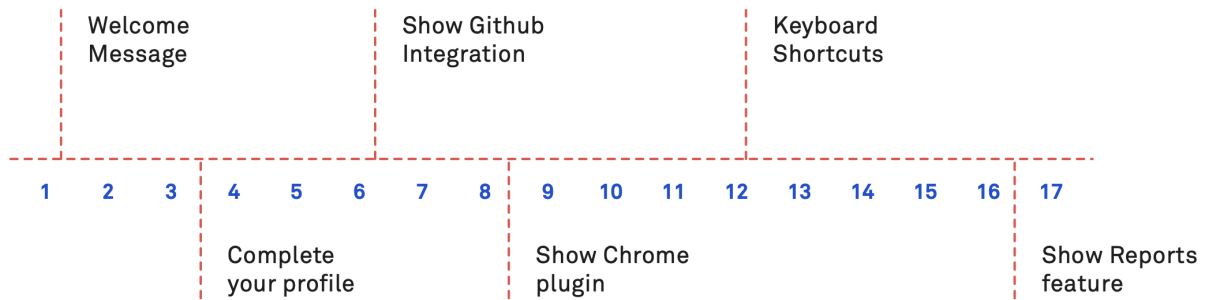
Листи приходять не в контексті і не вчасно, і з більшою ймовірністю відволікають користувача, ніж цікавлять його. Реакція - помістити лист в архів і перестати читати наступні повідомлення.

Показувати функції за допомогою поширених запитань або документації означає, що їх побачать тільки якщо у користувача проблема і він звернувся до документації. Це не ідеальний час відволікати його з іншими функціями.

Ми рекомендуємо нашим клієнтам створювати розклад надсилання повідомлень, щоб поступово показувати деякі функції. Коли ви добре розумієте ваших користувачів, ви зможете вирішити, які другорядні функції радують і в який момент вони корисні.

Після виділення таких функцій, це питання своєчасних повідомлень. Інтерком дає змогу регулювати надсилання протягом серії сесій. Як наслідок, щоразу, коли користувач входить до застосунку, ви показуєте йому все більше й більше приводів залишитися з вами.

Нижче приклад розкладу для залучення від одного з наших клієнтів:



Розклад надсилання повідомлень за кількістю входів у додаток

Завжди, в кожному повідомленні пишiть, що ви готові допомогти, якщо маєте запитання. Це ключ до отримання зворотного зв'язку, який допомагає покращувати процедуру залучення.

з Анонс функцій і поліпшень прямо в застосунку

Користувачі не помічають коли ваш процес розробки сповільнюється. Вони заходять у застосунок, щоб користуватися продуктом, а не відстежувати вашу розробку. Однак, якщо в продукті нічого не змінюється довгий час, їх легко відволікти новиною про випуск нової функції конкурентом, корисної або дріб'язкової.

Єдине, коли користувачів цікавлять функції - коли вони перебувають у застосунку. Це якраз те місце, де їх потрібно анонсувати.

Ми помічали десятикратне збільшення в комунікації з користувачами, коли використовували повідомлення в додатку порівняно з електронною поштою. При цьому були ще плюси. Наприклад, коли ми повідомили про функцію поділитися розташуванням користувачів на карті, наші клієнти одразу клікнули на посилання і почали ділитися картами. Я не уявляю, щоб від email був би подібний ефект.

Створення розкладу надсилання повідомлень

Сценарій: Ви продакт менеджер, який хоче збільшити залученість користувачів, коли вони перебувають поза офісом. Для цього ви розробили мобільну версію вашого застосунку. Тепер вам потрібно розповісти користувачам про нього.

Слабкий спосіб - це розіслати email усім зареєстрованим користувачам, просто зараз, з посиланням на мобільний застосунок. Що буде далі?

Деякі клієнти отримають його коли перебувають в офісі. Хтось отримає користуючись вашим веб-додатком на телефоні. Отримають і ті, хто не користувався вашим продуктом кілька років. Хтось о 4 ранку, інші о 10 ранку.

Ваша аналітика покаже результати, що 4.43% людей клікнули за посиланням у листі. І ви будете розчаровані, тому що розсилка була великим і стресовим проєктом.

Чи можна зробити краще?

Альтернативні підходи

Ви можете розділити клієнтів на дві групи: хто заходив у застосунок із телефона і хто ні. Це дасть змогу створити відповідне повідомлення для кожної групи.

Ви можете створити постійне повідомлення у веб-додатку для тих, хто заходить з мобільного телефону, щоб вони знали про мобільний додаток. Ви можете надіслати email усім, хто вийшов із застосунку, тому що вони ще пам'ятають про ваш застосунок. Ви можете створити кумедне повідомлення, з жартами про конкурентів і включити в нього смішні картинки. Завдяки всьому цьому, люди починають говорити про вас.

Ви можете перейняти у Slack їхній стиль повідомлень про релізи. Ці повідомлення цікаво і смішно читати. Наприклад:.

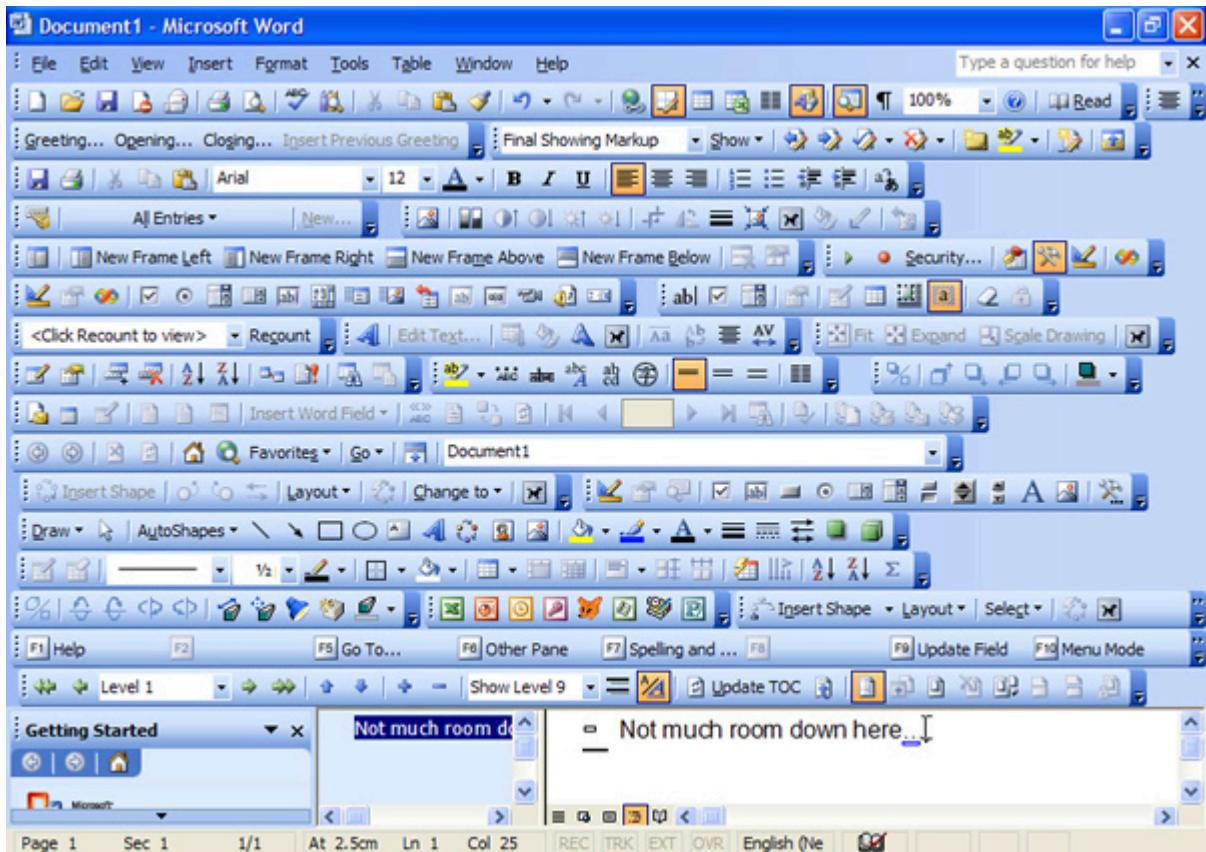
"Виправлено: Ми виправили вразливість, завдяки якій Gif можна було вимовляти як "Жиф". **Виправлено:** Коли клавіатуру прибираєш свайпом донизу, вікно з повідомленнями раптом прокручувалося назад догори, це було кумедно, але не те що нам потрібно."

Усі ці варіанти виходять за рамки "Давайте надішлемо всім листа", який обирають багато компаній.

Не соромтеся видаляти функції

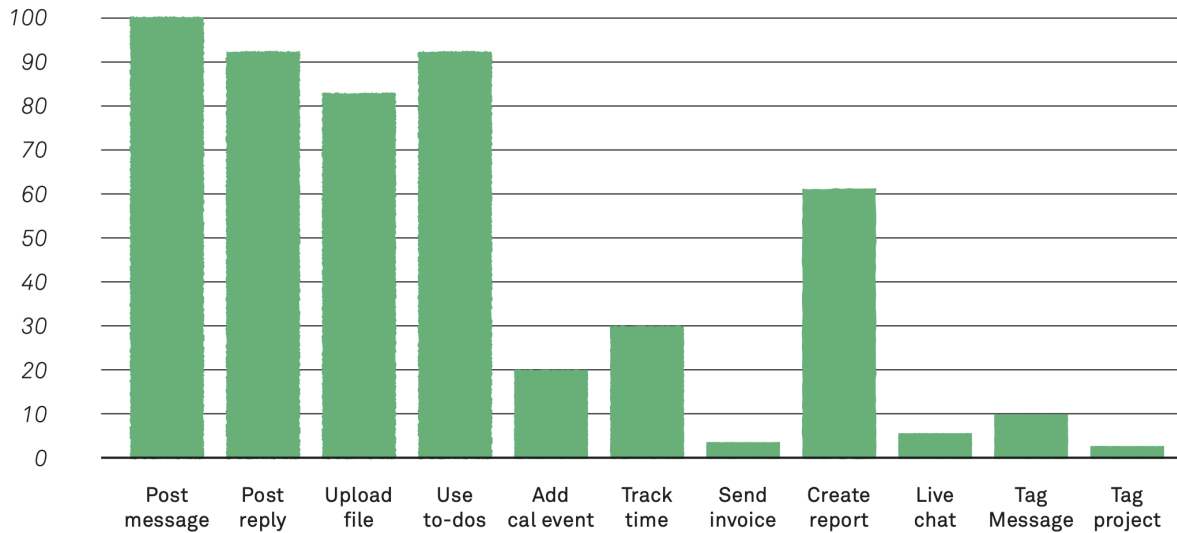
Коли ви думаєте про перевантажені функціями продукти, що спадає на думку? Нескінченні панелі інструментів, налаштування та опції, вірно?

Був час, коли жодна розмова про призначений для користувача інтерфейс або стратегію продукту не обходилася без згадки скріншота, на якому був зображений Microsoft Word з усіма панелями інструментів, залишаючи лише невелике місце внизу для документа. Це чудовий приклад того, як важлива простота.



Продукти на кшталт цього показують, що перевантаженість функціями обходиться недешево, ось чому вам треба легко розлучатися з функціями. Користувачі просто не в змозі освоїти всі ці функції. Проблема в тому, що скріншоти, на кшталт Microsoft Word, не говорять, що робити далі. Звісно, тут занадто багато непотрібного, але що робити з цим?

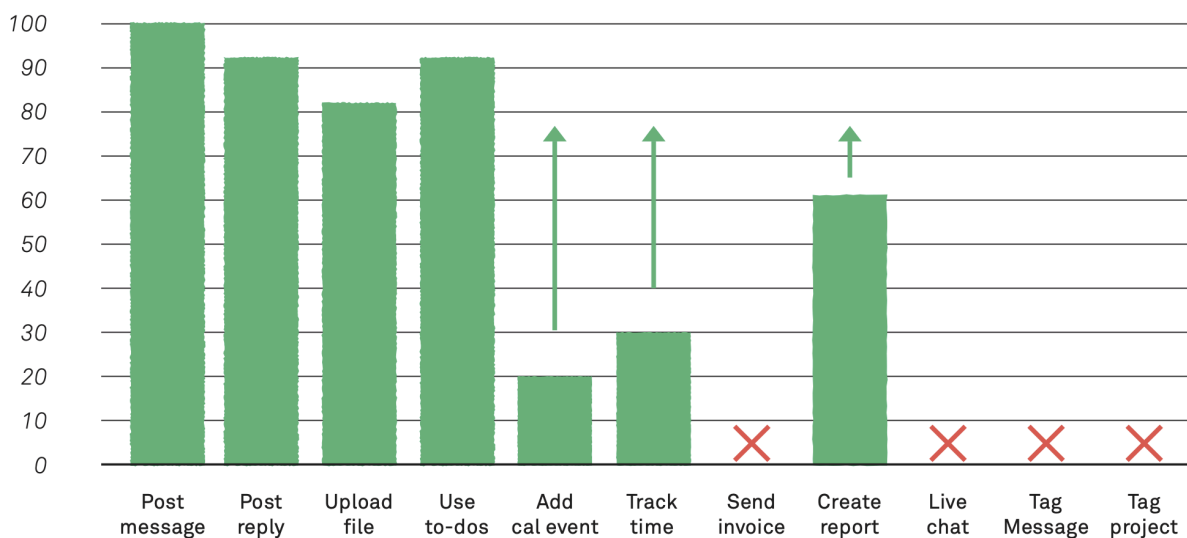
Щоб розв'язати проблему перевантаженості функціями, потрібно визначити, якими з них користуються більшість клієнтів, а якими ні. Знову час для аудиту використання функцій, про який розповідалося в першому розділі.



Виконайте повторний аудит функціоналу

Якщо у вас довгий хвіст маловикористовуваних функцій - ви збилися з курсу або ваша стратегія "створювати все, що попросять". Друге може звучати дивно, але продукти на кшталт Microsoft Word добре впоралися з цим планом. Але ви, ймовірно, не створюєте Microsoft Word. Якщо ви хочете створити добре визначений продукт, вам потрібно без жалю видаляти функції.

Дійте



Функції не приймають з різних причин: користувачі не бачать користі в них, вони занадто складні, вони заховані десь у продукті, багато інших. Ось чому важливо говорити "ні". Коли ви бачите, що лише 8% користувачів використовують її, потрібно ухвалити рішення: видалити чи зберегти.

Видаляючи функцію, ви можете приховати її від нових користувачів, а наявним - надіслати повідомлення, пояснивши своє рішення і даючи час, щоб вони могли підготуватися. Або ви просто можете прибрати її з інтерфейсу і ніколи більше не згадувати про неї. Обидва варіанти працюють, і різняться кількістю додаткової роботи. Що більше користувачів, то акуратнішими мають бути дії з видалення.

Збереження функції, яку мало використовують, потребує спілкування. Вам необхідно виділити користувачів, які користуються нею і з'ясувати чому. Знайти користувачів, які не користуються і, також, з'ясувати чому. У цьому може допомогти такий інструмент як Інтерком, який дає змогу таргетувати користувачів і почати спілкування за лічені секунди.

Буває, достатньо анонсувати функцію в застосунку. У деяких наших клієнтів прийняття функцій зросло на 30% після надсилання повідомлення в потрібний момент роботи в застосунку. Також у нас є клієнти, які отримали достатньо відгуків від їхніх користувачів, щоб стало зрозуміло, що функція не потрібна, і її видалення буде безболісним.

Хороші власники продуктів допускають лише кілька невдалих функцій, а великі - видаляють усі негайно.

Висновок: у процесі

Нам довелося ухвалювати складні рішення про те, які думки і частини включити в цю книгу. Ми намагалися сфокусуватися на статтях, які були б корисними і професіоналам, і тим, хто нещодавно прийшов у

продакт-менеджмент. Наше бачення про те, як створювати продукти і керувати ними все ще в роботі і це наша перша спроба узагальнення матеріалів.

Якщо ця книжка була корисною для вас, заходьте на наш блог "[Всередині Інтерком](#)", де ми регулярно ділимося думками про продакт-менеджмент і пов'язані теми, як-от дизайн, бізнес стартапів і успіхи клієнтів.

У нас немає всіх відповідей, але зростає пристойний список запитань.

Дякую, що прочитали цю книгу

Поділіться оригіналом книги з друзями -

<https://www.intercom.com/resources/books/intercom-product-management>

Або цим перекладом

Не забудьте підписатись на мій [телеграм канал](#)!