

# LIFETIME VALUE:

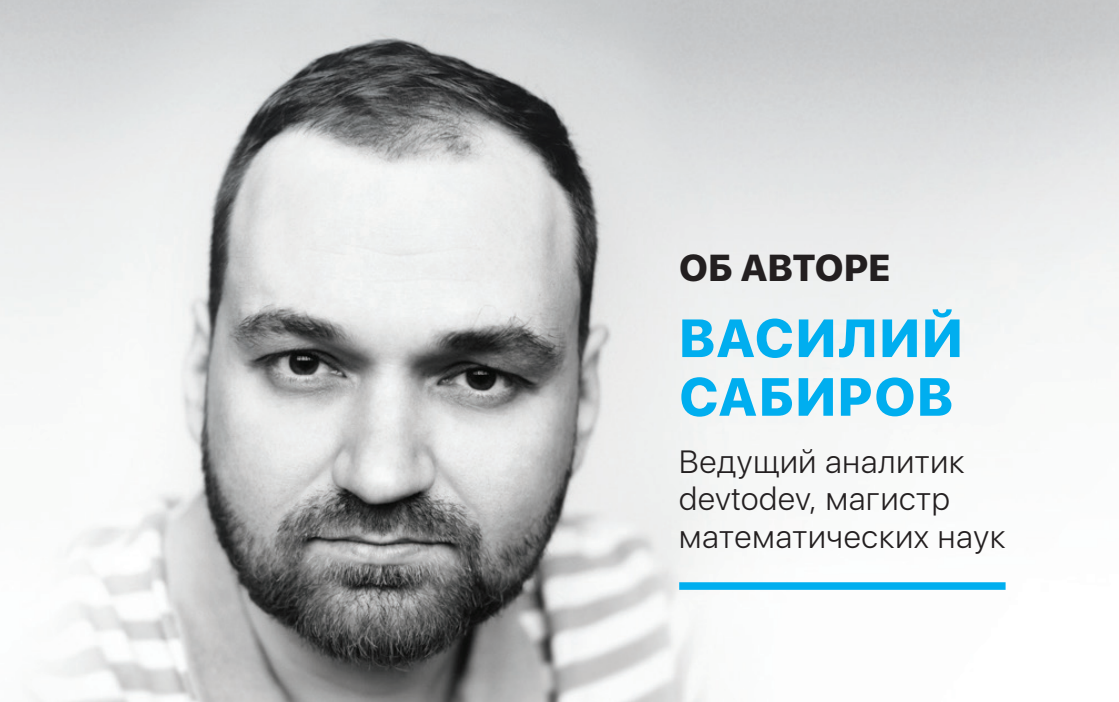
ГЛАВНАЯ МЕТРИКА  
ПРОЕКТА



**devtodev**

A full-cycle analytics solution for game developers

[www.devtodev.com](http://www.devtodev.com)



**ОБ АВТОРЕ**

## **ВАСИЛИЙ САБИРОВ**

Ведущий аналитик  
devtodev, магистр  
математических наук

---

Василий занимается аналитикой с 2011 года, специализируется на приложениях и игровых проектах.

В [devtodev](#) Василий помогает сделать тему игровой аналитики популярной. Он автор книг, статей, вебинаров и образовательных курсов, которые дают возможность разобраться в тонкостях игровой аналитики и улучшить проекты.

Также Василий ведет ряд игровых проектов в рамках аутсорса аналитики devtodev. Своими консультациями и экспертными советами он уже помог более чем 30 компаниям увеличить свои показатели.

# СОДЕРЖАНИЕ

1. Введение

2. Что такое Lifetime Value

3. Для чего необходимо знать LTV?

4. Как считать LTV?

➤ Метод 1. Постфактум

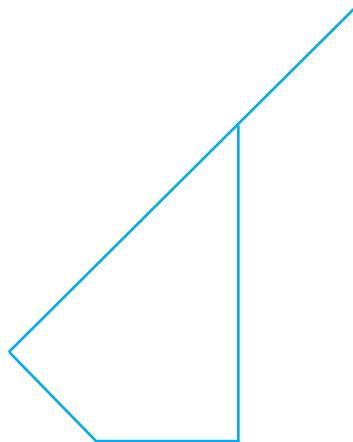
➤ Метод 2. «Метод Шарикова»

➤ Метод 3. Через Lifetime и ARPU, простой способ

➤ Метод 4. Через Lifetime и ARPU, сложный способ

➤ Метод 5. Через накопительный ARPU

5. Заключение



# 1 ВВЕДЕНИЕ

Привет! Мы аналитическая компания [devtodev](https://devtodev.com), и сейчас вы читаете книгу «Lifetime Value: главная метрика проекта».

После её прочтения вы сможете ответить на следующие вопросы:

- Что такое Lifetime Value (LTV)?
- Для чего необходимо знать эту метрику?
- На какие показатели она оказывает влияние?
- Какие способы расчёта LTV существуют?

Если вы открыли эту книгу и уже читаете введение, значит, эта тема действительно вам интересна. Мы попытались свести воедино все наши знания и навыки работы, касающиеся LTV.

Поэтому очень надеемся, что эта книга будет для вас интересной и полезной.



ПРИЯТНОГО  
ЧТЕНИЯ!

# 2 ЧТО ТАКОЕ LIFETIME VALUE

**LTV, она же Lifetime Value, она же Customer Lifetime Value, она же CLV,** — это показатель ценности клиента, которую он приносит за все время в проекте. Она показывает, сколько денег в среднем принесёт один пользователь за всё время использования продукта.

Показатель LTV универсален, рассчитывается и в веб-аналитике, и в мобильной аналитике. Метрику считают для большинства видов продуктов, будь то кофейни Starbucks, мобильные операторы, банки, SaaS-продукты или игры.

LTV по проекту в целом можно пользоваться, но более точный результат даёт показатель, рассчитанный по отдельным каналам.



LTV > CPI

Это главная формула всего анализа трафика и главное условие эффективности привлечения. Пользователь должен приносить больше денег, чем было потрачено на его привлечение в проект.

Под CPI (Cost Per Install) в данном случае мы имеем в виду среднюю стоимость привлечения одного пользователя по всем каналам сразу.

Если вам привычнее аббревиатура CPA (Cost Per Acquisition), традиционная для веб-продуктов, используйте ее в дальнейших формулах в этой статье.

# 3 ДЛЯ ЧЕГО НЕОБХОДИМО ЗНАТЬ LTV?



## LTV ПОМОГАЕТ ОЦЕНИТЬ КАЧЕСТВО ИСТОЧНИКОВ ТРАФИКА

На самом деле, что средний CPI, что средний CPA — показатели достаточно условные, ведь как правило мы платим одному партнёру сумму А, другому — сумму В, третьему — сумму С, а общий средний CPI — это, скорее всего, значение, которое не равно ни А, ни В, ни С.

**LTV лучше считать отдельно по каналам привлечения, по кампаниям.**

При этом может получиться так, что общая средняя LTV будет больше общего среднего CPI, однако в разрезе каналов привлечения будут и неэффективные каналы, где это условие не выполняется. Что делать в такой ситуации?

Можно, конечно, сразу отключить попавший в немилость источник трафика. Однако эффективнее будет детально изучить его, «разрезать» по кампаниям, странам и платформам, отключить те из них, где LTV меньше, чем CPI.

А ещё лучше — ввести подобный анализ в регулярную практику и отключать неэффективные SubID, которые появляются у поставщика трафика.



## LTV ПОЗВОЛЯЕТ ОТСЛЕЖИВАТЬ ДИНАМИКУ ПРОЕКТА

LTV основывается на значении многих метрик. На нее влияют и удержание пользователей (Retention), и доля платящих (Paying Share), и доход с платящего пользователя (ARPPU). Вместо того, чтобы отслеживать динамику нескольких метрик, вы можете отслеживать динамику LTV, — это покажет вам, насколько эффективны изменения, которые вы вносите в проект.

Если LTV растёт от месяца к месяцу — прекрасно, продолжайте в том же духе. Если же она падает (а у большинства проектов LTV имеет нисходящий тренд по временной оси) — пора принимать меры и улучшать проект.



## LTV НЕОБХОДИМА ДЛЯ РАСЧЁТА ROI

Метриками оперируют аналитики, а деньги дают собственники и инвесторы. И этим серьезным людям важно знать, окупятся ли их вложения. Для этого придумана метрика ROI (Return On Investment), которая учитывает в себе и LTV, и стоимость привлечения.

ROI можно считать по-разному, мы сейчас говорим о следующей формуле:

$$ROI = \frac{LTV}{CPI} \times 100\%$$

По результатам расчётов ROI должен быть больше 100%.

Мы в [devtodev](#) рекомендуем также рассчитывать ROI за определённые фиксированные интервалы времени от момента регистрации, или первого входа пользователя:

$$ROI \text{ N days} = \frac{\text{CUMULATIVE ARPU N days}}{CPI} \times 100\%$$



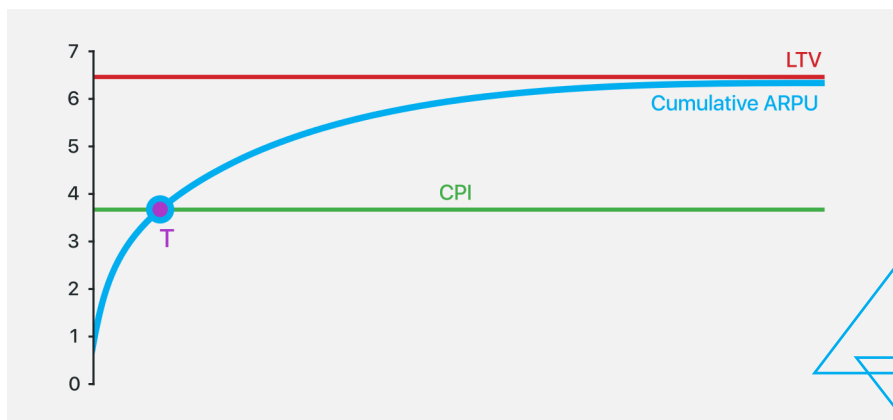
Здесь мы вводим новую метрику Day N Cumulative ARPU, которая демонстрирует, сколько денег в среднем принёс один пользователь за первые N дней использования продукта. Подбирая различные N, вы лучше поймете динамику ROI и сможете рассчитать ещё один важный показатель. А именно...



## С ПОМОЩЬЮ LTV МОЖНО РАССЧИТАТЬ ПЕРИОД ОКУПАЕМОСТИ ПРОЕКТА

Вы сможете узнать, когда же окупятся деньги, вложенные в проект.

Смотрим график:



Синяя линия — это показатель Cumulative ARPU, он показывает, сколько денег в среднем приносит один пользователь за N дней использования продукта.

Показатель LTV — это предел Cumulative ARPU при N, стремящемся к бесконечности (хотя на практике берут фиксированные значения N вроде 120, 180, 360 дней).

Если бизнес работает хорошо, и трафик окупается, значит, есть такая точка T, в которой синяя линия, то есть деньги, принесённые пользователем, становится выше, чем зелёная, то есть деньги, потраченные на пользователя.

И тот день, в который произошло это важное событие, и называется **периодом окупаемости**. Теперь мы можем точно сказать собственнику, когда отобьются вложенные деньги и когда ROI превысит 100%.





## ЗНАНИЕ LTV НЕОБХОДИМО ДЛЯ ПЛАНИРОВАНИЯ ИЗДЕРЖЕК

Вернемся к основной формуле:

$$\text{LTV} > \text{CPI}$$

При расчетах важно знать о понятии чистой LTV, то есть **LTV за вычетом прочих затрат**: комиссии магазина, комиссии издателя и роялти, налогов в конце концов.

С CPI тоже не всё просто. Чтобы начать закупать трафик, надо сначала договориться (добавляем зарплату менеджера), подписать договор (добавляем зарплату юриста), интегрироваться (добавляем зарплату программиста), и это мы ещё не берём в расчёт фиксированную плату за подписание у некоторых контрагентов. Поэтому от CPI мы перейдём к **эффективной цене привлечения eCPI** (по аналогии с эффективной банковской ставкой).

Как правило, в проекте существуют ещё и затраты на поддержание активности пользователя: техническая поддержка, комьюнити-менеджмент, сервера и другие. Итоговая формула приобретает такой вид:

$$\text{Чистая LTV} > \text{eCPI} + \begin{matrix} \text{Издержки на 1} \\ \text{пользователя} \\ \text{(переменные, постоянные)} \end{matrix}$$

Из нее следует, что издержки нужно планировать так, чтобы условие выполнялось после вычета всех комиссий из LTV и прибавления всех затрат к CPI.



## LTV ПОМОЖЕТ СПРОГНОЗИРОВАТЬ БУДУЩИЕ ПОСТУПЛЕНИЯ

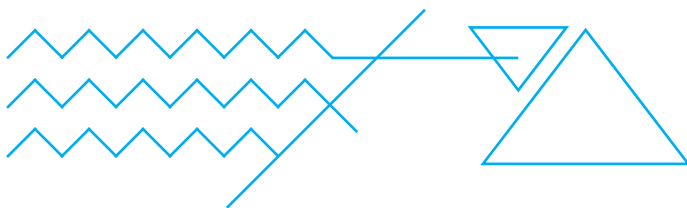
Если вы умеете прогнозировать LTV, да ещё и рассчитываете её в разрезе каналов, стран, платформ и т.д., то во-первых, респект вам, а во-вторых, вы вполне сможете спрогнозировать, сколько денег получите через N месяцев.

Например, вы сможете ответить на такие вопросы:

- что будет с выручкой через 3 месяца, если мы сейчас сократим платный трафик на 50%;
- сколько денег мы получим от неё к концу года, если мы выйдем на рынок новой страны в апреле;
- как это отразится на нашей выручке, если мы внесём в проект изменение, которое увеличит удержание пользователей на 3%;
- когда окупится трафик, который мы закупили у партнёра X и так далее.

Как видим, LTV — важнейший показатель в аналитике проекта. Но есть одна трудность: чтобы посчитать эту метрику, нужно время, а времени, как правило, нет.


Если вы считаете Lifetime Value за короткий срок, прогноз получится не самым точным. Если вы делаете расчет за длительный период, то вопрос прогнозирования перестаёт быть актуальным, будущее наступает нас.



# 4 КАК СЧИТАТЬ LTV?

Вопрос расчёта Lifetime Value рано или поздно встаёт перед разработчиками мобильных приложений. Методов расчёта придумано множество, и по поводу того, как считать LTV, существует сколько людей, столько же и мнений.

Вот, например, скриншот про расчёт LTV из книги «Database Marketing: Analyzing and Managing Customers» — кстати, хорошая и мощная книга по аналитике и маркетингу, если вы любите hardcore:


$$\begin{aligned} E[X(t) | \lambda, p] &= \frac{1 - e^{-\lambda p t}}{p} \\ P(\tau > t | \lambda, p) &= e^{-\lambda p t} \\ E[Y(t) | X = x, t_x, T, \lambda, p] \\ &= \frac{p^{-1} (1 - p)^x \lambda^x e^{-\lambda T} - p^{-1} (1 - p)^x \lambda^x e^{-\lambda (T + p t)}}{L(\lambda, p | X = x, t_x, T)} \end{aligned}$$

WITH

$$L(\lambda, p | X = x, t_x, T) = (1 - p)^x \lambda^x e^{-\lambda T} + \delta_x > 0 p (1 - p)^{x-1} \lambda^x e^{-\lambda t_x}$$

| *Расчёт LTV, Database Marketing: Analyzing and Managing Customers*

Но в рамках нашей книги мы опишем наиболее распространённые методы, обозначим их плюсы и минусы. Данные методы подходят прежде всего для описания модели free-to-play.



## МЕТОД 1. ПОСТФАКТУМ

Начнём с простого. Этот метод выделяется на фоне всех последующих, так как он не моделирует LTV и не прогнозирует её, а считает фактическую LTV.

Для этого метода необходимо взять когорту пользователей, которые уже точно покинули проект, посмотреть, сколько денег принесла вся эта когорта, затем поделить эту сумму на размер когорты.

Желательно, чтобы пользователи были зарегистрированы примерно в одно время - в один месяц, а лучше — в один день.

На практике же этот метод слабо применим, так как обязательно найдётся хотя бы один человек из когорты, который до сих пор активен, как бы давно не регистрировалась когорта.

А потому на практике LTV именно моделируют, а не рассчитывают по факту. И все последующие методы будут именно моделировать будущую LTV, а не оценивать прошлую.



## МЕТОД 2. ВЗЯТЬ ВСЁ И ПОДЕЛИТЬ, ИЛИ «МЕТОД ШАРИКОВА»























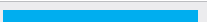
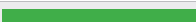
Наиболее быстрый, но грубый метод. Берём весь доход приложения за период и делим на общее количество пользователей за тот же период.



$$LTV = \frac{\text{TOTAL REVENUE ЗА ПЕРИОД}}{\text{TOTAL USERS ЗА ПЕРИОД}}$$

## → ПРИМЕР

Допустим, наш проект запустился в январе, а мы сейчас находимся в конце декабря. Помесячная статистика за год выглядит вот так:

	MAU		Revenue	
January		2 000		\$ 26 000
February		2 500		\$ 45 000
March		2 650		\$ 26 500
April		2 600		\$ 52 000
May		2 400		\$ 26 400
June		2 700		\$ 51 300
July		2 800		\$ 53 200
August		2 850		\$ 45 600
September		2 900		\$ 46 400
October		2 800		\$ 44 800
November		2 600		\$ 26 000
December		2 600		\$ 49 400
	Yearly Active Users		Total Revenue	
		14 550		\$ 492 600

- +** **Плюс** у этого метода только один: считается довольно быстро, буквально в одно действие.
- **Минус** заключается в очевидной неточности метода, которая может быть обусловлена, например, следующими причинами:
  - Не учитывается доход от тех пользователей, которые уже успели стать активными (попали в знаменатель), но еще не успели принести доход (который попал бы в числитель).
  - В расчёт попадают значения метрик приложения с самого начала его жизни; не стоит забывать, что приложения имеют свой жизненный цикл, и как правило, в начале своего жизненного цикла показатели лучше, чем спустя некоторое время после. В этом же методе все этапы жизни приложения объединены.
  - Также в этом методе трудно посчитать LTV отдельно для каждого пользовательского сегмента — для этого нужно заранее знать размер сегмента и количество денег, принесенных пользователями этого сегмента.



## МЕТОД 3. ЧЕРЕЗ LIFETIME И ARPU, ПРОСТОЙ СПОСОБ

Формула этого метода такова:

$$\text{LTV} = \text{Lifetime} \times \text{ARPU}$$

Глядя на эту формулу, можно задаться вопросом, что такое Lifetime и как её считать.

**Lifetime** — это метрика, которая показывает, сколько дней среднестатистический пользователь пользуется вашим приложением от первого до последнего входа.

Однако, ждать последнего входа пользователя часто приходится долго, поэтому этот показатель обычно определяет период неактивности, после которого пользователь считается «отвалившимся».

**Существует два способа расчёта Lifetime:** простой и сложный.

Для этого метода мы возьмем простой, как и обещано в заголовке:

- Мы определяем некоторый период неактивности, то есть время, после которого пользователь скорее всего уже не вернётся в приложение. Определяют это либо на основании значений Retention, либо, что чаще, «экспертно». Обычно «экспертно» это значение задают равным одной или двум неделям.
- Каждый день мы смотрим на пользователей, у которых в этот конкретный день истек период неактивности.
- Для каждого пользователя вычисляем количество дней от его первого визита до текущего дня.
- Рассчитываем среднее значение по всем пользователям. Это и есть Lifetime.

Ну а ARPU (в данном случае  $ARPU = ARPAU$ ) рассчитывается как дневной Revenue, делённый на DAU. Умножаем Lifetime на ARPU и получаем LTV.

## ➔ ПРИМЕР

Допустим, на дворе 20.04.18, и мы помечаем тех, кто не заходил уже более 7 дней, как неактивных.

	Install Date	Last Activity Date	Days from last activity	Days of activity
User 1	16.04.18	16.04.18	4	0
User 2	01.04.18	13.04.18	7	12
User 3	01.03.18	01.04.18	19	31
User 4	15.03.18	13.04.18	7	29
User 5	16.03.18	20.04.18	0	35
User 6	01.01.18	13.04.18	7	102
User 7	02.03.18	05.04.18	15	34
User 8	20.03.18	04.04.18	16	15
User 9	10.04.18	13.04.18	7	3
User 10	20.04.18	20.04.18	0	0

Таковых набралось трое, и средний период от даты установки до даты последней активности у них равен  $(12 + 29 + 3) / 3 = 14.7$  дней.

### Почему именно 7 дней мы задали как период неактивности?

Скажем так, экспертно. Для разных приложений эта граница будет вести себя по-разному. Кто-то выбирает неделю, кто-то две недели, кто-то, например, туристические сервисы, — до месяца. Просто определитесь сами, через сколько дней вы будете считать пользователя неактивным.

#### ⊕ Плюсы метода:

- Простота расчётов. Рассчитать Lifetime таким образом нетрудно, ещё легче рассчитать ARPU. А перемножить одно на другое сможет любой школьник.
- Можно рассчитывать LTV хоть каждый день.
- LTV можно рассчитать по каждому пользовательскому сегменту в отдельности.

⊖ **Минусы метода** вновь заключаются в неточности, которая в этом случае обусловлена следующими причинами:

- Значение сильно зависит от периода неактивности, задаваемого, как правило, экспертно (как и сделали мы в нашем примере).
- Мы умножаем среднее значение Lifetime на среднее значение ARPU, получаем накопленную ошибку.
- При расчёте Lifetime мы смотрим на тех пользователей, которые уже покинули приложение. При расчёте же ARPU мы смотрим на пользователей текущего дня. Получается, что множества пользователей, формирующих Lifetime и ARPU, не пересекаются: Lifetime считается по данным прошлых дней, ARPU — по текущему дню.
- Сильное предположение о неизменности ARPU. Мы берём ARPU лишь за один день и на его основании прогнозируем LTV на множество дней вперёд.

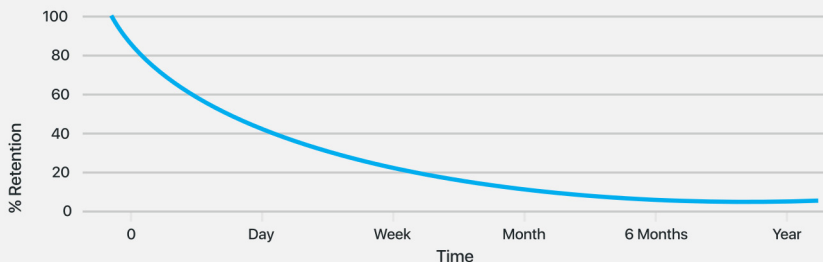
## ➤ МЕТОД 4. ЧЕРЕЗ LIFETIME И ARPU, СЛОЖНЫЙ СПОСОБ

Формула метода точно такая же:

$$\text{LTV} = \text{Lifetime} \times \text{ARPU}$$

Но Lifetime тут считается немного сложнее и получается намного точнее.

Вспомним, как выглядит график Retention:





Дело в том, что Lifetime — это площадь фигуры под графиком Retention, иначе говоря — интеграл от Retention по времени.

Но прежде чем считать интеграл, надо построить саму функцию Retention. В этом случае вам предстоит смоделировать эту Retention самостоятельно и по модельному значению отвечать на интересующие вас вопросы.

## → Моделирование Retention

Помните, в школе вы строили кривые по точкам? Вот именно здесь эти знания вам и пригодятся. Единственное, приготовьтесь, что в итоге уравнения будут чуть сложнее, чем в школе.

Прежде всего, отметим, что при моделировании Retention можно пользоваться любыми статистическими и аналитическими пакетами.

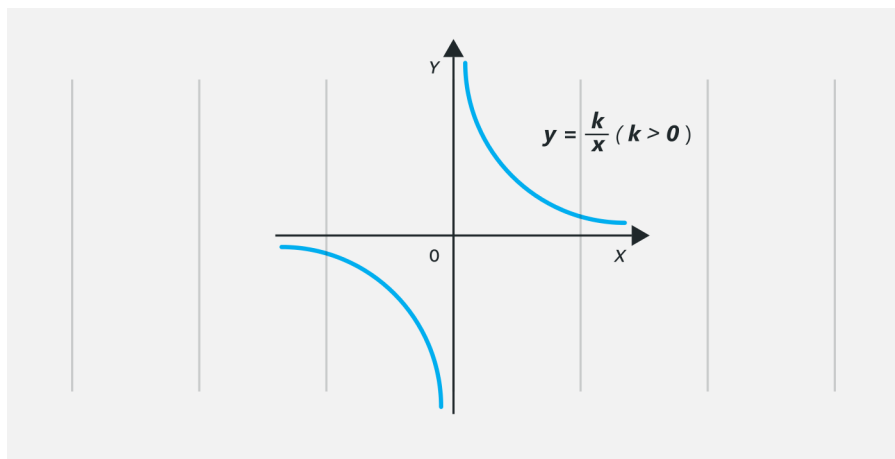
Всё зависит от ваших предпочтений и уровня владения той или иной программой. Но в рамках этой книги мы воспользуемся бесплатным и общедоступным программным обеспечением:

- **Open Office**, а именно его электронные таблицы и «Решатель» (Solver) во вкладке «Сервис»;
- «**Нелинейный решатель**», который нужно поставить отдельным бесплатным плагином.

Еще мы будем пользоваться **механизмом аппроксимации**, то есть приближением фактических значений математическими формулами.

Делая аппроксимацию, важно, во-первых, выбрать правильную функцию (которая бы изгибалась в нужных местах) и, во-вторых, верно подобрать её коэффициенты, чтобы разница между моделью и фактом была минимальной.

Итак, какой же из функций можно аппроксимировать удержание?  
На ум (тем, кто заканчивал школьный курс математики) приходит гипербола, и это верная ассоциация.



Рассмотрим несколько гиперболических уравнений ( $X$  в уравнении означает номер периода: дня, недели, или месяца). Начнём с простого уравнения гиперболы  $\frac{A}{X}$ , затем будем усложнять его, добавляя различные коэффициенты:

Retention =  $\frac{A}{X}$

Retention =  $\frac{A}{(X+B)}$

Retention =  $\frac{A}{(X+B+C)}$

Retention =  $\frac{A}{(X^D+B+C)}$

Неизвестные A, B, C, D — это коэффициенты, которые нам предстоит найти. Но наша задача состоит не только в этом.

Найдя в каждой из формул значения коэффициентов, мы затем должны будем выбрать самое оптимальное из этих уравнений.

То есть, итог каждого из уравнений — отдельная кривая. Мы будем сравнивать эту кривую с фактическими значениями, которые, надо сказать, не всегда идеально вписываются в модель, и выбирать ту из кривых, которая лучше повторяет факт.

Критерием будет минимум суммы квадратов отклонений, что означает, что мы воспользовались методом наименьших квадратов, между фактическими и модельными значениями.

В Excel это делается с помощью **СУММКВРАЗН (SUMXMY2)**.

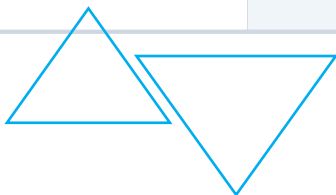
В Open Office эта функция нами не обнаружена, но это не проблема: рассчитываем в отдельном столбце отклонения (просто как разность между модельными и фактическими значениями), возводим их в квадрат, затем суммируем квадраты отклонений.

Для оптимизации нам пригодится Solver. Притом, учитывая и квадраты отклонений, и гиперболический вид функции, нужен именно нелинейный.

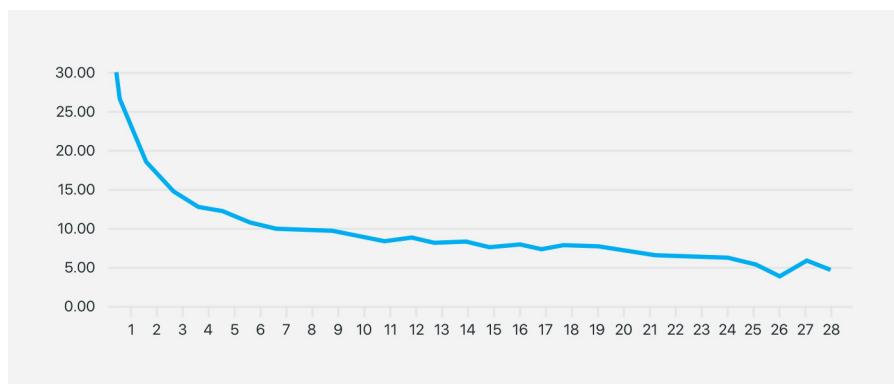
Здесь и далее мы пользовались поочерёдно DEPS Evolutionary Algorithm и SCO Evolutionary Algorithm, за стартовые данные новой итерации брали значения коэффициентов, полученные на предыдущей итерации.

Процесс заканчивался тогда, когда сумма квадратов отклонений с новой итерацией уменьшалась не более чем на 0.01.

<div> <div>fx</div> <div>✗</div> <div>✓</div> <div>=(B16-C16)^2</div> </div>			
	B	C	D
	basic data	A/X	
1	28.32	38.71	107.99
2	19.02	19.35	0.11
3	15.36	12.90	6.02
4	13.07	9.68	11.49
5	12.27	7.74	20.46
6	10.84	6.45	19.26
7	10.12	5.53	21.04
8	10.05	4.84	27.18
9	9.80	4.30	=(B16-C16)^2
10	9.12	3.87	27.54
11	8.27	3.52	22.57
12	8.64	3.23	29.34
13	8.17	2.98	26.93
14	8.26	2.76	30.93
15	7.71	2.58	26.27
16	7.77	2.24	28.67
17	7.34	2.28	25.66
18	7.81	2.15	32.00
19	7.60	2.04	30.94
20	7.04	1.94	26.02
21	6.65	1.84	23.11
22	6.40	1.76	21.53
23	6.25	1.68	20.86
24	5.91	1.61	18.50
25	5.03	1.55	12.15
26	3.48	1.49	3.97
27	5.83	1.43	19.28
28	6.40	1.38	9.11
		678.44	



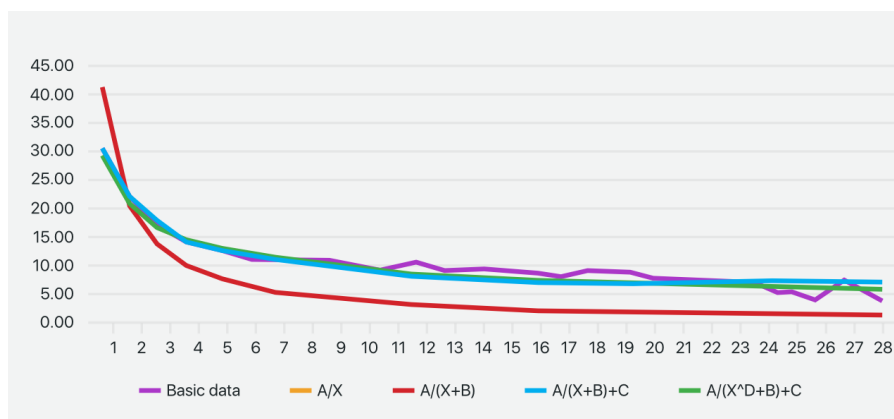
За исходные данные возьмём удержание демо-проекта [devtodev](#) (вы можете повторить всё то же самое самостоятельно, демо доступно без регистрации) за 28 дней.



Как видно из графика, данные далеко не идеальны, и для нашей задачи это отлично подходит. Редко на практике вы получаете в руки идеальные данные, а задачу решать всё равно надо.

Будем пытаться для каждой из выбранных функций подобрать такие значения коэффициентов, чтобы построить кривую, как можно более близкую к исходным данным.

Вот что получается:



Как видно, жёлтая  $\frac{A}{X+B}$  и красная  $\frac{A}{X}$  линии совпали, но от исходной кривой они очень далеки, а вот зелёная  $\frac{A}{X+B} + C$  и бордовая  $\frac{A}{X^D+B} + C$  линии довольно точно повторили исходную кривую.

При этом бордовая линия, если судить по сумме квадратов отклонений, повторяет стартовые данные точнее всего:

СУММА КВАДРАТОВ ОТКЛОНЕНИЙ			
$\frac{A}{X}$	$\frac{A}{X+B}$	$\frac{A}{X+B} + C$	$\frac{A}{X^D+B} + C$
678,44	674,13	17,99	11,55

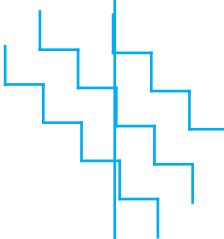
Таким образом, мы прощаемся с уравнениями  $\frac{A}{X+B}$  и  $\frac{A}{X}$  и идём дальше.

Если обратить внимание на кривую, то видно, что с каждым днём она меняется всё меньше.

Нам это напомнило про логарифмическую кривую, и мы подумали: а что если вместо  $X$  в уравнение подставить  $\ln(X)$ , не улучшит ли это наши результаты?

Поэтому следующим шагом давайте сравним результаты лучшей функции с  $X$  и  $\ln(X)$ . Единственное, в одном из случаев добавим под логарифм коэффициент  $E$ :

- Retention =  $\frac{A}{(X+B)} + C$
- Retention =  $\frac{A}{(X^D+B)} + C$
- Retention =  $\frac{A}{[\ln(X)^D+B]} + C$
- Retention =  $\frac{A}{[\ln(X+E)^D+B]} + C$



Рассмотрим получившиеся квадраты отклонений:

СУММА КВАДРАТОВ ОТКЛОНЕНИЙ			
$\frac{A}{X+B} + C$	$\frac{A}{X^D+B} + C$	$\frac{A}{\ln(X)^D+B} + C$	$\frac{A}{\ln(X+E)^D+B} + C$
17,99	11,55	10,66	10,57

### Какие выводы можно сделать?

- Лучше всего аппроксимируют кривые с четырьмя и пятью переменными.
- Замена X на LN(X) пусть и немного, но улучшает аппроксимацию.

На этом этапе мы прощаемся с кривой  $\frac{A}{X+B} + C$ .  
Она не выдержала конкуренции.

Остались три кривые, однако неизвестно, как они работают на длинных дистанциях. Мы тестировали их лишь на первых 28 днях.

Вполне вероятен случай, что если вместо X подставить большое значение (скажем, 365), то они уйдут в минус, что невозможно по определению удержания.

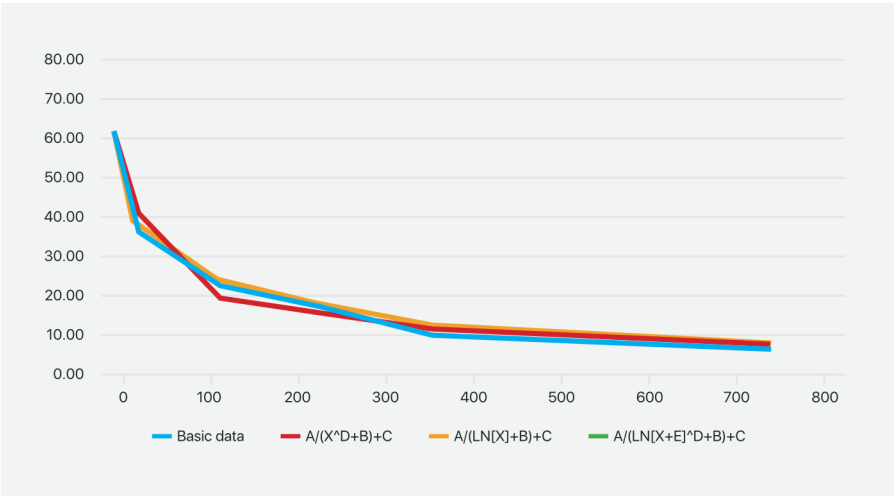
Поэтому, раз уж мы определили трёх финалистов, давайте следующим этапом протестируем, как они могут справляться с более долгосрочным удержанием. Мы просто взяли несколько примеров долгосрочного удержания и протестировали наши кривые на каждом из них:

- за 12 недель,
- за 18 месяцев,
- за 720 дней.

Пример про 720 дней рассмотрим графически.

Retention rate in %	BILLION-DOLLAR SOCIAL NETWORK	Good Game	Normal App
D 1	70 %	40 %	30 %
D 7	58 %	20 %	10 %
D 30	37 %	10 %	1 %
D 120	23 %	1%	0 %
D 365	11 %	0 %	0 %
D 720	8 %	0 %	0 %

Данные были взяты из статьи «How to Measure the Success of Your App», и в данном случае мы пытались повторить приведённую статистику по удержанию в социальной сети:



Видно, что все три кривые хорошо справились с аппроксимацией, однако красная линия немного сильнее выдаётся на фоне синей — отклонение у неё максимальное.



Теперь делимся результатами по всем трём примерам.  
 Напомним, чем меньше сумма квадратов отклонений, тем лучше:

СУММА КВАДРАТОВ ОТКЛОНЕНИЙ			
$\frac{A}{X+B} + C$	$\frac{A}{X^D+B} + C$	$\frac{A}{\ln(X)^D+B} + C$	$\frac{A}{\ln(X+E)^D+B} + C$
12 недель	1,29 (2-3 место)	1,29 (2-3 место)	1,22 (1 место)
18 недель	0,82 (1 место)	1,02 (3 место)	0,93 (2 место)
720 дней	7,80 (3 место)	4,26 (1-2 место)	4,26 (1-2 место)

Если бы это был чемпионат, то победила бы последняя кривая. Однако, согласитесь, её преимущество не так явно, особенно учитывая, как незначительно отличаются друг от друга суммы квадратов отклонений. Поэтому победителями, как на детском празднике, мы признаем все три кривые, и все три кривые мы можем рекомендовать для аппроксимации удержания.

Также хотелось бы рассказать про значение каждого из коэффициентов на примере функции  $\frac{A}{\ln(X)^D+B} + C$ .

КОЭФФИЦИЕНТ	НА ЧТО ВЛИЯЕТ	ПРИМЕР 1	ПРИМЕР 2
A	на изгиб и масштаб кривой	 A=5	 A=10
B	на изгиб/горизонтальность кривой	 B=1	 B=10
C	на положение кривой по вертикальной оси	 C=1	 C=5
D	на скорость отвала на первых днях (чем выше коэффициент, тем больше график похож на прямой угол)	 D=1	 D=5

Вывод, к которому мы приходим, следующий: для аппроксимации Retention хорошо подходят три вида кривых:

- **Retention** =  $\frac{A}{X^D+B} + C$
- **Retention** =  $\frac{A}{\ln(X^D)+B} + C$
- **Retention** =  $\frac{A}{n(X+E)^D+B} + C$



Победители известны, но теперь мы хотим огласить несколько важных моментов, которые нужно иметь в виду при аппроксимации удержания:

➔ **Если у вас мало исходных точек, то лучше использовать ту кривую, в которой меньше коэффициентов.**

И запомните: никогда не используйте кривую, у которой неизвестных коэффициентов больше, чем у вас точек!

Скажем, если у вас есть лишь три точки удержания (допустим, день 1, день 7, день 28), то максимальное количество коэффициентов, которые вы можете использовать — это три, и в этом случае лучше всего подойдет функция  $\frac{A}{X+B} + C$ .

➔ **Вы всегда вольны менять оптимизационную функцию так, как вам вздумается.**

Мы в [devtodev](https://devtodev.com) воспользовались стандартным МНК и использовали простую сумму квадратов отклонений.

Допустим, вам не так важно поведение удержания в другие периоды, но вы хотите, чтобы модельное и фактические значения удержания точь-в-точь совпали на 180 днях.

Поэтому для оптимизационной функции отклонение на 180 днях можно взять с большим коэффициентом или в большой степени.



**Мы даём вам не универсальную рекомендацию, а просто наш опыт решения нескольких разовых задач.**

Не исключено, что есть и другие функции, более точно аппроксимирующие показатели удержания. Но те функции, которые использовали мы, дали хороший результат.

Итак, удержание мы смоделировали, но и это ещё не всё, но мы уже близко.

Дальше по-прежнему можно выбрать сложный или простой метод.

Сложный метод заключается в нахождении интеграла от функции Retention.

Напомним, что:

$$\int \frac{dx}{x} = \ln |x| + C$$

Простой же метод заключается в том, чтобы, пусть и примерно, разделить кривую Retention на сегменты в зависимости от значения Lifetime. Например, на пользователей, ушедших через день, проживших в приложении от 2 до 7 дней, от 8 до 30 дней, от 1 до 3 месяцев, свыше 3 месяцев.

Чем больше сегментов, тем лучше. Для каждого сегмента посчитать по таблице Retention процент пользователей (вес сегмента), относящихся к нему, а затем посчитать средневзвешенный Lifetime по всем сегментам.

Но какой бы метод вы ни выбрали, вы столкнётесь с вопросом, до какого момента считать LTV (в случае с интегралом, это будет правый край области интегрирования, в случае с суммой — количество дней в самом последнем сегменте).

И здесь вновь существует два метода решения: простой и сложный.

**Простой метод** заключается в том, что правый край задаётся экспертно.

**Сложный метод** заключается в использовании дисконтирования и нахождении ставки дисконтирования WACC.

Признайтесь, вы не ожидали увидеть здесь финансовую математику? Дело в том, что тысяча долларов сейчас и тысяча долларов завтра — это разные суммы денег. Завтрашняя тысяча долларов сегодня будет равна девятистам долларам или около того, в зависимости от выбора ставки дисконтирования.

Формула такова:

$$PV = \frac{CF_1}{1+WACC} + \frac{CF_2}{(1+WACC)^2} + \dots + \frac{CF_N}{(1+WACC)^N} = \sum_{i=1}^N \frac{CF_i}{(1+WACC)^i}$$



Здесь PV (Present Value) — текущая стоимость будущих денег, CF<sub>i</sub> — деньги, которые вы получите через *i* временных периодов, WACC (Weighted Average Cost of Capital) — та самая ставка дисконтирования.

### Как её найти?

Обычно WACC делают равным фактической рентабельности капитала в среднем по фирме. Также можно приравнять его к желаемой рентабельности капитала, либо к рентабельности капитала альтернативных проектов.

Если вы не поняли этот абзац, спросите у своих финансистов, они наверняка знают WACC вашей компании.

Итак, зная WACC, вы сможете дисконтировать будущие временные потоки, а следовательно, в качестве правого края интегрирования выбрать хоть бесконечность. Дело в том, что добавление WACC делает из вашей суммы (или из вашего интеграла) бесконечно убывающую последовательность, у которой можно найти сумму.

Будем считать, что Lifetime мы посчитали. Теперь же считаем ARPU (Revenue/DAU), умножаем ARPU на Lifetime и получаем LTV.

#### ⊕ Плюсы метода:

- Точность. Lifetime рассчитан очень точно, погрешность в нём минимальна.
- Побочным эффектом от расчёта такого метода является то, что вы бонусом получаете ещё и прогноз Retention на сколько угодно дней.
- Возможность посчитать LTV для каждого сегмента в отдельности.

#### ⊖ Минусы метода:

- Сложно считать, хотя опытный аналитик при наличии всех данных посчитает вам LTV за пять минут.
- Вновь предположение о неизменности ARPU во времени. Можно немного перестраховаться и взять в расчёт не ARPU за один день, а среднедневной ARPU за Lifetime, это увеличит точность.



## МЕТОД 5. НАКОПИТЕЛЬНЫЙ ARPU, ИЛИ TOP-DOWN

Второе название метода взято из материала Wooga, что даёт +10 к доверию к данному методу. Из этого же материала взята и картинка:



## → ПРИМЕР

Допустим, к вам в проект пришла группа новых игроков, и вы стали за ней следить. Вы замеряете, сколько денег принёс вам в среднем один игрок из этой группы за 7 дней, за 14, за 28, и так далее. То есть, по сути, вы переходите от обычного ARPU к накопительному за N дней.

Ну а зная Cumulative ARPU за 7, 14, 28 и т.д. дней, мы вновь сможем построить математическую модель кривой, которая будет прогнозировать значения Cumulative ARPU за сколько угодно дней. Будем искать уравнение кривой вида:

$$F(t) = A + \ln(t+B)$$



где **t** — количество дней от первого визита пользователя, **F(t)** — будущее уравнение, **A** и **B** — коэффициенты модели.

Вновь рассчитываем сумму квадратов отклонений и минимизируем её за счёт подбора оптимальных значений коэффициентов A и B.

Если же у вас есть больше значений Cumulative ARPU (скажем, за 60 и 90 дней), то можно добавить в уравнение дополнительные слагаемые вида  $Cx^t$  или  $D/t$ , это может повысить точность.

Ну и в целом — здесь нет одного уравнения, гарантированно дающего минимальное отклонение. Экспериментируйте с видом уравнения!

Путём нескольких итераций вы таки получите уравнение, которое вас устроит. Теперь, подставив в это уравнение нужное вам значение t, вы получите Cumulative ARPU(t), что по сути и будет равняться LTV.

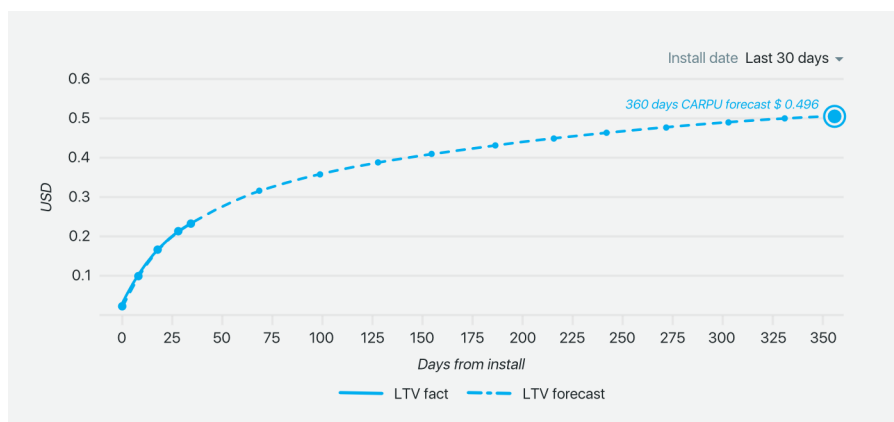
## → Как выбрать значение t для расчёта LTV?

- Можно взять Lifetime.
- Можно вновь задать это t экспертно.
- Можно вернуться к дисконтированию и добавить в получившееся уравнение знаменатель  $\frac{CF_N}{(1+WACC)^N}$ .

В этом случае рано или поздно на графике станет намечаться асимптотическое значение (как на картинке выше — примерно \$3.70, выше которого LTV быть не сможет. Вот это значение и берите).

### → ПРИМЕР

Вот как работает прогноз по методу Cumulative ARPU в системе [devtodev](#).



Мы имеем фактические данные о накопительном ARPU пользователей за их первые 7, 14, 20 и 30 дней жизни в системе. Больше иметь не можем, так как в данном отчёте выбраны пользователи, Install Date у которых ограничен интервалом «last 30 days» (смотрите левый верхний уровень скриншота).

Математической аппроксимацией мы строим кривую и рассчитываем накопительный доход за  $t = 360$  дней, который мы и считаем в нашей системе метрикой LTV.

Затем эта кривая ещё корректируется за счёт информации о схожих проектах в системе [devtodev](#). Схожесть определяется такими показателями, как жанр и платформа, а также близостью значений качественных метрик проекта: Retention, доля платящих пользователей, ARPU.

Итак, мы рассмотрели множество методов расчёта LTV, которые, как вы могли заметить, упорядочены от наименее точного к наиболее точному. Выбирайте тот метод, который вам по душе, рассчитывайте свою LTV и принимайте правильные решения.

А теперь **главное правило LTV**: делите пользователей на сегменты, и считайте LTV каждого сегмента в отдельности. Это даст вам и более высокую точность, и больше поводов для принятия правильных решений по вашему продукту.



# 5 ЗАКЛЮЧЕНИЕ

В заключении выделим основные моменты книги:

- Lifetime Value (LTV) – это метрика, показывающая, сколько денег принёс пользователь за всю его жизнь в проекте.
- LTV универсальна. С её помощью вы сможете оценивать качество трафика, планировать издержки, рассчитывать другие показатели вашего проекта и прогнозировать будущие поступления денег. Она рассчитывается и в веб-аналитике и в мобайле. Также, её считают для большинства видов продуктов.
- LTV имеет огромное количество вариантов расчёта, значительно отличающихся по сложности вычислений. Определённых рекомендаций по способам расчёта нет, выбирайте тот, который больше всего подходит вашему проекту.

**ЖЕЛАЕМ ВАМ РОСТА LTV  
И ВСЕХ ДРУГИХ МЕТРИК,  
КРОМЕ ОТТОКА!**

А ЕСЛИ НУЖНЫЕ МЕТРИКИ НЕ  
БУДУТ РАСТИ — **DEVTODEV**  
ВАМ ВСЕГДА ПОМОЖЕТ.

# ЧТО ПОЧИТАТЬ ПО ТЕМЕ:

Данная книга написана на основе следующих статей:

- » Для чего необходимо знать показатель LTV:  
<https://habrahabr.ru/company/devtodev/blog/298158/>
- » Как считать Lifetime Value: обзор методов:  
<https://habrahabr.ru/post/292312/>
- » Моделируем удержание пользователей  
(в поисках оптимальной формулы):  
<http://app2top.ru/marketing/modeliruem-uderzhanie-pol-zovatelej-v-poiskah-optimal-noj-formuly-99392.html>

Также могу рекомендовать следующие источники:

- » Database Marketing: Analyzing and Managing Customers  
(International Series in Quantitative Marketing):  
<https://www.amazon.com/Database-Marketing-Analyzing-International-Quantitative/dp/1441903321>
- » Two Methods for Modeling LTV with a Spreadsheet:  
<https://www.slideshare.net/EricSeufert/ltv-spreadsheet-models-eric-seufert>
- » The Ultimate Guide to SaaS Customer Lifetime Value (LTV):  
<https://blog.chartmogul.com/the-ultimate-guide-to-saas-customer-lifetime-value-ltv/>
- » How to calculate the lifetime value: the review of methods:  
[https://www.gamasutra.com/blogs/VasiliySabirov/20150903/252866/How\\_to\\_calculate\\_the\\_lifetime\\_value\\_the\\_review\\_of\\_methods.php](https://www.gamasutra.com/blogs/VasiliySabirov/20150903/252866/How_to_calculate_the_lifetime_value_the_review_of_methods.php)

Образовательный центр devtodev представляет:

Новый онлайн-курс

# ИГРОВАЯ АНАЛИТИКА: ОТ ОСНОВ К ПРОДВИНУТЫМ МЕТОДАМ

Этот курс подходит для всех, кто интересуется созданием игр и геймдизайном, вне зависимости от уровня аналитических знаний и способностей.

## ВЫ НАУЧИТЕСЬ:

- Прогнозировать доход игры
- Выводить продукт на точку безубыточности
- Находить неэффективные источники трафика
- Разбираться в основных метриках продукта
- Анализировать данные и визуализировать их
- Находить «слабые» места и точки роста продукта
- Улучшать игровой опыт пользователей и удерживать их



4 МОДУЛЯ  
16 ЛЕКЦИЙ



РАЗБОР  
КЕЙСОВ



СВОБОДНЫЙ  
ГРАФИК



СЕРТИФИКАТ



АНАЛИТИЧЕСКАЯ ПЛАТФОРМА,  
ПРЕДНАЗНАЧЕННАЯ ДЛЯ РАЗРАБОТЧИКОВ ИГР

С DEVTODEV ВЫ СМОЖЕТЕ:

Считать все базовые метрики игры в реальном времени

Находить, где и почему отваливаются пользователи

Оптимизировать tutorial и первую игровую сессию

Оценивать эффективность игровых уровней и локаций

Находить узкие места в экономике игры

Прогнозировать, сколько денег принесет пользователь

Анализировать поведение платящей аудитории

