

Станислав Домбровский

Email: yeswolf@gmail.com

Phone: +7 911 976 56 28

TG: [basic_87](#)

Цели / сферы

Менеджмент

- Продуктовый менеджмент и маркетинг
- Руководство командой и организация процессов

Разработка ПО

- Сферы, связанные с UI/UX
- IDE / тулинг
- Мобильные / встраиваемые операционные системы

Содержание

- [Опыт](#)
 - [JetBrains](#)
 - [e-legion](#)
 - [Goodwill LTD](#)
 - [Fotostrana](#)
 - [Axmor](#)
- [Образование](#)
- [Выступления](#)
- [Конференции](#)
- [Материалы и публикации](#)
- [Прочие интересы](#)

Опыт

JetBrains

Маркетолог в команде [AppCode](#) (октябрь 2015 - март 2022)

Пришел в 2015-м году в команду [AppCode](#) в момент, когда в мире iOS основной язык программирования (Objective-C) начал бурно меняться на Swift. Сам AppCode - это платная среда разработки под Apple-based системы, которая конкурирует с бесплатным стандартом в отрасли (Xcode), созданным Apple.

Что именно драйвил я и чего удалось достичь за 6.5 лет по AppCode:

- Уменьшили порог входа в продукт - проработали обучающие материалы на каждом шаге миграции, покрыли статьями / видео все основные сферы IDE, убрали много детских болезней, мешавших комфортной работе после первого запуска, встроили в IDE обучающий плагин ([IDE Features Trainer plugin](#)). Стоит отметить, что на текущий момент AppCode - один из немногих продуктов JetBrains с полноценным Migration Guide, разработанным мной.
- Реализовали ряд стратегически важных для IDE новых интеграций:
 - Поддержали Swift Package Manager. Долгое время было неочевидно, насколько востребованным будет пакетный менеджер, который Apple создала в момент выхода Swift в OSS, так как на тот момент превалировал пакетный менеджер, созданный внутри сообщества разработчиков (CocoaPods), и ситуация не менялась в течение нескольких лет. Тем не менее, проанализировав ситуацию, поняли, что в будущем это с большой вероятностью не мертворожденный эксперимент, а часть проектной модели будущих проектов, в результате к моменту, когда это случилось (по сути через 3 года после анализа ситуации), мы были к этому полностью готовы и использовали готовое ядро.
 - Swift на Linux ([from - to](#)), [Swift на Windows](#) и Swift на сервере ([CLion](#), [AppCode](#)). Нельзя сказать, что это движение от Apple было успешным, и для нас это было очевидным с самого начала (опять же, исходя из анализа ситуации). Тем не менее это позволило хорошо использовать инвоповод в сообществе, выстроить коммуникацию с рядом лидеров мнений, с которыми ранее контакта не было, и в итоге еще и позволило улучшить работу частей IDE, напрямую несвязанных с этим направлением. В итоге до сих пор CLion с Swift-плагином, выделенным из AppCode - это без преувеличения единственная полноценная IDE для разработки под кросс-платформенный Swift.
 - Интеграция SourceKit - по сути одна из первых полноценных интеграций LS (Language Server) среди всех IDE от JetBrains, если вообще не первая. Мы смогли

- переиспользовать часть функциональности конкурента, за счет этого уделить больше времени работе над более критичным для нас функционалом (более сложными, чем показ ошибок / предупреждений частями IDE), попутно сделать автодополнение, навигацию (а в перспективе и многие другие действия IDE) доступными до завершения индексации.
- Вынесли сборку, отладку, запуск проекта и тестов за индексацию - сделали их доступными почти сразу же после открытия проекта. Это был один из основных сценариев, которые мешали крупным проектам использовать AppCode - крупные проекты требуют существенного времени для первоначального построения символов, необходимого для работы практически всех частей IDE. Чем быстрее доступны наиболее критичные из них - тем больше желание использовать продукт.
 - Встроили полноценную поддержку баз данных (из [DataGrip](#)) и реализовали [Code coverage](#).
 - Поддержали ряд популярных библиотек (например, Quick/Nimble для BDD).
- Выстроили личный контакт с рядом крупных клиентов (как текущих, так и потенциальных).
 - Построили процесс выбора конференций - отсекали непринципиальные события, выделили основные, сделали framework для принятия решений об участии, который позволил отслеживать конференции и их полезность в динамике. Учитывая выбор между 10-15 событиями в течение года, необходимость варьировать состав команды на конференции - это сильно автоматизировало процесс планирования.
 - Формализовали разницу в функциональности между AppCode и платформой IntelliJ, а также в уровне поддержки языков внутри AppCode (три основных языка - C/C++/Objective-C/Swift имели разные подмножества реализованного), что позволило проще планировать очередную итерацию разработки и постепенно эту разницу уменьшать.
 - Автоматизировали ряд процессов публикации релиза с моей подачи - исключили ручные действия там, где они все еще оставались.
 - Переработали в несколько итераций структуру вебсайта - графика, тексты, структура страниц.
 - Вычистили help от исторически “случайно мертвых” страниц (не только для AppCode, а вообще для всех продуктов компании)
 - Формализовали процесс создания обучающих материалов - так, чтобы единично созданная обучающая статья была еще и скриптом видео, статьей в help продукта, источником графики для tips, встроенных в IDE, соцсетей и прочего.

Кроме AppCode:

- За 4 года я разработал в несколько итераций видение нового продукта - [Kotlin Multiplatform Mobile plugin](#), который разрабатывался нами как будущая полноценная IDE для кросс-платформенного Kotlin. Результат прост - финальное понимание продукта попало в точку, что было проверено практикой и отзывами от сообщества.

По отделу маркетинга:

- Автоматизировал публикацию статей засчет [плагина](#) для конвертации Google Docs в нормальный HTML. Google Docs удобен для проработки текстов и их ревью, но публикация на различных ресурсах подразумевает собой минималистичный HTML, в котором неплохо бы проверить типичные ошибки. Вручную это увеличивает время на публикацию кратно. Плагин в итоге вырос до мини-IDE для анализа естественного текста, несмотря на внешнюю простоту.

В целом, я шел от продукта, не останавливаясь на прямых обязанностях и имея большую свободу действий в формате “если видишь возможность сделать полезное - делай”.

[e-legion](#)

Менеджер проектов (июнь 2014 — сентябрь 2015)

Руководил разработкой мобильных приложений для нескольких крупных компаний (все известные банки итп).

Что было сделано:

- Полностью перестроил процесс оценки и попутно пресейла. В итоге оценка стала биться с реальностью, пресейл стал продавать существенно больше и получил для этого прямую аргументацию, а заказчик получил предсказуемый процесс, где сдвиг если и случается - то редко, но если уж случился - то понятно, куда конкретно.
- Построил простой процесс фиксации проблем по проектам в ходе совещаний, который позволил их решать, а не только обсуждать раз за разом.
- Вытащил несколько проектов из минуса в ноль и прибыль в короткие сроки.
- Вырастил несколько хороших разработчиков внутри команды.
- В целом, успешно руководил парой мобильных команд по 3-5 человек, выделенными на проект QA (до 4 человек), выделенными на part-time дизайнерами (до 3 человек), аналитиками, распределенной территориально командой субподрядчиков.

Goodwill Ltd

Лид команды разработки (февраль 2013 — январь 2014)

Руководил двумя компактными командами разработки под Android и iOS: - Оптимизировал размер команды, в основном на iOS-направлении - оставили лучших, ускорили за счет этого разработку, порезали косты. - В принципе отстроил удобный процесс разработки / тестирования итп. - Улучшил работу продукта, убрав проблемы в архитектуре приложений. - В итоге довел приложения до публикации в Google Play / App Store.

Fotostrana

Продуктовик в приложении под Android (февраль 2012 — октябрь 2012)

Пришел как разработчик, а в итоге стал продуктовым, который сделал все необходимое для того, чтобы первая версия дэйтинговых мобильных приложений компании появилась и начала приносить реальные деньги. В данном случае “все” означает “от разработки самого мобильного приложения до создания первых прототипов структуры, утверждения их, разработки спецификации API приложения, руководства реализацией выделенных сервисов чата, платежной системы итп”. Позиция фактически совмещала в себе разработчика, архитектора, менеджера продукта, маркетолога, и, пожалуй, немного дизайнера.

Итог - приложение было сделано за полгода с нуля и на момент публикации вышло в top-50 в соцсетях и top-10 по поиску в Google Play. Сервисы, созданные для приложения, в дальнейшем активно переиспользовались мобильной версией ФС. Кроме всего прочего с момента публикации за счет простой встроенной монетизации (на тот момент это буквально один экран покупки премиума) приносило реальный доход. Стоит отметить, что текущая версия приложения предельно далека от созданной на тот момент.

Axmor

Ведущий разработчик под мобильные ОС (август 2007 — февраль 2012)

- Начал как разработчик с нулевым опытом на позиции QA. За первые два года стал senior по разработке под Windows Mobile (C#), за следующие два - senior по разработке под Blackberry.
- Сделал больше 15-ти проектов практически под все основные мобильные ОС - (Windows Mobile, Blackberry, iOS, Android).
- В течение двух лет разрабатывал и поддерживал Evernote for Blackberry. Мы сделали из обертки над мобильной версией сайта полноценное нативное приложение.

Nokia Portfolio Challenge

Evernote for Meego Harmattan (ноябрь 2011 - декабрь 2011)

За месяц разработал единственный на тот момент клиент Evernote для Meego Harmattan, с которым до того не работал совершенно. В итоге выиграл конкурс, получил денежный приз, приложение за время жизни получило около 8к установок по миру. В дальнейшем открыл доступ к коду, он был использован для разработки прочих клиентов Evernote для Meego-based систем, как стартовая точка ([BlueWhale](#) or [EverSome](#)).

Образование

[Новосибирский государственный университет](#), рейтинги:

- Бакалавр, информатика и вычислительная техника: 2009 - 2011
- Магистр, информатика и вычислительная техника: 2004 - 2009

Выступления

- [Thinking like an IDE](#)

Конференции

- 2019:
 - [#Pragma](#)
 - [try! Swift NYC](#)
 - [mDevCamp](#)
 - [Mobius](#)
 - CodeMobile
 - iOSCon
- 2018:
 - iOSCon
 - [Mobile Era](#)
 - [MBLT Dev](#)
 - [ServerSide.swift](#)
- 2017:

- [#Pragma](#)
- [ADC](#)
- [NSSpain](#)
- [MobiConf](#)
- [MCE](#)
- [iOSCon](#)
- 2016:
 - Apps World London
 - [MobiConf](#)
 - [GDC Europe](#)
 - [mDevCamp](#)
 - [MCE](#)
 - [mdevcon](#)

Материалы и публикации

- В блоге:
 - [Блог AppCode](#), October 2015 - March 2022
 - Статьи в web help AppCode:
 - * [Quick start guide](#)
 - * [Migration guide](#)
 - [Habrahabr](#)
- Видео / скринкасты (видеоряд, как правило, создан мной; озвучка делалась либо developer advocate, либо подрядчиком, финальный монтаж - на мне):
 - [Quick Start Guide](#).
 - [Create a SwiftUI application in AppCode](#).
 - [Debugging in AppCode](#).
 - [Getting started with Git in AppCode](#).
 - [Testing in AppCode](#).
 - [Database support in AppCode](#).
 - [Swift Package Manager support in CLion](#).
 - [Server-side Swift in CLion](#).
- Плагины / тулы / части IDE:
 - [IDE Features Trainer plugin](#) - переработал ядро языковой поддержки, сделал часть для Swift.

- [SwiftLint plugin](#) - сделал первую версию плагина, передал затем другу в поддержку.
- [Semantic highlighting](#) для Swift.
- Поддержка шрифтов San Francisco (системные для macOS) во всех IDE от JetBrains.
- Переработал встроенные цветовые схемы для редактора в AppCode.
- Переработал встроенную раскладку, имитирующую дефолтную раскладку Xcode.

Прочие интересы

- В свободное время делаю проекты по embedded (например, [вот такие](#)).
- Играю на MusicMan Bongo 5, езжу на байке Ghost SE 1800.