

10.02.2021

Экономика сопровождения программного обеспечения

Программисты бесконечно спорят о том, является ли их профессия искусством или инженерией. Профессиональное звание программиста может содержать слово "инженер", а подразделение в компании может объединить всю разработку программного обеспечения в категорию "инженерия". Однако, если вы некоторое время проводите с программистами, вы, вероятно, слышали слово "изящно" в описании некоторых неопишуемых качеств программного обеспечения. Эти качества соответствуют эстетическим принципам, которые больше связаны со вкусом и стилем, чем с функциональными требованиями.

Программное обеспечение преодолевает неуверенный и непонятый разрыв между очень практическими потребностями бизнеса (создание отчетов по заработной плате, управление крупными машинами на заводе, моделирование пластикового корпуса потребительского устройства) и более абстрактными требованиями людей, ответственных за проектирование, реализацию, развертывание и поддержку программного обеспечения (легкость изменения, простота понимания, возможность мониторинга). Этот разрыв не является пропастью между инженерией и искусством; во многих отношениях, качества, которые разработчики хотели бы видеть в программном обеспечении, представляют собой компромиссы, присущие многим инженерным дисциплинам, в то время как бизнес-требования, которые должно выполнить программное обеспечение, имеют свою собственную нереальность.

Однако любое программное обеспечение, написанное для достижения бизнес-целей, должно соответствовать хотя бы одному основному стандарту: обеспечивать достаточную ценность, чтобы оправдать инвестиции, необходимые для его создания, развертывания и поддержки. Если производитель изделий тратит 100 миллионов долларов на строительство нового завода, но может получить только 50 миллионов долларов выручки за срок службы этого завода, инвестиция провалилась. (Цифры оказыва

ются хуже, чем кажутся сначала; предположительно, первоначальная инвестиция была чистой текущей стоимостью гораздо большей суммы. Другими словами, компании-производителю изделий было бы лучше инвестировать в что-то, что приносит даже 1% или 2% дохода, чем инвестировать во что-то, что теряет деньги. Это может показаться очень странным и педантичным моментом, но даже простое окупаемость инвестиции часто считается провалом с бизнес-точки зрения.)

Какова рентабельность инвестиций в программное обеспечение?

Бизнес может обеспечить ценность с помощью программного обеспечения одним из двух способов: увеличивая доходы (получая больше денег) или повышая производительность (снижая затраты). Это те же два варианта, что и у большинства других бизнес-операций, за исключением финансов (в качестве несвязанного примера рассмотрите, как выкуп акций может улучшить финансовое положение инвесторов в публичной компании). Многие компании рассматривают программное обеспечение, в целом ИТ, как центр затрат, в который поступают деньги и, в идеале, производительность улучшается по всей компании. Хотя новый идеал Кремниевой долины компаний, которые предоставляют услуги, основанные исключительно на программном обеспечении (облачный хостинг, онлайн-сбор платежей, видеоконференции), имеют иное представление о экономике программного обеспечения, многие или большинство компаний не видят программное обеспечение как средство увеличения доходов.

Если это так, то как компания измеряет эффективность своих инвестиций в программное обеспечение, предназначенное для повышения производительности?

На практике, плохо.

Представьте юридическую фирму без электронной почты, телефонов с несколькими линиями или даже компьютеров с текстовыми процессорами. (У него все еще есть факс и копир и кулер для воды.) В какой-то степени, идея о том, что у каждого адвоката, помощника адвоката и юридического ассистента должен быть компьютер на ее столе и доступ в интернет, так же является частью затрат на ведение бизнеса, как и идея о том, что у каждого адвоката должен быть стол. (Кто измеряет эффективность партнера в этой фирме, который никогда не учился печатать и который пишет все свои документы от руки, ожидая, что

юридический ассистент, который выставляет счет по 70 долларов в час, расшифрует его заметки? Конечно, не б

едный клиент, который не видит этой неэффективности в счете!)

Дорогая юридическая библиотечная служба, такая как Lexis-Nexis или Westlaw, будет утверждать, что траты тысячи долларов в год на одно рабочее место сэкономят сотрудникам часы усилий, потраченных на изучение библиотеки пыльных законов, которые медленно устаревают в том коридоре за кафетерием. Они даже могут назвать долларовую сумму этой ценности. Доверяете ли вы этой цифре? Вы можете измерить стоимость исследования с и без этой услуги и выяснить, как вы можете удовлетворить клиентов или выставить больший счет с и без этой услуги. Вам будет сложно измерить, насколько эффективнее ваша работа с широким доступом к цифровой библиотеке и более крупным индексом, однако. Удалось бы вам составить успешное исковое заявление, если бы вы не наткнулись на эту интересную юридическую теорию, которая случайно появилась в поиске? Наткнулись бы вы на соответствующий прецедент, просматривая полки вживую?

(Бизнес-ценность подписки на такую службу сильно разнится, если вы ведете внутреннее юридическое обслуживание. Производительность для одного клиента, который платит вам зарплату, очень отличается от производительности для множества клиентов, которым вы напрямую выставляете счет.)

Конечно, решение о лицензировании программного обеспечения сильно отличается от решения о создании программного обеспечения. Одна из причин, по которой Lexis-Nexis и Westlaw стоят так дорого, заключается в том, что они несут ответственность за работоспособность своего программного обеспечения год за годом. В отличие от печатной книги, они не могут выпустить единственное издание, которое прослужит до тех пор, пока не отпадет переплет и не поблекнет печать. Эфемерные качества программного обеспечения и постоянное развитие техники означают, что программное обеспечение требует постоянных инвестиций в поддержание, чтобы оставаться полезным. Если вы хотите, чтобы программное обеспечение делало больше (приносило больше дохода, улучшало производительность дальше), вы должны соответственно вложить больше ресурсов в программное обеспечение.

Если вы не инвестируете ничего в программное обеспечение, оно будет все больше отставать по полезности. Это одно из слепых пятен, которые компании часто демонстрируют, когда они рассматривают программное обеспечение.

Если бы вы строили математическую модель полезности программного обеспечения и его возврата, вы могли бы начать с идеи, что решение о том, стоит ли инвестировать в создание программного обеспечения, должно выглядеть примерно так:

$\$инвестиции \leq \text{Чистая приведенная стоимость } \ожидаемого_дохода

Другими словами, если вы полагаете, что создание программного обеспечения обойдется вам в 1 миллион долларов и вы ожидаете, что оно принесет 1,1 миллиона долларов нового дохода или улучшения производительности за год, и если 10% возврат приемлем, то у вас все идет хорошо.

(Да, финансовый расчет отличается в зависимости от того, является ли ценность, которую производит программное обеспечение, выручкой или прибылью, но если вы уже знаете это, вы также знаете, как делать математический расчет по-другому, и это не меняет сути.)

Отбросим тот факт, что очень сложно оценить стоимость создания программного обеспечения, прежде чем вы его создали (дизайн программного обеспечения чрезвычайно сложен из-за его внутренней сложности, измерение прогресса в программном обеспечении сложно из-за его эфемерности, и неизвестные тенденции умножаются в областях, где растут нематериальные активы). Это прогнозы. Вы можете потратить меньше. Вы можете получить больше. Вы должны рассматривать эти числа как наиболее понятные вам оценки. По меньшей мере, это уравнение поможет вам измерить результат инвестирования в программное обеспечение. (Пройдите его в обратном порядке, чтобы выяснить, сколько вы должны были заплатить за ценность, которую вы действительно осознали.)

А что, если вы хотите смоделировать затраты на поддержку программного обеспечения? Стоит ли инвестировать в его поддержание? Формула выглядит примерно так же, за исключением того, что на этот раз она включает ожидаемые годовые затраты на поддержку:

$\$ \text{начальные_инвестиции} + \text{дисконтированная стоимость } \$ \text{затрат на техническое обслуживание} \leq \text{Чистая приведенная стоимость } \$ \text{ожидаемого_дохода}$

Учет ожидаемых затрат на обслуживание на протяжении всего срока жизни проекта помогает учесть те способы, которыми вы могли бы лучше использовать эти деньги позже. (Доллар сейчас стоит больше, чем доллар в будущем. Конечно, вы можете не платить эти затраты на обслуживание сейчас, но если вы решаете инвестировать в это программное обеспечение сейчас, вы берете на себя эти затраты на обслуживание таким же образом, как взятие кредита означает, что вы берете на себя затраты на обслуживание долга на срок жизни кредита.)

Этот последний расчет намного менее надежен, чем первый, по простой причине: его переменные представляют больше неизвестных качеств. Как вы измеряете затраты на обслуживание программного обеспечения?

Источники затрат на обслуживание программного обеспечения

Обслуживание программного обеспечения зависит от множества факторов. Этот список не является исчерпывающим, и не все факторы применимы во всех ситуациях:

- стоимость поддержания аппаратной платформы для его работы
- лицензионные платы за аппаратные и программные зависимости
- стоимость IT-работников для мониторинга оборудования и программного обеспечения
- затраты на обучение новых пользователей
- затраты на перенос на новые платформы
- стоимость исправления ошибок
- стоимость добавления новых функций
- затраты на взаимодействие с существующими или новыми системами
- затраты на обучение новых сотрудников поддержки

За исключением некоторых специализированных ситуаций (например, монополия, монополия или картель, где поставщики могут устанавливать баснословные лицензионные или поддерживающие платежи), самой большой единичной стоимостью обслуживания программного обеспечения является персонал. Отсюда и последний пункт в списке: если вы создали программное обеспечение, на котором зависит ваш бизнес, вы обнаружите, что платите за экспертизу людей, которые могут его поддерживать. Если вы их не удержите, вы заплатите как прямые, так и косвенные затраты на найм и удержание людей, которые должны узнать детали этого программного обеспечения.

Измерение этих затрат сложно. Но также сложно измерить затраты на потерю производительности и, особенно, на упущенные возможности.

Управление расходами на обслуживание программного обеспечения

Если управление программным обеспечением кажется устрашающим, то это потому, что это действительно так. Сущностная природа программного обеспечения представляет сложность, которую можно только контролировать, но никогда полностью не удержать. Любой современный проект программного обеспечения, работающий на универсальных компьютерах (сервер, ноутбук, планшет, даже телефон), зависит от взаимодействия десятков или сотен миллионов строк кода, нескольких уровней оборудования, а также бесчисленных поставщиков и разработчиков. Эти компоненты меняются ежегодно.

Природа бизнеса не менее хаотична, особенно из-за присутствия крупнейшего источника хаоса в обоих предприятиях: людей. Действия или бездействие одного сотрудника могут стоить компании миллиарды, в то время как действия или бездействие разработчика программного обеспечения могут привести к появлению ошибки или реализации функции, которые могут сделать новые идеи возможными или вызвать бесчисленные ошибки.

Отношение к программному обеспечению как к центру затрат, к сожалению, может усугубить отрицательное.

Возможно, основной трудностью в понимании стоимости программного обеспечения является его сложность. Какие из десятков миллионов строк кода оказывают наибольшее влияние на ваш бизнес? Если бы у вас было всего несколько тысяч долларов для инвестиций в внесение одного изменения, где бы вы увидели наибольший результат? Даже если бы у вас была возможность сосредоточить свои усилия в одном месте по всему стеку, слишком много выбора для оценки. Вы потратили бы огромное количество времени, усилий и денег на рассмотрение всех ваших вариантов.

Возможно, проблема слишком велика. Возможно, затраты и ожидаемые значения легче оценить в меньших количествах.

Например, что если у вас есть команда, полная работников, которые каждый день тратят час на перенос данных из таблиц в веб-прилож

ение. Если каждый работник стоит вам \$14 в час, и у вас десять работников, это \$140 в день, \$700 в неделю и \$35,000 в год.

Предположим, разработчик, который стоит вам \$700 в день, потратит неделю — \$3500 — на реализацию функции загрузки таблиц и прямого импорта данных. Через пять недель вы окупите свои инвестиции, улучшив продуктивность десяти сотрудников.

У этого подхода, конечно, есть свои ограничения. Не все функции так легко количественно оценить, да и они не все так эффективны. Однако хорошее в программных функциях в том, что они, как правило, усиливают друг друга. Каждая реализованная возможность автоматизации открывает потенциал для дальнейших возможностей. (Ваши работники не только освобождаются для работы над другими вещами, но и испытывают меньше усталости, и, следовательно, у вас может улучшиться удержание персонала.)

Неразумно ожидать, что вы сможете измерить прямые экономические выгоды каждой написанной строки кода, но возможно проанализировать ожидаемые и реализованные выгоды каждой реализованной функции с определенной детализацией. Возможно, правильная детализация на уровне рабочей недели разработчика. Возможно, это день. (Требуемый уровень детализации напрямую связан с объемом прямого контроля над процессом разработки. При точных измерениях на еженедельном уровне возможно свести в общую сумму затраты, ожидаемые выгоды и реализованные выгоды.)

Эта модель, конечно, требует базового набора затрат на поддержание работы программного обеспечения в целом (аппаратное обеспечение, лицензирование, доступный персонал для чрезвычайной поддержки), а также ожидаемые периодические выгоды (если программное обеспечение перестанет работать, каковы будут затраты на производительность или доход?). Однако, учитывая эти затраты, можно проанализировать ожидаемую оставшуюся стоимость программного обеспечения по сравнению со стоимостью замены или устаревания.

Мало компаний справляются с этим хорошо.

Программное обеспечение плохо понимается на всех

уровнях, особенно в части затрат и выгоды. Но так необязательно должно быть. Даже такая простая экономическая модель, как представленная здесь, может дать представление о том, что вы покупаете и что получаете. Даже упражнение в форме вопросов о затратах и выгоде на уровне недели работы разработчика может раскрыть информацию, которой у вас ранее не было — информацию, которая может помочь вам принимать лучшие решения о том, как управлять и оплачивать технологии.