

Как не надо программировать

Преамбула. Работаю в США с 1990 года. Работал software developer, technical leader, software development manager, в разных компаниях. Накопил некоторый опыт работы, которым и хочу поделиться в ненавязчивой форме.

Тут вся братва говорит напиши как надо программировать да напиши. А чего подумал я, кучу лет я это делаю. Вот возьму и напишу. Ну сначала я лучше расскажу как это бывает, потом расскажу как не надо программировать, а если вам кто-нибудь скажет, что он знает как надо не верьте. Для начала про государственные контракты ну, а потом про другие и вообще как пойдет.

Как это бывает

Ну сначала много бессмысленных людей совершенно не понятным образом получают контракт. Это я так – для прикола сказал, что не понятно, на самом деле понятно. Любая контрактная кампания, если она хочет остаться таковой, первым делом нанимает бывших правительственных работников. Убиваем сразу двух зайцев. Во первых, они всех там знают, во вторых, те кто еще там, могут рассчитывать на эту кампанию после выхода на пенсию. Остальные бегающие, это балласт – они создают шум, генерируют документацию и мешаются. А если контракт получили, то другая группа обеспечивает не формальные отношения на фоне гольфа, душевных разговоров и «братования». Ежели всего этого нет, то на следующий срок могут и послать, вне всякой зависимости от технического качества.

Предположим проект новый. Сначала забираем деньги по максимуму. Сажаем кучу народу и все опять бегает в полной конфузии. А кого только там нет.

Тут мне придется переключиться на английский, не знаю русских терминов. Project managers, Quality Assurance managers, Configuration managers, Testing managers, Just Managers, Random People. И совсем не много тех кто чего-то понимает. А со стороны заказчика не лучше. Если есть хоть кто-нибудь, кто хоть немного знает, чего надо делать, это большая удача. Его необходимо идентифицировать как можно быстрее и вынимать из него это знание как можно быстрее, а то или сдохнет или уйдет. Что очень не просто, потому, что он думает совсем по другому и ни хрена не понимает в “software development”. Но с ним надо быть ласковым и понимающим – упустишь не будет того чего надо (“functional or user requirements”).

Процесс определения того чего надо самый трудоемкий и сложный. Потому, что кроме того кто знает есть еще много тех, кто думает, что знает и еще больше тех кто притворяется, что знает. Ну поди разберись! Однако свои тоже очень мешаются. Они суют заказчику всякую ерунду типа крутых методологий и “detailed schedule” на следующие 5 лет поминутно. А среди своих надо отыскать того, кто понимает чего говорит тот, кто знает и может это внятно навалить на бумаге. Таких тоже мало.

Ну худо бедно определились. Тут бы и работать. Но ... Сейчас попробую пояснить.

Начнем из далека. “software development “ шибко молодая индустрия. И шибко прыгучая. Полно неграмотного и неквалифицированного народа. Вы когда нибудь видели чтобы электрик унитаза чинил, а архитектор собственноручно клал кирпичи, а маляр рассчитывал конструкции?

- Ты кто такой будешь?

- Ну, я это, код могу писать...

- А ты, енту, как ее тудить в коромысло, будь она не ладна, жаву, кумекаешь?

- Ну я, это тут маленько сервлетками баловался...

- Вот и славно, стало быть буишь у нас за главного едрить архитекта. Буишь жейтуи (J2EE) рассекать, крутую

дизайну делать и ваще...

- Да я это, токмо, код

- Да ты не ссы, пацан, не боги горшки обжигают...

Чтобы дом строили по приблизительным чертежам используя не опробованные технологии? Не видели, а в этой индустрии сплошь и рядом. А вы видели заказчика который просит в разгаре строительства десятиэтажного дома добавить еще три этажа и сарай в придачу и соединить их мостом. А тут нормальненько, в порядке вещей.

- А это чего то у вас тут на экране, срань какая-то!?

- Ну как же вы нам сказали что без этого низзя

- Кто? Мы?

- Ну эта, ну юзеры...

- Какие такие юзеры!?

- Ну обнакновенные. Вон там стояли

- Да это не наши юзеры. Эта с другого департамента...

Когда строят дом все понимают, что это такое будет до того. Когда строят "software" то понимают только после и то не все. Но многие думают, что судьбу можно обмануть. Они говорят мы тут щас по бырому прототип настрогаем на коленке покажем заказчику и отточим функциональность, а уж потом... Увы потома практически никогда не бывает. Потом избушку на курих ногах надстраивают до небоскреба. Он все время почему-то разваливается.

Ну как их, методологии

А как же методологии? Методологии простые должны быть. Когда сложно и замысловато то не работает. А просто это приблизительно вот таким макаром:

- В конце концов определится с тем чего надо (functional requirements) и на бумажке изложить.

- Вцепиться в того кто знает и выяснить про все шаги и варианты которые юзеры будут осуществлять. Называется "use cases"

- Ежели гуя (GUI) нужна то назююкать на основании "use cases" "screen flow" и пристать с ентим к тем кто знает штоб проверил. Ну енто просто какие экраны, какие данные там есть и откуда куда можно попасть

- Ежели нужно общаться с software из другого software то определить APIs (интерфейсы сталобить шоб вызывать разну функцанльность)

- Ущучить из того чего надо "business objects" (для этой цели нужен крутой архитектор рассекатель, их оченно мало, не брать того пацана котрый сервлетками баловался и избегать теоретиков произносящих умные слова, а при пристальном взоре выясняется, что последний раз писал код на коболе в 74 году, ох сколько таких расхаживает с проекта на проект и пузырь надувает)

- надывать из того чего надо "business rules" и "validation" (архитект должен долго и нудно мучить того кто знает вместе с тем кто может внятно изложить на бумаге)

- присандалить "business rules" и "validation" к "business objects"

- попробовать состыковать "screen flow" с "business objects"

Тут как правило полная ерунда получается. Затык. Потому что "screen flow" и "business objects" разными людьми производились.

Так, тут, я господин, читатель, маленько обосрался. Забыл одну важную заковыку. Это prototyping называется. У ней много всяких причин имеется. Ну к примеру взбрело заказчику чегонибудь свеженого там relational database to XML или Add hock Web based reports или еще какуюнибудь загогулину. Ежу понятно что самим ломаться глупо, этого всего писано-понаписано. Но выбрать не тривиально. Сажаем братву, чтобы осуществляли prototyping. И шоб начинали с "open source". Ну тут все производители как коршуны налетают и начинают свое пихать. Увы не всегда рекомендации братвы по результатам прототипирования совпадают с решением принимаемым мудрым руководством. Или хочут оне, чтобы туча юзеров одновременно обслуживались сервером с офигительной скоростью и штоб 24x7. Это значит круглосуточно. Ну это нужно проверить. Вообще весь этот prototyping надо в параллели со вторым пунктом осуществлять.

Щас погоди, мы где остановились та... А.. вспомнил. Худо бедно состыковали "screen flow" с "business objects". А тут, хлобысть нам про меж глаз, тот кто знает на стороне заказчика вспомнил одну «малюсенькую» деталь, которая в корне меняет практически все. И теперя все по новой. А срока то не сдвигаем. Доблестный менежмент бьет себя в грудь и обещает заказчику «шо усе буит в срок». А братва уже аж вспотела вся. В конце концов все просто устают елозить туда-сюда и помолясь к дизайну приступают. А тут, что за черт, оказывается через два месяца уже надо первый релиз сдавать. А у нас и не стояло...

Как насчет дизайну

Всенепрременно дизайн необходим. А какой? А такой который удовлетворяет известным функциональным требованиям с возможностью его дополнить. А может давай его "generic" сделаем, чтобы удовлетворял еще не известным. Заказчик вчерась в носу поковырялся, глаза закатил и размечтался: «А в во втором релизе мы вот тут финтфлюшек желаем и чтобы ваще...». Наши быстренько все записали и велели углубить и одженерить дизайну, чтобы в будущем мог даже на дудке играть и макароны варить. Позвольте господу вас спросить, а будет ли этот второй релиз? Ежели мы тут будем такой обалденно женерик дизайн фигачить то и первого релиза не будет. А кто вам сказал, во втором релизе финтифлюшки понадобятся, может и вовсе нас попросят кандебобером плясать? Тот кто знает, отвечаете вы. Практика убедительно показывает, что никто, никогда не знает, что будет и какие финтифлюшки понадобятся. И если уж по полной правде то перый релиз это «прототип на стероидах». Знание и понимание того, что надо наконец приходит когда этот первый релиз готов.

Ну чего, будем код писать?

Так он уже готов ентот код. Это последний прототип. На настоящий код времени нет. Ну иногда все-таки пишут код, приблизительный такой код (я вам потом проясню, что такое приблизительный код). А как код писать, вы все сами знаете. Ну все, абсолютно, все знают как код писать! А я так считаю, правильный код это такой код где нет этих, как их там, условных переходов (if statement), совсем нет и все! Потому что правильное решение оно как известно одно, какие там варианты. Циклы (loops), будь они не ладны, отменить, ну какого хрена, хоть один нормальный человек будет делать одно и тоже практически бесконечно. Так, что у нас еще, а в базу пихать и обратно вынимать, тоже бред каой-то. Ну скажите мне зачем пихали-то, ведь тут-же хотят обратно. А про энту File I/O просто не говорите мне. В проклятой жаве куча каких-то классов наделано, а все равно кажанный раз вокруг код надо зююкать или в open source лазить. UI просто отменить, действует на нервы деталями и мелочевкой. А всю эту бессмысленную HTML запретить под страхом отключения итернета. Ну сами подумайте у вас на столе Pentium 4 а вы только можете HTML интерпретировать и гонять всю эту кучу дерьма туда и обратно засоряя и без того загаженный интернет. А поменялось то одно поле!

Ну в общем ежели дизайна сложилась, то код из нормального программиста вылетает мухой. Главное не забыть прокомментировать ентот код.

Удивительные, я вам доложу, комментарии можно повстречать если покопаться в коде. Ну например, как вам нравится комментарий, что дальше код идет не правильный и автор сам не уверен будет ли он работать. Или, что дальше код очень плохой и автору стыдно за него.

Удивительное дело, но в конце концов, внезапно выясняется, что писать больше нечего. Приходится

тестировать.

Тестируют все!

Особенно хорошо тестируют сами программисты. Все тесты проходят на ура. Тестировать ведь нужно строго то, что закодировал, весь ваш error handling мы в гробу видали. Вообще приличный программист с тестером даже за руку не здороваются. Потому как на оси эволюции программист это человек прямоходящий, а тестер как бы питекантроп. Тестер он ведь сам ничего навалить не может, поэтому, чужой код и ломает. Особенно программист нелюбит "anal testers". Это которые въедливые, и прицепляются со всякой фигней.

Когда закончим мужики?

Ну а как понять сколько нужно времени, чтобы все это забалтывать? Опыт и интуиция плюс и несколько формальных правил. Сколько нужно времени чтобы определить то, что надо, ваще не понятно, про это даже думать глупо. А про дизайн, кодирование, тестирование используем следующую адгоритму. Спрашиваем братву сколько-сколько у вас классов нарисовалось, а сколько экранчиков, а табличек в базе? Берем бывалого программиста за жопу и с пристрастием допрашиваем сколько ему нужно времени чтобы выдать на гора средней сложности экранчик, и класс и допрашиваем дибиея (DBA) про базу. Умножаем енто число на количество экранов, классов, табличек и еще на два, а потом умножаем на фактор невезения (bad luck factor). Фактор невезения бывает всегда, это когда все идет как по маслу, а потом облом и кранты. Нет ни одного проекта без обломов и крантов. В зависимости от степени фактапности (fucked up) проекта ентот фактор варьируется от 1.5 до 1.8. Потом выясняем количество штыков и пытаемся понять каков коэффициент параллельности. Ну это когда что-то можно параллельно рубать. Одни штопают экраны другие наяривают middle tier, третье базу. Предположим четверо гавриков шуруют middle tier, это вовсе не значит, что они это сделают в четыре раза шустрее чем один. Только в два. А шестеро только в два с половиной. С учетом всех этих соображений можно ответить на вопрос: «Когда закончим, мужики» при средней длительности проекта от трех до восьми месяцев, с точностью до недели. Не верите? Зуб даю! Проверено неоднократно.

Люди базы

Целью жизни для людей базы является неумное стремление нормализовать базу до десятого уровня нормализации и при этом они испытывают состояние близкое к оргазму. Кроме того в их задачу входит полное устранение middle tier, в виду его полной ненужности, а чего, говорят они, все здесь, в наших родненьких stored procedures и ненаглядных triggers. А к этим таблицам не будет вам прямого доступа, щас вам view навалием. А будете выпендриваться вообще лишим доступа. В идеале кроме базы ничего не должно существовать. База есть вещь в себе и для себя!

Все про менеджеров

Кто не может програмировать идет в тестеры, кто не может тестировать идет в менеджеры. Ну это я конечно слегка преувеличил, для красного словца. Но некая доля истины здесь имеется. Между нами говоря, незавидная у менеджера доля. Он ведь прекрасно осознает, что без него можно обойтись. Поэтому, бедолага, ведет себя не всегда адекватно. А тут еще надо с клиентом умные разговоры вести, заседать на бесконечных митингах с начальством и выслушивать этих проклятых программистов. Если так получилось, что вы менеджер, очень не повредит:

- иметь некий опыт программирования
- попробовать понять в общих чертах чего нужно делать
- попытаться понять в общих чертах чего уже сделано
- обращаться к программистам по техническим вопросам
- никогда не говорить нет или да заказчику и не давать обещаний предварительно не посоветовавшись с программистами.

Менеджеры бывают нескольких видов.

Менеджер – линейное звено. Этот менеджер имплементирует функцию у которой на выходе то-же самое, что и на входе. Полностью контролируемая особь, поддается дрессировке, послушен и покладист. Толку от него как от козла молока но и вреда не приносит. Великолепный переадресатор e-mails в обе стороны. На программистов смотрит преданно и дает лапу.

Менеджер – генератор хаотического шума. Этот менеджер имплементирует функцию случайных чисел. Практически не контролируем, иногда агрессивен, иногда ласков. От него можно ожидать практически всего. Часто бывает женского пола. Выдает клиенту случайную и ничем не обоснованную ерунду. Периодически просит рассказать, что-же такое XML и почему он перевернул мир.

Менеджер – приносящий пользу. Этот вид встречается очень редко и занесен в красную книгу. Может внятно донести до клиента простую техническую мысль. Иногда дает разумные советы клиенту и полностью осознает круг своего незнания. Служит буфером между враждебным и бестолковым внешним миром в лице клиента и начальства и программистом.

Ошивающиеся

На любом проекте всегда появляются люди которые просто ошиваются. Никто никогда не знает, что они делают, они просто ошиваются, ошиваются на митингах и в коридорах. Ошиваются с умным, сосредоточенным видом, что-то записывая в блокноты и периодически рассылая абсолютно не нужные бумажки. Ярким примером являются ошивающиеся по причине Quality Assurance. Или например блюдущие Security.

Величайшей загадкой является предназначение этой Quality Assurance. Не подвластна тайна сия простому смертному.

- Чего-чего вы делаете? Вопросает наивный программист.

- Мы следим за тем, чтобы вы, засранцы, все делали согласно утвержденным стандартам. - Каким стандартам?

- Вот этим.

- Дык, ежели этим стандартам следовать то на дизайн, кодировку и тестирование времени не остается, одни бессмысленные документы буим калякать

- Ну, между нами говоря, мы понимаем, что это не реально

- Дык, чего тогда...

- Ну мы тоже кушать хотим и клиент платит

- Ну хрен с вами, ошивайтесь по малу, только под ногами не мешайтесь

Индусы в деле

А причем тут индусы, спросит наивный читатель. Причем, ой как причем. Индусы заполняют software industry как тараканы. Обладают «запахом и вкусом» которые создают специфическую атмосферу, поэтому нельзя не коснуться этой животрепещущей темы. Введем несколько ключевых понятий. Одно из основных это индокритическая масса. Индокритическая масса возникает при наличии хотя бы одного индуса менеджера и пары-тройки индусов программистов. Следующее понятие это – индоцепная реакция. Индоцепная реакция возникает спонтанно при наличии индокритической массы. Приводит к бурному и неконтролируемому увеличению индокритической массы. Оновной функцией индокритической массы является политическая деятельность, программирование это побочный продукт. День прошедший без политической интриги считается полностью пропавшим. Элементы индокритической массы обмениваются информацией с околосветовой скоростью и обладают невероятной говнистостью. Мозг индуса программиста так хорошо натренирован на многоходовых политичеких итригах, что программирование дается ему играючи. Задачей любого программиста не индуса является не допущение индокритической массы.

Русская братва

Ну что, уважаемый читатель, щас буишь про себя читать. Не боись, надо же и когда нибудь правду узнать про себя. Русский программист (в основном еврей) хотя последние лет пять-семь и коренных русачков можно повстречать, делится на две категории - на программистов и програмахеров. Я надеюсь не надо объяснять, что такое програмахер. Интересно, что частенько програмахеры эволюционируют в программистов. Во времена «золотой лихорадки» практически все превратились в програмахеров. Суровые годы лопнувшего экономического пузыря приостановили процесс тотальной програмахеризации. И тем кому не удалось

эволюционировать в программистов пришлось туго.

Каковы же отличительные черты русского программиста? Русский программист никогда не чинит чужого, бессмысленного, объектно не ориентированного, спагетти кода.

- Че, не работает?

- Ща мы енто дерьмо выкинем и мухой напишем наш родной, мудрый, объектно ориентированный, офигительный код

- Усе, готово

- А протестировать...

- Че, тестировать!? У нас код работает правильно и всегда!

Русский программист немедленно сносит всю операционку и ставит свою. Пользуется только "cracked software" и "open source". Скорость генерации кода приближается к световой. При наличии трех четырех русских программистов на проекте характерен туннельный эффект самопроизвольного возникновения кода. Русский программист говорит по русски, даже с представителями других национальностей. Предпочитает использовать русские матерные выражения для сообщений об ошибках. При подходе менеджера кладет ноги на стол и продолжает говорить по телефону.

Ну, а как же отличить програмахера? Програмахер практически не заметен.

Очень тихо сидит за своим столом и шепотом выясняет по телефону у знакомого русского программиста, что же ему делать с массивом. Все время улыбается и делает вид, что не понимает по русски. Всем своим видом дает понять, что начал програмировать на вижувал бейсике еще в третьем классе. Несчастен и очень устает на работе.

Немного про китайцев

По сравнению с индусом китаец практически безвреден. Китайский программист это такая большая, раскосая, офигительно усидчивая жопа. Ежели материализовать все то время которое китаец тратит на написание кода в дрова и поджечь, то от кода останется одна большая дыра. Китаец никогда не переписывает, китаец трудолюбиво шьет заплатки, при достаточной длительности проекта слойность заплаток достигает бесконечности, но при этом, как это не парадоксально, перед вами все равно первоначальный код. При наличии некоего количества китайцев на проекте необходимо покидать здание на время ланча. Подогреваемая пища может убить неподготовленного фантастическим ароматом.

Commercial Software Development

Пришла пора поговорить о Commercial Software Development. Это все, что не является государственным заказом, а производится на продажу или для внутреннего потребления. Ежели наивный читатель полагает, что тут дело обстоит получше, чем в доблестном государственном секторе, то он, увы жестоко ошибается. Хотя есть некоторые отличия, о которых я коротенько поведаю.

Заказчиком является Marketing или Business departments. Похоже, что эти ребята в массе своей являются смесью Хлестакова с Остапом Бендером.

Они замечательно научились делать вид, что знают, что можно будет продать через пол года. Но поведать об этом увы не могут, потому как беспрестанно мотаются по важным митингам и надувают пузырь. Эти люди обладают замечательной неспособностью концентрироваться и излагать свои запутанные мысли, могут бесконечно генерировать пустые слова и они всем своим видом дают вам понять, что им доступно высшее знание и только они знают куда дуют «ветры рынка».

- Во что бы то ни стало необходимо буквально завтра выпустить на рынок этот продукт, а то проклятые конкуренты все нахрен захватят!

- Ну, а функциональность то какая?

- Ну как, нам надо, это, и то, и что бы плясало

- А поподробней

- Некогда нам с вами разговаривать, у нас презентация, встреча с потенциальным покупателем и разъезды

- Ну чего делать то...

- Да отстаньте вы, с вашими глупостями...

- Ну чего, мужики, будем сами функциональность изобретать...

- Да, что это вы нам такое нагадили!?

- Ну, вы же заняты все время

- Все поменять!!!

- Дык, расскажите

- Да некогда нам, что вы как дети малые

А потом вдруг бац, всех продали, все верх дном, кто где, ничего не понять. Новые людишки, как из под земли. Вроде новые, а от старых не отличишь.

- Это кто вам сказал, что это "hot product"?

- Да вот, которые до вас тут...

- Кто. Эти, да они ни уха ни рыла !!!

Ежели и удастся чего нибудь, произвести, то частенько продукт уже давно не нужен или столько потратили на его создание, что аж дух захватывает, лучше убрать с глаз долой и опять презентации, митинги с потенциальными заказчиками, разъезды, разъезды...

Заключение

Надеюсь развлек и не обидел.

Автор: Артур Скальский © Газета.Ru IT'S MY LIFE..., МИР 👁 4583 08.01.2006, 18:36 📄 377

URL: <https://babr24.com/?ADE=26887> Bytes: 21975 / 21825 Версия для печати

 [Порекомендовать текст](#)

Поделиться в соцсетях:

Также читайте эксклюзивную информацию в соцсетях:

- [Телеграм](#)

- [ВКонтакте](#)

Связаться с редакцией Бабра:

newsbabr@gmail.com

Автор текста: **Артур
Скальский.**

НАПИСАТЬ ГЛАВРЕДУ:

Телеграм: @babr24_link_bot
Эл.почта: newsbabr@gmail.com

ЗАКАЗАТЬ РАССЛЕДОВАНИЕ:

эл.почта: bratska.net.net@gmail.com

КОНТАКТЫ

Бурятия и Монголия: Станислав Цырь
Телеграм: @bur24_link_bot
эл.почта: bur.babr@gmail.com

Иркутск: Анастасия Суворова
Телеграм: @irk24_link_bot
эл.почта: irkbabr24@gmail.com

Красноярск: Ирина Манская
Телеграм: @kras24_link_bot
эл.почта: krasyar.babr@gmail.com

Новосибирск: Алина Обская
Телеграм: @nsk24_link_bot
эл.почта: nsk.babr@gmail.com

Томск: Николай Ушайкин
Телеграм: @tomsk24_link_bot
эл.почта: tomsk.babr@gmail.com

[Прислать свою новость](#)

ЗАКАЗАТЬ РАЗМЕЩЕНИЕ:

Рекламная группа "Экватор"
Телеграм: @babrobot_bot
эл.почта: equatoria@gmail.com

СТРАТЕГИЧЕСКОЕ СОТРУДНИЧЕСТВО:

эл.почта: babrmarket@gmail.com

[Подробнее о размещении](#)

[Отказ от ответственности](#)

[Правила перепечаток](#)

[Соглашение о франчайзинге](#)

[Что такое Бабр24](#)

[Вакансии](#)

[Статистика сайта](#)

[Архив](#)

[Календарь](#)

[Зеркала сайта](#)