

Экстремальное проектирование?

Рассуждения автора (<http://offline.computerra.ru/2005/589/38836/>) сводятся к простой мысли: кодирование — это не черная и тупая работа.

Кодирование — это принятие решений, и кодер — такой же участник процесса проектирования, как и архитектор проекта.

Следствие первое. То, что мы привыкли называть проектированием (то, что до кодирования), — штука, возможно, нелишняя, но и не принципиальная.

Следствие второе. Тестировать, тестировать и еще раз тестировать. Ибо исходный текст — это просто проект, и только тест позволит убедиться в его корректности.

То есть, по сути, нам предлагается еще одна апология экстремального программирования.

Я убежденный противник экстремального программирования и дочерних практик, таких как Agile Development. Кратко: XP не предоставляет возможность гарантировать сколь-нибудь разумное качество результата. Этот способ применим, только если программисты очень высокого класса. Он не работает на больших проектах, а эффективность многих методик (в частности, парного программирования) вообще никем никогда не была подтверждена на практике или хоть сколь-нибудь логично обоснована.

Простая мысль: если другая методика разработки софта дает на 10% большую предсказуемость результата ценой на 100% (!) больших затрат программистских ресурсов, но без парного программирования, то она уже выигрывает у XP с его парным программированием (которое суть то же самое увеличение затрат на 100%).

На мой взгляд, XP — это своеобразная попытка революционным путем свергнуть «власть» аналитиков и менеджеров над программистами. Действительно — если аналитик ставит задачи невнятно, а менеджер только спрашивает «когда?», а больше не делает ничего, то XP может оказаться наилучшей альтернативой. Но это решение из серии «если летать так и не вышло, давайте хоть быстро бегать». Правильный подход — не заменять нормальные методики разработки софта на дикорастущее XP, а выгнать нерадивых менеджеров и аналитиков, набрать профессионалов и заниматься разработкой программного обеспечения с умом.

В то же время многие практики XP вполне разумны. (Впрочем, увы, большинство реально применимых практик экстремального программирования заимствованы из других методик. То есть само по себе XP принесло мало полезного.)

Утверждение «Исходный код является последней стадией проектирования» имеет такое же право на жизнь, как и утверждение «Дом является последней стадией проектирования». Нравится? Правда, нравится? Давайте я проиллюстрирую построение дома по технологии экстремального программирования.

- Вы (будущий житель) приходите на стройку к рытью котлована и сидите за спиной у тракториста. Воняет? Купите одеколону. Сидеть придется до готовности дома.
- Тракторист спрашивает вас: «Ну, как, котлован достаточно глубокий?» Откуда вам знать? Ну, вы же заказчик. Должны знать. Заодно определитесь с тем, какое сечение должны иметь силовые кабели, можно ли соединять газовые трубы сваркой, и какого типа нужен цемент.
- Когда дом вчерне построен, приезжает эдакая дура с чугунной чушкой на веревке и долбасит по стене. Если дом разваливается, его стоят заново, укрепив получше в тех местах, где треснуло. Пожалуйста — придумайте сами, почему вы должны оплатить строительство трех домов, а получить только один. Я не знаю.
- Когда вы включите в розетку чайник и провода загорятся, ваша претензия не будет принята — в вашем присутствии розетку проверяли втыканием лампочки, и вы согласились, что она работает нормально.

Я предвижу возражение по пункту 3: мол, если дом не выдерживает теста, то второй вам строят бесплатно. Оно столь же наивно, сколь и вера в бесплатную медицину. Если программист писал код трижды, заказчик все равно оплатит его работу, и не важно, будет ему эта сумма выставлена в счете явно или скрыта в завышенной стоимости работы. Чудес не бывает — или всю работу оплачивает заказчик, или исполнитель разоряется. Не достроив, кстати, дома.

Экстремальное программирование — вообще ужасно наивная идея. Например, посадить заказчика рядом с программмером, чтобы он рассказывал, как делать, можно, только если:

- Заказчик ужасно мало зарабатывает и тратить его время не жалко. Правда, неясно, откуда у него деньги на заказ разработки софта. Впрочем, так бывает, если и заказчик, и исполнитель — студенты, а работа сводится к стряпне сайта за полсотни баксов.
- Заказчик на самом деле понимает, что он хочет получить. Но это заказчик из сказки, в жизни они не встречаются. Поэтому существуют аналитики.
- Заказчик помнит, что он хотел вчера и почему он этого хотел. Так тоже не бывает, поэтому если требования к продукту не фиксируются, разработка начинает осциллировать: вчера заказчик просил красненькое, а сегодня он хочет толстенькое, а что красненькое с толстеньким несовместимо — он уже забыл. Посему программмер пять раз переделывает с красненького тоненькое на синенькое толстенькое и назад.

Можно долго говорить о том, насколько XP отличается от детского лепета, но есть и другая сторона медали. Коль скоро XP появилось, тому были какие-то предпосылки. Как бы само решение ни было несерьезно, нужно посмотреть на причины, которые к нему привели, и, может быть, предложить иное решение.

А. Автор утверждает, что проектирование не закончено, пока последняя строка кода не написана и не оттестирована. По сути это означает следующее: никакой проект не транслируется в код 1:1. В настоящее время это утверждение вполне справедливо.

Проблема рассинхронизации кода и проекта общеизвестна, и способов борьбы с ней немало. Однако вряд ли стоит решать ее, низводя проектирование до уровня кода. Правильнее было бы поднимать код до уровня проекта. В обоих случаях происходит некое слияние проекта и его реализации, но задача автоматизированного получения кода по проекту проще, чем задача превращения языка программирования в хороший проектный инструмент. Давайте идти вверх, а не вниз. Не нужно проектировать на C++, нужно сделать так, чтобы код на C++ генерировался из проекта нажатием одной кнопки.

В. Автор фактически подтверждает, что XP ориентирована на высококлассных программистов и не работает, если программист — среднего уровня. «Пусть все программеры будут классными», — говорит автор. Иллюстрирую: не будем проектировать автомобили, не будем давать рабочим технологические карты. Вместо этого найдем таких рабочих, чтобы сами по месту разбирались, что к чему. Наивность этого подхода подтвердил Форд запуском первого конвейера около ста лет тому назад. То есть XP применять можно, но — вот беда! — экономически невыгодно.

С. Терминологический вопрос. Несущественно.

D. В рамках авторской теории это правда, за рамками — не имеет смысла. Так что обсуждать нечего. Резюме: несерьезно. Все это очередной завуалированный разговор на тему «не хочу ничего делать, кроме кодирования и — так уж и быть — тестирования». Почему? Да потому, что не умеет. Так пусть учится!