

Почему рефакторинг работаэ?

Программы имеют два типа ценности: что они могут сделать для тебя сегодня и что они могут сделать для тебя завтра. Бóльшую часть времени, которое мы программируем, мы сфокусированы на том, что мы хотим от программы сегодня.

Исправляем ли мы баги или добавляем новый функционал, мы делаем сегодняшнюю программу более ценной, делая её более способной.

Ты не можешь программировать долго, не понимая, что то, что система делает сегодня – это только часть Истории. Если ты можешь сделать сегодняшнюю работу, но ты это делаешь так, что не сможешь завтра завершить завтрашнюю, ты проиграл. Хотя ты знаешь, что ты должен сделать сегодня, ты не вполне уверен на счёт завтра. Быть может, вы будете делать то, быть что-то иное, быть может то, чего ты себе ещё не можешь представить.

Я знаю достаточно для того, чтобы сделать сегодняшнюю работу. Я не знаю достаточно для того, чтобы сделать завтрашнюю.

Но если я буду работать только для сегодня, то буду не в состоянии полностью работать завтра.

Рефакторинг является одним из способов выйти из этой связки. Когда ты видишь, что вчерашнее решение не имеет смысла сегодня, ты изменяешь решение. Сегодня ты можешь делать сегодняшнюю работу. Завтра часть сегодняшнего понимания будет казаться тебе нивной и ты опять изменишь его.

Что делает программу такой, что с ней сложно работать? На мой взгляд, это следующие 4 вещи:

- Программы, которые сложно читать, сложно изменять.
- Программы, в которых дублируется логика, сложно изменять.
- Программы, которые требуют определённого рабочего окружения, что требует изменение запущенного кода, сложно изменять.
- Программы с большим кол-ом ветвлений сложно изменять.

Т.о. мы хотим, чтобы программы было легко читать, чтобы вся логика, встерчающаяся в программе, была специфична для данного конкретного места, чтобы текущее поведение не было под угрозой и чтобы вся логика была выражена настолько просто, насколько это возможно.

Рефакторинг – это процесс, который берёт рабочую программу и увеличивает её ценность не изменением её внешнего поведения, а повышая её качество, что даёт возможность продолжать разработку быстрее.

Что я должен сказать своему менеджеру?

Как сказать своему менеджеру о рефакторинге – один из основных вопросов, который мне задавали. Если менеджер немного соображает в технике, то ввести его в тему будет не очень сложно. Если менеджер *действительно* ориентирован на качество, тогда стоит уделить внимание качеству. Здесь использование рефакторинга является хорошим способом сделать работу. Тонны исследований показывают, что обзоры являются хорошим способом устранения багов, тем самым повышая скорость разработки. Посмотри на любую книгу об обзорах, проверках или о процессе разработки программного обеспечения (ПО) для дальнейшего цитирования. Это должно показать большинству менеджеров ценность обзоров. Затем короткий шаг введения рефакторинга, как способа получения комментариев в коде после обзоров.

Конечно, многие люди скажут, что качество важно, но более важно соответствовать расписанию. В этом случае я даю более спорный совет: “Не говорите!1 Подрывает? Я так не считаю”. Разработчики ПО – профессионалы. Их работа – это разрабатывать ПО настолько быстро, насколько они в состоянии. Мой опыт показывает, что рефакторинг – это большая помощь в более быстрой разработке ПО. Если мне надо добавить новый функционал, но проект не предназначен для требуемых изменений, я нахожу, что быстрее отрефакторить и добавить новый функционал. Если мне требуется исправить баг, я должен понять, как программа работает и я нахожу рефакторинг самым быстрым способом сделать это. Менеджер хочет, чтобы я делал вещи настолько быстро, насколько я могу, а как я их буду делать – это мой бизнес. Самым быстрым способом является рефакторинг, поэтому я использую рефакторинг.