

## Глава 1

# Обучение на собственном опыте

Я не знаю никого, кто мог бы, положив руку на сердце, сказать: «Сегодня я пишу программное обеспечение лучше, чем когда-либо». Всем, кто не может этого сказать, было бы полезно обучиться более эффективным методам работы. Моя книга дает возможность ускорить прохождение этого квеста.

Обретение опыта — это форма обучения, которая лучше всего способствует закреплению знаний и умений. Но она же и самая болезненная. В попытках попробовать новые подходы мы часто терпим неудачу. Мы должны преодолевать трудности и прикладывать усилия, чтобы освоить новые методы и понять, когда и как их использовать.

К счастью, есть другой путь. Кривую обучения можно сделать более полой, опираясь на уроки, советы и приемы людей, которые уже получили знания и опыт. В книге собраны полезные идеи в области разработки ПО и управления проектами. Эти жемчужины мудрости я обрел на личном опыте и наблюдал в работе других. Ваш собственный опыт и уроки могут быть другими, и вы можете соглашаться не со всем, о чем я рассказываю. Это нормально — опыт каждого уникален. Тем не менее все описанное было весьма ценным для меня и пригодилось в моей карьере программиста.

## МОЯ ТОЧКА ЗРЕНИЯ

Позвольте начать с короткого рассказа о моем прошлом, чтобы объяснить, как я усвоил эти уроки. Свой первый курс по программированию я прослушал в 1970-м в колледже. В основе курса лежал, конечно

же, язык FORTRAN. Следующим летом я приступил к выполнению своего первого задания. Нужно было автоматизировать некоторые операции в отделе бухгалтерии в моем колледже, причем сделать это я должен был в одиночку. У меня за плечами было уже два курса по программированию, так что я чувствовал себя полноценным инженером-программистом. Я на удивление легко справился с заданием, если учесть мой небольшой опыт. Несмотря на еще два года учебы, все остальное, что я узнал о программировании, я почерпнул сам из сторонних источников и от своих коллег. В прошлом такое неофициальное начало карьеры не было чем-то необычным, поскольку разработка программного обеспечения привлекала многих людей из разных слоев общества, которые не всегда имели формальное образование в области информатики.

С самого начала я много работал с программным обеспечением: разрабатывал требования, архитектуру и дизайн пользовательского интерфейса, писал код, тестировал его, занимался управлением проектами и писал документацию, наконец, пытался совершенствовать качество и процесс разработки. Попутно я защитил докторскую диссертацию по органической химии. Уже тогда треть моей диссертации состояла из программного обеспечения, которое анализировало экспериментальные данные и моделировало химические реакции.

В начале карьеры в компании Eastman Kodak, превратившейся в огромную и успешную корпорацию, я использовал компьютеры, чтобы заниматься проектированием и анализом экспериментов. Вскоре я перешел на должность программиста и занялся созданием приложений для исследовательских лабораторий Kodak, а затем несколько лет руководил небольшой группой разработчиков. Я обнаружил, что моя научная подготовка позволяет мне использовать более систематический подход к разработке программного обеспечения.

В 1983 году я написал первую статью о программном обеспечении. С тех пор из-под моего пера вышло много статей и восемь книг на темы, связанные с программированием. Как независимый консультант и преподаватель, начиная с 1997 года я оказал услуги почти 150 компаниям и государственным учреждениям их многих сфер. Эта деятельность позволила мне увидеть, какие методы эффективно работают в программных проектах, а какие — нет.

Ко многим своим идеям, связанным с разработкой программного обеспечения и управлением проектами, я пришел на личном опыте. Одни

из них были полезными, другие — ошибочными. Некоторые идеи я почерпнул из опыта моих клиентов, в основном из проектов, потерпевших крах. Никто не вызывает консультанта, когда все идет хорошо. Я написал эту книгу, чтобы вам не приходилось медленно и мучительно проходить все те же уроки на личном опыте. Один опытный инженер-программист, прочитавший этот список уроков, сказал: «С каждым из этих уроков связан шрам (или несколько), полученный при его усвоении».

## О КНИГЕ

Эта книга содержит 60 уроков, связанных с разработкой и управлением программным обеспечением. Уроки разбиты на шесть групп, каждая из которых описывается в отдельной главе.

- Глава 2. Требования.
- Глава 3. Проектирование.
- Глава 4. Управление проектами.
- Глава 5. Культура и командная работа.
- Глава 6. Качество.
- Глава 7. Совершенствование процессов.

Глава 8 содержит заключительный общий урок, о котором следует помнить всегда. Для удобства все 60 уроков перечислены в приложении.

Я не пытался составить исчерпывающие списки уроков в названных областях. В каждой накоплено так много знаний, что никто не сможет представить более или менее полный список. Кроме того, я не рассматриваю другие важные и наиболее очевидные аспекты разработки ПО: программирование, тестирование и управление конфигурацией. Обсуждение этих процессов вы найдете в трудах других авторов, например, в таких книгах, как:

- *Programming Pearls*<sup>1</sup> Джона Бентли (Jon Bentley) (Bentley, 2000);
- *Lessons Learned in Software Testing* Джема Канера (Cem Kaner), Джеймса Баха (James Bach) и Брета Петтикорда (Bret Pettichord) (Kaner, Bach, Pettichord, 2002);

---

<sup>1</sup> Бентли Дж. Жемчужины программирования. 2-е изд. — СПб.: Питер, 2002.

- *Code Complete* Стива Макконнелла (McConnell, 2004);
- *Software Engineering at Google*<sup>1</sup> Титуса Уинтерса (Titus Winters), Тома Мэншрека (Tom Manshreck) и Хайрама Райта (Hyrum Wright) (Winters, Manshreck, Wright, 2020).

Темы и уроки в этой книге в значительной степени независимы, поэтому вы можете знакомиться с ними в любой последовательности. Каждая глава начинается с краткого обзора соответствующей области разработки программного обеспечения. Затем несколько врезок под названием «Первые шаги» предлагают вам поразмышлять о предыдущем опыте работы, прежде чем погрузиться в уроки главы. Эти врезки предлагают задуматься о проблемах, с которыми сталкиваются команды в данной области, влиянии этих проблем и возможных глубинных причинах.

Каждый урок начинается с краткого изложения основной мысли, далее следует обсуждение и приводятся рекомендации, которые команды могут применять в своей практике. Читая каждую главу, думайте о том, как описанные приемы можно использовать в вашей ситуации.



Значок с изображением книги на полях, как здесь, указывает на реальную историю из личного опыта или опыта моих клиентов или коллег. Все истории реальны, хотя имена участников изменены, чтобы сохранить конфиденциальность.



Ключ на полях указывает на основные моменты в описании каждого урока.



Некоторые уроки содержат ссылки на другие уроки. В таких случаях приводится этот значок со стрелкой.

Врезка «Следующие шаги» в конце каждой главы подскажет вам, как применить полученные сведения в вашем проекте, команде или организации. Независимо от вида проекта, над которым вы работаете, его жизненного цикла или создаваемого продукта, ищите идею, которую несет каждый урок, и думайте, как вы можете адаптировать ее, чтобы помочь проекту стать более успешным.

Попробуйте проработать материал всех врезок вместе с коллегами. В начале всех обучающих курсов, которые я вел, участники объединялись в небольшие группы и обсуждали задачи, стоявшие перед их

<sup>1</sup> Винтерс Т., Маншрек Т., Хайрам Р. Делай как в Google. Разработка программного обеспечения. — СПб.: Питер, 2021.

командами («Первые шаги»). В конце те же группы изучали варианты решения этих задач, проводили мозговой штурм, посвященный тому, как решить их с помощью материалов курса («Следующие шаги»). Мои студенты увидели, что в эти дискуссионные группы полезно включать представителей разных заинтересованных сторон, так как они высказывали свою точку зрения на функционал проекта и его продвижение. Благодаря всестороннему обсуждению можно достичь глубокого понимания текущей деятельности и творчески подойти к выбору решений.

Я надеюсь, что многие из моих уроков найдут отклик и мотивируют вас попробовать что-то новое в своих проектах. Однако вы вряд ли сможете изменить все сразу. Разные люди, команды и организации воспринимают новый опыт с разной скоростью. Заключительная глава «Что дальше?» поможет вам наметить путь, как реализовать уроки на практике. В ней предлагается установить приоритеты изменений, которые вы хотели бы внести, и разработать план действий, который поможет перейти из точки, в которой вы находитесь сегодня, в точку, куда хотели бы попасть.

## ПРИМЕЧАНИЕ О ТЕРМИНОЛОГИИ

В этой книге я использую термины «система», «продукт», «решение» и «приложение» более или менее взаимозаменяемо. В каждом случае я имею в виду просто конечный результат вашего проекта, поэтому, пожалуйста, не ищите особого смысла в этих терминах. Независимо от того, работаете ли вы с корпоративной или государственной информацией, сайтами, коммерческими программными продуктами или аппаратными устройствами со встроенным ПО, уроки и соответствующие практики найдут свое применение.

## ВАШИ ВОЗМОЖНОСТИ

Если вы действительно не относитесь к той редкой категории практиков, которые уже создают программное обеспечение лучше, чем кто-либо другой, то у вас всегда найдется что улучшить. Мы все — и отдельные разработчики, и группы, и организации — должны постоянно расширять свои возможности. Мы все хотим иметь как можно меньше шрамов.

Младший разработчик по имени Захари Минотт (Zachary Minott, 2020) поделился своей методикой, которая помогла ему превзойти более опытных разработчиков. Минотт описал этический подход, согласно которому он признает отсутствие у себя тех или иных знаний и обязуется изучать и применять новые знания на практике. Он сказал: «Если у меня и есть какая-то сверхспособность, то это способность быстро учиться и сразу же применять новые знания в своей работе». Минотт открыл важнейший механизм, позволяющий ему постоянно совершенствоваться в своей сфере.

---

Нам всем нужно постоянно совершенствоваться,  
мы все хотим иметь как можно меньше шрамов.

---

Возможно, вы решите пройти курс обучения, чтобы освоить новый навык или усовершенствовать привычные методы работы. Пока вы учитесь, невыполненная работа продолжает накапливаться. Из-за спешки велик соблазн проигнорировать новые знания и продолжать работать по старинке. Вы можете чувствовать себя вполне комфортно, поскольку текущих знаний и умений пока достаточно. Но это не путь к совершенствованию.

В каждом проекте я выделил две области, в которых нужно совершенствоваться. Я тратил какую-то часть времени, чтобы изучить новые темы и попытаться применить новые знания. Удача не всегда сопутствовала мне, но мой подход позволил постепенно накопить навыки, сослужившие мне хорошую службу.

Я призываю вас поступить так же. Не просто читайте книгу — делайте следующий шаг. Решите, как вы и ваши коллеги сможете применять описанные здесь методы и что, по вашему мнению, они дадут вам. Составьте список практик, о которых хотели бы узнать больше, а затем применяйте их. Так постепенно вы достигнете прогресса.