## Парное программирование

В XP используется достаточно оригинальная практика **парного программирования**, при которой два человека сидят за одним компьютером и пишут код вместе. Такой режим работы может показаться непривычным для людей, привыкших рассматривать программирование как занятие для одиночек. Тем не менее он может стать эффективным инструментом для очень быстрого создания высококачественного кода, потому что многие люди, занимающиеся парным программированием, сообщают, что пары выполняют больший объем работы, чем два программиста по отдельности.

**Парное программирование способствует концентрации внимания, упрощает поиск ошибок, стимулирует коллективные обсуждения и помогает участникам изучить все части кодовой базы.**

**Райан и Ана помогают друг другу сохранить концентрацию внимания.**

**Если Райан зайдет в тупик, Ана может взять дело в свои руки и направить его дальше — и наоборот.**

**Они постоянно обсуждают проблемы, над которыми работают в данный момент, и совместно придумывают решения.**

**Вряд ли вам захочется работать спустя рукава, когда кто-то работает рядом.**

**Состав пар постоянно меняется, поэтому все участники получают возможность поработать над каждой частью системы.**

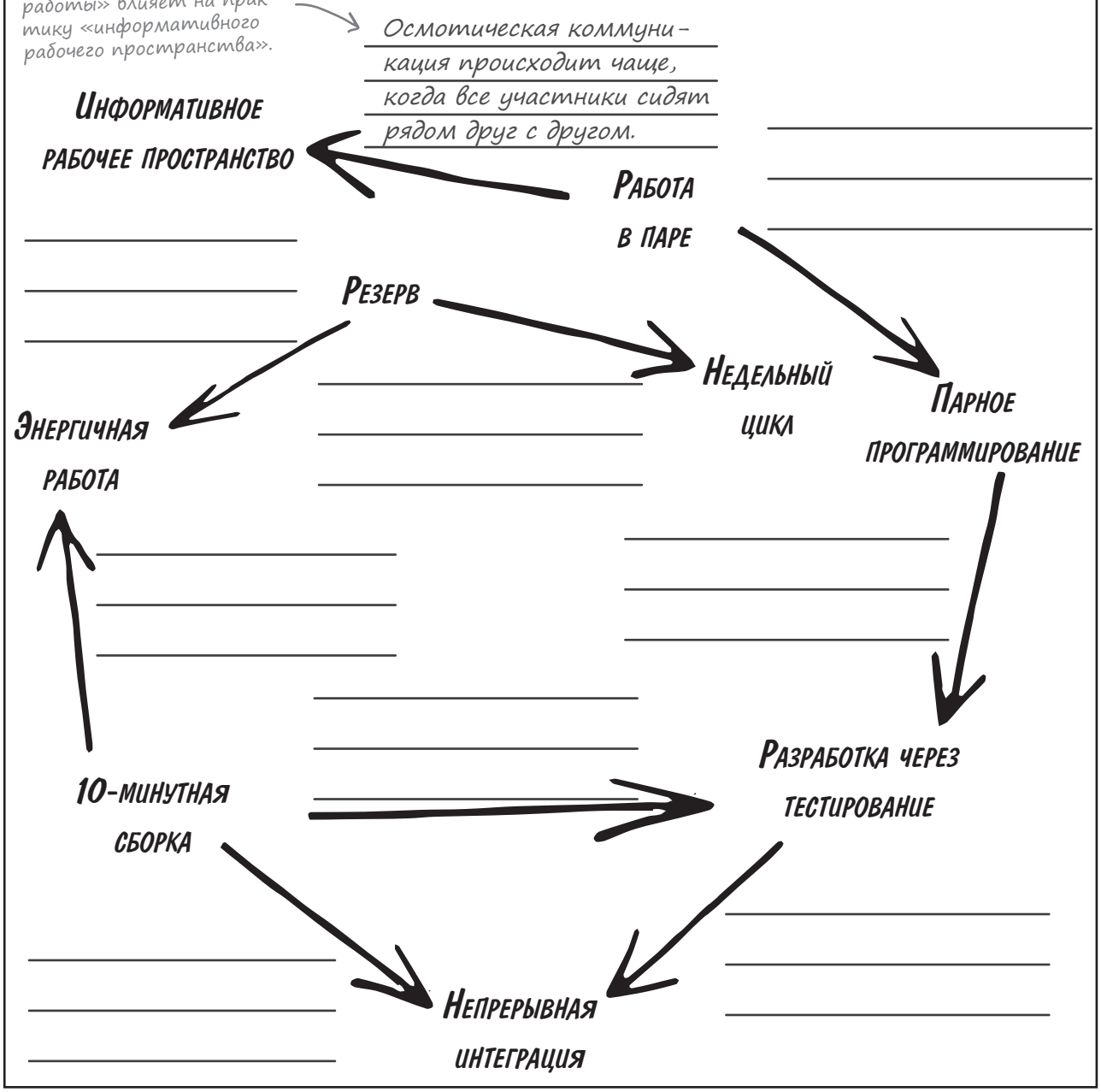
**Вы не всегда знаете, насколько хорошо поняли суть идеи, пока не попытаетесь ее кому-нибудь объяснить.**

**На каждое изменение смотрят две пары глаз, и они выхватывают множество мелких ошибок, которые могли бы породить массу проблем в будущем.**

## Возьми в руку карандаш

Чтобы помочь вам взяться за дело, мы заполнили этот пропуск — он показывает, как практика «командной работы» влияет на практику «информативного рабочего пространства».

Практики XP полезны сами по себе, но они становятся особенно эффективными при комбинированном применении. Внизу мы изобразили несколько практик XP и соединили их стрелками. Рядом с каждой стрелкой — пустые строки. В каждом наборе пустых строк запишите один способ взаимодействия практики, ОТ которой выходит стрелка, к практике, К которой она ведет — способ, который бы подкреплял и поддерживал последнюю.



## Возьми в руку карандаш
### Решение

У этого упражнения есть много разных решений, что объясняется большим количеством способов, которыми практики XP могут подкреплять и поддерживать друг друга. Мы записали некоторые способы, которые, на наш взгляд, являются важными. А ваши ответы похожи на наши?

**Практики XP совместно формируют экосистему, которая ведет к более качественному, гибкому и простому в сопровождении коду.**

## Часто Задаваемые Вопросы

**В: Практики «нахождения в одном помещении» и «парного программирования» требуют, чтобы все находились в одном офисе. Означает ли это, что глобальные или распределенные команды не могут использовать XP?**

О: Многие глобальные и распределенные команды используют XP. Команды XP знают, что при совместном размещении у них больше времени для общения, их работа реже прерывается телефонными звонками и они могут совместно использовать информативное рабочее пространство. В распределенной команде, где все работают в разных офисах и общаются по электронной почте или телефону, это невозможно. Однако важная часть майндсета XP заключается в том, что каждая практика направлена на повышение эффективности команды. Если какие-то практики просто невозможны, команда работает с тем, что возможно.

**В: Разве это не означает, что они не практикуют «чистую» методологию XP?**

О: Действительно, эффективная команда знает, что «***чистой методологии» XP*** не существует. Команды XP всегда ищут пути к совершенствованию. Нет «идеального» состояния, которого они пытались бы достичь; команда просто пытается лучше делать то, что она делает. При бездумном следовании практикам среда быстро теряет весь запас энергии. Кроме того, все разговоры о недостаточной «чистоте» неуважительны. **Не доставайте людей по поводу «чистоты» XP** — у них создается ощущение, что вы судите их и их работу. От этого ничего реально не изменится, люди только будут ненавидеть вас и XP.

**В: Означает ли это, что я могу просто отбрасывать все практики, которые мне не нравятся?**

О: Нет, не можете. Практики XP тщательно проектировались с расчетом на работу друг с другом, и при совместном использовании они помогают команде интегрировать ценности XP в майндсет. Например, команды начинают понимать такую ценность XP, как общение, когда они размещаются вместе и формируют информативное рабочее пространство. Когда команды решают выкинуть ту или иную практику, обычно это происходит из-за того, что ее майндсет несовместим с одной или несколькими ценностями, и практика кажется неудобной. Лучшее, что можно сделать в подобном случае — **по-настоящему попытаться применить практику**. Часто это помогает команде сменить майндсет, что в свою очередь помогает всем участникам лучше работать и строить более качественные продукты.

**В: Разве непрерывная интеграция не означает простого создания сервера сборки?**

О: Нет. Сервер сборки — программа, которая периодически загружает новейший код из системы контроля версий, запускает автоматизированную сборку и уведомляет команду о возможных ошибках. На самом деле это хорошая мысль — почти в каждой команде гибкой разработки используется сервер сборки. Тем не менее сервер сборки — не то же самое, что непрерывная интеграция. Непрерывная интеграция означает, что каждый участник команды активно (и непрерывно!) интегрирует новейший код, написанный его коллегами, в свою рабочую папку. Путаница с терминами возникает из-за того, что сервер непрерывно «интегрирует» код из системы контроля версий в собственный репозиторий и уведомляет команду о каждом закреплении кода, который не компилируется или вызывает сбои при тестировании. Тем не менее это не заменяет необходимости своевременного обновления рабочей папки на компьютере каждого участника.

**В: Не понимаю. Если у нас имеется сервер построения, который непрерывно интегрирует код, разве это не сокращает объем работы на каждого участника?**

О: Действительно, если каждый участник команды будет непрерывно интегрировать новейший код из системы контроля версий в свою рабочую папку, работы будет больше, чем при создании сервера сборки. Но если участники будут полагаться исключительно на уведомления электронной почты от сервера сборки, которые будут сообщать о рассинхронизации, это часто кончается плохо. Например, может выясниться, что при закреплении кода, нарушающего сборку, все начинают злиться на вас, поэтому вы начинаете закреплять свой код намного реже обычного. Или команда просто привыкнет к сообщениям «нарушения сборки» от сервера сборки и начнет игнорировать их, накапливая уведомления в папках. С другой стороны, если каждый человек знает, что через каждые несколько часов он обязан прерывать текущую работу и интегрировать код из системы контроля версий в рабочие папки, ошибки сборки происходят редко — а когда это произойдет, команда быстро заметит проблему и будет совместно работать над ее исправлением.

**В: Значит, непрерывная интеграция — всего лишь вопрос достаточного уровня дисциплины в команде?**

О: Не совсем. Когда команда действительно хорошо освоила такие практики, как непрерывная интеграция, 10-минутные сборки или разработка через тестирование, снаружи все выглядит так, словно команда идеально дисциплинирована, но на самом деле дисциплина здесь ни при чем. Команда делает все это, ***потому что смысл этих действий понятен всем***. Все участники команды просто сознают, что работа замедлится, если, допустим, они не приложат усилий к ускорению сборки или построению модульных тестов перед написанием кода. Их не надо подгонять, на них не надо орать или отчитывать — потому что им не приходит в голову ***не*** делать все это.

**В: Я работаю в команде контроля качества. Разработка через тестирование означает, что тестировщики должны написать мои модульные тесты до того, как я напишу код?**

О: Нет. Вы сначала пишете свои модульные тесты, а затем код, обеспечивающий их прохождение. Дело в том, что модульные тесты должны быть написаны тем же человеком, который пишет код, — это объясняется тем, что в процессе написании тестов вы много узнаете о проблеме, над которой работаете, а это способствует повышению качества кода.

Не нравится мне все это. Такое впечатление, что парное программирование — напрасная трата времени. Если два человека работают вместе, разве они не начинают **строить код вдвое медленнее**?

## Парное программирование — чрезвычайно эффективный подход к программированию.

Работа в парах помогает сосредоточиться и устраняет многие отвлекающие факторы (например, запуск браузера или проверку электронной почты). Лишняя пара глаз помогает обнаружить ошибку на ранней стадии вместо того, чтобы тратить время на их отслеживание в будущем. Но еще важнее то, что это означает **постоянное сотрудничество с участниками команды**. Программирование — интеллектуальная деятельность: в процессе написания кода разработчику постоянно приходится решать непростые задачи и головоломки, одну за другой. Обсуждение этих задач и головоломок с коллегами — действительно эффективный способ их решения.

*Вот почему люди, изначально настроенные против парного программирования, нередко изменяют свое отношение после того, как они опробуют эту практику в течение нескольких недель.*

**Оправданно ли использование слова «иррациональный»? На наш взгляд — да. Парное программирование — простой, понятный и (давайте признаем) непримечательный рабочий процесс, многие люди делают это каждый день. Слишком сильная и негативная реакция на нечто столь повседневное иррациональна по определению.**

Хорошо, допустим. Но... Вы уверены? Честно говоря, **вы меня не убедили**. Мне все равно кажется, что с парным программированием что-то не так.

## «Что-то не так» кажется тогда, когда практика вступает в конфликт с вашим майндсетом.

Думаете, что вы лучший программист из всех окружающих? Относитесь к программированию как к одиночному занятию? В таком случае вы будете ощущать иррациональную неприязнь к парному программированию. Считаете себя «звездой», окруженной сплошными идиотами, которые и пары строк толкового кода не напишут? Тогда вы будете испытывать *чрезвычайно сильную иррациональную враждебность* к парному программированию. Ключевое слово здесь — «**иррациональную**»: да, вы можете придумать причины и обоснования для своего отношения к парному программированию, но в самом центре лежит *ощущение того, что эта практика не подходит* для вас и для вашей команды. Собственно, само понятие «иррациональный» определяется как «решение, принятое на основании эмоций, а не доводов разума».

Но потом многие хорошие программисты в реальных проектах обнаруживают, что их «младшие» коллеги не только не отстают от них (кто бы подумал!). Более того, когда они по-настоящему пытаются применить парное программирование — не просто выполняют весь положенный ритуал, но и действительно стараются заставить его работать, — программирование идет намного быстрее. Наконец, более «медленные» коллеги начинают перенимать многие навыки и приемы, и вся команда совершенствуется вместе.

Простите, так и не убедили. Парное программирование — это **плохо**, и ничто из того, что вы скажете, моего отношения не изменит. Означает ли это, что методология XP **не подходит мне и моей команде**?

### Значит, вы и ваша команда не разделяют ценности команд XP.

Ценности команд XP — целенаправленность, уважение, смелость и обратная связь. Если вы действительно цените все это, парное программирование выглядит абсолютно логично. Если вы цените целенаправленность, вы понимаете, как парное программирование помогает вам и вашим коллегам оставаться на пути к цели. Если вы цените уважение, идея парной работы с коллегами не вызовет у вас иррационального протеста, потому что вы уважаете их и их способности. Если вы цените смелость, вы сможете преодолеть собственное неудобство и опробуете то, что может помочь команде. А если вы цените обратную связь, лишняя пара глаз, рассматривающая каждую строку написанного кода, не помешает.

С другой стороны, если несколько последних предложений кажутся вам банальными, чрезмерно упрощенными, идеалистичными или даже глупыми, это значит, что **вы не разделяете те же ценности**, что и эффективные команды XP.

**Если вы пытаетесь освоить практику, которая не соответствует вашему майндсету или культуре команды, обычно эта практика «не приживается», и все кончается механическим повторением положенных действий.**

И что? Что случится, если я не разделяю ценности XP?

### Переход на новые практики требует работы, и общие ценности стимулируют всех участников выполнить эту работу.

Если команды пытаются освоить методологию, не соответствующую культуре команды, обычно это заканчивается плохо. Команда пытается внедрить новые практики, какие-то из них даже временно работают. Но в конечном итоге все выглядит так, словно вы и ваша команда просто пытаетесь механически повторить эти практики. Они воспринимаются как бремя, которое не приносит никакой пользы, и через несколько недель или месяцев команда возвращается к прежнему состоянию дел.

Но это вовсе не означает, что ситуация безнадежна! Это означает лишь то, что вам и вашей команде *стоило поговорить о ценностях* ***перед*** *тем, как опробовать эти практики.* Если вы с самого начала начнете работать над проблемами культуры, вам будет намного проще перейти на XP (или любую другую методологию!), а вероятность того, что новая методология приживется, заметно повышается.

Раз уж речь зашла об усовершенствовании функционирования команды, вернемся к Райану и Дне. →