



ГЛАВА СЕДЬМАЯ
**КАК НАПИСАТЬ
КАЧЕСТВЕННЫЕ
СПЕЦИФИКАЦИИ**

[Почитать описание, рецензии и купить на сайте МИФа](#)

Однажды я поспорил с программистом, который утверждал, что спецификации нам не нужны. Он поднял на смех толстенный документ, взятый мною у босса, а заодно и меня. Коллега рассуждал так: если задача настолько сложна, что объяснение занимает 50 страниц, — за нее вообще не стоит браться. Во всей бумажной работе он видел признак нарушенной коммуникации и координации в команде и недоверия к сотрудникам. «Зачем столько бюрократии?» — спросил он, подразумевая, что доскональное планирование лишь вредит делу.

Я спросил, готов ли он утверждать то же самое в отношении проектной документации многоэтажного дома, в котором живет, или трехуровневой дорожной развязки, по которой добирается на работу. По-видимому, этот вопрос он слышал не впервые и улыбнулся. Он сказал, что в перечисленных вещах идет речь о законах физики и строительном материале, от чего сфера ПО далека, и стал отстаивать методы с минимальными спецификациями, такие как экстремальное программирование. Мы быстро сошлись по двум пунктам. Во-первых, проектирование ПО по сравнению с традиционным более гибкое, менять его намного легче и оно редко ставит на кон человеческую жизнь. Однако мы также признали, что, поскольку перед нами стоят сложные задачи, есть бюджет, сроки, а команда зависит от наших решений, мы должны убедиться, что все будет сделано как надо. Во-вторых, для нашего проекта нужно нечто, подходящее именно для такого типа деятельности и именно для таких людей, как мы. Некий письменный документ, который решает реальные проблемы команды, ускоряет работу и повышает вероятность качественного результата (причем документ придется периодически обновлять, никого не расстраивая). Если такой документ появится, то независимо от названия и формата, мой собеседник охотно им воспользуется. Мы пересмотрели процесс спецификаций и превратили его в то, что должно сработать в нашей немногочисленной команде. Я вернулся к боссу, вкратце рассказал о нашей беседе, и мы нашли компромисс. Так мы попрощались с громоздким шаблоном спецификаций размером с налоговый кодекс.

Главный урок этой истории в том, что ни в одной сфере нет единственно правильного способа написать спецификации или документировать работу. Спецификации должны соответствовать потребностям проекта и людей, которые их пишут и читают. И точно так же, как сайты или программные продукты нуждаются в проектировании, чтобы найти оптимальные подходы к потребителю, — спецификации требуют анализа и итераций.

Однако многие опытные специалисты, которых я знаю, попались в ловушку и убеждены, что есть только один способ правильно написать спецификации (или как бы вы ни называли этот документ) — тот, которым они пользовались прошлый раз. Иногда нить тянется к самому первому проекту, над которым они работали. Они предполагают, что, поскольку эти проекты не обернулись полной катастрофой, тот способ, которым они составили (или хотели составить) спецификации, позитивно отразился на результате — хотя это утверждение абсолютно голословное (к примеру, проект мог оказаться успешным вопреки неадекватным спецификациям). Хуже того, если не возникал вопрос, как и почему составлены спецификации, никто в команде не сможет понять, насколько эффективен этот процесс или насколько он способствует или не способствует плодотворной работе команды. (Аналогично отсутствие правильных вопросов о качественном коде мешает понять, насколько хорош или плох код имеющийся.)

Моя цель в этой главе — объяснить основные идеи. Во-первых, спецификации проекта выполняют три задачи: гарантируют создание нужного продукта; дают контрольную точку, которая знаменует завершение этапа планирования; а также позволяют провести тщательный обзор и получить обратную связь от людей относительно направления проекта. Эти три задачи крайне важны, и вряд ли что-то еще, кроме письменных спецификаций, выполняет их все одновременно. Одной этой причины достаточно, чтобы я целиком и полностью отстаивал необходимость подготовки документа. Во-вторых, большинство жалоб по поводу спецификаций легко устранить, если, конечно, авторы понимают частые ошибки их написания и знают, для каких целей они пишутся.

ЧТО МОГУТ И ЧЕГО НЕ МОГУТ СПЕЦИФИКАЦИИ

Спецификации, как и видение, представляют собой форму общения: передают важную информацию простым и понятным способом. Но при неграмотном исполнении их сложно читать и утомительно воплощать. Они выбивают из колеи всех, кто с ними соприкасается. Зачастую паршивые спецификации получаются у команд, которые питают пристрастие к многостраничным документам. Чаще всего слабые или неудачные попытки рождаются из непонимания потенциала спецификаций.

Перечислим список важных задач, которые спецификации выполняют для проекта:

- эффективно описывают функционал продукта;
- помогают проектировщикам прояснить решения, призывая к конкретике;
- создают условия для проверки, вопросов и обсуждений подробных планов до начала реализации;
- передают информацию от одного многим;
- создают ориентир для всей команды относительно плана работы (если составить спецификации на этапе проектирования, можно вносить в них изменения и обновления, отражая этапы развития идеи) [1];
- отображают график этапов работы, чтобы команда сосредоточилась;
- страхуют на случай, если автора (или авторов) собьет автобус [2];
- ускоряют, улучшают и увеличивают регулярность грамотных обсуждений;
- упрощают тимлидам возможность дать обратную связь и задать стандарт качества;
- дают команде (и автору) объективный взгляд и уверенность.

Что спецификации не могут и не должны делать:

- исключать какие-либо обсуждения между членами команды;
- доказывать команде, какой автор умный;

- доказывать, как важен тот или иной функционал (и почему его нельзя исключить);
- навязывать людям философский подход;
- дать автору возможность похвастаться своими навыками в Visio и UML.

Тимлиды должны составить такой список для команды. Каждый, кому предстоит прочитать или написать спецификации, обязан изучить его и заранее высказаться. Возможно, некоторые пункты списка посеют в команде сомнения. Можно провести краткое обсуждение — максимум полчаса. Даже переброска парой слов даст представление о том, что это даст, и команда сможет предложить более эффективный способ. Если есть шаблон по спецификациям для всей команды, он должен соответствовать принятым в результате обсуждения критериям.

КАКИЕ СПЕЦИФИКАЦИИ НУЖНЫ

Каждая методология для разработки ПО или для проектного менеджмента определяет спецификации по-разному. В этом нет ничего плохого. Есть четыре базовых типа информации, которая попадает в спецификации. Самый простой способ обсудить их — представить, что они предназначены для четырех разных документов. Главное, чтобы нужная информация исходила от людей, которые в ней разбираются, и чтобы формат для чтения был удобным. В небольших командах разные типы спецификаций часто объединяют. В крупных командах их лучше разделить (но связь должна оставаться) и даже поручить разным сотрудникам.

- **Требования.** Спецификации отмечают все задачи и обязанности, которые следует выполнить. Иными словами, вы собираете все требования по работе и задаете ориентир. В идеале это список условий, от которых зависит успех проекта, описывающих итоговый результат, но не объясняющих, как его достичь. В любом случае требования следует определить до начала проектирования (хотя их можно изменить

и обновить позже), и они должны опираться на видение. Для ясности их следует дополнить функциональными спецификациями (то есть понять, насколько данный план соответствует требованиям).

- **Функционал.** Спецификации описывают поведение продукта по пользовательскому сценарию. Они вытекают из проектирования и иллюстрируют функциональность программного обеспечения через пользовательский интерфейс (если таковой имеется), а также подробно указывают, как все должно работать с нетехнической точки зрения. Они также демонстрируют, как изменится клиентский опыт по завершении работы, и содержат перечень конкретных проектировочных задач. Этот список отличается от требований тем, что определяет проектное решение, удовлетворяющее требованиям, включая пользовательский интерфейс и другие нетривиальные элементы. При грамотном исполнении функциональные спецификации предельно просты — к примеру, вполне достаточно скриншотов с объяснениями.
- **Технические спецификации.** Эти документы подробно описывают подход инжиниринга, необходимый для реализации функциональных спецификаций. Детализация должна быть достаточной для описания самых сложных или многократно используемых компонентов, которые могут задействовать другие программисты, и для обоснования рабочих компонентов, необходимых для функциональной спецификации. Иногда глубина и техническая сторона последних исключает потребность в отдельных технических спецификациях.
- **Список рабочих элементов.** Примерно то же самое, что WBS. Список рабочих элементов описывает каждое задание по программированию, необходимое для реализации функциональных спецификаций. Его следует разбить на детали, которые отделяют элементы разного уровня важности, предусматривая сроки (в днях), с учетом ограничений по их продолжительности, допустим, день или полдня (это решают программисты). Составление списка — целиком и полностью привилегия программистов, и главный программист, а также, возможно,

МП должны проверять и подчищать эти списки. (Строго говоря, списки рабочих элементов — это не спецификации, а план их реализации. Однако они настолько важны и связаны со спецификациями, что я не смог найти более подходящего места, чтобы представить их.)

- **Критерии тестирования и критерии завершения этапа.** После составления функциональных спецификаций нужно подготовить критерии тестирования. В них войдут приоритетные тестовые сценарии для новой функциональности, а также критерии качества кода в этих сценариях для выполнения конкретных задач текущего этапа (критерии завершения см. в главе 15).

Позвольте привести пример объединения разных типов спецификаций. Каждый раз, когда я работал с большой командой, мы писали и функциональные, и технические спецификации. Мы составляли списки требований, опираясь на видение, проверяли их с командой и клиентом, а затем размещали в начале функциональных спецификаций. Список рабочих элементов составляли программисты (обычно в простой электронной таблице), затем его копировали или привязывали к функциональным спецификациям. Проще говоря, наши основные спецификации включали множество других сведений, которые мы сейчас перечислили.

Самый простой подход к этим четырем типам спецификаций — приблизительный хронологический порядок: требования, функционал, технические параметры и рабочие элементы. Как во многих задачах, каждый тип информации становится основой для следующего. Чем больше команда и чем сложнее проект, тем четче должна быть разница между этими типами спецификаций.

КТО ОТВЕЧАЕТ ЗА СПЕЦИФИКАЦИИ

В большой команде МП или проектировщики занимаются функциональными спецификациями; разработчики берут на себя технические спецификации. Их нужно написать так, чтобы любой прочитавший поверил:

авторы действительно знают друг друга и часто общаются. Зачастую технические спецификации намного короче (и менее содержательны для читателя), потому что их аудитория меньше и программисты в принципе не любят писать то, что не компилируется. Однако технические спецификации, поддерживающие проектные идеи из функциональных спецификаций, должны сочетаться друг с другом.

Бизнес-аналитики, клиенты и МП зачастую пишут требования. Все зависит от того, кому поручено составить требования и какова природа проектной команды (небольшая наемная или большая штатная). Составление списка рабочих элементов ложится на плечи того, кто руководит командой программистов. В крупных организациях это, как правило, ведущий программист.

Небольшие команды обычно менее структурированы. Не всегда жестко предусмотрено, кто что делает и даже какие документы нужно написать. МП или ведущий программист пишут один документ — сборную солянку всех четырех типов информации, причем переходят с одного типа на другой в зависимости от потребностей команды. Это нормально, если люди получают то, что им нужно и когда им это нужно.

СПЕЦИФИКАЦИИ – ЭТО НЕ ПРОЕКТИРОВАНИЕ

Две предыдущие главы касались проектирования — как работать с идеями и превратить их в план. Крайне важно разделить проектирование и планирование от написания спецификаций для них. Спецификации должны, насколько возможно, наиболее полно отражать план и принятые решения, а не составлять этот план на ходу. Чем меньше разница между этими двумя задачами, тем сложнее выполнить их. Каждый из этих процессов сам по себе достаточно труден, и пытаясь одолеть оба процесса одновременно, вы усиливаете вероятность завала.

Авторы спецификаций должны понимать разницу подходов к ним и к проектированию. Работа над спецификациями — это не исследование и создание, а формулировка и объяснение, голос того, кто объясняет, а не творит. При написании спецификаций (или любого другого документа) важно помнить, что метод, который позволил нам что-то выяснить, не всегда самый лучший для растолкования этого явления кому-то другому.

ОПИСАНИЕ КОНЕЧНОГО ПРОЕКТНОГО РЕШЕНИЯ И КАК ЕГО ПОСТРОИТЬ

Функциональные и технические спецификации можно сочетать в одном документе, но их следует разбить на два раздела. В худших спецификациях, какие я читал, этого не сделали. Автор, несмотря на весь свой интеллект и талант, пытался описать идею и одновременно объяснить, как ее реализовать. С первых же строк стало очевидно, сколько времени потрачено зря [3]. Документ содержал огромные скрупулезные диаграммы, отражающие взаимосвязь между объектами и компонентами, а также подробную информацию о том, как ими будут пользоваться клиенты. В итоге получилась эстетически прекрасная и идеально отшлифованная катастрофа. Спецификации выглядели внушительно, но спустя пять минут чтения и тщетных попыток понять смысл у меня появилось желание придушить их автора (и, по-видимому, его команда отреагировала так же). Он несколько раз пытался ознакомить со своим детищем сотрудников, но столкнулся с непониманием и даже агрессией.

Хотя автор признал, что потерял цель и что спецификации сами по себе не так важны, но все же настаивал на своем подходе. Он заявил, что программистам нужен ориентир по ожидаемым сценариям поведения, а также высокоуровневые детали относительно взаимосвязи между объектами, и вполне разумно объединить их. На мой взгляд, даже если специалисту требуется информация обоих типов, нет причин вываливать

ее одновременно. Намного проще писать и читать что-то одно, а не затрагивать множество тем. Хорошие спецификации зачастую содержат несколько уровней проектирования: во-первых, клиентский опыт на языке пользователя, во-вторых, высокоуровневый обзор базовых объектов и архитектуры и, в-третьих, сложные и подробные задачи проектирования.

ХОРОШИЕ СПЕЦИФИКАЦИИ УПРОЩАЮТ ПРОЦЕСС

Самое сложное для технарей — правильно выбрать, какие детали исключить и в какое время это сделать. Посетив множество занятий по логике и математике, я понял, что лучшие преподаватели знают, когда пропустить несущественный, хотя и важный материал и как вернуться к нему, когда студент (или читатель) будет готов к этому. Правильные спецификации опираются на этот же принцип. Главное выходит на первый план. Людям необходимо понимать предстоящую работу. После чтения спецификаций процесс создания продукта должен стать для них более осмысленным, а вопросы, которые они смогут задать менеджеру и другим членам команды, — более конкретными. Стремитесь к этому. Разумеется, всем не угодишь, но старайтесь охватить основных участников проекта.

Конечно, в сложных объектных моделях и детализированных интерфейсах не избежать трудностей. Некоторые моменты требуют дополнительных пояснений и больше времени на понимание. Не волнуйтесь: так и должно быть. Зачастую трудности — это сомнительные отговорки, за которыми прячутся некачественные спецификации или посредственный подход. Смысл спецификаций — описать все элементы работы так, чтобы облегчить людям их понимание. В худшем случае сотрудникам понадобится больше времени, чтобы разобраться в спецификации, чем спроектировать сам продукт. Однако, как и с любым письменным текстом, критерии крайне субъективны. Ясность, понятность или сложность

спецификаций зависит от восприятия человека и его степени доверия лидерам команды.

Мне бы хотелось дать вам несколько советов по грамотному описанию задач и обозначить, чего следует избегать.

- **Позаимствуйте удачные описания элементов работы из других спецификаций (даже чужих).** Грамотно используйте гипертекст и поищите полезные обзоры в интернете, если необходимо (кстати, тимлиды должны поощрять такие поиски). Нет нужды изобретать колесо и заново описывать все на свете.
- **Избегайте жаргона и непонятных выражений.** Не используйте их, если не уверены, что они помогут читателю. Или, выражаясь более абстрактно, сократите объем вероятных умышленных обфускаций концептуального характера через максимально сжатое и емкое преобразование макроконцепций в однозначные выражения информации и упразднение избыточных лингвальных накоплений.
- **Держитесь за старые спецификации.** Они будут хорошими ориентирами, если вы не знаете, как лучше представить концепцию или составить диаграмму.
- **Пишите спецификации для конкретных читателей.** Даже в команде из 10 человек есть 4–5, которые больше всего нуждаются в спецификациях. Добавьте к этой группе умного человека, которого вы знаете, но не состоящего в команде и не знакомого с конкретной технологией, которой вы пользуетесь. Как бы вы объяснили ему простую концепцию?
- **Не увлекайтесь Visio и блок-схемами.** Сохраняйте «платонические отношения» со всеми инструментами. Обычно диаграммы интересны только их автору и не настолько полезны для проекта, как ему кажется. Иногда один абзац понятного текста или схематичная, нарисованная от руки иллюстрация лучше, чем UML-диаграмма из 500 элементов. Инструменты и диаграммы могут быть эффективны, просто не теряйте объективности.

- **Это справочное руководство или спецификации?** Последние не должны представлять собой полный справочник по программному интерфейсу приложения или описывать каждый случай либо возможный сценарий. Абсолютно разумно объяснить 10–15 самых распространенных или важных сценариев и подготовить отдельный документ, где вы подробно перечислите все остальное.
- **Прежде чем углубляться, используйте псевдокод или обычный разговорный язык, чтобы описать сложные алгоритмы на высоком уровне.** Как мы уже отмечали, многоуровневый подход к объяснению может быть самым быстрым способом донести информацию до читателей, даже до самых умных. Как минимум хорошее резюме и обзор принесут большую пользу.

И еще одна хитрость, которая всегда мне помогала: если кто-то запутался в черновике ваших спецификаций (вы узнаете это, если подкупите кого-нибудь их почитать), выделите пять минут и объясните то, что непонятно. Когда человек поймет, о чем речь, спросите, можно ли иначе отразить этот момент в спецификациях. Не факт, что вы получите хороший совет, но ваше понимание улучшится хотя бы потому, что вы расширите кругозор. Умение посмотреть на конкретную концепцию с другой точки зрения повышает вероятность найти более удачное решение. Как автор спецификаций, помните, что хорошую обратную связь легче получить, если попросить об этом, а не ждать, пока человек сам догадается высказаться.

УБЕДИТЕСЬ, ЧТО НУЖНОЕ ПРОИЗОЙДЕТ

Спецификации определяют ряд замыслов. Они утверждают: «Если все пойдет так, как мы ожидаем, по окончании работы получится то, что указано в этом документе», — то есть все сценарии поведения и функционал (или их значительный процент), отраженные в спецификациях, должны воплотиться в программе по завершении работы. Хотя вполне возможно, что вы напишете спецификации, а уже на следующий день

ситуация изменится. Тогда документ следует обновить: он должен отражать новую ситуацию и новые стремления.

На уровне инжиниринга цель спецификаций — сообщить об этих стремлениях всем заинтересованным. Необходимо иметь достаточно подробностей по ожидаемому сценарию поведения проекта, чтобы написать тестовые сценарии и прогнозы. Отделы маркетинга, подготовки документации и другие, задействованные в проекте, зададут свои вопросы о том, каким будет конечный продукт. Ответы помогут им выполнить свои задачи. Технической поддержке и менеджерам по работе с клиентами нужно понять, как все работает, чтобы запланировать эту самую техподдержку и обеспечить ее.

Один из лучших вопросов, который можно задать людям после того, как они прочитали спецификации: «У вас есть все необходимое, чтобы выполнить ваши задачи максимально эффективно?» Нацелившись на читателей, вы заинтересуете их. Они станут активнее использовать спецификации для реальной работы.

КТО, КОГДА И КАК

Как и в формулировке видения, крайне важно, чтобы у спецификаций был один автор. Все участники проекта должны внести в них свой вклад — комментарии, дополнительный контент, но отфильтровать все это и выстроить упорядоченный документ должен один сотрудник. Причина проста: если хотите, чтобы спецификации нормально читались, нельзя поручать формирование их частей разным людям.

Самые вероятные кандидаты на составление спецификаций — менеджер проекта, проектировщик или ведущий программист. Межфункциональный, по сути, документ должен писать тот, кто принимает решения на этом уровне. Функциональные и технические спецификации обязаны соответствовать списку рабочих элементов, который составили программисты. Если разработчики и проектировщики работали вместе

на этапе проектирования, подобное соответствие станет естественным продолжением сотрудничества и приятной возможностью коллективно спланировать работу, а не началом длинного процесса дебатов и разочарований.

По этой причине, а также по многим другим работа над спецификациями должна начинаться уже на этапе проектирования. Параллельно с подготовкой прототипов и исследованием идей намечаются предварительные решения, и можно набросать примерный вариант спецификаций (первые черновые варианты). Их допустимо не разглашать какое-то время, пока не появится достаточное количество описаний, которые пригодятся больше чем одному человеку. В ходе совместных обсуждений постепенно растет понимание оптимального проектного решения. Эти дискуссии должны быть отражены в спецификациях. Когда в процессе проектирования останутся только две альтернативы, работа над спецификациями уже должна идти полным ходом. При всего двух альтернативах спецификации должны охватить все элементы и разработку, необходимые для обеих (например, поисковая система, которая нужна для обоих дизайнов), а также высокоуровневый список остальных важных решений и областей их потенциального применения.

ДЛЯ ОДНОГО ИЛИ ДЛЯ МНОГИХ

Для большинства МП спецификации — удобное место для хранения информации, полезной только для них. Обычно от разных людей накапливается столько вопросов, что единый документ спецификаций становится, на первый взгляд, самым простым местом для их отслеживания. К сожалению, для всех, кроме МП, это мусор. Спецификации не должны превращаться в рабочий дневник автора (хотя справедливости ради следует отметить, что рабочий дневник помогает найти достойные идеи многим ученым и инженерам). Чем больше команда и чем специализированнее задачи, тем серьезнее встанет проблема.

Однако одна из важных функций спецификаций — помочь МП. Поскольку ему нужно организовать всю работу и руководить процессом, именно он будет вносить изменения и читать этот документ чаще, чем кто-либо другой, чтобы отслеживать конкретные, детализированные фрагменты информации о проекте.

Итак, если вы пишете спецификации, подумайте, какие подробности нужны только менеджеру по проекту, а какие представляют ценность для всей команды. Самый простой способ — отделить объяснение сценариев поведения и функций в спецификациях от трудностей или вопросов, связанных с этими описаниями. Можно, к примеру, составить один простой список открытых вопросов в конце спецификаций.

КОГДА СПЕЦИФИКАЦИИ ГОТОВЫ?

Для любого графика разработки, предусматривающего этап планирования, написание и проверка спецификаций — естественное завершение. Теоретически, после этого команда должна знать большинство деталей по итоговому продукту и принципы его построения. Проект готов к запуску, и с планировщиков и проектировщиков все внимание перемещается на программистов и тестировщиков.

КОГДА ПОРА ОСТАНОВИТЬСЯ?

Когда спецификации готовы? Это довольно спорный вопрос. Всегда остаются нерешенные вопросы или зависимость от других компаний и проектов, которые еще не совсем обозначились. Штамп «выполнено» может означать самые разные уровни завершенности и качества в зависимости от проекта и команды.

Здесь нет правильных и неправильных ответов: иногда риск слабых спецификаций перевешивается сжатыми сроками или иными соображениями. Как и с любым другим высокоуровневым аспектом проекта

(качество кода, устойчивость, производительность), только лидеры команды могут решить, когда считать работу выполненной. И, конечно, чем больше итераций в общей стратегии инжиниринга, тем больше свободы в написании спецификаций.

Общее правило, однако, неизменно: чем грамотнее спецификации, тем выше вероятность своевременного достижения результата. В таком случае закономерен вопрос: какая степень вероятности вам нужна? Стоит ли тратить время, чтобы улучшить спецификации на 5%? Или программисты и МП смогут выяснить эти детали по ходу работы? Простых ответов нет: придется опираться на собственное чутье. Думаю, опыт управления проектами значит здесь больше, чем мастерство программирования или писательский талант.

Однако важно отметить: какого бы уровня завершенности вы ни ожидали, единственный способ завершить спецификации — проверка. Нужно, чтобы тимлиды (и читатели спецификаций) проверяли текст и давали обратную связь. (Подробнее об этом процессе — в следующем разделе.)

ЧТО ДЕЛАТЬ С ОТКРЫТЫМИ ВОПРОСАМИ

Как бы хорошо команда ни справилась с процессом проектирования, во время написания спецификаций всегда остаются неразрешенные вопросы. Если оставить их без внимания, разразится катастрофа. Многие неприятности в середине проекта вызваны именно неучтенными вопросами спецификаций. Крайне важно, чтобы МП взял на себя сбор и проверку этих вопросов и поощрял команду обратить на них внимание как можно раньше. Непростая задача для неопытных менеджеров проекта, так как они настолько заняты другими задачами по спецификациям, что не уделяют достаточно времени открытым вопросам. Возможно, нужно один раз пережить печальный опыт столкновения с нерешенным вопросом на более позднем этапе проекта, чтобы осознать, насколько важно отслеживать эти трудности как можно раньше.

Эффективное решение открытых вопросов зависит исключительно от усидчивости. Кто-то должен одновременно исследовать потенциальные проблемы и записывать их. Все предельно просто. Записав вопросы, их можно приоритизировать, поручить людям и решить. Если вы не уделите этому время, вероятность возникновения серьезных проблем будет зависеть от воли случая, а не ваших знаний и навыков.

Если вы все же тем или иным образом отслеживаете трудные моменты, предлагаю вам основные вопросы, которые помогут приоритизировать и доработать их.

- Когда нужно решить эту проблему? Кто наиболее компетентен, чтобы принять необходимые решения?
- Можно ли изолировать эту проблему, к примеру, свести ее до конкретного компонента или сценария? Или она влияет на всю функцию либо на весь проект?
- Перечислите возможные варианты решения проблемы, которые еще находятся на рассмотрении. (К примеру, мы будем использовать ASP или PHP, но не JSP.) Как каждое возможное решение повлияет на спецификации?
- Можно ли вычеркнуть эту проблему? Как она на самом деле влияет на клиента в самом приоритетном пользовательском сценарии?
- Возможно ли разделить проблему на небольшие задачи, которые можно (и следует) делегировать другим людям?
- Кто или что мешает решению проблемы и что делается для устранения препятствия? (Решение может носить технический или политический характер.)

Если набралось много серьезных вопросов и сложно разделить их на более мелкие задачи, значит что-то пошло не так на этапе проектирования и/или руководство команды допустило ошибку. Поиск решения проблемы выходит за рамки управления открытыми вопросами (глава 11).

Как закрыть пробелы спецификаций

Если вы уделяете открытым вопросам достаточно внимания, проблемы в графике вполне можно закрыть, спрогнозировав решение этих вопросов. Основная идея, которую часто насмешливо называют «заплатками», показана на рисунке 7.1.

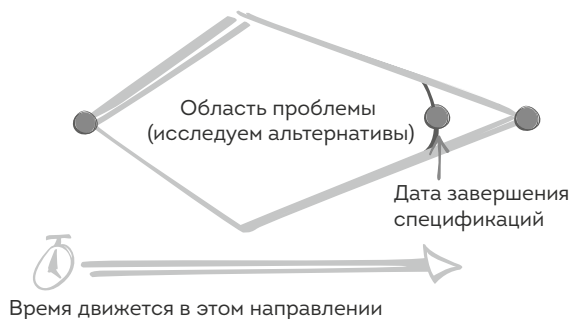


Рис. 7.1. Закрываем пробелы проектирования и спецификаций

Если все сделать правильно, спецификации можно доработать и завершить вовремя даже при наличии нерешенных вопросов по проектированию. «Заплатки» сопряжены с некоторым риском: вы прогнозируете, насколько эффективно ваша команда решит оставшиеся вопросы, вместо того чтобы ждать, когда она на самом деле решит их. Однако этот риск редко можно назвать серьезным. Все зависит от того, насколько хорошо вы понимаете оставшиеся проблемы и насколько верны ваши предположения относительно них. Возьмем, к примеру, две команды. У команды А длинный, но досконально осмысленный список проблем. У команды Б список короткий, но она совершенно ничего в нем не понимает. Какая команда уложится в сроки? Я бы поставил на команду А (включите ее гимн, пожалуйста). Мой врожденный скептицизм говорит, что короткий список проблем команды Б указывает на то, что она нашла далеко не все открытые вопросы по спецификациям. Команда А потратила больше времени на изучение своих

открытых вопросов и лучше подготовлена ко всем трудностям, которые ждут ее впереди.

Важно отметить, что закрыть пробелы не значит бросить проектную работу, необходимую для принятия окончательных решений. Это значит, что МП пытается сделать шаг назад и объективно оценить ситуацию, чтобы уложиться в график.

Чтобы помочь вам закрыть пробелы, перечислим несколько вопросов по каждой открытой проблеме.

- Есть ли рабочие элементы, которые нужно выполнить, независимо от выбранного варианта? Если да, их следует оценить и добавить в спецификации. При необходимости к выполнению этих элементов можно приступить до того, как вы закончите спецификации.
- Может ли МП или проектировщик решить вопрос? Приведет ли его решение к появлению новых рабочих элементов? (К примеру, можно работать параллельно с программистом, начиная с тех рабочих элементов, которые абсолютно понятны, а МП в это время будет искать варианты решения открытых вопросов.)
- Есть ли возможные альтернативные решения проблемы, которые все еще находятся на рассмотрении?
- Какая из альтернатив наиболее дорогостоящая? Постарайтесь оценить рабочие элементы, опираясь на этот подход, и преобразовать спецификации и список рабочих элементов в худший возможный проектный план.
- Это центральный или основной компонент? Где программисту придется внедрить его? Допустимо ли построить его позже, на этапе реализации? Зависят ли от него другие компоненты?
- Этот вопрос можно локализовать, сузить, разделить или вычеркнуть? Если нет, поставьте его на первую строчку списка приоритетов.

Далеко не всегда удастся закрывать пробелы успешно. Возможно, у вас будет уверенный старт, и дело пойдет как по маслу, но вы все

равно увидите, что вам предстоит далекий путь. Несмотря на это, никогда не помешает приложить усилия и закрыть все пробелы. Неопытным командам зачастую нужен «пинок», чтобы они встретились лицом к лицу со своими трудностями (техническими или другими). Менеджеры же не всегда понимают всю сложность ситуации, пока не нахлынут неприятности. Вероятно, стоит действовать заранее, а не ждать, когда ваш график окажется под угрозой.

ПОЧЕМУ ТАК ВАЖНО ДОДЕЛАТЬ СПЕЦИФИКАЦИИ ВОВРЕМЯ

В графике проекта должен быть отведен конкретный день для завершения спецификаций. Возможно, это самый важный день для менеджеров как участников проекта. Зачастую написание спецификаций — их основной или, возможно, единственный значимый вклад в проект. Как только спецификации будут готовы, внимание МП переместится на управление проектом, включая помощь команде во время перехода на этап разработки.

После завершения спецификаций подход проектной команды должен измениться. Всем следует понимать, что на этом этапе обсуждения подошли к концу: многие серьезные решения уже приняты. Команда преодолела немало трудностей, связанных с поиском оптимальных проектных решений и составлением упорядоченного плана. Теперь задача МП — убедиться, что все участники процесса разделяют эту позицию и вклад каждого из них учтен.

ВНИМАНИЕ!

Один на один – лучший способ сообщить людям, что вы цените их работу. Не рассчитывайте, что электронное письмо, адресованное всей команде, много значит для каждого из сотрудников. Загляните в офис или позвоните по телефону. Короткий разговор «вживую» гораздо весомее, чем любое электронное письмо.

Хотя мероприятия по поднятию командного духа и мотивационные речи — дело непростое, должно быть командное признание выполненной в срок работы. Чем проще, тем лучше: отпустите людей пораньше с работы, устройте обед на свежем воздухе или целую неделю угощайте их бесплатным пивом и закусками. Любое позитивное вкрапление в рутину (например, выезд) — лучший способ помочь командам перейти на новый этап и «перезарядить батарейки», готовясь к выполнению предстоящих в ближайшие недели и месяцы задач.

ПРОВЕРКА И ОБРАТНАЯ СВЯЗЬ

Самая большая ошибка, которую допускают со спецификациями, — ожидание официальной проверки для получения обратной связи. Проверки нужны для совершенствования, а не для того, чтобы сделать первый обзор и принять окончательное решение. Это еще одна причина, по которой процесс проектирования так важен: он показывает, что проектные решения прошли множество итераций и авторам предоставлялось немало шансов внести в спецификации удачные предложения. Тимлиды должны создать условия для неформальной проверки на ранних этапах и выложить черновой вариант спецификаций в интранете. Однако это не значит, что собрания по проверке последних — пара пустяков; каждый должен подходить к этой задаче с предельно четким пониманием, чего от него ждут.

Есть разные способы проверить спецификации, но большинство подразумевает проведение собрания. Насколько формальным или неформальным оно будет, зависит от лидеров команды. В любом случае цель — ответить на два вопроса: «Спецификации достаточно подробны, чтобы стать нашим ориентиром на этапе разработки?» и «Будет ли результат соответствовать требованиям и задачам проекта?». Конечно, можно задать другие, более конкретные вопросы, но все они опираются на эти два ключевых момента. Проверка должна быть направлена на поиск обоснованных ответов.

КАК ПРОВЕРЯТЬ СПЕЦИФИКАЦИИ

Проверка спецификаций — это командный процесс. Хотя в центре внимания находятся документ и его авторы, задача — убедиться, что все участники проекта согласны с тем, что сказано в этом документе. Самый простой и быстрый способ сделать это — собрать всех в одной комнате, чтобы каждый услышал ответы на вопросы, которые будут заданы. Я видел, как проверка спецификаций проводится по электронной почте или конференц-связи, и не могу сказать, что доволен результатами. Как только появлялся спорный вопрос, я жалел, что не нахожусь рядом с командой, чтобы использовать доску и дать объяснения в реальном времени. Чем сложнее спецификации, тем важнее собрать всех в одной комнате. (Если вы вынуждены работать виртуально и хотите, чтобы вся команда участвовала в проверке, проведите обсуждение в небольших группах по три-пять человек. Для таких сложных задач, как проверка спецификаций, конференц-связь и видеоконференции в больших группах быстро превращаются в трагикомедии.)

Следует «застолбить» на час или два конференц-зал за несколько дней до собрания. Если спецификации готовы для проверки (по мнению автора и по критериям, определенным лидерами команды), их следует отправить всем участникам собрания в качестве приглашения. Насколько я помню, мне удалось это сделать всего несколько раз. Чаше я предупреждал о собрании примерно за неделю и информировал всех, что они получат документ по электронной почте за 24 часа до собрания. Некоторым такой срок не нравился, но мой опыт показывает, что это самый эффективный способ довести до людей обновленный документ и заставить их прочитать его. Если предупредить заблаговременно, люди, скорее всего, запланируют чтение именно на этот 24-часовой период.

Точно так же я считаю, что чтение спецификаций — обязательное и вполне справедливое требование для всех участников собрания. Люди, которым нужно прочитать спецификации, сделают это. Если же они

не могут найти время, чтобы хотя бы пролистать документ и разглядеть очевидные проблемы, эта работа не приоритетна для них, и на собрании им делать нечего.

При соответствующих полномочиях я делал чтение спецификаций до собрания обязательным требованием для всей команды. Это гарантирует две вещи. Во-первых, сокращает количество участников, потому что придут только те, кому это действительно нужно. Вероятность того, что комната будет переполнена лишними критиками, отпадает сама собой. Во-вторых, собрание пройдет намного быстрее, потому что у всех будет примерно одинаковое понимание ситуации. Те, кто не читал спецификации, выдадут себя соответствующими вопросами. Если эти вопросы обоснованы, их следует рассмотреть; если они подробно разъясняются в спецификациях, будет справедливо попросить этих членов команды прочитать соответствующий раздел и поговорить с автором спецификаций после собрания.

КТО БУДЕТ НА СОБРАНИИ И КАК ЕГО ПРОВЕСТИ

На собрании должен присутствовать по крайней мере один человек из каждого основного отдела (программирование, тестирование и т. д.), а также любые другие значимые участники проекта (бизнес, проектирование, документация). Проверка спецификаций — открытое мероприятие для всей команды: если тестировщики или программисты заинтересовались документом и удосужились его прочитать, их присутствие приветствуется, даже если они не работают над конкретной функцией. Лидеров команды тоже можно пригласить: они вправе сами решать, участвовать в собрании или нет. Если они хорошо справляются со своей работой, то наверняка владеют достаточной информацией и вполне могут принять участие, к примеру, в обсуждении самых спорных моментов. С другой стороны, если команда неопытная, возможно, лидерам стоит посещать каждое собрание.

Его проводит МП (или автор спецификаций). Процесс простой: нужно отвечать на вопросы. Если вопросов нет (то есть мы живем в сказочной

стране), в комнате присутствуют нужные люди и они довольны спецификациями, то собрание закончено. Одни менеджеры любят проводить критический обзор прототипа, и это хорошо. Другие предпочитают пошаговый разбор документа, раздел за разделом. Лично я считаю это пустой тратой времени (если я написал хорошие спецификации, и все их прочитали, зачем так подробно разбирать текст?), однако некоторые команды любят это. Используйте все, что подходит для вашей конкретной ситуации. Главное, чтобы люди участвовали в здоровой дискуссии, задавали правильные вопросы и вместе искали ответы.

Присутствующие обсуждают каждый вопрос, пока не найдут ответ, удовлетворяющий спросившего, или не добавят новый пункт в список открытых вопросов. Когда последних больше не останется, МП зачитает список (можно также воспользоваться доской) и решит, стоит ли проводить еще одно собрание. Если причин для нового созыва нет, спецификации признаются проверенными, и останется только найти решения открытых вопросов.

После завершения проверки спецификаций МП должен ввести временные ограничения для поиска ответов на новые вопросы и проблемы. Сразу же после собрания следует разослать всем участникам электронное письмо с кратким резюме открытых вопросов, сроками их решения и датой следующего обсуждения (если она намечена в графике). Это особенно важно, если какой-либо открытый вопрос мешает одному из членов команды делать свою работу. По сути, такие блокирующие вопросы следует обозначить на обсуждении спецификаций и уделить им особое внимание.

СПИСОК ВОПРОСОВ

Есть вопросы, которые нужно задавать на каждом обсуждении спецификаций, учитывая распространенные ошибки и проблемы, которые наблюдаются год за годом. Даже если эти сложные вопросы не связаны

с конкретными проблемами, они все равно вынуждают команду взглянуть более критично на то, что она делает. Помните: это не итоговый экзамен — все могут знать вопросы заранее. В ваших интересах убедиться, что люди пришли на собрание подготовленными.

Поскольку проектирование и написание спецификаций — процесс оптимистичный, участники собрания просто обязаны проявить скептицизм и искать все, что могло ускользнуть от внимания автора. (Только не перегибайте палку! Не нужно проявлять чрезмерную жесткость и доводить людей до слез. Если команда не готова к обсуждению спецификаций, тимлиды виноваты в этом не меньше рядовых ее членов.) Даже если все присутствующие знают правильные вопросы, кто-то должен копнуть глубже, чтобы доискаться до нужных ответов.

Вот мой список, хотя я призываю вас доработать эти вопросы и добавить свои.

- Соответствует ли спецификациям список рабочих элементов программиста? Как каждый основной компонент спецификаций соотносится с каждым рабочим элементом? На каком этапе проектирования существует наибольшая вероятность обнаружить неучтенные рабочие элементы?
- Где уязвимые места проектирования? Какие компоненты или интерфейсы можно назвать слабыми? Почему их нельзя улучшить?
- Каковы самые сильные стороны? В чем мы больше всего, а в чем меньше всего уверены? Опираются ли наши сильные стороны и наша уверенность на самые важные компоненты?
- У нас нужный уровень качества? Это проектное решение будет таким же надежным, производительным и практичным, как требует видение? Реалистичны ли прогнозы тестирования?
- Почему более простое проектное решение не будет оптимальным? Нам действительно нужны все эти сложности и функциональности? Какие факты или обоснования не позволяют нам пойти по пути упрощения?

- Какие зависимости присутствуют в этом проектном замысле? Есть ли технологии, корпорации, проекты или другие спецификации, которые могут провалиться и навредить или помешать нашей работе? У нас есть план действий для внештатных ситуаций?
- Какие проектные элементы могут измениться с наибольшей вероятностью? Почему?
- Обладают ли специалисты по тестированию, документации, маркетингу и другим функциям всей необходимой информацией, чтобы выполнить работу на высочайшем уровне? Какие у них вопросы и как они будут решены? Или у нас есть обоснованные причины проигнорировать их?
- Каковы основные опасения менеджеров проекта, программистов и тестировщиков относительно этих спецификаций? Или функционала?
- Можно ли позаимствовать код из других функций, которые строятся для этого проекта?
- Выполнили ли мы требования по доступности и локализации пользовательского интерфейса?
- Каковы риски безопасности? Почему неразумно избавляться от них? Указаны ли они в спецификациях, включая возможные решения (например, модели угроз)?
- Располагаем ли мы надежными доказательствами, что конкретные пользователи могут прибегнуть к UI-дизайну, чтобы успешно выполнить то, что им нужно?

РЕЗЮМЕ

- Спецификации выполняют три задачи: убедиться, что мы делаем нужный продукт, наметить в графике контрольную точку, которая означает завершение этапа планирования, и создать условия

для подробного обсуждения и обратной связи от разных участников на протяжении всего проекта.

- Спецификации решают только определенные проблемы. Лидеры команды должны четко понимать, с чем именно они пытаются разобраться с помощью спецификаций, а что требует другого подхода.
- Хорошие спецификации упрощают информацию. Это прежде всего формат общения.
- Спецификации сильно отличаются от проектирования.
- Нужно четко обозначить, кто пишет спецификации.
- Закрыть пробелы — один из подходов к открытым вопросам и скорому завершению спецификаций.
- Процесс обсуждения спецификаций — простейший способ контрлировать их качество.

УПРАЖНЕНИЯ

А. Вам понадобится большая упаковка LEGO и еще один менеджер проекта. Разделите LEGO на две части с одинаковым количеством и типом фигурок. Повернитесь спиной друг к другу, и пусть один из вас построит что-то из своего набора (неважно что). После этого тот, кто строил, должен объяснить (только словами) другому, как сделать точно такой же предмет. Сравните результаты. Затем повторите, поменявшись ролями.

Б. Почему МП стремятся использовать спецификации для целей, для которых они не предназначены? Какую проблему они пытаются (тщательно) решить?

В. Что говорит качество спецификаций о МП? Можете ли вы догадаться, судя только по спецификациям, каким будет качество программного продукта?

Г. Представьте действие, которое вы делаете часто и практически не задумываясь, например завязываете шнурки, ставите будильник или

включаете DVD. Запишите, как вы это делаете — так, чтобы другой человек смог выполнить ваши инструкции. Нарисуйте иллюстрацию. Попробуйте сами выполнить свои инструкции, слово в слово, или попросите кого-то другого сделать это. Обратите внимание на результат, скорректируйте инструкции и сделайте снова.

Д. Найдите худшие спецификации, какие вы только видели (обратитесь к друзьям, коллегам, ко всем, чья работа предполагает их составление). А потом попросите найти лучшие спецификации. Составьте ваш собственный список общих характеристик, позитивных и негативных.

Е. Как убедиться, что в спецификациях достаточно подробностей? Как понять, что вы перегнули палку или не дотянули до нужного уровня?

Ж. Среди ваших знакомых есть фанатик Visio, UML или других инструментов? Имеете ли вы доказательства, что эта зависимость приводит к некачественным спецификациям? Сделайте одолжение своей команде: боритесь за качество. Соберите всех людей, которые читают его спецификации, — пусть подпишут петицию за сокращение объема Visio-документов. Передайте ее менеджеру проекта. Включите туда же список задач, которые спецификации могут и не могут выполнить.

З. Если вы знаете, что до завершения спецификаций осталось всего несколько дней, как убедиться, что оставшееся время будет использовано эффективно? Чем может помочь команда? Что следует сделать, чтобы максимально увеличить вероятность плодотворного обсуждения готовых спецификаций?

И. Представьте такой сценарий: вы написали блестящие спецификации с потрясающими иллюстрациями, понятные читателям и досконально обдуманные. Но вашему лучшему инженеру они не нравятся. Ему не нравится не только то, как они написаны, но и идеи, которые они отражают. До завершения спецификаций осталось всего два дня. Что вы можете сделать? Что следует сделать в следующий раз, чтобы предотвратить подобную ситуацию?



[Почитать описание, рецензии
и купить на сайте](#)

Лучшие цитаты из книг, бесплатные главы и новинки:



[Mifbooks](#)



[Mifbooks](#)



[Mifbooks](#)