

# Оглавление

---

Yes, we did! Предисловие к первому русскоязычному изданию.....	3
Предисловие к редакции 2017 года.....	4
Оглавление.....	7
Благодарности.....	12
Предисловие Джеффа Сазерленда.....	13
Предисловие Майка Кона.....	15
Пролог: Эй! А Scrum-то работает!.....	16
Пролог: второе издание, 8 лет спустя.....	17
<b>Вступление.....</b>	<b>18</b>
Предупреждение.....	20
Зачем я это написал.....	20
Так что же такое Scrum?.....	20
<b>Как мы работаем с product backlog'ом.....</b>	<b>22</b>
Дополнительные поля для user story.....	26
Как мы ориентируем product backlog на бизнес.....	27
<b>Как мы готовимся к планированию спринта.....</b>	<b>29</b>
<b>Как мы планируем спринт.....</b>	<b>32</b>
Почему без product owner'а не обойтись.....	33
Почему качество не обсуждается.....	35
Планирование спринта, которое никак не заканчивается.....	37
Распорядок встречи по планированию спринта.....	38
Определяем длину спринта.....	39
Определение цели спринта.....	41
Как product owner может влиять на то, какие истории попадут в спринт?.....	43
Как команда принимает решение о том, какие истории включать в спринт?.....	45
Планирование, основанное на интуиции.....	45
Планирование, основанное на методе оценки производитель- ности.....	47

Какую технику мы используем для планирования?.....	51
Почему мы используем учетные карточки.....	52
Критерий готовности.....	56
Оценка трудозатрат с помощью игры в planning poker.....	57
Уточнение описаний историй.....	59
Разбиение историй на более мелкие истории.....	61
Разбиение историй на задачи.....	61
Выбор времени и места для ежедневного Scruma.....	63
Когда пора остановиться.....	63
Технические истории.....	64
Как мы используем систему учёта дефектов для ведения productbackloga.....	68
Свершилось! Планирование спринта закончено!.....	69
<b>Как мы доносим информацию о спринте до всех в компании....</b>	<b>70</b>
<b>Как мы создаём sprint backlog.....</b>	<b>73</b>
Формат sprint backloga.....	74
Как работает доска задач.....	75
Пример 1 – после первого ежедневного Scruma.....	75
Пример 2 – еще через пару дней.....	77
Как работает burndown-диаграмма.....	78
Тревожные сигналы на доске задач.....	79
Эй, как насчет отслеживания изменений?.....	80
Как мы оцениваем: в днях или часах?.....	81
<b>Как мы обустроили комнату команды.....</b>	<b>82</b>
Уголок обсуждений.....	83
Усадите команду вместе.....	84
Не подпускайте product owner’а слишком близко.....	86
Не подпускайте менеджеров и тренеров слишком близко.....	87
<b>Как мы проводим ежедневный Scrum.....</b>	<b>88</b>
Как мы обновляем доску задач.....	89
Как быть с опоздавшими.....	91
Как мы поступаем с теми, кто не знает, чем себя занять.....	91
<b>Как мы проводим демо спринта.....</b>	<b>94</b>

Почему мы настаиваем на том, чтобы каждый спринт заканчивался демонстрацией.....	95
Чеклисты по подготовке и проведению демо.....	96
Что делать с «не демонстрируемыми» вещами.....	97
<b>Как мы проводим ретроспективы.....</b>	<b>98</b>
Почему мы настаиваем на том, чтобы все команды проводили ретроспективы.....	99
Как мы проводим ретроспективы.....	100
Как распространять уроки между командами.....	102
Изменения. Быть или не быть.....	103
Типичные проблемы, которые обсуждают на ретроспективах...104	
«Нам надо было больше времени потратить на разбиение историй на подзадачи».....	104
«Очень часто беспокоят извне».....	104
«Мы пообещали выполнить огромный кусок работы, а закончили только половину».....	105
«У нас в офисе бардак и очень шумно».....	105
<b>Slack-время между спринтами.....</b>	<b>106</b>
<b>Как мы планируем релизы и составляем контракты с фиксированной стоимостью.....</b>	<b>110</b>
Определяем свою приёмочную шкалу.....	111
Даём временные оценки наиболее важным историям.....	112
Прогнозируем производительность.....	114
Сводим всё в план релиза.....	115
Корректируем план релиза.....	117
<b>Как мы сочетаем Scrum и XP.....</b>	<b>118</b>
Парное программирование.....	119
Разработка через тестирование (TDD).....	120
TDD и новый код.....	122
TDD и существующий код.....	122
Эволюционный дизайн.....	124
Непрерывная интеграция (Continuous integration).....	125
Совместное владение кодом (Collective code ownership).....	125

Scrum и XP: заметки с передовой. Второе издание. Х. Книберг	
Информативное рабочее пространство.....	125
Стандарты кодирования.....	125
Устойчивый темп работы.....	126
<b>Как мы тестируем.....</b>	<b>128</b>
Скорее всего, вам не избежать фазы приёмочного тестирования.....	129
Минимизируйте фазу приёмочного тестирования.....	130
Повышайте качество, включив тестировщиков в Scrum-команду.....	132
Тестировщик – это «последняя инстанция».....	132
Чем занимается тестировщик, когда нечего тестировать?..	133
Повышайте качество – делайте меньше за спринт!.....	135
Стоит ли делать приёмочное тестирование частью спринта?..	135
Соотношение спринтов и фаз приёмочного тестирования.....	136
Подход №1: «Не начинать новые истории, пока старые не будут готовы к реальному использованию».....	138
Подход №2: «Начинать реализовывать новые истории, но наивысшим приоритетом ставить доведение старых до ума».....	139
Неправильный подход: «Клепать новые истории».....	140
Не бегите быстрее самого медленного звена системы.....	141
Возвращаясь к реальности.....	142
<b>Как мы управляемся с несколькими Scrum-командами.....</b>	<b>143</b>
Сколько сформировать команд.....	144
Виртуальные команды.....	145
Оптимальный размер команды.....	147
Синхронизировать спринты или нет?.....	148
Почему мы ввели роль «тимлида».....	149
Как мы распределяем людей по командам.....	151
Нужны ли узкоспециализированные команды?.....	153
Подход №1: команды, специализирующиеся на компонентах.....	153
Подход №2: универсальные команды.....	154

Scrum и XP: заметки с передовой. Второе издание. Х. Книберг	
Стоит ли изменять состав команды между спринтами?.....	155
Участники команды с частичной занятостью.....	156
Как мы проводим Scrum-of-Scrums.....	157
Scrum-of-Scrums уровня продукта.....	158
Scrum-of-Scrums уровня компании.....	159
Чередование ежедневных Scrum'ов.....	160
«Пожарные» команды.....	161
Разбивать product backlog или нет?.....	163
Подход первый: Один product owner – один backlog.....	164
Подход второй: Один product owner – несколько backlog'ов...	165
Подход третий: Несколько product ownerов – несколько backlog'ов. ....	166
Параллельная работа с кодом и ветки.....	167
Ретроспектива для нескольких команд.....	168
<b>Как мы управляем географически распределёнными командами..</b>	<b>170</b>
Оффшорная разработка.....	173
Члены команды, работающие из дому.....	174
<b>Памятка ScrumMastera.....</b>	<b>176</b>
В начале спринта.....	177
Каждый день.....	178
В конце спринта.....	179
<b>Заключительное слово.....</b>	<b>179</b>
Список рекомендованной литературы.....	181
Обавторе.....	182

# Благодарности

Первый черновик этой книги был написан за выходные, но это был супер интенсивный уик-энд - 150% фокус-фактора! :o)

Огромное спасибо моей жене Софии и детям Дейве и Дженни за что, выдержали мою асоциальность в те дни, и родителям Софии - Еве и Йёргену за то, что пришли помочь с семейными хлопотами.

Благодарю также всех моих коллег из компании Crisp в Стокгольме и людей в Scrum User Group на Yahoo за вычитку и помощь в улучшении этой книги.

И, в конечном счёте, спасибо всем моим читателям, которые поставляют бесперебойный поток полезной обратной связи. Я особенно рад слышать, что эта книга зажгла стольких из вас опробовать методы гибкой разработки!

# Предисловие Джеффа Сазерленда

Командам необходимо знать основы Scrum'a. Как создать и оценить product backlog? Как получить из него sprint backlog? Как работать с burndown-диаграммой и вычислять производительность(velocity) своей команды? Книга Хенрика – это базовое руководство для начинающих, которое поможет командам перейти из состояния «мы пробуем Scrum» в состояние «мы успешно работаем по Scrum'у».

Хорошая реализация Scrum'a становится всё важнее и важнее для команд, которые хотят получить инвестиции. Я выступаю в качестве тренера по гибким методологиям для группы компаний с венчурными инвестициями, помогая им в стремлении вкладывать деньги только в настоящие Agile-компании. Глава группы инвесторов требует от компаний, составляющих инвестиционный портфель, ответа на вопрос, знают ли они производительность своих команд. Многих этот вопрос ставит в тупик. Будущие инвестиции требуют от команд знания собственной производительности разработки программного обеспечения.

Почему это так важно? Если команда не знает собственной производительности, следовательно product owner не может разработать стратегический план развития продукта с достоверными датами релизов. Без такого плана компанию может постигнуть неудача, в результате чего инвесторы потеряют свои деньги.

С этой проблемой сталкиваются разнообразные компании: большие и маленькие, старые и новые, с финансированием и без. Во время недавнего обсуждения реализации Scrum'a компанией Google на лондонской конференции я решил узнать у аудитории, состоящей из 135 человек, кто из них использует Scrum? Я получил утвердительный ответ лишь от тридцати человек. Затем я поинтересовался, соответствует ли их процесс Nokia-стандарту итеративной разработки. Итеративная разработка – это ключевое положение Agile Manifest'a: «Постарайтесь предоставлять версии работающего программного обеспечения как можно чаще и раньше». В результате проведения ретроспектив с сотнями Scrum-команд в течение нескольких лет, Nokia выработала некоторые базовые требования к итеративной разработке:

- Итерации должны иметь фиксированную длину и не превышать шести недель.
- К концу каждой итерации код должен быть протестирован отделом качества (QA) и работать как следует.

Из тридцати человек, которые сказали, что работают по Scrum'у, лишь половина подтвердила, что их команды придерживаются первого принципа Agile Manifest'a и соответствующую Nokia-стандарту.

Затем я спросил их, придерживаются ли они Scrum-стандарта, разработанного Nokia:

- У Scrum-команды должен быть один product owner и команда должна знать, кто это.
- У product owner'a должен быть один product backlog с историями и их оценками, выполненными командой.
- У команды должна быть burndown-диаграмма, а сама команда должна знать свою производительность.
- На протяжении спринта никто не должен вмешиваться в работу команды.

Из тридцати команд, внедряющих Scrum, только у трёх процесс разработки соответствовал стандартам Nokia. Я думаю, что только эти три команды получают дальнейшие инвестиции от венчурных капиталистов.

Основная ценность книги Хенрика состоит в том, что если вы будете следовать его советам, то у вас будет и product backlog, и оценки для product backlog'a, и burndown-диаграмма. Вы также будете знать производительность вашей команды и сможете использовать все наиболее важные практики высокоэффективных Scrum-команд. Вы пройдёте Nokia Scrum-тест, за что инвесторы оценят вас по достоинству. Если вы – начинающая компания, то, возможно, вы получите такие жизненно важные для вашего проекта финансовые вливания. Возможно вы – будущее разработки программного обеспечения, вы – создатель нового поколения программ, которые станут лидерами рынка.

Джефф Сазерленд (**Jeff Sutherland**),  
доктор наук, со-автор Scrum



# Предисловие Майка Кона

И Scrum, и XP (экстремальное программирование) требуют от команд завершения вполне осязаемого куска работы, который можно предоставить пользователю в конце каждой итерации. Эти итерации планируются таким образом, чтобы быть короткими и фиксированными по времени. Такая целенаправленность на выпуск рабочего кода за короткий промежуток времени означает только одно: в Scrum и XP нет места теории. Agile-методологии не гонятся за красивыми UML моделями, выполненными при помощи специальных case-средств, за созданием детализированных спецификаций или написанием кода, который сойдёт на все случаи жизни. Вместо этого Scrum и XP команды концентрируются на том, чтобы завершить необходимые задачи. Эти команды могут мириться с ошибками в ходе работы, но они понимают, что лучшее средство выявить эти ошибки – это перестать думать о софте на теоретическом уровне анализа и дизайна, и, закатав рукава, полностью посвятить себя созданию продукта.

Именно акцент на действии, а не на теории ярко выделяет эту книгу среди прочих. То, что Хенрик разделяет эти взгляды, видно с самых первых страниц книги. Он не предлагает нам длинное описание того, что такое Scrum; вместо этого он просто ссылается на необходимые веб-ресурсы. Первым делом Хенрик начинает с описания того, как его команда работает со своим product backlog'ом. Затем он проходит по всем элементам и практикам правильно поставленного agile-проекта. Без теоретизирования. Без справочных данных. Ничего этого не нужно: книга Хенрика – не философское объяснение, почему Scrum работает или почему мы должны делать так, а не иначе. Это описание того, как работает одна успешная agile-команда.

Хенрик предлагает набор избранных практик и описывает живые примеры, чтобы помочь нам понять, как использовать Scrum и XP на передовой.

Майк Кон (**Mike Cohn**),  
автор книг ***Agile Estimating and Planning***  
и ***User Stories Applied***.

# Пролог: Эй! А Scrum-то работает!

Scrum работает! По крайней мере, нам он подошёл (я имею в виду проект моего клиента из Стокгольма, чьё имя я не хотел бы называть). Надеюсь, вам он тоже подойдёт! И, возможно, именно эта книга поможет вам на пути освоения Scrum'a.

Это был первый случай в моей жизни, когда я увидел как методология (ну да, Кен [Кен Швебер – соавтор Scrum'a], – фреймворк) работает «прямо из коробки». Просто подключи и работай. И при этом все счастливы: и разработчики, и тестеры, и менеджеры. Вопреки всем передрягам на рынке и сокращению штата сотрудников, Scrum помог нам выбраться из сложнейшей ситуации, позволил сконцентрироваться на наших целях и не потерять свой темп.

Не хочется говорить, что я удивлён, но... так и есть. После беглого знакомства с парой книг по теме, Scrum произвёл на меня хорошее впечатление, даже слишком хорошее, чтобы быть похожим на правду. Так что неудивительно, что я был настроен слегка скептически. Однако после года использования Scrum'a, я настолько впечатлён (и большинство людей в моих командах тоже), что, скорее всего, буду использовать Scrum во всех новых проектах, ну, разве что кроме случаев, когда есть веская причина не делать этого.

# Пролог: второе издание, 8 лет спустя

Прошло восемь лет, а эта книга всё ещё довольно популярна. Вау! Я никогда бы не мог предсказать, что небольшая книга вроде этой может оказать так много влияния! Я регулярно встречаю команды, менеджеров, коучей и тренеров во всех уголках мира, которые пользуются этой книгой как первичным руководством по гибкой разработке.

Но вот в чём вещь: я столько всего нового узнал с 2007 года и так многому научился! Так что эта книга очень сильно нуждается в обновлении.

Со времени издания этой книги, у меня был шанс поработать с большим количеством agile-команд и lean-экспертами; некоторые даже стали мне чем-то вроде наставников. Отдельное спасибо Jeff Sutherland, Mary и Tom Poppendieck, Jerry Weinberg, Alistair Cockburn, Kent Beck, Ron Jeffries и Jeff Patton - я не могу себе представить лучшей группы советчиков!

Я также помог огромному числу компаний применить эти идеи на практике: компаниям в кризисе, а также сверх-успешным компаниям, которые желают стать ещё лучше. В итоге, это было довольно головокружительное путешествие!

Когда я перечитываю первое издание этой книги, я поражаюсь, с каким количеством вещей я всё ещё согласен. Но там также есть страницы, которые я бы хотел вырвать, воскликнув: «Какого \$%# я тогда думал? Не делайте так никогда! Есть намного лучшие способы!»

Но, как говорят, что написано пером, не вырубишь топором. Что есть, то есть. Но никто мне не может запретить добавить комментарии!

Так вот это и есть второе издание - версия с авторскими аннотациями к оригинальной книге. Что-то вроде director's cut - специальной режиссёрской редакции. Думайте об этом, как если я стою у вас за плечом, пока вы читаете эту книгу, комментируя то тут, то там, подбадривая вас, а иногда подсмеиваясь или чертыхаясь.

Вот так выглядят мои комментарии. Всё остальное (кроме этого предисловия) - оригинальная книга, как есть. Только эти тёмные рамочки - мои новые мысли и рефлексия ко второму изданию.

Я также привнесу несколько новых примеров из ряд компаний, в основном из Spotify (так там я проводил большую часть своего времени последнее время), но также и в других компаниях.

Enjoy!  
Хенрик, март 2015.

# ЧАСТЬ 1

Вступление

Собираетесь начать практиковать Scrum у себя в компании? Или вы уже работаете по Scrum'у пару месяцев? У вас уже есть базовые понятия, вы прочитали несколько книг, а, возможно, даже получили сертификат Scrum Master'a? Поздравляю!

Однако, согласитесь, какая-то неясность всё равно остаётся.

По словам Кена Швебера, Scrum – это не методология, это фреймворк. А это значит, что Scrum не дает готовых рецептов, что делать в тех или иных случаях. Вот незадача!

Но у меня для вас есть хорошая новость: я расскажу, как именно я практикую Scrum... очень подробно и со всеми деталями. Однако, и без плохой новости не обойдётся: я расскажу всего лишь о том, как практикую Scrum я. Это значит, что вам не обязательно делать всё точно так же. На самом деле, в зависимости от ситуации, я и сам могу делать что-то по-другому.

Достоинство Scrum'a и одновременно самый большой его недостаток в том, что его необходимо адаптировать к вашей конкретной ситуации.

Моё видение Scrum'a формировалось на протяжении целого года и стало результатом экспериментов в команде численностью около 40-ка человек. Одна компания попала в крайне сложную ситуацию: постоянные переработки, авралы, серьёзные проблемы с качеством, проваленные сроки и прочие неприятности. Эта компания решила перейти на Scrum, но внедрить его толком у неё не получалось. В итоге эта задача была поручена мне. К тому времени для большинства сотрудников слово «Scrum» было просто набившим оскомину термином, эхо которого звучало время от времени в коридорах без какого-либо отношения к их повседневной работе.

Через год работы мы внедрили Scrum на всех уровнях компании: поэкспериментировали со всевозможными размерами команд (от 3 до 12 человек), попробовали спринты различной длины (от 2 до 6 недель) и разные способы определения критерия готовности, разнообразные форматы product и sprint backlog'ов (Excel, Jira, учетные карточки), разные стратегии тестирования, способы демонстрации результатов, способы синхронизации Scrum-команд и так далее. Также мы опробовали множество XP практик: с разными способами непрерывной интеграции, с парным программированием, TDD (разработкой через тестирование), и т.д. А самое главное – разобрались, как все это дело сочетается со Scrumом.

Scrum подразумевает постоянный процесс развития, так что история на этом не заканчивается. Я уверен, упомянутая мной компа-

ния будет продолжать двигаться вперед (если в ней и дальше будут проводить ретроспективы спринтов) и постоянно находить новые и новые способы эффективного применения Scrum'a, учитывая особенности каждой из сложившихся ситуаций.

### Предупреждение

---

Эта книга не претендует на звание «единственно правильного» учебного пособия по Scrum'у! Она всего лишь предлагает вам пример удачного опыта, полученного на протяжении года в результате постоянной оптимизации процесса. Кстати, у вас запросто может возникнуть ощущение, что мы всё поняли не так.

Эта книга отражает моё субъективное мнение. Она никоим образом не является официальной позицией Crisp'a [от переводчика – консалтинговая компания, в которой работает Хенрик] или моего нынешнего клиента. Именно поэтому я специально избегал упоминания каких-либо программных продуктов или людей.

### Зачем я это написал

---

Во время изучения Scrum'a я читал книги, посвящённые Scrum'у и Agile'у, рылся по различным сайтам и форумам, проходил сертификацию у Кена Швебера, засыпал его вопросами и проводил кучу времени, обсуждая Scrum с моими коллегами. Однако одним из наиболее ценных источников знаний стали реальные истории внедрения Scrum'a. Они превращают Принципы и Практики в... ну, в то, как вы это делаете. Они помогли мне предвидеть типичные ошибки Scrum-новичков, а иногда и избежать их.

Итак, вот и выпал мой шанс поделиться чем-то полезным. Это моя реальная история.

### Так что же такое Scrum?

---

Ой, простите, я совсем забыл про новичков в Scrum'е и XP. Если вы к ним относитесь, вам не мешало бы посетить следующие веб-ресурсы:

- <http://agilemanifesto.org/>
- <http://www.mountaingoatsoftware.com/scrum>
- <http://www.xprogramming.com/xpmag/whatisxp.htm>

Ознакомьтесь также со «Scrum Guide» - руководством по Scrum. Сегодня - это официальное описание Scrum, поддерживаемое

обоими авторами Jeff Sutherland и Ken Schwaber. <http://www.scrumguides.org>

Нетерпеливые же могут продолжать читать книгу дальше. Я объясню все основные Scrum'овские термины по ходу изложения.

Я надеюсь, что эта книга станет стимулом для тех, кто так же не против поделиться своими мыслями на счёт Scrum'а. Пожалуйста, не забывайте сообщать мне о них!

# ЧАСТЬ 2

Как мы работаем с  
product backlogom

[Купить книгу на сайте kniga.biz.ua >>>](http://kniga.biz.ua)



Product backlog – это основа Scrum’а. С него все начинается.

Ээээ... В начальной точке беклога продукта нет в. Хороший продукт начинается от нужд пользователей и видения, как их удовлетворить. Беклог продукта - это результат уточнения и проработки этого видения до конкретных поставляемых частей продукта. Путь от видения до беклога может быть весьма сложным, и появилось множество техник для заполнения этой пустоты. Такие техники, как user story mapping (читайте книгу Jeff Patton, она - просто супер!), lean UX, impact mapping и др. Но не используйте это, как причину для составления слишком детальных планов в начале разработки (big up-front design)! Позвольте беклогу продукта появляться итеративно, как и всё остальное.

По существу, product backlog является списком требований, историй, функциональности, которые упорядочены по степени важности. При этом все требования описаны на понятном для заказчика языке.

Элементы этого списка мы будем называть «историями», *user story*, а иногда элементами backlog’а.

Описание каждой нашей истории включает в себя следующие поля:

- ID – уникальный идентификатор – просто порядковый номер. Применяется для идентификации историй в случае их переименования.
- Название – краткое описание истории. Например, “Просмотр журнала своих транзакций”. Оно должно быть однозначным, чтобы разработчики и product owner (владелец продукта) могли примерно понять, о чем идет речь, и отличить одну историю от другой. Обычно от 2 до 10 слов.
- Важность (Importance) – степень важности данной задачи, по мнению product owner’а. Например, 10. Или 150. Чем больше значение, тем выше важность.

Я предпочитаю не использовать термин “приоритет”, поскольку обычно в этом случае 1 обозначает наивысший приоритет. Это очень неудобно: если впоследствии вы решите, что какая-то история еще более важна, то какой «приоритет» вы тогда ей поставите? Приоритет 0? Приоритет -1?

- Предварительная оценка (initial estimate) – начальная оценка объема работ, необходимого для реализации истории по сравнению с другими историями. Измеряется в story point’ах. Приблизительно соответствует числу “идеальных человеко-дней”.

- Спросите вашу команду: “Если собрать команду из оптимального количества людей, то есть не слишком большую и не слишком маленькую (чаще всего из двух человек), закрыться в комнате с достаточным запасом еды и работать ни на что не отвлекаясь, то, сколько дней тогда понадобится на разработку завершённого, протестированного продукта, готового к демонстрации и релизу? “. Если ответ будет “Для трёх человек, закрытых в комнате, на это потребуется 4 дня”, это значит, что изначальная оценка составляет 12 story point’ов.

- В этом случае важно получить не максимально точные оценки (например, для истории в 2 story point’а потребуется 2 дня), а сделать так, чтобы оценки верно отражали относительную трудоёмкость историй (например, на историю, оцененную в 2 story point’а потребует примерно в два раза меньше работы по сравнению с историей в 4 story point’а).

- Как продемонстрировать (how to demo) – краткое пояснение того, как завершённая задача будет продемонстрирована в конце спринта. По сути, это простой тестовый сценарий типа “Сделайте это, сделайте то – должно получиться то-то”.

- Если у вас практикуется Test Driven Development (разработка через тестирование или кратко TDD) , то это описание может послужить псевдокодом [пр. Переводчика: в программировании специальный способ записи любого алгоритма] для приемочного теста.

- Примечания – любая другая информация: пояснения, ссылки на дополнительные источники информации, и т.д. Обычно она представлена в форме кратких тезисов.

Product backlog (пример)					
ID	Название	Важность	Предварительная оценка	Как продемонстрировать	Примечания
1	Депозит	30	5	Войти в систему, открыть страницу депозита, положить на счет €10, перейти на страницу баланса и проверить, что он увеличился на €10.	Нужна UML диаграмма последовательности. Пока что не стоит беспокоиться про шифрование данных.
2	Просмотр журнала личных транзакций	10	8	Войти в систему; перейти на страницу транзакций; положить деньги на счет; вернуться на страницу транзакций; проверить, что новая транзакция появилась в списке.	Чтобы избежать больших запросов к базе данных, стоит воспользоваться пространственным выводом информации. Дизайн такой же, как и у страницы просмотра пользователей.

Мы экспериментировали и с другими полями, но в итоге именно эти 6 оказались для нас самыми применимыми.

Есть две вещи, которые я почти всего делаю по-другому. Первое: я не использую колонку «важность». Вместо этого, я просто упорядочиваю список. Практически все инструменты работы с беклогами предоставляют возможность перетаскивания элементов (даже Excel и Google Spreadsheet - если вы осмелитесь освоить этот секретный набор супер-инструментов). Так проще и быстрее.

Во-вторых, никаких человеко-дней. Оценки сложности даются в story points или t-shirt sizes (размеры футболок): S/M/L или же оценок нет вообще. Но об этом позже.

Обычно product backlog хранится в Excel таблице с возможностью совместного доступа (несколько пользователей могут редактировать файл одновременно). Хотя официально документ принадлежит product owner'у, мы не запрещаем другим пользователям редактировать его. Ведь разработчикам довольно часто приходится заглядывать в product backlog, чтобы что-то уточнить или изменить оценку предполагаемых трудозатрат.

По этой же причине, мы не помещаем product backlog в систему контроля версий, а храним его на сетевом диске. Этот простейший способ позволяет избежать взаимных блокировок доступа и конфликтов синхронизации изменений.

Однако почти все остальные артефакты хранятся у нас в системе контроля версий.

Excel? Ну и ну, вот это были деньки! Я бы сейчас ни за что не посоветовал использовать Excel для управления беклогом продукта, кроме разве что его облачной версии. Беклог продукта должен жить в виде online-документа, доступного всем для постоянной работы и одновременной модификации. Так, что используйте один из множества специальных инструментов (Trello, LeanKit, Jira), либо же Google Spreadsheet (самый практичный выбор).

---

### Дополнительные поля для user story

---

Иногда мы используем дополнительные поля в product backlog'е. В основном для того, чтобы помочь product owner'у определиться с его приоритетами.

- Категория (track). Например, “панель управления” или “оптимизация”. При помощи этого поля product owner может легко выбрать все пункты категории “оптимизация” и установить им низкий приоритет.