



Профиль

Публикации (1)

Комментарии (7)

Избранное

10 августа 2014 в 12:57

## Scrum — как эффективно работать без project-менеджера

 Agile\*

### Вместо введения

За последние 3 года работы мне довелось работать в самых различных ипостасях: исследователем, разработчиком и руководителем проектов. Есть различные стили управления: западный (когда предоставляется большая свобода в коллективе и многое построено на доверии, уважении, личной организованности отдельного индивидуума) и восточный (когда штрафуются каждое опоздание, жестко фиксируются сроки, во главе угла стоит железная дисциплина коллектива и если человек не справился с поставленными целями — наступает расставание). Руководитель проекта должен сочетать в себе два этих элемента: яблоко и кнут, подпускать людей к себе, чтобы разработчики вам доверяли, но и соблюдать субординацию, так как отношение-отношениями, а нацеленность на результат должна быть всегда.

Но куда важнее: как вы двигаетесь к поставленной цели, как организуете свой рабочий процесс... В этой статье хотелось бы поделиться с достопочтенной публикой одной из наших непрофессиональных видео-лекцией, которую мы снимали для себя. Думаю, в каждом коллективе наступает такой момент, когда что-то может идти не совсем так, как хотелось бы. Хочется каких-то изменений и лучше прежде всего начинать их с себя. Как говорится — если хотите изменить мир, то стоит это начать прежде всего с вас самих же и вашего ближайшего окружения.

Для удобства сделал субтитры к видео, чтобы смотреть было проще. Замечу лишь, что это не профессиональная видео-лекция и лектор нигде эту методологию не читает специально. Дина Насырова (Тим Лидер из Fujitsu) пришла к нам в знак уважения, чтобы помочь наладить процесс работы коллектива и заодно поделилась своим собственным богатым опытом. Встреча прошла год назад — с тех пор много воды утекло. Но спустя время до сих пор вспоминаю ее, так как информация представленная в ней мне очень сильно пригодилась.



### Расшифровка видео лекции полностью (субтитры)

**Дина:** Откуда вообще все пошло. Мы сидели, пили чай и выяснили, что вам нужен project manager. Правильно?

**Команда:** Да!

**Дина:** Вот и я спросила: «А точно нужен, так?». И я обещала прийти и почитать лекцию. Как я себе вижу, что такое Scrum и как вообще можно разрабатывать без начальника. Кто-нибудь из вас знает, что такое Scrum?

**Команда:** Я читал. Ссылку Виктор скинул. Заметки с передовой

**Дина:** Вот, короче, и я здесь, тоже разместила ссылку на эту книжку. С моей точки зрения — это одна из лучших таких книжек, без лишних таких вступлений — зачем этот Scrum, бла-бла-бла. Просто ты берешь, как бы tool, да, и применяешь. Вот этим она и мне нравится. На ее основе я сделала вот такую вот небольшую презентацию. Я сама, конечно, тоже читала Хенри Книберга и вот эту книжку только. Этой книжки реально хватает, чтобы начать. Почему разработка без начальника, да? Потому что — по сути такого понимания как team leader или project manager или кто-то в такой вот роли, который вот стоит наверху, да, и с такой вот палкой нет. И вот это вот, (показывает) делает как это мотивацию, да. Представляете какая будет мотивация да? Покажу на картинке (рисует). Такой сотрудник вот, ослик. Похож на ослика, нет? Как его можно мотивировать? Можно такую вот морковку добавить. Побежит он вперед? Побежит, наверное, да? Можно другим способом мотивировать. Такой волшебный пендель. Побежит вперед? Побежит. Есть еще один вид мотивации, когда такая морковка изнутри его мотивирует. Изнутри человека жжет. С моей точки зрения самая правильная мотивация.

Но это я отвлеклась, на самом деле речь совсем не о том. Речь о том, что если программисты достаточно зрелые, команда сложилась, то эта вот штука сверху с палкой она нужна опционально по сути. И сейчас я буду рассказывать, как можно изнутри команды распределить ответственность таким образом, чтобы и разработка шла как можно быстрее и, соответственно, все были счастливы. Так вот Роберт Челдини говорит про то, кто такой scrum мастер и просто команда. Кстати сколько Вас человек сейчас?

**Команда:** Нас. Ну 9. А программистов из них 5.

**Дина:** 5-6 программистов. Это в принципе команда — самый идеальный вариант для Scrum. Потому что 4 как-то непонятно, да. 4 человека всегда могут договориться. И никакая методология не нужна. А 10 — это уже много. Потому что я по себе знаю. Человека, который вблизи — ты уже не видишь. Не держишь в поле внимания.

Нужно прописать все, до последней буквы как вы будете разрабатывать. Почему — это займет именно столько и сказать — на посмотри — вот заказчик. Будет так работать? Это то, что тебе нужно, да? Заказчик скажет — ну, наверное — это то, что мне нужно, и Вы садитесь разрабатывать. Приходите в свое помещение и разрабатываете. Через полгода приходите и говорите — «на». Поработали, разработали и протестировали. Заказчик говорит — ну клево конечно, но мне это не совсем уж и надо, наверное. Уже не совсем то надо. А Вы говорите — ну посмотри. Мы же с тобой договаривались? Вот что договаривались — то и сделали. Это вот waterfall модель. Сначала собираются требования. Потом Вы значить их approve. Ребята у кого как с английским? Мне стоит переводить? Требования — согласование с заказчиком. Потом значит разработка, потом тестирование и потом все — на этом разработка заканчивается.

При этом, что у заказчика вот в эти полгода происходило? Может у него вообще все изменилось — Вас это вообще не касается? Он заплатил, хороший аванс дал — нормально. Есть такое понятие как Scrum, да? Он зачем? Вот мне кажется в Вашем случае — это оптимальная модель. Потому что — это итеративная разработка. Вот эта вся вот лабуда (показывает), да? Она происходит много-много-много-много раз за жизнь и время разработки. Почему — потому что, вообще-то все может измениться — особенно у Вас. Да, вот почесал репу заказчик, пришел-сказал — блиин, вот я такую клевою штуку нашел. Давайте ее имплементировать. Оценивать вот. И Scrum — он именно для таких команд и для таких вот быстро развивающихся и меняющихся проектов. Когда заказчик может в принципе играть с ценой проекта. Да? То есть он может понимать... сейчас лучше покажу. Начнем да. С чего начинается Scrum? Scrum начинается с того, что Вы создаете Product backlog. Это по сути список требований заказчика. Сажают его перед собой и говорят — давай рассказывай, что тебе нужно. Конечно, в этой книжке по-другому, написано, но — с моей точки зрения требования должны выглядеть именно следующим образом. Пользователь, что-то делает и система как-то реагирует. То есть пользователь заходит и вводит логин-пароль. Система его аутентифицирует или говорит, что логин-пароль не правильный. Вот этот как бы user story. И она должна быть сформулирована именно с точки зрения заказчика — это business story, там не должно быть, что коннектимся в базе данных и использовать такой temple — это не правильно — это про бизнес хотелки, вот с классическим продуктом. Хитро улыбается — громко кричал, что он в команде, а на самом деле (показывает на руководителя проекта). User story. После того как эти user story записаны. Нужно для каждой из них сделать идентификатор стандартный. У каждой из них должно быть имя, именно в таком виде, что пользователь заходит, и система как-то отвечает...Важность, первоначальный estimate — первоначальная оценка, скажем так, да. То есть сколько это примерно может занять, очень примерно. Как продемонстрировать. Пользователь заходит, логиниться и видит результат и всякие заметки разные. И будет это выглядеть примерно так. Это я взяла из книжки. Вот это мне не очень нравится (рисует). Но так тоже можно лучше так — чем никак. Да, что мы делаем, чтобы воссоздать. Сажаем product owner перед собой, выписываем все до единого хотелки. Потом садимся все вместе. Делаем примерную оценку нашей задачки и потом просим его — поставь важность. Важность от 50 до 150, каждый task: какой самый важный, а какой не самый важный. Для чего — для того, чтобы выдирать только самое важное. То есть сначала мы делаем только самое важное. А потом уже все остальное, не важное, а потом может его совсем выкинуть. Понятно? Вопросы есть какие-нибудь? И так мы сформировали product backlog. Потратили несколько дней — выжили все соки из заказчика. Что он нам сделал — проставил важность. Что мы делаем дальше. Нужно бы рассказать про первый спринт. Но я хочу поговорить знаете, о чем? Где должен, как должна сидеть команда — как Вы думаете? То есть может она сидеть где-угодно? По отделам надо рассказывать? Надо сажать всех вместе, убиться, но посадить всех вместе. То есть люди должны видеть друг друга. Scrum мастер в принципе может вообще не приходиться на работу.

**Команда:** А вопрос можно — если я удалено работаю.

**Дина:** Ну с этим тоже можно работать. В книжке кстати есть такой кейс. Ну понятно, что идеального мира не бывает. Но вообще круто было бы, чтобы все друг друга видели. У меня просто был и есть опыт — я вижу, как формируются команды. То есть первая стадия — совсем разные люди. Посадили их вместе — они начинают шутить. Пройдет полгода — они будут кушать вместе ходить. Вот с этого момента и начинается команда. Это влияет и воздействует, к сожалению, надо понимать, что да — хочется работать удаленно, хочется из дома работать

**Команда:** Да, нет просто дело не в этом. Я в другом городе живу. По-другому, не получается.

**Дина:** С этим можно жить, я про идеальный случай, это не смертельно. Где должен сидеть product owner?

**Команда:** На возвышении, наблюдать за всеми.

**Дина:** На самом деле он должен работать с командой каждый день. Он должен быть в шаговой доступности. Но не там же. То есть у него не должно быть возможности прийти и сказать — а что Вы тут делаете? Этого не должно быть. Он где-то рядом. Но не сказать, что вместе с командой сидит в одном и том же месте. И так. Мы сформировали наш product backlog. Вот наши task. То есть, например, занимает 50 важность (часов, да?) Но это не часы — а user story. Что такое user story? Точнее story point. Когда вы начинаете estimate product backlog вы собираетесь вместе и определяете сколько в идеальном мире займет времени реализация вот этого. В идеальном мире, это как только ты пришел в 8 утра, вечером ушел в 6 часов и все это время: кушать не ходил, чай пить не

ходил, общаться не ходил — только работал. Это называется Story Point и эта штука вот так вот оценивается. Это скажем 30, это 10. Это очень важно, это не очень. Ну и так далее. Сейчас нам нужно сформировать такую штуку, которая называется sprint backlog. Это то, что Вы возьмете в первую итерацию. Помните же, что мы ведем итеративную разработку. Одна итерация — это один спринт. Можно варьировать длину спринта, то есть спринт может быть недельный. Он может быть месячный, трехмесячный — это в принципе зависит от команды. Вот, чем хорош длинный спринт — как Вы думаете?

**Команда:** Можно что-нибудь большое замутить.

**Дина:** А чем он плох?

**Команда:** Этот большой может быть слишком криво замучено.

**Дина:** Потому что мы может быть не поняли, что от нас хочет заказчик. Мы творим это целый месяц, а оказалось не то. Ну все правильно — нужно найти для себя самую удобную позицию. Еще бывает, что люди за 2 недели не вработываются — вот он только раскачался — а спринт закончился. Что делать?

**Команда:** А можно задачу сохранить если он не доделал? Пускай дальше будет. Но смотрите да. Для чего вообще Scrum?

**Дина:** Он для того, чтобы после каждой итерации. Был какой-то buildt. Какая-то новая фишка? То, что можно показать. И то что было не доделано — считается, что не работает.

**Команда:** Но же всякое бывает. Человек, например, заболел или просто не успел. Что делать? Наверняка такие кейсы бывали?

**Дина:** Бывают такие кейсы. Бывают, что и люди меняются. Но. Мы говорим о том, что в конце мы должны что-то показать product owner. Хорошо, мы покажем чуть поменьше. Никто не умрет. Другой вопрос — у нас будет некая статистика, что обычно мы выполнили за 1 спринт. 1 спринт, например, у нас занимал -, например, 280 часов. Мы считали, что мы за 200 сделаем. Сделали скажем за 160. А в следующем хотели 180 — а сделали опять 160. Следующий раз мы планируем 160. У нас есть — история. И это же дает нам возможность достаточно серьезно, гораздо серьезно, чем в модели waterfall планировать. Но примерно зная. Что 160, да. Мы делим 160 на 200. И получаем некий фокус-фактор. Это процент, то — как мы идем. Потом на эту цифру мы умножаем в Sprint Backlog. И выясняем, что мы закончим через пару лет. Добавляйте нам разработчиков — примерно так. Так Sprint Backlog и мы берем туда вот эту историю и вот эту историю (показывает). Для нас это как бы начало формирования Sprint Backlog. Это мы делаем на специальной планерке, которая делается в начале каждого спринта. Это называется планирование. Мы все вместе собираемся, мы смотрим на то, какие приоритеты. И берем вот эти вот task. Как Вы думаете product owner надо звать? Он уже все тут проставил. Надо звать? Еще не показываем ничего? На самом деле у меня была возможность попробовать и так, и этак. У нас был product owner, который был совсем не с нами, и он говорил — ну ладно, я буду на демо приходить — что Вы от меня хотите, это и так слишком много. Вот и был такой, который был с нами. Тот, который был с нами реально — мы его просили приходить на этот митинг и на этом митинге вот эти вещи вот пересматриваются (показывает). То есть backlog такие очень примерные. И вот здесь могут они пересматриваться. Например, тут на 50, а 60 на самом деле. И когда он вот это вот видел. И говорил: ребят, мы так не договаривались — почему это так. И начинаешь здесь объяснять, здесь. И он сидит и думает. Может это не сейчас тогда. И отводит, например, эту штуку. Вниз (показывает). Так, что мы делаем еще. На этом же планировании мы определяем цель самого спринта. То есть когда вот эти штуки берем — дело на самом деле очень сложное. Для меня возможно самая сложная штука была. Потому что ты выдираешь все эти user story. Они могут быть с друг другом не очень связаны. все равно должна быть цель. А вот это как раз основная цель спринта.

**Команда:** А как цели на мелкие делить?

**Дина:** Сейчас расскажу. Чуть попозже, хорошо. После того как мы определили цель, мы должны цель, длину спринта, как мы все это покажем через 2 недели, через 3 недели. Потом нужно определить время ежедневного митинга. А вообще бывает у Вас ежедневный митинг?

**Команда:** Планерки раз в неделю бывают. Но они не технического плана

**Дина:** Что такое ежедневный митинг. На самом деле сейчас да, у меня разработки совсем маловато — у меня сервис. Тем не менее мы делаем ежедневный митинг. Я просто не знаю, как Вам объяснить — но это работает. Люди собираются. Самое лучшее, что происходит — люди начинают вариться в своем собственном соку. Каждый из них очень быстро отвечает на несколько вопросов. Что ты сделал? Что ты собираешься делать? И какие проблемы у тебя возникают? Причем с проблемами бывает очень феерично. Когда человек сам в собственном соку варится — он может столкнуться с проблемой, когда кто-то в команде знает, как ее решить. А тут получается, что вопрос обращен к команде — кто-то обязательно поднимает руку и говорит — слушай да ведь это очень просто. И все вертится, серьезно, намного быстрее. И тот эффект, что люди просто не варятся в своем собственном соку. И я часто видела, как одна половина команды после этого митинга собирается и оживленно так

жестикულიрует. Как это в действительности надо сделать. Это очень важный фактор. Просто я настойчиво рекомендую каждый день друг на друга смотреть. И отвечать на эти вопросы. Это достаточно серьезно помогает. И притом программисты они же любят сесть. И ни с кем не общаться. Если у Вас есть новенькие — Вы почувствуете насколько он быстрее войдет, так как он каждый день в лицо Вам смотрит, а не сидит там. Когда лучше делать митинг: с утра, или вечером, или в обед? Кому как

**Команда:** а смысл обсуждать какие-то там проблемы? Если ты за целый день ничего там не смог сделать? А под вечер скажешь — я целый день сидел. Что-нибудь завтра сделаешь.

**Дина:** Мы сейчас объясним. Очень многие говорят, что надо с утра делать. Но мы делаем в обед. Есть минусы, что люди очень тяжело вспоминают, что они делали вчера.

**Команда:** Ну как с утра. Например, за час прийти на работу. И митинг, чтобы мозги разрулились.

**Дина:** Мы лично пришли, что лучше в обед, хотя в книжке написано, что с утра. Так как мои люди приходят в обед.

**Команда:** Утро время растяжимое на самом деле. У нас тоже график немного плавающий

**Дина:** У меня нет желания заставлять приходиться их к восьми. Зачем? Если, например, он придет и 4 часа будет фигней страдать. Я лучше в 12 сделаю. Там есть понятие опоздал, что самое характерное. Но есть люди, которые к 12 опаздывают. Как Вы думаете, что я делаю. Чтобы люди не опаздывали? Что можно сделать, чтобы люди не опаздывали?

**Команда:** Лишить премии его. Тенденция, однако. Морковку показать. Вот помнишь подход сзади. Для особых случаев или поставить банку

**Дина:** На самом деле в книжечке представлен подход, когда они делают общую копилку и человек, который опоздал приходит и молча кладет денежку. И они в конце недели пьянствуют за собранные деньги. У нас это не работает. Спокойно кладет денежки, на следующий день опять опаздывает, на следующий день опять опаздывает. Не знаю, Вам поможет нет. Лично я купила ушки — хрюшки. И целый день надо с этими ушками ходить. 2 раза было достаточно было.

Вот собственно сформировали backlog. Это карты, которые предназначены для покера, известно Вам? (показывает). То есть после того как мы tasks собрали. Надо их заново оценить — так как мы могли ошибиться здесь. Что мы делаем, команда собирается, как Вы. Представьте, что у каждого из Вас колода карт. Мы будем делать линейки, кто-то делает. И каждый молча выкладывает карту. И я, например, смотрю — один выложил сто. Другой выложил час — и у меня возникает вопрос в честь чего так? А почему единица? Ну, например, я знаю, как это делать, так как уже это делал. Заготовки есть. Спрашиваю — а почему 100? Я не знал, как делать. Тогда мы просим того, кто знает объяснить. И делаем еще один тур — все снова выкладывают карты. Второй раз обычно уже одни и те же цифры. Но я беру больше — ну на всякий случай. Так идет планирование — sprint backlog. Еще, что можно сделать. Обычно user story достаточно все-таки поверхностные. Поэтому, в книге написано, что они тоже так делают. Приходим к тому, что эти истории, они дробятся на маленькие tasks. То есть там, например, админку, или в базе данных 3 таблицы. Это бизнес истории. Это истории для программистов. Что делать? И вот такими вот мелкими tasks.

**Команда:** А кто вот устанавливает этот вот самый вес? Тикета и вообще самой истории? Сколько времени?

**Дина:** Вот вы сами сидите и делаете. Никого нет рядом. Есть только scrum master, но он не начальник. Он может вообще со стороны прийти. Вам только помогает наладить процесс. У него больше нет никаких функций. Если кто-то, например, саботирует он этого человека выявляет. Например, тот, кто сидит и говорит — а я не знаю, как это все работает. Звезда такая. Он просто с ним говорит — зачем так делаешь? В принципе начальника нет

**Команда:** Получается это будет работать если только все люди будут мотивированны.

**Дина:** Ну что значит мотивированны?

**Команда:** Ну как я имею в виду что кто-то приходит на работу просто отсидеть, ну есть же такие то же — планктон.

**Дина:** Scrum этих людей не терпит, я говорю, что это же все сразу видно будет. Просто работа сразу встанет. Ну если, например, он вчера говорил, что он делал 3 таблицы, позавчера говорил. Сегодня он опять это говорит. Надо включить мозги и подумать, тот ли он человек. Он что здесь делает. Таблицы? Вот собственно эти мелкие желтые бумажки (показывает). Вот это вот разбитые таски. Таски могут брать совсем разные люди.

**Команда:** Вот все-таки непонятно. Кто разбивает задачи на таски. Программисты сами разбивают задачи на

таски? И оценивают тоже сами, то есть, если не правильно разобьют? Ни тех директор, никто?

**Дина:** А кто кроме Вас — мне тоже интерес этот вопрос? Откуда тех директор знает, что внутри разработки. Кто кроме Вас знает, что внутри разработки? Тех директор показал, что снаружи. Я говорю product owner. Если ты знаешь, что этот таск тебе нужно делать — ты и оценишь его нормально.

**Команда:** А споры приветствуются? Да Вас много. На основании нескольких карточек все складывается? А если получилось, что ты сказал, что сделаешь быстро, а в итоге ты напоролся на косяк какой-нибудь?

**Дина:** А так всегда и бывает. Я же рассказывала про фокус. То есть всегда получается так, что мы говорим 160 story point. А делаем первый спринт никогда больше 80. Хорошая цифра — это когда мы взяли 160 и сделали 120, вообще прекрасно. Люди всегда оценивают больше. Он всегда себя лучше считает

**Команда:** Да и не все подводные камни не всегда видно просто

**Дина:** И я о том же — ну всякое же бывает. Сидишь — думаешь не о том с утра. Там жена накричала. Это ведь тоже случается. Никуда от этого не денешься. Собственно, идем дальше или еще вопросы есть?

**Команда:** Я не пойму до конца. Ну вот кто-то считает. Что задачу можно разбить на 2 подзадачи, другой можно на 5? Кто прав?

**Дина:** Обсуждаете. А там споры приветствуются? Scrum мастер зачем? чтобы Вас немножко вести. Он не будет участвовать в споре. Но если он видит, что спор не в тему, то кто-то там свои амбиции выражает. То в таком случае он этот спор остановит

**Команда:** То есть он как бы все равно получается руководит процессом — и он говорит, что будем 5 подзадач делать точку ставит?

**Дина:** Он руководит процессами. Если у Вас грамотный Scrum мастер, то он никогда не скажет — будем 5 делать. Нет, он просто, ну посмотрит, что люди говорят и решит, что нужно 5 делать или проведет голосование или будет как-то по-другому, или будет как-то разговаривать. Он сам не скажет, что 5. Он не отвечает за результат. Он отвечает, что у Вас процесс хорошо идет. все.

**Команда:** Нет, но он не сам это придумал он на основании мнений это сказал

**Дина:** Это решается по процессу. Если Вы 2 часа уже разговариваете — он Вас остановит. Но он не будет за Вас решать все равно

**Команда:** по типу как управление людьми, процессами

**Дина:** тут в том-то и ценность, что такое человека, который решит за Вас какой таск Вам делать, сколько это времени займет и когда вы будете делать — такого человека просто нет. То есть это некий коллективный разум. Но на самом деле он. Вас призывает достаточно серьезно собраться и к ответственности. Не каждая команда к этому готова. Но так как Вы уже давно работаете вместе — у Вас уже должна быть. То есть ответственность она на Вас полностью лежит. Никто за Вас не будет делать, никто вам не говорил этого. Вы сами все это вынесли наружу и отвечаете за результат

**Команда:** Извините, а можно еще вопрос: вы говорили, что, сначала один сказал там сделал 10 оценку или 100, а product owner должен подождать пока оценки выровняются или он может выбрать?

**Дина:** нет, нужно выровнять оценки. Еще в моей практике было — если спор идет слишком долго, если один говорит 5 часов, другой 100. Личные амбиции. Я говорю — а кто это возьмет? То есть — если ты возьмешь — оставим 5 часов. Это не хорошо делать, так как таски должны образовываться по иному принципу. Но если уже совсем деваться некуда, то оставим вот так вот. Теперь, когда мы сформировали product backlog. У нас есть куча тасков они оценены, мы знаем, что делать, и дальше делаем такую вот штуку. Есть такая вот сетка (рисует)и сюда все это втыкается и после этого происходит все. То есть план боард готов, и мы должны начать, как-то спринт, мы должны начать что-то делать. Здесь есть очень важная вещь — мы все эти таски назначаем. Но мы сразу все таски для людей не назначаем. Мы смотрим наверху самые важные задачи. Внизу не очень важные. Здесь есть задача, которую мы может быть возьмем. Если у нас останется время — скорее всего нет. Вот здесь вот куча тасков (показывает). И каждый должен знать, что сначала нужно брать таск, у которого самый больший приоритет?

**Команда:** А люди сами задачи берут?

**Дина:** Да, в этом и цимус. Каждый человек имеет право взять таск. У меня на самом деле, когда я читала это был самый страшный момент. Вдруг если кто-то скажет, что я это делать не буду. И какой-то таск останется совсем

просто так висеть. Че делать? Реально вообще ни разу за все время так не случилось. Я не знаю, что буду делать? Наверное буду взывать ребят к тому что — команда Вы где? Почему вот так вот происходит? Разве мы не ответственны за результат? Реально ни разу такое не случилось. Видимо, когда команда сложилось и у нее есть ответственность за результат, то возникает понимание, что кто-то должен сделать. Есть еще одна такая фишка здесь. К 5 или 6 спринту. Если достаточно большая команда, то становится понятно, что, один человек постоянно делает одно и то же, потому что он знает хорошо, как это все работает изнутри. То есть он начал делать админку, так и там ковыряется. Он начал делать какой-то большой запрос. А рядом куча маленьких запросов. Вот он это все и делает — потому что я знаю, как это сделать быстрее, да. Тут нужно если есть скрам мастер толковый — он скажет нет, не ты это делаешь, потому что он послезавтра заболит, и никто не знает, что он внутри делал. Лучше это будет дольше, но каждый по немножечко попробует. То есть здесь у скрам мастера есть вето. Почему ты не делаешь — естественно с объяснениями.

**Команда:** Извините, а можно вопрос, мне кажется, что для нас для нашего проекта эта стадия не подойдет. Потому что, например, у нас одни люди занимаются допустим аудио ядром, другие видео ядром и нет смысла их дергать — так как это очень разные области своя специфика есть

**Дина:** панацеи нет на самом деле. Даже с waterfall, как бы я не любила waterfall, да?

**Команда:** Я не про waterfall, я про то не менять людей из-за задач. Он, например, делает кусок, успешно его делает, то пускай делает. Его пока неким заменить, но с другой стороны — Он быстро делает качественно, почему бы и нет?

**Дина:** Но может в действительности это будет не Скрам

**Команда:** Но я просто говорю может нам какой-то гибрид подойдет больше на первое время хотя бы?

**Дина:** Мне тоже так кажется. Вы смотрите сами. Я говорю нет панацеи, нет серебряной пули, всегда нужно что-то менять. Брать что-то и тихонечко под себя что-то адаптировать. Вот если вы считаете, что не нужно людей менять — то это Ваше решение собственно.

**Команда:** Просто мне кажется, что Ваш скрам. Вообще все это от джава разработки. Но не таких вот проектов. У нас гетерогенный проект, грубо говоря. А он такой должен быть: там веб-сайт, допустим делаешь базу данных. И любые люди веб сайты умеют делать и базы данных. Просто вот заказная разработка сделать сайт, например, магазин. Люди делают его и этот скрам идеально подходит. У нас R&D разработка прорывная. Нам не совсем...

**Дина:** Но Скрам как раз и на R&D

**Команда:** Я говорю совсем разные сферы аудио и видео. Вообще совсем разные сферы. Человек может быть в аудио 10 лет работать. Там у него ученая степень. А в видео он ни бельмеса не понимает. Если его так менять через неделю. Это будет просто каша какая-то. И общий вектор будет нулевой прирост. Чтобы между аудио, вот задача. Тоже далеко не монолит может получиться. Не, не, я не говорю глобально, может их просто глобально не перепихивать, более мелкими задачами. Это я согласен если два человека на аудио. Они могут, пускай друг друга дублируют. Но не надо их на видео переключать, так как смысла не будет. Получается такой скрам лимитед. Мы же не ради скрама все это затееваем? Но мне это интересно, чтобы это применять, а чтобы не так просто прослушали и все.

**Дина:** Это можно делать на доске, но мы делали все это на скрам воркс. Есть такая штука. И есть еще джайра. В джайре мне показалось не так удобно. Потому что джайра — это все таки не про это. И потому что в джайре есть — накатываешь такую штуку, которая делает из джайры доску для скрама. Но мне скрам воркс понравился больше. Но это дело вкуса на самом деле. В принципе можно делать все это на доске. Тупо берешь стикеры и лепишь на досочку. На доске это лучше, потому что ты это видишь. Все таки скрам воркс надо открыть. Это же лень, да.

Спринт кончился. Что мы делаем дальше? Мы показываем заказчику и всем кто интересуется этим делом — что мы сделали. Заказчик обязательно, продукт оунер обязательно. Мы показываем ему это все. Вообще хорошо приглашать на демку кого-то со стороны. Потому что сразу такой мандраж — а что мы покажем. Людей это достаточно сильно подхлестывает, что мы можем показать, чтобы это было не стыдно. После того как завершилась демонстрация нужно сделать еще один волшебный митинг. Он называется ретроспектива и идет где-то часа 1,5. То есть мы все вместе собираемся и смотрим наш фокус фактур. Сколько поинтов сделали. Смотрим на наш баклэг и проговариваем что можно было сделать иначе. Что можно было сделать лучше, где мы прокосячили, что мы прокосячили, почему так получилось? И эта штука вот — это возможно самое важное из всех митингов. Как это происходит. Такая вот ретроспектива

Еще я хотела рассказать, не знаю как у Вас. Но когда мы начинаем проект всегда есть технический таск, который тяжело оформить как юзер стори. Это такие вещи как развернуть энвайронмент, сделать некое исследование. Как это будет работать? Как это объяснить продукт оунеру? Но он у Вас вроде толковый. Он, наверное, поймет. У меня лично были проблемы с этим. Я лично пришла к тому, что у

меня первый спринт, всегда технический, просто нужно придать как данность и объяснить-понять, что происходит, попробовать пару вариантов. И всегда есть технические задачи, в которых нужно всегда приучить своего продукт оунера, что есть технический таск и мы его сюда вставляем, что без тестирования жить нельзя. Это мы сейчас сидим и разбираемся с тестированием и это займет столько-то сторипоинтов. Ну и собственно тестирование. У вас есть какое-нибудь не автоматическое тестирование? Не юнит тесты? Что делать с тестированием — есть несколько подходов. Можно сделать тестирование внутри спринта. То есть таск закончен. Отдаешь его тестеру. И он тестирует. А пока тестировать нечего — он сидит и пишет таск кейсы. Мы так пробовали и у нас не получилось. Что мы делали? То есть некоторые разработчики, они тянут все до последнего дня, практически все. А потом на тестера все это падает мертвым грузом да? Извините, что я должна делать? Вот итерации и мы что придумали, не знаю будет у вас это все работать или нет? Разрабатываем, а тестер пока что-то делает? Он вообще не с нами получается. Мы отдаем ему этот билдт и начинаем следующий. При этом какую-то часть этого билда закрашиваем под баги. То есть мы знаем, что в стори поинте будет баг фиксинг. Они прилетят обязательно вот, она пока тестирует, она выкатывает баги, мы их тут же чиним, собственно так и жили. По-другому не получилось. Может у Вас как-то получится?

**Команда:** Сколько процентов вы на баги закладываете?

**Дина:** Сначала больше — потом меньше

**Команда:** Процентом 30-20?

**Дина:** Сначала ближе к 40 даже, а потом качество повышается и уже поменьше

**Команда:** А еще можно такой вопрос? Спросить? А че делать если человек ушел в отпуск на месяц, и он отпадает из этой системы?

**Дина:** То есть, есть же сколько один человек стори поинтов делает в один спринт среднестатистический и вычитаем — получается криво конечно, но хоть как-то. То есть мы брали на это количество стори поинтов, но уже меньше

**Команда:** А по каждому статистика велась индивидуально?

**Дина:** Нет вообще просто. Если вести по каждому — то это будет разброс и шатание. Кто-то скажет, что у меня выработка 50 стори поинтов, а у вас там 10, надо же, на команду

**Команда:** Да согласен. Конкуренция убивает команду да? Да?

**Дина:** К сожалению, это так, на самом деле, что можно делать если 2 команды, то можно вывешивать результаты. Вообще доска наглядно показывает — сколько стори поинтов взяли, сколько сделали, а если 2 доски стоят, то все наглядно видно, видно все, понимаете, что начинается?

**Команда:** Это называется морковку подвесить. Для нас актуально. Можно допустим по видео и аудио. Ну, когда команда потолще станет.

**Дина:** Знаете завод Форда — знаете, что он делал? У него посменная выработка и один раз ночная смена залажала. Он пришел и всех послушал. Покивал и спросил сколько вы сделали за смену? Ему говорят, например, 9 машин. Он взял мел и большим размашистым нарисовал 10 и ушел. Следующая смена пришла и говорит, что это вообще — им объяснили сколько мы сделали штук. Короче на следующий день там было 10, потом 11 — это морковка просто надо уметь. Собственно, все. Я забыла представиться. Меня зовут Дина. Я работаю в Фуджитсу, новичков надо так обучать, вообще-то оно действует и кстати я могу рассказать. Прочитала замечательную книжку Роберта Чалдини вот про морковки, различные конфигурации как раз оттуда.

**Команда:** А еще вопрос — то есть про общую статистику. По каждому программисту допустим? Мотивировать, например, у одного программиста столько-то очков у другого столько-то. Это все очень видно на самом деле

**Дина:** Тут я опять же говорю, что команда должна быть уже достаточно зрелой. И вот в один прекрасный момент у нее должно быть право, что вот этот вот конкретный человек, что каждый раз лагает и что надо другого. Это должно быть возможно, иначе нет прав у тебя — значит нет и ответственности. Вообще существенная экономия получается на продукт менеджере. Все все делают. Можно повысить себе зарплату. Сказать, что мы работаем без продукт менеджера. Если 2-3 спринта пройдут на ура, то можно фактически не приходить

**Команда:** А вот можно вопрос, вот у себя в Фуджитсу вы для скрама какого рода проекты используете?

**Дина:** У меня сервис. Вот такой вот сервис — это поддержка, то есть у меня 60 приложений разнокалиберных и для них нужно, надо чтобы они жили. То есть если они упадут у заказчика будут достаточно серьезные денежные убытки. Поэтому все сделано так. По большей части мы пользуемся вот этой вот штукой (рисует). ITIL — это штука, которая тебе рассказывает, как поддерживать приложения по науке. Есть 1,2,3,4,5 линия поддержки. 1

линия

это СД тест, девочки, которые отвечают за колл центр, потом 2 линия прилетают таски, что сделано — что не работает. На 2 линии сидят по большей части тоже девочки, потому что с 4 линии к ним приходит инструкция. Им приходит тикет, они знают, где его взять и что сделать. Или очень простой таск решить, который не требует там сильных каких-то знаний, тут по большей части инциденты решаются. На третью линию уходит, то, что здесь не решено, либо что-то безумное — инцидент повторяется миллион 20 раз и он уходит сюда = это уже проблема. То есть сюда приходит проблема — почему у нас приложение из раза в раз падает. Сидишь решаешь. Если таск сначала занимает 2-3 часа, то дальше от 5 до 16. Если что-то совсем критическое и не понятно, что делать, то это уходит на 4 линию — обычно это вендор — человек, который все это сделал, который знает, что внутри, что за архитектура. Эти люди, конечно тоже знают, что за архитектура, но у них нет возможности исправить. Сюда приходит обычно. Но если сюда приходит 1000, сюда 200, а сюда приходит 20, вот так это все и работает. То есть ты не держишь, ни кучу одинаковых спецов, которые стоят дорого у тебя здесь 500 девочек, здесь 20 умудренных опытом женщин, здесь 15 мужчин, а здесь такие вот с бородами. Но это собственно о том как это все сделать, каких людей подобрать. Мы пользуемся этой штукой. Но раз мы все это делаем, то нам прилетает разработка. Но она не такая. Она на 3-4-5 месяцев. У нас поддержка 60 проектов. И вся команда все 60 поддерживают. Там на самом деле уже 2 команды. Больше 8 человек не удержишь в внимании. Поэтому уже 2. 12-13 человек. И у каждой есть вот эта вот 4 линия

**Команда:** А давно? Сколько лет они вместе в команде? Они сработанные получается?

**Дина:** Там просто серьезные изменения будут. Мы будем набирать народ. Но сейчас они сработанные. Правда вот здесь команды по 2-3 человека. Мы делаем скрам, но он такой быстрый получается. Очень быстрый получается, все друг друга знают и скрам митинг, например, это сделал, это сделал, ну ладно — молодец, но потому что — они сидят рядом уже давно. Большой скрам у нас бывает, но уже не часто, когда я работала на разработке — было чаще

**Команда:** На разработке, наверное, интереснее скрам? Поддержка — это просто — ты какие-то баги решаешь?

**Дина:** хотелки какие-то выполняешь. Да, да, иногда правда бывает с нуля. Он не настолько прикольный там нет такого, команда друг друга знают и иногда я даже не хожу — они сами. Только доску вижу. Причем мы все в скрам ворксе делаем. Не дозрели до нормальной доски. Может вы дозреете? Может мы ленивые потому что? Это ведь надо двигать же.

**Команда:** а вот там надо общий доступ надо иметь в скрам ворксе? Да?

**Дина:** Это такая вот штука, которая все это организует виртуально. Вообще лучше ее может в первый раз использовать — потому что она тебя ведет — ты можешь ошибиться — а она говорит — теперь таски создай давай. То есть это такое приложение. Он древний, но ничего ведь в скраме не менялось (написала). Он там есть бесплатный — вообще-то он безумно дорогой

**Команда:** Дина, скажите вот, мне не понятно, поинты какие-то, 10 поинтов, как высчитываются?

**Дина:** Как вы считаете? У Вас задачи какие-нибудь есть живые?

**Команда:** Да я не знаю. Вы можете на своих рассказать. Как это вообще соизмеряется с временем? Но мне, например, нужно 2 часа. Я за 2 часа делаю. Все. Точка.

**Дина:** Вот 2 стори поинта. Это идеальное время. Мы просто говорим себе, что они идеальные и вот только я занимаюсь ими.

**Команда:** но я понял чисто сел 2 часа и работаешь — не отвлекаешь и даешь результат. Отключил скайп — да именно так?

**Дина:** в сети конечно встречаются другие определения — у нас это так прижилось. Можно, например, в попугаях мерить. Но это чисто вот переключиться тяжело

**Команда:** почему тогда не ауэрс назвали, а стори поинт? воркинг ауэрс?

**Дина:** хороший вопрос, обязательно его задам Кену Швайберу

**Команда:** просто дело привычки, да?

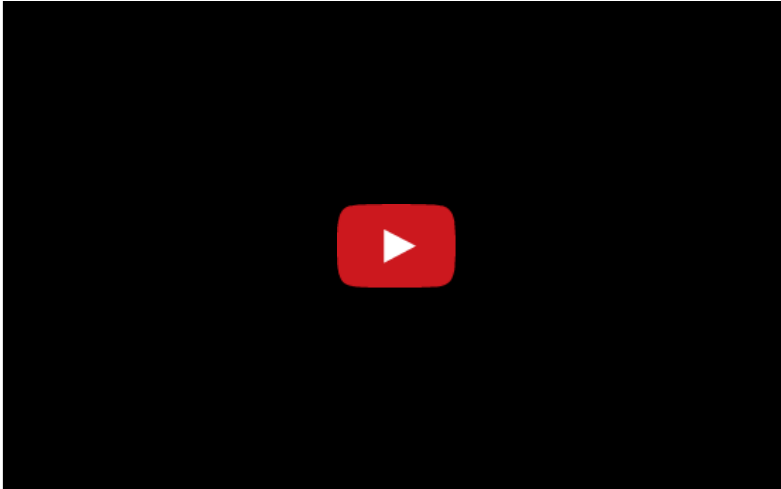
**Дина:** Но я говорю, что в сети есть всякое. Например, у Кена это попугаи. Я сама привыкла — потому что я оцениваю в часах. Сколько времени нужно. Как-то странно да? То есть надо придумать метрику, чтобы все ее поняли и все в ней ориентировались. Поэтому я все-таки предпочитаю идеальные часы

**Команда:** Спасибо! Было очень интересно!



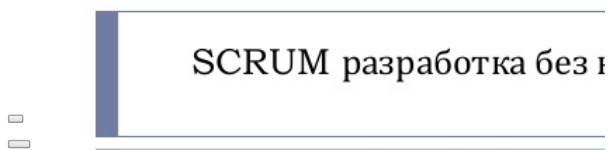
**Дина:** Всегда счастлива!

## Видео-лекция



Лектор — Дина Насырова (Dina Nasyrova), software & solution team leader, FUJITSU Global Delivery Center, Russia. Она разрешила мне использовать видео материалы лекции и презентацию

## Презентация



## Вместо заключения

Представленная лекция вызвала большую дискуссию у нас. И надолго запомнилась у меня в памяти. Некоторые элементы мы стали применять, какие-то — нет. Но речь сейчас не об этом. Дело в том, что мы впервые получили возможность ознакомиться с представленной моделью ведения разработки от практика своего дела. И тем полезнее было услышать мнение эксперта. Применять Scrum или нет, это дело исключительно каждого. Но его наличие невозможно не отрицать. Буду рад, если представленный материал будет кому-то полезен и позволит поменять процесс разработки в новую сторону.

скрам, scrum, agile, team leader, project management, методология, управление проектами, управление командой, управление стартапом



## Комментарии (42)

alexpic 10 августа 2014 в 14:07 #

+2 ↑ ↓

Думаю, из субтитров спокойно можно выкинуть все слова-паразиты, смотреть станет гораздо проще.

RealSpeaker 10 августа 2014 в 15:52 # h ↑

0 ↑ ↓


Окей, сделаем. Пока полностью выложил весь материал в статью. Беседа получилась очень интересной!

 **gandjustas** 10 августа 2014 в 14:37 # +3 ↑ ↓

Мда... пересказ Scrum from trenches в не очень хорошем исполнении.

Как обычно упущены вопросы:

- 1) Как планировать багфикс и то что называется operations. Если учитывать в беклоге, то неясно как это оценить. Если не учитывать, то начинает velocity скакать, причем непредсказуемо. А если будут две команды — dev и ops, то нужен уже ПМ, как координатор действий команд.
- 2) Как приоритезировать PBI. В 90% внедрений скрамов это очень сложная проблема. Если все отдать команде, то начнут делать то что понятнее и/или интереснее. Если отдать все бизнесу, то возникнет то возникает ступор, ибо разные стейкхолдеры считают свои фиши важнее всех других. Нужен опытный product owner, который сможет это разруливать, а его зачастую нет.
- 3) Как выработать правильный definition of done, который с одной стороны не снижает темп, а с другой не позволяет релизить откровенное говно. Часто внедрения скрама начинают буксовать именно из-за этого.

 **RealSpeaker** 10 августа 2014 в 15:56 # 0 ↑ ↓

Спасибо за Ваш комментарий.

Сначала поместил отрывок в статью из видео лекции. Но после Вашего замечания решил дополнить. Спасибо.

1. Бакфиксинг рассматривали (теперь дополнено) — очень важный пункт
2. С приоритетами тоже добавился материал. Как их определять вместе с продукт оунером. И как важно ходить на первые спринты продукт оунеру, чтобы откоррелировать приоритеты.
3. Скрам ради скрама, это не есть хорошо. Надо этого избегать — Вы правы. Панацеи нет как таковой. Можно лишь использовать некоторые полезные инструменты.


Спасибо огромное за ценные замечания!

 **Pavel\_L** 10 августа 2014 в 15:35 # 0 ↑ ↓

Используем Scrum.

Команда 5 человек, все senior-level, разрабатываем ХД и связанные ETL-процессы (Postgres, MS SQL, MySQL, AWS).

Мой опыт: небольшие задачи, бьющиеся на спринты (2 недели), выполняются «на ура». Проблема — с долгосрочным планированием и с задачами, которые на спринты бьются плохо. В первую очередь это касается системной архитектуры: несколько раз уже случалось так, что с таким трудом реализованный функционал приходилось спустя какое-то время полностью переделывать.

 **RealSpeaker** 10 августа 2014 в 15:57 # 0 ↑ ↓

Спасибо. Будет здорово прочитать от Вас практический разбор работы Вашего коллектива со Скрам.

 **Pavel\_L** 13 августа 2014 в 14:48 # 0 ↑ ↓

Боюсь, в ближайшее время не получится — просто времени нет...

 **dbondarev** 10 августа 2014 в 17:24 # 0 ↑ ↓

Эх, говорит, что ретроспектива — самый важный из митингов, но совсем не уделяет ему внимания в рассказе :(

 **RealSpeaker** 10 августа 2014 в 17:44 # 0 ↑ ↓

Да, немного сказано про ретроспективу. Одна это не умоляет важность ее использования.

 **ihs** 10 августа 2014 в 18:29 # +2 ↑ ↓

User Story не совсем о требованиях, а скорее о проблемах пользователей.

Story Points не имеют ничего общего с приоритезацией, потому как приоритезирует Product Owner, а Story Points — удел команды. Помимо всего прочего Story Points **не стоит приравнивать** к человеко-часам.

Scrum — о командной ответственности, так что задачи на Sprint Planning берёт на себя команда, а не отдельные личности. У Скрам-мастера нет блин никакого права вето в принципе.

Много воды в лекции и неточностей в деталях, которые превратят скрам в организации в скрамцею.

 **Conung\_ViC** 10 августа 2014 в 20:27 # 0 ↑ ↓

такой вопрос.

Если стоит задача реализовать фишу с применением технологии, которой раньше никто из команды не пользовался. Почему так — вопрос не рассматриваем. И возникает ситуация, когда изначально простая фиша кажется реализуемой, скажем, за неделю.

Но из-за не знания технологии может начаться этап «исследования» который может затянуться и на 3-4 недели.

Как поступать в скраме с такими задачами?

 **gandjustas** 10 августа 2014 в 20:40 # +1 ↑ ↓

Нужно из исследования выделить задачу, которую можно сделать **и показать** менее чем за неделю. Не забывать про правило числа Пи — сложность изучения новой технологии обычно в Пи раз недооценивается.

 **Conung\_ViC** 10 августа 2014 в 20:45 # 0 ↑ ↓

я столкнулся с такой штукой.

Есть система, и мне надо с ней интегрироваться. В определенные моменты на мой взгляд, она ведет себя совершенно не штатно. Соответственно, можно ломать мозг самостоятельно, можно писать в техподдержку. Как показывает практика, ТП отвечает в среднем через 1-2 суток. Ну и решение проблемы может занимать 4-5 вопросов-ответов с ТП. При этом, при планировании разработки этой фичи по интегрированию, мы даже не предполагали возможность наличия какой-то проблемы.

Это получается задача с просроченным временем выполнения. А как планировать в дальнейшем? Было и наоборот, когда фича, запланированная на неделю, пилилась за полдня...



**gandjustas** 10 августа 2014 в 20:58 # h ↑

+2 ↑ ↓

Не удалось сделать в пределах итерации — записали фейл, посмотрели какое значение velocity получилось, посчитали даты завершения, поняли сколько еще итераций можно потратить на попытки.

Суть гибких методологий в том, что не надо заранее план фиксировать. План фиксируется **только** на одну итерацию, а дальше пересматривается по результатам.



**Conung\_ViC** 10 августа 2014 в 20:59 # h ↑

0 ↑ ↓

хм...

А если при том заказчик требует календарный план проекта с момента подписания договора до старта пром. эксплуатации?

То agile, scrum и прочие слова — не для нас?



**gandjustas** 10 августа 2014 в 21:49 # h ↑

+1 ↑ ↓

И вы реально верите, что если заранее составить план, особенно детальный на длительный срок, то все очно по нему пройдет? «Нет, сынок, это фантастика» (с)

Или есть проблема передоговориться с заказчиком, о том что и когда он получит? Особенно если заказчик уже получает ценность, а не кормится «завтраками».



**Conung\_ViC** 10 августа 2014 в 21:51 # h ↑

0 ↑ ↓

хм. я — нет, не верю.

и вроде умом понимаю, что вы правы, и что при составлении плана на длительный срок предусмотреть все — почти невозможно, но каждый раз при необходимости переноса срока, возникает чувство что облажался по полной...

С другой стороны, этот план иногда является приложением к договору. И особо упорные заказчики, чисто теоретически, за просрочки могут начать взыскивать неустойку.



**gandjustas** 10 августа 2014 в 22:04 # h ↑

+1 ↑ ↓

А не делать план приложением к договору? Этого даже при госконтрактах не требуется. Если заказчик хочет детальный календарный план и штрафы за просрочки, то он пытается закрыть риски. Это значит вы не управляете **рисками заказчика**.

Очень редко бывают случаи когда вас хотят намеренно нагнуть, потому что все понимают, что исполнитель может сделать полное говно, которое даже формальную проверку пройдет, но пользоваться этим будет невозможно. Кроме того менеджер заказчика на таком проекте подпортит свою репутацию. Поэтому всем нужно не формальное закрытие проекта, а получение value.

С помощью гибких подходов вы поставляете value больше и раньше, чем при «водопадном» подходе. В итоге заказчик более доволен, с ним легче договориться. В том числе заключить допник, который снимит уже **ваши риски**.



**Conung\_ViC** 10 августа 2014 в 22:06 # h ↑

0 ↑ ↓

к (сожалению? счастью?) я не продакт-менеджер, и эти вопросы не согласовываю. Для меня это факт, на который я, как программмер, повлиять не могу.

Ну тема, все равно интересная. Спасибо за комментарии и разъяснения.



**ihs** 10 августа 2014 в 22:18 # h ↑


+1 ↑ ↓


Идеология у скрама несколько иная.

На полгода-год планов никаких быть не может, разве что крупными мазками могут быть обозначены хотелки. Единственная более-менее осязаемая часть беклога — нарезанные User story на пару-тройку спринтов. Всё это выливается из-за одного из пунктов Agile манифеста: готовность к изменениям важнее следования первоначальному плану.

В том числе сам подход о найме команд в идеале должен строиться иначе. Стейкхолдер платит не за проект в целом, а платит команде её стоимость в спринт. Тут же и раскрывается смысл итераций —

инкремент продукта, который по завершению спринта и релиза в продакшн повысит прибыль, например. И по факту, стейкхолдеры в праве в любой момент прекратить финансирование команды. При «Водопаде», если на середине разработки стейкхолдеры прекратят финансирование, то у них на руках остаётся не работающее нечто и бесполезное ТЗ, которое даже не приносит понимания: зря или нет они потратили кучу денег на разработку. При скраме же у них есть продукт с конкретным набором фич, который уже давно работает или не работает (как одна из причин, почему они прекратили инвестирование). Они так же могут банально прекратить инвестирование, если посчитают, что стоимость спринта, например, не соизмерима с инкрементом продукта.


 **ihs** 10 августа 2014 в 21:05 (комментарий был изменён) #    +1  

Выбор продукта, с которым необходимо интегрироваться для решения конкретной User story, лежит на плечах команды. Соответственно, если аналогов продукту нет, которые более адекватны и с временем ответа, и стабильностью работы, то есть пара выходов, но все они сводятся к совету  [gandjustas](#).

Во-первых на планировании спринта (а лучше на груминге, задолго до планирования) рассказать Владельцу продукта, что у такой-то User story есть внешняя зависимость (аналогов которой нет, поэтому приходится нам с ней работать), которая может подпортить итог спринта. Он в свою очередь может изменить приоритет, может ему это будет ОК.

Во-вторых всё в ваших руках, и если вы закоммитились на какую-то User Story, то должны стараться сделать всё, чтобы не облажаться командой на демо: это может и чаще долбить суппорт «внешней зависимости» (всей командой по 100500 раз на день, чтобы чаще отвечали), быть может предложить им помощь, если они не способны решать проблемы быстро (в ваших же интересах, чтобы они шевелились быстрее, так как в первую очередь вы потеряете деньги от их неторопливости), возможно реализовать самим то, что предлагает другая контора.

Это, конечно, рассуждения о сферическом коне, ибо по описанию не совсем понятен масштаб бедствия.

 **Conung\_ViC** 10 августа 2014 в 21:06 #    0  

да не, в принципе ваш комментарий вполне в тему и без конкретики. Примерно так и поступаем. Спасибо

 **ihs** 10 августа 2014 в 21:12 #    0  

А что касается

Было и наоборот, когда фича, запланированная на неделю, пилилась за полдня...

Если вы завершаете спринт быстрее и у вас остаётся адекватное количество времени до демо, то вы вполне можете смело подойти к Владельцу продукта и взять из беклога ещё одну-другую User Story на текущий спринт. Это проще, конечно, если беклог «начёсан» на пару спринтов вперёд.

 **opium** 10 августа 2014 в 22:51 #    0  

собственно скрам это агайл то есть гибкая разработка, сделайте спринты поменьше и никто вам не мешает сделать переоценку и обсудить задачу.

 **leorush** 10 августа 2014 в 22:19 (комментарий был изменён) #    0  


Нашел недавно отличный инструмент, позволяющий синхронизировать лекцию на youtube с презентацией [slideshare](#) [presentz.org](#) Возможно пригодится.

 **clockworkbird** 11 августа 2014 в 23:09 #    +1  

Спасибо. Сама статья довольно поверхностная, но такой общий взгляд тоже важен — для понимания в целом. За комментарии отдельное спасибо — много ценного почерпнул.

 **RealSpeaker** 13 августа 2014 в 10:56 #    0  

Вам спасибо и всем, кто принял участие в обсуждении!

 **DeBovuar** 14 августа 2014 в 17:29 #    0  

Последние 3 года для большей часть проектов использую Scrum и могу сказать, что если первоначально применяли его как он описан в Scrum Guide, то со временем он деформировался прочими активностями и практиками PMBoK и ITIL. Поэтому когда читаю успех применения Scrum, появляется комплекс, что это только у моих заказчиков есть бюджет и он заранее хочет знать когда же проект завершится, а еще почему изменений он вроде как не так много и добавил, а проект расширился больше его/ее ожиданий=)

 **clockworkbird** 5 сентября 2014 в 15:24 #    0  

Возникают те же самые вопросы.

В свое время нативно наработали практически тот же scrum, но не хватало учета скорости + с ростом проекта возникало все больше задач по обслуживанию самого проекта (миграция данных, разнос по серверам, наработка инфраструктуры, оптимизация, исследования,...) и в результате нам казалось что мы делаем очень много, а заказчику — что очень мало.

Навскидку scrum гораздо выгоднее разработчикам, для заказчика плюсы в первую очередь в гибкости (поскольку если фиксируется бюджет и сроки, то уже и менять задачи разработчики не дадут) ну и в поточности — можно работать по проекту непрерывно.

Интересно было бы услышать мнение руководителей разработки и заказчиков (со стороны разработчиков процесс понятен).



ihs 6 сентября 2014 в 15:19 # h ↑

0 ↑ ↓

Со стороны Владельца продукта, радеющего за свой продукт, профит в том, что у него есть возможность грамотно расставлять приоритеты Историй под ожидания и желания end-user'ов, условий рынка, каких-то шагов конкурентов под каждый спринт, а не сидеть на жопе и ждать, когда проект, который пилиться 9 месяцев по ТЗ появится наконец-таки на свет, но, например, уже никому не нужный, потому что многие условия/требования изменились и проект в том виде, как он проектировался изначально уже на рынке ничего не стоит.



clockworkbird 8 сентября 2014 в 17:55 # h ↑

0 ↑ ↓

Вы готовы часть недоделанного функционала переносить в следующий апдейт, а текущий при этом оплачивать полностью?

Мне, как разработчику, подход нравится, а вот как к этому относится заказчик — интересно узнать.



ihs 8 сентября 2014 в 19:19 # h ↑

0 ↑ ↓

Если это первый проваленный спринт, то да, почему нет? Если в каком-либо последующем спринте произойдёт такая же ситуация (такие же условия провала), то это уже повод подумать о новой команде, потому как они не оценили, почему они провалили спринт и ничего для этого не сделали, чтобы в будущем этого не повторялось. Для команды и существуют ретроспективы, чтобы оценивать, что понравилось за спринт и что можно улучшить, в том числе проанализировать, почему завалили спринт/конкретную историю (если конечно завалили) и что можно сделать в следующий раз, чтобы дойти до конца спринта «без потерь».

Но это всё разговоры о сферическом коне в вакууме и ситуации бывают разные. У меня есть примеры и где несколько спринтов подряд команда разработки фейлила одну и ту же критически важную фичу для бизнеса и Владельцу продукта при этом было ОК, ибо в основном провалы были из-за внешних зависимостей, о которых и Команда разработки, и Владелец продукта знали заранее до начала спринта, ибо ещё на грумминге они все вылезали.

У них со временем скрам превратился, конечно, в эдакое подобие его и им скорее всего лучше бы подошёл водопад, но они из скрама почерпнули часть полезных идей, как для команды, так и для бизнеса в целом: если вкратце, то это пожалуй умная работа в бэклогом и улучшенная/налаженная коммуникации как внутри команды, так и в целом со своими клиентами.

Но у меня складывается впечатление, что вы пытаетесь всё же совместить водопад со скрамом. Точнее вы как команда разработчиков пытаетесь работать по скраму, а заказчик привык работать с водопадом и вы ему не рассказали правил игры в которую подписались играть. В таком случае вы находитесь в заведомо проигрышной ситуации.



clockworkbird 8 сентября 2014 в 19:43 # h ↑

0 ↑ ↓

Я пока не совмещаю )

У нас чистый водопад, и я пытаюсь понять, что как будет, если перейти на scrum.

В скрам'е все нравится, и вроде бы скрам решает все вопросы, которые сейчас встают.

Но пока все теоретически. Думаю, как наиболее безболезненно «мигрировать», в основном, беспокоит вопрос готовности заказчиков.



ihs 8 сентября 2014 в 21:23 # h ↑

0 ↑ ↓

Хех, прошу прощения за поспешность в выводах. :)

С точки зрения разработки и доносимой ценности Заинтересованным лицам (к которым относится и заказчик) scrum несколько интереснее, да. В том числе демонстрации по окончании каждой итерации позволяют Заинтересованным лицам не находиться в неведении, а на каждой демонстрации понимать на каком этапе вы находитесь и в какую сторону движетесь. Принимая от них же обратную связь можно в том числе будет корректировать путь к достижению цели.

В принципе, многие беды из-за отсутствия нормальной коммуникации между людьми, в том числе в командах разработки. То же непонимание Заказчика, как заинтересованного лица, при решении задач из категории «технического долга» корнем лежат в том, что эти проблемы скорее всего не были объяснены ему с близкой ему экономической точки зрения, чем бизнесу это грозит в будущем и так далее.

Думаю, что если вы способны будете объяснить сильные стороны нового для заказчика подхода для разработки с экономической точки зрения, то у него не будет причин наставлять на чём-то другом. Главное быть честным перед ним и рассказывать всё как есть и в этом случае он будет лучше вовлечён в проект и, возможно, сможет помочь в каких-то вопросах решение которое у него займёт меньше времени, чем у вас, что в итоге ему принесёт профита больше.



clockworkbird 5 сентября 2014 в 13:10 #

0 ↑ ↓

У меня вопрос — как абстрактные Story Points в итоге трансформируются в часы? Получается, что все-таки 1 Story Points имеет фиксированную временную оценку?

 return\_true 5 сентября 2014 в 14:45 # h ↑

0 ↑ ↓

Нет, не имеет. Это просто уровень сложности. Вот мы знаем, что такая-то задача была оценена в N SP. Не важно сколько. Когда появляется новая задача мы просто сравниваем её и понимаем, что скорее всего это тоже N или N + 3. Дальше, через несколько спринтов команда понимает, что в среднем она выполняет K SP. И когда планируется новый спринт можно примерно понимать, сколько задач можно взять на следующий спринт.

Кроме того, оценка может меняться, например, мы замечаем, что у нас все задачи хорошо разбиты и мы оцениваем их в 8, почти всегда. Ну, тогда можно перекалибровать оценки, и сказать, что простая user story может стать... ну, допустим 3 SP. Это если нам хочется более точной оценки. Т.е. нам часы на самом деле не важны. Оцениваем не часы, а сложность. Если хочется трекать время, то можно просто логировать его в задаче. Может быть полезно, для аутсорс-команд, где оплата идёт за часы.

 return\_true 5 сентября 2014 в 14:50 # h ↑

0 ↑ ↓

Кроме того, задача в 1 SP может занять разное время у разных разработчиков. Зависит от опыта, продуктивности, да много от чего зависит. Но «сложность» задачи от этого не поменяется, правда?

 clockworkbird 5 сентября 2014 в 15:16 # h ↑

0 ↑ ↓

Спасибо, большое за ответ.

Вроде все понимал, но традиционный подход заказчиков: «сколько денег и времени?» и частые делдлайны внесли свою лепту.

Правильно ли я понимаю, что время (а точнее выполненный объем в единицу времени) можно сказать только по прошествии нескольких спринтов и оно так же может меняться от раза к разу, а работа идет по принципу «от забора и до обеда» — время вышло, не успели — не важно, решится уже в следующем спринте за доп.оплату?

И еще вопрос — как заказчик в таком случае оценивает эффективность? Как мотивируется команда на увеличение скорости работы? Есть ли какие-то механизмы для этого в scrum или решается внутри каждой команды по своему?

 ihs 6 сентября 2014 в 15:26 # h ↑

0 ↑ ↓

Вы пытаетесь совместить обычный водопад со скрамом, поэтому у вас возникают такие проблемы.

В скраме фиксированная оплата за спринт, фактически — нет никаких стоимостей проектов или чего-то подобного. Заказчик по истечении некоторых спринтов (за каждый из которых вы привнесли ему какую-то Ценность на основе его ожиданий) либо доволен результатами и продолжает с вами колбасить проект, либо нет и ищет других. При этом заказчику очень просто сравнивать ценно ли то, что вы сделали за спринт с тем, сколько он за него отвалил денег.

 return\_true 9 сентября 2014 в 12:37 # h ↑

0 ↑ ↓

Когда я работал в аутсорсе оплата за работу шла по такой схеме:

У нас был продакт оунер со стороны заказчика. Он приезжал в офис на планирование спринта. Он знал, что он от нас хочет на следующий релиз. Мы говорили, сколько из его «хотелок» готовы выполнить. Он выбирает те, которые ему нужны больше.

После спринта и демонстрации он подписывал «приёмку» и релиз уходил на тестирование на стороне заказчика.

Во время спринта мы так же логировали время, потраченное на задачи. По ним и выставлялся счёт. Не очень удобно, но так было проще платящей стороне.

 clockworkbird 9 сентября 2014 в 12:44 # h ↑

0 ↑ ↓

Спасибо, полезно.

Scrum, по сути фреймворк, и каждый, мне кажется, нарабатывает свою схему.

 return\_true 9 сентября 2014 в 13:37 # h ↑

0 ↑ ↓

Так и есть. Все скрамы, которые я видел, отличаются и адаптируются под свои реалии.

Только зарегистрированные пользователи могут оставлять комментарии. [Войдите](#), пожалуйста.