



Профиль

Публикации (2)

Комментарии (73)

Избранное

11 сентября 2008 в 03:00

Панацея ли Scrum — 2!

Управление проектами*

Этот топик, есть продолжение вот этого топика: habrahabr.ru/blogs/pm/39308/

Итак. Продолжим повествование на тему Скрама.

Перед тем, как описать саму методологию, чуть коснёмся классификации. Часто приходится слышать: у нас XP методология, а у нас Скрам, а у нас Agile девелопмент. Всё это имеет право на жизнь. Но, на мой взгляд, желательнее правильно соотносить эти понятия.

XP — это набор принципов и подходов, меняющих традиционное отношение к разработке современного программного продукта.

Agile – это общее название группы гибких и достаточно успешных методик организации ведения проектов. Методик, которые, как правило, опираются на принципы XP и предполагают быструю итеративную разработку, тесное общение с заказчиком и, за счёт тесного общения членов команды, минимизацию создания дополнительных сущностей (артефактов/документов), которые традиционно создавались в процессе работы над проектом. Минимизацию всего того, что создаётся помимо самого программного кода.

Scrum – это одна из конкретных Agile-методик. То есть чёткий шаблон (каркас) организации управления проектом. Со своими терминами и понятиями. Если вы близко следуете именно этому шаблону, то можно сказать, что вы ведёте разработку по Скраму.

Я часто сталкиваюсь с другой классификацией, когда XP называют Agile методологией. Сложно сказать... На мой взгляд, XP — постулирует основополагающие принципы. В один ряд с XP становятся другие концептуальные подходы к разработке: Водопад, Спираль, RUP... А Agile это уже конкретный набор методик, так или иначе описывающих технологические моменты управления проектом. Хотя... это уже терминологические нюансы. Но иметь в виду нужно — термины эти употребляются с разной степенью обобщения, что порой способно вызвать некоторую словесную путаницу.

А сейчас непосредственно о Scrum.

Для начала организации проекта по Скраму, в команду вводится две важные персоны. Это Product Owner и Scrum Master.

Product Owner – человек, который чётко знает, что в конечном итоге нужно заказчику от программного продукта. В идеале, это представитель заказчика. Причём не той части заказчика, которая платит деньги, а той, которая будет непосредственно этим продуктом пользоваться. В реале достичь этого не всегда возможно, поэтому данную роль может выполнять сотрудник компании, который отвечает за контакт с заказчиком. Звонит, ездит, говорит по скайпу и уточняет, выясняет, выдавливает...

В любом случае – именно и только от этого человека исходят все пожелания (требования) к будущему программному продукту. Его слово тут основополагающее и по сути единственное. Это закон.

Scrum Master – очень особенный человек. В его задачу входит сделать так, чтобы в разработке постоянно сопутствовал дух вовлечённости/участия всех членов команды, и при этом не были упущены важные детали, которые должны быть прояснены на скрам-митингах. Скрам митинг является главным местом где принимаются решения по организации проекта.

И что важно – этот человек не является формальным начальником для команды. Но именно он должен правильно организовать обсуждение деталей проекта, “навязав” встречам соответствующий дух, чтобы они НЕ проходили в виде – а теперь пусть отчитается начальник транспортного цеха! Но вся информация должна быть донесена и цели выполнены. Я бы сказал, что этого человека можно назвать тамадой проекта. :-)

А вот теперь, перейдём к тому, как строится непосредственно сам процесс разработки. А точнее процесс управления.

Как мы помним, в XP, на принципы которого и опирается Скрам, разработка должна строиться таким образом, чтобы

идти к цели итеративными витками, на каждом из которых проект приобретает всё более и более законченный вид, и уже с первого витка начинает дышать. Пусть и в виде одной только заставки.

В Скраме, такой единичный временной забег (итерация) называется спринт (Sprint). В последний день (и час) каждого спринта мы должны иметь работающую версию проекта, в которой уже есть на что глянуть. С каждым новым спринтом, мы имеем всё более и более обрастающий мясом (функциональностью) проект. И так, спринт за спринтом, забег за забегом, на выходе получаем готовое изделие. Учитывая, что после каждого забега проект как-то уже работает и, стало быть, мы имеем что показать, то тем самым мы имеем и возможность оперативно получить обратную связь от будущих пользователей и, соответственно, оперативно учесть их замечания. Чуть точнее подстроив будущее изделие под нужды заказчика. Причём эта обратная связь опирается не на представления будущего пользователя о том, как это будет выглядеть в конце, а на том, что он уже смог посмотреть. Что крайне важно в любом проекте. Поскольку увы, представления разных людей могут отличаться очень и очень сильно.

Тут очевиден вопрос – а какой должна быть длительность спринта? Хороший вопрос. Чем меньше – тем лучше. Но – вы должны иметь работающий проект (с новой функциональностью!) после каждого спринта. Подобрать длительность спринта это опыт и искусство. Но, Скрам говорит – эта длительность должна быть не более нескольких недель. Иначе это уже будет не XP, а водопад. :-) Одна, две, три, в крайнем случае — четыре недели. Большой интервал расслабляет заказчика. Маленький интервал? Так вы сделать ничего заметного не успеете. Или не сможете собрать. Такие вот грустные дела.

А теперь — перейдём к механике процесса. Попутно, введём ещё несколько новых понятий. Начинается всё с создания Product Backlog. Я намеренно вначале употребляю англоязычные термины, что бы не запутать читателей, читающих тексты о скрам в английском варианте. По ходу, ненавязчиво, использую разумный перевод на русский.

Итак, Product Backlog – это полный список пожеланий (stories) заказчика, который формирует и ведёт Product Owner (менеджер продукта). Линейный список. Дополняется постоянно. Он может быть создан в excel. Там сконцентрировано всё то, что заказчик хочет видеть в конечном продукте. И все замечания, которые он вносит по ходу работы. А-абсолютно все. Дабы ничего не забыть. Верно, иногда это напоминает: 1. Ремонт сарая (приоритет — 20). 2. Построение коммунизма (приоритет — 6). Но вот так. Скрам не настаивает на том, что бы список был иерархическим, учитывал зависимость пунктов... и так далее. Нет. Не спрашивайте меня почему, но нет! Хотя, по уму. Может быть и да.

Канонически, каждый пункт, должен состоять из нескольких колонок: Номер, Название, ПРИОРИТЕТ, приблизительная оценка срока выполнения, пояснения к пункту и столбец — КАК ПРОДЕМОНСТРИРОВАТЬ РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ. Как вы поняли – приоритет выставляет менеджер продукта. Остальное решает команда.

И теперь – движение.

В начале каждого спринта, собирается собрание. Там вся команда (с продуктовым менеджером и скрам-мастером внутри). Грубо и привычно говоря – митинг. Производственное совещание. Долгий такой митинг. Часа на три. И на нём решается, какие пункты (stories) попадут в реализацию на текущий забег. И формируется второй список – Sprint backlog. То, что попадёт в реализацию на текущий забег из Product backlog. Понятно, что это подмножество Product backlog. Но этого мало. Главное – фиксируется цель забега. Что и как мы сможем показать представителю заказчика на финише данного спринта. И, что самое интересное, делается оценка сроков разработки. Публично и весело. Давая понять менеджеру продуктов, что действительно удастся сделать на данном забеге. А что, действительно не удастся. А дальше?

А дальше — хлопок стартового пистолета и забег. Как вы поняли, это спринтерский забег, а не стайерский. И уж тем более не марафонский. В легкоатлетическом марафоне всё просто – там хотя-бы понятно куда бежать. В разработке софта, траектория забега будет постоянно корректироваться. И даже конечная точка может поменяться.

А что бы всем было понятно, что мы ещё бежим, и, что важно, бежим вместе и в нужном направлении, каждый день проводятся небольшие митинги. Под чутким руководством Скрам мастера. Живые такие производственные совещания на полчаса, где каждый с энтузиазмом рассказывает сколько он пробежал и где споткнулся. Они, как раз, и называются схватками. Что в переводе на английский означает Scrum. Как мужики в регби, в борьбе за мячик. Или, как мужики в американском футболе. В борьбе за тот же мячик. Что, порой, более впечатлительно, хотя и туповато. И не важно, что человек делает в проекте. Локоть товарища он должен чувствовать всегда! В идеале, все локти всех товарищей. Эти схватки лучше утром проводить. Это не сборище уставших в конце дня людей. Это ведь тусовка энергичных спринтеров, остановившихся на двадцатом метре полосы, что бы хлебнуть добротного кофе в хорошей компании.

В конце дистанции каждого спринта– показ. Показ результатов спринта. Всегда. В идеале — прямо заказчику. Ну какой смысл показывать самим себе то, что уже сто раз видели в забеге? Нет. Именно показ кому-то. Ну, на крайняк, руководителю компании. Или начальнику отдела кадров. И обсуждение личных достижений.

А дальше? А всё!

Рывок – результат! Рывок – результат! Осязаемый результат. Осязаемый для тех кто смотрит, а не для самих разработчиков.

Что имеем в сухом остатке? А вот:

1. Формируем Product Backlog
2. Оцениваем сложность проекта и длительность спринта
3. Компонуем Sprint Backlog на забег.
4. Непосредсно забег! (с обсуждениями каждый день)
5. Показ!
6. Дополняем Product Backlog.
7. Компонуем Sprint Backlog на забег.
8. Непосредсно забег! (с обсуждениями каждый день)
9. Показ!
10. Дополняем Product Backlog.
11. Компонуем Sprint Backlog на забег.
12. Непосредсно забег! (с обсуждениями каждый день)
13. Показ!
14. Дополняем Product Backlog.
15. ...
16. ...
- 17.... Ну вы поняли...
18. Получаем производственную премию за успешно сделанный проект.
19. Отлёт в отпуск (или поиск нового места работы).

На самом деле, конечно же — не всё. Скрам, это очень командная методология. Все эти схватки, в процессе которых формируется и работает команда, очень важные штуки. А скрам-мастер должен быть психо-волшебником, способным изошрённо проводить ежедневные схватки таким образом, что бы сплотить и мотивировать команду, выяснить результат работы на текущий день, и тактично скорректировать цели.

Методология Скрам предлагает разные техники для успешного результата. Это и приятное помещение для митингов. Большое. Светлое. С мягкими креслами. И с комнатными растениями. Это и использование большого количество бумаги для стикеров, что бы визуально показать на доске, как и куда мы двигаемся. Предварительно разлинеев доску на лево и право. На менее важное и более важное. Это и тонкие психологические методики оценки времени (planning poker). Команда должна работать крайне напряжённо, но с комфортом и поддержкой друг-друга.

И ещё, Скрам очень аккуратно и внятно настаивает на использовании концептуальных принципов XP. Особенно на test-driving development. Невозможно вести интенсивную разработку и не писать тестов. Да, это трата времени В первый раз. Затем – экономия времени. Особенно после рефакторинга. Это очень важный принцип. Не стоит его недооценивать!

Думаете закончилось пиво? Нифига! Просто решил пока закончить.

Теперь, в русле тактичных и умных комментариев, будем смотреть как бы это правильноЕй поднять критическую часть обсуждения в третьей части эпопеи. :-)

Если ей конечно суждено будет быть.

 scrum, eXtreme programming






Комментарии (25)

 **zarincheg** 11 сентября 2008 в 03:52 #

+2  

Статья хорошая, доступно все описано. Хотелось бы ещё увидеть реальные, практические примеры успешного внедрения данной методики, и если есть такая возможность, то примеры внедрения в различных условиях(ну т.е 5 разработчиков, 10 разработчиков... совмещение различных ролей и тп.).

 **Steamus** 11 сентября 2008 в 04:27 #  

+4  

М-м-м... Я не думаю, что есть много людей способных рассказать это в таком срезе. Термину XP чуть больше десятка лет. Скрам? Скрам, как более-менее очерченный методологический фреймворк (если вообще об этом можно стопроцентно говорить), появился года два-четыре назад. Как часть Agile методики, которая также этим словом была обозвана лет как семь, не более. Учитывая, что мало-мальский серьёзный проект занимает год и более, видимо трудно определённо говорить о большом опыте внедрении этой методологии в столь разных условиях.

Я бы сказал так — тот, кто будет это утверждать, скорее всего лукавит. :-)

**HongKilDong** 11 сентября 2008 в 10:06 # h ↑

+1 ↑ ↓

Интересно, откуда такие данные?

Предлагаю пользоваться википедией — SCRUM-подход к программной разработке впервые был описан более 20 лет назад!!! Кому стало интересно — за подробностями: [en.wikipedia.org/wiki/Scrum_\(development\)#History](http://en.wikipedia.org/wiki/Scrum_(development)#History) (ссылка бьется парсером, для достоверного воспроизведения авторского замысла пользуйте копи-паст :])

В свете этих обстоятельств логично предположить что XP-методология строилась уже от опыта использования SCRUM

**bishop3000** 11 сентября 2008 в 10:48 # h ↑

+2 ↑ ↓

По той же википедии самый важный пункт — это:

«In 2001 Schwaber teamed up with Mike Beedle to write up the method in the book Agile Software Development with SCRUM.»

Именно примерно с 2001-2002 началось массовое увлечение SCRUM, а до этого были только одиночные команды, поэтому нельзя говорить, что SCRUM-у уже 20 лет. 20 лет первому упоминанию, но 5-6 лет началу массового использования.

Это одна из самых молодых методологий сейчас.

**alexlobach** 11 сентября 2008 в 13:47 # h ↑

0 ↑ ↓

Внедрение методологии Scrum в компании «Заказные ИнформСистемы»

**rsultanov** 20 сентября 2008 в 20:16 # h ↑

0 ↑ ↓

Так же очень интересно внедрил ли кто-нибудь подобное для разработки сайтов. У нас в компании используется технология водопада. И я не представляю, как создание сайта можно превратить в несколько параллельных этапов т.к. измененный дизайн надо переверстывать и переподключать. К примеру, программировать функции можно ещё когда дизайн не нарисован, если знать хорошо, что будет представлять новый модуль и когда придет верстка «натянуть» на свой функционал. А вот верстать заранее никак нельзя пока не будет нарисован макет =(.

**dan** 11 сентября 2008 в 06:43 #

+4 ↑ ↓

Отличная статья, спасибо автору!

Как человек, имевший практику scrum, хотел бы добавить несколько пунктов:

*) Scrum Board и Sprint Burndown chart

Очень важно для команды видеть текущее положение дел в спринте. Для этого просто необходимо иметь доску с текущими задачами. И не стоит пренебрегать графиком, чтоб избежать в дальнейшем всяких факапов.

*) Bug fixing

Один из возможных факапов, который практически всегда растягивает спринт. В этом случае имеет смысл выделить отдельную команду (2-3 человека) и ввести для них отдельный спринт длительностью 1 день. Для всеобщей вовлеченности можно менять людей от спринта к спринту между командами девелоперов и баг-фиксеров.

Для такой команды правила scrum могут (и, наверное, должны) быть подкорректированы.

*) wow-фактор

Также может выбить из итерации фактор «начальника». Когда итерация уже запланирована и идет на всех парах, неожиданно появляется «начальник» и со звездами в глазах начинает «впаривать» про то, что Product Team накреативила новую фицу, которую обязательно нужно сделать «вчера». Без нее бизнес встанет, бонусы урежутся, да и вообще всё пойдет к чертям. Такие факапы надо сразу (коллективным разумом) перенаправлять к Product Owner'у, чтобы тот объяснил «начальнику» всю его неправоту вмешательства в спринт. Команда — ответственный организм. Она подписалась под задачами — она их должна выполнить полностью и в срок.

**DaemonI** 11 сентября 2008 в 13:43 # h ↑

0 ↑ ↓

И все-таки я не могу понять, как для багфиксинга можно выделять 1 день. Конечно можно подтянуть 2-3 более-менее серьезных бага, но достаточно запутанные баги таким образом не выловить. Для этого нужно провести в дебагере 3-4 дня. Не будет ли подобный подход к багфиксингу способствовать тому, что достаточно серьезные баги будут кочевать из итерации в итерацию, т.е. из релиза в релиз?

**andry** 11 сентября 2008 в 19:59 # h ↑

0 ↑ ↓

я такие вещи просто в спринт ставлю, в какой — зависит от Product owner-а, зачем отдельный спринт выделять не понятно?!

**bishop3000** 11 сентября 2008 в 11:00 #

+1 ↑ ↓

Автор — молодец, хотя его статьям не хватает немного «взгляда с той стороны», т.к. похоже сам SCRUM он не использовал. Добавлю:

>>Я часто сталкиваюсь с другой классификацией, когда XP называют Agile методологией. Сложно сказать...

XP — безусловно Agile, т.к. проповедует ту же гибкость. Но большой плюс XP — это набор конкретных артефактов, как достичь гибкости и производительности. Минус — мало артефактов, как заставить команду быть командой. А это то, чему много внимания удалено в SCRUM.

>>... Долгий такой митинг. Часа на три...

Не забываем, что SCRUM — это Agile. После первого же митинга на 3 часа все напишут письма, что им такие митинги не нужны и следующий будет уже полчаса.

У нас сейчас daily meeting (ежедневный митинг) 5-15 минут, изредка до получаса, если что-то интересное обсуждаем.

Montly meeting (ежемесячный митинг) — 30-60 минут.

Montly Retrospective (когда команда раз в месяц собирается и обсуждает спринт) — это у нас долго проходит, т.к. мы ходим в кафе и едим и пьем пиво за счет работодателя. И там уже обсуждаем :)

>> Долгий такой митинг. Часа на три. И на нём решается, какие пункты (stories) попадут в реализацию на текущий забег. И формируется второй список – Sprint backlog.

Для оптимизации бэклог создает не вся команда, а только Скрам мастера, продакт оунеры и менеджеры. Остальная команда работает, как обычно и никто на долгий митинг не отвлекается — удобно.

Ну и самое главное в конце:

SCRUM — это чуть ли не единственная методология, которая направлена на развитие команды и постоянную поддержку командного духа. Именно команда и ее мотивация — это главный инструмент SCRUM. Все остальное — это только почва и инструменты для того, чтобы мотивировать команду на работу.

А если команда мотивирована, то она сама всё сделает максимально хорошо и быстро.

 **rgaliull** 11 сентября 2008 в 11:21 #

+1 ↑ ↓

стараюсь использовать именно эту технологию. Не знал что она существует и формализованна. Не пишем код, занимаемся аналитикой и написанием всяких отчетов по экономике. С утра короткий митинг, в 11 обсуждение за кофе. Вечером подведение итогов.

 **puumku** 11 сентября 2008 в 11:27 #

0 ↑ ↓

Большое спасибо за статью. Очень интересно и познавательно.

 **ScorpAL** 11 сентября 2008 в 11:31 #

+6 ↑ ↓

Ну а теперь дам повод покидаться минусами.

На мой взгляд все это большой блеф.

Ребята придумали название для очевидных вещей и теперь зарабатывают на этом деньги.

Вспомните AJAX.

Он был. Он был еще до того как появилось название AJAX.

Однако, когда этой технике присвоили название, появились мастера, тренера, конференции и т.д. и т.п. по этой тематике.

То же самое и с Agile.


Очевидным и самым естественным вещам присваивают название, пиарят это название, выдают за что-то новое.

Вот скоро будет очередная коференция по Agile.

Компании попробуют применить это у себя.

Будут радоваться и кричать «А мы работаем по Agile!». Гордиться этим.

Но никто не хочет понимать что это у вас уже есть. Это наиболее естественный процесс построения работы.


 **yar** 11 сентября 2008 в 12:22 # [h](#) ↑

+2 ↑ ↓

Да, есть такое :)

Про Scrum слышу первый раз, хотя последние года 3 работаю по очень похожей схеме.

С разными командами/конторами/заказчиками по разному, но направление именно «естественный процесс построения работы» :)

 **vybe** 11 сентября 2008 в 13:15 # [h](#) ↑

+3 ↑ ↓

Отчасти с Вами согласен. Но лишь отчасти. Да, все мы умные и все мы и так все знаем и не учите нас работать.=)

Напридумывали тут.=) Да, и раньше у команды был начальник (не важно как его называли) и заказчик, да, и раньше собирались с пивом или на крайняк с кофе побеседовать о тонкостях проекта и что порешать. Но пока это все не приобретает системность и испорченность, получить от подхода максимальную эффективность не получится.

Т.е. пока кто-то главный не скажет «мы работаем по Agile (или другой методологии), а это значит то, то и это..» толку от разработки будет довольно мало.

То же самое, как мне кажется, и в других областях, хотя, наверное, и не во всех.

 **artfh** 11 сентября 2008 в 15:52 # [h](#) ↑


+1 ↑ ↓

Agile/Scrum — естественные и очевидные процессы с точки зрения разработчика. Подумал, сделал, проверил — и так много раз в день. А вот для заказчика/менеджера это принципиально новый подход. И пиар, мастера, тренера, конференции — рассчитаны на них, на то что б изменить их «сознание».

 **scorp13** 11 сентября 2008 в 12:22 #

-1 ↑ ↓

На мой взгляд, XP — постулирует основополагающие прнципы.

 **mrTempl** 11 сентября 2008 в 12:54 #

+1 ↑ ↓

В целом можно устать от таких забегов. Это может подойти если вся команда сангвиники или холлерики, оных просто это не устроит. Либо нужна особая уличная магия для особой мотивации. тут как было видео с конференции и там был пример как команда разрабатывала БД для десткого дома за бесплатно в не рабочее время и с большим энтузиазмом. Для описанного выше это очень понадобится.



vybe 11 сентября 2008 в 13:21 # [h](#) ↑

0 ↑ ↓

Мне кажется, дело в том, что Вы немного не верно поняли слово Спринт. Это не значит, что в этот страшный период все разработчики должны рвать себе попы (простите за мой французский). Все работают спокойно и в нормальном режиме. Добиться такого нормального процесса разработки без нервов и авралов — как раз и задача продукт менеджера. А если ставить нереальные задачи на спринты — это и правда всем очень быстро надоест...



ivoyager 11 сентября 2008 в 13:37 #

0 ↑ ↓

Очень приятная статья, хорошо и живо написана! Задумался о применении подобной методологии в далеких от разработки областях...



shalomman 11 сентября 2008 в 14:13 #

0 ↑ ↓

Интересно, как в Скраме работают с stories, которые занимают больше 1-ого витка разработки (например 5 недель). Очевидно, что должно быть какое-то двухвитковое планирование или типа того...



Castro 11 сентября 2008 в 16:47 # [h](#) ↑

0 ↑ ↓

Важно создать подробный Product Backlog, а большие задачи разбить на субзадачи. Поэтому если нужно написать как-то крупный модуль, но важно иметь список всех шагов к выполнению, оценить их по времени и взять на Ваши 5 недель такое количество, которое предположительно можете успеть. Тут самое тонкое место — это оценка времени, потому что задачи есть разные и разные участники одну и ту же задачу могут сделать потратив разное время. Для этого устраивается во время планирования спринта planning poker, который позволит более-менее достоверно оценить временной промежуток для конкретной задачи. Если важно не брать слишком много задач, а брать столько, сколько сможете реально сделать. То есть использовать принцип «Взятые задачи должны быть сделаны качественно на 100%». Этот принцип позволит выполнять разработку последовательно и не возвращаться к написанному коду в будущем.



legato_di 11 сентября 2008 в 15:12 #

+1 ↑ ↓

Статья прекрасна! Мне понравилась) Хотелось бы увидеть в дальнейшем развитие этой темы, а также может какие-нибудь приемы, позволяющие укрепить команду) Ну и вообще хорошо было бы услышать пример использования этой методологии в жизни.)



andry 11 сентября 2008 в 20:12 #

+1 ↑ ↓

Еще, как мне кажется, важный момент. При планировании, нужно учитывать, что оценки по времени выполнения больше 1-го дня, не есть гут и нужно разбивать на подзадачи. Во-первых избавляемся от студенческого синдрома, во-вторых у любого члена команды, каждый день есть обоснованное ощущение, что все движется в перед.



paurock 13 сентября 2012 в 20:05 #

0 ↑ ↓

Шикарно написано, читается на одном дыхании, спасибо! Хоть нашел наконец, что такое спринт по-русски!))

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