

## Introduction to Scrum

### Введение в Scrum

Recorded by Michael James

Записано Майклом Джеймс (Michael James)

[Existing slide with MJ]

Welcome to Module 1 of CollabNet's Scrum Training Series: Introduction to Scrum.

Разрешите представить, первый модуль обучающей серии о скраме (Scrum) фирмы CollabNet : Введение в SCRUM.

This is a brief introduction to topics that are covered in greater depth in the other modules.

Этот модуль содержит краткое введение в темы, которые подробнее описанные в следующих разделах.

I'm Michael James.

Меня зовут Майкл Джеймс (Michael James).

I help organizations do Scrum and related Agile practices.

Я помогаю организациям внедрять и использовать скрам и сопутствующие гибкие методологии.

[new slide: Thumbnail of Scrum Reference Card, clicks through to online document]

This module is subdivided into three chapters.

Этот модуль разделен на 3 раздела.

At the end you'll find a challenging quiz that's been shown to increase scores on certification tests and may be a prerequisite to your certification class.

В конце модуля будет предложено пройти тест, который поможет вам при сдаче сертификационного экзамена или поможет оценить ваш уровень подготовки.

I also recommend downloading the six-page illustrated Scrum Reference Card.

Я также рекомендую скачать иллюстрированные справочные карты скрам на шести страницах.

[existing slide: Scrum Framework]

Scrum is a framework for dealing with complex work such as new product development. Скрам это платформа для решения сложных задач, таких как разработка нового продукта.

It is an alternative to traditional approaches that were more suited to manufacturing and construction.

Он представляет собой альтернативу для традиционного подхода, который более подходит для производства и строительства.

Why do you need an alternative?

Почему вообще нам понадобился альтернативный подход?

[existing slide: “OLD” assembly line]  
[cue scratchy music from the 1940s or 50s]

In the previous century, most people’s work required more of a focus on execution than innovation.

В прошлом веке большинство людей были вынуждены заниматься выполнением заранее определенной работы, а не разработкой нововведений.

People with defined roles used defined models and defined best practices to execute defined plans that didn’t change very quickly.

Эти люди, в рамках заранее определенных ролей, моделей и практик, выполняли работы по заранее определенным планам, которые менялись достаточно редко.

[new slide: “OLD” assembly line is superimposed on the lower left quadrant of the Stacey diagram, in the “Predictable” zone.]

Life was more predictable because we knew more about *what*  
Жизнь была более предсказуемой, потому что мы знали чего

[slide or animation emphasizes “Requirements”]

we were trying to accomplish, and *how*  
мы пытаемся достигнуть, и как

[slide or animation emphasizes “Technology”]  
we would accomplish it.  
мы будем это делать.

People like Henry Ford used ideas from Frederick Taylor and H.L Gantt to do predictable or repeatable work.

Такие люди как Henry Ford, используя идеи Frederick Taylor и H.L Gantt делали работу людей предсказуемой и повторяемой.

[zoom out to existing slide: Stacey diagram]  
[scratchy old music stops, or changes to modern music, whatever that is]

Today, a lot of the predictable, repeatable work is done much faster, by machines.  
Сегодня большинство этой повторяемой и предсказуемой работы выполняется машинами и гораздо быстрее.

Things change more quickly.  
Все стало меняться быстрее.

To stay competitive, we need to inspect and adapt more quickly, and deal with greater uncertainty.

Чтобы оставаться конкурентоспособными, мы должны адаптироваться быстрее и работать в условиях все большей неопределенности.

A transparent framework imposing timeboxes and feedback loops can help us master uncertainty.

Прозрачная схема работы с применением ограничений по времени и быстрой

реакцией на отзывы клиентов, может помочь нам справиться с работой в условиях такой неопределенности.

[new slide: “Only *learning organizations* will be able to keep up with the future.”]

Only *learning organizations* will be able to keep up with the future.

Только постоянно обучающиеся организации смогут идти в ногу со временем.

[return to existing slide: Stacey diagram]

[new slide: Stacey diagram with framework image in the chaotic zone]

Scrum is a framework for learning about work and the processes we use to do it.

Скрам это основа для изучения и совершенствования той работы и рабочих процессов, которые мы используем.

It’s an attempt to put chaos in a box, making the most of uncertainty.

Это попытка упорядочить хаос максимально используя неопределенность

While it’s mostly been used for developing new software products, it may be useful for other kinds of complex work.

Хотя в основном скрам используется для разработки новых программных продуктов, этот подход может быть использован для других видов сложных работ.

[existing slide: blank circuit diagram]

Scrum introduces feedback loops

Скрам основывается на обратных реакциях

[existing slide: colored circuit diagram]

encouraging us to inspect and adapt the

подталкивая нас к постоянным проверкам и адаптациям

[new slide: circuit + product measuring instrument only]

product that we’re building and the

того продукта, который производим

[existing slide: circuit + both instruments]

processes we’re using to build that product.

и тех процессов, которые мы используем для этого.

[existing slide: balanced Agile Movement]

Scrum is associated with the Agile movement described at <http://agilemanifesto.org/>.

Скрам непосредственно связан с Agile движением описанным <http://agilemanifesto.org/>.

We value:

Наши основные принципы:

....

individuals and interactions over processes and tools,

Люди и их взаимодействие важнее процессов и инструментов,

working software over comprehensive documentation,  
работающий продукт важнее исчерпывающей документации,

customer collaboration over contract negotiation, and  
сотрудничество с заказчиком важнее контракта,

responding to change over following a plan.  
готовность к изменениям важнее следования первоначальному плану.

[existing slide: Agile Movement tips left]  
This doesn't mean that we don't value the things on the right.  
То есть, не отрицая важности того, что справа,

[optional animation: balance teeters]  
We do value the things on the left more.  
мы всё-таки больше ценим то, что слева.

[new slide: Running Tested Features diagram]  
The basic intent of all the Agile approaches (and they are quite similar), is an early and sustainable rate of valuable feature delivery.  
Основная цель всех Agile подходов (все они достаточно похожи) - как можно более ранние и стабильные поставки ценного для клиента продукта.

Early in our product development cycle, we will want to see some features working, rather than one huge release at the end.  
Мы хотим видеть часть требований реализованными, еще на ранней стадии нашего цикла разработки, а не в конце, во время большого выпуска продукта .

Ideally, we'd be able to continue incremental improvements indefinitely.  
В идеале, мы должны быть способны бесконечно продолжать выпускать расширения продукта, на каждом шаге улучшая его.

Product development is ongoing, with frequent releases, rather than a one-shot "project."  
При этом вся разработка продукта ведется частыми релизами, а не одним большим выпуском продукта в конце цикла разработки.

[old scratchy music comes back]  
[existing slide: waterfall diagram]  
Here's what Scrum is NOT:  
На что не похож Scrum :

The attempt to use traditional Gantt ideas for developing software was first described by Dr. Winston Royce in 1970.  
Попытка использовать традиционные идеи Gantt для разработки программного продукта впервые описаны Dr. Winston Royce в 1970 году.

He drew a picture, a bit like this one, showing a series of phases.  
Он представил схему подобную указанной, с последовательностью фаз-переходов.

[cycle through existing slide sequence showing relay-race baton handoffs]

One phase connects to the next phase connects to the next phase and each one has a hand off.

Каждая фаза здесь соединяется со следующей, которая в свою очередь соединяется с последующей, составляя общую последовательность.

In theory, you would do all of your analysis in the beginning, get that completely done, and then you would start on your design, get that completely done, and then start on your code, get that all the way done, and then integrate, etc.

В теории вам следовало бы провести полный анализ, после его окончания перейти к дизайну, выполнить его полностью, и перейти к кодированию, после полного выполнения заняться интеграцией и т.д.

That was a nice theory about how things could work, and we suppose it could work if we had perfect knowledge in the beginning, and never made mistakes along the way. Это была отличная теория, описывающая как должно быть, и мы полагаем это может сработать при условии что мы имеем идеальное знание о том что нам нужно и как этого достичь, в начале разработки, и ни когда не совершаем ошибок при реализации.

The next sentence in Dr Winston Royce's paper (unfortunately, no one read this far), said

Следующее предложение в статье Dr Winston Royce (к сожалению ни кто не дочитывает до этого места) , гласит

[slide: Dr. Royce in the barrel falling down the waterfall]

"I believe in this concept but, the implementation described above is risky and invites failure."

«Я верю в эту концепцию, но внедрение ее рискованно и может повлечь за собой провал.»

[slide sequence: blindfolded team members fail to hand off batons to each other]

The reason it fails for complex work: We rarely have perfect knowledge at the beginning.

Причина такого провала для сложных задач: Мы редко имеем полное представление о задаче в начале работ.

In fact, we know less about our project when we are starting out then we're ever going to know in the future.

Фактически, в начале работ, мы знаем о проекте меньше, чем когда-либо в последующем.

Today is the dumbest day of the rest of our project.

И этот момент один из самых бессмысленных.

But a plan-driven approach requires us to make our most important decisions right at the beginning, when we know the least.

Ведь в соответствии с поэтапным подходом к проекту мы должны сделать наиболее важные решения в самом начале работы, когда знаем о проекте менее всего.

Waterfall projects eventually descend into anarchy, when the actual work finally happens with low quality and too much overtime.

В конце концов это приведет нас к анархии, и когда работа будет выполнена мы увидим что она выполнена с низким качеством и большой переработкой.

[existing slide: ScrumMaster throws waterfall into blender]

Scrum throws all those phases in the blender.

Скрам смешивает все эти этапы.

All the ingredients are mixed into every Sprint.

Каждый этап присутствует в каждом спринте(Sprint).

[existing slide: ScrumMaster with shot glasses]

Instead of dividing the work into activity-specific phases, we divide it into fixed-length iterations, called Sprints.

Вместо разделения всей работы по функциональным этапам работ, мы разделяем весь проект на итерации фиксированной длины, называемые спринт(Sprint).

Every Sprint contains some combination of analysis, design, actual implementation, testing, planning for the future, maybe we'll actually be deploying or shipping more frequently.

Где каждый спринт содержит некоторую комбинацию анализа, дизайна, внедрения, тестирования, планирования, как если бы мы действительно выпускали версии, только гораздо чаще.

[optional future animation: shot glasses turn into rainbow circles, then add the loops on top making Scrum swirls]

[existing slide: Scrum swirl + unicycle, bicycle, tandem bicycle]

Right from the first Sprint, a Scrum team tries to build a *working, tested, potentially shippable* product increment, even if it starts out small.

С первого же спринта, скрам команда пытается выпустить работающую, оттестированную и потенциально готовую к отправке версию, даже если она очень мала.

Every Sprint they demonstrate what little bit of shippable product they have to everyone.

Каждый Sprint они демонстрируют всем часть нового реализованного продукта.

Customers often need to see the wrong product before they can specify what they really need.

Клиенту часто необходимо увидеть хоть какие-то наработки, прежде чем он сможет сформулировать, что действительно ему нужно.

With short iterations and more feedback we have a better chance of discovering the *right* product.

С короткими итерациями и постоянными отзывами клиентов на продукт, у нас есть большой шанс создать именно необходимый нужный продукт.

While the Product Owner doesn't have to actually ship every Sprint, it's the team's job to make it possible.

В то время как владелец продукта (Product Owner) совершенно не обязан делать релиз каждый Sprint, задача команды сделать возможным отправку продукта к концу каждого спринта.

Sprints are officially 30 days long and most Scrum teams are doing them in less, such as two weeks.

Официально спринт ограничивается длительностью в 30 дней, однако большинство команд практикует двух недельные спринты.

[existing slide: Scrum swirl, etc. + cross-functional team]

To make a shippable product every two weeks, we're going to want people with skills in testing, in design, in business requirements, in coding, all on one *cross-functional, selforganizing team*.

Чтобы выпускать готовый продукт каждые две недели нам конечно необходимы специалисты в тестировании, дизайне, кодировании, бизнес аналитике , т.е. все специалисты в одной много-функциональной самоорганизующейся команде.

Now, instead of waiting for the end to start our testing, we'll start testing with our very first Sprint.

Теперь для перехода к тестированию нам нет необходимости ожидать окончания этапа разработки, мы приступаем к тестированию сразу же на нашем первом Sprint.

Instead of doing all the design up front, we'll do a bit of design every Sprint, until it becomes *continuous design*.

И вместо полного дизайна вначале проекта, мы выполняем его понемногу на каждом спринте, получая непрерывное моделирование и разработку.

We'll do small amounts of continuous redesign and refactoring to avoid *technical debt*.

Мы делаем постоянные, не большие шаги по реконструкции и рефакторингу, что бы избежать серьезных технических проблем.

Every Sprint combines all aspects of the work.

При этом каждый спринт содержит все аспекты работы.

A Scrum team should collaborate together instead of working in phases with handoffs.

И Scrum команда непрерывно работает вместе, вместо того что каждому работать отдельно на своем этапе, к которому он приступает, после передачи результатов предыдущего этапа.

[existing slide: Scrum is hard]

Scrum brings challenges to individuals, teams, and organizations.

Конечно скрам это вызов сотрудникам, командам и всей организации.

If it's not disrupting your organization, you're probably not doing Scrum.

Если его внедрение не изменяет всю вашу организацию, возможно скрам вам не нужен.

That's the case most of the time people claim to be doing Scrum.

Это тот случай, когда люди утверждают что занимались скрам.

You usually find that they've overlooked the parts of Scrum that would provide the most benefit because they're too hard to do to.

И вы обнаруживаете, что они игнорировали как раз те части скрам которые дают наибольший эффект, и игнорировали их потому что они сложны в использовании.

The purpose of Scrum's strict rules are to reveal organizational impediments that you can start fixing, if you have the courage and commitment.

Цель строгих правил скрам - выявить все препятствия для скрам внутри организации и начать устранять их, если у вас есть мужество и решимость для этого.

Healing the organizational impediments exposed by trying to do Scrum is more important than "doing Scrum."

Устранение таких, выявленных в ходе внедрения скрам, организационных барьеров важнее чем просто «занятие» скрам.

But this path to mastery starts with learning the framework properly.

Конечно, путь к мастерству начинается с изучения соответствующих практик.

[existing slide: Roles, Meetings, Artifacts]

Scrum provides a structure of roles, meetings, rules and artifacts.

В скрам предусмотрена определенная структура ролей, собраний, правил и артефактов.

Let's do a quick overview of them now.

Давайте быстро взглянем на них.

[existing slide: 3 roles]

There are only 3 roles defined by Scrum and we'll go through each one: Product Owner, Scrum Development Team, and ScrumMaster.

В скрам существует 3 роли : Product Owner (владелец продукта) , Scrum Development Team (Scrum команда разработчиков), and ScrumMaster(скрам мастер). Рассмотрим их подробнее.

[existing slide: Product Owner]

The Product Owner is the single individual responsible for return on investment (or *ROI*) of the product development effort.

Владелец продукта - отдельный сотрудник исключительно ответственный за ROI (окупаемость/возврат инвестиций) всех усилий по разработке продукта.

The Product Owner mostly exerts that influence through the prioritization of the product backlog.

Он влияет на развитие продукта через приоритезацию работ из product backlog (список не завершенных работ по продукту).

The Product Owner is the final arbiter of requirements questions.



Именно владелец продукта конечный арбитр всех вопросов связанных с требованиями.

That doesn't mean that he'll give you all of the detailed requirements up front, or even all the details at the beginning of each Sprint.

Это не значит, что он сразу определит для вас все детальные требования к продукту или определит их к началу Sprint.

But the Product Owner does make the final call about those things.

Но владелец продукта принимает конечное решение по этим вопросам.

[new slide: Product Owner has vision. For example, a thought bubble with a light bulb in it.]

The Product Owner must have the vision behind the product development.

Владелец продукта должен иметь общее видение развития продукта не зависимо от разработчиков.

Given high uncertainty, it often doesn't make sense to have a detailed roadmap, but a vision is important.

С учетом низкой предсказуемости общего процесса, часто не имеет смысл разрабатывать детальную дорожную карту, но общее видение чрезвычайно важно.

[new slide: stakeholders funnel concerns through the Product Owner]

If anyone else wants anything from the team, they need to work through the Product Owner to get what they want.

Если кто-либо желает изменить продукт и повлиять для этого на команду, он должен обращаться и работать с владельцем продукта.

Sometimes we call the Product Owner "the single, wringable neck."

Иногда мы зовем владельца продукта - «единственная скручиваемая шея».

He's the one person making the prioritization decisions for that team.

Он единственный кто может принимать решения по приоритетам задач в рамках проекта.

He makes the business decisions for that team, focused more on the *what* than on the *how*.

Он принимает решения для команды, сосредотачиваясь на том, что надо сделать больше чем на том как это делать.

[existing slide: Scrum Development Team]

Now we have the Scrum Development Team.

Теперь о скрам команде.

The Scrum Development Team is a cross functional group responsible for self Organizing to build a "potentially shippable product increment" every Sprint.

Скрам команда разработчиков это кросс-функциональная самоорганизующаяся группа ответственная за выпуск «потенциально готового к выпуску продукта» в конце каждого спринта.

This is hard to do in the beginning, and today more and more teams are learning how to do it.

Вначале это тяжело организовать, но сегодня все больше команд учатся делать это.

You don't want to be the *last* team that learns how to do this!

И вы наверняка не хотите стать последней командой изучившей этот подход!

Organizations use the word "team" lightly.

Организации достаточно абстрактно используют понятие «команда».

Think back to a time in your life you were on a *real* team, where you could all count on each other.

Вспомните из своей жизни участие в действительно крепкой команде, в которой каждый мог рассчитывать на других и наоборот.

That's the kind of Scrum team I want you to create.

Я бы хотел, что бы создали именно такую команду.

If we keep the same team together in the right environment full time, with effective Retrospectives, Sprint after Sprint, they'll probably get better at working with each other. Если мы сможем организовать стабильную команду, работающую полное рабочее время в правильном окружении, и еще использующую эффективные ретроспективы спринт за спринтом, эта команда, возможно, все лучше и лучше будет работать вместе.

They may become a *learning team*, the building block of a *learning organization*.

Она станет самообучающейся командой, как часть большой самообучающейся организации.

There are no externally imposed hierarchy or job titles on this team.

В такой команде нет привнесенной иерархии или особых должностей.

We leave openings for leadership to emerge naturally, and for control to flow from person to person.

Мы оставляем возможность для естественного проявления лидерства и для общего контроля осуществляемого каждым участником.

[new slide: derive from small\_vs\_large\_team\_illustration]

The Scrum Development Team is a small group of people, ideally between 4 and 9 people.

В идеале скрам команда, это команда состоящая из 4-9 человек.

We have millions of years of practice dealing with groups about the size of a family.

У нас есть миллионы лет практики в управлении командами такого размера - это наши семьи.

Large teams don't self organize effectively until they divide into smaller teams, ideally *feature teams* with minimal interdependencies.

Большие команды не могут эффективно самоорганизовываться, до тех пор пока они не разбиты на небольшие команды, а именно небольшие правильно укомплектованные команды с минимальными взаимосвязями между командами.

[existing slide: Team Room]

Team collaboration emerges most naturally in a team room.

Взаимодействие в команде лучше всего поддерживается, если команда размещена в общей комнате.

[existing slide: planet-sized impediment]

If your organization is spread around the world, you're probably already suffering poor collaboration and coordination; Scrum will quickly bring this problem to the surface.

Если ваша организация распределена по всему миру, вы возможно уже испытываете проблемы с взаимодействием и коммуникациями, скрам еще больше выявит эти проблемы.

You may experience breakthroughs by getting your remote people into one room for one or two Sprints in the beginning, and then every few months after that.

Собрав всю распределенную команду вместе в одну комнату на первые пару спринтов в начале проекта, вы можете получить необходимый толчок в развитии проекта, поддерживая развитие, собираясь вместе раз в несколько месяцев в последующем.

Information sharing tools help with the practical problem of sharing information, but tools by themselves don't transform organizations.

Конечно, инструменты предназначенные для обмена информацией помогают нам, но инструменты сами по себе не могут изменить организацию.

[existing slide: ScrumMaster]

The ScrumMaster is the most misunderstood role in Scrum.

Скрам мастер наименее понятная роль в скраме.

If I'm your ScrumMaster, you're not my ScrumSlave.

Если я ваш скрам мастер это не значит что вы мои скрам рабы.

Its the opposite.

Как раз наоборот.

The ScrumMaster has no management authority over the team.

Скрам мастер не менеджерская должность.

If you are the project manager or line manager of the team, by definition you are not the ScrumMaster. (Sometimes it turns out you're actually a Product Owner.)

Если вы менеджер проекта или руководитель отдела, по определению вы не скрам мастер. (Иногда оказывается что вы владелец продукта.)

Scrum intentionally leaves out the project manager role.

В скраме намеренно не существует роли менеджера проекта.

[existing slide: ScrumMaster + PO + Team]

The responsibilities of project management are split up among the Product Owner and the Team, with the ScrumMaster acting as a kind of facilitator.

Обязанности по управлению проектом разделены между: владельцем продукта, командой и скрам мастером, действующим как консультант и организатор.

[optional new slide: ScrumMaster protects the team from wolves.]

[or return to ScrumMaster slide]

The ScrumMaster protects the team from distractions and interruptions, gets things out of the way of natural team self-organization, removes impediments affecting the team, facilitates the process, helps teach people how to use Scrum, promotes improved engineering practices, enforces timeboxes, provides visibility, and somehow does all this without any management power.

Скрам мастер защищает команду от отвлекающих факторов и прерываний, убирает препятствия для естественной самоорганизации команды и иные препятствия влияющие на команду, помогает общему процессу и обучает использованию скрама, способствует улучшению инженерных практик, обеспечивает соблюдение временных рамок артефактов, обеспечивает общую прозрачность процесса и все это делает без какого либо влияния и понуждения со стороны менеджмента.

How is that possible? It turns out “power” isn’t even the most powerful type of influence. Как такое возможно?

It turns out “power” isn’t even the most powerful type of influence.

Оказывается «власть» не самый мощный тип влияния.

[new slide: illustrate the ScrumMaster checklist]

For further information on the elusive ScrumMaster role, see the “Example ScrumMaster’s Checklist”, an example list of things the ScrumMaster would be concerned with fixing.

Для получения большей информации о «трудно уловимой» роли скрам мастера, воспользуйтесь «Примером контрольного листа скрам мастера», с примерным списком задач о которых должен беспокоится скрам мастер.

[existing slide: artifacts]

The two important artifacts in Scrum are the Product Backlog and the Sprint Backlog. Два важных артефакта скрама - бэклог продукта (Product Backlog) и бэклог спринта (Sprint Backlog).

[existing slide: Product Backlog]

The Product Backlog is a one-dimensional, force ranked list of customer-centric features, prioritized by the Product Owner.

Бэклог продукта это одномерный, специально упорядоченный список клиентских пожеланий, приоритезируемый владельцем продукта.

It’s a list of everything we might ever do.

Это список всего, что мы можем сделать в продукте.

If it’s not in the backlog, it doesn’t exist.

Если чего-то нет в этом списке, его не существует.

Anyone can add items to the Product Backlog, but the Product Owner has got to prioritize them, and the ScrumMaster's got to make it visible.

Всякий может добавить что либо в этот список, но только владелец продукта может его приоритезировать, и скрам мастер должен сделать этот список доступным для всех.

A well formed Product Backlog does not contain tasks, only well-formed Product Backlog Items (or PBIs) which might be written in user story form, or maybe use case scenarios.

Правильно составленный бэклог продукта не содержит конкретных заданий, а только правильно составленные пункты (PBI), в которых описаны краткие пользовательские требования (user story) или сценарии использования.

[existing slide: Force Ranked]

Force ranked means there is only one thing in the top position.

Специально упорядоченный обозначает, что существует только один пункт с максимальным приоритетом.

Getting organizations to do this is usually a breakthrough.

Добиться этого, достижение для организации.

[existing slide: Sprint Backlog]

The Sprint Backlog is what have we committed to do right now to meet our goals for the current Sprint.

Спринт бэклог - это тот список, который мы наметили сделать сейчас, что бы достигнуть целей нашего спринта.

It has an end date.

Он имеет дату окончания работ.

It has the committed Product Backlog Items representing the *what*, and Sprint Tasks representing the *how*.

У него есть согласованные пункты для выполнения (PBI) , которые говорят нам что надо сделать и конкретные задания, которые говорят нам как мы будем это делать.

[existing slide: Meetings]

Now, let's have an overview of the meetings in Scrum.

Теперь давайте посмотрим, какие типы совещаний есть в скраме.

[existing slide: Meetings flowchart]

There are four meetings defined by Scrum and a fifth one that just about everyone has found useful to do.

Четыре типа совещаний проводятся в рамках скрама, и пятый тип который используется многими и можно считать рекомендованным.

The four meetings defined by Scrum: The Sprint Planning Meeting, The Daily Scrum, The Sprint Review Meeting, and the Sprint Retrospective Meeting

Вот совещания предусмотренные в скраме : Планирование спринта, Ежедневный

скрам, Обзор спринта и Ретроспектива спринта.

The fifth one has no official name so we'll call it the Backlog Refinement Meeting.  
Пятый вид совещаний не имеет официального именованя, назовем его  
Переработка бэклога.

[calendar]

Here's an example of how those meetings might fit into a two-week Sprint.  
Вот пример того как эти совещания распределяются в спринте.

I personally like to end the Sprint on a Friday so we don't work over the weekend.  
Лично я предпочитаю заканчивать спринт в пятницу, чтобы не перерабатывать на выходные.

[existing slide: Sprint Planning Meeting]

At the Planning Meeting, the Team and the Product Owner negotiate which items will be committed to the sprint.

Цель совещания Планирование спринта - согласование командой и владельцем продукта списка тем, которые войдут в спринт.

The Team pulls the top priority items from the Product Backlog,  
При этом команда выбирает наиболее приоритетные задачи из бэклог продукта

[existing slide: Sprint Planning Meeting with Sprint Backlog]

commits them to the Sprint Backlog, breaks them into smaller tasks typically, and decides whether it's the right amount of work for them to do and if they're clear about what they are going to do.

помещая их в бэклог спринта, обычно разбивая на более мелкие задания, определяя трудозатраты на выполнение этих заданий, и добиваясь общего четкого понимания выбранных задач.

They plan one Sprint.

Так они планируют спринт.

[existing slide: Daily Scrum]

[@VJ: add 15-minute timer]

During Sprint Execution the Team meets once per day for a 15 minute standup, or Daily Scrum.

В ходе выполнения спринта, команда встречается ежедневно на 15 минутное совещание «на ногах», или Ежедневный скрам.

If we're collaborating we're meeting all the time of course, but this one official meeting is defined where we stand up and report to each other.

Конечно, если мы работаем вместе мы встречаемся постоянно, но это официальное мероприятие со своим местом проведения и формой отчета друг другу.

We report to each other.

Здесь мы отчитываемся именно друг перед другом.

Not to the Scrum Master, not to the Product Owner, not to any kind of boss but to the

Team.

Ни скрам мастеру, ни владельцу продукта, ни какому либо начальнику, а вашей команде.

I report to the six other people that are my team members.

Т.е. я отчитываюсь остальным шести членам нашей команды.

I look them in the eye and I tell them what I did yesterday (“here’s what I did yesterday”), and I tell them what I’m going to do today, and I tell them what impedes me, what are my blockers.

Я глядя в глаза коллегам рассказываю им, что я делал вчера, что собираюсь делать сегодня, что мешает мне выполнить мои задания.

Then I toss the ball to someone else on my team and they give the same report to the rest of the team.

Потом я передаю эстафету и другой участник команды держит такой же отчет перед остальными участниками команды.

[existing slide: Sprint Review Meeting]

At the Sprint Review Meeting, the team demonstrates a potentially shippable product increment to the Product Owner and anyone else who’s interested (sometimes called stakeholders).

На Обзоре спринта команда демонстрирует разработанное обновление продукта, потенциально готовое к отправке клиенту, владельцу продукта и другим заинтересованным лицам (совладельцами).

The Product Owner will declare which items are done, which items did not meet their acceptance criteria.

Владелец продукта описывает, что было сделано, что нет.

We can measure velocity and we’ll get feedback from stakeholders about how we’re doing with the product.

Мы можем измерить скорость разработки и собрать отзывы о продукте от всех заинтересованных лиц.

A lot of times the feedback is “Hey guys you did what you said you’d do, but now that we see it we realize that we need something else. And we couldn’t have known that until we saw the *wrong* product.”

Много раз мы слышали «Ребята вы сделали, то что сказали что сделаете, но теперь, когда мы видим продукт, мы понимаем что нам надо что то иное. И мы не могли этого знать, пока вы не показали нам этот не верный продукт.»

People seem to need a functioning piece of software to react against before they can specify what they really want.

Часто для того что бы понять что действительно хотят люди, им необходимо продемонстрировать часть уже работающего ПО.

[Repeat existing slide: Product feedback loop]

Regular Sprint Review Meetings provide feedback about the product and its emerging requirements.

Регулярные обзоры спринтов позволяют получить отзывы на продукт и способствуют появлению новых требований.

[new slide: Sprint Retrospective Meeting]

Every Sprint ends with a Sprint Retrospective Meeting for the team to inspect and adapt their own process.

Каждый спринт заканчивается ретроспективой, для того чтобы команда проверила и скорректировала свои внутренние процессы.

We inspect and adapt the way we worked together during the last iteration.

Мы инспектируем и адаптируем, как мы работали вместе в ходе последнего спринта.

We typically talk about what went well, what could be improved, what we learned and what still puzzles us.

Обычно мы обсуждаем, что прошло хорошо, что можно улучшить, чему мы научились и что до сих пор вызывает у нас затруднения.

We give feedback to each other.

Мы собираем отзывы друг от друга.

These kinds of things will come up in the Retrospective.

Именно такие вещи происходят на ретроспективе.

Some practical techniques for doing this well are covered in Module 6.

Несколько практических техник для использования описаны в Модуле 6.

This is really the key of the whole thing: the Team eventually takes ownership of their own process.

И это ключевое совещание, т.к. команда становится, в конце концов, хозяином собственных процессов.

[repeat existing slide: feedback loops with instrument measuring “Process”]

Regular Sprint Retrospectives provide feedback about the *process* the team uses to build the product.

Регулярные ретроспективы позволяют получить отзывы о процессах, которые использует команда при разработке продукта.

Remember: Scrum is a framework for learning about products and the processes we use to build them.

Помните: скрам это только основа для изучения цикла продуктов и процессов, которые мы используем для их разработки.

[new slide: Backlog Refinement Meeting]

The fifth meeting isn't given a name officially in Scrum.

Пятое совещание не имеет официального именованя в скраме.

I'm going to call it the “Backlog Refinement Meeting.”

Я собираюсь назвать его - совещание по уточнению бэклога.



It might also be called Backlog Grooming, or maybe we'd do some of this work in a Release Planning Meeting.

Это также можно назвать «стрижка» бэклога, возможно мы сделаем это совещание частью совещания по планирования релиза.

During Backlog Refinement, the Team and the Product Owner get together and they look ahead a little bit into the next few items in the Product Backlog, the items that are candidates for the next couple Sprints.

В ходе такой «стрижки», команда и владелец продукта собираются вместе и рассматривают следующие за текущим спринтом задачи, которые фактически являются кандидатами на включение в последующие пару спринтов.

They clarify them, break the big Product Backlog Items (or *epics*) into small Product Backlog Items (for example: *user stories*) so they could imagine doing a few of them in a Sprint, get some input about prioritization, consider dependencies, etc.

Вместе они рассматривают такие крупные задачи, разбивая их в более мелкие подзадачи (для примера: пользовательские истории), предполагая что некоторые из них войдут в последующие спринты, определяют приоритеты задач, их связи и т.д.

While this work could also be done in the Sprint Planning Meeting, through experience, people have found it preferable to do it in a separate meeting on a different day.

Несмотря на то, что эту работу можно включить в планирование спринта, исходя из опыта, мы видим что люди предпочитают выполнять такую оценку на отдельном совещании в отдельный день.

[new slide: conclusion]

That was a brief overview of Scrum.

Это был краткий обзор скрам.

To learn a little more about Scrum, download the Scrum Reference Card and the ScrumMaster's Checklist.

Чтобы изучить скрам немного больше, скачайте контрольные карточки скрам и чек листы скрам мастера.

To learn a lot more about Scrum, attend one of our Scrum classes, or try our other training modules that go into more depth.

Чтобы узнать существенно больше, присоединяйтесь к одному из наших занятий, или попробуйте другие наши обучающие модули для большего погружения в тему.

To get coaching in Scrum, or tools such as ScrumWorks, TeamForge, or Subversion, give us a call here at CollabNet.

Если вам необходимо обучение скраму или такие инструменты как ScrumWorks, TeamForge, или Subversion, звоните нам в CollabNet.

Also drop us a line to give feedback on this training module.

Также пишите нам, если у вас есть отзывы об этом модуле.