

Пример чек-листа для Скрам-мастеров

An Example Checklist for ScrumMasters (ScrumMasterChecklist.org)

Michael James ([MJ4Scrum@gmail.com](mailto: MJ4Scrum@gmail.com))

Переведено 14 сентября 2007

(обновлено 30 июня 2017)

Фасилитатор со 100% загрузкой?	1
Часть I — Как дела у моего Владельца продукта?	2
Часть II — Как дела у моей команды?	3
Часть III — Как поживают наши инженерные практики?	4
Часть IV — Как идут дела в нашей организации?	5
Заключение	6
Инструкция	6
Карточка организационного препятствия	7

Фасилитатор со 100% загрузкой?

Опытный Скрам-мастер ([Scrum Master](#)) может работать с 2-3 командами одновременно. Если вы готовы ограничить свою роль организацией встреч, контролем отведенного на них времени (timebox), реакцией на препятствия, о которых сотрудники сообщают в явном виде, то сможете справиться с такой ролью, уделяя ей не 100% своего времени. Скорее всего, команда станет эффективнее и при таком объеме участия — ситуация станет лучше, чем была до внедрения Скрама ([Scrum](#)).

Но если вы хотите видеть команду, которая достигает результатов, ранее считавшихся невозможными в трансформируемой компании, и при этом не перегорает на работе, то серьезно обдумайте возможность стать первоклассным Скрам-мастером, а такой Скрам-мастер может работать только с одной командой. Мы рекомендуем на начальном этапе выбирать одного Скрам-мастера на одну команду из 7 человек.

Если вы не можете определить все предстоящие вам задачи, то прислушайтесь к вашему Владельцу продукта ([Product Owner](#)), к вашей команде, обратите внимание на инженерные практики команды и на всю организацию вне вашей команды. И хотя единого рецепта для всех нет, я опишу некоторые моменты, которые многие Скрам-мастера упускают из виду. Пожалуйста, отметьте приведенные ниже строки символами ✓, Δ, ?, N/A в соответствии с описанной на последней странице инструкции.

Часть I — Как дела у моего Владельца продукта?

Скрам-мастер повышает эффективность Владельца продукта, помогая поддерживать Бэклог продукта ([Product Backlog](#)) и План релиза. Обратите внимание, что только Владелец продукта может приоритизировать Бэклог.

- Приоритезирован ли Бэклог продукта в соответствии с его/ее последними соображениями?
- Все ли требования и пожелания к продукту и от всех ли заинтересованных лиц зафиксированы в Бэклоге? Помните: Бэклог постоянно изменяется.
- Имеет ли Бэклог управляемый размер?
 - Чтобы поддерживать контролируемое число элементов (items) в Бэклоге, в верхней его части должны быть мелкие и детальные элементы, а в нижней части — элементы с более общим описанием. Детально анализировать задачи, расположенные *не* в верхней части Бэклога продукта, контрпродуктивно, поскольку в ходе систематических обсуждений продукта с заинтересованными лицами и заказчиками требования к нему будут меняться.
- Может ли каждое требование (особенно из верхней части Бэклога) быть лучше описано в формате пользовательской истории (User story), удовлетворяющей критериям INVEST (независимая, обсуждаемая, ценная, готовая к оценке, небольшая, проверяемая)¹?
- Объяснили ли вы своему Владельцу продукта о техническом долге и как его избегать?
 - Одним из способов является добавление автоматического тестирования и рефакторинга в критерии готовности ([Definition of Done](#), DoD) каждого элемента Бэклога.
- Ведется ли Бэклог в виде *информационного радиатора*, легко доступного для всех заинтересованных лиц?
- Если вы используете инструменты автоматизации управления Бэклогом, то все ли умеют их использовать? Такие инструменты представляют определенную опасность, поскольку без активного участия Скрам-мастера обычно превращаются в *информационные холодильники*.
- Можете ли вы сделать информацию доступнее, демонстрируя распечатки?
- Можете ли вы сделать информацию доступнее, создав большие и понятные диаграммы, графики или схемы?
- Помогли ли вы своему Владельцу продукта организовать элементы Бэклога в соответствующие релизы или группы по приоритетам?
- Все ли участники знают, насколько план релиза соответствует реальности?
 - Вы можете попробовать продемонстрировать всем графики сгорания продукта/релиза (Product/Release Burndown Charts²), после того как элементы Бэклога признаются выполненными (done) в ходе каждого Обзора спринта ([Sprint Review](#)). Графики показывают как скорость реализации элементов Бэклога, так и скорость добавления новых, позволяя отследить изменения состава и сроков проекта.

¹ xp123.com/articles/invest-in-good-stories-and-smart-tasks/

² Mike Cohn, Agile Estimation and Planning, 2005.

- Скорректировал ли Владелец продукта план релиза после последнего Обзора спринта?
 - Лишь небольшое число Владельцев продукта *перепланируют* релиз каждый спринт. Может потребоваться отложить некоторые работы на будущие релизы, поскольку появятся более важные задачи.

Часть II — Как дела у моей команды?

Несмотря на то, что Скрам-мастеру стоит подавать личный пример применения навыков и взаимодействия с членами команды (например, участвовать в парном программировании), довольно легко увязнуть в технических задачах слишком глубоко.³ Поэтому не забывайте, что основными задачами Скрам-мастера по отношению к команде являются:

- Находятся ли члены команды большую часть времени в состоянии *потока*, без разрыва контекста? Некоторые характеристики этого состояния⁴:
 - понятные цели (ожидания и правила ясны, цели достижимы и соответствуют навыкам и умениям исполнителя);
 - фокусировка, высокая степень концентрации на ограниченной зоне внимания;
 - абстрагирование от внешнего мира, объединение действия и его осознания;
 - прямая и моментальная обратная связь (успехи и неудачи в процессе выполнения очевидны, так что поведение может быть скорректировано при необходимости);
 - баланс между способностями и сложностью задачи (задачи не слишком простые и не слишком сложные);
 - чувство личного контроля над ситуацией и текущей деятельностью;
 - задача приносит удовлетворение сама по себе, поэтому ее выполнение дается легко.
- Хорошо ли члены команды относятся друг к другу, встречаются ли вне работы, радуются ли успехам друг друга?
- Поддерживают ли члены команды высокие стандарты при работе друг с другом и поощряют друг друга к профессиональному росту?
- Есть ли проблемы/возможности, которые команда не обсуждает, поскольку в этом случае ее члены чувствуют себя некомфортно?⁵
- Пробовали ли вы различные форматы и места для проведения Ретроспектив спринта ([Sprint Retrospective](#))?⁶
- Фокусируется ли команда на Цели спринта? Возможно, в середине спринта вам следует повторно рассмотреть критерии готовности для элементов Бэклога, взятых в спринт.
- Отражает ли доска задач то, что реально делает команда? Остерегайтесь “темной материи” невыявленных задач и задач длительностью более 1 дня. Задачи, не

³ Предложение переведено в соответствии с уточнениями автора: “A Scrum Master with skills in things like TDD should *sometimes* help the rest of the team learn these skills and habits by pair programming and mob programming with them. But since technical work tends to expand, there's a risk the Scrum Master would neglect the other Scrum Master responsibilities by focusing on the technical work too much.”

⁴ Mihaly Csikszentmihalyi, *Flow: The Psychology of Optimal Experience*, 1990.

⁵ Marshall Rosenberg, *Nonviolent Communication: A Language of Life: Life-Changing Tools for Healthy Relationships*, 2003. Also consider enlisting a professional facilitator who can make uncomfortable conversations more comfortable.

⁶ Derby/Larson, *Agile Retrospectives: Making Good Teams Great*, 2006.

обещанные к выполнению в спринте, являются препятствиями для реализации этих обещаний.

- Состоит ли ваша команда из 3-9 специалистов? В команде достаточно навыков для разработки потенциально готового к релизу продукта?
- Своевременно ли обновляется доска задач команды? Актуальна ли представленная на ней информация?
- Видны ли команде артефакты самоорганизации (доска задач, график сгорания, любые другие)? Удобны ли они для команды?
- Защищены ли эти артефакты от назойливого вмешательства? Избыток внимания извне к ежедневной деятельности команды может мешать ее внутренней прозрачности и самоорганизации.
- Самостоятельно ли члены команды берут себе задачи?
- Отражена ли необходимость выплаты технического долга явным образом в критериях готовности (DoD), которые постепенно делают ваш код более приятным для работы?
- Оставляют ли члены команды свои звания/должности за пределами рабочего места, совместно отвечая за все аспекты работы (тестирование, документирование, пр.)?

Часть III — Как поживают наши инженерные практики?

- Есть ли в разрабатываемой вами системе кнопка «Нажми, чтобы проверить», с помощью которой любой сотрудник вашей или другой команды мог бы легко обнаружить допущенные им ошибки регресса (сломанную функциональность, которая работала до момента внесения изменений)? Обычно это достигается с помощью инструментов xUnit (JUnit, NUnit, пр.).
- Найден ли оптимальный баланс между автоматическим end-to-end тестированием (функциональным тестированием) и автоматическим модульным/компонентным тестированием?
- Использует ли команда для написания тестов язык, совпадающий с языком разработки? Совместная работа не становится легче от использования редких скриптовых языков или средств записи действий пользователя, которыми умеет пользоваться только часть команды.
- Удалось ли команде определить “серую зону”, не покрытую ни компонентными, ни системными тестами?⁷
- Уведомляет ли сервер непрерывной интеграции (CI)⁸ автоматически о появлении ошибки регресса? Может ли время появления такого уведомления снижено до часов или минут? (“Ежедневные билды для слабаков”, — Kent Beck)
- Включены ли результаты всех тестов в результат работы сервера непрерывной интеграции?

⁷ blogs.collab.net/agile/junit-is-not-just-for-unit-testing-anymore and Michael's answer: "I know that many people advocate pure unit tests, e.g. Michael Feather's suggestion that they should return in 10 ms. But with my aerospace background I'm concerned about errors in the larger system made out of the units. Pure "black box" system tests take a long time to run, but I think there's an area between pure unit tests and pure system tests that should be explored."

⁸ [MartinFowler.com/articles/continuousIntegration.html](https://martinfowler.com/articles/continuousIntegration.html)

- Осознали ли члены команды всю прелесть непрерывного проектирования и постоянного рефакторинга⁹ как альтернативу исчерпывающему проектированию до начала разработки (Big Design Up Front)?
 - Рефакторинг имеет четкое определение: изменение внутренней структуры системы без изменения внешнего поведения. Рефакторинг должен происходить несколько раз в час при обнаружении: дублирования кода, сложных логических условий (видны по избыточным отступам и длинным методам), плохих имен идентификаторов, избыточной связности между объектами и т. п. Производить рефакторинг с уверенностью в результате возможно только при покрытии автоматическими тестами. Пренебрежение рефакторингом сделает сложным изменение продукта в будущем, особенно потому, что тяжело найти хорошего разработчика готового работать с плохим кодом.
- Включают ли ваши критерии готовности элементов Бэклога (DoD) полное покрытие автоматизированными тестами и рефакторинг? Чтобы выполнить эту задачу, изучите методы “разработки через тестирование” (TDD).
- Применяется ли парное программирование большую часть времени?
 - Парное программирование существенно повышает степень поддерживаемости кода и снижает количество дефектов. Но этот метод сталкивается с проблемой взаимодействия людей и иногда занимает большее время (особенно если вы измеряете производительность в строках кода, а не в поставляемой функциональности). Попробуйте ввести в своей команде специальные дни парного программирования и вы увидите, что некоторые разработчики предпочитают работать таким образом.

Часть IV — Как идут дела в нашей организации?

- Достаточно ли количество коммуникаций между командами? “Scrum of Scrums” — всего лишь один из способов добиться такого взаимодействия, и не всегда лучший.
10
- Могут ли ваши команды независимо друг от друга реализовывать работающую функциональность даже при пересечении архитектурных границ?¹¹
- Встречаются ли ваши Скрам-мастера друг с другом для устранения организационных препятствий?
- Сообщается ли об организационных проблемах в офис директора по разработке? Может ли быть определена стоимость в долларах, потерянного времени выхода на рынок (time-to-market), уровне снижения качества, нереализованных возможностях клиентов?
 - Но будьте осторожны и не повторяйте ошибку Кена Швабера, которого уволили за то, что он слишком сильно подталкивал организацию к изменениям: “Мертвый Скрам-мастер — бесполезный скрам мастер.”¹²
- Относится ли ваша организация к тем редким организациям, в которых карьерный путь совпадает с общими целями ваших команд? Отвечайте “нет”, если в вашей

⁹ Martin Fowler, Refactoring: Improving the Design of Existing Code, 1999.

¹⁰ See less.works/less/framework/coordination-and-integration.html for alternatives.

¹¹ [FeatureTeamPrimer.org](https://featureteamprimer.org)

¹² Ken Schwaber, Agile Project Management with Scrum, 2004. The “dead ScrumMaster” quote is from a story in Ken’s book. He was fired because he pushed the organization too hard.

организации в карьерном смысле¹³ выгоднее заниматься разработкой или архитектурой, чем тестированием, автоматизацией или документированием.

- Признавала ли отраслевая пресса или другие независимые источники вашу организацию одной из лучших организаций для работы или лидером в вашей индустрии?
- Создаете ли вы “обучающуюся организацию”?

Заключение

Если вы сможете ответить на большинство этих пунктов, и у вас еще останется время до конца дня, то дайте мне знать. Нет готовой формулы для овладения мастерством. Приведенные выше пункты списков могут как помочь в вашей ситуации, так и оказаться бесполезным. После того, как вы начнете понимать, что могли бы сделать для изменения ситуации, вы можете заметить, что боитесь делать это. Это знак, что вы на правильном пути.

Инструкция

Если вы получили этот чек-лист как учебное задание, и ваш текущий (или последний) работодатель пробовал что-либо подобное Скраму, пожалуйста, заполните список в соответствии полученным опытом. Отметьте каждый пункт следующим образом:

- √ если сделано хорошо
- Δ если “может быть улучшено” и “я знаю с чего начать”
- ? если “может быть улучшено, но как?”
- N/A если “не применимо” или “не принесет пользы”

Если ваш текущий (или последний) работодатель не пробовал внедрять Скрам, то заполните список следующим образом:

- √ если “сделано хорошо” или “может быть легко сделано”
- Δ если “будет тяжело, но я знаю с чего начать”
- ? если “будет тяжело, и я не знаю с чего начать”
- N/A если “не применимо” или “не принесет пользы”

Заполнив все пункты, опишите 2-6 организационных препятствий с помощью “Карточки организационного препятствия” (ниже) независимо, взята проблема из этого чек-листа или нет. Осталось выбрать из этих препятствий те, для которых есть хотя бы 1% вероятность успеха.

Чек-лист был впервые переведен 14 сентября 2007, автор неизвестен (сообщите мне, если знаете). В 2017 году чек-лист обновил Дмитрий Блинов (Dmitriy.Blinov@mail.ru, SkillsCup.com), уточняя детали у автора. Ревизию провел Денис Тучин (DenisTuchin@gmail.com).

¹³ Alfie Kohn, Punished By Rewards: The Trouble with Gold Stars, Incentive Plans, A's, Praise, and Other Bribes, 1999.

Карточка организационного препятствия

Проблема внешне выглядит так:

Первопричина (используйте вопрос “Почему?” пять раз):

Влияние на бизнес:

Влияние на эмоции:

Запрос на конкретное действие:

Карточка организационного препятствия

Проблема внешне выглядит так:

Первопричина (используйте вопрос “Почему?” пять раз):

Влияние на бизнес:

Влияние на эмоции:

Запрос на конкретное действие: