

Примерный чек-лист для скрам мастеров (ScrumMasters)

Michael James
(mj4scrum@gmail.com)
14 сентября 2007
(обновлено 24 июля 2012)

Координатор на полный рабочий день?

Компетентный скрам мастер может работать с двумя-тремя командами одновременно. Если вы готовы ограничить свою роль организацией встреч, контролем временных периодов и реакцией на препятствия, о которых в явном виде сообщают сотрудники, вы можете справиться с такой ролью, уделяя ей только часть своего времени. Команда возможно все равно сделает шаг вперед, по сравнению с ситуацией до скрам (Scrum), и ничего катастрофического не произойдет.

Но если вы представляете себе команду, для которой настали великие времена, когда она может выполнить задачи, которые прежде даже не рассматривались, в рамках общей преобразенной компании - задумайтесь над тем чтобы стать Отличным скрам мастером (ScrumMaster).

Такой Отличный скрам мастер может работать только с одной командой.

Мы рекомендуем – один выделенный скрам мастер на команду из семи, или около того, человек, особенного вначале работы.

Если вы все же не можете понять все задачи которые вам необходимо выполнить, прислушайтесь к вашему владельцу продукта (ProductOwner), к вашей команде, обратитесь к инженерным практикам вашей команды, и всей организации вне вашей команды. Поскольку нет общего рецепта, я опишу некоторые вещи, которые как я видел, выпускают из вида скрам мастера. Пожалуйста отметьте √, Δ, ?, или N/A, в соответствии с инструкцией описанной на последней странице.

Часть I - Как дела у моего владельца продукта?

Скрам мастер повышаете эффективность работы владельца продукта, помогая поддерживать бэклог продукта и план релиза. (Обратите внимание что только владелец продукта может приоритизировать бэклог (Backlog).)

- Приоритизирован ли бэклог продукта (Product Backlog) в соответствии с его/ее последними соображениями?
- Все ли требования и пожелания на продукт от всех заинтересованных лиц охвачены бэклогом? Помните бэклог постоянно изменяется.
- Имеет ли бэклог управляемый размер? Что бы поддерживать контролируемое число задач в бэклоге, задачи по мере продвижения к верхней части списка должны быть все более мелкими и подробными, при этом в нижней части остаются задачи с общим описанием. Контрпродуктивно подробно анализировать задачи расположенные не в топе бэклога.

Ваши требования к продукту будут меняться по мере обсуждений владельцев и заказчиков продукта с разработчиками.

- Может ли любое требование (особенно из верхней части бэклога) быть лучше описано как независимая, согласованная, имеющая ценность, готовая к оценке, небольшая и готовая к тестированию пользовательская история (User story)¹?
- Рассказали ли вы своему владельцу продукта о «технической задолженности» и как избегать ее появления? Здесь одним из рецептов является включение автоматического тестирования и рефакторинг в критерий выполнения (DoD) для каждой задачи.
- Доступен ли бэклог в виде информационной доски, информационного «радиатора», непосредственно доступного для всех участников?
- Если вы используете инструменты для автоматизации управления бэклогом, все ли умеют эффективно их использовать? Такие инструменты представляют определенную опасность и без активного участия скрам мастера превращаются в информационный «охладитель».
- Можете ли вы помочь сделать доступнее информацию, демонстрируя распечатки?
- Можете ли вы помочь сделать доступнее информацию, создав большую понятную диаграмму?
- Помогли ли вы своему владельцу продукта организовать задания из бэклога в соответствующие релизы или группы по приоритетам?
- Все ли участники знают соответствует ли еще план релиза реальности? Вы можете попробовать продемонстрировать каждому графики сгорания продукта/релиза (Product/Release Burndown Charts)², после того как задачи признаются выполненными в ходе совещания по обзору спринта (Sprint Review Meeting). Графики показывают как скорость выполнения задач, так и скорость добавления новых, позволяя отследить на ранних этапах тенденции в изменении содержания и сроков проекта.
- Скорректировал ли владелец продукта план релиза после последнего совещания обзора спринта? Небольшое число владельцев продукта, которые отправляют адекватно протестированный продукт во время, перепланируют релиз каждый спринт. Может потребоваться отложить некоторые работы на будущее, поскольку появятся более важные.

Часть II - Как дела у моей команды?

В то время как вас поощряют руководить, постоянно приводя примеры взаимодействия с членами команды в их работе, существует риск того что вы упустите из внимания технические задачи. Рассмотрите следующие основные обязанности по отношению к команде :

- Находятся ли большинство своего времени члены команды в потоке, без разрыва контекста? Некоторые характеристики этого состояния³:
 - Понятные цели (ожидания и правила различимы, цели достижимы и соответствуют навыкам и умениям);

¹ <http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>

- Концентрирование и фокусирование, высокая степень концентрации на ограниченном поле внимания;
- Потеря чувства самосознания, объединение действия и осознания;
- Прямая и моментальная обратная связь (успехи и неудачи в процессе деятельности очевидны, так что поведение может быть скорректировано по мере необходимости);
- Баланс между возможностями и задачами (выполнение не является не слишком легким, ни слишком сложным);
- Чувство личного контроля над общей ситуацией или текущей деятельностью;
- Деятельность действительно приносит удовлетворение, как следствие легкость действий.

- Хорошо ли относятся члены команды друг к другу, развлекаются ли вместе, радуются ли за успехи друг друга?
- Поддерживают ли члены команды при работе друг с другом высокие стандарты и поощряют друг друга к росту профессионализма?
- Есть ли препятствия для работы в команде о которых не хотят обсуждать члены команды, поскольку они слишком не удобные?⁴
- Пробовали ли вы различные форматы и места для проведения ретроспектив по спринту (Sprint Retrospective Meetings) ?⁵
- Держит ли команда в фокусе цели спринта? Возможно вам следует провести в середине спринта сверку и пересмотреть критерии готовности для задач из бэклога, назначенных на спринт.
- Отражает ли доска задач то что реально делает команда? Остерегайтесь «темной материи» не раскрытых задач и задач длительностью более одного дня. Задачи не относящиеся к целям спринта являются препятствиями к достижению этих целей.
- Состоит ли ваша команда из 3-9 специалистов, с достаточными уровнями квалификации для разработки продукта потенциально готового к релизу?
- Обновляется ли ваша доска задач регулярно?
- Доступны ли для команды инструменты самоменеджмента (доска задач, график сгорания задач, ..), удобно ли ими пользоваться ?
- Защищены ли такие инструменты от назойливого вмешательства? Избыток внимания извне команды к ежедневной деятельности сотрудников может препятствовать внутреннему общению и самоменеджменту.
- Вызываются ли члены команды самостоятельно для решения задач?
- Включены ли задачи связанные с техническим долгом в задачи бэклога, которые постепенно делают ваш код приятным для работы?

² Mike Cohn, Agile Estimation and Planning. (2005).

³ Mihaly Csikszentmihalyi, Flow: The Psychology of Optimal Experience(1990).

⁴ Kerry Patterson, Crucial Conversations: Tools for Talking When Stakes are High (2002). Also consider enlisting a professional facilitator who can make uncomfortable conversations more comfortable.

⁵ Derby/Larson Agile Retrospectives: Making Good Teams Great (2006)

- Оставляют ли члены команды свои должности за пределами командной комнаты, коллективно отвечая за все аспекты согласованной работы (тестирование, документирование и т.д.)?

Часть III - Как поживают ваши инженерные практики?

- Есть в вашей системе для разработки кнопка «нажми что бы проверить», с помощью которой каждый из вашей или другой команды мог бы просто обнаружить допущенные ошибки регресса (сломанную функциональность работавшую до изменений)? Обычно это достигается с помощью xUnit (JUnit, NUnit, и т.п.).
- Существует ли оптимальный баланс между автоматическим системным полным тестированием (функциональным тестирование) и автоматическим юнит тестирование?
- Использует ли команда при системном и юнит тестировании язык совпадающий с языком разработки? Совместная работа не становится лучше от использования проприетарных скриптовых языков или инструментов записи действий, которыми только часть команды умеет пользоваться.
- Затрагивает ли команда «серую зону» между системным тестированием и юнит тестированием?⁶
- Существует ли на сервере непрерывной интеграции⁷ автоматическое уведомление в случае появления ошибки регресса? Может ли скорость появления такого уведомления снижена до часов или минут? («Ежедневные билды для слабаков» Kent Beck)
- Входят ли все тесты в результат работы сервера непрерывной интеграции?
- Поняли ли члены команды всю прелесть непрерывного дизайна и постоянного рефакторинга⁸, как альтернативу Большого дизайна До Разработки? Рефакторинг имеет четкое определение: изменение внутренней структуры системы без изменения внешнего поведения. Рефакторинг должен случаться несколько раз в час, где бы не было дублирование кода, сложная условная логика (видная по избыточным отступам и длинным методам), плохое именование идентификаторов, избыточная связь между объектами и т.п.. Выполнять рефакторинг с уверенностью возможно только при покрытии автоматическими тестами. Пренебрежение рефакторингом делает сложным изменение продукта в будущем, особенно с тех пор как стало тяжело найти хорошего разработчика готового работать с плохим кодом.
- Включают ли ваши критерии выполнения задачи для каждого функционального задания из бэклога полное автоматизированное покрытие тестами и рефакторинг? Изучение методов разработки через тестирование (TDD) увеличит возможность достижения такой задачи.

⁶ <http://blogs.collab.net/agile/2007/03/07/junit-is-not-just-for-unit-testing-anymore/>

⁷ <http://www.martinfowler.com/articles/continuousIntegration.html>

⁸ Martin Fowler, Refactoring: Improving the Design of Existing Code (1999)

- Применяется ли парное программирование при большинстве разработок? Парное программирование радикально повышает уровень пригодности кода к сопровождению и уменьшает количество багов. Но этот метод сталкивается с проблемой взаимодействия людей и иногда занимает большее время (если вы измеряете производительность в строчках кода чаще чем в рабочей функциональности). Введите для примера дни парного программирования. И наверняка некоторые разработчики предпочтут работать таким образом.

Часть IV -Как дела в вашей организации?

- Достаточное ли количество меж группового взаимодействия в вашей организации? Скрам скрама (Scrum of Scrums) единственный путь добиться такого взаимодействия, и не обязательно самый лучший.
- Могут ли ваши команды независимо друг от друга реализовывать работающую функциональность, даже при пересечении архитектурных границ?⁹
- Встречаются ли ваши скрам мастера друг с другом, для устранения возможных препятствий на уровне организации?
- Когда это необходимо передаются ли организационные проблемы в офис директора по разработке? Может ли быть определена цена в долларах потерь от позднего выхода на рынок, потерь от падения качества, потерь клиентов от нереализованных возможностей? (Но учтите на ошибках Ken Schwaber : «Мертвый скрам мастер бесполезный скрам мастер»¹⁰)
- Относится ли ваша организация к тем редким организациям где карьерный путь совпадает с общими целями ваших команд? Отвечайте «нет» если в вашей организации в карьерном смысле¹¹ выгоднее заниматься разработкой или архитектурой, чем тестированием, автоматизацией или документированием.
- Признавала ли отраслевая пресса или другие независимые источники вашу организацию как одну из лучших организаций для работы, или лидером в вашей индустрии?
- Создаете ли вы из вашей организации «обучающуюся организацию»?

⁹ <http://FeatureTeamPrimer.org/>

¹⁰ Ken Schwaber, Agile Project Management with Scrum (2004)

¹¹ Alfie Kohn, Punished By Rewards: The Trouble with Gold Stars, Incentive Plans, A's, Praise, and Other Bribes(1999)

Заключение

Если вы сможете ответить на большинство этих пунктов и у вас еще останется время до конца дня, дайте мне знать.

Нет готовой формулы для овладения мастерством. Этот список может как помочь вам в вашей ситуации, так и быть бесполезным.

После того, как вы начинаете понимать, что вы могли бы сделать, чтобы изменить всю ситуацию, вы поймете, что вы боитесь делать это. Это знак вам - вы на правильном пути.

Карточка организационной проблемы

Проявление проблемы :

Первопричина (Используй «Почему?» пять раз) :

Влияние на бизнес:

Эмоциональное воздействие:

Запрос на конкретное действие:

Карточка организационной проблемы

Проявление проблемы :

Первопричина (Используй «Почему?» пять раз) :

Влияние на бизнес:

Эмоциональное воздействие:

Запрос на конкретное действие:

Инструкция

Если вы получили этот чек-лист как учебное задание и ваш текущий (или последний) работодатель пробовал что либо подобное скраму, пожалуйста заполните так как вы это видели в вашей организации. Отметьте каждый пункт следующим образом :

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

Или если ваш текущий работодатель (или самый последний) не пробовал внедрение скрам , заполните следующим образом :

- ✓ (для “сделано хорошо” или «может быть просто сделано»)
- Δ (для “может быть проблемно и я знаю с чего начать”)
- ? (для “может быть проблемно я не знаю с чего начать”)
- N/A (для “не применимо” или “не принесет пользы”)

Когда вы заполните все пункты, опишите 2-6 препятствий с помощью Карточки организационной проблемы, независимо взята ли эта проблема из этого чек листа или нет. Осталось выбрать те в которых у вас есть хотя бы 1% надежды на успех.