

РАЗ — НАКЛОН, ДВА — ПОКЛОН

**&must
read**

Гибкий подход к управлению позволяет постоянно подгонять долгоиграющий проект к требованиям рынка, но взамен требует культурных изменений в самой компании.



ПЕРЕВОРАЧИВАЕТ с ног на голову классические проектные ограничения и при этом работает; меньше формализации и больше непосредственного

общения и доверия — все это о гибком подходе к ведению проектов. Пока наибольшее распространение agile-управление (так его называют) получило в сфере разработки программного обеспечения. Но кто знает, куда эта практика переключается дальше. Ее гибкость и ориентация команды разработчиков на результат — мечта любого заказчика, ведь требования рынка сейчас меняются постоянно. С такой же регулярностью могут вноситься и изменения в проект, конечно, при условии, что управление построено по принципу agile. Его суть заключается в том, что команда, работающая над проектом, вполне способна к самоорганизации — она сама может выбрать лучший путь к достижению результата. В таком случае менеджер проекта превращается скорее в помощника, который убирает преграды на пути команды. Заказчик в свою очередь получает возможность постоянно корректировать свой запрос. Изменилась ситуация — изменились вводные от заказчика. В теории все просто. О том, как работает этот подход на практике и при каких условиях приносит максимальную пользу, «&» рассказал **Станислав Спиваков**, технический директор *Крайтек Украина*, компании, занимающейся разработкой компьютерных игр, в которой уже несколько лет гибко ведется работа над проектами.

ЛЕГКОСТЬ В ДЕТАЛЯХ

Agile — это название семейства облегченных подходов к работе над проектами. Сформировалось это семейство в конце 1990-х в ответ на излишнюю громоздкость и формализацию водопадного подхода к управлению. Agile не содержит руководства к действиям и конкретных практик, это скорее принципы и ценности, которые

Уоллак. Фото: О. Березко «А».



При гибком подходе к управлению в начале работы над проектом часто никто точно не знает, что в итоге должно получиться

ЦЕННОСТИ, отраженные в agile-манифесте разработчика ПО

- Личности и их взаимодействие важнее, чем процессы и инструменты.
- Работоспособное ПО важнее, чем обширная документация.
- Сотрудничество с заказчиком важнее, чем переговоры по контрактам.
- Умение реагировать на изменения важнее, чем следование плану.
- Удовлетворение клиента за счет ранней и бесперебойной поставки ценного ПО.
- Приветствуются изменения требований даже на конечных этапах разработки. Это может повысить конкурентоспособность продукта.
- Частая поставка рабочего ПО (каждый месяц, неделю или еще чаще).
- Тесное ежедневное общение заказчика с разработчиками на протяжении всего проекта.
- Проект строится вокруг команды мотивированных личностей, которые обеспечены нужными условиями работы, поддержкой и доверием.
- Наиболее эффективный метод передачи информации – личная беседа.
- Работающее ПО – лучший измеритель прогресса.
- Спонсоры, разработчики и пользователи должны иметь возможность неопределенно долго поддерживать постоянный темп работы.
- Постоянное внимание совершенствованию технического мастерства и дизайна.
- Простота – искусство, не делать лишней работы – необходимость.
- Команда регулярно размышляет над тем, как работать более эффективно, и адаптирует свое поведение к изменяющимся обстоятельствам.

ми руководствуются успешные команды. Наиболее распространенная из методологий получила название scrum. Термин взят из регби – в игре так называют момент, когда все члены команды собираются вместе. Собственно, по-настоящему командная работа и является ядром гибкого подхода к управлению проектами. Его приверженцы уверены в том, что строгое следование изначальному плану работы не обязательно обеспечит проекту успех. Наоборот, своевременное реагирование на изменение ситуации и следующий за этим выход за изначально заданные рамки может привести к победе. По сути, проект разбивается на множество коротких этапов (итераций), занимающих две-четыре недели. Каждую такую итерацию можно назвать проектом в миниатюре: у нее есть свои приоритеты, свои задачи, свой хоть и небольшой, но план. Завершенная итерация обеспечивает прирост функциональности конечного продукта. Кроме того, после реализации каждого такого небольшого этапа в начальный план действий могут вноситься некоторые изменения. Необходимо это, к примеру, если изменились требования рынка или появилась новая информация о продуктах конкурентов.

Agile-подход может работать как внутри отдельно взятой команды, так и в крупной организации. Для начала разберемся с командой. Итак, группа профессионалов получает входящий набор задач от заказчика. Каждой задаче присваивается свой уровень важности. Уровень определяет заказчик, команда же, в свою очередь, предлагает сроки, в которые она может выполнить то или иное задание. Как только задание на ближайшую итерацию определено, команда самоорганизуется для его выполнения. То, как именно будет достигнут результат, целиком зависит от людей. Потому изначально принимается, что каждый член команды – профессионал в своем деле, притом мотивированный на работу над проектом. Иногда возможность самому придумать себе правила работы может оказаться ширмой для



Приветствуются изменения требований даже на конечных этапах разработки.

Это может повысить конкурентоспособность продукта

некомпетентных или недобросовестных сотрудников. Но свобода действий обязательно вскрывает проблему мотивации и компетентности. Если люди находятся на недостаточном уровне самосознания, чтобы работать по принципам agile, в команде они долго не продержатся. Что касается мотивации участников проекта, то рецептов слома льда много. К примеру, можно изначально собрать команду из лидеров мнений и «продать» им идею agile-подхода. Специалисты, профессионально переросшие статические структуры своих компаний, также мотивированы к работе по принципу гибкого управления проектом. Такие люди с легкостью и желанием идут на самоорганизацию – это позволяет им найти свою нишу, попробовать что-то новое.

К построению agile полезно привлечь внешних консультантов. Они найдут способы привести команду в нужное состояние или же смогут точно определить, что эта команда самоорганизации не подлежит. И такое случается. Хотя с консультантами может возникнуть другая проблема: пока они есть, команда работает, как только уходят, все возвращается на круги своя. Самый серьезный вызов для внедрения гибкого подхода – уметь перестроиться. Команде нужно заработать доверие руководства, руководству – доверие команды, ведь когда все завязано на результат, на поверхность выходят все скрытые застаревшие проблемы организации.

ПРАКТИКИ ВЗАИМОДЕЙСТВИЯ

Чтобы сделать работу уже сформированной команды слаженной, существует несколько хороших практик. Например, daily stand-up, которую также называют scrum. Все просто: ежедневно участники команды ненадолго собираются и общаются друг с другом. При этом каждый должен ответить на следующие вопросы: что я делал вчера,

что буду делать сегодня и какие у меня возникли проблемы в ходе работы? Такие запланированные встречи, во-первых, делают действия участников проекта синхронными, а во-вторых, позволяют избежать ситуаций, когда над одной проблемой параллельно работают несколько человек, тогда как решением другой никто не занимается. В команде должен быть участник, выполняющий роль организатора процесса, – так называемый scrum-мастер, который следит за тем, чтобы каждая встреча команды достигала своей цели. Ведь профессионалы – люди творческие, легко могут о чем-то забыть, и кому-то надо следить за тем, чтобы они не теряли связь с реальностью. Scrum-мастер оказывает команде много услуг, но он совсем не похож на типичного менеджера. Кстати, менеджеру в традиционном понимании этого слова крайне тяжело перебраться в школу организатора групповой работы.

Другим примером эффективной практики может служить daily build. Ее суть заключается в том, что каждый день, например, в отрасли разработки ПО, у команды есть диск с версией конечного продукта, который мы ежедневно дорабатываем в соответствии с графиком приоритетов. Преимущество этой практики в том, что проблему с продуктом мы можем увидеть сразу. Такую версию можно показать заказчикам или инвесторам, а в случае с веб-проектами – открыть доступ реальным пользователям. Кроме того, в любой момент, если, допустим, закончились средства на дальнейшую разработку, есть что-то, что можно, грубо говоря, положить в коробку и продавать.

Важный этап в процессе самоорганизации – проведение ретроспектив. По окончании определенного этапа работы команда собирается и обсуждает, что прошло хорошо, а что не так, как хотелось бы. Ретроспективы дают возможность проанализировать и при необходимости подправить правила, по которым команда работала раньше.

Самоорганизация команды – это, конечно, хорошо, но ведь есть еще и заказчик. Scrum предполагает, что взаимодействие команды с product owner, то есть человеком, представляющим клиента, должно осуществляться постоянно. У команды всегда должна быть возможность уточнить задачу. Это важно, так как в процессе работы часто всплывают мелкие неточности, а иногда и радикальные изменения требований. Чем раньше станет известно об изменениях, тем лучше, но на то, чтобы остановить итерацию, перепланировать микроцикл, нужно согласие представителя клиента.

Для организации работы с заказчиком в scrum применяют следующий подход. Внутри организации-исполнителя выби-

