

ВОЗМОЖНОСТИ И АКТУАЛЬНАЯ ПРОБЛЕМАТИКА ПРИ ИНТЕГРАЦИИ МЕТОДОВ ЭКСТРЕМАЛЬНОГО ПРОГРАММИРОВАНИЯ И ПРИНЦИПОВ BUSINESS AGILITY.

© 2019 Мухин Кирилл Юрьевич

преподаватель, Департамент менеджмента

Финансовый Университет при Правительстве Российской Федерации, Россия, Москва

E-mail: K. Y. Mukhin@gmail.com

© 2019 Родина Мария Александровна

Финансовый Университет при Правительстве Российской Федерации, Россия, Москва

E-mail: rodina.mashulka@yandex.ru

В статье рассматривается один из наименее растиражированных Agile-методов — экстремальное программирование (XP). Концепция экстремального программирования оценивается в контексте релевантных принципов существования компаний на высококонкурентных рынках (business agility), во многом методологически противопоставляя ее фреймворку Scrum (как наиболее распространённому из гибких подходов и прикладному взгляду на Agile-ценности). Авторы объясняют причины невысокой популярности XP среди компаний, в той или иной степени применяющих agile в управлении корпоративными проектами, через анализ точек соприкосновения и актуальные реалии рынка. При этом в работе отражен действенный инструментарий проектного управления, позволяющий эффективно наладить работу проектных команд, базируясь на методологии экстремального программирования, а также приведены примеры успешного практического применения данной методологии в компаниях «Pivotal Labs» и «Додо Пицца».

Ключевые слова: XP, Экстремальное программирование, Business agility, Agile, Scrum, Проектный менеджмент, Гибкие подходы к управлению проектами, IT-проекты.

1. Занимаемое XP место среди Agile-семейства методов

Так называемые “Agile-методики” управления проектами уже прочно вошли в деятельность IT-компаний и компаний из других сфер. Одна из тенденций последних лет — тотальный переход менеджмента к “business-agility” далеко за пределами предметной области управления проектами, что должно позволять компаниям быстрее и дешевле адаптироваться к постоянно меняющимся условиям рынка. Повсеместное использование подходов Agile и их комбинирование с директивными стандартами управления проектами породило множество гибридов. Сложно даже перечислить полный пул Agile-инструментов, применяемых в компаниях, ведущих деятельность в абсолютно разных отраслях.

На текущий момент, в 2018 году все agile-практики можно разделить на универсальные, которые можно успешно применять в самых разных сферах бизнеса, и IT-практики, специфика которых ограничивает возможность применения конкретной практики вне проектов IT-отрасли. К последней группе относится и

“экстремальное программирование”, которое изначально было создано как метод управления проектами по разработке программного обеспечения.

По состоянию на 2017 год (рисунок 1), самой используемой agile-практикой стал Scrum (58% процентов опрошенных).

Преимущества Scrum на первый взгляд очевидны и могут быть сформулированы следующим образом:

1. Обучение требуется только проектному менеджеру, scrum-мастеру и владельцу продукта, а не всей команде проекта;

2. На рынке труда есть достаточно много scrum-коучей и agile-коучей, которые способны помочь проектным командам перейти к Scrum и повысить эффективность проектов и результативность проектной деятельности в компании;

3. Не требует больших изменений в проектной команде: изменения касаются, в основном, процессов планирования и коммуникации;

4. Легко пройти сертификацию и обучение, на рынке достаточно количество 2-х дневных курсов;

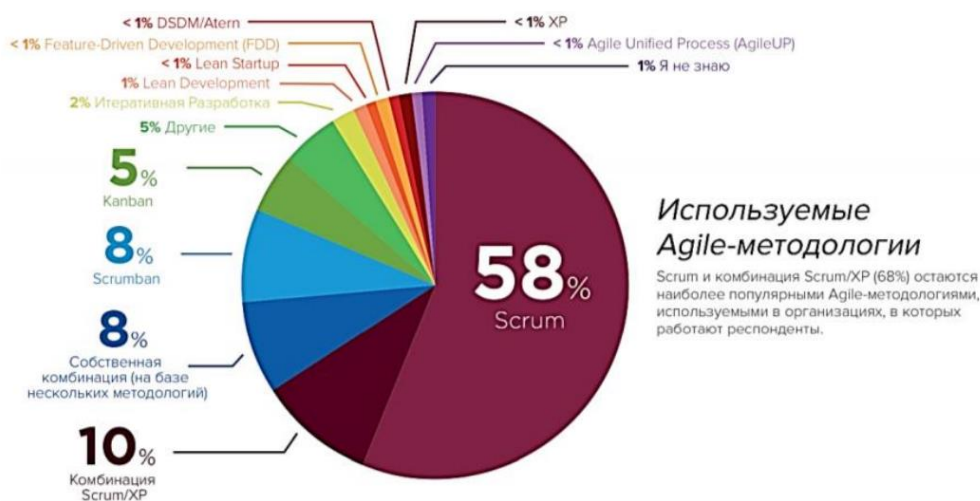


Рисунок 1. Использование Agile-методов. Источник: «State of Agiletm», 2017 [10]

5. Может применяться не только в IT-среде, является достаточно универсальной практикой (например, подходит для проектов в сфере дизайна);

6. Одна из наиболее директивных agile-практик, что облегчает переход от классических методов управления проектами.

При этом на втором месте по популярности расположился гибрид из Scrum и XP (10%). XP же в чистом виде используют около 1% организаций [5].

“Экстремальное программирование” появилось в 1996 году, за 5 лет до написания Agile Manifesto. Автором метода является Кент Бек, один из 17 специалистов, участвующих в подписании Agile Manifesto. Он также является создателем “TDD” (Test Driven Development) — самостоятельной it-методологии, которая входит в число практик экстремального программирования и может применяться как совместно со всеми практиками XP, так и отдельно даже в организациях, которые еще не осуществили переход

к гибким методологиям.

Так, в проектах, осуществляемых в рамках каскадной модели управления проектами в компании ООО «М. Видео Менеджмент», используются автоматические тесты, написанные до появления кода после согласования требований к функционалу.

2. Ценности, принципы и инструменты XP

Спустя 21 год XP оказалась малоиспользуемой agile-практикой, методы которой постоянно подвергаются критике со стороны профессионального сообщества. При этом, по аналогии со Scrum, суть метода экстремального программирования состоит из практик, ценностей и принципов. Существует 5 основных ценностей, которые команде необходимо разделять команде для успешного внедрения XP и повышения эффективности работы проектной команды [1]. В таблице 1 приведено сравнение ценностей Scrum и XP, чтобы наглядно проиллюстрировать, чем конкретно данные методы идеологически и

Таблица 1. Сравнение ценностей Scrum и XP. Разработано авторами.

| Ценности Scrum | Ценности Extreme Programming |
|---|---|
| Открытость: информация является открытой и доступной для всех членов проектной команды | Коммуникация: каждый член команды знает, что делают остальные |
| Обязательство: все члены команды должны стремиться к достижению поставленных перед командой целей | Простота: разработчики стараются создать максимально простое и прямое решение |
| Фокус: направленность всех сил и внимания на взятые командой обязательства | Обратная связь: постоянное тестирование и обратная связь держат качество продукта под контролем |
| Смелость: нужно иметь смелость сказать нет стабильности и сделать командной привычкой непрерывное совершенствование | Мужество: каждый член группы нацелен на выбор лучших решений для проекта, даже если это означает отказ от неудачных решений или требует иного подхода |
| Уважение: каждый член команды должен уважать всех остальных | Уважение: каждый член команды ценен для проекта |

технически отличаются друг от друга.

Многие ценности Scrum и XP схожи между собой: уважение, открытость и коммуникация, смелость и мужество. При этом в различиях кроется коренное различие: такие ценности XP, как простота и обратная связь, еще раз указывают, что данная методология применима исключительно для it-проектов, в то время как ценности Scrum (фокус и обязательство) говорят о применимости в любом проекте и направленности на достижение проектных целей.

Ценности не работают без практик, как и практики не дают эффекта без принятия ценностей. При этом многие специалисты отмечают важную роль ценностей. По их мнению, использование только практик не приведет проектные команды к желаемому результату. В лучшем случае, улучшение результата будет крайне незначительным — «лучше, чем ничего». В худшем случае, команда может отметить снижение результативности работы и отказаться от дальнейшего внедрения agile-методологий в работу.

Для поддержания ценностей используются 13 основных практик XP [3]:

1. Вся команда
2. Игра в планирование (планирование релиза и планирование итераций, которые в XP короче, чем в Scrum)
3. Частые релизы версий
4. Пользовательские тесты (определяются заказчиком проекта)
5. Коллективное владение кодом
6. Непрерывная интеграция кода (вывод разработанного кода в продуктив)
7. Стандарты кодирования (часто используется словарь для описания основных используемых команд, чтобы избежать дублей и упростить написание отдельных частей кода)
8. Метафора системы (сравнение с чем-то, что хорошо известно команде, которое придумывает архитектор или другой человек, представляющий результат целиком)
9. Устойчивый темп (отсутствие спадов и подъемов в темпе работы, 40-часовая рабочая неделя без переработок и работы в выходные)
10. Разработка, основанная на тестировании (тесты пишутся до написания кода) (TDD — Test Driven Development)
11. Парное программирование (2 человека, которые сидят за одной рабочей станцией, которая состоит из компьютера и одного или нескольких мониторов)

12. Простой дизайн (работа над поставленной задачей, без угадывания будущей функциональности, отсутствие workarounds (обходных решений) и так называемых «костыльных» решений — временных решений, не укладывающихся в целевое решение, которые поддерживают функционал продукта до момента создания целевого решения)

13. Рефакторинг (постоянное улучшение дизайна системы с целью упрощения)

3. Критика XP

Наиболее спорной из вышеперечисленных практикой является парное программирование. По сути, это работа двух разработчиков за одним компьютером, что можно описать пословицей «одна голова хорошо, а две лучше». По результатам исследований эффективности данного метода [2], можно сделать вывод, что работа в паре идет с обычной скоростью или немного (15%) медленнее. При этом качество кода ощутимо повышается. Так, получаемый в результате парного программирования код содержит на 60% меньше ошибок и технических долгов (необходимая часть функционала, которая переносится в «backlog» следующего спринта и ведет к созданию «костыльных» решений).

Критика экстремального программирования зачастую включает в себя замечания к Agile в целом. В частности, противники внедрения XP выделяют такие недостатки метода, как [8]:

1. XP не работает без вовлечения команды, и эта проблема не решается внедрением agile-методологии
2. XP может использоваться как средство для получения денег от клиента, при этом у клиента отсутствует понимание конечного продукта
3. Отсутствие структуры и необходимой документации
4. Работает только для разработчиков высшего уровня (senior)
5. Недостаточный дизайн программного обеспечения, отсутствие необходимых спецификаций
6. Требуется постоянных встреч, что увеличивает время разработки и затраты заказчиков
7. Требуется слишком много изменений в организационной структуре и культуре
8. Возможно затягивание переговорного процесса для заключения договоров
9. Может быть неэффективным методом,

если требования к одной и той же части кода изменяются в течение разных итераций, в итоге получается дублирование работы, чего можно было бы избежать при следовании первоначальному плану

10. Невозможно дать реалистичные оценки трудозатрат на проект, так как в начале проекта нет понимания объема требований

11. Риск отклонения от первоначального содержания повышается из-за отсутствия задокументированных детальных требований

12. Agile базируется на поставке частей функционала, при этом нефункциональные атрибуты качества сложно записать в виде пользовательских историй

При этом несложно заметить, что критикой непосредственно XP-инструментария, а не всех гибких подходов управления проектами, является лишь пункт «работает только для разработчиков высшего уровня» (что, кстати, не является доказанным утверждением). Внедрение и использование XP позволяет разработчикам начального и среднего уровней получить необходимый опыт быстрее, чем работа при традиционном подходе.

“Эмпирический анализ показывает, что хотя абсолютное большинство респондентов говорят о позитивных результатах от внедрения различных Agile-практик на всех уровнях в любых проектах, они отмечают сложности масштабирования Agile. Две наиболее часто отмечаемые сложности — это корпоративная культура, расходящаяся с Agile-ценностями (63%) и недостаточные навыки и опыт использования Agile-подходов (47%)” [11] (Рисунок 2).

Более того, данная статистика ничего не говорит о «систематической ошибке выжившего

(“survivorship bias”), т.е. о тех случаях и кейсах, когда «не получилось» — ведь здесь фигурируют исключительно “best-practices” при внедрении Agile. Хотя вполне очевидно, что большее количество итераций приводит к возникновению большого количества проблем, которые необходимо решить, зачастую в режиме цейтнота. На этом этапе и должна включиться команда проекта, задача которой вовремя вносить необходимые корректировки по содержанию.

Соответственно можно постараться выделить основные причины низкой популярности именно XP, по возможности очищенных от влияния смещенных оценок, касающихся Agile в целом [9] (основным исследовательским вопросом данной работы, при этом, является — насколько заслуженной и обоснованной является данная “расстановка сил”):

1. *Акцент на разработке и кодировании.* В то время как проектный менеджмент делает акцент именно на управлении, экстремальное программирование обращает больше внимания на техническую сторону проекта. Иными словами, XP не создан для проектных менеджеров, он создан непосредственно для команды.

2. *Требует больших инвестиций.* Хотя утверждение, что работать по принципам XP могут только разработчики высшего уровня, не является полностью верным, но лучшим решением для XP-команды являются именно разработчики-звезды, которые способны писать работающий простой код, пригодный для покрытия автоматическими тестами. С разработчиками-звездами не любят работать проектные менеджеры, потому что этот тип разработчиком лучше знает, как должна функционировать система, они реже идут на уступки и их



Рисунок 2. Сложности в ходе внедрения и масштабирования Agile. Источник: «State of Agiletm», 2017 [10]

практически невозможно уговорить. А еще такие специалисты редко встречаются. Для внедрения Scrum часто достаточно отправить менеджера проекта на обучение, длящегося 2 дня. Для внедрения XP необходимо обучить всю команду, что значительно дороже. Особенно учитывая то, что разработчиков в крупных проектах может быть в 20 раз больше, чем проектных менеджеров. Возрастают затраты на инфраструктуру. Автоматические тесты требуют больше производительности и мощности оборудования, чем ручные. Необходимы большие изменения в организационной структуре, что также часто требует затрат. Отдельная графа расходов — XP-тренеры. Их трудно найти, их труд стоит дорого, потому что они являются настоящими мастерами (можно даже сказать, ремесленниками) программного обеспечения. Зато они заботятся о качестве больше, чем кто-либо. И они непоправимые идеалисты.

3. *Это иррационально.* Пользовательское тестирование означает более близкое знакомство бизнес-пользователей с кодом. И требует значительно больше времени, пусть даже и добавляет намного больше ценности продукту. Написание тестов до разработки является самой правильной практикой, потому что на выходе команда получает функциональность нужного качества. Но это неочевидно для большинства людей, у которых нет видения ценности наличия тестов до старта разработки. По их мнению, это пустая трата времени. Оценка в story points не дает понимания сроков разработки. Часто тратится дополнительное время для перевода оценки в затрачиваемые на разработку часы. Парное программирование ведет к увеличению количества программистов в штате компании. Для руководства нелогичным выглядит выделять 2 FTE на задачу, с которой может справиться один человек.

4. *Это слишком сложно.* Рефакторинг, простой дизайн и написание тестов до разработки требует много усилий со стороны разработчиков. Именно отсюда идет идея о том, что XP создан исключительно для профессионалов. Специалистам, работающим с XP, необходимо постоянно держать в уме множество вещей, о которых обычно вспоминают только после проваленного теста системы (написанного после разработки, что немаловажно).

5. *Легче заметить неуспешное внедрение XP в компании.* Успешное внедрение XP ведет к

повышению ценности создаваемого программного обеспечения, что не так заметно, как увеличение сроков на разработку, затрат на инфраструктуру и коуча. Положительные результаты часто воспринимаются как должное или даже независимое от XP последствие. Отрицательные результаты, на которые было потрачено много усилий со стороны проектной команды и руководства, будут восприняты в штыки и не дадут дальше внедрять XP в жизнь компании.

4. “Лучшие практики” использования методики XP

Самым успешным примером внедрения экстремального программирования можно считать компанию Pivotal Labs, создателей Pivotal Tracker [4]. Компания успешно реализует такие практики, как: парное программирование, TDD, рефакторинг, непрерывная интеграция, ежедневные stand-up'ы, совместная работа с клиентом, ретроспективы, использование трекера задач (при работе именно над этим продуктом — Pivotal Tracker — компания впервые стала использовать в своей работе agile-практики), использование демо-среды для приемки работы проектным менеджером, близость проектного менеджера к работе команды, понедельное планирование и специальные дни для старта/финиша проекта.

Разработчики в компании — это не те люди, которые сидят в наушниках, обособленные от мира вокруг. У них нет собственных столов и компьютеров, вся работа проходит в парах.

В компании есть собственная философия «неуверенности», которая обозначает принципы работы. Так, у разработчиков нет спецификаций. Часто команда не знает, куда хочет прийти. А то, что команда знает, может измениться уже на следующий день. Работа будет длиться до тех пор, пока получившийся продукт не будет полностью удовлетворять заказчика. Поэтому в компании нет понятия «быть завершенным».

Основным преимуществом XP компания называет именно TDD. До написания тестов не начинается кодирование. Именно тесты содержат информацию о том, что нужно клиенту. В компании заменяют всю проектную документацию тестами, потому что они понятны и бизнес-пользователям, и разработчикам, и их легче держать в актуальном состоянии, тогда как проектная документация устаревает, как только вы ее публикуете. Ее необходимо долго согласовывать, в то время как требования и потребности

меняются очень стремительно. Разработчикам не нужны комментарии в коде, объясняющие, почему код написан так и что именно он делает. Все это уже определено в тестах.

При этом крайне важна роль руководителя проектов. Проектный менеджер должен поддерживать backlog в актуальном состоянии, представлять приоритеты задачам, распределять работу между парами, своевременно принимать выполненные задачи, чтобы темп работы оставался стабильным, планировать работу и поддерживать команду разработчиков. Проектный менеджер дает команде ориентиры, вселяет уверенность и не мешает команде.

В российской практике мало компаний, практикующих XP. На прошедшей в конце марта 2018 года 12-й глобальной конференции по гибкому управлению процессами «Agile Days» об экстремальном программировании рассказывал только один спикер — Олег Блохин из компании «Додо Пицца» [7].

В «Додо Пицце» постоянно практикуют XP. Особенно часто в компании используют парное программирование. При этом используют не только классические пары из программистов,

но и пары программистов с юристами, бизнес-аналитиками, инженерами инфраструктуры.

Компания использует такие практики, как: рефакторинг, короткие итерации, постоянная интеграция, парное программирование, пользовательские тесты [6].

Резюмируя вышеизложенное, хотелось бы привести цитату Джона Симпсона, отражающую доминирующее мнение большинства экспертов, включая авторов: “Не существует ни одного совершенного процесса для управления проектом — серебряной пули — Waterfall, PMBoK, ICB от IPMA, Agile, Scrum, XP или иного... Вместо этого большинство использует гибридный, постоянно развивающийся процесс, который наилучшим образом соответствует потребностям именно их проекта.” [11]. Таким образом, XP представляет собой отличную практику, как плацдарм для разработки программного обеспечения. Но для ее повсеместного внедрения требуется не просто изменить работу компаний, но и провести значительную работу по внедрению ценностей экстремального программирования в деятельность проектных команд.

Библиографический список

1. *Стеллман, Эндрю*. Постигая Agile. Ценности, принципы, методологии / Эндрю Стеллман, Дженнифер Грин; пер. с англ. С. Пасерба. — М.: Манн, Иванов и Фербер, 2017. — 448 с.
2. *Бындю, А.* Экстремальное программирование: Pair Programming [Электронный ресурс] — <https://habrahabr.ru/post/151563/>
3. *Гранько, О.* Экстремальное программирование (XP) не для слаонервных [Электронный ресурс] — <https://worksection.com/blog/extreme-programming.html>
4. Как работают создатели Pivotal Tracker: о разработке, управлении и найме людей [Электронный ресурс] — <https://habrahabr.ru/post/194776/>
5. Какие подходы из Agile наиболее популярны в 2017 году? [Электронный ресурс] — <http://project-management.zis.by/agile/kakie-podhody-iz-agile-naibolee-populjarny-v-2017-godu.html>
6. О том, как устроено it в «Додо Пицца» [Электронный ресурс] — <http://dodois.com>
7. Agile Days — конференция о гибких методологиях управления разработкой ПО и не только [Электронный ресурс] — <https://agiledays.ru/#speaker-516>
8. Extreme Programming [Электронный ресурс] — https://en.wikipedia.org/wiki/Extreme_programming#Criticism
9. *Partogi, Joshua* 5 reasons why eXtreme Programming isn't popular [Экстремальное программирование] — <https://medium.com/agility-path/5-reasons-why-extreme-programming-isnt-popular-83790418b901>
10. 11-th annual report «State of Agiletm». Version One inc., 2017
11. *Мухин К. Ю.* Следуя “гибким” путем от устойчивости к клиентоцентричности: возможности и подходы // Современная наука: актуальные проблемы теории и практики. Серия: экономика и право. — 2018. — № 1. — с. 34–43
12. *Мухин К. Ю.* Новая эра Agile: есть ли будущее у традиционных подходов в управлении проектами?! // Экономические науки. — 2018. — № 159. — с. 17–21
13. *Мухин К. Ю.* Agile-трансформация: миф или революционный управленческий инструментарий? // Научное обозрение: теория и практика. — 2017. — № 11. — с. 33–48

14. Мухин К.Ю., Антохина Ю.А. Специфика требований к Agile-трансформации как инструменту инновационного обновления менеджмента компании // Вестник Тихоокеанского государственного университета. — 2018. — № 4 (51). — с. 109–116
15. Мухин К.Ю., Родина М.А. Квинтэссенция гибкости: анализ возможных проблем и эволюция ситуационных решений в проектном управлении // Инициативы XXI века. — 2018. — № 3–4. — с. 8–11
16. Родина М.А. Экстремальное программирование: проблемы при внедрении // Приоритетные направления проектного менеджмента в цифровой экономике: Сб. научн. тр. студентов, аспирантов и преподавателей Финансового университета при Правительстве РФ. — М.: СВИВТ, 124 с. — 2018. — В. 8. — Т. 4. — с. 36–46.
17. Родина М.А. Экстремальное программирование (xp): подходы и опыт «гибкого» внедрения в современных компаниях // Взгляд молодых ученых на проблемы устойчивого развития: Сборник научных статей по результатам IV Международного конгресса молодых ученых по проблемам устойчивого развития. Роль менеджмента в стимулировании экономического роста. — 2019. — с. 142–148.
18. Фунтов В.Н., Парамонов Д.В., Малоземов С.Н. Гибкое управление в негибкой отрасли. // НИР. Российский журнал управления проектами. — 2017. — № 1. — с. 25–36
19. Ризби Д., Сазерленд Д., Такеучи Х. Новый рецепт инноваций: модель agile. Как освоить модель, которая меняет саму суть управления // Harvard Business Review-Россия. Август, 2016. с. 39–42.
20. Книберг Х. Scrum и XP: заметки с передовой / Х. Книберг. — Киев: InfoQ, 2007.
21. West D., Gilpin M., Grant T., Anderson A. Water-Scrum-Fall Is The Reality Of Agile For Most Organizations Today // Forrester Research, Inc. 2011.
22. Schwaber K. and Sutherland J. The Definitive Guide to Scrum: The Rules of the Game, 2013.
23. Бобровский С. Самый эпический провал agile-методологий, PCWeek, 05/2015. [Электронный ресурс] — <https://www.pcweek.ru/idea/blog/idea/4908.php/>
24. Makabee H. Конец Agile: смерть от примитивизма. Практика проектирования систем / H. Makabee. 2016. — С. 10.
25. Книберг Х., Скарин М. Scrum и Kanban: выжимаем максимум. — Киев: InfoQ, 2010. — 78 с.
26. Augustine, Sanjiv. Managing Agile Projects. // Prentice Hall. 2005.
27. Successful Solutions Through Agile Project Management. ESI Int. 2010
28. Mark Flynn. Agile: the good, the bad and the ugly. Project Smart. 2011
29. Craig Larman. Agile and Iterative Development: A Manager's Guide (Agile Software Development Series). Cockburn, Alistair and Highsmith, Jim (Series Editors). 2003