СОВРЕМЕННЫЕ ПОДХОДЫ К ПРОЕКТАМ. МЕТОДОЛОГИИ. AGILE: SCRUM

MODERN METHODS OF PROJECT MANAGEMENT. METHODOLOGIES. AGILE: SCRUM

УДК 005

Мокшин Владимир Васильевич, кандидат технических наук, доцент, доцент кафедры «Автоматизированных систем обработки информации и управления» Казанский национальный исследовательский технический университет им. А. Н. Туполева – КАИ, Россия, г. Казань

Гайнутдинова Алсу Маратовна, студентка, 2 курс, Институт компьютерных технологий и защиты информации, Россия, г. Казань

Самсонов Сергей Олегович, студент 2 курс, Институт компьютерных технологий и защиты информации, Россия, г. Казань

Mokshin Vladimir Vasilievich

Gainutdinova Alsu Maratovna, alsu471704116@gmail.com

Samsonov Sergey Olegovich

Аннотация

В данной работе рассматриваются подходы для работы в проектах, основанный на методологии Agile. Описываются и сравниваются методы Scrum, эффективных методах управления проектами. Как и где требуется использовать методологию и почему она может быть лучше. Делается вывод, о том, что методология Agile и входящий в нее фреймворк Scrum позволяют более продуктивно управлять ИТ-проектами, создавать качественные программные

продукты в рамках данных проектов и в других сферах деятельности, таких как продажи, банки, телеком и т. д.

Annotation

This paper discusses approaches for working in projects based on the Agile methodology. It describes and compares the methods of Scrum, effective methods of project management. How and where to use the methodology and why it might be better. It is concluded that the Agile methodology and the Scrum framework included in it allow for more productive management of IT projects, creating high-quality software products within these projects and in other areas of activity, such as sales, banks, telecom, etc.

Ключевые слова: Agile, Scrum, работа в проекте, команда, общение с заказчиком, результативная работа в IT.

Keywords: Agile, Scrum, project work, team work, communication with the customer, effective work in IT.

1. Введение

Сегодня разработка IT проектов стала полноценным бизнесом, а иногда даже обогнала всех в этом понимании. Появилось множество профессий для работы с заказчиком и командой разработчиков. Поделили работу и создали один слаженный механизм. Так же и появились свои методологии для работы этого механизма.

Методология разработки софта — организация труда, включающая идеологические принципы, план, контроль над процессами, подход к сотрудникам.

Проще говоря методологии — это опыт, накопленный ошибками и успехами. Усвоив азы разработки, возможно избежать полных провалов, достичь наилучшего успеха и распределить работу в команде.

Проект менеджмент в IT системе стремительно развивается и каждый раз трансформируется в свои новые реализации. Сегодня появилось новое модное слово Agile в этой сфере. Нам презентуют гибкую схему, которая готова меняться в любой ее стадии. Многие компании используют новые методы и подходы Agile. Мы рассмотрим, как и почему используют такие подходы. И нужно ли гнаться за модой.

2. Agile

Так что же такое Agile? Если открыть толковый Оксфордский словарь, то можно найти 2 определения:

1. Able to move quickly and easily. (Способен двигаться быстро и легко.)
2. Able to think and understand quickly. (Умеет быстро думать и понимать)

Agile или гибкая методология разработки, обобщающий термин для подходов, которые берут начало из Манифеста гибкой разработки программного обеспечения(Agile manifesto) и 12 принципах.

Agile Manifesto — основной документ, содержащий описание ценностей и принципов гибкой разработки программного обеспечения, разработанный в феврале 2001 года на встрече 17 независимых практиков нескольких методик программирования, именующих себя «Agile Alliance». Манифест[1] содержит 4 ценности и 12 принципов.

Принципы:

1. Наивысшим приоритетом является удовлетворение потребностей заказчика, благодаря регулярной и ранней поставке ценного программного обеспечения.
2. Изменение требований приветствуется, даже на поздних стадиях разработки.
3. Работающий продукт следует выпускать как можно чаще, с периодичностью от пары недель до пары месяцев.
4. На протяжении всего проекта разработчики и представители бизнеса должны ежедневно работать вместе.

5. Над проектом должны работать мотивированные профессионалы. Чтобы работа была сделана, создайте условия, обеспечьте поддержку и полностью доверьтесь им.

6. Непосредственное общение является наиболее практичным и эффективным способом обмена информацией как с самой командой, так и внутри команды.

7. Работающий продукт — основной показатель прогресса.

8. Инвесторы, разработчики и пользователи должны иметь возможность поддерживать постоянный ритм бесконечно.

9. Постоянное внимание к техническому совершенству и качеству проектирования повышает гибкость проекта.

10. Простота — искусство минимизации лишней работы — крайне необходима.

11. Лучшие требования, архитектурные и технические решения рождаются у самоорганизующихся команд.

12. Команда должна систематически анализировать возможные способы улучшения эффективности и соответственно корректировать стиль своей работы

12 принципов можно объединить в 4 идеи:

1. **Люди и взаимодействие** важнее процессов и инструментов.

2. **Работающий продукт** важнее исчерпывающей документации.

3. **Сотрудничество с заказчиком** важнее согласования условий контракта.

4. **Готовность к изменениям** важнее следования первоначальному плану.

Сложно назвать Agile методологией, ведь по своей сути это фреймворк, с которым в дальнейшем мы работаем и трансформируем его в нечто новое. Некоторые могут сказать, что Scrum или Kanban это Agile, и в конечно счете сделают колоссальную ошибку, потому что новый собранный фреймворк, основанный на ценностях agile.

В классической модели проектного подхода разработки можно выделить основные этапы Рис.1:

1. Инициация
2. Планирование
3. Разработка
4. Реализация и тестирование
5. Выход в продакшн

Если обратить внимание, то каждый из этапов идет четко друг за другом. Например: мы не можем начать тестирования до конца разработки.

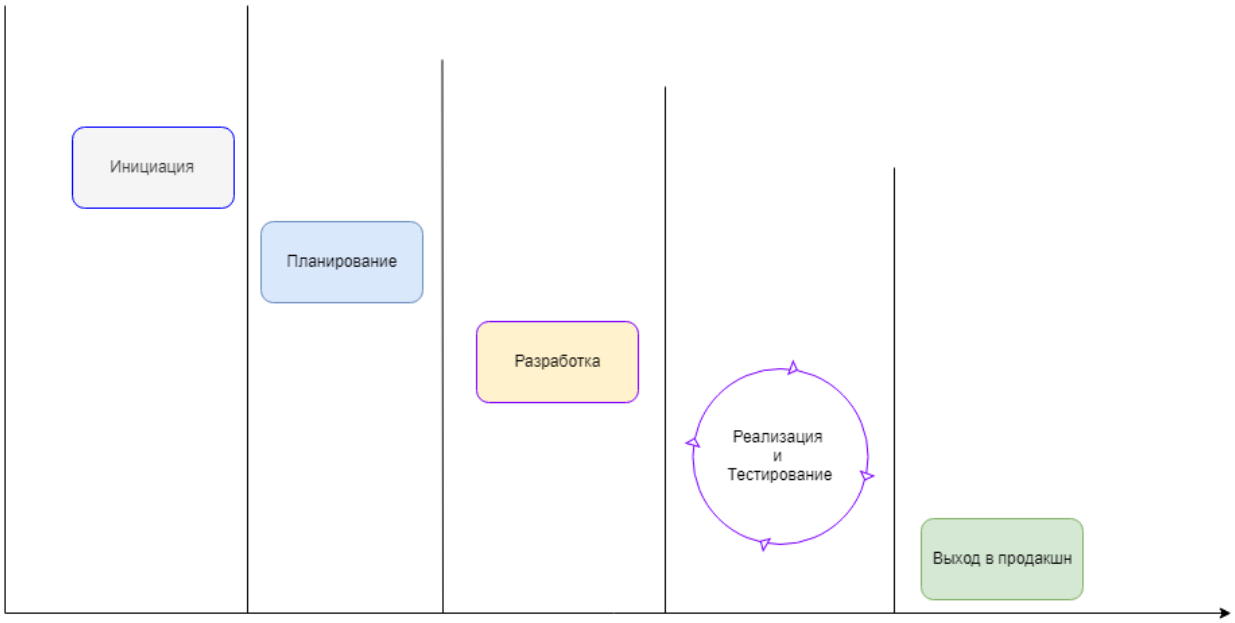


Рис.1 Графическое представление классической модели

Данный подход ориентирован на проекты, в которых есть строгие ограничения по последовательности выполнения задач, но такой подход не всегда годится под IT проектные решения. Если мы планируем не изменчивость проекта и точно знаем какой нам нужен результат можно остановиться и на этой модели, а вот Agile предлагает более гибкую структуру рис. 2:

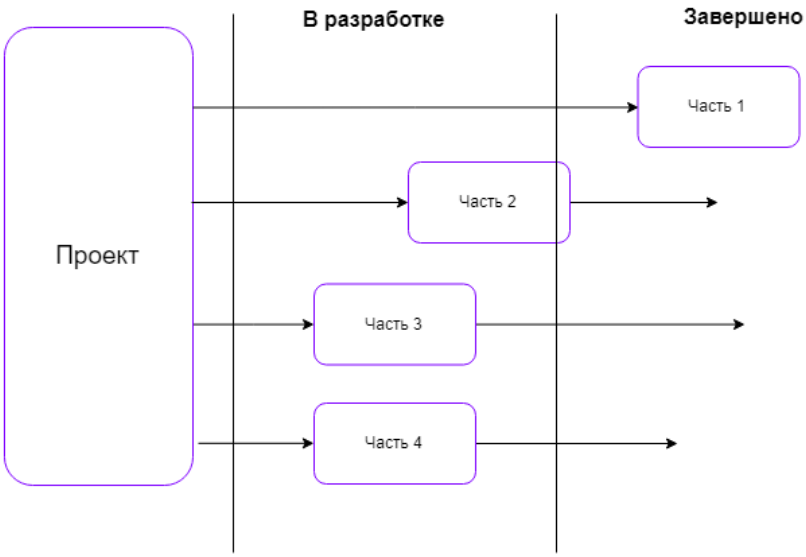


Рис.2 Графическое представление Agile методологий

Самое главное достоинство Agile – его гибкость и адаптивность. Он может подстроиться под практически любые условия и процессы организации. Именно это обуславливает его нынешнюю популярность и то, сколько систем для различных областей было создано на его основе. Мы готовы к любым изменениям от заказчика, нужно понимать изменчивость рынка и во время реализации проекта в мире может появиться что-то новое, и текущее решение может уже не устраивать заказчика. Команда должна быть готова и должна понимать важность таких решений.

3. Scrum.

Как уже говорилось ранее Agile это начальный фреймворк и из него вытекают другие новые и все полезны в разной мере. Согласно статистике[3] от компании ScrumTreck, в России используют в основном Scrum, Kanban, Свои собственные наработки и совмещенные методологии Рис. 3.

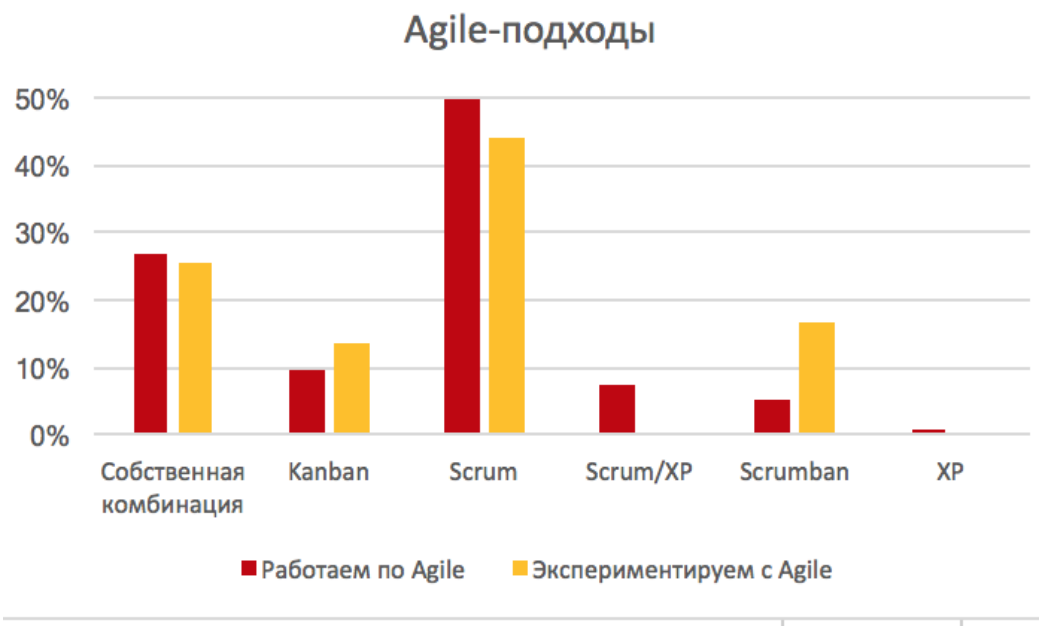


Рис.3 Статистика по использованию Agile в РФ

Конечно же мир не останавливается на Scrum, Kanban, XP и Собственный комбинации. Почему их нет в этой статистике? Методологии Agile пришли к нам из Европейских стран, где взгляд на бизнес в целом другой. Многие методологии просто не могут прижиться в России сейчас, многие компании и сейчас имеют не уверенное желание использовать новое. Почему так популярен Scrum? Он довольно прост в понимании и очень эффективный в команде.

Каждый фреймворк по-своему хорош, но у них разные цели. Когда речь заходит о Kanban мы рассчитываем на работу в известной нам сфере и ожидаем быструю реакцию на запросы. Наш рабочий процесс становится как конвейер и работа не стоит на месте. Когда мы говорим о Scrum мы подразумеваем работу с малоизвестным продуктом, где мы нацелены на качественную программную реализацию и готовы создавать его столько времени сколько на это потребуется.

Scrum – это фреймворк для разработки поставки и поддержки комплексных продуктов. Основа Scrum это скрам-команда. Команда делится на три роли:

1. Scrum-мастер
2. Владелец продукта
3. Разработчик

Рассмотрим каждую из ролей.

Разработчик:

В скрам-команде обычно от 3 до 9 человек. Разработчики образуют команду разработки. В команду разработки входят: аналитики, дизайнеры, разработчики, тестировщики и т.д. **Владелец продукта** (человек от бизнеса): отвечает за бизнес результат команды.

Scrum-мастер:

Отвечает за то, чтобы команда работала эффективно.

Владелец продукта собирает ценные пожелания от заказчика и сохраняет их в специальном артефакте, который называется бэклог продукта, бэклог продукта может меняться и меняется на всем протяжении разработки продукта.

Бэклог – это артефакт Scrum, который содержит в себе все функции, улучшение и исправления для будущей реализация в продукте. Это такой список пожеланий заказчика, который обладает определенными особенностями.

Свойства бэклога:

1) Плоский
2) Приоритезированный

Что значит Плоский и Приоритезированный список? Это означает, что мы можем однозначно сказать, какой элемент бэклога нам требуется реализовать вслед за текущим. Принцип приоритезации выбирается очень простой. Сверху бэклога должны находиться самые ценные элементы и при этом самые маленькие, то есть такие, которые команда разработки может реализовать быстрее всего.

3) Неоднородный

Неоднородный список означает, что он может включать в себя элементы разного размера. Как правило, выделяют три типа элементов по размеру. Это пользовательская история. Главное требование к таким элементам одно. Команда разработки должна иметь возможность реализовать его в продукте за один спринт. Следующий элемент, они чуть побольше называются Ipic или по русски эпосами. Они уже не помещаются в один спринт и большие наименее определенные пожелания заказчика называются темами. Рис. 4

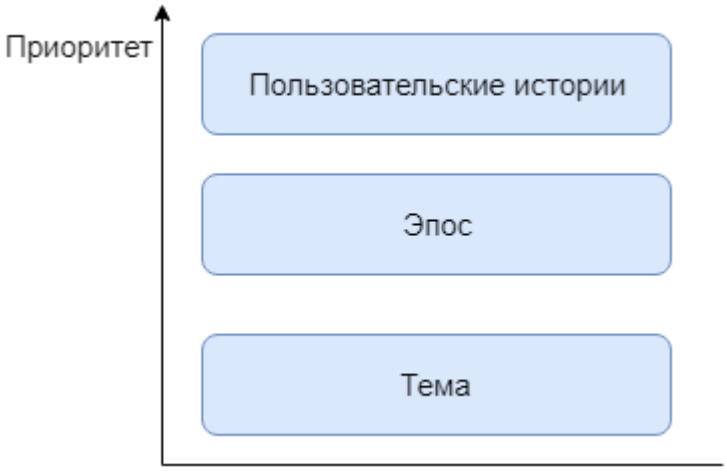


Рис.4 Приоритеты в бэклоге

Пример пользовательской истории: “Я, как пользователь такой-то роли, хочу или могу выполнить определенные действие в нашем продукте” так мы понимаем, что нужно пользователю от нашей системы.

4) Ценностно-ориентированный

5) Живой

В бэклоге должны рассматриваться реальные задачи, которые нужны заказчику, бэклог должен вести нас к определенной цели. Вся работа делится на итерации, в Scrum данные итерации называются спринт.

К концу каждого спринта команда разработки создает инкремент продукта, то есть новую версию продукта с новыми ценными для заказчика функциями. Рис.5

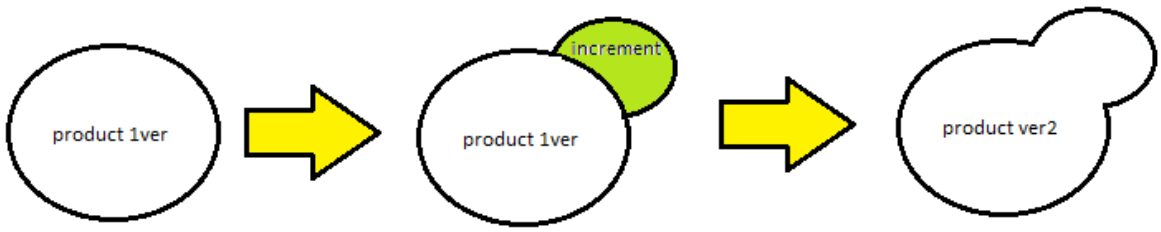


Рис.5 Инкремент продукта

После мы обсуждаем версию продукта с заказчиком и получаем фидбэк (обратную связь) по выпущенной версии продукта. Это нужно что бы понимать, что мы делаем то, что нравится заказчику и что он будет доволен нашим продуктом, на таких обсуждениях собираются все стейкхолдеры(заинтересованные лица). Бывают такие ситуации, что заказчик попросил одно, но, когда увидел реализацию понимает, что ему нужно что-то другое, поэтому команда разработки выявляет такие моменты на ранних стадиях. После каждой итерации мы понимаем в какую сторону двигаться что бы создать качественный продукт. Рис.6.

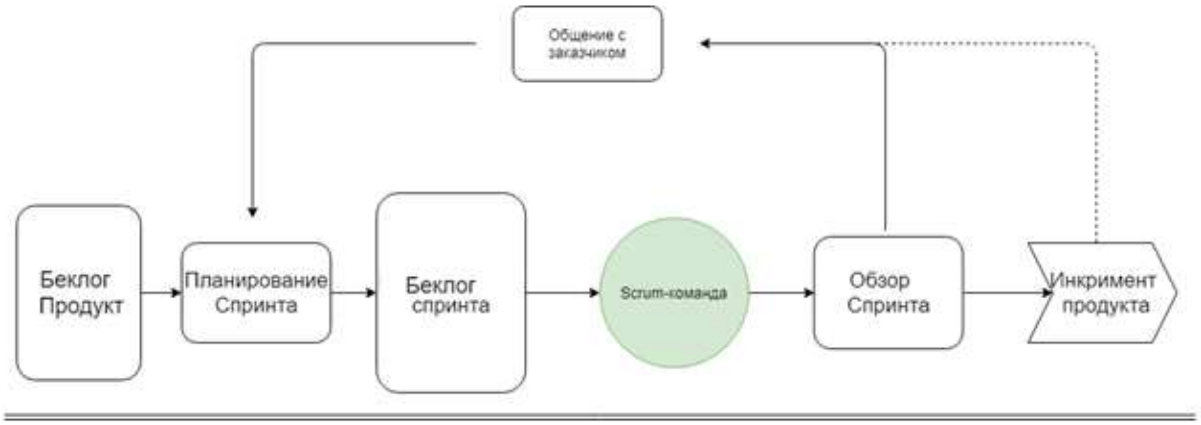


Рис.6 Графическое представлении Scrum-модели

Мы поговорили о том, что такое бэклог продукта, но что такое бэклог спринта, в планировании мы рассматриваем бэклог продукта и разбиваем пользовательские истории на задачи, которые возможно решить в рамках одного спринта. В бэклоге же мы берем эти задачи и составляем свой бэклог для спринта и за этот бэклог отвечает уже команда разработки, потому что взялась эти задачи реализовать. Как же команда понимает, что может выполнить ту или иную задачу. В scrum есть метод оценки времени и сложности пользовательской истории и подзадач. На первых спринтах это довольно сложно сделать, но начиная с 3го спринта команда может уже это делать, ориентируясь на результаты прошлых бэклогов. Мы отказываемся от оценки в часах, потому что это всегда субъективное мнение и не точное. В планировании спринта мы даем оценку задачам по их сложностям. Например, задача А имеет сложность равной 1-це, тогда при оценке задачи В, если она в два раза сложнее мы дадим оценку 2. Нам не обязательно оценивать сложность проекта начиная с 1цы, мы можем сразу начать использовать 100ни и 1000, некоторые используют ряд Фибоначчи, чтобы уйти как можно дальше от числовых значений. Есть команды, которые оценивают сложность в марках машин. Но есть один минус в этом подходе, кому-то задача может показаться легкой, а кому-то очень даже сложной.

В этих случаях придумали оценку стиле игры в покер. Допустим 5 человек из команды разработчиков решили оценить задачу. Они берут карточки и ставят свои оценки задаче, после они показывают оценки и обсуждают их.

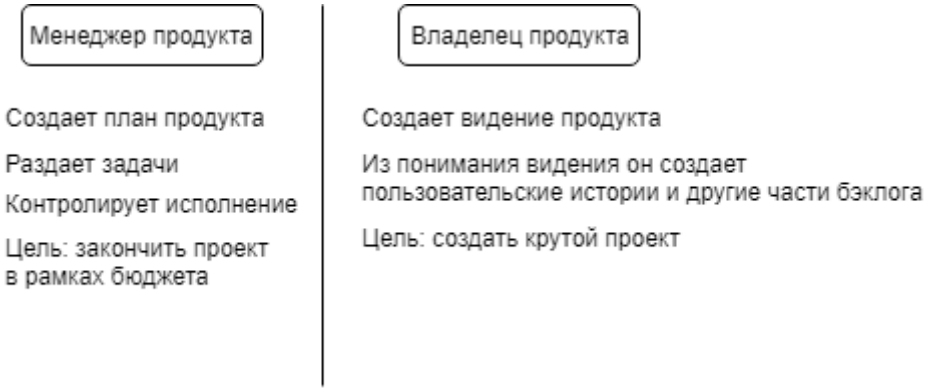
Если разрыв между оценками не большой, то берется среднее между ними. Например, между 4 и 6 мы можем выбрать 5. Но что делать если разрыв между оценками очень большой? Например: 10 2 3 4 и 7, тогда человек с оценкой 10 и с оценкой 2 высказываются почему задача сложная или наоборот легкая. Может один не учел некоторые сложности, которые знает второй. Или 1 из них знает легкий вариант решения этой задачи. После обсуждения происходит переоценка.

Функции Владельца продукта.

1. Вовлекает заказчиков в процесс разработки
2. Формирует видение продукта

3. Доносит видение и бэклог до команды и заказчиков
4. Владеет бэклогом(управляет и делает видимым всем)
5. Отвечает за бизнес результат

Многим может показаться, что Владелец продукта похож на менеджера проекта, давайте рассмотрим:



Владелец продукта не раздает задачи, он собирает видение продукта и формирует бэклог, что бы команда разработки могла выбрать и разработать нужны части проект в той или иной последовательности. Владелец продукта заинтересован в качестве продукта, а не в его наличие.

Роль Scrum-мастера.

Проводит и помогает проводить встречи. Устраняет препятствия на пути команды. Воспитывает в команде самоорганизацию. Он воспитывает команду и помогает ей эффективно воспользоваться ресурсами, что они имеют.

Таблица 1.

Сравнение Scrum-мастера и проект менеджера.

Проект менеджер	**Скрам мастер**
Составляет план проекта	Проводит и помогает проводить встречи
Раздаёт задачи	Помогает участникам договориться
Контролирует исполнение	Устраняет препятствия на пути команды
Координирует работу	**Цель:** построить эффективную команду
Цель: закончить проект	

4. Гибкость против статичности

Как говорилось ранее слово Agile очень модно и многие компании опираясь на успех других пытаются полностью перейти на этот фреймворк. Специалисты советуют внедрять Agile постепенно на пилотных проектах внутри компании. И стоит учитывать так же что нам нужны мотивированные люди. Но всем ли компаниям и всем ли проектам так нужен Scrum, Kanban, XP и другие фреймворки?

Конечно же нет. Scrum хорошо будет работать в мало известной проектной задаче, где мы будет выдвигать столько гипотез сколько это нужно. Но как понять, насколько известен нам тот или иной проект. Гипотеза основана на теории запутанности [4] Рис.7

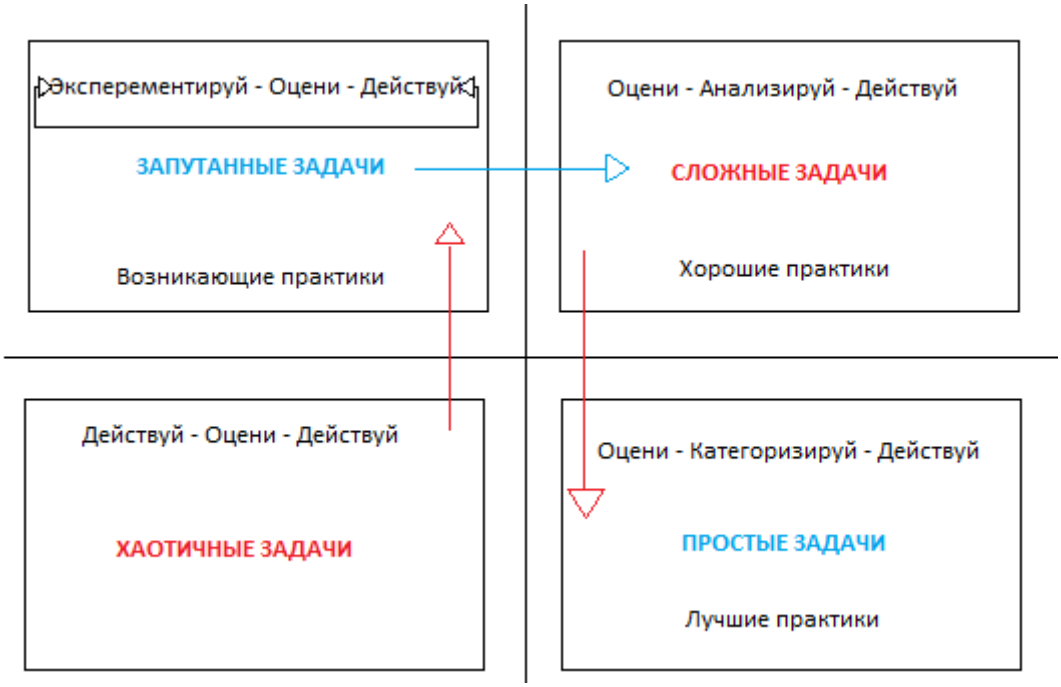


Рис.7 Теорема запутанности

Если задача ясна и понятна, мы можем оценить ее и видим вектор, в котором можно двигаться, это простая задача. Если после оценки задачи нам требуется анализ для решения задачи, то это уже сложная задача. Если мы не понимаем задачу и не можем ее оценить мы выдвигаем гипотезу и пытаемся действовать до тех пор, пока мы не найдем решение. И Хаотичные задачи, в которой максимально ничего не известно, мы стараемся действовать и в конечно итоге реагировать. Если объяснять простыми словами, то Теория состоит в том, что из Хаотичной задачи мы можем перейти в Запутанную. Из запутанной в Сложную и из сложной в простую. Для Scrum подходят запутанные задачи или сложные. Мы ожидаем что столкнемся с трудностями и постепенно их решаем. На рис. 8 мы можем увидеть графическое представление задач, которые можно решить scrum.

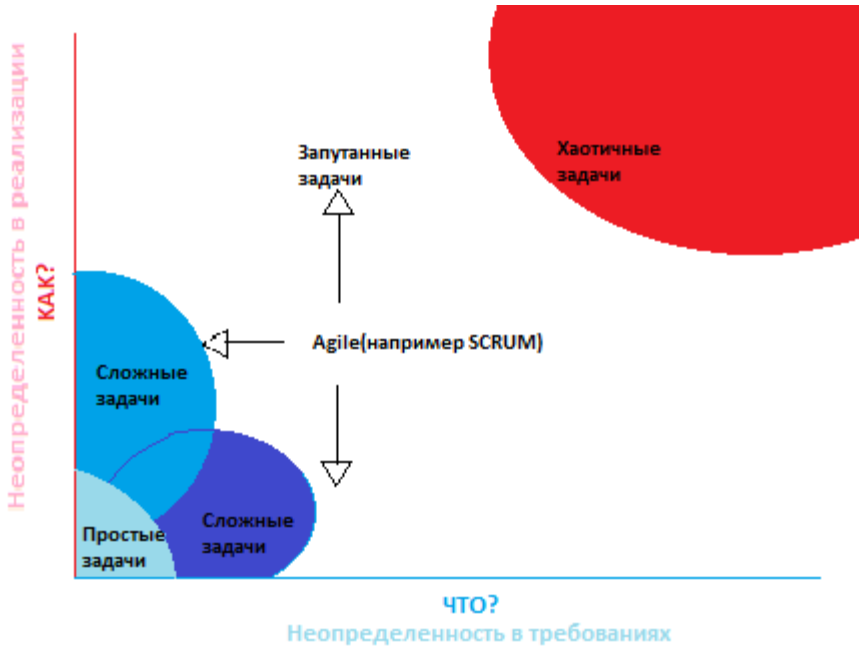


Рис.8 Задачи для Scrum

Из этого следует понять, что простые системы и хаотичные не требуют вмешательства Agile методологий, но и не отрицает их использование. Да, действительно можно использовать данные подходы в простых процессах, но это будут лишние затраты и возникновение не оправданных рисков. Поэтому в этом случаем специалисты проект менеджмента рекомендуют использовать классическую модель или, например, систему Waterfall(водопад)

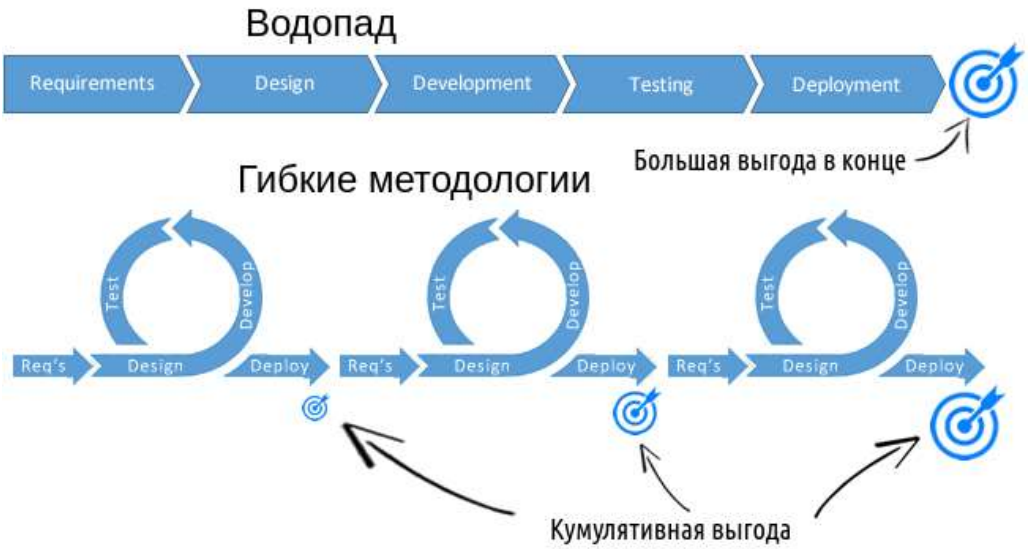


Рис.9 Сравнение Водопада и Agile

Сильные стороны классического проектного менеджмента.

Довольно часто говорится о том, что классический водопадный подход устарел, но он и не думает сдавать позиции. Большим плюсом данного подхода является то, что он требует от Заказчика и руководства компании определить, что же они хотят получить, уже на первом этапе проекта. Раннее включение привносит определённую стабильность в работу проекта, а планирование позволяет упорядочить реализацию проекта. Кроме того, этот подход подразумевает мониторинг показателей и тестирование, что совершенно необходимо для реальных проектов различного масштаба.

Потенциально, классический подход позволяет избежать стрессов ввиду наличия запасного времени на каждом этапе, заложенного на случай каких-либо осложнений и реализации рисков. Кроме того, с правильно проведённым этапом планирования, руководитель проектов всегда знает, какими ресурсами он обладает. Даже если эта оценка не всегда точная.

5. Заключение Мы рассмотрели, что такое Agile и один из самых популярных фреймворков – Scrum. Новый фреймворк позволит создать отличный новый продукт, где не будет потерь времени в ожидании задач. Стоит ли использовать только Agile подходы? Конечно нет. Классические подходы уже не раз показали свою эффективность, поэтому перед использованием нового Фреймворка стоит задуматься, а стоит ли тратить ресурсы на легкие и понятные задачи.

Литература

1. Гузик В. Ф., Гушанский С. М., Касаркин А. В. Использование квантовой запутанности для моделирования параметра согласованности в задачах теории игр // Известия ЮФУ. Технические науки. 2014. №4 (153).
2. Лопатин Д. Н. Agile - новый уровень мотивации в менеджменте // ОмГТУ. 2012. №4
3. Гугаев Кирилл Валерьевич Границы применимости компонентов Scrum // Вестник евразийской науки. 2018. №3
4. Общие ресурсы по Agile Manifesto [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Agile_Manifesto (Дата обращения 20.05.2021).
5. Курс Agile и Scrum в работе над проектами [Электронный ресурс]. URL: https://ru.coursera.org/learn/upravleniya-proektami-agile-scrum (Дата обращения 10.02.2021).
6. Общие ресурсы по исследованию ScrumTrek [Электронный ресурс]. URL: https://fomag.ru/news/issledovanie-scrumtrek-agile-v-rossii-vnedryaetsya-bystro-i-daleko-ne-tolko-v-it/ (Дата обращения 28.04.2021).

Literature

1. Guzik VF, Gushansky SM, Kasarkin AV Use of quantum entanglement for modeling the consistency parameter in problems of game theory // Izvestiya SFU. Technical science. 2014. No. 4 (153).
2. Lopatin DN Agile - a new level of motivation in management // OmSTU. 2012. No. 4

3. Gugaev Kirill Valerievich Limits of applicability of Scrum components // Bulletin of Eurasian Science. 2018. No. 3
4. General resources for Agile Manifesto [Electronic resource]. URL: https://ru.wikipedia.org/wiki/Agile_Manifesto (Date of treatment 05/20/2021).
5. Course Agile and Scrum in the work on projects [Electronic resource]. URL: https://ru.coursera.org/learn/upravleniya-proektami-agile-scrum (Date of treatment 02/10/2021).
6. General resources for research ScrumTrek [Electronic resource]. URL: https://fomag.ru/news/issledovanie-scrumtrek-agile-v-rossii-vnedryaetsya-bystro-i-daleko-ne-tolko-v-it/ (Date of treatment 04/28/2021).