

## **Преимущества использования гибких методологий разработки программного обеспечения**

*Родионова Ольга Игоревна*

*Санкт-Петербургский государственный университет им. Петра Великого  
студент*

*Ушачёв Артём Андреевич*

*Санкт-Петербургский государственный университет им. Петра Великого  
студент*

### **Аннотация**

В статье рассматриваются гибкие методологии разработки программного обеспечения в качестве инструмента для повышения эффективности деятельности компаний и управления проектами в сфере ИТ.

**Ключевые слова:** гибкие методологии, управление проектом, разработка программного обеспечения.

## **Advantages of use of flexible methodologies of software development**

*Rodionova Olga Igorevna*

*Peter the Great Saint-Petersburg Polytechnic University  
student*

*Uschachev Artem Andreevich*

*Peter the Great Saint-Petersburg Polytechnic University  
student*

### **Abstract**

In the article are considered flexible methodologies of software development as the tool for increase in efficiency of activity of the companies and project management in the field of IT.

**Keywords:** flexible methodologies, project management, software development.

На сегодняшний день подходы к управлению проектами сильно отличаются от тех, что впервые использовались во второй половине прошлого века в качестве инструмента для повышения эффективности деятельности компаний. Сегодня невозможно представить себе успешную организацию практически в любой сфере деятельности, в которой не практикуется использование одного или ряда подходов к управлению проектами. Внедрение и использование различных подходов способно качественно повысить эффективность и экономическую отдачу организации.

Ситуация начала приобретать новый характер с развитием информационных технологий. Создание программных продуктов для персональных компьютеров превратилось в самостоятельную сферу промышленности. Потребности заказчиков не перестают расти, а сами проекты становятся все более масштабными с любых точек зрения. Ужесточаются требования, ограничения по времени и ресурсам, растут риски. В таких условиях в организациях, специализирующихся на разработке программного обеспечения возникает острая необходимость во внедрении стандартов и методологий для более эффективной отдачи и предсказуемости результатов.

Использование различных автоматизированных информационных систем для управления проектами способно повысить рентабельность бизнеса, не нанося при этом вреда в виде огромного количества «бумажной» работы. Особенно актуальной сферой для внедрения и использования различных подходов к управлению проектами является ИТ [1]. Связано это со спецификой данного вида деятельности: во-первых разработка программного обеспечения является творческим процессом, в котором попросту отсутствуют четкие пошаговые инструкции для каждого из членов коллектива разработчиков, а во-вторых использование подобных подходов способно наладить правильные схемы коммуникации внутри коллектива разработчиков, что в реальной жизни является очень непростой задачей.

Справиться со второй задачей может помочь внедрение определенного набора формальностей и правил, которые подтолкнут разработчиков выполнять свои обязанности качественно и в срок, но при этом не разрушат дружественный климат в коллективе. Если методология разработки будет выбрана правильно, эта задача будет выполнена.

На сегодняшний день существует огромное количество различных методологий разработки программного обеспечения, учитывающих специфику больших и маленьких команд разработчиков, крупных и мелких проектов и так далее. В связи с этим необходимо каждый раз точно определять, какая именно методология подходит больше всего в данном конкретном случае для данной компании.

Для выбранной отрасли наиболее подходящими являются, так называемые «гибкие» методологии разработки программного обеспечения. Гибкими методологиями разработки (agile методология) называют серию подходов к разработке программного обеспечения, направленных на использование итеративной модели разработки, динамическое формирование новых требований и обеспечение реализации этих требований в результате непрерывного взаимодействия внутри самоорганизующихся рабочих коллективов, состоящих из специалистов различного профиля. Гибкие методологии зарекомендовали себя как эффективная практика организации труда небольших рабочих коллективов, в которых выполняется однородная творческая работа [2]. Они представляют из себя золотую середину между слишком перегруженным процессом разработки (как в тяжеловесных методологиях) и полным его отсутствием.

Подавляющее большинство гибких методологий направлены на минимизацию рисков посредством разбиения процесса разработки на серии коротких циклов, называемых итерациями, (отсюда и происходит название итеративных моделей) которые обычно длятся не больше месяца. Каждая такая итерация выглядит как небольшой самостоятельный проект, на выходе из которого решается очерченный ранее ряд задач. Таким образом, каждая итерация приводит к приросту функциональности будущего продукта. Каждая итерация включает в себя набор базовых этапов: планирование, анализ требований, проектирование, программирование, тестирование и документирование. Итеративность приводит к частой смене версий, что позволяет поддерживать тесный контакт с заказчиком, а также оперативно реагировать на изменения требований. По окончании каждой итерации команда анализирует результаты и проводит переоценку приоритетов разработки.

Гибкие методологии в основном делают упор на непосредственное общение лицом к лицу. Большинство команд, работающих по правилам гибких методологий, расположены в одном рабочем пространстве. Одним из субъектов такого общения и участником рабочего пространства обязательно является представитель заказчика (product owner — заказчик или его полномочный представитель, определяющий требования к продукту). Эту роль может выполнять менеджер проекта, бизнес-аналитик или сам заказчик. В рабочее пространство могут также быть включены тестировщики, дизайнеры интерфейсов, технические писатели и менеджеры, все зависит от масштабов и целей конкретного проекта.

Основной метрикой в гибких методологиях является рабочий продукт. Так как приоритет в этих методологиях отдан непосредственному общению, объём классической письменной документации в таких проектах достаточно мал. Это привело к критике гибких методологий как недисциплинированных.

Ещё на заре своего появления семейство гибких методологий установили ряд принципов, противоречащих классическим методологиям:

- Команды должны сосредотачиваться на людях и налаживании коммуникаций между ними, вместо выстраивания жестких формализованных процессов;
- Команды должны концентрироваться на разрабатываемом продукте, вместо того чтобы писать большие объемы технической и эксплуатационной документации, которую представляет исключительно формальный интерес;
- Исполнитель не должен заставлять заказчика ограничивать контракт жесткими условиями, вместо этого должны выстраиваться здоровые партнерские отношения с целью достижения оптимальных результатов;
- Команда всегда готова к изменениям требований к программному продукту, потому что члены команды понимают, что мир вокруг меняется и то, что месяц назад казалось абсолютно необходимым в проекте, сейчас уже не нужно вообще.

Более формально принципы гибких методологий отражены в тезисах, сформулированных основателями семейства гибких методологий в документе, который получил название Agile Manifesto. Agile Manifesto содержит 4 основные идеи и 12 принципов.

Основные идеи:

- Люди и взаимодействие важнее процессов и инструментов;
- Работающий продукт важнее исчерпывающей документации;
- Сотрудничество с заказчиком важнее согласования условий контракта;
- Готовность к изменениям важнее следования первоначальному плану.

Принципы, которые разъясняет Agile Manifesto:

- Удовлетворение клиента за счёт ранней и бесперебойной поставки ценного программного обеспечения;
- Приветствие изменений требований даже в конце разработки (это может повысить конкурентоспособность полученного продукта);
- Частая поставка рабочего программного обеспечения (каждый месяц или неделю или ещё чаще);
- Тесное, ежедневное общение заказчика с разработчиками на протяжении всего проекта;
- Проектом занимаются мотивированные личности, которые обеспечены нужными условиями работы, поддержкой и доверием;
- Рекомендуемый метод передачи информации — личный разговор (лицом к лицу);
- Работающее программное обеспечение — лучший измеритель прогресса;
- Спонсоры, разработчики и пользователи должны иметь возможность поддерживать постоянный темп на неопределённый срок;
- Постоянное внимание улучшению технического мастерства и удобному дизайну;
- Простота — искусство не делать лишней работы;
- Лучшие технические требования, дизайн и архитектура получаются у самоорганизованной команды;
- Постоянная адаптация к изменяющимся обстоятельствам.

Подводя итог, можно сказать, что гибкие методологии скорее адаптивны, а не предсказуемы. Для тяжеловесных монументальных методологий необходимо детальное планирование большого объема разработок, и такой подход работает - но до тех пор, пока не начнутся изменения, ведущие появлению новых или к упразднению намеченных ранее задач. Отсюда и идет естественное сопротивление изменениям тяжеловесными методологиями. Гибкие же методологии, в свою очередь, приветствуют и безболезненно адаптируются к изменениям. В отличие от тяжеловесных, они были задуманы как процессы, которые адаптируют изменения и только выигрывают от них, даже в том случае, когда изменения

происходят в них самих. Также гибкие методологии ориентированы на человека, а не на процесс. В них ясно заявлено о необходимости учитывать в работе природные качества человеческой натуры, а не действовать им наперекор. Кроме этого в гибких методологиях особо подчеркивается, что работа по созданию программных продуктов должна приносить удовольствие.

### **Библиографический список**

1. Разу М. А., Бронникова Т. М., Разу Б. М., Титов С. А., Якутин Ю. В. Управление проектом. Основы проектного управления: учебник / кол. авт.; под ред. проф. М. А. Разу. М.: КНОРУС, 2006 768 с.
2. Товб А. С., Ципес Г. Л. Управление проектами: стандарты, методы, опыт. М.: ЗАО «Олимп-Бизнес», 2003. 240 с.