

Содержание:

image not found or type unknown



Гибкая методология разработки программного обеспечения

Гибкая методология разработки программного обеспечения ориентирована на использование итеративного подхода, при котором программный продукт создается постепенно, небольшими шагами, включающими реализацию определенного набора требований. При этом предполагается, что требования могут изменяться. Команды, использующие гибкие методологии, формируются из универсальных разработчиков, которые выполняют различные задачи в процессе создания программного продукта.

При использовании гибких методологий минимизация рисков осуществляется путём сведения разработки к серии коротких циклов, называемых итерациями, продолжительностью 2 -3 недели. Итерация представляет собой набор задач, запланированных на выполнение в определенный период времени. В каждой итерации создается работоспособный вариант программной системы, в которой реализуются наиболее приоритетные (для данной итерации) требования заказчика. На каждой итерации выполняются все задачи, необходимые для создания работоспособного программного обеспечения: планирование, анализ требований, проектирование, кодирование, тестирование и документирование. Хотя отдельная итерация, как правило, недостаточна для выпуска новой версии продукта, подразумевается, что текущий программный продукт готов к выпуску в конце каждой итерации. По окончании каждой итерации команда выполняет переоценку приоритетов требований к программному продукту, возможно, вносит коррективы в разработку системы.

Принципы и значение гибкой разработки

Для методологии гибкой разработки декларированы ключевые постулаты, позволяющие командам достигать высокой производительности:

- люди и их взаимодействие;
- доставка работающего программного обеспечения;
- сотрудничество с заказчиком;
- реакция на изменение.

Люди и взаимодействие. Люди - важнейшая составная часть успеха. Отдельные члены команды и хорошие коммуникации важны для высокопроизводительных команд. Для содействия коммуникации гибкие методы предполагают частые обсуждения результатов работы и внесение изменений в решения. Обсуждения могут проводиться ежедневно длительностью несколько минут и по завершению каждой итерации с анализом результатов работ и ретроспективой. Для эффективных коммуникаций при проведении собраний участники команд должны придерживаться следующих ключевых правил поведения:

- уважение мнения каждого участника команды;
- быть правдивым при любом общении;
- прозрачность всех данных, действий и решений;
- уверенность, что каждый участник поддержит команду;
- приверженность команде и ее целям.

Для создания высокопроизводительных команд в гибких методологиях кроме эффективной команды и хороших коммуникаций необходим совершенный программный инструментарий.

Работающее программное обеспечение важнее всеобъемлющей документации. Все гибкие методологии выделяют необходимость доставки заказчику небольших фрагментов работающего программного обеспечения через заданные интервалы. Программное обеспечение, как правило, должно пройти уровень модульного тестирования, тестирования на уровне системы. При этом объем документации должен быть минимальным. В процессе проектирования команда должна поддерживать в актуальном состоянии короткий документ, содержащий обоснования решения и описание структуры.

Сотрудничество с заказчиком важнее формальных договоренностей по контракту. Чтобы проект успешно завершился, необходимо регулярное и частое общение с заказчиком. Заказчик должен регулярно участвовать в обсуждении принимаемых решений по программному обеспечению, высказывать свои пожелания и замечания. Вовлечение заказчика в процесс разработки программного обеспечения необходимо создания качественного продукта.

Оперативное реагирование на изменения важнее следования плану. Способность реагирования на изменения во многом определяет успех программного проекта. В процессе создания программного продукта очень часто изменяются требования заказчика. Заказчики очень часто точно не знают, чего хотят, до тех пор, пока не увидят работающее программное обеспечение. Гибкие методологии ищут обратную связь от заказчиков в процессе создания программного продукта. Оперативное реагирование на изменения необходимо для создания продукта, который удовлетворит заказчика и обеспечит ценность для бизнеса.

Постулаты гибкой разработки поддерживаются 12 принципами. В конкретных методологиях гибкой разработки определены процессы и правила, которые в большей или меньшей степени соответствуют этим принципам. Гибкие методологии создания программных продуктов основываются на следующих принципах:

1. Высшим приоритетом считать удовлетворение пожеланий заказчика посредством поставки полезного программного обеспечения в сжатые сроки с последующим непрерывным обновлением. Гибкие методики подразумевают быструю поставку начальной версии и частые обновления. Целью команды является поставка работоспособной версии с течением нескольких недель с момента начала проекта. В дальнейшем программные системы с постепенно расширяющейся функциональностью должны поставляться каждые несколько недель. Заказчик может начать промышленную эксплуатацию системы, если посчитает, она достаточно функциональна. Также заказчик может просто ознакомиться с текущей версией программного обеспечения, предоставить свой отзыв с замечаниями.
2. Не игнорировать изменение требований, пусть даже на поздних этапах разработки. Гибкие процессы позволяют учитывать изменения для обеспечения конкурентных преимуществ заказчика. Команды, использующие гибкие методики, стремятся сделать структуру программы качественной, с минимальным влиянием изменений на систему в целом.
3. Поставлять новые работающие версии ПО часто, с интервалом от одной недели до двух месяцев, отдавая предпочтение меньшим срокам. При этом ставится цель поставить программу, удовлетворяющую потребностям пользователя, с минимумом сопроводительной документации.
4. Заказчики и разработчики должны работать совместно на протяжении всего проекта. Считается, что для успешного проекта заказчики, разработчики и все заинтересованные лица должны общаться часто и по многу для

целенаправленного совершенствования программного продукта.

5. Проекты должны воплощать в жизнь целеустремленные люди. Создавайте команде проекта нормальные условия работы, обеспечьте необходимую поддержку и верьте, что члены команды доведут дело до конца.
6. Самый эффективный и продуктивный метод передачи информации команде разработчиков и обмена мнениями внутри неё - разговор лицом к лицу. В гибких проектах основной способ коммуникации - простое человеческое общение. Письменные документы создаются и обновляются постепенно по мере разработки ПО и только в случае необходимости.
7. Работающая программа - основной показатель прогресса в проекте. О приближении гибкого проекта к завершению судят по тому, насколько имеющаяся в данный момент программа отвечает требованиям заказчика.
8. Гибкие процессы способствуют долгосрочной разработке. Заказчики, разработчики и пользователи должны быть в состоянии поддерживать неизменный темп сколь угодно долго.
9. Непрестанное внимание к техническому совершенству и качественному проектированию повышает отдачу от гибких технологий. Члены гибкой команды стремятся создавать качественный код, регулярно проводя рефакторинг.
10. Простота - искусство достигать большего, делая меньше. Члены команды решают текущие задачи максимально просто и качественно. Если в будущем возникнет какая-либо проблема, то в качественный код имеется возможность внести изменения без больших затрат.
11. Самые лучшие архитектуры, требования и проекты выдают самоорганизующиеся команды. В гибких командах задачи поручаются не отдельным членам, а команде в целом. Команда сама решает, как лучше всего реализовать требования заказчика. Члены команды совместно работают над всеми аспектами проекта. Каждому участнику разрешено вносить свой вклад в общее дело. Нет такого члена команды, который единолично отвечал бы за архитектуру, требования или тесты.
12. Команда должна регулярно задумываться над тем, как стать ещё более эффективной, а затем соответственно корректировать и подстраивать свое поведение. Гибкая команда постоянно корректирует свою организацию, правила, соглашения и взаимоотношения.

Вышеприведенным принципам, в определенной степени, соответствуют ряд методологий разработки программного обеспечения:

AgileModeling

набор понятий, принципов и приёмов (практик), позволяющих быстро и просто выполнять моделирование и документирование в проектах разработки программного обеспечения;

AgileUnifiedProcess(AUP)

упрощенная версия IBM RationalUnifiedProcess(RUP), которая описывает простое и понятное приближение (модель) для создания программного обеспечения для бизнес-приложений;

OpenUP

это итеративно-инкрементальный метод разработки программного обеспечения. Позиционируется как лёгкий и гибкий вариантRUP;

AgileDataMethod

группа итеративных методов разработки программного обеспечения, в которых требования и решения достигаются в рамках сотрудничества разных кросс-функциональных команд;

DSDM

методика разработки динамических систем, основанная на концепции быстрой разработки приложений (RapidApplicationDevelopment, RAD). Представляет собой итеративный и инкрементный подход, который придаёт особое значение продолжительному участию в процессе пользователя/потребителя;

Extremeprogramming (XP)

экстремальное программирование;

Adaptive software development (ADD)

адаптивная разработка программ;

Featurerivenddevelopment (FDD)

разработка, ориентированная на постепенное добавление функциональности;

GettingReal

итеративный подход без функциональных спецификаций, использующийся для веб-приложений;

MSFfogAgileSoftwareDevelopment

гибкая методология разработки ПО компании Microsoft;

Scrum

устанавливает правила управления процессом разработки и позволяет использовать уже существующие практики кодирования, корректируя требования или внося тактические изменения.

Ключевые термины

Гибкая методология разработки программного обеспечения

методология, ориентированная на использование итеративного подхода, при котором программный продукт создается постепенно, небольшими шагами, включающими реализация определенного набора требований.

Итерация

набор задач, запланированных на выполнение определенный период времени.

Вывод

Гибкая методология разработки программного обеспечения ориентирована на использование итеративного подхода, при котором программный продукт создается постепенно. Программный продукт создается за несколько итераций, включающих реализацию определенного набора требований. Итерации имеют длительность 2 -3 недели. Результатом итерации является промежуточный вариант работоспособного программного обеспечения. Для методологии гибкой разработки декларированы ключевые постулаты и принципы. Принципам гибкой разработки ПО, в определенной степени, соответствуют ряд методологий.