Гибкая методология разработки — Agile

JSP & Servlets
15 уровень, 2 лекция

ОТКРЫТА

Agile Model

Гибкая (Agile) методология помогает снизить риск при разработке ПО за счет перевода рабочего процесса в несколько небольших циклов. Такие циклы называются итерациями, как правило они длятся от двух до трех недель.

Итерация похожа на небольшой программный проект, который состоит из заданий, каждое из которых улучшает функциональность. К ним относятся: составление плана, оценка требований, согласование проекта, написание кода, тестирование и создание технической документации.

Одной итерации обычно недостаточно для полноценного релиза ПО. Однако, Agile хорош тем, что небольшие части проекта готовы к оценке в конце каждой итерации. Это позволяет участникам команды менять приоритеты для дальнейшей работы, не дожидаясь финального релиза.

Применяя “гибкую” методологию разработки можно видеть конкретный результат уже после каждой итерации. То есть, разработчик может понять, соответствует ли результат его работы требованиям или нет. Это один из важных плюсов гибкой модели.

Что касается минусов, то при использовании Agile иногда трудно оценить затраты трудовых ресурсов и бюджет проекта. Если же брать варианты практического применения гибкой модели, то наиболее известным среди них считается экстремальное программирование (XP).

В основу XP входят краткие встречи участников команды, которые проходят каждый день, и регулярные собрания (раз в неделю или реже). На ежедневных митингах (daily standup) обычно обсуждаются:

- текущие результаты работы;
- перечень задач, которые нужно выполнить каждому участнику команды;
- возникшие трудности и варианты их разрешения.

Манифест

Agile представляет собой целое направление в разработке, поэтому правила работы по нему декларированы в специальном документе — Agile Manifesto. Сюда входят как практики, так и принципы, по которым должна работать команда.

Agile Manifesto состоит из 4-х основополагающих идей и 12 принципов.

Основные идеи:

- сотрудничество между разработчиками важнее, чем инструменты;
- рабочая версия продукта имеет приоритет над документацией;
- взаимопонимание между командой и заказчиком важнее чем условия контракта;
- первоначальный план всегда можно изменить, если это необходимо.

Что касается 12 принципов Agile, то вот они:

- главным приоритетом является соответствие готовой программы ожиданиям заказчика;
- изменение условий допускается на любом этапе, даже на финале разработки (если это способно повысить качество и конкурентоспособность ПО);
- регулярная поставка рабочих версий программного продукта (каждые 14 дней, месяц или ежеквартально);
- залог успеха — регулярное взаимодействие между заказчиком и разработчиками (желательно ежедневное);

- проекты нужно строить среди тех, кто в них заинтересован, таких людей нужно обеспечить необходимыми условиями для работы и всяческой поддержкой;
- лучший способ обмена информацией в команде — личная встреча;
- рабочая версия ПО — лучший показатель прогресса;
- все заинтересованные лица должны иметь возможность поддерживать нужный темп работы в течение всего процесса создания ПО;
- техническое усовершенствование и хорошее проектирование улучшают гибкость;
- важно придерживаться простоты и не создавать лишнее;
- лучшие результаты получаются у тех команд, которые умеют самоорганизовываться;
- участники команды должны регулярно думать о способах улучшения своей эффективности путем изменения рабочего процесса.

Согласно манифесту Agile, хороший процесс разработки ПО напрямую зависит от людей, которые задействованы в этом процессе. Для этого нужно как можно более эффективно организовать их взаимодействие, создать максимально организованную команду.

Методологии

В Agile Manifesto существует также несколько методологий, которые объясняют ценности и принципы:

- Agile Modeling;
- Agile Unified Process;
- Agile Data Method;
- Rapid Application Development (DSDM);
- Essential Unified Process;
- Extreme programming;
- Feature driven development;
- Getting Real;
- OpenUP;
- Scrum.

Agile Modeling — это перечень принципов, терминов и практик, использование которых ускоряет и упрощает разработку моделей ПО и документацию.

Цель Agile Modeling состоит в улучшении моделирования и создании документации. Важно учесть, что сюда не входит кодирование, тестирование или вопросы, связанные с контролем над проектом, развертыванием и поддержкой. Однако эта методология включает проверку кода.

Agile Unified Process — методология, которая упрощает для пользователей приближение (модель). Обычно применяется для разработки коммерческого софта.

Agile Data Method — несколько похожих методологий, в которых условия заказчика достигаются благодаря сотрудничеству нескольких команд.

DSDM — этот подход отличается от остальных тем, что в нем наряду с разработчиками активное участие принимают пользователи будущего продукта.

Feature driven development — методология разработки, имеющая временное ограничение: “каждую функцию нужно реализовывать не дольше, чем за две недели”.

Стоит учесть, что если сценарий использования невелик, его можно считать функцией. Если же он существенный, то его нужно разделить на несколько функций.

Getting Real — итеративная методология, при которой сначала ведется разработка интерфейса программы, а лишь затем ее функционала.

OpenUP — метод разработки, который разделяет цикл проекта на четыре стадии: начальную, стадию уточнения, конструирования и передачи.

Согласно принципам Agile, независимо от длительности работы нужно обеспечить всем заинтересованным лицам и участникам команды способ ознакомления и принятия решений. Благодаря этому можно эффективно осуществлять контроль

над ситуацией и вовремя оценивать промежуточные результаты. План проекта определяет жизненный цикл, а итоговым результатом следует считать стабильный релиз приложения.

Что касается Scrum, то он регулирует правила управления процессом разработки и позволяет применять уже имеющиеся практики кодирования с возможностью корректировки условий или внесением изменений. Использование этой методологии позволяет увидеть и устранить отклонения от ожидаемого результата на ранних стадиях разработки.

Давайте ознакомимся с этим чуть подробнее…

‹ Предыдущая лекция

Следующая лекция ›

RUSH

JavaRush — это интерактивный онлайн-курс по изучению Java-программирования с нуля. Он содержит 1200 практических задач с проверкой решения в один клик, необходимый минимум теории по основам Java и мотивирующие фишки, которые помогут пройти курс до конца: игры, опросы, интересные проекты и статьи об эффективном обучении и карьере Java-девелопера.

ПОДПИСЫВАЙТЕСЬ

ЯЗЫК ИНТЕРФЕЙСА

Русский

СКАЧИВАЙТЕ НАШИ ПРИЛОЖЕНИЯ

ДОСТУПНО В Google Play

Загрузите в App Store