ЛЕКЦИЯ 33

Тема: Основные методы применения практики в XP разработке
Цель: изучить стратегии проектирования программного продукта

13 ПРАКТИК ЭКСТРЕМАЛЬНОГО ПРОГРАММИРОВАНИЯ

XP — строго упорядоченный процесс.

Простые решения, имеющие высший приоритет, в отличие от проектных решений, которые пока не нужны, а могут (в условиях изменения требований и операционной среды) и вообще не понадобиться.

1. **Вся команда-** обязательно входит представитель заказчика. В нее обязательно входит представитель заказчика. Заказчик выдвигает требования к продукту и расставляет приоритеты в реализации функциональности. Со стороны исполнителей в команду входят разработчики и тестировщики, направляющий команду, и менеджер, который обеспечивает команду ресурсами.

Все участники проекта с применением XP работают как одна команда. Заказчики которые ставят задачи(что нужно выполнить) и исполнители, которые реализуют эти задачи с применением метода экстремального программирования.

2. **Игра планирования** — быстрое определение области действия следующей реализации путем объединения деловых приоритетов и технических оценок. Заказчик формирует область действия, приоритетность и сроки с точки зрения бизнеса, а разработчики оценивают и прослеживают продвижение (прогресс).

Планирование в XP проводят в два этапа — *планирование релиза и планирование итераций*. Если команда не успевает выполнить все задачи к дате релиза, то релиз не отодвигается, а режется часть функционала, наименее важная для заказчика т.е. просто сокращается часть ф-й.

3. **Маленькие релизы версий** (Small releases) — быстрый запуск в производство простой системы. Новые версии реализуются в очень коротком (двухнедельном) цикле.

В XP версии выпускаются часто, но с небольшим функционалом.

Во-первых, маленький объем функциональности легко тестировать и сохранять работоспособность всей системы.

Во-вторых, каждую итерацию заказчик получает часть функционала, несущую бизнес-ценность.

4. **Пользовательские тесты**- Заказчик сам определяет автоматизированные приемочные тесты, чтобы проверить работоспособность очередной функции продукта. Команда пишет эти тесты и использует их для тестирования готового кода.

Заказчик сам решает, что он хочет тестировать и как. Т.е. он может потребовать тесты в определяет объёме только по функционалу основных компонентов или только к пользовательскому интерфейсу. А уже разработчики реализуют эти тесты

5. **Коллективное владение кодом** — любой разработчик может улучшать любой код системы в любое время.

В XP любой разработчик может править любой кусок кода, т.к. код не закреплен за своим автором. Кодом владеет вся команда.

6. **Непрерывная интеграция кода**— система интегрируется и строится много раз в день, по мере завершения каждой задачи. Непрерывное регрессионное тестирование, то есть повторение предыдущих тестов, гарантирует, что изменения требований не приведут к регрессу функциональности.

Это значит, что новые части кода сразу же встраиваются в систему — команды XP заливают в репозиторий новый билд каждые несколько часов и чаще.

Во-первых, сразу видно, как последние изменения влияют на систему. Если новый кусок кода что-то сломал, то ошибку найти и исправить в разы проще, чем спустя неделю.

Во-вторых, команда всегда работает с последней версией системы.

7. **Стандарты кодирования** — должны выдерживаться правила, обеспечивающие одинаковое представление программного кода во всех частях программной системы. Можно создать свои стандарты или использовать готовые.

Когда кодом владеют все, важно принять единые стандарты оформления, чтобы код выглядел так, как будто он написан одним профессионалом. Можно выработать свои стандарты или принять готовые.

8. **Метафора системы** — вся разработка проводится на основе простой, общедоступной истории о том, как работает вся система.

Метафора системы — это ее сравнение с чем-то знакомым, чтобы сформировать у команды общее видение.

Обычно метафору системы продумывает тот, кто разрабатывает архитектуру и представляет систему целиком.

9. **Устойчивый темп** (40-часовая неделя) — как правило, работают не более 40 часов в неделю. Нельзя удваивать рабочую неделю за счет сверхурочных работ.

XP команды работают на максимуме продуктивности, сохраняя устойчивый темп.

При этом экстремальное программирование *негативно относится к переработкам* и пропагандирует 40-часовую рабочую неделю. Т.е. в день не более 8 часов

10. **Разработка основанная на Тестирование** — непрерывное написание тестов самими программистами для определенных модулей, до написания самого кода; функционал покрывается тестами на 100%. заказчики пишут тесты для демонстрации законченности функций.

Самая сложная часть. В XP тесты пишутся самими программистами, причем ДО написания кода, который нужно протестировать.

При таком подходе каждый кусок функционала будет покрыт тестами на 100%.

Когда пара программистов заливают код в репозиторий, сразу запускаются модульные тесты. И ВСЕ они должны сработать. Тогда разработчики будут уверены, что движутся в правильном направлении.

11. **Парное программирование** — весь код пишется двумя программистами, работающими на одном компьютере.

Представьте двух разработчиков за одним компьютером, работающих над одним куском функциональности продукта.

Из двух вариантов решения проблемы выбирается лучший, код оптимизируется сразу же, ошибки отлавливаются еще до их совершения.

В итоге имеем чистый код, в котором хорошо разбираются сразу двое разработчиков.

12. **Простой дизайн**— проектирование выполняется настолько просто, насколько это возможно в данный момент, не пытаемся предугадать будущий функционал.

делать только то, что нужно сейчас, не пытаясь угадать будущую функциональность.

13. **рефакторинг —** это процесс постоянного реконструирование дизайна системы, чтобы привести его в соответствие новым требованиям. Рефакторинг включает удаление дублей кода, повышение связности и снижение сопряжения.

Постоянное улучшение дизайна т.е. постоянные рефакторинги, поэтому дизайн кода всегда остается простым.

Простой дизайн и непрерывный рефакторинг дают синергетический эффект — когда код простой, его легко оптимизировать.

Контрольные вопросы:

1. что такое рефакторинг?

2. Что такое игра планирования?

3. Что такое стандарты кодирования?

4. Метафора системы?

5. Что подразумевается под устойчивым темпом?

6. Что подразумевает разработка основанная на тестировании?

7. Как происходит парное программирование?

8. Что такое пользовательские тесты?

9. Что подразумевается под маленькими релизами версий программы?

10. Практики экстремального программирования, опишите

Список использованных источников:

1. Технологии разработки программного обеспечения С.А. Орлов
2. Технологии разработки программного обеспечения В.В. Бахтизин, Л.А. Глухова
3. Искусство IT-проектирования Скотт Беркун
4. https://finswin.com/projects/osnovnye/sroki-realizacii-proekta.html