HEICA

TEMPUS

Организация процесса конструирования

HEICA

Цели

- Стратегии конструирования
- Тяжеловесные и облегченные процессы
- XP- процесс

Акронимы

- XP –Экстремальное программирование

Стратегии конструирования ПО

Характеристики стратегий

Стратегия конструирования	В начале процесса определены все требования?	Множество циклов конструирования?	Промежуточное ПО распространяется?
Однократный проход	Да	Нет	Нет
Инкрементная (запланированное улучшение продукта)	Да	Да	Может быть
Эволюционная	Нет	Да	Да

Используемые модели и процессы:
Водопадная,
Инкрементная,
RAD,
Спиральная,
Компонентно-ориентированная
RUP

Тяжеловесные
и
облегченные процессы

Тяжеловесные процессы

- Традиционно для упорядочения и ускорения программных разработок предлагались строго упорядочивающие тяжеловесные (heavyweight) процессы.
- В этих процессах прогнозируется весь объем предстоящих работ, поэтому они называются прогнозирующими (predictive) процессами. Порядок, который должен выполнять при этом человек-разработчик, чрезвычайно строг — «шаг вправо, шаг влево — виртуальный расстрел!».

Облегченные процессы (1)

- К 2000 году существовало уже целое множество так называемых легковесных (lightweight) методологий, привлекательные отсутствием бюрократизма, характерного для тяжеловесных (прогнозирующих) процессов. Новые процессы воплотили в жизнь разумный компромисс между слишком строгой дисциплиной и полным ее отсутствием. Иначе говоря, порядка в них достаточно для того, чтобы получить разумную отдачу от разработчиков.
- В 2001 году группа создателей и экспертов по различным легковесным методологиям провела семинар, на котором были сформулированы основные принципы гибкой разработки ПО (так называемый AgileManifesto). На том же семинаре было предложено новое название легковесных методологий – гибкая разработка (agilesoftwaredevelopment).

Облегченные процессы (2)

Общими особенностями гибких методологий являются:

- Ориентированность на людей – как разработчиков, так и заказчиков (согласие участвовать в разработке). Считается, что умение собрать в проектной команде «правильных» людей определяет успех или неудачу проекта в значительно большей степени, чем любые процессы или технологии.
- Использование устных обсуждений вместо формальных спецификаций везде, где это возможно.
- Итеративная разработка с возможно более короткой (в разумных пределах) продолжительностью итерации, при этом в результате каждой итерации выпускается полноценная работающая версия продукта.
- Ожидание изменений – в гибком процессе проектная команда не пытается зафиксировать требования в начале проекта и затем следовать жестко определенному плану. Изменения могут быть сделаны на сколь угодно позднем этапе проекта.

XP - процес

XP - процесс (1)

- Методология XP была создана Кентом Беком (KentBeck) в 1996 году в ходе попытки спасти провальный проект по разработке системы расчета зарплаты для компании Крайслер.
- XP наследует все общие принципы гибких методологий, достигая их при помощи двенадцати инженерных практик. Ниже описаны самые интересные из специфических технологий и практик XP:

 В проектной команде должен постоянно работать так называемый представитель заказчика – он обладает детальной информацией о необходимой функциональности, определяет приоритеты отдельных требований, оценивает качество создаваемой системы. Технически, представитель заказчика может быть и сотрудником фирмы разработчика – менеджером продукта, бизнес-аналитиком и т.п.

XP - процесс (2)

- Пользовательские истории – короткие неформальные описания прецедентов использования системы. В XP истории являются основным и, вместе с приемочными тестами, единственным средством спецификации требований. Поскольку истории очень лаконичны, участникам проекта обычно требуются более детальная информация по функциональности системы – они получают ее непосредственно от представителя заказчика.
- Версии генерируются каждые две недели, поэтому разработчики и заказчик должны прийти к соглашению о том, какие истории будут осуществлены в пределах двух недель. Полную функциональность, требуемую заказчику, характеризует пул историй; но для следующей двухнедельной итерации из пула выбирается подмножество историй, наиболее важное для заказчика.

XP - процесс (3)

- Разработка через тестирование (testdrivendevelopment) – в XP становится особенно важным, чтобы весь создаваемый код был покрыт автоматическими юнит-тестами (почему это так, станет понятно дальше). Этого можно добиться при помощи простого правила – новый код может быть написан исключительно для того, чтобы увеличить число успешно проходящих юнит-тестов. Фактически это означает, что перед реализацией новой функции разработчик должен создать соответствующий тест, а написание кода должно завершиться в тот момент, когда начнет проходить новый, а также все существующие тесты. Если после этого окажется, что новая функция реализована не полностью, необходимо создать еще один тест и повторить весь цикл заново.

XP - процесс (4)

- Архитектура системы должна быть максимально простой. XP не рекомендует проектировать в расчете на будущее развитие системы; идеальная архитектура должна не более чем поддерживать существующую функциональность. Цель такого минималистичного подхода к проектированию – избежать бесполезных инвестиций в архитектурные решения, которые часто оказываются выброшенными после очередного изменения требований. Вместо этого архитектура постоянно изменяется и развивается вместе с системой.

HEICA

TEMPUS

XP - процесс (4)

- Постоянное изменение архитектуры требует постоянной переработки и улучшения кода – рефакторинга. В XP поощряется коллективное владение кодом – увидев возможность улучшения в любом компоненте системы, разработчик может провести необходимые рефакторинги вне зависимости от того, кто является основным разработчиком компонента. Возможные ошибки, внесенные рефакторингом, должны быть тут же обнаружены автоматическими тестами.
- Все изменения, сделанные разработчиками, после автоматического тестирования практически сразу попадают в основной репозиторий. Таким образом этап интеграции как таковой отсутствует или, что то же самое, происходит постоянно. XP называет эту практику непрерывной интеграцией.

XP - процесс (4)

- Использование парного программирования - не означает, что на двух разработчиков в организации должен быть выделен только один компьютер, однако большая часть написания кода должна проходить в парах. Считается, что при этом общая эффективность разработки повышается за счет более продуманных решений, меньшего количества ошибок, тщательного написания юнит тестов и т.п.
- Продолжительность рабочей недели не должна превышать 40 часов. По сравнению с обычной практикой постоянных переработок, в средне- и долгосрочной перспективе это повышает производительность проектной команды за счет уменьшения стресса и переутомления.

Жизненный цикл проекта в XP (1)

- Жизненный цикл проекта в XP состоит из последовательности релизов. Каждый релиз – это полноценная версия продукта, которую может использовать заказчик, и содержащая дополнительную функциональность по сравнению с предыдущим релизом. Релиз появляется в результате одной или нескольких итераций, длящихся от одной до четырех недель.
- В XP не рекомендуется тратить много времени на планирование; сам процесс планирования называется игрой (planninggame). Подробный план составляется только на очередную итерацию и ближайшие один-два релиза.
- Планирование релиза состоит из следующих шагов:

Жизненный цикл проекта в XP (2)

- Заказчик формулирует свои требования в виде историй, которые оцениваются по трудоемкости разработчиками. Оценки трудоемкости делаются в так называемых идеальных днях – времени, которое некий воображаемый разработчик потратит на реализацию истории при полном отсутствии отвлекающих факторов, параллельных задач, перерывов и т.п.
- Истории сортируются по приоритету, рискам, сложности реализации.
- Определяется фактическая производительность команды (какое число реальных дней соответствует идеальному дню).
- На основании фактической производительности определяется, какие истории войдут в очередной релиз и за какое время он будет завершен.

Жизненный цикл проекта в XP (3)

Итерация планируется аналогичным образом.

- Предварительно определяется набор историй для итерации.
- Истории разбиваются на задачи (tasks), которые распределяются между разработчиками. В отличие от историй, которые описывают поведение системы с точки зрения пользователя, задачи составляются на техническом уровне, например «модифицировать схему базы данных» или «провести рефакторинг».

Жизненный цикл проекта в XP (4)

- Разработчики оценивают свои задачи. Казалось бы, это ненужное повторение операции оценки историй, однако на этом этапе уточненные оценки должны быть более реалистичными. Поскольку задачи теперь оценивают их непосредственные исполнители, в оценках учитывается индивидуальная производительность, зависящая от опыта, знакомства с используемыми технологиями и т.п.
- На основе уточненных оценок задач подсчитывается общая загрузка команды. В зависимости от загрузки некоторые истории могут быть перенесены на следующую итерацию или, наоборот, добавлены в текущую.
- Периодически в ходе проекта измеряется фактическая производительность команды. Если она начинает сильно отличаться от значения, которое было использовано при планировании, график выхода релизов должен быть пересмотрен.

Жизненный цикл проекта в XP (5)

XP рекомендует:

- первая реализация должна иметь длительность 2-6 месяцев,
- продолжительность остальных реализаций — около двух месяцев,
- каждая итерация длится приблизительно две недели,
- численность группы разработчиков не превышает 10 человек.
- XP-процесс для проекта с семью реализациями, осуществляемый за 15 месяцев