ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ ГОРОДА МОСКВЫ

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

**ПЕРВЫЙ МОСКОВСКИЙ ОБРАЗОВАТЕЛЬНЫЙ КОМПЛЕКС**

# **Выбор и обоснование модели проектирования**

**программного обеспечения в виде приложения и базы данных для Marathon Skills 2016**

*Могучев Андрей Александрович*

Студент

Группа 22 ИС

**Москва,**

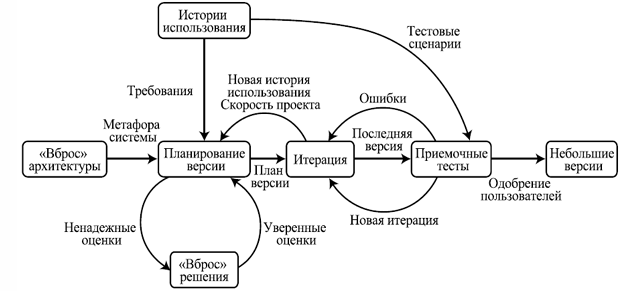
**2019– 2020 учебный год**

При проектировании программного обеспечения в виде приложения и базы данных для Marathon Skills 2016 рекомендуется использовать **Экстремальное программирование**(англ. Extreme Programming, XP) — одну из гибких методологий разработки программного обеспечения.

Методология ХР позволяет применить полезные традиционные методы и практики разработки программного обеспечения.

Практика выполнения ревизии кода, заключающая в проверке одним программистом кода, написанного другим программистом, в «экстремальном» варианте представляет собой «парное программирование», когда один программист занимается кодированием, а его напарник в это же время непрерывно просматривает только что написанный код.

Объединение нескольких специалистов и подключение их в качестве соавторов в общем репозитории позволит исполнить несколько приемов методологии ХР и получить преимущество перед другими методологиями.



**Основные приемы**

Двенадцать **основных приёмов** экстремального программирования могут быть объединены в четыре группы:

***Короткий цикл обратной связи*** (Fine-scale feedback)

— *Разработка через тестирование* (Test-driven development). Заказчики заранее пишут тесты, демонстрирующие основные возможности системы, чтобы можно было увидеть, что система действительно заработала.

— *Игра в планирование* (Planning game) — быстро сформировать приблизительный план работы и постоянно обновлять его по мере того, как условия задачи становятся всё более чёткими.

— *Заказчик всегда рядом* (Whole team, Onsite customer) — XP утверждает, что заказчик должен быть всё время на связи и доступен для вопросов.

— *Парное программирование* (Pair programming) — предполагает, что весь код создается парами программистов, работающих за одним компьютером либо в общем репозитории. Один из них работает непосредственно с текстом программы, другой просматривает его работу и следит за общей картиной происходящего.

***Непрерывный, а не пакетный процесс***

— *Непрерывная интеграция* (Continuous integration) — В XP интеграция кода всей системы выполняется несколько раз в день после того, как разработчики убедились в том, что все тесты модулей корректно срабатывают.

— *Рефакторинг* (Design improvement, Refactoring) — XP подразумевает, что однажды написанный код в процессе работы над проектом почти наверняка будет неоднократно переделан.

Тесты модулей позволяют разработчикам убедиться в том, что их код работает корректно. Они также помогают другим разработчикам понять, зачем нужен тот или иной фрагмент кода и как он функционирует. Тесты модулей также позволяют разработчику без каких-либо опасений выполнять рефакторинг. Разработчики XP переделывают написанный ранее код для того, чтобы улучшить его.

— *Частые небольшие релизы* (Small releases) — версии (releases) продукта должны поступать в эксплуатацию как можно чаще. Работа над каждой версией должна занимать как можно меньше времени. При этом каждая версия должна быть достаточно осмысленной с точки зрения полезности для бизнеса.

***Понимание, разделяемое всеми***

— *Простота* (Simple design) — XP исходит из того, что в процессе работы условия задачи могут неоднократно измениться, а значит, разрабатываемый продукт не следует проектировать заблаговременно целиком и полностью. Проектирование должно выполняться небольшими этапами, с учётом постоянно изменяющихся требований. В каждый момент времени следует пытаться использовать наиболее простой дизайн, который подходит для решения текущей задачи, и менять его по мере того, как условия задачи меняются.

— *Метафора системы* (System metaphor) — Архитектура — это представление о компонентах системы и их взаимосвязях между собой. Разработчикам необходимо проводить анализ архитектуры программного обеспечения для того, чтобы понять, в каком месте системы необходимо добавить новую функциональность, и с чем будет взаимодействовать новый компонент.

Метафора системы (system metaphor) — это аналог того, что в большинстве методик называется архитектурой. Метафора системы даёт команде представление о том, каким образом система работает в настоящее время, в каких местах добавляются новые компоненты, и какую форму они должны принять.

— *Коллективное владение кодом* (Collective code ownership) или выбранными шаблонами проектирования (Collective patterns ownership) -означает, что каждый член команды несёт ответственность за весь исходный код. Таким образом, каждый вправе вносить изменения в любой участок программы. Парное программирование поддерживает эту практику: работая в разных парах, все программисты знакомятся со всеми частями кода системы. Важное преимущество коллективного владения кодом — в том, что оно ускоряет процесс разработки, поскольку при появлении ошибки её может устранить любой программист.

*— Стандарт кодирования* (Coding standard or Coding conventions) — в рамках XP необходимо добиться того, чтобы было сложно понять, кто является автором того или иного участка кода, — вся команда работает унифицировано, как один человек. Команда должна сформировать набор правил, а затем каждый член команды должен следовать этим правилам в процессе кодирования. Перечень правил не должен быть исчерпывающим или слишком объёмным. Задача состоит в том, чтобы сформулировать общие указания, благодаря которым код станет понятным для каждого из членов команды.

***Социальная защищённость программиста (Programmer welfare)***

— *40-часовая рабочая неделя* (Sustainable pace, Forty-hour week).