

## Вопросы к РК

### Методики разработки ПО

1. Назовите 10 навыков/умений/личностных качеств (в порядке убывания важности), которые необходимы разработчику ПО.

- *Усидчивость*
- *Умение долго находиться в глубокой концентрации*
- *Развитое логическое мышление*
- *Постоянное обучение*
- *Упорство*
- *Умение находить компромиссы*
- *Стрессоустойчивость*
- *Рассудительность*
- *Коммуникабельность*
- *Любопытство*

2. Какие из методик ХР вы считаете полезными, а какие - нет, и почему? В ответах укажите 10 методик. Комментарии для каждой методики обязательны.

- *Игра в планирование*

*Эффективная методика в случае нахождения общего языка с заказчиком. Разработчики должны адекватно иметь оценивать временную характеристику каждой микрозадачи.*

- *Тестирование до начала разработки*

*Ускоряет разработку, так как сразу четко определяет требования к программному продукту. При любом изменении требований заказчика необходимо дописать (или переписывать) тесты.*

- *Парное программирование*

*Основано на понятии Code review. Увеличивается качество кода, уменьшается количество ошибок. Без постоянной смены партнеров становится менее эффективным. Необходима хорошая командная работа.*

- *Рефакторинг*

*Методика улучшения кода. По моему мнению, должна входить во все остальные. Без тестов является разрушительной.*

- *Коллективное владение кодом*

*Частично входит в парное программирование. При большой текучке кадров, есть риск переписывания рабочего куска кода по несколько раз, так как «кому-то что-то показалась эффективнее и надо это проверить». При большом объеме продукта вырастает риск неучена некоторый зависимостей.*

- *Непрерывная интеграция*

*Уменьшает риск ошибок при итоговой интеграции. Увеличивается размер работающего кода на каждой итерации.*

- Присутствие заказчика

В теории эффективно. Однако возможен риск того, что заказчик «будет чересчур любопытным».

- 40-часовая рабочая неделя

Может стать неэффективной при жестком указании рамок рабочего дня. Попытки изменить биологические часы работников могут дорого обойтись компании (например в виде сна на рабочем месте). 40 часов — действительно оптимальный объем рабочих часов, однако эффективнее всего давать работником право распределять их по рабочей неделе (или даже по всей неделе, давая возможность работать удаленного в выходные дни).

- Стандарт кодирования

*Best Practice.* Уменьшает время интеграции новых умов в команду. Облегчает понимание кода.

- Смена позиций

Позволяет подвергнуть код всестороннему осмотру с разных точек зрения.

3. Какие роли в программных проектах вы знаете? Чем они занимаются и за что отвечают? Минимум 3 ответа.

- *Project Manager*

Как правило отвечают за связь команды разработчиков с заказчиками. Ведут общее управление проектом, следят за сроками и так далее.

- *Team Leader* + команда разработчиков

Являются основным средством написания программного продукта. *Team Leader* отвечает за команду и направляет ее на путь истинный. В отличие от *project Manager* непосредственно участвует в написании кода

- QA

Тестировщики, обеспечивают соответствие программного продукта необходимым требованиям, а также находят всевозможные ошибки.

- Дизайнеры

Отвечают за внешнюю составляющую продукта.

- Системные аналитики

Общаются с заказчиками. Уточняют их требования и дорабатывают до вида, понятного команде разработчиков.

- Системный архитектор

Проектирует ПО. Определяет основное устройство и архитектуру программного продукта. Принимает непосредственное участие в построении ПО. Принимает наиболее важные решения относительно продукта.

Качество кода

1. Назовите не менее 5 признаков, которыми должен обладать, на ваш взгляд, хороший код. Что даёт в итоге каждый такой признак?

- Следование стандарту оформления (Code Style) + адекватные названия переменных + отсутствие магических чисел

*Легкое сопровождение и читаемость.*

- Хорошее покрытие кода тестами

*Обеспечивает необходимое и определенное поведение приложение в любой ситуации. Обеспечивает точное соответствие требованиям.*

- Полное (не по объему) документирование кода

*Опять же отличное сопровождение.*

- Отсутствие повторяющихся кусков кода

*Сигнал о хорошей архитектуре ПО*

- Хороший код === безопасный код

*Ограждает от проблем в дальнейшем. Позволяет предотвратить большие плачевные инциденты.*

2. Иногда, когда говорят о программном коде, употребляют понятия «связность» и «связанность». Причём их часто путают. А как вы понимаете связность и связанность?

*И то, и другое является характеристикой ПО. Связанность модулей определяет степень зависимости модулей друг от друга. Связанность должна быть минимальной. Связность определяет, насколько модуль является простым, то есть насколько хорошо отвечает одной перед ним поставленной задачи и не имеет ничего лишнего.*

3. Что такое сопряжение? Каким оно бывает? Приведите 3 примера (с кодом) сопряжения, которые использовать не желательно.

*Сопряжение — мера, в которой сущности в коде связаны между собой.*

*Примеры: все атрибуты класса не private, один класс отвечает за несколько обязанностей, один класс завит от содержимого другого класса.*

Тестирование

1. В каких случаях тесты в проекте писать не нужно? Назовите минимум 3 случая.

*Когда проект в образовательных целях; на первых этапах разработки, когда пишется прототип;*

2. Приведите пример кода, для которого невозможно написать юнит-тесты. Пример должен содержать минимум 3 класса.

Паттерны проектирования

1. Назовите 10 паттернов, которые, на ваш взгляд, наиболее часто используются

*MVC*

*Singleton*

*Proxy*

Фабрика

Адаптер

Итератор

Хранитель

Шаблонный метод

Фасад

Декоратор

2. За счёт чего достигается экономия памяти при использовании паттерна «Приспособленец»?

*За счет выделения изменяемых и неизменяемых свойств. Эффективен там, где необходимо использовать большое количество объектов с минимальным различиями. И вместо одинаковых экземпляров использует только один.*