Оглавление

[Занятие 1 2](#_Toc411322606)

[Ответы на тест 2](#_Toc411322607)

[Первый вопрос 2](#_Toc411322608)

[Второй вопрос 2](#_Toc411322609)

[Третий вопрос 2](#_Toc411322610)

[Четвертый вопрос 3](#_Toc411322611)

[Практическое задание 3](#_Toc411322612)

[Тестирование чайника 3](#_Toc411322613)

[Занятие 2 6](#_Toc411322614)

[Ответы на тест 6](#_Toc411322615)

[Первый вопрос 6](#_Toc411322616)

[Третий вопрос 6](#_Toc411322617)

[Четвертый вопрос 6](#_Toc411322618)

[Практическое задание 7](#_Toc411322619)

[Занятие 8 8](#_Toc411322620)

[Ответы на тест 8](#_Toc411322621)

[Первый вопрос 8](#_Toc411322622)

[Второй вопрос 8](#_Toc411322623)

[Третий вопрос 8](#_Toc411322624)

[Четвертый вопрос 9](#_Toc411322625)

[Пятый вопрос 9](#_Toc411322626)

Занятие 1

## Ответы на тест

Первый вопрос

i. To find defects.- да, это дна их основных целей тестирования

ii. To gain confidence in the level of quality. - повысить уверенность в уровне качества - да, тоже подходит

iii. To identify the cause of defects. - определить причину дефектов - нет. Причина дефектов это ошибка программиста, либо проблемы аппаратного свойства, либо какие-то внешние причины (типа магнитных бурь). Со всем этим лучше, чтобы разбирался программист. Конечно же, тестировщик может догадываться о причинах дефектов, но это не его цель. Его цель находить сами дефекты, а не их причину.

iv. To prevent defects. - да. Тестировщик, как человек, влияющий на весь процесс разработки и вовлеченный в него, может вносить свои предложения как предотвратить дефекты, изменив немного сам процесс. Например, тестировщик может предложить использовать code style-ы (это повысит сопровождаемость)...

То есть правильный ответ на первый вопрос С.

Второй вопрос

Which is the MOST important advantage of independence in testing?

A An independent tester may find defects more quickly than the person who wrote the  
software. – нет, это зависит от опыта человека, а не от того тестер он или нет  
B An independent tester may be more focused on showing how the software works than  
the person who wrote the software. – нет, как раз наоборот. Зачастую именно разработчики фокусируются на том, чтобы показать, КАК работает то, что они создали  
C An independent tester may be more effective and efficient because they are less  
familiar with the software than the person who wrote it. – нет. Чаще (особенно это касается больших систем) именно тестеры знают функциональность ВСЕЙ системы. Разработчик в основном знает только свою ЧАСТЬ  
D An independent tester may be more effective at finding defects missed by the person  
who wrote the software. – да. Разработчик написал код, отладил (то есть выловил часть ошибок и исправил). Тестер ищет те ошибки, которые еще остались после отладки

Правильный ответ D.

Третий вопрос

testable - это поддающийся проверке.

Есть характеристики ПО, про которые мы не можем точно сказать, удовлетворяет ли им ПО. Например, удобство использования, сопровождаемость. Мы можем как-то выразить это количественно, но сказать 100% удобно или 100% сопровождаемо мы не можем.

Поэтому:

A The system shall be user friendly - не подходит

B The safety-critical parts of the system shall contain 0 faults - это в принципе невозможно (см. Принцип1)

C The response time shall be less than one second for the specified design load - да, это мы может точно определить

D The system shall be built to be portable - не можем точно это протестировать и дать ответ Да или Нет

Правильный ответ С.

Четвертый вопрос

A To find faults in the software - да, может быть целью тестирования

B To access whether the software if ready for release - да, тоже подходит

C To demonstrate that the software doesn’t work - да, вполне возможно

D To prove that the software is correct - нет. Принцип1 - мы не может доказать, что дефектов нет. Принцип2 - невозможно протестировать все. Принцип7 - даже если не найдены дефекты, это не доказывает правильность ПО

Правильный ответ D.

## Практическое задание

Тестирование чайника

Хочу вас немного поправить.

Вы пишите: "В наші обов'язки не входить тестувати (перевіряти) дизайн (форму, тип) приладу. Ми перевіряємо те, що можна описати цифрами. Не забираємо роботу в інших (дизайнерів, проектантів)"

Это не совсем правда. Тестеры проверяют дизайн, удобство (это так называемое usability тестирование). Просто мы не можем точно сказать Да, чайник удобный или Да, чайник производительный.

Это такие характеристики, которые трудно поддаются тестированию, НО мы можем дать количественную оценку удобства или производительности.

Например:

99 человек из 100 смогли включить чайник без инструкции

30 человек из 100 без инструкции разобрались, как очистить чайник

во время очистки чайника 28 человек из 30 сделали несколько ошибок (список ошибок прилагается)

чайник кипятит максимальное количество воды за 20 секунд

чайник с накипью кипятит максимальное количество за 30 секунд

и т.д.

Просто по тому, что я написала выше, мы не можем сказать, что чайник удобен или производителен.

Но это может помочь разработчикам (программистам, дизайнерам, аналитикам) разобраться, КАК улучшить чайник.

Вот примерный список того, что мы не можем протестировать в работе чайника:

- его работу в космосе

- без дополнительного оборудования мы не можем проверить его устойчивость к перепадам напряжения

- как он нагревает НЕ воду, а другие жидкости

Все из этого списка относится к рискам. Мы это не проверяем, но потребитель может захотеть все это проверить \ узнать. Потребитель может столкнуться с проблемами, так как мы не все проверяли.

Мультиварку вы протестировали хорошо :)

А вот список того, что не может быть протестировано неполный :)

- работу в космосе

- без дополнительного оборудования мы не можем проверить устойчивость к перепадам напряжения

- приготовление блюд, не описанных в инструкции

Ваш список

1) Wash the dishes;

2) Play like radio;

3) Go shopping, etc.

я бы заменила одним пунктом - "Использование не по назначению".

Занятие 2

## Ответы на тест

Первый вопрос

A Source-code inspections are often used in component testing – нет. Инспекция – это вид статического тестирования. Он обычно применяется ко всему коду (ну или к каким-либо документам), основывается на формальной процедуре. Во время инспекции находятся ошибки типа: нарушение стандартов разработки и несоответствие документации более высокого уровня.  
B Component testing searches for defects in programs that are separately testable – да. Компонентное тестирование применяется к каждому модулю отдельно. Поэтому и найденные дефекты могут тестироваться по-отдельности (не имеет значения, работает ли вся система в целом или нет).  
C Component testing is an important part of user acceptance testing – нет. Приемочное и компонентное тестирование относятся к разным уровням тестирования.  
D Component testing aims to expose problems in the interactions between software and  
hardware components – нет. Компонентное тестирование проверяет работу каждого из компонентов отдельно.

Правильный ответ В.

Третий вопрос

A All document reviews involve the development team – не обязательно. В оригинале говорилось про тестеров.  
B Some, but not all, development activities have corresponding test activities – нет. Всем и каждой активности девелоперов должна быть соответствующая тестовая активность.   
C Each test level has test objectives specific to that level – да. Это даже описано в «характеристиках качественного тестирования в любом ЖЦ разработки ПО».  
D Analysis and design of tests begins as soon as development is complete – нет. Это относиться в водопадной модели разработки ПО. Мы с вами говорили, что это и есть ее минут, что тестировать начинают уже после того, как весь код написан.

Правильный ответ С.

Четвертый вопрос

Чтобы протестировать функцию, программист должен написать \_\_\_ , который вызывает эту функцию и передает ей тестовые данные.

A Stub – нет. Заглушка ничего не вызывает, она имитирует еще не реализованный компонент.  
B Driver – да. Драйвер – это и есть средство тестирования, которое заменяет компонент, обеспечивающий вызов другого компонента. Подходит по определению.  
C Proxy – нет. Прокси - объект, который контролирует доступ к другому объекту через перехват всех вызовов. Не подходит оп определению.  
D None of the above – нет.

Правильный ответ В.

## Практическое задание

Предположим, ручка состоит из стержня, колпачка и корпуса.

Компонентное тестирование: тестируем стержень, колпачок и корпус отдельно.

1) пишет ли стержень на обычной бумаге, которая лежит на столе горизонтально

2) пишет ли стержень на вертикальной поверхности, и насколько долго

3) вытекает ли паста из стержня, если его перевернуть пером вверх, как быстро

4) есть ли внутренняя резьба на колпачке

5) если ли внешняя резьба на корпусе

6) все материалы соответствуют стандартам

и т.д.

Интеграционное тестирование:

1) Собираем ручку: стержень входит в корпус

2) Собираем ручку: колпачок закручивается на корпусе, его невозможно просто стащить, не раскручивая

и т.д.

Системное тестирование:

1) Ручка в собранном виде пишет на гладкой поверхности

2) Ручка в собранном виде пишет на негладкой поверхности

3) Ручка пишет на горизонтальной поверхности

4) Ручка пишет на вертикальной поверхности

5) Нажим, необходимый для использования ручки, не превышает общепринятых норм (чтобы рука не быстро уставала от письма)

и т.д.

Приемочное тестирование (проводит либо покупатель - Бета, либо человек, отвечающий за закупки - контрактное, либо человек на предприятии, где ручки производят - Альфа):

1) Ручка пишет

2) Колпачок не спадает

и т.д.

Занятие 8

## Ответы на тест

Первый вопрос

Test planning has which of the following major tasks?

1. Determining the scope and risks, and identifying the objectives of testing - да
2. Determining the test approach (techniques, test items, coverage, identifying and interfacing the teams involved in testing, test ware) - да
3. Reviewing the Test Basis (such as requirements, architecture, design, interface) – нет. Эта информация используется, когда планируются тесты (это Test analysis and design направление в процессе тестирования)
4. Determining the exit criteria - да

Правильный ответ А

Второй вопрос

Deciding how much testing is enough should take into account:

1. Level of Risk including Technical and Business product and project risk – да (это risk-based approach)
2. Project constraints such as time and budget – да (если есть ограничения по времени и бюджету, то для тестирования выбираются более приоритетные области ПО)
3. Size of Testing Team – нет. Размер команды тестировщиков не подскажет, что и сколько нужно тестировать , и на что надо обратить особое внимание
4. Size of the Development Team – нет. Размер команды разработчиков вообще не поможет определить объем тестирования. Для определения объема тестирования может помочь проектная документация, анализ рисков, определение более приоритетных областей ПО

Правильный ответ С

Третий вопрос

The structure of the results for each level of the designated testing activities is covered in the Standard for Software Test Documentation IEEE 829 and is called as:

A Anomaly Report – нет. Этот документ для описания неожиданной проблемы, с который столкнулся тестировщик. Он не описывает результаты тестирования для каждого из уровней тестовых активностей.

B Defect Report – нет. Это отчет об ошибке, а не отчет о результатах тестирования.

C Test Defect Report – нет. Такого вида документов не бывает.

D Master test report – да. Используется, чтобы подвести итоги результатов тестирования на каждом из уровней.

Правильный ответ D.

Четвертый вопрос

IEEE 829 test plan documentation standard contains all of the following except:

A test items – это есть (см. формат Level Test Plan)

B test deliverables – есть. Тут описываются отчеты, которые будут предоставлены как результат тестиварония

C test tasks - есть

D test environment - есть

E test specification – нет (это часть Level Test Design)

Правильный ответ Е.

Пятый вопрос

Testing should be stopped when:

A all the planned tests have been run – нет. Я бы ответила Да, если бы не было пункта Е. Ведь действительно мы не всегда может выполнить все запланированные тесты. Количество тестов может зависеть от рисков (ну и от ограничения времени и бюджета)

B time has run out – нет. Если вышло время, то мы просто задерживаемся с тестированием, и это не значит, что мы должны все остановить. Хотя иногда действительно принимают решение, что время – самый важный фактор. Но это не обычное решение, и оно должно быть оговорено заранее.

C all faults have been fixed correctly – нет. Во-первых, странное слово all. Во-вторых, релиз в основном делают не только с целью пофиксить баги, еще чтобы реализовать какую-то новую функциональность, например.

D both a) and c) – нет.

E it depends on the risks for the system being tested – да (risk-based approach).

Правильный ответ Е.

## Практическое задание

Тестирование web- и desktop-приложений немного отличается. Для web- более тщательно проверяют безопасность, тестирование на разных конфигурациях и производительность.