01. What is Integration Testing?

Testing of combined parts of an application to determine if they function together correctly. Usually performed after unit and functional testing. This type of testing is especially relevant to client/server and distributed systems.

Integration Testing - мы имеем несколько систем. Идея в том что мы входим в одну систему(Система это сумма двух или трех модулей которая имеет определенную функциональность) и конкретный модуль и получаем output потом дальше. Функциональность одной системы запускает функциональность другой системы и приводит к какому то конечному результату. Пример RTCCA- real time credit card authorization( проверка кредитки при покупке)

Вы покупаете что то в онлайн магазине . Shopping bag - check out - RTCCA

RTCCA проверяют

1 credit card number ( exist or not)

2 authentication (разрешение на ввод) имеет 2 уровня защиты:

key itself and address (пример ключ от дома, если вор нашел ключ , ему так же еще надо найти дверь)

Credit Card Authentication:

- CC Number(visible)

- CC Name( ваше full name)(visible)- Expiration Date(visible)- CVC core(visible)

- ZIP Code (invisible)

- Unusual activities(invisible)когда уезжаешь в другую страну, могут карту заблокировать.Или покупаешь что то дорогое(Rolex), а обычно дешевое(кофе)

02. What is Build Acceptance Testing?

Testing whether the new build is ready for comprehensive testing.

Build Acceptance Test (BAT) - или Basic Acceptance Test

Результат этого тестирования - готов ли этот build к всеобъемлющему(полному) тестированию.

Is the current build is ready for comprehensive testing.

Когда выходит новый билд. Менеджер раздал задание кто что делает. И выясняем что не работет. Выясняем работают ли основные функциональности. Show stoppers(Это основные функциональности, если они не работают нет смысла дальше тестировать) = Facebook - Like doesn't work or eCommerce - Checkout не работает. Если Like не работает нет смысла начинать полное тестирование.(Ночью происходит тестирование на основные функциональности и утром все получают имейл - build is ready for comprehensive testing. Но если что то не работает надо быстренько это поправить)

03. What is Localization Testing?

Is the process of customizing a software application that was originally designed for a domestic market so that it can be released in foreign markets.

Localization testing стр 28 - насколько ваш аппликейшн работает на других языках. Языки другие знать не надо чтоб тестировать.

Язык интерфейса - Log in=Вход= entrada.

Язык ввода информации - Field Вход можно писать на любом языке который поддерживает app.Tools for testing переводчик, beyond compare

Функциональность - в чем оплачиваете в рублях или долларах(IP address выясняет где вы и предлагает оплату в зависимости от страны. Через user agent выясняется ваш IP address) меры длины, курс валют currency, температура, формат date англ MM/DD/YYYY, европ DD/MM/YYYY, японская YYYY/MM/DD, Latvian jan.DD/YYY. Эта функциональность переключается либо автоматически либо самому надо поменять (зависит от дизайна программы)

20 основных я зыков интерфейса. Делятся на две категоии single bite - 8 bits. Азтиатские языки Double byte - 16 bits.

Продакт менеджер пишет все на англ отправляет фрилансерам нужного языка и они переводят текст. Потом делают документацию.

04. What is Compatibility Testing?

Testing whether software is compatible with other elements of a system with which it should operate. To check the compatibility of the system across different operating systems, browsers, add-ons and screen resolutions.

Compatibility testing - тестируется в 4 направлениях: Тестируем апп на разных OS(Win 7 &Wind 8, Mac 9 & Mac 10 как мин на 2), browser(Firefox, Safary, Chrome, Explorer), ads on(Java, Flash), screen resolution(разное разрешение экранов самые популярные )

05. What is Performance Testing?

Is the process of determining the speed or effectiveness of a software. This process is measuring the response time at which a system functions.

It is performed to determine how fast some aspect of a system performs under a particular workload. It can demonstrate that the system meets performance criteria. It can compare two systems to find which performs better.

05. What is Performance Testing?

Is the process of determining the speed or effectiveness of a software. This process is measuring the response time at which a system functions.

It is performed to determine how fast some aspect of a system performs under a particular workload. It can demonstrate that the system meets performance criteria. It can compare two systems to find which performs better.

Тестирование производительности является процесс определения скорости или эффективности компьютерных, сетевых, программного обеспечения или устройства. Этот процесс может включать в себя количественные анализы, в лаборатории, такие как измерение времени отклика или количество MIPS (миллионов инструкций в секунду), при котором система функций. Качественные атрибуты, такие как надежность, масштабируемость и совместимость также могут быть оценены. Тестирование производительности часто делается в сочетании с стресс-тестирования.

Тестирование производительности можно убедиться, что система удовлетворяет требованиям заявленных заводом-изготовителем или продавцом.Процесс может сравнить два или несколько устройств или программ с точки зрения таких параметров, как скорость, скорость передачи данных, пропускной способности, пропускная способность, эффективность и надежность.

Тестирование производительности также может быть использован в качестве диагностического средства в поиске связи узкие места. Часто система будет работать гораздо лучше, если проблема решена в одной точке или в одном компоненте. Например, даже самый быстрый компьютер будет функционировать плохо на современном Web, если соединение происходит при только 40 до 50 Кбит (килобит в секунду).

Performance testing is the process of determining the speed or effectiveness of a COMPUTER, network, software program or device. This process can involve quantitative tests done in a lab, such as measuring the response time or the number of MIPS (millions of instructions per second) at which a system functions. Qualitative attributes such as reliability, scalability and interoperability may also be evaluated. Performance testing is often done in conjunction with stress testing.

Performance testing can verify that a system meets the specifications claimed by its manufacturer or vendor. The process can compare two or more devices or programs in terms of parameters such as speed, data transfer rate, bandwidth, throughput, efficiency or reliability.

Performance testing can also be used as a diagnostic aid in locating communications bottlenecks. Often a system will work much better if a problem is resolved at a single point or in a single component. For example, even the fastest computer will function poorly on today's Web if the connection occurs at only 40 to 50 Kbps (kilobits per second).

Slow data transfer rate may be inherent in hardware but can also result from software-related problems, such as:

Too many applications running at the same time

A corrupted file in a Web browser

A security exploit

Heavy-handed antivirus software

The existence of active malware on the hard disk.

Effective performance testing can quickly identify the nature or location of a software-related performance problem.

06. What is Documentation Testing?

Documentation is “Any written or pictorial information describing, defining, specifying, reporting, or certifying activities, requirements, procedures, or results”. Documentation is the product itself. If the documentation is poor, non-existent, or wrong, it reflects on the quality of the product and the vendor.

Тестирование документации гарантирует, что письменные разъяснения о том, как использовать систему практически соответствует тому, что система делает. Это значительная часть, чтобы проверить, так как горящие исправления, как правило, изменить поведение системы в виде, что сворачивает в соответствии с документацией.Хороший тестировщик должен знать, что все виды документации пользователя, должны быть проверены: это включает в себя операционные руководства, а также обычный документации для конечного пользователя в приложение. Документация может быть проверена, задавая пользователи пытаются использовать новую систему, используя документацию. После этого вы должны выбрать тех пользователей, которые знакомы с их собственное доменное (работа, приложения, администрирование и т.д.), но которые не знакомы с новой системой.

Documentation testing is a non-functional type of software testing.

It is a type of non-functional testing.

Any written or pictorial information describing, defining, specifying, reporting, or certifying activities, requirements, procedures, or results’. Documentation is as important to a product’s success as the product itself. If the documentation is poor, non-existent, or wrong, it reflects on the quality of the product and the vendor.

As per the IEEE Documentation describing plans for, or results of, the testing of a system or component, Types include test case specification, test incident report, test log, test plan, test procedure, test report. Hence the testing of all the above mentioned documents is known as documentation testing.

This is one of the most cost effective approaches to testing. If the documentation is not right: there will be major and costly problems. The documentation can be tested in a number of different ways to many different degrees of complexity. These range from running the documents through a spelling and grammar checking device, to MANUALLY reviewing the documentation to remove any ambiguity or inconsistency.

Documentation testing can start at the very beginning of the software process and hence save large amounts of money, since the earlier a defect is found the less it will cost to be fixed.

Documentation testing assures that the written explanations of how to use the system practically match what the system does. This is significant part to test since last minute fixes usually change the behavior of the system in a way, which swerves from documentation. A good software tester should know that all kinds of user documentation must be tested: this comprises operational guides as well as the usual application end user documentation. Documentation can be tested by asking users try to use the new system using the documentation provided. After this you should choose those users who are conversant with their own domain (operation, application, administration and so on) but who are not conversant with the new system.

07. What is Portability Testing (Cross-Platform)?

Is the testing of a software application can be moved from one platform to another (compatibility of your application across different platforms).

Когда разработчики работают на продуктах, которые являются портативными или взаимодействовать на разных платформах, он является общим для использования одной из этих платформ в качестве предпочтительного платформы для развития. (Заметим, что "платформа" не относятся не только к OperatingSystems, а также может относиться к различным серверными СУБД, CORBA ORB-от разных производителей, и так далее.) Платформа разработки может быть тот, который наиболее часто используется клиентами, или может быть одним с лучших инструментов разработки и отладки.

В такой ситуации, может быть силы, которые вызывают UnitTests будет работать на других платформах лишь эпизодически или не работать вообще. Это может привести к неприятным сюрпризам. При разработке кросс-платформенных продуктов, запуск тестов постоянно на всех поддерживаемых платформах необходимой частью TestDrivenDevelopment. В противном случае, вы действительно только работает определенное количество тестов. Вы не можете AlwaysBeReadyToShip если вы не постоянно работает полный набор тестов для всех платформ.

When developers work on products that are portable or interoperable across platforms, it is common to use one of those platforms as the preferred platform for development. (Note that "platform" does not only refer to OperatingSystems; it can also refer to different database backends, CORBA ORBs from different vendors, and so on.) The development platform may be the one that is most commonly used by customers, or may be the one with the best development and debugging tools.

In such a situation, there may be forces that cause UnitTests to be run on other platforms only sporadically, or not run at all. This can lead to nasty surprises. When developing cross-platform products, running the tests continuously on all supported platforms is a necessary part of TestDrivenDevelopment. Otherwise, you are really only running a subset of the tests. You can not AlwaysBeReadyToShip if you aren't continuously running the full set of tests for all platforms.

Some tips for cross-platform testing:

When developing a product that has to work across platforms, set up the cross-platform tests immediately--don't put it off until later.

Set up an automated script that will run the tests on all platforms at least once per day (perhaps after each DailyBuild).

Use a cross-platform scripting language to drive automated test scripts, and use the exact same script for all platforms. If that is not possible, find some other way to guarantee that the same tests are being run everywhere.

Whenever tests fail unexpectedly, compare the results on different platforms. If the tests succeed on some platforms but fail on others, it's often a sign that the code is not as portable as is thought.

Whenever a new test is added, the developer who writes the test should be encouraged to immediately run it on all platforms (but this may not be feasible in all environments).

Don't use MockObjects in place of the cross-platform elements. You need to run tests against the real stuff to ensure portability and interoperability. Also beware of using platform emulators for your tests.

08. What is Usability Testing?

Testing for 'user-friendliness'. Usability testing focuses on measuring a human-made product's capacity to meet its intended purpose.

Юзабилити-тестирование относится к оценке продукт или услугу, проверив ее с представительными пользователей. Как правило, во время теста, участники будут пытаться завершить типовых задач, а наблюдатели смотреть, слушать и принимает.Цель состоит в том, чтобы выявить любые проблемы с юзабилити, собирать качественные и количественные данные и определить удовлетворение Участника с продуктом.

Usability testing refers to evaluating a product or service by testing it with representative users. Typically, during a test, PARTICIPANTS will try to complete typical tasks while observers watch, listen and takes notes. The goal is to identify any usability problems, collect qualitative and quantitative data and determine the PARTICIPANT'S satisfaction with the product.

To run an effective usability test, you need to develop a solid test plan, recruit participants , and then analyze and report your findings.

Benefits of Usability Testing

Usability testing lets the design and development teams identify problems before they are coded. The earlier issues are identified and fixed, the less expensive the fixes will be in terms of both staff time and possible impact to the schedule. During a usability test, you will:

Learn if participants are able to complete specified tasks successfully and

Identify how long it takes to complete specified tasks

Find out how satisfied participants are with your Web site or other product

Identify changes required to improve user performance and satisfaction

And analyze the performance to see if it meets your usability objectives

You Do Not Need a Formal Lab

Effective Usability Testing does not require a formal usability lab for testing. You can do effective usability testing in any of these settings:

Fixed laboratory having two or three connected rooms outfitted with audio-visual equipment

Room with portable recording equipment

Room with no recording equipment, as long as someone is observing the user and taking notes

Remotely, with the user in a different location (either moderated or unmoderated)

Factors Affecting Cost

Your testing costs depend on

Type of testing performed

Size of the team assembled for testing

Number of PARTICIPANTS for testing

Number of days you will be testing

Remember to budget for more than one usability test. Building usability into a Web site (or any product) is an iterative process. Consider these elements when budgeting for usability testing:

Time: You will need time to plan the usability test. It will take the usability specialist and the team time to become familiar with the site and pilot test the test scenarios. Be sure to budget in time for this test prep as well as running tests, analyzing the data, writing the report, and presenting the findings.

Recruiting Costs: Consider how or where you will recruit your PARTICIPANTS. You will either need to allow for staff time to recruit or engage a recruiting firm to schedule participants for you based on the requirements.

Participant Compensation: If you will be compensating participants for their time or travel, factor that into your testing budget.

Rental Costs: If you do not have monitoring or recording equipment, you will need to budget for rental costs for the lab or other equipment. You may also need to secure a location for testing, a conference room for example, so consider this as well.

It’s important to keep in mind that usability testing is not just a milestone to be checked off on the project schedule. The team should have a goal for why they are testing and then implement the results.

09. What is Ad Hoc Testing?

A testing phase where the tester tries to 'break' the system by randomly trying the system's functionality.  Testers have significant understanding of the software before testing it.

Специальная тестирование неформальной и импровизационный подход к оценке эффективности продукта.Специальная тест обычно проводится только один раз, если дефект не найден.

Обычно используются в разработке программного обеспечения, специальных тестирование проводится без плана действий и любые действия, предпринятые как правило, не документированы. Тестеры могут не иметь детальное знание требований к продукции. Специальная проверка также называют выборочной проверки и обезьяны тестирования.

Потому что подход не является методическая, специальная тестирование может пропустить недостатки, которые будут найдены в более структурированной системы тестирования. Тем не менее, отсутствие формальных требований также означает, что очевидные недостатки могут присутствовать быстрее, чем если бы они были подходить в более систематической основе.

Исследовательское тестирование, в котором продукт разработан, изучаются и проверяются одновременно, иногда предлагается как уточнение этой специальной модели.

Ad hoc testing is an informal and improvisational approach to assessing the viability of a product. An ad hoc test is usually only conducted once unless a defect is found.

Commonly used in software development, ad hoc testing is performed without a plan of action and any actions taken are not typically documented. Testers may not have detailed knowledge of product requirements. Ad hoc testing is also referred to as random testing and monkey testing.

Because the approach is non-methodical, ad hoc testing can miss flaws that would be found in a more structured testing system. However, the lack of formal requirements also means that obvious flaws can be attended to more quickly than if they had to be approached in a more systematic fashion.

Exploratory testing, in which a product is developed, studied and tested simultaneously, is sometimes proposed as a REFINEMENT of the ad hoc model.

10. What is Acceptance Testing (UAT)?

Final testing based on specifications of the end-user. Conducted to determine whether or not a system meets a set of agreed acceptance criteria.

Формальное тестирование по отношению к потребностям пользователей, требований и бизнес-процессов, проводимых для определения, удовлетворяет ли система критериям приемки и предоставить пользователю, клиентов или других уполномоченных лица, чтобы определить, является ли или не принять систему. Итоговое тестирование на основе спецификаций конечного пользователя. Проводятся для определения, соответствует ли система набора согласованных критериев приемки. Принятие Тестирование уровень тестирования программного обеспечения, где система тестируется на приемлемость.

Цель этого теста заключается в оценке соответствия системы с бизнес-требованиями и оценить, является ли приемлемым для поставки.

Final testing based on specifications of the end-user. Conducted to determine whether or not a system meets a set of agreed acceptance criteria. Acceptance Testing is a level of the software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system’s compliance with the business requirements and assess whether it is acceptable for delivery.

Formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system.

ANALOGY

During the process of manufacturing a ballpoint pen, the cap, the body, the tail and clip, the INK CARTRIDGE and the ballpoint are produced separately and unit tested separately. When two or more units are ready, they are assembled and Integration Testing is performed. When the complete pen is integrated, System Testing is performed. Once System Testing is complete, Acceptance Testing is performed so as to confirm that the ballpoint pen is ready to be made available to the end-users.

METHOD

Usually, Black Box Testing method is used in Acceptance Testing. Testing does not normally follow a strict procedure and is not scripted but is rather ad-hoc.

When is it performed?

Acceptance Testing is performed after System Testing and before making the system available for actual use.

Who performs it?

Internal Acceptance Testing (Also known as Alpha Testing) is performed by members of the organization that developed the software but who are not directly involved in the project (Development or Testing). Usually, it is the members of Product Management, Sales and/or Customer Support.

External Acceptance Testing is performed by people who are not employees of the organization that developed the software.

Customer Acceptance Testing is performed by the customers of the organization that developed the software. They are the ones who asked the organization to develop the software. [This is in the case of the software not being owned by the organization that developed it.]

User Acceptance Testing (Also known as Beta Testing) is performed by the end users of the software. They can be the customers themselves or the customers’ customers.

11. What is Exploratory Testing?

Informal software test that is not based on formal test plans or test cases, testers may be learning the software as they test it.

* As its name implies, exploratory testing is about exploring, finding out about the software, what it does, what it doesn’t do, what works and what doesn’t work. The tester is constantly making decisions about what to test next and where to spend the (limited) time. This is an approach that is most useful when there are no or poor specifications and when time is severely limited.
* Exploratory testingis a hands-on approach in which testers are involved in minimum planning and maximum test execution.
* The planning involves the creation of a test charter, a short declaration of the scope of a short (1 to 2 hour) time-boxed test effort, the objectives and possible approaches to be used.
* The test design and test execution activities are performed in parallel typically without formally documenting the test conditions, test cases or test scripts. This does not mean that other, more formal testing techniques will not be used. For example, the tester may decide to us boundary value analysis but will think through and test the most important boundary values without necessarily writing them down. Some notes will be written during the exploratory-testing session, so that a report can be produced afterwards.

Как следует из названия, Exploratory testing является исследованием, выяснением о программном обеспечении, что он делает, что он не делает, что работает, а что не работает. Тестер во время тестирования постоянно принимает решение что делать дальше, и на изучение чего потратить (ограниченное) время. Это подход, который является самым полезным, когда есть слабая (или ее нет вообще) спецификация и когда время сильно ограничено.

Exploratory testing это практический подход, при котором тестеры занимаются в минимальной степени планированием и максимально - выполнением теста.

Планирование предполагает создание тестового Устава, краткую декларацию о масштабах кратковременных тестов (от 1 до 2 час), целях и возможных подходах, которые будут использоваться.

Тест-дизайн и исполнение тестов выполняются параллельно, как правило, без формального документирования условия испытаний, тестов и сценариев тестирования. Это не означает, что другие, более формальные методы тестирования не будут использоваться. Например, тестер может принять решение об объемах граничного тестирования, проверить самые важные граничные значения и не обязательно записывать их. Некоторые замечания будут записаны во время поисково-сессии тестирования, так что отчет может быть произведено после этого.

12. What is Sanity Testing?

Brief test of major functional elements of a piece of software to determine if it’s basically operational.

After receiving a Software build with the minor issues fixes in code or functionality, Sanity testing is carry out to check whether the bugs reported in previous build are fixed & there is regression introduced due to these fixes i.e. not breaking any previously working functionality. The main aim of Sanity testing to check the planned functionality is working as expected. Instead of doing whole regression testing the Sanity testing is perform.

Sanity tests helps to avoid wasting time and cost involved in testing if the build is failed. Tester should reject the build upon build failure.

After completion of regression testing the Sanity testing is started to check the defect fixes & changes done in the software application is not breaking the core functionality of the software. Typically this is done nearing end of SDLC i.e. while releasing the software. You can say that sanity testing is a subset of acceptance testing. We can also say Tester Acceptance Testing for Sanity testing. Sanity testing is narrow & deep approach of testing, it needs to concentrate limited & main features of testing in detailed.

Here are the few consolidated points of Sanity testing:

-Sanity testing follows narrow and deep approach with detailed testing of some limited features.

-Sanity testing is typically non-scripted.

-Sanity testing is a sub-set of regression testing.

-Sanity testing is cursory testing to prove software application is working as mention in the specification documents & meets the user needs.

-Sanity testing is used to verify the requirements of end users are meeting or not.

-Sanity testing to check the after minor fixes the small section of code or functionality is working as expected & not breaking related functionality.

Узконаправленное тестирование достаточное для доказательства того, что конкретная функция работает согласно заявленным в спецификации требованиям. Является подмножеством регрессионного тестирования. Используется для определения работоспособности определенной части приложения после изменений произведенных в ней или окружающей среде. Обычно выполняется вручную. Отличие санитарного тестирования от дымового (Sanity vs Smoke testing) - эти виды тестирования имеют "вектора движения", направленнные в разные стороны. В отличии от дымового (Smoke testing), санитарное тестирование (Sanity testing) направлено вглубь проверяемой функции, в то время как дымовое направлено вширь, для покрытия тестами как можно большего функционала в кратчайшие сроки.

13. What is Smoke Testing?

Typically an initial testing effort to determine if a new software version is performing well enough to accept it for a major testing effort.  A quick-and-dirty test of the major functions.

Smoke testing, in the context of software development, is a series of test cases that are run before the commencement of more rigorous tests. The goal of smoke testing is to verify that an application's main features work properly. A smoke test suite can be automated or a combination of manual and automated testing.

Smoke testing is a preliminary test which is used to catch the high-level functional errors of an application. If the tests fail, then further testing of the application stops, and the build is refused for additional testing until the build passes the smoke test.

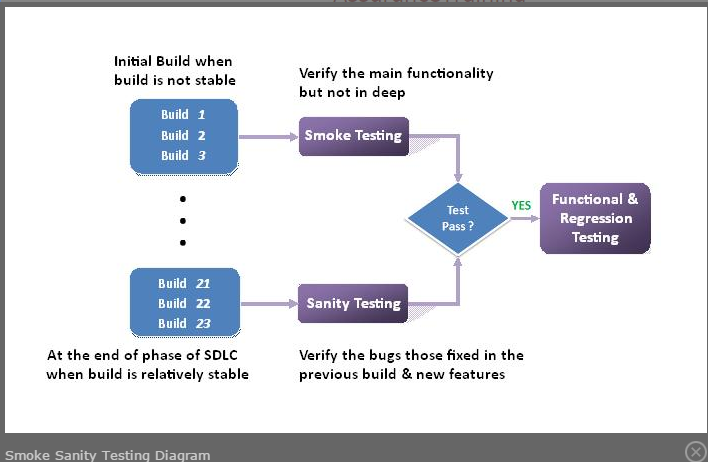
The term itself is derived from the electronic industry. While repairing a hardware component, if the equipment works with no smoke, the component is said to have passed the test.

Smoke testing is the initial testing process exercised to check whether the software under test is ready/stable for further testing.

The term ‘**Smoke Testing**’ is came from the hardware testing, in the hardware testing initial pass is done to check if it did not catch the fire or smoked in the initial switch on.

Prior to start *Smoke testing* few test cases need to created once to use for smoke testing. These test cases are executed prior to start actual testing to check **critical functionalities of the program is working fine**. This set of test cases written such a way that all functionality is verified but not in deep. The objective is not to perform exhaustive testing, the tester need check the navigation’s & adding simple things, tester needs to ask simple questions “Can tester able to access software application?”, “Does user navigates from one window to other?”, “Check that the GUI is responsive” etc.

Here are graphical representation of Smoke testing & Sanity testing in software testing



The test cases can be executed manually or automated; this depends upon the project requirements. In this types of testing mainly focus on the important functionality of application, tester do not care about detailed testing of each software component, this can be cover in the further testing of application.

The Smoke testing is typically executed by testers after every build is received for checking the build is in testable condition. This type of testing is applicable in the Integration Testing, System Testing and Acceptance Testing levels.

Smoke testing -это серия тестов, которые выполняются до начала более строгих тестов. Целью Smoke testing является проверка, что основные функциональности работают должным образом. Smoke testing может быть автоматизированым или сочетанием ручного и автоматизированного тестирования.

Smoke testing- предварительный тест, который используется, чтобы поймать на высоком уровне функциональные ошибки приложения. Если испытания окажутся неудачными, дальнейшее тестирование приложения останавливается и build не направляется для дополнительной проверки, пока он не проходит Smoke testing.

14. What is Alpha Testing?

Testing of an application when development is nearing completion (at the Alpha Stage of the development), minor design changes may still be made as a result of such testing.

Alpha testing is one of the most common software testing strategy used in software development. Its specially used by product development organizations.

This test takes place at the developer’s site. Developers observe the users and note problems.

Alpha testing is testing of an application when development is about to complete. Minor design changes can still be made as a result of alpha testing.

Alpha testing is typically performed by a group that is independent of the design team, but still within the company, e.g. in-house software test engineers, or software QA engineers.

Alpha testing is final testing before the software is released to the general public. It has two phases:

In the first phase of alpha testing, the software is tested by in-house developers. They use either debugger software, or hardware-assisted debuggers. The goal is to catch bugs quickly.

In the second phase of alpha testing, the software is handed over to the software QA staff, for additional testing in an environment that is similar to the intended use.

Alpha testing is simulated or actual operational testing by potential users/customers or an independent test team at the developers’ site. Alpha testing is often employed for off-the-shelf software as a form of internal acceptance testing, before the software goes to beta testing.

Альфа-тестирование является одним из наиболее распространенных стратегий тестирования программного обеспечения, используемого в разработке программного обеспечения. Его специально используется организациями по развитию продукта.

Этот тест проводится в поле зрения разработчика. Разработчики наблюдают за пользователями и контролируют/исправляют проблемы.

Альфа-тестирование это тестирование приложения, когда подходит время релиза. Незначительные изменения в конструкции еще могут быть сделаны в результате альфа-тестирования.

Альфа-тестирование обычно выполняется группой, которая не зависит от проектной группы, но все еще в пределах компании, например, в отделе тестирования программного обеспечения тестерами или software QA engineers.

Альфа-тестирование является окончательным тестированием перед выпуском для широкой публики. Она состоит из двух этапов:

На первом этапе альфа-тестирования, программное обеспечение проходят проверку в доме разработчиков. Они используют либо отладчика программного обеспечения или аппаратных помощь отладчики. Цель состоит в том, чтобы поймать ошибки быстро.

На втором этапе альфа-тестирования, программное обеспечение переданы сотрудникам QA, для дополнительной проверки в среде, которая похожа на среду предполагаемого использования.

Альфа-тестирование моделируется или фактических тестируется потенциальными пользователями/клиентами или независимой группой испытателей на месте (в присутствии) разработчиков. Альфа-тестирование часто используется как форма внутренних приемо-сдаточных испытаний, прежде чем программное обеспечение идет в стадии бета-тестирования.

15. What is Beta Testing?

Testing when development and testing are essentially completed and final bugs and problems need to be found before final release.

Beta testing comes after alpha testing and can be considered a form of external [user acceptance testing](http://en.wikipedia.org/wiki/User_acceptance_testing). Versions of the software, known as [beta versions](http://en.wikipedia.org/wiki/Beta_version), are released to a limited audience outside of the programming team. The software is released to groups of people so that further testing can ensure the product has few faults or [bugs](http://en.wikipedia.org/wiki/Computer_bug). Sometimes, beta versions are made available to the open public to increase the [feedback](http://en.wikipedia.org/wiki/Feedback#In_organizations) field to a maximal number of future users.

Бе́та-тести́рование ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) beta testing) — интенсивное использование почти готовой версии продукта (как правило, [программного](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) или [аппаратного обеспечения](https://ru.wikipedia.org/wiki/%D0%90%D0%BF%D0%BF%D0%B0%D1%80%D0%B0%D1%82%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5)) с целью выявления максимального числа [ошибок](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B3) в его работе для их последующего устранения перед окончательным выходом ([Релизом](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D0%B8%D0%B7_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5))) продукта на рынок, к массовому потребителю.

В отличиелагает привлечение добровольцев из числа обычных будущих пользователей продукта, которым доступна упомянутая предварительная версия продукта (так называемая бета-версия).

Такими добровольцами (их называют бета-тестерами) часто движет любопытство к новому продукту — любопытство, ради удовлетворения которого они вполне согласны мириться с возможностью испытать последствия ещё не найденных (а потому и не исправленных) ошибок. Кроме любопытства, мотивация может быть обусловлена желанием повлиять на процесс разработки и в итоге получать более удовлетворяющий их нужды продукт и многим другим.

Кроме того, открытие бета-тестирования может использоваться как часть стратегии продвижения продукта на рынок (например, бесплатная раздача бета-версий позволяет привлечь широкое внимание потребителей к окончательной дорогостоящей версии продукта), а также для получения предварительных отзывов о нём от широкого круга будущих пользователей.

Бета-версия не является финальной версией продукта, поэтому разработчик не гарантирует полного отсутствия ошибок, которые могут нарушить работу компьютера и/или привести к потере данных. Хотя и в финальных версиях таких гарантий разработчики, как правило, не дают.

16. What is Security Testing?

Security testing (penetration testing) includes a review of the security configuration of each component of the system and a review of known vulnerabilities within routers, firewalls, operating systems and other technology deployed including web, mail and database servers. Application security testing is therefore performed that includes an analysis of web application code, to locate weaknesses that could be exploited at an HTTP/URL level.

Цель тестирования безопасности является выявление угроз в системе и измерить его потенциальных уязвимостей. Это также помогает в обнаружении всех возможных рисков безопасности в системе и помогают разработчикам в фиксации этих проблем с помощью кодирования.

Типы тестирования безопасности:

Есть семь основных типов тестирования безопасности как в Open Source тестирования безопасности методологии руководства. Они объясняются следующим образом:

Vulnerability Scanning: Это делается через автоматизированного программного обеспечения для сканирования системы от известных сигнатур уязвимостей.

Security Scanning: Это включает в себя определение сетевых и системных недостатков, а позже предоставляет решения для снижения этих рисков. Такое сканирование может выполняться как для ручного и автоматического сканирования.

Penetration testing: Этот вид тестирования имитирует нападение с хакера. Это тестирование включает в себя анализ конкретной системы для проверки потенциальных уязвимостей к внешним попытки взлома.

Risk Assessment: Это тестирование включает в себя анализ рисков безопасности, наблюдаемых в организации. Риски классифицируются как низко-, средне- и Хай. Это тестирование рекомендует контроля и мер по снижению риска.

Security Auditing: Это внутренняя проверка приложений и операционных систем для уязвимостей. Аудит также может быть сделано с помощью строка за строкой проверки кода

Ethical hacking: Это взлома системы программного обеспечения организации. В отличие от хакеров, которые крадут для собственных прибылей, намерение состоит в том, чтобы разоблачить недостатки безопасности в системе.

Posture Assessment: Это объединяет сканирование безопасности, - взлома и оценок риска, чтобы показать общий уровень безопасности организации.

Vulnerability Scanning: This is done through automated software to scan a system against known vulnerability signatures.

Security Scanning: It involves identifying network and system weaknesses, and later provides solutions for reducing these risks. This scanning can be performed for both Manual and Automated scanning.

Penetration testing: This kind of testing simulates an attack from malicious hacker. This testing involves analysis of a particular system to check for potential vulnerabilities to an external hacking attempt.

Risk Assessment:This testing involves analysis of security risks observed in the organization. Risks are classified as Low, Medium and High. This testing recommends controls and measures to reduce the risk.

Security Auditing: This is internal inspection of Applications and Operating systems for security flaws. Audit can also be done via line by line inspection of code

Ethical hacking: It's hacking an Organization Software systems. Unlike malicious hackers ,who steal for their own gains , the intent is to expose security flaws in the system.

Posture Assessment: This combines Security scanning, Ethical Hacking and Risk Assessments to show an overall security posture of an organization.

17. What is Install/Uninstall Testing?

Testing of full, partial, or upgrade install/uninstall processes.

Web system often require both client-side and server-side ddinstalls. Testing of the installer checks that installed features function properly--including icons, support documentation , the README file, and registry keys. The test verifies that the correct directories are created and that the correct system files are copied to the appropriate directories. The test also confirms that various error conditions are detected and handled gracefully.

Testing of the uninstaller checks that the installed directories and files are appropriately removed, that configuration and system-related filea are also appropriately removed or modified, and that the operating environment is recovered in its original state.

Тестирование установки направленно на проверку успешной инсталляции и настройки, а также обновления или удаления программного обеспечения.

В настоящий момент наиболее распространена установка ПО при помощи инсталляторов(специальных программ, которые сами по себе так же требуют надлежащего тестирования, описание которого рассмотрено в разделе "[Особенности тестирования инсталляторов.](http://www.protesting.ru/testing/types/installationfeatures.html)").

В реальных условиях инсталляторов может не быть. В этом случае придется самостоятельно выполнять установку программного обеспечения, используя документацию в виде инструкций или readme файлов, шаг за шагом описывающих все необходимые действия и проверки.

В распределенных системах, где приложение разворачивается на уже работающем окружении, простого набора инструкций может быть мало. Для этого, зачастую, пишется план установки (Deployment Plan), включающий не только шаги по инсталляции приложения, но и шаги отката (roll-back) к предыдущей версии, в случае неудачи. Сам по себе план установки также должен пройти процедуру тестирования для избежания проблем при выдаче в реальную эксплуатацию. Особенно это актуально, если установка выполняется на системы, где каждая минута простоя - это потеря репутации и большого количества средств, например: банки, финансовые компании или даже баннерные сети. Поэтому тестирование установки можно назвать одной из важнейших задач по обеспечению качества программного обеспечения.

Именно такой комплексный подход с написанием планов, пошаговой проверкой установки и отката инсталляции, полноправно можно назвать тестированием установки или Installation Testing.

18. What is Comparison Testing?

Comparing software weaknesses and strengths to competing products.

Сравнение программного обеспечения с другими продуктами

Говоря простыми словами, тестирование сравнение тип тестирования, где тестеры сравнить сильные программный продукт и слабых с другим программным обеспечением, которые в настоящее время доступны на рынке. Тест Сравнение является очень хорошим показателем того, насколько конкурентоспособными и полезным программный продукт будет для конечных пользователей вскоре после его коммерческого выпуска.

Тестирование программного обеспечения выполняется, сравнивая параллельно несколько элементов. В каждом из сравниваемых компонентов необходимо выделить такую информацию, как рабочая характеристика и параметры, чтобы потом их сравнить.

Сравнительное тестирование программного обеспечения выполняют либо как исследовательское тестирование, либо как более серьезный эксперимент с контрольной и экспериментальной группами, в зависимости от того, какой результат Вы хотите получить.

Любопытно, что лучшие результаты тестирования будут в том случае, если сравниваемые элементы абсолютно отличаются, а не когда у них есть нечто общее. Причиной тому может быть:

Чтобы сравнить кардинально разные элементы, тестировщик затрачивает больше усилий, ведь он должен отойти от привычной схемы действий.

Чтобы сравнить кардинально разные элементы, нужен более глубокий анализ, поскольку в то время, как у схожих элементов подобная структура, у противоположных – абсолютно иная.

19. What is Automated Testing?

Automated Test (tests without manual intervention) is ability to playback predefined actions, compare the results to the expected behavior and report the success or failure. It’s the best way to improve the accuracy, increase the efficiency and test coverage of your software testing, and supports unattended test runs.

Автоматизированное тестирование программного обеспечения (Software Automation Testing) - это процесс верификации программного обеспечения, при котором основные функции и шаги теста, такие как запуск, инициализация, выполнение, анализ и выдача результата, выполняются автоматически при помощи инструментов для автоматизированного тестирования.

Преимущества автоматизации тестирования:

Повторяемость – все написанные тесты всегда будут выполняться однообразно, то есть исключен «человеческий фактор». Тестировщик не пропустит тест по неосторожности и ничего не напутает в результатах.

Быстрое выполнение – автоматизированному скрипту не нужно сверяться с инструкциями и документациями, это сильно экономит время выполнения.

Меньшие затраты на поддержку – когда автоматические скрипты уже написаны, на их поддержку и анализ результатов требуется, как правило, меньшее время чем на проведение того же объема тестирования вручную.

Отчеты – автоматически рассылаемые и сохраняемые отчеты о результатах тестирования.

Выполнение без вмешательства – во время выполнения тестов инженер-тестировщик может заниматься другими полезными делами, или тесты могут выполняться в нерабочее время (этот метод предпочтительнее, так как нагрузка на локальные сети ночью снижена).

Automated testing is as simple as removing the "human factor" and letting the computer do the thinking. This can be done with integrated debug tests, to much more intricate processes. The idea of the these tests is to find bugs that are often very challenging or time intensive for human testers to find. This sort of testing can save many man hours and can be more "efficient" in some cases. But it will cost more to ask a developer to write more lines of code into the game (or an external tool) then it does to pay a tester and there is always the chance there is a bug in the bug testing program. Reusability is another problem; you may not be able to transfer a testing program from one title (or platform) to another. And of course, there is always the "human factor" of testing that can never truly be replaced.

Other successful alternatives or variation: Nothing is infallible. Realistically, a moderate split of human and automated testing can rule out a wider range of possible bugs, rather than relying solely on one or the other. Giving the testere limited access to any automated tools can often help speed up the test cycle.

20. What is Agile Testing?

Agile [means moving quickly] Testing is a software testing practice that follows the principles of the agile manifesto:

       •   Individuals and interactions over processes and tools

       •   Working software over comprehensive documentation

       •   Customer collaboration over contract negotiation

       •   Responding to change over following a plan.

Agile testing involves testing from the customer perspective as early as possible, testing early and often as code becomes available and stable enough from the unit level testing.

Agile testing is a [software testing](http://en.wikipedia.org/wiki/Software_testing) practice that follows the principles of [agile software development](http://en.wikipedia.org/wiki/Agile_software_development). Agile testing involves all members of a cross-functional agile team, with special expertise contributed by testers, to ensure delivering the business value desired by the customer at frequent intervals, working at a sustainable pace. [Specification by example](http://en.wikipedia.org/wiki/Specification_by_example) is used to capture examples of desired and undesired behavior and guide coding.

Agile development recognizes that testing is not a separate phase, but an integral part of software development, along with coding. Agile teams use a "whole-team" approach to "baking quality in" to the software product. Testers on agile teams lend their expertise in eliciting examples of desired behavior from customers, collaborating with the development team to turn those into executable specifications that guide coding. Testing and coding are done incrementally and iteratively, building up each feature until it provides enough value to release to production. Agile testing covers all types of testing. The Agile Testing Quadrants provide a helpful taxonomy to help teams identify and plan the testing needed.