Whether it is small calculator software with only the basic arithmeticoperations, or an online enterprise solution; there are two categories of applications.  
**a.**[**Desktop**](http://www.softwaretestinghelp.com/difference-between-desktop-client-server-and-web-testing/)  
**b.**[**Web**](http://www.softwaretestinghelp.com/web-application-testing/)

Не смотря на то, является ли приложение маленьким калькулятором, которое считает примитивные арифметические операции или онлайн солюшеном для крупного предприятия, выделяют 2 категории приложений:

**a.**[**Desktop**](http://www.softwaretestinghelp.com/difference-between-desktop-client-server-and-web-testing/)  
**b.**[**Web**](http://www.softwaretestinghelp.com/web-application-testing/)

**Difference between Desktop, Client server and Web testing**

Можно выделить 3 видатестирования:

* **Desktop application testing,**
* **Client server application testing and**
* **Web application testing.**

Все они отличаются средой – энварментом, в котором они тестируются.

**Десктопные приложения** запускаются на персональных компьютерах и ворк станциях, таким образом, когда вы тестируете десктоп приложение, вы фокусируетесь на оперделенном энварменте. Вы сможете протестировать приложение в таких категориях как GUI, functionality, Loadи например бэкэнт – т. е. БД.

В **клиент-серверных приложениях** есть 2 стороны для тесnирования: приложение, которое работает на сервере, а также ехе-приложение на каждой клиентской машине. Здесь можно тетсирвоать приложение относительно таких аспектов: GUI на обоих сторонах, functionality, load, client-server interaction, backend. При этом вы должны учитывать количество клиентов и серверов в тестовых сценариях.

Здесь имеется **front-end и backend.** Например, приложения, разработанные на C++, Java, VB, C#. Бэкэндом для таких приложений может быть MSAccess, SQLServer, Oracle, Sybase, Mysql, Quadbase.

Тестирование **Веб-приложений** немного отличается, в частности тем, что тестер не может настолько контролировать и управлять приложением, как в предыдущих случаях. Приложение может быть загруженным на сервере, расположение которого тестер может ии не может знать. Ехе-приложение не устанавливается на сервере. Необходимо тестировать веб-приложение на разных браузерах и ОС. Поэтому в основном тестируется браузерная совместимость и совместимость ОС, отслеживание ошибок, статические страницы, бэкэнд тестинг и нагрузочное тестирование.

**Web Server**

* Компьютер, который подсоединет к Интернет. Он владеет Веб-контентом и конфигурирован для того, чтобы делиться этим контентом.
* Веб-сервер отвечает за принятие http запросов от клиентов, которые являются Веб-браузерами, и обслуживаемые или Веб-страницы, которые обычно являются HTML документами и слинкованными объектами (изображениями и т.д.)

Например: {…}

- A computer that is connected to the Internet. Hosts Web contentand is configured to share that content.

- Webserver is responsible for accepting HTTP requests from clients,which are known as Web browsers, and serving them Web pages,which are usually HTML documents andlinked objects (images,etc.).

• Examples:

oApache HTTP Server from the Apache Software

Foundation.

oInternet Information Services (IIS) from Microsoft.

oSun Java System Web Server from Sun Microsystems,formerly Sun ONE Web Server, Planet Web Server,and Netscape Enterprise Server.

oZeus Web Server from Zeus Technology

**Web Client**

* Наиболее часто в форме Веб-браузера, напр. IE, Netscape.
* Используется для навигации Веб-контента Веб-серверов для просмотра.

**ApplicationServer**

– Anapplicationserver– это компьютер в сети, предназначенный для запуска нескольких software applications. Этоттерминтакжеотноситсякsoftware, котороебылопроинсталлированонатакомкомпьютередля того, чтобы обслуживать другие приложения. Applicationserverпредоставляеттакженаборmiddlewareдлятого, чтобыapplicationsмогливзаимодействоватьв соответствии с различными аспектами качества:reliability, security, nonrepudiation, andsoon.

ВзаимодействияосуществляютсячерезWebвформеHTMLandXML,в качестве связующего звена для различных баз данных, и в качестве ссылок между системами и устройствах, начиная от масшатабных приложений до небольших информационных устройст

– An application server располагает business logic для client applications черезразличныеprotocols, возможновключая HTTP. Болеетого the application server управляетвсемисвоимиресурсами

– Ex: JBoss (Red Hat), WebSphere (IBM), Oracle Application Server 10g (Oracle Corporation) and WebLogic (BEA)

-An application server is a server computer in a computer network dedicated to running certain software applications. The term also refers to the software installed on such a computer to facilitate the serving of other applications. Application server products typically bundle middleware to enable applications to intercommunicate with various qualities of service — reliability, security, nonrepudiation, and so on. Application servers also provide an API to programmers, so that they don't have to be concerned with the

operating system or the huge array of interfaces required of a modern web-based application. Communication occurs through the web in the form of HTML and XML, as a link to various databases, and, quite often, as a link to systems and devices ranging from huge legacy applications to small information devices, such as an atomic clock or a home appliance.

– An application server exposes business logic to client applications through various protocols, possibly including HTTP. the server exposes this business logic through a component API, such as the EJB (Enterprise JavaBean) component model found on J2EE (Java 2 Platform, Enterprise Edition) application servers. Moreover, the application server manages its own resources. Such gate-keeping duties include security, transaction processing, resource pooling, and messaging

– Ex: JBoss (Red Hat), WebSphere (IBM), Oracle Application Server 10g (Oracle Corporation) and WebLogic (BEA)

**ThinClient**

Это компьютер в клиент-серверной архитектуре, который имеет мало или вообще не имеет логики, так что он зависит от центального сервера для выполнения каких-либо действий. Архитектура устроена таким образом, что обработка данных в большей степени осуществляется на сервере.

– A thin client is a computer (client) in client-server architecturenetworks which has little or no application logic, so it has to dependprimarily on the central server for processing activities. It isdesigned to be especially small so that the bulk of the dataprocessing occurs on the server.

**Thickclient**

Это клиент, который производит обработку данных самостоятельно, и связан с сервером исключительно для обращения к базе данных.

– It is a client that performs the bulk of any data processingoperations itself, and relies on the server it is associated withprimarily for data storage.

**World Wide Web (the Web)**

- Модель обмена информацией, которая построена на базе Internet, использует HTTP протокол и браузеры (например, Internet Explorer) для доступа к веб-страницам, отформатированным в HTML, которые связаны с помощью гиперссылок. Web это только часть Интернета (из других целейИнтернет можно выделить электронную почту (через SMTP), Usenet, обмен мгновенными сообщениями и передачи файлов (через FTP).

**–** An information sharing model that is built on top of the Internet,utilizes HTTP protocol and browsers (such as Internet Explorer) toaccess Web pages formatted in HTML that are linked via hyperlinksand the Web is only a subset of the Internet (other uses of the

Internet include email (via SMTP), Usenet, instant messaging andfile transfer (via FTP)

**HTTP (Hypertext Transfer Protocol)**

БазовыйпротоколWorldWideWeb.Определяет, как сообщения форматируется и передается по протоколу TCP / IP сети для веб-сайтов.Определяет, какие действия веб-серверов и веб-браузеров предпринять в ответ на различные команды.

**–** Underlying protocol of the World Wide Web. Defines how messagesare formatted and transmitted over a TCP/IP network for Websites. Defines what actions Web servers and Web browsers take inresponse to various commands.

Stateless: В программировании, без протокола протокол связи, который рассматривает каждый запрос какнезависимую транзакцию, которая не имеет отношения к какой-либо предыдущий запрос так, чтобыобщение состоит из независимых пар запросов и ответов.Без гражданства, протокол не требует серверасохранить информационную сессию или статуса друг о партнерских связей на протяжении нескольких запросов.

- HTTP является stateless. Преимущество данного протокола это то что он не хранит информацию о пользователях между запросами, но это заставляет использовать альтернативные методы хранения пользовательскиой информации, корорый ранее пользовался какими-либо ресурсами.Общий метод решения этой проблемы связано с использованием отправки и запросов cookies.

**–** HTTP is stateless. The advantage of a stateless protocol is that hostsdon't need to retain information about users between requests, butthis forces the use of alternative methods for maintaining users'state, for example, when a host would like to customize content for

a user who has visited before. The common method for solving thisproblem involves the use of sending and requesting cookies. Othermethods are session control, hidden variables, etc

* Пример: когда вы вводите URLв браузер, HTTPкоманда посылает Webсерверу запрос на загрузку и передачу запрашиваемой веб-старницы.

**–** example: when you enter a URL in your browser, an HTTPcommand is sent to the Web server telling to fetch and transmit therequested Web page