* **Imp: link** <http://www.guru99.com/software-testing.html>
* <http://www.etestinghub.com/what_is_softwaretesting.php>

**New DTV features**

Broadcasting has always been a one way proposition: what you see is what you get. Watching TV has, until now, been a passive activity. In the beginning, just seeing the moving images was spellbinding. As technology moved forward, color, then sophisticated graphics and live remote broadcasts were added. Today, viewers can even vote live, albeit via their telephones, on some programs. The quest has been to provide the viewer with as much information as possible while expanding their involvement with the programs.

Today some will say that this visual information deluge is beyond usable to the point of distracting. (Read ["Lower the visual volume"](http://broadcastengineering.com/mag/broadcasting_lower_visual_volume/) from the October 2005 Editorial in *Broadcast Engineering*.) Have you ever been watching a news program, then catch something on the crawl, where your attention shifts to the elimination of the original program? As you wait for the crawl to return the information you’re looking for, you're miss the entire newscast.

Television viewing might be more enjoyable if the viewer could control the appearance of extra info, graphics and alerts. In other words, allowing the audience to customize their viewing experience could benefit both viewer and broadcaster.

The advent of DTV brings us to the verge of a personalized, converged, more Internet-like television viewing experience. That is, if the proper standards are implemented.

**DTV standards**

In 1995, when the ATTC completed tests of the Grand Alliance prototype, there was little precedent for two-way communication between viewer and content provider. The Internet had yet to take off as a consumer item. Still, the standards committee well understood the need to provide extra capability in the MPEG transport stream for datacasting and other enhanced TV features.

Emerging standards have sought global interoperability. The DVB Multimedia Home Protocol (MHP) is the DVB specification for the technical implementation of interactive TV applications. The DTV Application Software Environment (DASE) was the initial ATSC interactive enabling standard. CableLabs has their version, the Open Cable Application Protocol or (OCAP) for cable platforms. But, that leaves us, again, with more than one platform to support.

Fortunately, the ATSC A/101, Advanced Common Application Platform (ACAP) standard harmonizes the ATSC DASE and Cable Labs OCAP standards while also providing functionality for Globally Executable MHP (GEM), which apply to DVB applications. The intent is to offer an interoperable standard so that receiver manufacturers can develop interactive-capable consumer devices.

**Infrastructure technology implementation**

There are three types of features that DTV standards enable: enhanced features, interactive features and networked or Internet applications.

Enhanced (eTV) features are those applications that can be implemented without a back channel. Interactive (iTV) features require a back channel. Network or Internet features expand the TV viewing experience beyond the living room or immediate environment.

Enhanced features include local scenarios. Applications are downloaded to the receiver and all events are triggered by user input. The applications reside in the receiver. An EPG is an example of local interactivity.

When a back channel is available, access to remote information can be requested interactively by the user and then downloaded to the DTV receiver. An example would be a VOD movie.

Networking features and connections to the Internet are still in development. This will lead to personal home digital networks. But interoperability, security and rights management are issues that must be resolved before such applications can become a reality.

**New advertising opportunities**

The ability to deliver customized and targeted advertising is something all broadcasters and content providers need to think about. This is considered the "Holy Grail" of Madison Avenue. Let's look at one targeted advertising workflow example.

As with any complex project, careful planning must be the starting point, specifically emphasizing the intended end result. In this case, the goal is to promote and get the viewer to purchase a vacation package.

The client has decided to target three groups: retirees, families and couples. To sell to these three groups, an infomercial will be produced. The infomercial will be made available as VOD. Other features will include the ability for the viewer to email a link for this commercial to a friend or access the advertiser's website from the TV set. An option to locate a near-by local travel agent will be an on-screen option. Finally, the program will allow the viewer to schedule and pay for a trip — all through the TV.

The usual ad production process is followed, the ad is storyboarded and content must be produced. While some content is universal to all three audiences, there are some scenes that are unique to each audience. For the families, the video/audio may highlight group activities. For the singles, romantic locals may be key. For the retirees, perhaps the content emphasizes more relaxed and restful locations. Appropriate video is captured and returned to the post room for composition.

Now, it’s time to assemble the interactive application. First, there needs to specific icons developed, which indicate the two-way nature of the video. These must be linked to specific triggers that will launch behind-the-scenes tasks. Appropriate security must also be enabled.

To implement these interactive features, new authoring tools will be needed. These are available from several vendors, but be sure that the production tools will properly embed the needed codes and data to properly control the down-line functions that must be completed.

The "data" is fed into the MPEG transport stream multiplexer. It must be compliant with appropriate specifications such as ATSC A/90. In any case, it cannot exceed the 19.39Mb/s maximum data rate.

The video is transmitted and then received by the consumer. The television or STB now decodes the audio, video and data. Data is then placed on the DSM-CC data carousel and appropriate icons are displayed.

Inside the receiver or STB, the viewer has stored key personal information. These demographics may include information that the viewer is married and has no children. This means that when the VOD is displayed, instead of a scene of noisy children playing, a couple is shown at some romantic get away. The ad concludes with a viewer-activated lower-third graphic with contact information for the nearest local travel agent based on the viewer's zip code.

Let's say the viewer is interested in receiving more information. She clicks on the information icon. In an OTA system, the Directed Channel Change (DCC) PSIP feature is activated. Based on the viewer's profile, the DCC jumps to a relevant looping infomercial on an SD multicast subchannel. Or, with a cable STB, a specific VOD could be launched.

Finally, the trip is booked and the Television-commerce (T-commerce) transaction is securely completed. A link to this ad is forwarded via email to a friend whose 10th wedding anniversary is coming.

Sound far fetched? It's not. All that's needed is to connect the various pieces of the data technology puzzle and broadcasters will have developed an entire new class of profit-making services.

**Making contact**

Unfortunately, until ACAP/OCAP/MHP-capable receivers are available, interactivity, targeted advertising and T-commerce will be limited to STB implementations.

Samsung demonstrated the first ACAP-ready HDTV, in conjunction with Aircode, a Korean company specializing in DTV-related iTV and commerce applications, in the DTV Drafthouse at NAB2005. Set-top boxes from vendors like Motorola and Scientific-Atlanta are already OCAP capable.

Traditional broadcasters must adapt. This is the survival of the fittest, where the consumer could vote you out of business with their dollars. Broadcasters must realize that DTV is not just about audio, video and graphics. It is about creating an interactive experience.

We'll look at some new examples of DTV interactivity in our next newsletter. If you have specific questions or comments, send them to [editor@primediabusiness.com](mailto:editor@primediabusiness.com).

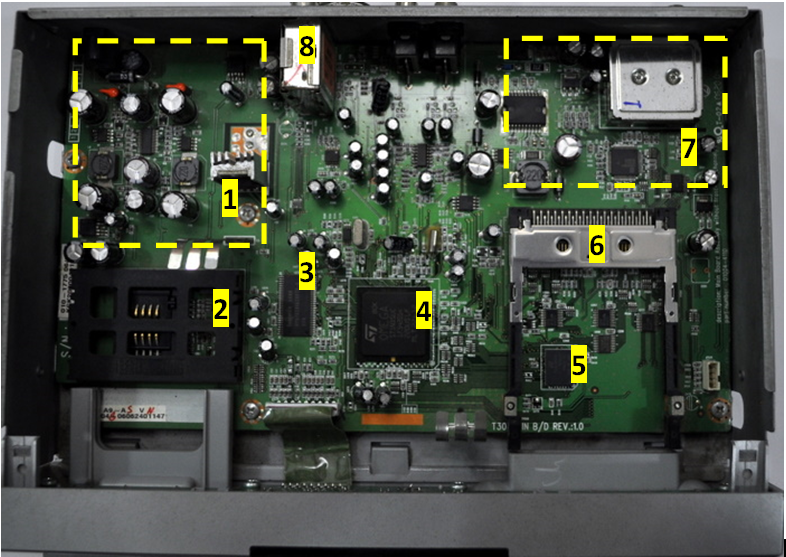
Read more: <http://broadcastengineering.com/infrastructure/Dtv-features-new-20051123/#ixzz20TvOS0cR>

## Inside Set Top Box (STB)

Set Top Box or STB has become an integral part of TV viewing in many parts of the world. We commonly see this sleek looking device sitting on side of TVs. Though this device looks slim and simple but it is one of the most complex embedded systems today. STBs are increasing their feature set day by day. Few of the common features in current generation STBs are time shift mode viewing, recording, Internet based viewing, video on demand, Full High definition video output etc.  
STB is very complex embedded system; it consists of 30+ hardware blocks and similar number of software drivers. STB has lot of computing power distributed across main processor and various co-processors. In few of top end STBs if we add operating frequencies of all co-processors then it would be in range of 3-4 GHz.

**Hardware**

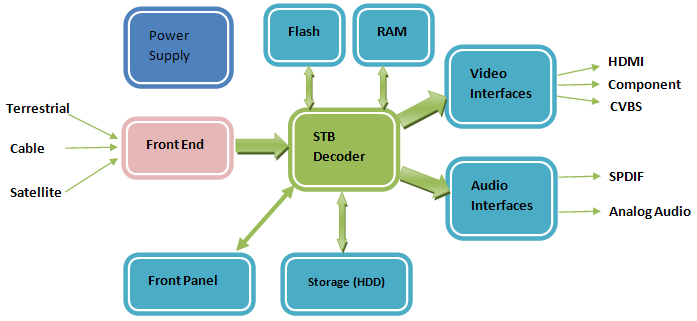
A typical STB would look similar to one shown in following image. This is picture of standard definition (SD) satellite based STB being used at my home. Number of components used is fairly less compared to complexity of this system. This credit goes to the main STB decoder chip which integrates a lot of hardware components required into a single chip.

[](http://embeddedtutorial.com/wp-content/uploads/2010/09/STB_PCB.PNG)

Inside view of Satellite Set Top Box

**STB blocks**1. Power Supply  
2. Smart Card Slot  
3. RAM  
4. STB Decoder  
5. Flash  
6. DVB-CI Slot  
7. Satellite Front End  
8. RF Modulator

The above picture shows STB circuit board and major components on the board. A more logical relationship between various components is shown in following block diagram.

[](http://embeddedtutorial.com/wp-content/uploads/2010/09/SetTopBoxHardware.PNG)

Set Top Box Hardware Blocks

**STB Decoder:** This is heart of the whole system. In current generation STBs most of the features required by STB system are integrated in STB decoder chips. This level of integration is called System On Chip (SoC). STB SoCs contain a large number of blocks ranging from de-multiplexer to decoders and peripherals like USB, SATA etc. We will discuss STB SoC in more details in next post [Inside Set Top Box Part 2](http://embeddedtutorial.com/2010/09/inside-set-top-box-stb-part-2-decoder-soc-and-software/).

**Front End:** Front End part of STB is responsible for receiving the broadcasted signal, demodulating the signal and outputting digital data output for STB decoder chip. Depending upon broadcasting environment terrestrial or satellite or cable front end will be used. Front end unit consist of 3 main blocks tuner to tune correct frequency, demodulator to demodulate as per standard and forward error correction (FEC) unit for data recovery.

**Power Supply:**This is the main power source for board. This unit generates different voltage required by various components on board. Input to this unit can be main line AC (220/110) or DC 12V via standalone power adapters.

**Flash:** This is used to store boot loader, main application and other user specific non volatile data. Different STBs uses different sizes of flash ranging from 8MB to 64MB.

**RAM:** RAM is used to store all intermediate data (such as decoded video/audio buffers) and application variables. In many cases main application is also copied to RAM and is executed from RAM to speed up the operation (as RAM is faster compared to Flash). RAM size ranges from 32MB in standard definition STBs to 256MB in some top end Full HD STBs.

**Video Interfaces:** STB decoder chip outputs video data in analog or digital format. To make these signals compatible with external devices, special circuitry like filter and physical connector are required. Current generation STBs provide many video output formats such as CVBS, S Video, Component video and HDMI.  
Audio Interfaces: STB decoder chip outputs audio data in analog as well as digital format. In some cases high quality DACs are used to convert digital data into analog format. Digital data is also transmitted in digital format using SPDIF standard.

**Storage:** Few STBs also work as digital video recorders. To aid storage of programs some storage device (HDD) is added via any of the interfaces (SATA, eSATA, ATAPI or USB) provided by decoder chip.

**Front Panel:** This is STBs interface to external world. Front panels are different for different boxes. But most of them provide IR input/output, Status LEDs, 7 segments or LCD and few switches to configure set top box. These features are controlled by parallel IOs of main decoder chip. In some cases a dedicated microcontroller is added to front end to reduce processing load for main chip and also to reduce the number of wires going from front panel to main PCB.

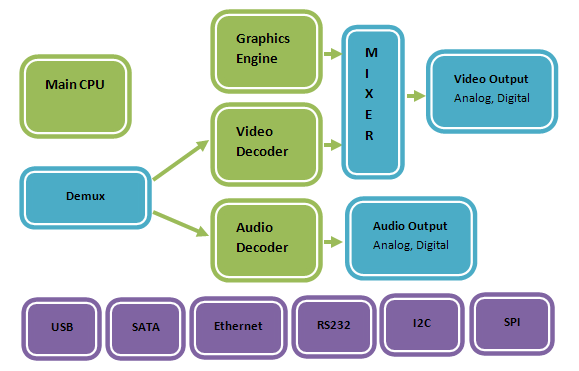
**DVB-CI Slot:** This slot is provided to support various conditional access schemes. Conditional Access providers provide compatible DVB-CI cards to be used with STB. The DVB-CI card decrypts the channels encrypted by Conditional Access provider as per user’s subscription policies.

**Smart Card Slot:** This slot is provided to use smart card for Conditional Access implementation. Unique subscriber ID is stored on each card. Smart card is also used in decrypting the channels.

**RF Modulator:** This is used to modulate Audio and video into RF signal. This is mainly to be used with older TVs which have only RF input and no composite (CVBS) input.

This is second part of main article [Inside Set Top Box](http://embeddedtutorial.com/2010/09/inside-set-top-box-stb/), Board hardware and basic blocks are discussed in [previous article](http://embeddedtutorial.com/2010/09/inside-set-top-box-stb/).

**STB Decoder SoC:**  
STB Decoder is one of the most complex system on chip (SoC). There is generally one main processor and lot of co-processor doing dedicated processing.  
A typical decoder SoC will have following blocks.

[](http://embeddedtutorial.com/wp-content/uploads/2010/09/SetTopBoxDecoder.PNG)

Set Top Box Decoder blocks

**Main CPU:** This is the main CPU executing the STB application. It is a general purpose CPU with lot of development tools available. Its speed range from 200MHz on standard definition devices to near 1GHz on High Definition devices. The main CPU is generally based on industry standard core such Super H or ARM to improve ease of tools and reusable stack availability.

**Demux:** Demux is a dedicated co-processor to de-multiplex the digital transport stream into audio, video and other data. Demux checks the input stream for errors and protocol compliance and filters the required data into desired buffers (Audio, video)

**Video Decoder:** This co-processor is responsible for converting compressed video (MPEG) data into basic video format. Current generation decoders have programmable video decoders, so video decoders can support a variety of formats such as MPEG2, H264, VC1 etc.

**Graphics Engine:**This co-processor is dedicated to graphics acceleration. Its main task is to draw pictures and menus for user interface (UI). This unit is becoming more powerful these days with introduction of 3D menus.

[](http://embeddedtutorial.com/wp-content/uploads/2010/09/STB_MIxer.PNG)

Set Top Box Mixer operation

**Mixer:** This block is responsible for mixing the video output and graphics output and producing a final single image. This is also responsible for ordering of video and graphics plane and transparency settings. As shown is image there are two planes one video in background on video plane and other is rectangular window showing program information on graphics plane. These two planes are mixed by mixer to generate a single image.

**Video Output:** Final result after mixing video decoder and graphics accelerator outputs is provided to video output block. This block is responsible for outputting data as per required standards (PAL, NTSC, SECAM, HDMI). This block generates output in analog format using DACs. This block also generates output in digital format using HDMI convertors.

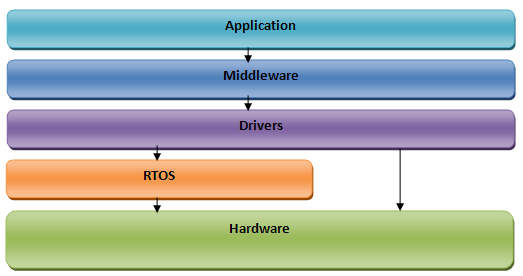
**Audio Decoder:** Audio decoder converts the compressed audio data into basic audio data. Audio decoders are also programmable these days. They can be programmed to support any audio standard. Audio decoder generally supports MPEG, AAC, Dolby formats.

**Audio Output:** Audio output is fed to audio output block. This block provides audio output in analog format using internal DACs and in digital format using SPDIF convertor.

**Peripherals:** Other than main decoders and CPU a lot of peripheral devices are supported by STB SoCs for providing various features.  
**USB:** For record/playback on external storage  
**SATA:** used to connect HDD for providing digital video recording facility.  
**Ethernet:** Input source for IP based STBs  
**UART:**Debug port. Sometimes also used for software upgrade in field  
**I2C:** Used by main STB SoC to communicate with external peripheral devices such as front end, SCART controller, HDMI controller etc.  
**SPI :** Used for connecting to non-volatile storage on serial flash devices.

**Software**

STB Software is organized as layered architecture as shown in block diagram below. RTOS is generally STB company proprietary or some industry standard OS such as Linux. Software drivers are written for all hardware blocks and some software components. A typical STB has 30-40 different drivers. Middleware is generally used to standardize the interfaces from drivers to application so that device independence can be provided. Final application is on top of middleware and it usually remains same for one service provider across different STBs.

[](http://embeddedtutorial.com/wp-content/uploads/2010/09/SetTopBoxSoftware.PNG)

Set Top Box Software Architecture

There are normally two application loaded on each STB. One is the boot loader and other is main application. Boot loader is responsible for downloading main application Over The Air (Application broadcasted over air by service provider) and updating the main application. The main application is responsible for all the features/ functionality which the end user sees.

STB software application has very high complexity. A typical STB application including drivers consists of 0.5-0.6 million lines of code.

So STB does a great job of hiding this advance level of complexity in its simple form factor. Its application is also designed by keeping various users in mind, so the complexity of software is completely hidden from end users perspective.

|  |
| --- |
| **1. What is the MAIN benefit of designing tests early in the life cycle?**  It helps prevent defects from being introduced into the code.  **2. What is risk-based testing?**  Risk-based testing is the term used for an approach to creating a test strategy that is based on prioritizing tests by risk. The basis of the approach is a detailed risk analysis and prioritizing of risks by risk level. Tests to address each risk are then specified, starting with the highest risk first.  **3. A wholesaler sells printer cartridges. The minimum order quantity is 5. There is a 20% discount for orders of 100 or more printer cartridges. You have been asked to prepare test cases using various values for the number of printer cartridges ordered. Which of the following groups contain three test inputs that would be generated using**[**Boundary Value Analysis**](http://www.guru99.com/equivalence-partitioning-boundary-value-analysis.html)**?**  4, 5, 99  **4. What is the KEY difference between preventative and reactive approaches to testing?**  Preventative tests are designed early; reactive tests are designed after the software has been produced. |

**5. What is the purpose of exit criteria?**

To define when a test level is complete.

**6. What determines the level of risk?**

 The likelihood of an adverse event and the impact of the event

**7. When is used Decision table testing?**

 Decision table testing is used for testing systems for which the specification takes the form of rules or cause-effect combinations. In a decision table the inputs are listed in a column, with the outputs in the same column but below the inputs. The remainder of the table explores combinations of inputs to define the outputs produced.

Learn More About Decision Table Testing Technique in the Video Tutorial [here](http://www.guru99.com/software-testing-techniques-1.html)

**8. What is the MAIN objective when reviewing a software deliverable?**

To identify defects in any software work product.

**9. Which of the following defines the expected results of a test?**[**Test case**](http://www.guru99.com/test-case.html)**specification or test design specification.**

Test case specification.

**10. Which is a benefit of test independence?**

It avoids author bias in defining effective tests. 

**11. As part of which test process do you determine the exit criteria?**

Test planning. 

**12. What is beta testing?**

Testing performed by potential customers at their own locations. 

**13. Given the following fragment of code, how many tests are required for 100% decision coverage?**

**if width > length**

**then biggest\_dimension = width**

**if height > width**

**then biggest\_dimension = height**

**end\_if**

**else biggest\_dimension = length**

**if height > length**

**then biggest\_dimension = height**

**end\_if**

**end\_if**

4 

**14. You have designed test cases to provide 100% statement and 100% decision coverage for the following fragment of code. if width > length then biggest\_dimension = width else biggest\_dimension = length end\_if The following has been added to the bottom of the code fragment above. print "Biggest dimension is " & biggest\_dimension print "Width: " & width print "Length: " & length How many more test cases are required?**

None, existing test cases can be used. 

**15. Rapid Application Development ?**

Rapid Application Development (RAD) is formally a parallel development of functions and subsequent integration. Components/functions are developed in parallel as if they were mini projects, the developments are time-boxed, delivered, and then assembled into a working prototype. This can very quickly give the customer something to see and use and to provide feedback regarding the delivery and their requirements. Rapid change and development of the product is possible using this methodology. However the product specification will need to be developed for the product at some point, and the project will need to be placed under more formal controls prior to going into production.

**16. What is the difference between Testing Techniques and**[**Testing Tools**](http://www.guru99.com/testing-tools.html)**?**

Testing technique: – Is a process for ensuring that some aspects of the application system or unit functions properly there may be few techniques but many tools.

Testing Tools: – Is a vehicle for performing a test process. The tool is a resource to the tester, but itself is insufficient to conduct testing 

Learn More About Testing Tools  [here](http://www.guru99.com/testing-tools.html)

**17. We use the output of the requirement analysis, the requirement specification as the input for writing …**

User Acceptance Test Cases 

**18. Repeated Testing of an already tested program, after modification, to discover any defects introduced or uncovered as a result of the changes in the software being tested or in another related or unrelated software component:**

[Regression Testing](http://www.guru99.com/regression-testing.html)

**19. What is component testing ?**

Component testing, also known as unit, module and program testing, searches for defects in, and verifies the functioning of software (e.g. modules, programs, objects, classes, etc.) that are separately testable. Component testing may be done in isolation from the rest of the system depend-ing on the context of the development life cycle and the system. Most often stubs and drivers are used to replace the missing software and simulate the interface between the software components in a simple manner. A stub is called from the software component to be tested; a driver calls a component to be tested.

**20. What is functional**[**system testing**](http://www.guru99.com/system-testing.html)**?**

Testing the end to end functionality of the system as a whole.

**21. What is the benefits of Independent Testing**

Independent testers see other and different defects and are unbiased. 

**22. In a REACTIVE approach to testing when would you expect the bulk of the test design work to be begun?**

After the software or system has been produced.

**23. What are the different Methodologies in Agile Development Model?**

There are currently seven different Agile methodologies that I am aware of:

1. Extreme Programming (XP)
2. Scrum
3. Lean Software Development
4. Feature-Driven Development
5. Agile Unified Process
6. Crystal
7. Dynamic Systems Development Model (DSDM)

**24. Which activity in the fundamental test process includes evaluation of the testability of the requirements and system?**

A Test analysis and design.

**25. What is typically the MOST important reason to use risk to drive testing efforts?**

 Because testing everything is not feasible. 

**26. Which is the MOST important advantage of independence in testing?**

An independent tester may be more effective at finding defects missed by the person who wrote the software. 

**27. Which of the following are valid objectives for incident reports?**

**i. Provide developers and other parties with feedback about the problem to enable identification, isolation and correction as necessary.**

**ii. Provide ideas for test process improvement.**

**iii. Provide a vehicle for assessing tester competence.**

**iv. Provide testers with a means of tracking the quality of the system under test.**

i. Provide developers and other parties with feedback about the problem to enable identification, isolation and correction as necessary,

ii.Provide ideas for test process improvement,

iv.Provide testers with a means of tracking the quality of the system under test

**28. Consider the following techniques. Which are static and which are dynamic techniques?**

**i.**[**Equivalence Partitioning**](http://www.guru99.com/equivalence-partitioning-boundary-value-analysis.html)**.**

**ii. Use Case Testing.**

**iii.Data Flow Analysis.**

**iv.Exploratory Testing.**

**v. Decision Testing.**

**vi. Inspections.**

Data Flow Analysis and Inspections are static, Equivalence Partitioning, Use Case Testing, Exploratory Testing and Decision Testing are dynamic.

**29. Why are**[**static testing**](http://www.guru99.com/testing-review.html)**and dynamic testing described as complementary?**

Because they share the aim of identifying defects but differ in the types of [defect](http://www.guru99.com/software-defect.html) they find. 

**30. What are the phases of a formal**[**review**](http://www.guru99.com/testing-review.html)**?**

In contrast to informal reviews, formal reviews follow a formal process. A typical formal review process consists of six main steps:

1. Planning
2. Kick-off
3. Preparation
4. Review meeting
5. Rework
6. Follow-up.

**31. What is the role of moderator in review process?**

The moderator (or review leader) leads the review process. He or she deter-mines, in co-operation with the author, the type of review, approach and the composition of the review team. The moderator performs the entry check and the follow-up on the rework, in order to control the quality of the input and output of the review process. The moderator also schedules the meeting, disseminates documents before the meeting, coaches other team members, paces the meeting, leads possible discussions and stores the data that is collected.

Learn More About Review process in Video Tutorial [here](http://www.guru99.com/testing-review.html)

**32. What is an equivalence partition (also known as an equivalence class)?**

An input or output range of values such that only one value in the range becomes a test case. 

**33. When should configuration management procedures be implemented?**

During test planning.

**34. A Type of functional Testing, which investigates the functions relating to detection of threats, such as virus from malicious outsiders.**

Security Testing 

**35. Testing where in we subject the target of the test , to varying workloads to measure and evaluate the performance behaviors and ability of the target and of the test to continue to function properly under these different workloads.** Load Testing

**36. Testing activity which is performed to expose defects in the interfaces and in the interaction between integrated components is:**

Integration Level Testing 

**37. What are the Structure-based (white-box) testing techniques ?**

Structure-based testing techniques (which are also dynamic rather than static) use the internal structure of the software to derive test cases. They are com-monly called 'white-box' or 'glass-box' techniques (implying you can see into the system) since they require knowledge of how the software is implemented, that is, how it works. For example, a structural technique may be concerned with exercising loops in the software. Different test cases may be derived to exercise the loop once, twice, and many times. This may be done regardless of the func-tionality of the software. 

**38. When should be performed Regression testing ?**

After the software has changed or when the environment has changed

**39. When should testing be stopped?**

It depends on the risks for the system being tested

**40. What is the purpose of a test completion criterion?**

To determine when to stop testing 

**41. What can static analysis NOT find?**

For example memory leaks 

**42. What is the difference between re-testing and regression testing?**

Re-testing ensures the original fault has been removed; regression testing looks for unexpected sideeffects 

**43. What are the Experience-based testing techniques ?**

In experience-based techniques, people's knowledge, skills and background are a prime contributor to the test conditions and test cases. The experience of both technical and business people is important, as they bring different perspectives to the test analysis and design process. Due to previous experience with similar systems, they may have insights into what could go wrong, which is very useful for testing. 

**44. What type of review requires formal entry and exit criteria, including metrics?** Inspection **45. Could reviews or inspections be considered part of testing?**

Yes, because both help detect faults and improve quality 

**46. An input field takes the year of birth between 1900 and 2004 What are the boundary values for testing this field ?**1899,1900,2004,2005 

**47. Which of the following tools would be involved in the automation of regression test? a. Data tester b. Boundary tester c. Capture/Playback d. Output comparator.**

d. Output comparator 

**48. To test a function,what has to write a programmer, which calls the function to be tested and passes it**[**test data**](http://www.guru99.com/software-testing-test-data.html)**.**

 Driver

**49. What is the one Key reason why developers have difficulty testing their own work?**

Lack of Objectivity

**50.“How much testing is enough?”**

The answer depends on the risk for your industry, contract and special requirements. **51. When should testing be stopped?**It depends on the risks for the system being tested. 

**52. Which of the following is the main purpose of the integration strategy for**[**integration testing**](http://www.guru99.com/integration-testing.html)**in the small?**

To specify which modules to combine when, and how many at once.

**53. What is the purpose of a test completion criterion?**

 To determine when to stop testing 

**54. Given the following code, which statement is true about the minimum number of test cases required for full statement and branch coverage?**

**Read p**

**Read q**

**IF p+q> 100**

**THEN Print "Large"**

**ENDIF**

**IF p > 50**

**THEN Print "p Large"**

**ENDIF**

 1 test for statement coverage, 2 for branch coverage

**55. What is the difference between re-testing and regression testing?**

 Re-testing ensures the original fault has been removed; regression testing looks for unexpected side-effects. 

**56. Which review is normally used to evaluate a product to determine its suitability for intended use and to identify discrepancies?**

Technical Review.

**57. Why we use decision tables?.**

The techniques of equivalence partitioning and boundary value analysis are often applied to specific situations or inputs. However, if different combinations of inputs result in different actions being taken, this can be more difficult to show using equivalence partitioning and boundary value analysis, which tend to be more focused on the user interface. The other two specification-based tech-niques, decision tables and state transition testing are more focused on business logic or business rules. A decision table is a good way to deal with combinations of things (e.g. inputs). This technique is sometimes also referred to as a 'cause-effect' table. The reason for this is that there is an associated logic diagramming technique called 'cause-effect graphing' which was sometimes used to help derive the decision table

**58. Faults found should be originally documented by who?**

By testers. 

**59. Which is the current formal world-wide recognized documentation standard?**

There isn’t one. 

**60. Which of the following is the review participant who has created the item to be reviewed?**

Author

**61. A number of critical bugs are fixed in software. All the bugs are in one module, related to reports. The test manager decides to do regression testing only on the reports module.**

Regression testing should be done on other modules as well because fixing one module may affect other modules.

**62. Why does the boundary value analysis provide good test cases?**

Because errors are frequently made during programming of the different cases near the ‘edges’ of the range of values. 

**63. What makes an inspection different from other review types?**

It is led by a trained leader, uses formal entry and exit criteria and checklists.

**64. Why can be tester dependent on configuration management?**

Because configuration management assures that we know the exact version of the testware and the test object.

**65. What is a V-Model ?**

A software development model that illustrates how testing activities integrate with software development phases

**66. What is maintenance testing?**

Triggered by modifications, migration or retirement of existing software 

**67. What is test coverage?**

Test coverage measures in some specific way the amount of testing performed by a set of tests (derived in some other way, e.g. using specification-based techniques). Wherever we can count things and can tell whether or not each of those things has been tested by some test, then we can measure coverage. 

**68. Why is incremental integration preferred over “big bang” integration?**

Because incremental integration has better early defects screening and isolation ability 

**69. When do we prepare RTM (Requirement**[**traceability matrix**](http://www.guru99.com/traceability-matrix.html)**), is it before test case designing or after test case designing?**

The would be before. Requirements should already be traceable from Review activities since you should have traceability in the [Test Plan](http://www.guru99.com/test-plan.html) already. This question also would depend on the organisation. If the organisation do test after development started then requirements must be already traceable to their source. To make life simpler use a tool to manage requirements.

**70. What is called the process starting with the terminal modules ?**

Bottom-up integration 

**71. During which test activity could faults be found most cost effectively?**

During test planning 

**72. The purpose of requirement phase is**

To freeze requirements, to understand user needs, to define the scope of testing

**73. How much testing is enough?**

The answer depends on the risks for your industry, contract and special requirements **74. Why we split testing into distinct stages?** Each test stage has a different purpose.

**75. Which of the following is likely to benefit most from the use of test tools providing test capture and replay facilities? a) Regression testing b) Integration testing c) System testing d)**[**User acceptance testing**](http://www.guru99.com/user-acceptance-testing.html)

Regression testing

**76. How would you estimate the amount of re-testing likely to be required?**

Metrics from previous similar projects and discussions with the development team

**77. What studies data flow analysis ?**

The use of data on paths through the code.

**78. What is Alpha testing?**

Pre-release testing by end user representatives at the developer’s site.

**79. What is a failure?**

Failure is a departure from specified behaviour. 

**80. What are Test comparators ?**

Is it really a test if you put some inputs into some software, but never look to see whether the software produces the correct result? The essence of testing is to check whether the software produces the correct result, and to do that, we must compare what the software produces to what it should produce. A test comparator helps to automate aspects of that comparison.

**81. Who is responsible for document all the issues, problems and open point that were identified during the review meeting**Scribe 

**82. What is the main purpose of Informal review**

Inexpensive way to get some benefit 

**83. What is the purpose of test design technique?**

Identifying test conditions and Identifying test cases

**84. When testing a grade calculation system, a tester determines that all scores from 90 to 100 will yield a grade of A, but scores below 90 will not. This analysis is known as:**

 Equivalence partitioning 

**85. A test manager wants to use the resources available for the automated testing of a web application. The best choice is**Tester, test automater, web specialist, DBA 

**86. During the testing of a module tester ‘X’ finds a**[**bug**](http://www.guru99.com/software-defect.html)**and assigned it to developer. But developer rejects the same, saying that it’s not a bug. What ‘X’ should do?**

Send to the detailed information of the bug encountered and check the reproducibility

**87. A type of integration testing in which software elements, hardware elements, or both are combined all at once into a component or an overall system, rather than in stages.**

Big-Bang Testing 

**88. In practice, which Life Cycle model may have more, fewer or different levels of development and testing, depending on the project and the software product. For example, there may be component integration testing after component testing, and system integration testing after system testing.**

V-Model

**89. Which technique can be used to achieve input and output coverage? It can be applied to human input, input via interfaces to a system, or interface parameters in integration testing.**

Equivalence partitioning 

**90. “This life cycle model is basically driven by schedule and budget risks” This statement is best suited for…**

V-Model 

**91. In which order should tests be run?**

The most important tests first 

**92. The later in the development life cycle a fault is discovered, the more expensive it is to fix. why?**

The fault has been built into more documentation,code,tests, etc 

**93. What is Coverage measurement?**

It is a partial measure of test thoroughness. 

**94. What is Boundary value testing?**

Test boundary conditions on, below and above the edges of input and output equivalence classes.

**95. What is Fault Masking ?**

Error condition hiding another error condition.

**96. What does COTS represent?**

Commercial Off The Shelf. 

**97.The purpose of wich is allow specific tests to be carried out on a system or network that resembles as closely as possible the environment where the item under test will be used upon release?**

Test Environment

**98. What can be though of as being based on the project plan, but with greater amounts of detail?**

Phase Test Plan 

**99. What is exploratory testing?**

 Exploratory testing is a hands-on approach in which testers are involved in minimum planning and maximum test execution. The planning involves the cre-ation 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 typi-cally 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 use 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. 

**100. What is failure?**

Deviation from expected result to actual result