

Manual Testing Interview Questions for 2-3 YO

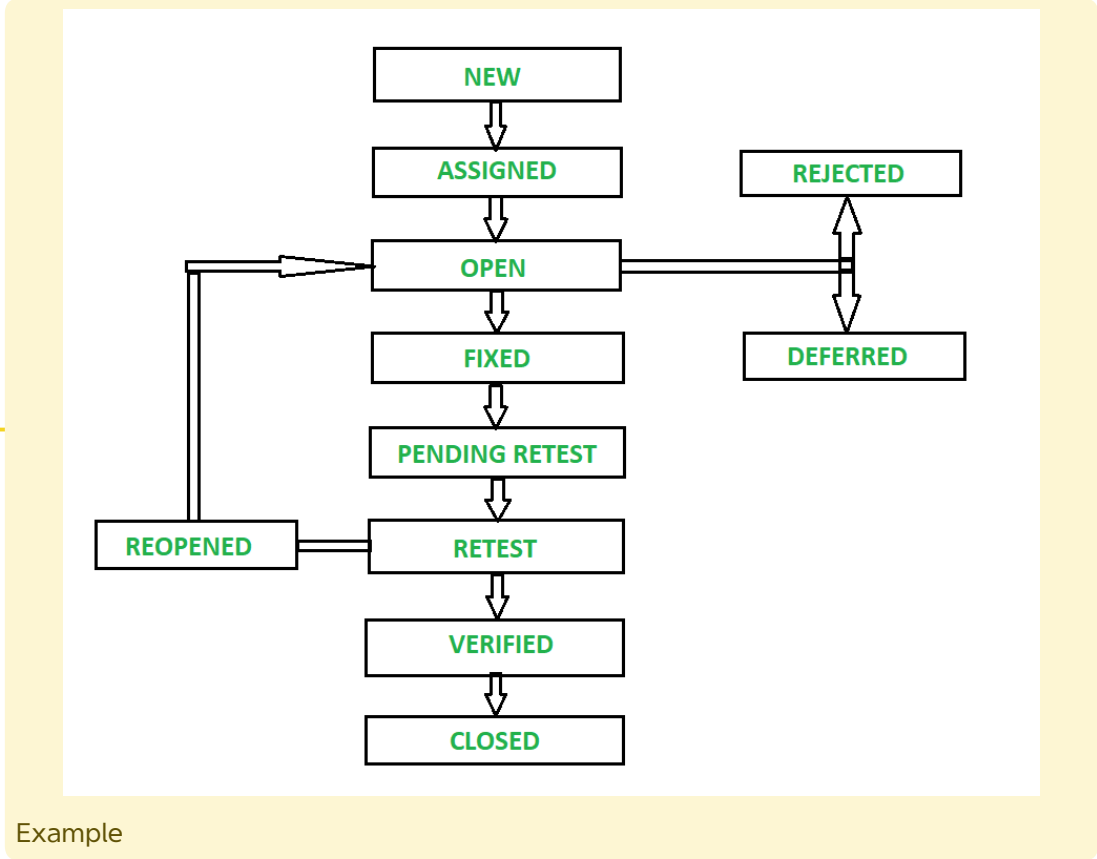
Q 1 - What is Requirement Traceability Matrix ?

- Download the RTM <https://scrolltest.com/rtm>
- What is RTM?
- It is used to Track the Requirements and check the current progress if requirements are met or not.
- Full Video on Different Type of RTM and more information <https://sdet.live/rtm>
- RTM is a document which traces or maps the Requirement with Test cases
- Help in planning release, map the correct req. to test cases to avoid bugs and changes, Improve Test Coverage.
- You can prevent defect leakage, good test coverage, improve quality.

Q 2 - What is Exploratory Test ?

- Explore the Application
- Tester don't create Test cases in Advance
- Check the functionality in Adhoc Manner
- We generally do scripted testing, ET is opposite of Scripted Testing
- Most of the time it is time boxed, 1 hour, 2 hour extra and in the end people share bugs and flows
- It is skilled based as it is not scripted
- When use exploratory testing?
 - The testing team has experienced testers
 - Early iteration is required
 - There is a critical application
 - New testers entered into the team
- Read more
- Test Charter 1
 - Charter- Analyze Login feature of Website
 - Areas- Login as Existing user with Username and Password
 - Login as Existing user with Google account
 - Login as Existing user with Facebook account
 - Enter incorrect username and password to verify validation message
 - Block your username and verify the validation message
 - Use Forgot Password to reset password
 - And so on
 - Test Environment & URL- SIT
 - Start- 9/18/2017 10:00 AM (Start date and time)
 - Tester - the tester's name
 - Duration-Normal (60 mins)
 - Task Breakdown- Bug Logging, New feature, Backward Compatibility
 - Testing Notes- This acts like a journal of exploratory testing done.
 - Open login page
 - Verify default screen
 - Verify Existing & New user account hyperlink exist
 - Open Existing User Login
 - Verify successful login with google& Facebook, username
 - Verify validations
 - Data files- screen shots, recordings, files
 - Bugs- Bugs found during testing. Each defect should be described in one sentence and include tracker id
 - Issues- Issues encountered during testing. Provide list with description in one line
 - Session Setup Time- % of time required to setup
 - Test Design and Execution- % of time spent on this phase
- Sample

Q 3 - Explain Defect life cycle.



Q 4 - 7 principles of software testing

- 7 principles of testing
 - Principle 1- Testing shows presence of defects
 - Principle 2- Exhaustive testing is impossible
 - Principle 3- Early Testing
 - Principle 4- Defect Clustering
 - Principle 5- Pesticide Paradox
 - Principle 6- Testing is context dependent
 - Principle 7- Absence of errors fallacy
- 1. Testing shows presence of defects
 - Software testing reduces the presence of defects.
 - Software testing talks about the presence of defects and doesn't talk about the absence of defects
 - Even multiple testing can never ensure that software is 100% bug free
- 2. Exhaustive testing is not possible
 - Exhaustive testing is impossible means the software can never test at every test cases
 - It can test only some test cases and assume that software is correct and it will produce the correct output in every test cases
- 3. Early Testing
 - The defect detected in early phases of SDLC will very less expensive.
 - For better performance of software, start software testing will start at initial phase
- 4. Defect clustering
 - In a project, a small number of the module can contain most of the defects.
 - Pareto Principle to software testing state that 80% of software defect comes from 20% of modules.
- 5. Pesticide paradox
 - Repeating the same test cases again and again will not find new bugs
 - It is necessary to review the test cases and add or update test cases to find new bugs.
- 6. Testing is context dependent:
 - Testing approach depends on context of software developed.
 - Different types of software need to perform different types of testing.
 - The testing of the e-commerce site is different from the testing of the Android application.
- 7. Absence of errors fallacy
 - If a built software is 99% bug free but it does not follow the user requirement then it is unusable.
 - It is not only necessary that software is 99% bug free but it also mandatory to fulfil all the customer requirements

Q 5 - Real World Scenarios Questions

How will you estimate a login page with dashboard and What type of Testing you will perform?

- app.vwo.com
- login and dashboard page
- Estimation in men days
- Type of testing performed?
- CBT?
- Don't ask or Assume

What to do if Web app or Mobile App that you are testing is not suitable to test?

- Smoke Testing
- Ignore the build
- Sent a report saying build is not testable

Developer said, Please don't add this bug to JIRA What to do?

- avoid brotherhood
- Raise every concern if possible

What to do if you have less time to test?

- Proper estimates
- Let them know about the risk
- Tell them about the process and risk and things you are doing
- Ask for the risk

When to stop testing

- Deadlines (release deadlines, testing deadlines, etc.)
- Test cases completed with certain percentage passed
- When the test budget is depleted
- Coverage of code or functionality or requirements reaches a specified point
- Bug rate falls below a certain level
- When Beta or alpha testing period ends