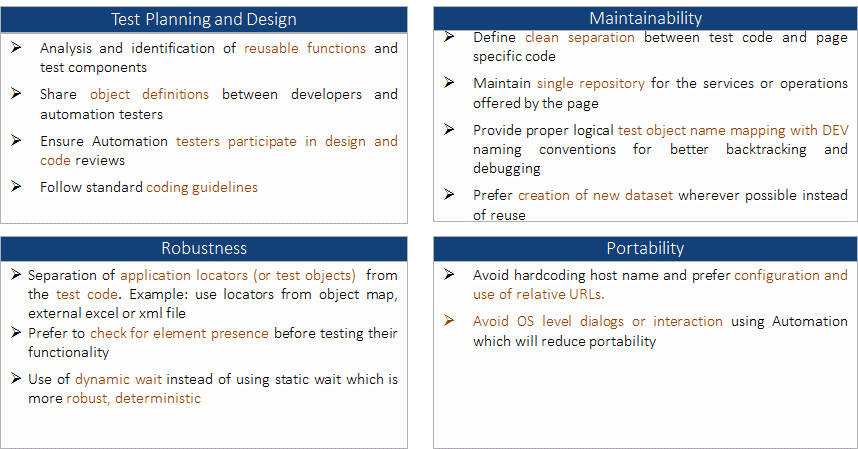
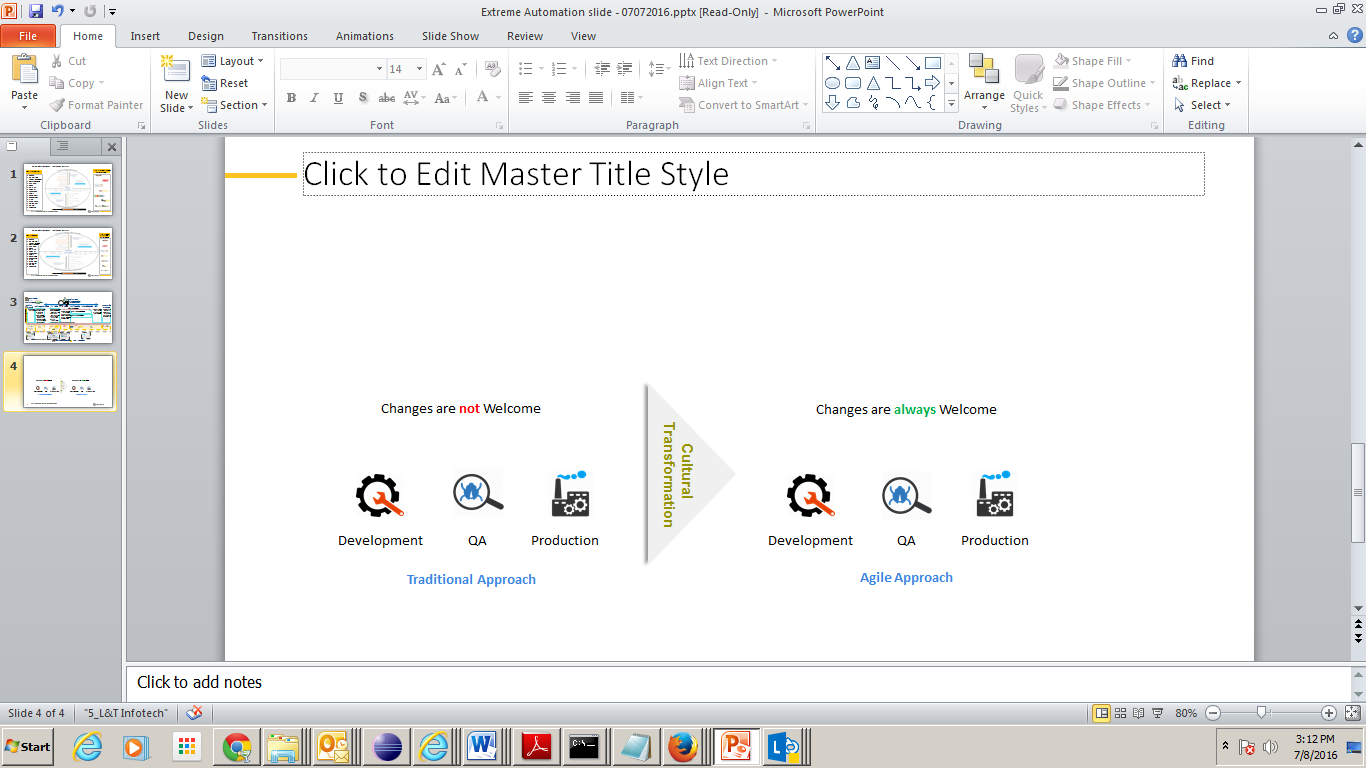
**Test Automation Best Practices:**

LTI understands that Test automation plays vital role in testing and success of automation primarily depends on Automation strategy and design. Below diagram represents guidelines need to be considered in automation life cycle,

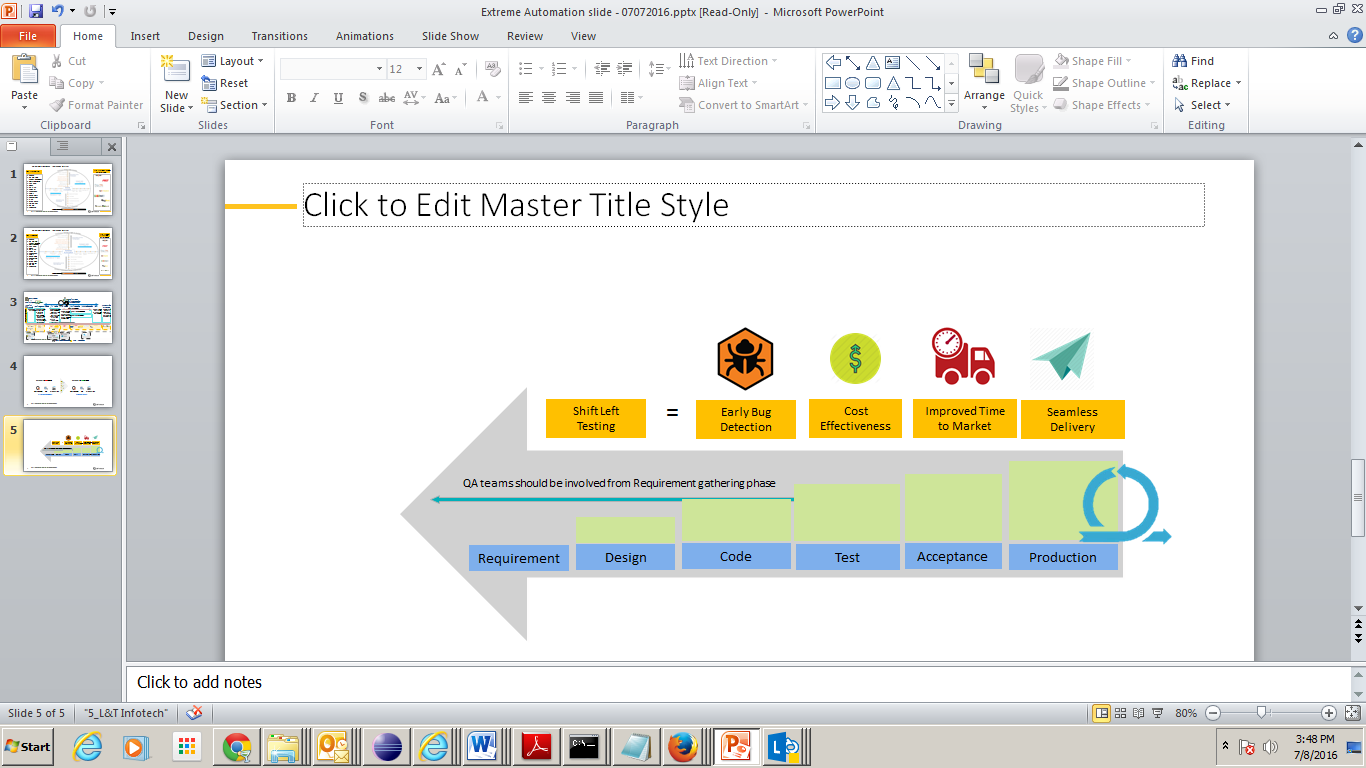


Below are detailed Automation best practices:

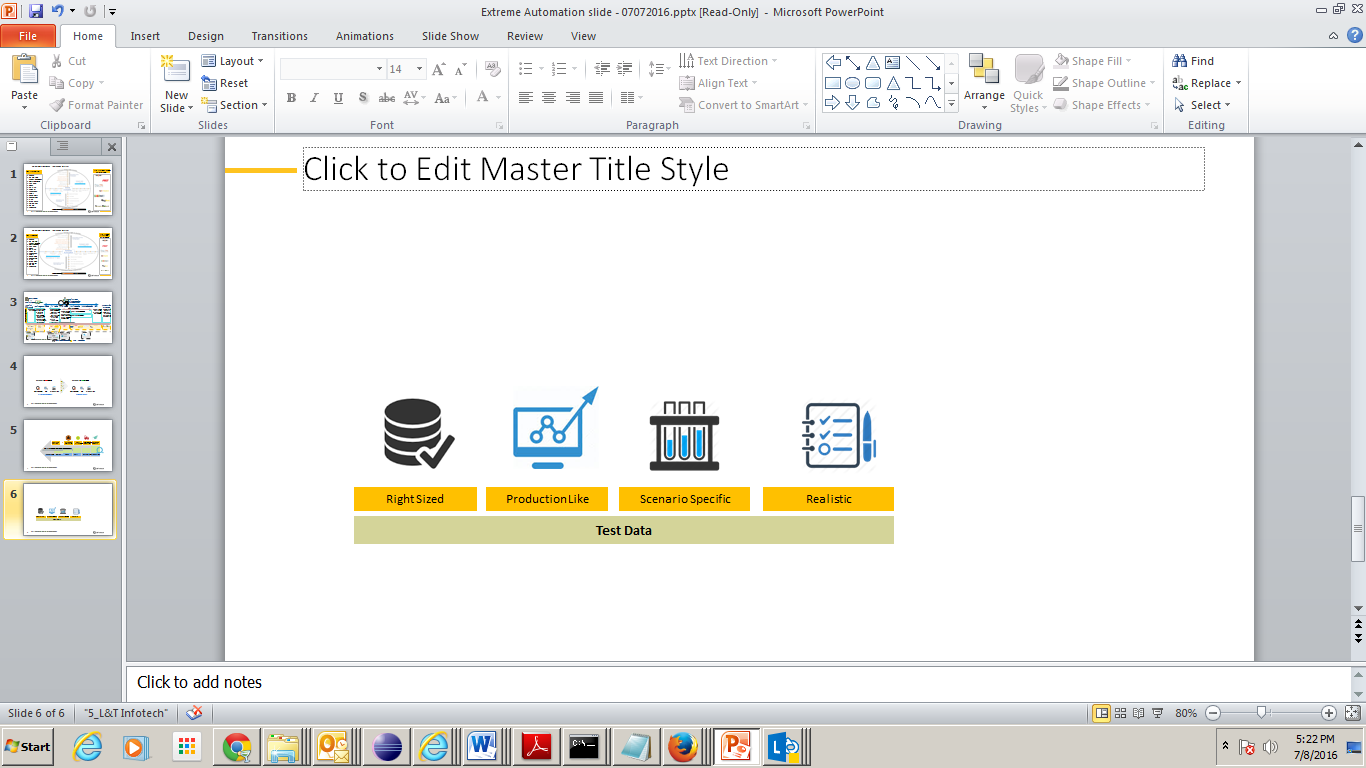
* **Mindset Change** – Change is constant in any agile development model. Organizations need to induct its people in adapting to the changed work culture. Becoming agile doesn’t happen in a day, it’s a mindset that one needs to cultivate continuously. Whole team approach is very much necessary, wherein the onus of quality is not the responsibility of any person in particular but of the entire team. Developers should also focus on QA aspects while the testers should look beyond the usual QA activities and also become more technically aligned. Developers should not fear making changes to their code. Additionally sustainable environment should be made available which is easy to use and understand. Intuitive processes and tools can be brought into practice to make this transition smoother.



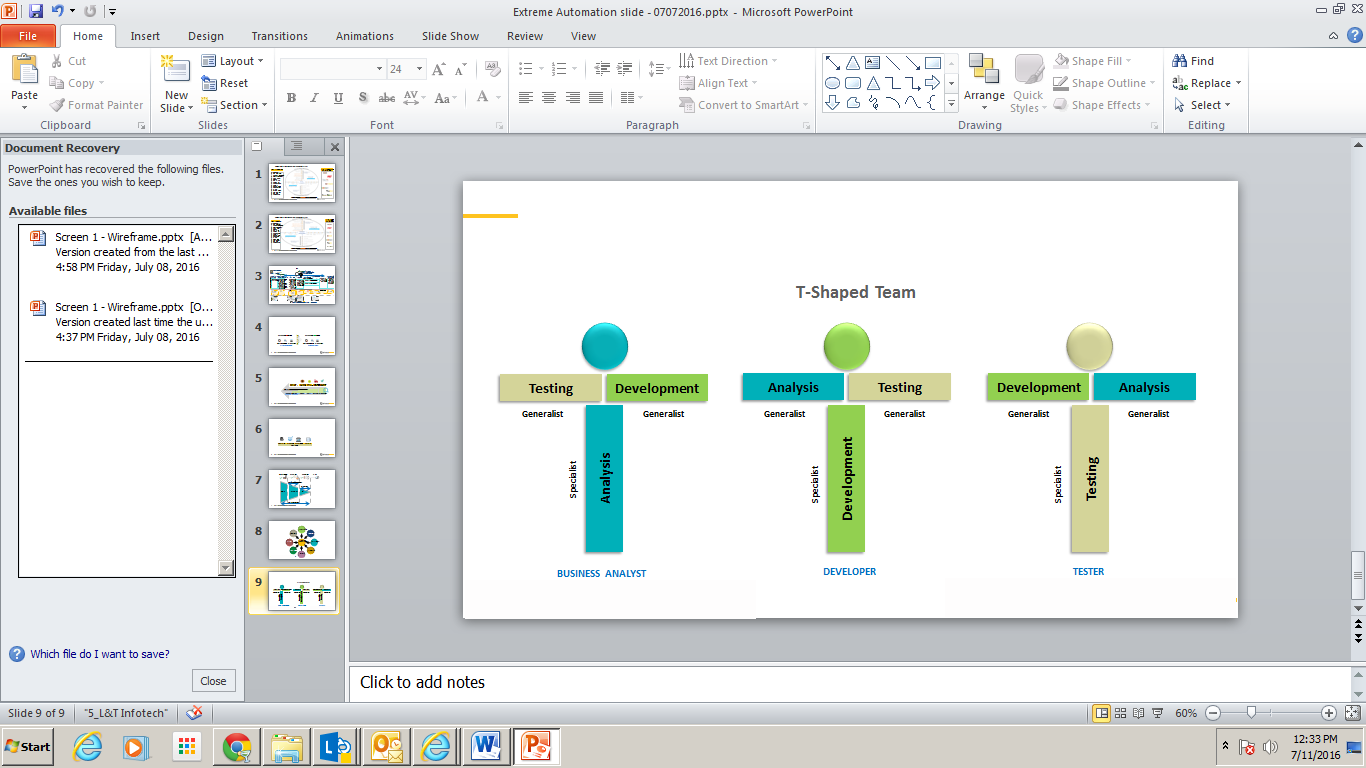
* **Test Early and Test Often** – Organizations are constantly trying hard to [reduce costs, increase revenue, and speed up time to market](http://techarcis.com/services/environment-management/). This is where lies the importance of Shift left testing. Test Automation should be introduced as early as requirement analysis and should be implemented across the application lifecycle. The earlier QA teams get involved in the life cycle of the project the better, and the more you test, the more bugs you find. Improving upstream quality is the key to success here. This could be achieved by introducing static testing requirements, boosting system test, assuring code quality among others.



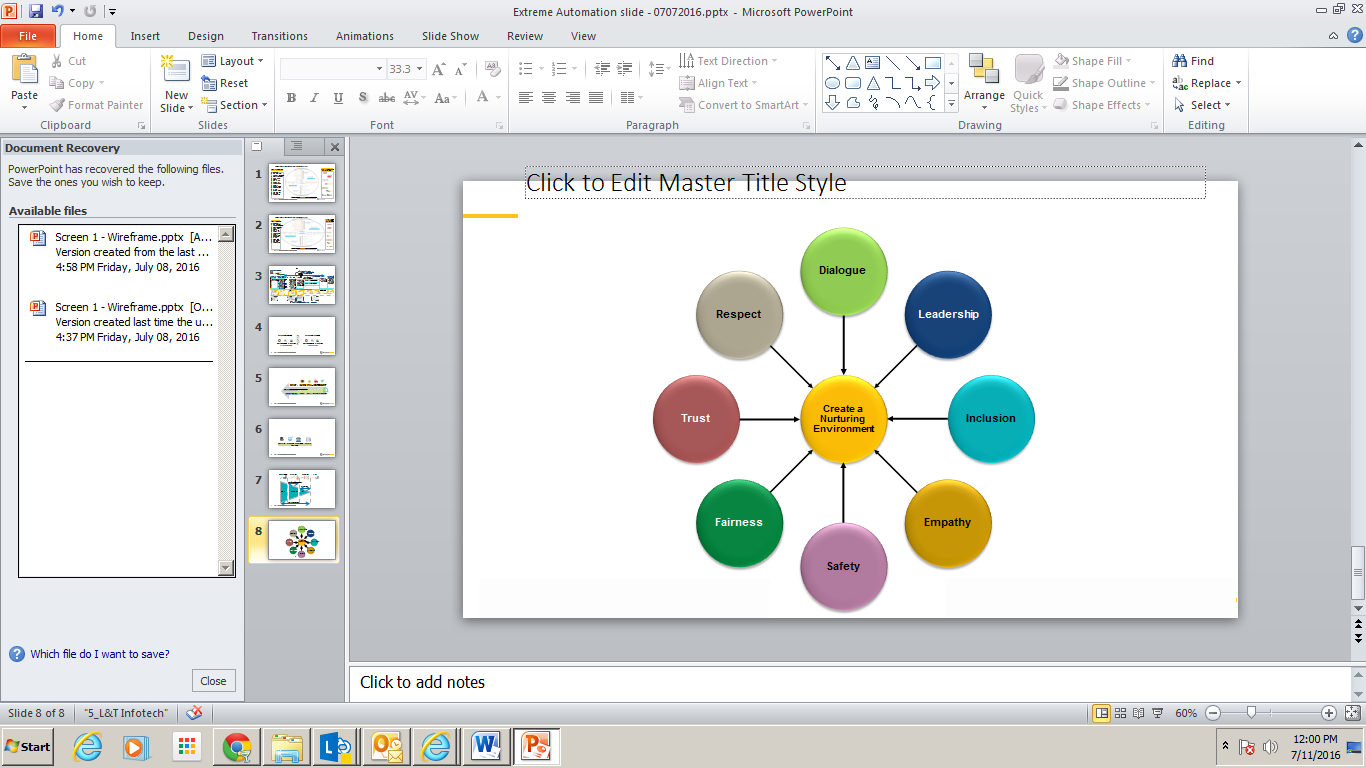
* **Work on the ‘Lean’ theme:** With CI, changes are split into small increments. Developers work towards creating a stable system by working in small batches and short cycles which is in-sync with the lean theme. This enables teams to work with shared code and increases the visibility into the development and quality of the system.
* **Create good quality Test data –** To increase the business and sustain growth, it is every organization’s responsibility to deliver high quality software. It could be about building a new application, migrating from one system to another or re-engineering; it has to be tested thoroughly with realistic, production-like, scenario-specific, right-sized test data in order to deliver high quality software. Every scenario needs to be tested with specific, right-sized, realistic test data to ensure the quality and efficiency. Hence, test data plays an important role for both functional and non-functional testing for delivering high quality software.



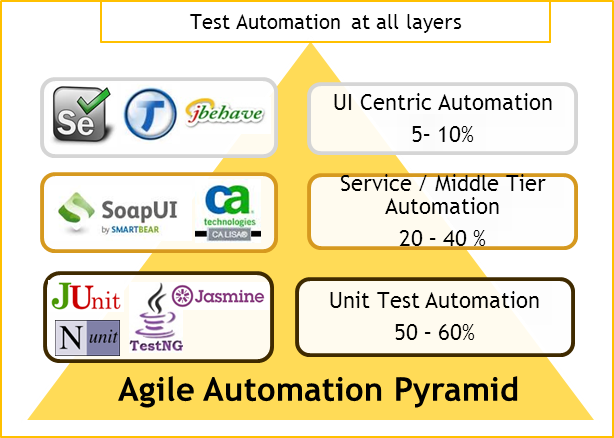
* **Encouraging cross-skilling in the team** – All members in an Agile team should have all the necessary skills to build and deliver the optimized solution, encompassing technical and business skills. The industry is moving towards a T-shaped model. Developing T-shaped skilled engineers does not only mean technical skills but also soft skills and domain skills. A team of t-shaped people, each with different skills and specialties, enables team members to complement one another and form a high performing team. Teams with these cross-functional members, are higher performing as they have less internal bottlenecks and contention for one person’s time.



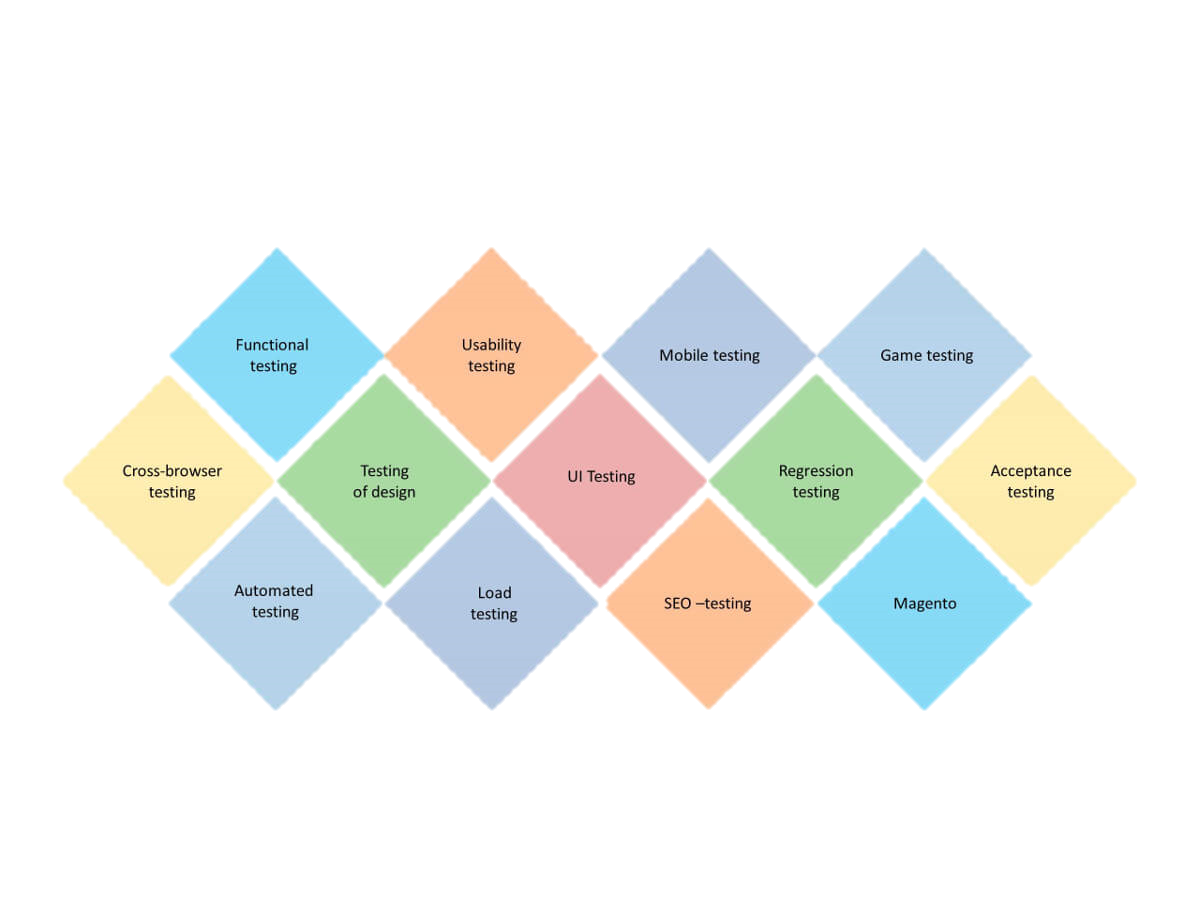
* **Collaboration across distributed teams** –Working with distributed teams introduces both technical and cultural diversity to any company, which is integral to excellence. In an agile development environment there is no place for individual heroism. It’s the whole team that succeeds or fails. For successful collaboration, mutual respect and open communication adds to project success. Pair programming, cross functional teams, frequent communication between stakeholders encourages team collaboration. In addition to that teams can also take help of many collaboration tools like Digital taskboards, video conferencing, online meeting etc.



* **Testing across multiple platforms and interfaces –** Such automation tool should be selected which is capable of testing across multiple platforms, technologies and interfaces. This ensures we can perform testing in a heterogeneous environment rather than having a homogenous environment. Organizations are expected to provide seamless customer experiences regardless of the platform.



* **Choosing a ‘standard’ Automation tool/Framework:** Selecting an automation tool that supports standard scripting language not only improves the productivity, it also reduces the learning curve.
* **Support for Various Levels Types -** To enhance the quality of software testing, and to produce a more unified testing methodology applicable across several projects, the testing process could be abstracted to different levels. This classification into different levels introduces some parallelism in the testing process as multiple tests could be performed simultaneously. It is important to test at different levels because sources of errors in the software can then be easily tracked and fixed



* **Remote Test Script Execution –** Remote test script execution allows the testers to do the testing in multiple machines at one time. It also allows the test to be executed in the machine where the test tool is not installed. If this comes with scheduling tests, synchronization etc, it will reduce the manual work further.
* **Speed up feedback process:** Development team and stakeholders should get intimation about the failure as soon as it occurs to prevent any delay in fixing them. For speeding up continuous build feedback, each sub-team may have their own stream and run continuous builds until they are free of critical errors, and then deliver to a higher level integration stream to run a complete team build.
* **Virtualization wherever possible -** Organizations are on the lookout for the latest technologies that allow businesses to run seamlessly with fewer resources while providing the infrastructure to meet current and future customer needs. Use of virtualization enables rapid deployment by isolating the application in a known and controlled environment. Virtualization makes it possible to maximize computer utilization while minimizing all associated overheads of downtime management, hardware requirement, maintenance and physical space. Service virtualization can be applied at all levels, even allowing integration testing to begin in parallel with unit testing.

