eBook

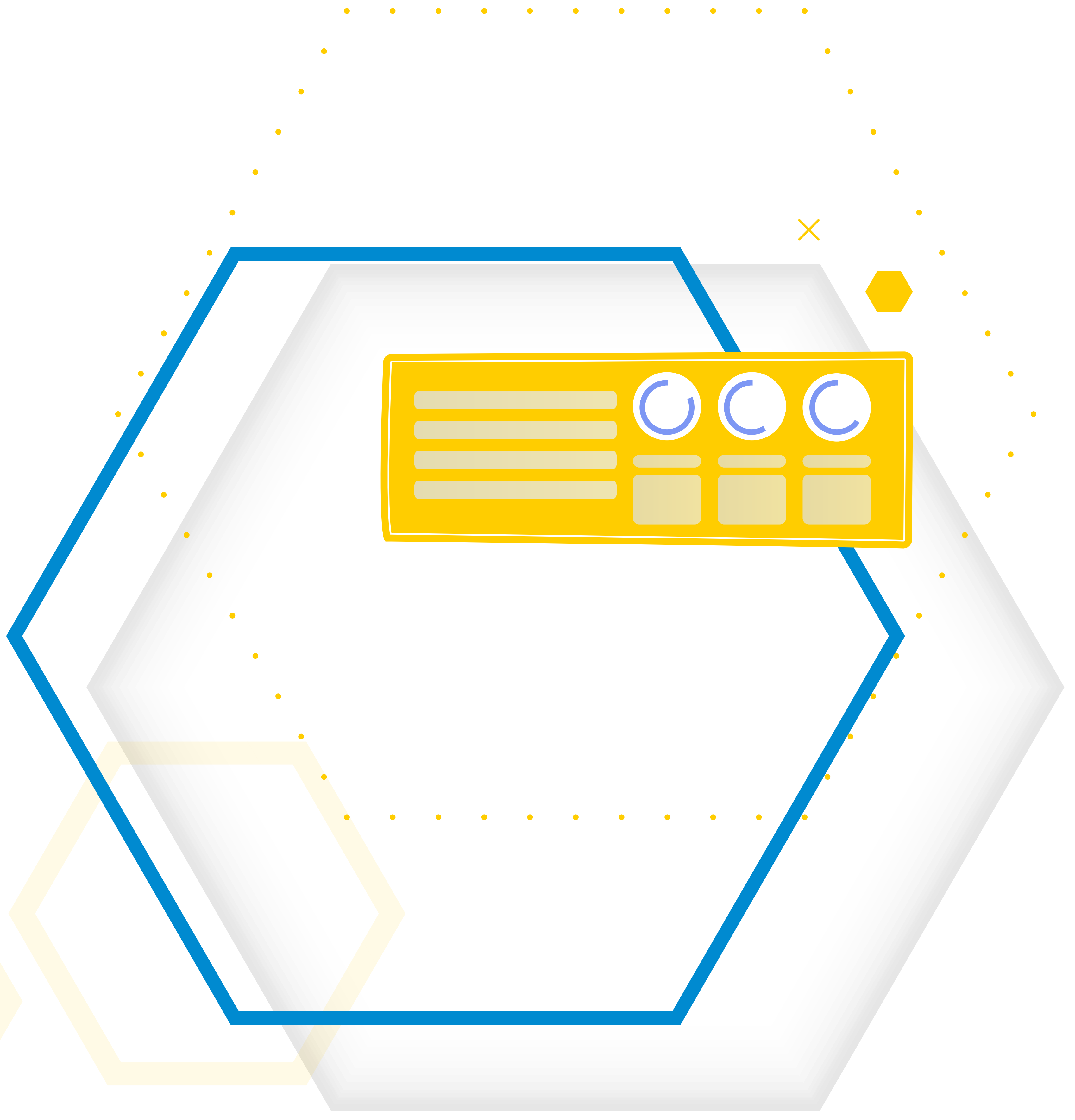
The Road to QA Maturity

ZUCI SYSTEMS

High-quality software products are hugely in demand. Customers are not satisfied with subpar software products. Amidst the cutthroat competition, IT companies must have a smart and efficient QA team with people and processes that always exceeds customer expectations. In the agile business model, QA teams play a significant role in delivering the best end products. With a streamlined QA process, your company can reduce operational costs while improving the organization's reputation.

A mature QA process is essential in agile project development. The right QA maturity model for your organization enables you to measure the QA practices currently used and determine improvement scope. With QA maturity assessment, gaps in the current QA system can be identified and rectified. This ensures a better result for the end-user. The Software Engineering Institute at Carnegie Mellon University has created a process maturity model effectively adopted by numerous organizations worldwide.

The QA maturity process model plays a major role in organizational structure. It will help improve the quality of the result and demonstrate the progress to important stakeholders. The improvement initiatives for delivered products will directly improve the organization's performance. The QA maturity model indicates the progress made and changes required for success in the future.

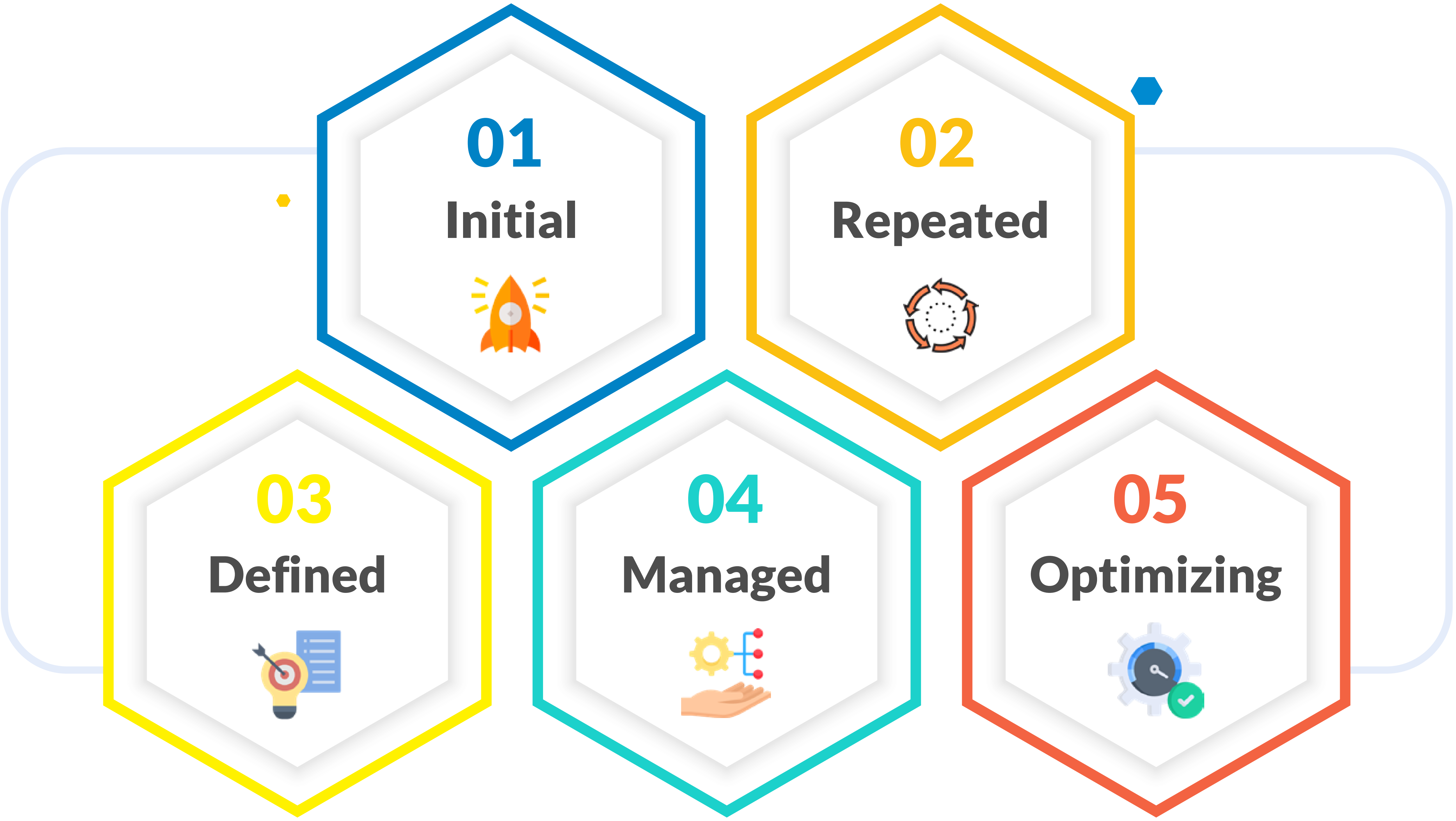


CHAPTERS

An Understanding Of Capability Maturity Model

The Custom QA maturity model for an organization can be derived from the most popular Capability Maturity Model (CMM) used worldwide. This model will eventually optimize any process during any stage of product development.

The five maturity levels of CMM are:



01 Initial

02 Repeated

03 Defined

04 Managed

05 Optimizing

By following the CMM model, an inconsistent process can be converted into a structured, managed, and optimized process. The CMM model will help analyze and improve existing QA processes to ensure compatibility across various development projects from a QA perspective.

The initial level of maturity is inherently chaotic. Testing requirements, user journey requirements, end-user experience requirements are not clearly defined. By creating a template for each requirement and documenting the requirements, the team members will clarify the requirements.

During the repeatable level of maturity, the QA team works in full force to develop processes, methodology, and standards. The well-documented plan has repeatable steps that can be used for completing similar work.

Once the test strategies, test cases and testing methods are finalized at the repeatable level, the process must be communicated to everyone in the team at the defined level of maturity. This helps the project development team use the processes and standards in every step of project development.

The processes are well defined and communicated in the managed level of maturity. Periodic monitoring and reviewing is the core job at this level. You can run automated tests at this stage. Periodical testing of automated scripts is mandatory.

The final stage of maturity is optimizing. The processes must be consistently audited at this stage, and improvement areas must be identified. With continuous optimization strategies, your organization can perfect the QA team to complete rigorous testing within a short time to deliver the best value product.

02

Roles of QA in SDLC/ releasing an awesome product

Delivering a product that works the way the customer expects is not easy. The testing process focuses on finding bugs that limit user experience. The development team has to fix the bugs before the product release. The QA team is responsible for finding bugs at the earliest to be fixed quickly. The QA team must also ensure that the identified bugs and errors are fixed before the release. QA can be defined as a set of proactive processes that are put in place to maintain the best quality in products. The Quality Control (QC) team uses the test cases of QA to identify errors and defects.

The QA team plays an important role throughout the Software Development Life Cycle (SDLC). Right from the requirement analysis till the deployment of the product, the QA team must collaborate with other development and testing teams to deliver the end product.

Requirement Analysis

In the requirement analysis phase of software development, the QA team must study the feasibility and prepare a risk analysis report. When mitigation tactics are determined initially, it will be easier to identify loopholes later. The functional and environmental requirements must be mentioned to avoid ambiguity. Proactive QA is necessary to identify test plans incorporating risks and tools to ensure a high-quality product in the end.

Design

The design can incorporate complicated designs, but the QA team must ensure that the design features are testable. During the design phase of SDLC, QA assumes the role of a proxy end user to review the design elements. This involves verifying Data Flow Diagram (DFD), Process diagram, ER diagram and use case to ensure a seamless end-user experience.

Development

The QA will enter the implementation phase during the development stage of SDLC. The development algorithm must be verified at every step. Working alongside the development team, the QA team will develop test cases to be used in the later testing phase.

Testing

The QA team must have the test cases ready before the software goes through the testing phase of SDLC. The QA team must work actively to follow the test plan thoroughly during the testing phase. The QC team will collaborate with the QA team to execute test cases. During this phase, errors and bugs are identified and reported early. As a part of the testing process, the QA team must work together with the development team to ensure that the reported bugs and errors are fixed before the software passes the testing phase of SDLC. The QA team is responsible for carrying out different types of testing such as functional testing, integration testing, regression testing, performance testing, security testing and user acceptance testing.

Documentation

The documentation phase is the last stage of SDLC, during which the user guide and manual are created from the perspective of the end-user. When the customer has any question regarding the usability of the software product, he will refer to the guide to get his answers. The QA team works with the documentation team to ensure that the user guide and manual have all the information that the end-user will want at any time.

Challenges plaguing the QA teams

Many organizations don't give vital importance to the QA teams. Often, testing and error fixing is considered an annoyance. Those companies that don't pay attention to bug fixing will not succeed. Competitors are always lurking around to provide a better quality product at a lower price. To win a client and retain long-lasting customer relationships, the QA team must be given the importance it deserves. The QA team should also advocate the necessity of testing in each phase of SDLC because the development team's goals will be different.

Unclear environment

Setting up the environment to mimic the end-user journey takes time, and the QA team has to work with an unstable environment for a long time. This will slow the testing process because the user journey may behave differently with different environments.

Improper use of tools

While creating a software product, it is not always easy to identify and use the right tools from the beginning. Sometimes, the developers and customers may have a specific set of tools that they can work with. These tools may not be suitable for the project to be handled. In that case, the QA team has to work with what they have while advocating what is needed.

Deadline issues

In most cases, development takes priority over QA. The time required for testing is misjudged, while development consumes more time. In that case, the QA team is forced to complete testing within a short period. This will result in ignoring some of the bug fixes, resulting in bad product quality in the end.

Using documents created by others

QA teams prefer creating their test procedures and documents. However, they may need to use the documentation created by others for some projects. This could result in time delays. Sometimes, the goals of different QA teams will be different and following documentation created by another team can be annoying.

Unexpected testing issues

The QA team focuses on identifying bugs and fixing them at the earliest possible time. However, testing and bug fixing may create unexpected results. It is common for software products to throw new errors when old bugs are fixed. The QA team must make room for such unexpected testing issues and complete quality assurance before the project release.

Ever-changing requirements

During agile project development, all requirements are analyzed and documented at the first stage of SDLC. However, these requirements can change from time to time based on client needs. Even a small change to the codebase must undergo various tests to ensure steadiness with the existing code. The QA team must ensure that the product is compliant with new requirements through continuous testing and monitoring.

Inadequate communication between development and testing teams

Often, the development team is dissatisfied with the testing team when bugs and errors are discovered. The testing team must collaborate with the development team from the beginning of product development to ensure that the errors don't escalate. Developers assume that testing is the final phase that needs a list of user journeys and requirements. However, a proper communication model should be established during SDLC to ensure that testing and development go hand-in-hand.

Lack of test automation experience

Numerous modern test automation tools are available for organizations to choose from. The QA team may lack the expertise and experience in using advanced test automation tools. These automation tools greatly help in reducing the time taken for testing. However, the QA team must focus on gaining knowledge and experience on these tools to use them effectively.

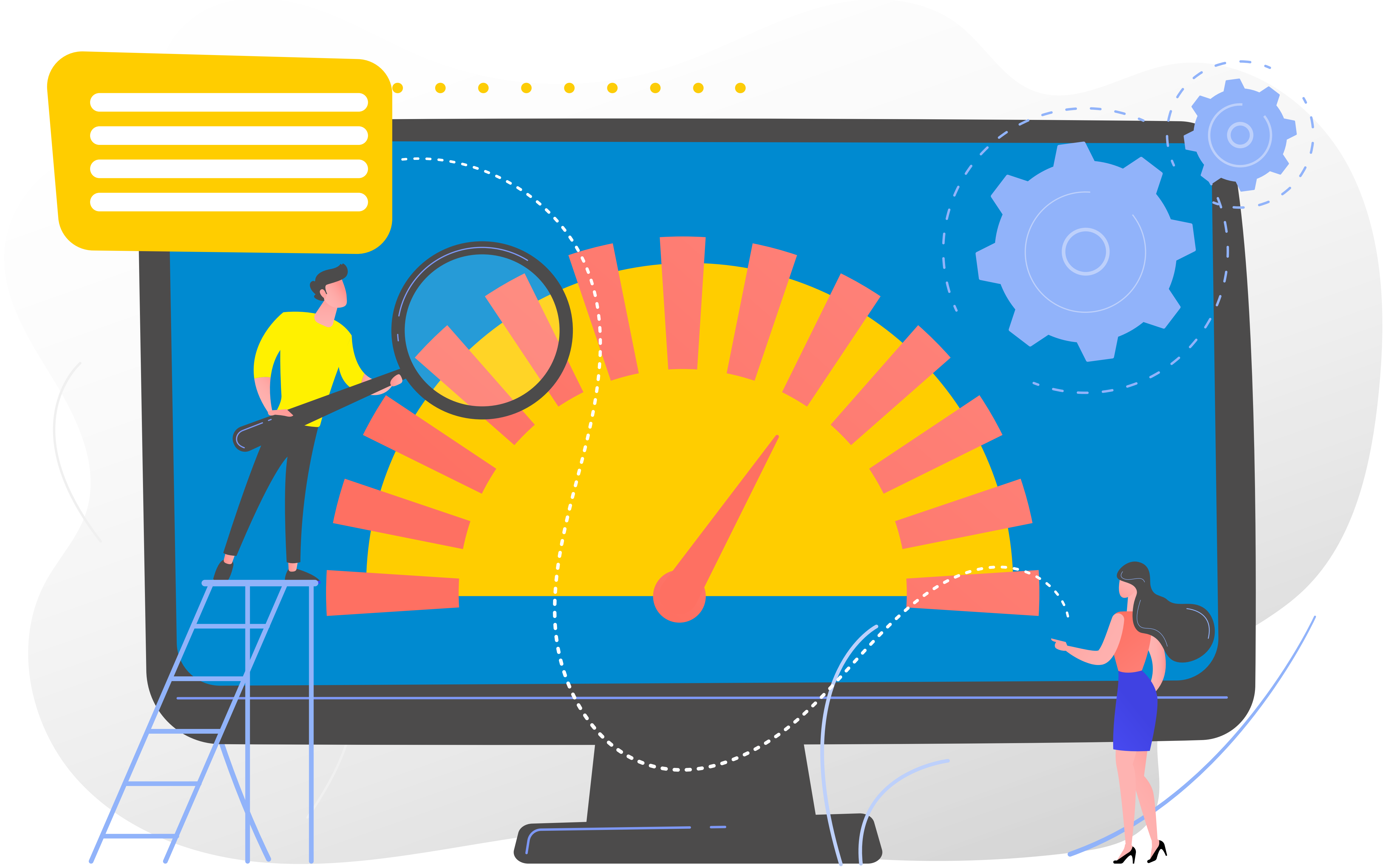
Inadequate user journey requirements

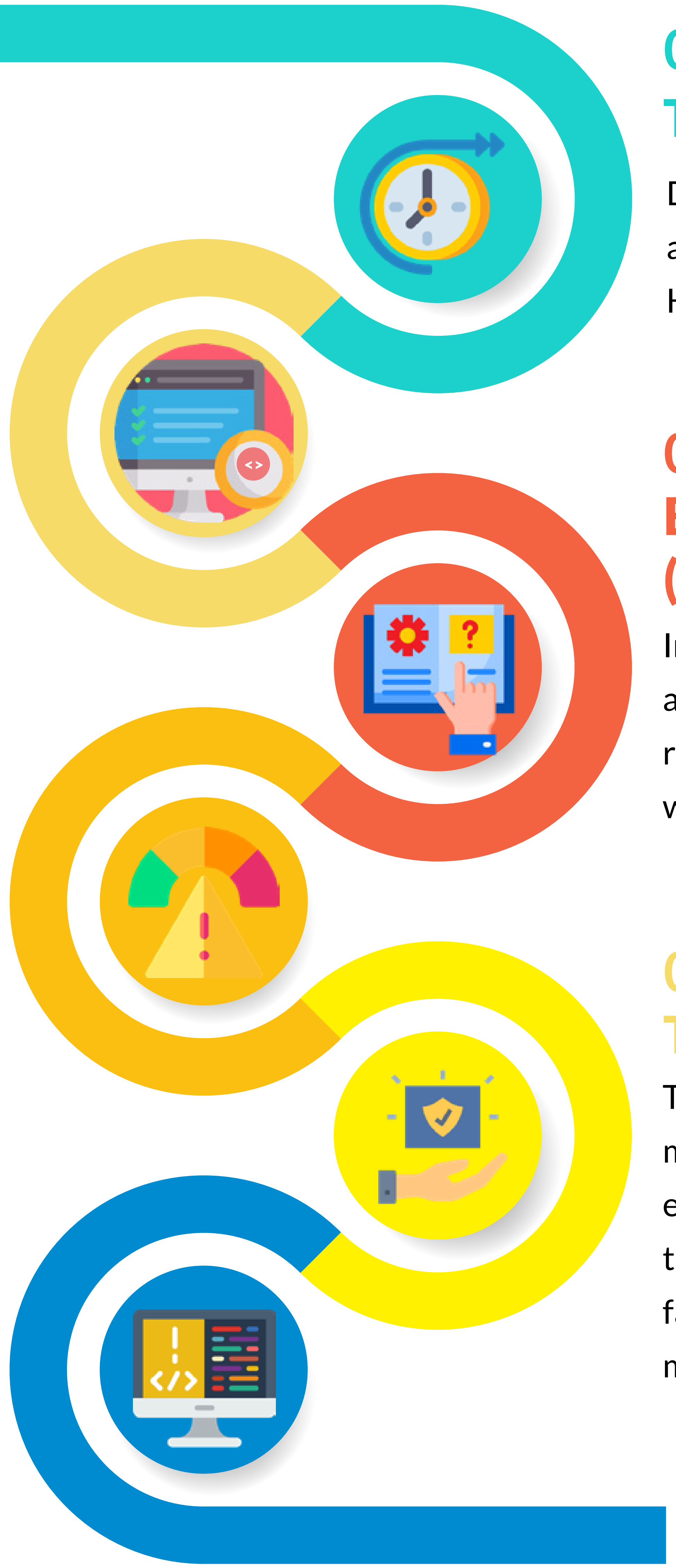
The test cases can be developed only when the user journey expectations are known. In many cases, the end-users don't know what exactly they want. This often results in ambiguous expectations, which can delay the testing phase. The QA team should have metrics to analyze and understand user journeys even if the requirement is not communicated.

Key metrics that show your QA maturity

The QA should focus on catching and fixing bugs early. However, when the QA team focuses on finding every bug in the software product, it would leave a never-ending road. The automated testing metrics should be determined based on your organization's acceptable level of quality. Your organization should test the efficiency of QA maturity metrics to optimize the QA process. If automated tools are deployed, the QA metrics must be adapted accordingly.

The role of the QA metrics is to measure the efficiency of the QA process and not the QA results. Even though each project has different QA goals, identifying QA maturity metrics will help in optimizing the QA process throughout the organization. This will improve time deliverables for different projects. The automated testing metrics should determine the effectiveness of QA testing. The metrics should also help analyze the performance of the QA team.





01 Test duration

Determining how long the testing takes is one of the important test automation metrics. Automated scripts take time to build and execute. However, if it saves valuable manual testing time, the automation is a success.

02 Test coverage

Even though test coverage is a vague metric, it is essential to determine QA maturity. Test coverage involves requirement coverage, path coverage and user journey coverage. The testing scenarios must replicate user journeys so that the end-user doesn't face unforeseen errors.

03 Equivalent manual test effort (EMTE)

In QA evaluation, EMTE is the most widely used to evaluate the efficiency of automation. Automating testing is not always suitable, especially when the requirement involves critical thinking. If the automated scripts take longer to write and much longer to maintain, that testing task is better off manually.

04 A number of mitigated risks

To achieve testing goals, the QA team must prioritize based on the risks involved. High-risk errors must be identified and rectified immediately. Sometimes, certain errors can be allowed on the end product as long as the requirements are met. Automated tests should be evaluated based on mitigated risks so that the overall risk for the product is considerable.

05 Test reliability

Test reliability is a useful QA maturity metric to make sense using failed tests. Test suite reliability evaluates whether failed tests are due to effective test suites or the presence of bugs. When testing fails because of test scenarios and environments, more time should be spent optimizing the process.

06 Time to fix

QA process doesn't end with identifying bugs and errors. The time taken to fix the errors should also be measured to determine the effectiveness of the process. The QA team often misses it because error fixing is handed over to the development team. This is why the Continuous Integration(CI)/ Continuous Delivery(CD) pipeline is always incorporated in agile models.

Key QA metrics that are under-utilized

- Test coverage
- Qualified builds
- Regression target
- Defect and flaky test trends

05

The next step:
Road to QA maturity

When the organizational QA has reached the final stage of QA maturity, your SDLC teams will be ready to adopt agile technologies. The teams can communicate clearly throughout SDLC. Software Testing Life Cycle (STLC) metrics and standards will be ready at the requirement phase. There won't be a need for fixing patches.

QA mature processes can be developed only *when the developers and testers in the team evolve into SDETs(Software development engineers in testing).* The developers and testers should write codes together. The QA plan should be clear and thorough right from the beginning. Rather than reading the requirement and design documents, the QA team must interact with the stakeholders to understand the product they are visualizing. Automating tests is the ultimate way to improve QA maturity.

Must-haves for an efficient test automation framework

- User-friendly framework
- Scope for scalability
- Integrations with test and defect management tools
- Reusability
- Supports cross-platform language and browsers
- Flexibility for a data-driven approach
- Have connectors to integrate with cloud-based solutions
- Easy integration into CI/CD pipelines

If the QA processes are automated, the testing experts can focus on the core concepts of testing instead of spending manual hours doing repetitive tasks. Identifying processes that can be automated is crucial to automation success. The automated tests must run independently with low maintenance, and anyone should run them. Shifting tests left is a step in the right direction of Agile and DevOps - which are aspects of mature QA teams. QA maturity assessment must be conducted periodically to ensure that the QA management process reaches the road to maturity. Once the assessment is complete, you can identify gaps and fine-tune the organizational QA process.

06

Final thoughts

The efficiency of the QA maturity model can only be determined with periodic assessment. The QA maturity score is useful for finding out where your organization stands to reach the final state of QA maturity.

Contact Zuci for QA maturity assessment to reduce operational costs and create awesome products. Our assessment consists of the evaluation of the following:

- QA Practices & Processes
- Test Automation Capabilities
- Toolsets
- Project methodologies & Communication

Based on the results, our consultants will help:

- Identify priority areas in your QA
- Examine current resources and create a plan for mentoring test teams
- Adopt industry-standard best practices and set up hardware and software, including test infrastructure and tools
- Establish quality metrics and quantify their success in phases
- Set up continuous testing, and feedback
- Periodic evaluation and provide realistic recommendations.

Find out how we can effectively improve your QA team's efficiency to deliver high-standard products.

Know more

ZUCI SYSTEMS

About Zuci

Zuci is a digital organization focused on the craft of assuring quality to software which we have perfected over the years. Zuci enables product teams to improve product quality, fasten release speed and enhance user experiences - with a perfect blend of test planning, engineering perfection, and customer-centricity..

Learn more at **zucisystems.com**