

WHITEPAPER

The Role of SDET's in Engineering Quality for Mobile Applications

TOP 6 TESTING ZONES FOR MAINTAINING CONTINUOUS QUALITY

Authored by
Philip Lazaro,
Principal Software Engineer,
Maveric systems.

Introduction

Every customer journey is unique.

But what's common is the need for speed. And the enormous risk associated with getting it wrong. In this volatile business ecosystem to re-engineer business models, and seize the next technology shift becomes key.

Towards this imperative, businesses today realize the edge digital transformation offers to

fuel their growth rhythms and carve out its competitive edge – be it through faster-time-to-market, or by digitizing their strategic approaches. And finally, the one unchanging truth in these disruptive times can be judged through the question: “Did your digital transformation create a seamless end user experience. If yes, how much of an enhancement did it bring?”

Achieving seamless customer experience through digital transformation, calls for specific attention to the vitality of quality engineering processes in the software development life cycle.



The Need for SDET's

Along with the industry insistence on practical solutions, more and more CXO's are being asked to upgrade their levels of service quality and delivery speeds. This market dynamic understandably raises most company's appetite for automated testing solutions.

Automated testing solutions, however, are not a magic bullet by themselves. They introduce a degree of uncertainty and potential risks: the native testing team may lack adequate knowledge to adapt it to their specific environment or be agnostic about

its problem shooting or later in the game, compatibility issues may surface. Additionally, implementing automated testing solutions does not preclude the necessary skills needed to create robust test designs or writing high performance testing scripts.

These then become the precise needs for SDET's to function in the complex world of Quality engineering.

Who are SDET's and Why they are so special?

SDET's are ideal customer advocates who understand and influence product design by aligning end user expectations. Not an entirely new concept, (Microsoft introduced it a decade earlier) the need for SDET's has risen considerably in the recent years. Simply put, Software Development Engineer in Test (SDET) is the 'jack-of-all-trades' IT professional who takes part in the entire software development process – designing, development and testing. Unlike a tester, SDET's know the entire system and focusses his development and testing knowledge on a system's testability, robustness and performance.

SDET's scope of delivery as compared to functional or automation testers is geographically vast and immensely deep. They are masters in the below streams

- ▶ Build robust and high quality test automation solutions
- ▶ Develop high performance code
- ▶ Assess product scalability, reliability and performance
- ▶ Participate in design and architectural decision making
- ▶ Perform high class debugging, and customizations
- ▶ Deploy environment /test automation framework

Let us next consider the top skills SDET's bring for succeeding in today's application development scenario:

Crystal clarity on the project scope - What are the critical testing aspects under purview – Is it stress, performance, load, security or vulnerability or a specific combination?

Thinking beyond Testing and Development

- How to objectively identify the risks and defects on a holistic application level from an end user's perspective?

Embracing business agility through DevOps and Agile approaches

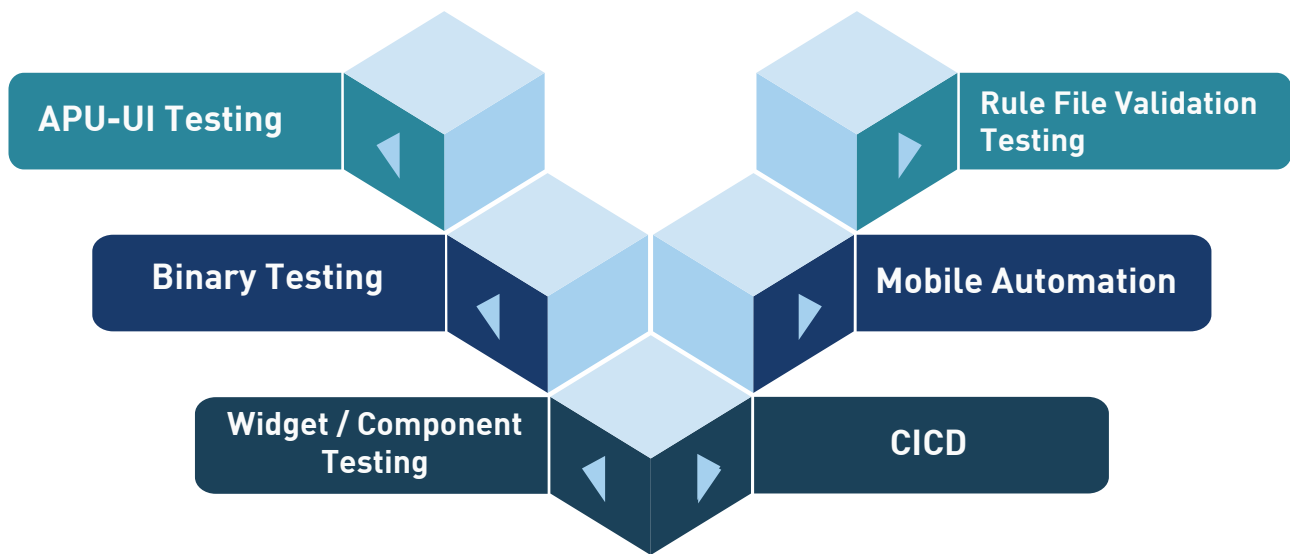
- SDET's will need to pre-empt next shifts in consumer preferences – thereby creating a culture of flexibility through collaboration

Building automation into the strategic objectives

- Automation is indispensable today, but SDET's need to hard wire automation at the strategic level – and not just be caught up in daily incremental execution tasks



Top 5 testing zones for mobile application testing with SDET's



1. API-UI Testing:

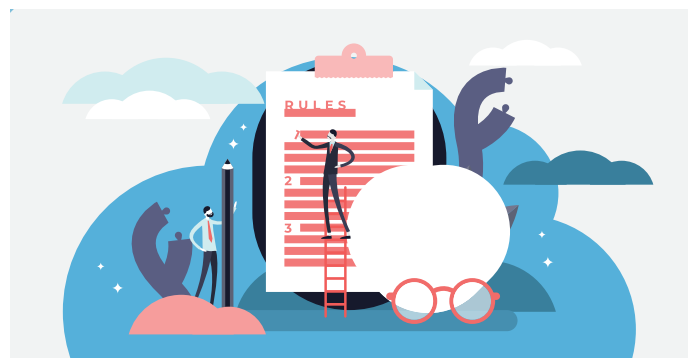
API Testing is a type of software testing that involves testing of application programming interfaces (API's) directly. Additionally, as a critical part of integration testing, API testing determines if they meet expectations for functionality, reliability, performance, and security. This testing revolves around the list of API's needed for that particular functionality from specification document provided by the business unit.

This phase of testing includes functionality testing in three different aspects – usability, errors, look and feel. The aim of this API-UI testing is to make sure the end user gets the functionality intended to be delivered with minimal user inputs, faster response times and higher security on traversing customer-client data. Thus, facilitating a superior customer experience.



2. Rule file validation Testing

Rule file validation is a process to compare JSON (Java Script Object Notation) formatted files, to look for missing tags or newly added tags in rule file using a JSON comparator. The functionalities in applications are turned on or off based on the flags set in rule files. Ultimately, it is the rule file that ensures the presence of functionality in the application when delivered to the customer.



3. Binary Testing

Post development, Binaries are generated for two variants Android & IOS. The next stage follows the Binary testing phase where focus is given to various touch points on each screen, this involves, Field valida-

tion, Element touch points validations (Buttons, Radio Buttons, Links etc.) & Transition validation to ensure the traversals to and forth from a screen is as expected and smooth.

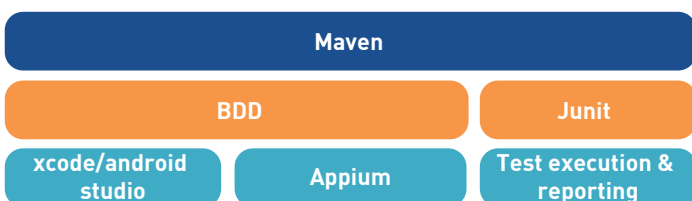


4. Mobile automation

As the name suggests, mobile automation is done on mobile devices. Whether it is on top of a WAP site or an app, automation can be introduced for increased, efficient testing productivity and reducing testing cycle times.

Test cases are first automated for android mobile application, iOS mobile applications and web applications. Thereafter, automation framework is used for improving execution control, parallel execution and report generation purposes. The automation framework involves two sub sets as in Behaviour Driven Development (BDD) used as documentation framework along with Junit / TestNg as an testing framework riding on appium server. BDD involves Gherkin language (code is laymen friendly) which revolves around keywords such as Given, When, Then, And.

Additionally, regression suite is kept ready to be executed during the regression phase.



5. Widget / Component Testing

Widget testing is focused on mobile apps built as individual widgets / components. The focus is primarily given to validate the individual widget / components for all its possible variations, combinations in the same or at different page level occurrences.

This testing also paves way for unit testing the individual widgets / component at local or WAP sites.



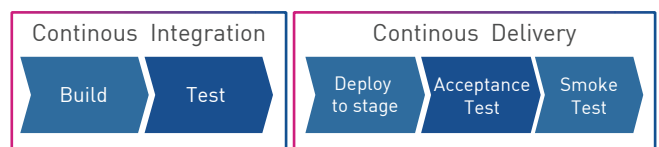
6. Continuous Integration and Continuous Delivery Testing (CI/CD)

Continuous Integration is about merging all codes from all developers into one central repo branch several times a day to avoid any future code conflicts. This process ensures that the main repo branch is maintained as the most current form of the source code. The technical goal of continuous integration is to establish a consistent and automated way to build, package, and test applications. With consistency in the integration process, teams are more likely to commit code changes more frequently, leading to a better collaboration and software quality. Continuous delivery picks up where continuous integration ends. With Continuous Deployment, every change made is automatically deployed. User acceptance testing. It automates the delivery of applications to selected infrastructure environments. As most teams work with multiple environments other than the production, such as development and testing environments, continuous delivery processes ensure that there is an automated way to push code changes to them.

Continuous Integration also takes place in the testing phase where all automation codes are merged to a single master or main branch thus facilitating to build test

points for the development code to be tested against the test code.

CI/CD automation testing then performs validation on the builds generated and carries out to be either deployed further to user acceptance or rolls back based on the gate value set in the automation testing (greater or equal to 75% pass rate).



CONCLUSION

SDET's are a single point of source for bringing inline automation for accelerating customer transformation journeys. The early quality engineering proficiency they facilitate is a key instrument for maintaining continuous quality. Several use cases beyond mobile applications such as websites, wearables and more, can be quality engineered thoroughly using SDET's. Therefore, launching any future digital transformation initiatives can be a breeze with continuous quality engineered with speed at an unprecedented scale.



ABOUT MAVERIC'S QE SERVICES

Maveric Systems are a world-class leader in QE services for the banking domain. For over two decades now, Maveric has partnered on 60+ major banking transformations. Our core promise of continuous quality is brought alive via Digital-age QE solutions, and cutting-edge test approach that is engineered for speed and geared to bring QE at scale. We foster a culture of "95% QE automation", through our tools, accelerators and matured frameworks, which are engineered with cognitive computing and predictive analytics for intelligent automation. To learn more about our QE business,

Please visit <https://maveric-systems.com/services/quality-engineering/quality-engineering-services>



www.maveric-systems.com

Email us at
marketing@maveric-systems.com

Maveric Systems Limited

INDIA | SINGAPORE | SAUDI ARABIA | UAE | UK | USA