

Research Article

## Approach for Mass Notification Device Testing

Brijesh Patil\*, Shiva Vyas†, Anand Nadgire† and Vaibhav Kulkarni†

†Mechanical Engineering Department, Savitribai Phule University Pune, MIT College of Engineering, Kothrud, Pune – 411038, India

Accepted 02 March 2016, Available online 15 March 2016, Special Issue-4 (March 2016)

### Abstract

*Diversity in devices, lack of systematic device testing strategy has always led to open questions in the customer's mind saying "Is the device tested correctly?" Testing of any device integrated with System Under Test (SUT) is certainly different and more complex than testing any traditional software application, as it is a comprehensive testing which includes device and the infrastructure of SUT. A systematic device testing approach plays a very important role in getting the device and its supporting traditional software application released in the market with in time ensuring high quality. This white paper focuses on the lessons learnt and experiences shared along with some guiding principles that can be considered at the time of planning a systematic testing strategy. This strategy can be followed for any Mass Notification application that includes manual as well as automation testing approach in order to cover functional and nonfunctional testing. This paper will also revisit some of the most common pitfalls and pain points that often are snowed under during the test execution phase for any Mass Notification device.*

**Keywords:** Mass Notification Device Testing, Device Testing, Mass Notification, Systematic device testing approach.

### 1. Introduction

As the number and variety of Mass Notification Systems (MNS) along with the diversity in Mass Notification Devices has grown exponentially over the last few years, organizations need to ensure that every Mass Notification System meets a high quality bar in order to prevent revenue loss, productivity lost, and damage to the brand reputation. Comprehensive Mass Notification Systems' devices testing strategy is essential for getting your application to market on time and within budget.

Some of the key elements which can be considered for effectively testing of Mass Notification devices which are integrated with SUT are as follows.

- 1) Categorization of device parameters - Define a systematic test strategy for each parameter of the device under test.
- 2) Types of testing - Consider different types of testing required (functional, performance, security, and compliance).
- 3) Target device selection - Create an optimal mix of simulator testing and physical device to maximize the test coverage
- 4) Test Automation - To reduce the cost of regression testing select an effective automation testing tool and maximize the use of automation.

The challenge of Mass Notification Devices testing can be effectively addressed by a test strategy that combines these elements with traditional best practices and processes for testing.

### 2. Problem Definition

It is generally observed that there is a lack of systematic approach while planning a test strategy for any device/s integrated with SUT which leads to many questions:

- 1) Is the device tested properly?
- 2) Is the software application (SUT) properly integrated with the device under test?
- 3) Is the response time of the device as expected or can we improve on it? And so on.

### 3. Methodology Strategy for Mass Notification Device Testing

#### 3.1. Categorization of device parameters:

The very first thing that can be considered for the Mass Notification Device under test is to categorize its various parameters and then define a systematic test strategy for each parameter. An example of a Mass Notification device "Adaptive" (Display device) is given in Table 1.

\*Corresponding author: Brijesh Patil

**Table 1** Device parameters

Type of Device	Protocol	Firmware Supported	Response Supported	Type of text supported	Type of text color supported	Scalability
Adaptive (Display device)	TCP/IP	1.1	5	Times New Roman	Red	3,100,200

Here one of the biggest challenges of a MNS device testing is its ability to scale. Scalability testing helps to determine whether your SUT scales with the workload growth as the MNS.

Device grows in numbers and complexity. Hence, scalability testing forms an essential part of the entire development and testing process of MNS.

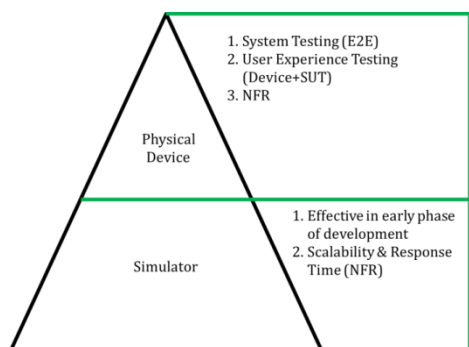
**Table 2** Device factors

Device Factor	Type of testing
Protocol	1. Network Security testing 2. Firewall Testing
Firmware	1. Firm ware compatibility testing 2. Performance testing w.r.t Firmware
Response Time	Functional and Non Functional testing (Performance Testing)
Type of Text Support	Functional /compatible testing- to cover various fonts of text
Type of text color supported	Functional /compatible testing- to cover various supported text color
Scalability	NFR- Nonfunctional testing

**3.2. Test Approach for Physical device vs Simulators.**

Use of simulators can be very effective in the early phase of development life cycle. Better test coverage can be achieved with respect to scalability and response time whereas testing on physical devices is very essential to understand the real time behavior of the device with the integrated software (SUT).

Below figure explains further how choice between Physical device vs Simulators can help in planning a better test strategy



**Fig.1** Choice between Physical device vs Simulators

**4. Challenges with Real device and simulators**

**4.1. Challenges with real device testing:**

Though testing with physical devices is recommended and generally followed, however it is observed that

more challenges are encountered in this area. General issues faced while testing with real devices are as below:

**Table 3** Challenges and Recommendations for real device testing

Sr. No	Challenges	Recommendation
1	Device Configuration complexity	Use of Automation scripts for configuration of device; However at the initial phase reference document containing all the configuration information can also be very helpful.
2	Network Connectivity issues	General practice is to have an independent test network where the device and SUT should be configured.
3	Third party software dependency for configuring the device.	Probably the solution would be to create automation scripts for configuring the third party software, however at the initial phase reference documents containing all the information.

**4.2. Challenges with simulator testing**

As discussed, earlier testing with simulators should be done in the early phase of the development life cycle, as simulators can be used to test the basic application functionality; General issues faced while testing with simulators are as below:

**Table 4** Challenges and Recommendations while testing with simulators

Sr. No	Challenges	Recommendation
1	SUT Knowledge	Gain Knowledge by conducting sessions with domain experts
2	Deep Knowledge of device and its integration with SUT	Conduct Domain knowledge sessions with System/Software Architect
3	Features in simulators Vs Features in actual devices	Clarity in understanding expectations from simulators at the same time it should be clearly communicated to the test team
4	Communication between simulator and SUT	Granular level understanding of communication between SUT and actual device e.g. Protocol/Fix parameters /Inputs and outputs format
5	Simulator framework	Well defined logging framework should be developed, which helps in differentiating the issue coming out from simulators and SUT

### 5. Real Time Test data

To understand the MNS device test approach more practically we have used some real time test data, which was collected during the time of test from real MNS device and simulators.

In the table for real time Mass Notification Device, we had focused on the response time of the single device, where as in the test coverage using simulators the focus was on scalability and on response time.

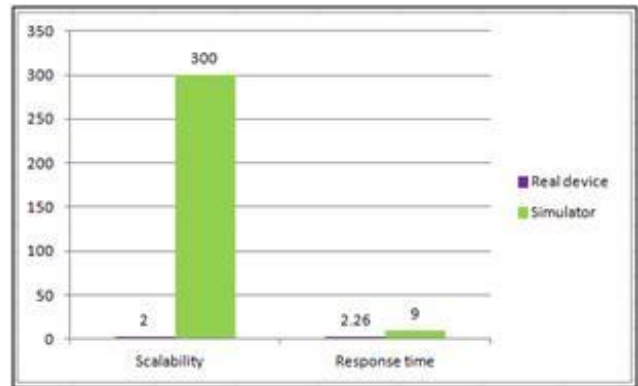
**Table 5** Results from real time Mass Notification device (Adaptive Display Device)

Device Type	Total	Failed	Delivery Time(ms)	Delivery Time (Sec)
Adaptive_Display_Driver	2	0	3300	3.3
Adaptive_Display_Driver	2	0	1847	1.847
Adaptive_Display_Driver	2	0	1793	1.793

**Table 6** Results from Mass Notification device simulator (Adaptive Display Device)

Device Type	Total	Failed	Delivery Time(ms)	Delivery Time (Sec)
Adaptive_Display_Driver	300	0	7153	7.153
Adaptive_Display_Driver	300	0	8956	8.956
Adaptive_Display_Driver	300	0	11000	11.00

The successes of over all industries demonstrate that breakthroughs in quality begin with a change in attitude. By increasing expectations for Mass notification device testing and providing testers with an efficient test strategy, is one of the way which will significantly increase quality, improve customer satisfaction, and gain a competitive edge.



### References

Rex Black. Managing the Testing Process, Wiley Publishing, 2002.  
 Cem Kaner, James Bach, and Bret Pettichord,(2001). Lessons Learned in Software Testing. Wiley Publishing,. ISBN-13: 978-0471081128. Software Engineering Technical Committee of the IEEE Computer Society. "IEEE Std 829-1998: IEEE Standard for Software Test Documentation." IEEE, 1998.