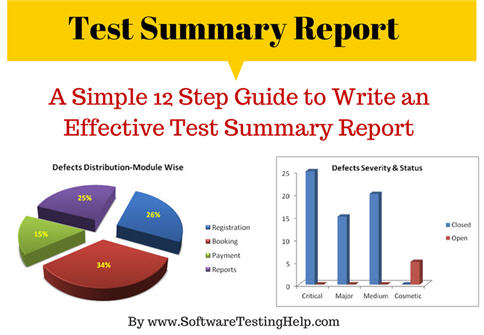
**A Simple 12 Steps Guide to Write an Effective Test Summary Report [with a Sample Report for Download]**

Several documents and reports are being prepared as part of Testing. Some are [Test Strategy doc](http://www.softwaretestinghelp.com/writing-test-strategy-document-template/" \o "Test strategy document), [Test Plan doc](http://www.softwaretestinghelp.com/test-plan-sample-softwaretesting-and-quality-assurance-templates/" \o "Test plan document), [Risk management Plan](http://www.softwaretestinghelp.com/risk-management-during-test-planning-risk-based-testing/" \o "risk management in software testing), Configuration management plan etc. Among these Test Summary Report is one such report which is prepared after the Testing is completed.

I have tried to explain the purpose of ‘*Test Summary Report*’ and provided a **sample Test Summary Report template along with an actual report for download.**

**What is a Test Summary Report?**

As we know, Software Testing is an important phase in SDLC and also it serves as the “Quality Gate” for the application to pass through and certified as “Can Go Live” by the Testing Team.

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/06/test-reporting.jpg)

Test Summary Report is an important deliverable which is prepared at the end of a Testing project, or rather after Testing is completed. The prime objective of this document is to explain various details and activities about the Testing performed for the Project, to the respective stakeholders like Senior Management, Client etc.

As part of [Daily status reports](http://www.softwaretestinghelp.com/how-to-write-software-testing-weekly-status-report/" \o "daily/weekly status report), daily testing results will be shared with involved stakeholders every day. But Test Summary Report provides a consolidated report on the Testing performed so far for the project.

**Recommended reading => [How to Report Test Execution Smartly (Status Report Template download)](http://www.softwaretestinghelp.com/test-execution-report/" \o "Test execution report)**

Assume that if the Client who sits in a remote location need to understand the results and status about a Testing project which was performed for a period of, say for example – four months, Test Summary Report will solve the purpose.

This is also an artifact required to be prepared as part of [CMMI process](http://www.softwaretestinghelp.com/cmmi-appraisals-testers-journey-as-an-appraisal-team-member-atm/" \o "CMMI process).

**What Test Summary Report contains?**

A typical **Test Report template** will contain the below information, however based on each Company’s format & practice, the contents may vary. I have also provided real examples for better understanding.

***At the end of this article you can download a test summary report sample.***

***12 Steps Guide to writing an effective test summary report:***

**Step #1: Purpose of the document**

*<Short description about the objective of preparing the document>*

***Example:***This document explains the various activities performed as part of Testing of ‘ABCD transport system’ application.

***Step #2:*Application Overview**

*<Brief description about the application tested>*

***Example:***‘ABCD transport system’ is a web based Bus ticket booking application. Tickets for various buses can be booked using the online facilities. Real time passenger information is received from a ‘Central repository system’, which will be referred before booking is confirmed. There are several modules like Registration, Booking, Payment and Reports which are integrated to fulfill the purpose.

***Step #3:*Testing Scope**

1. In Scope
2. Out of Scope
3. Items not tested

*<This section explains about the functions/modules in scope & out of scope for testing; Any items which are not tested due to any constraints/dependencies/restrictions>*

***Example:****A functionality verification which needs connectivity to a third party application cannot be tested, as the connectivity could not be established due to some technical limitations. This section should be clearly documented, else it will be assumed that Testing covered all areas of the application.*

**a) In Scope**  
Functional Testing for the following modules are in Scope of Testing

* Registration
* Booking
* Payment

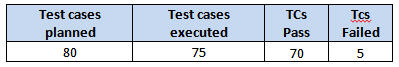
**b) Out of Scope**  
Performance Testing was not done for this application.

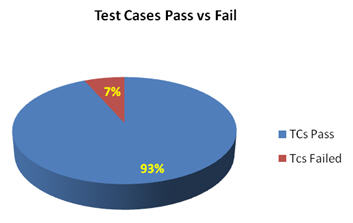
**c) Items not tested**  
Verification of connectivity with the third party system ‘Central repository system’ was not tested, as the connectivity could not be established due to some technical limitations. This can be verified during UAT (User Acceptance Testing) where the connectivity is available or can be established.

**Step #4: Metrics**

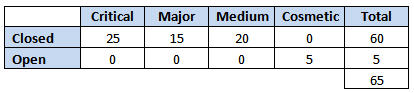
*<Metrics will help to understand the test execution results, status of test cases & defects etc. Required Metrics can be added as necessary. Example: Defect Summary-Severity wise; Defect Distribution-Function/Module wise; Defect Ageing etc.. Charts/Graphs can be attached for better visual representation>*

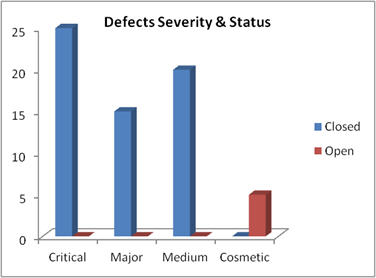
**a) No. of test cases planned vs executed**  
**b) No. of test cases passed/failed**

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/06/test-summary-report-1.jpg)

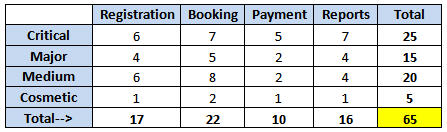
[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/06/test-summary-report-2.jpg)

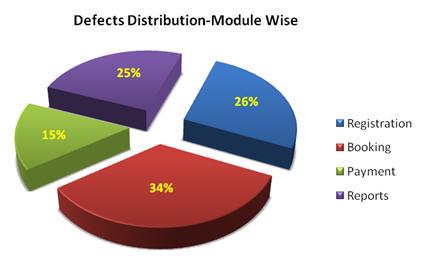
**c) No of defects identified and their Status & Severity**

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/06/test-summary-report-3.jpg)

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/06/test-summary-report-4.jpg)

**d) Defects distribution – module wise**

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/06/test-summary-report-5.jpg)

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/06/test-summary-report-6.jpg)

**Step #5: Types of testing performed**

1. Smoke Testing
2. System Integration Testing
3. [and Regression Testing](http://www.softwaretestinghelp.com/regression-testing-tools-and-methods/" \o "What is regression testing)

*<Describe the various types of Testing performed for the Project. This will make sure the application is being tested properly through testing types agreed as per Test Strategy.*

***Note:****If several rounds of Testing were done, the details can also be included here.>*

***Example:***  
**a) Smoke Testing**  
This testing was done whenever a Build is received *(deployed into Test environment)* for Testing to make sure the major functionality are working fine, Build can be accepted and Testing can start.

**b) System Integration Testing**

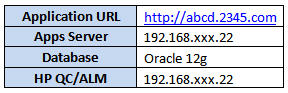
* This is the Testing performed on the Application under test, to verify the entire application works as per the requirements.
* Critical Business scenarios were tested to make sure important functionality in the application works as intended without any errors.

**c) Regression Testing**

* Regression testing was performed each time a new build is deployed for testing which contains defect fixes and new enhancements, if any.
* Regression Testing is being done on the entire application and not just the new functionality and Defect fixes.
* This testing ensures that existing functionality works fine after defect fix and new enhancements are added to the existing application.
* Test cases for new functionality are added to the existing test cases and executed.

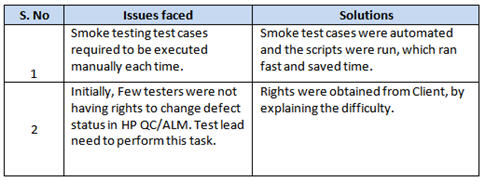
**Step #6: [Test Environment & Tools](http://www.softwaretestinghelp.com/test-bed-test-environment-management-best-practices/" \o "Test environment)**

*<Provide details on Test Environment in which the Testing is carried out. Server, Database, Application URL etc. If any Tools were used like Quality Center (now HP ALM) for logging defects>*

***Example:***  
[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/06/test-summary-report-7.jpg)

**Step #7: Lessons Learned**

*<This section is used to describe the critical issues faced and their solutions (how they were solved during the Testing). Lessons learnt will help to make proactive decisions during the next Testing engagement, by avoiding these mistakes or finding a suitable workaround>*

***Example:***  
[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/06/test-summary-report-8.jpg)

**Step #8: Recommendations**

*<Any workaround or suggestions can be mentioned here>*

***Example:***

* *Admin control for defect management tool can be given to Offshore Test manager for providing access to Testing team.*
* *Each time the onsite Admin need not be contacted for requests whenever they arise, thereby saving time due to the geographical time zone difference.*

**Step #9: Best Practices**

*<There will be lot of activities done by the Testing team during the project. Some of them could have saved time, some proved to be a good & efficient way to work, etc. These can be documented as a ‘Value Add’ to show case to the Stakeholders>*

***Example:***

* *A repetitive task done manually every time was time consuming. This task was automated by creating scripts and run each time, which saved time and resources.*
* Smoke test cases were automated and the scripts were run, which ran fast and saved time.
* Automation scripts were prepared to create new customers, where lot of records need to be created for Testing.
* Business critical scenarios are separately tested on the entire application which are vital to certify they works fine.

**Step #10: Exit Criteria**

*<Exit Criteria is defined as a Completion of Testing by fulfilling certain conditions like  
(i) All planned test cases are executed;*  
*(iI) All Critical defects are Closed etc.>*  
***Example:***  
**a)** All test cases should be executed – **Yes**  
**b)** All defects in Critical, Major, Medium severity should be verified and closed – **Yes**.  
**c)**Any open defects in Trivial severity – **Action plan prepared with expected dates of closure.**

*No Severity1 defects should be ‘OPEN’; Only 2 Severity2 defects should be ‘OPEN’; Only 4 Severity3 defects should be ‘OPEN’. Note: This may vary from project to project. Plan of Action for the Open defects should be clearly mentioned with details on when & how they will be addressed and closed.>*

**Step #11: Conclusion/Sign Off**

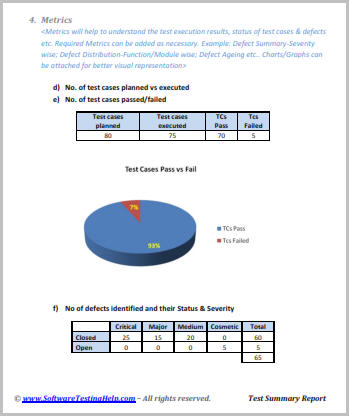
*<This section will mention whether the Testing team agrees and gives a Green signal for the application to ‘Go Live’ or not, after the Exit Criteria was met. If the application does not meet the Exit Criteria, then it can be mentioned as – “The application is not suggested to ‘Go Live’. It will be left with the decision of Senior Management and Client and other Stakeholders involved to take the call on whether the application can ‘Go Live’ or not.>*

***Example:***As the Exit criteria was met and satisfied as mentioned in Section 10, this application is suggested to ‘Go Live’ by the Testing team. **Appropriate User/Business acceptance testing** should be performed before ‘Go Live’.

**Step #12: Definitions, Acronyms, and Abbreviations**

*<This section mentions the meanings of Abbreviated terms used in this document and any other new definitions>*

**=> Download Sample Test Summary Report:**  
***[Click here to download](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/06/Sample-Test-Summary-Report-by-SoftwareTestingHelp.pdf" \o "sample test summary report example" \t "_blank)****a sample test report template with an example.*

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/06/test-report-template.jpg)

**Few points to note while preparing the Test Summary Report:**

* As part of Test Execution, collect all required information on the Testing performed. This will help to prepare a sound Test summary report.
* Lessons learned can be explained in detail, which will convey the Responsibility which was taken to solve these issues. Also this will be a reference for upcoming projects to avoid these.
* Similarly, mentioning the Best Practices will portray the efforts taken by the team apart from regular testing, which will also be treated as a “Value Addition”.
* Mentioning the Metrics in graphics form (Charts, Graphs) will be a good way to visually represent the status & data.
* Remember, Test summary report shall mention and explain the activities performed as part of the Testing, to the recipients to understand better.
* Few more appropriate sections can be added if required.

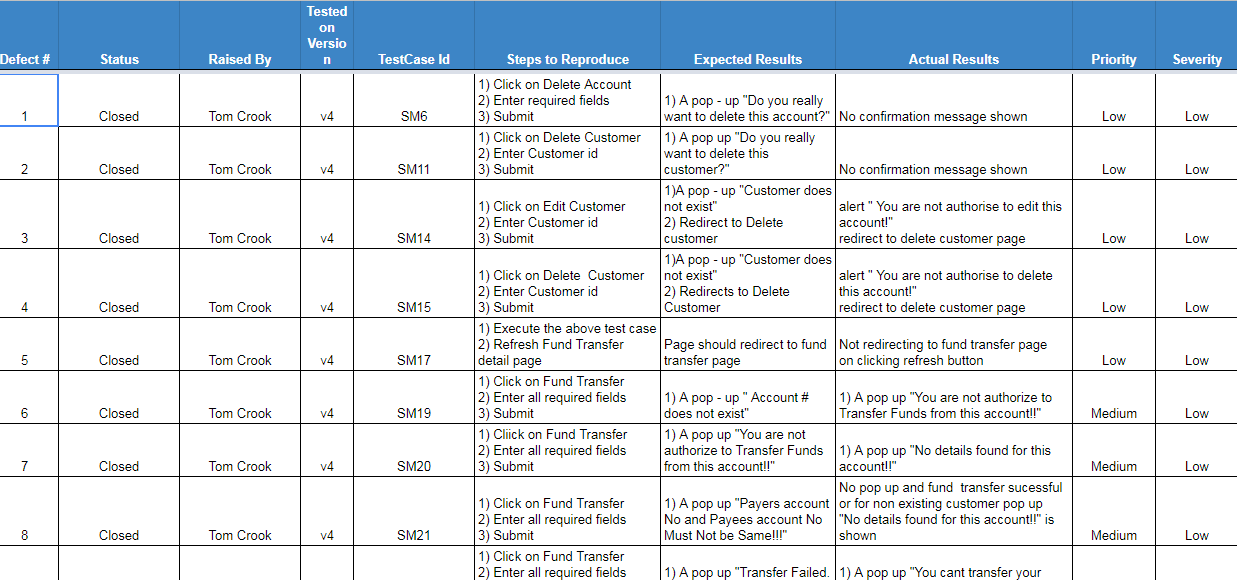
**Conclusion:**

Test summary report is an important deliverable and focus should be to prepare an effective document, as this artifact will be shared with various stake holders like senior management, client etc.

After performing an exhaustive testing, publishing the test results, metrics, best practices, lessons learnt, conclusions on ‘Go Live’ etc. are extremely important to produce that as an evidence for the Testing performed and the Testing conclusion.

*We have also made available the test report sample for download. It is a perfect example of how to prepare an effective test summary report!*

***About the author:*** This is a guest post by Baskar Pillai. He is having around 14 years of experience in Test management and end to end software testing. CSTE certified Testing professional, trainer, worked in IT majors like Cognizant, HCL, Capgemini and currently working as Test Manager for a large MNC.





# Mobile vs. Desktop: 10 Key Differences

From desktops to smartphones to an expanding galaxy of different connected screens and devices, we can safely say that access to the world’s information on-demand is part of our lives now, and for the foreseeable future. We’re at the end of the beginning and companies must evolve. People now expect connectivity wherever they are and on whatever device they choose — this is the new norm. It’s time for businesses to catch up and understand their connected customer.

As mobile web browsing becomes more common, it's increasingly important to design websites for mobile use as well as for desktop browsing. This can mean anything from tweaking your regular site so it looks good on Android and iOS devices, to CSS media queries, responsive layouts, and even completely separate websites for mobile users.

The question is: How do you make a website work well on a mobile browser? It helps to start out by thinking about how the mobile browsing experience differs from desktop browsing. The best way to understand this is to understand 10 important factors that make the mobile web different.  Once you understand the differences, you can use those factors to influence your mobile site design decisions.

**1. Smaller screens**

Let's start with one of the most obvious differences: Mobile browser displays are smaller than their desktop counterparts. "Smaller" here means two things:

* Physically smaller: Typically around 3-5 diagonal inches for a phone, although some larger phones can be up to 6, and 9-12 diagonal inches for a tablet. This compares to a typical notebook screen size of 13-17 inches, and a desktop screen size of 20-30 inches.
* Fewer pixels: Most mobile displays currently have fewer pixels than desktop displays. An iPhone 6s retina display is 1334 x 750 pixels, whereas the smallest-screen MacBook has a 2304 x 1440 display.

## [You don't have to guess!](https://www.paradoxlabs.com/blog/screen-resolutions/)

We've compiled a massive list of screen resolutions to help you plan content and images...

[Check it out](https://www.paradoxlabs.com/blog/screen-resolutions/)

[](https://www.paradoxlabs.com/blog/screen-resolutions/)

A smaller display means that the user can see a lot less information at once. Most modern mobile browsers compensate for this by allowing the user to zoom in and out easily, as well as adapting font sizes to make text more readable. Typically, though, the user will need to zoom in if they want to read the text in your page.

A smaller display also pushes more of your page content "below the fold," requiring users to scroll through your page content more often.

So, for your site to work well on a mobile browser, it needs to: present important information near the top of the page, use an easy-to-read font, and not overwhelm the user with too much content on the page. In addition, the page layout needs to be usable and it needs to look good in a small browser window too. This usually implies a simpler page layout than you might see on a regular website.

**2. Slower processors**

Although they are catching up, mobile devices generally have much less processing power than desktop computers. This is for various reasons, including cost and battery life. Therefore, mobile browsers take longer to render pages, and JavaScript-intensive pages can run very slowly.

Consider creating a simpler page layout for mobile browsers that uses less markup and CSS. If your site uses a lot of JavaScript for things like slideshows, interactive forms, and the like, you might want to optimize or minimize your JavaScript so it runs smoothly on mobile browsers.

**3. Less bandwidth**

While cellular network speeds are improving all the time, a typical 4G mobile device gets a 5-12 Mbps download rate, compared to an average of over 50 Mbps for broadband internet users.

What's more, most mobile plans aggressively limit the amount of data that can be downloaded each month. If your users find that your site sucks up a large chunk of their download allowance, they won’t be coming back!

It's always a good idea to make your site as bandwidth-efficient as possible, since every second that your users have to wait for your pages to load will increase their frustration. This is even more important when designing for the mobile web. According to [Google](https://www.doubleclickbygoogle.com/articles/mobile-speed-matters/), more than half of users will abandon a page if it takes longer than 3 seconds to load.

If your site contains large graphics or embedded videos that take up a lot of bandwidth, then you definitely want to look at creating a responsive, mobile-friendly site with smaller images and lower bit-rate video (or no video at all). Also, if you're providing downloadable PDFs or other content for your visitors, make sure the content isn't too big to download comfortably on a mobile device.

**4. Touch input**

One important aspect of many mobile devices is touch-based input. Rather than using a mouse, the vast majority of mobile users work with their devices using their fingers (some desktops do as well). This has several implications for mobile site designers, including:

* No hover events:  
  There is no mouse pointer, so there is no concept of "hovering" over a page element. Navigation menus and other controls that rely on the CSS hover pseudo-class or JavaScript mouse-over/mouse-out events won't work well on touch devices. Some mobile browsers use various tricks to compensate for this, such as firing a hover event when the user taps the element once, and a click event if they tap again. However, it's best not to rely on hover events for your mobile site's functionality.
* Less precision:  
  Clicking a 12-pixel-high text link with a mouse is no problem. Tapping the same link with big sausage fingers is a different story! Users can compensate for this by zooming in, but it's still awkward. This is one of the best arguments for creating a separate site just for mobile users, since you can then replace those fiddly text links with nice, large, touch-friendly buttons and other controls.
* Gestures:  
  Most modern touch devices allow the user to perform gestures using one or more fingers, such as swiping, pinching, and so on. You can use gestures to enhance the experience for your mobile users. For example, you can let your users swipe left and right to move between images in a gallery. Many JavaScript frameworks such as jQuery Mobile can generate events for various touch gestures, making it easier to add gesture support to your mobile sites.

**5. Tricky keyboards**

Unless you happen to be using an add-on, full-size keyboard with your mobile device, the chances are that you're typing on a tricky little plastic keyboard or tapping a minuscule on-screen keyboard. While these keyboards are much better than they used to be, typing on them is still far from being a pleasant experience. This means that mobile users hate typing long reams of text. Your mobile site can make life easier for them by:

* Using shorter URLs
* Adding an autocomplete function to text fields and search fields
* Providing an easy way to browse popular pages or products, rather than having to search
* Preemptively filling as much information as possible in web forms

**6. No, or limited, multitasking**

Multitasking — in the sense of being able to run more than one app at once — is finally starting to take off on mobile devices. That said, many devices still can't multitask, and even those that can, don't offer the power or flexibility of desktop multitasking.

This poor multitasking support can affect the way you design your mobile websites. For example, consider including Twitter/Facebook sharing buttons on every page of your site, so that users don't have to copy and paste your page's URL to a different window or app in order to share the page.

**7. Websites are not always viewed in browsers**

On the desktop, users nearly always surf the web in a browser. However,  86% of online smartphone time is [spent in apps](https://www.emarketer.com/Article/Smartphone-Apps-Crushing-Mobile-Web-Time/1014498) and only 14% is spent in web browsers.

This fact can have various subtle consequences for your mobile site designs:

* Often there's no URL bar - This means users can't glean any information or context by looking at your page URL, copy and paste the URL, or type in a new URL.
* Reduced screen real estate - Mobile browser viewports are small at the best of times, but websites viewed in apps are often crammed into even smaller spaces thanks to app toolbars, buttons, and other widgets surrounding the page.
* Limited functionality - Typically, browser features such as bookmarking, opening links in new windows, or printing are not available when viewing web pages within apps.

Including sharing buttons within your pages can make it easier for users to share your content while inside another app. Also, don't force people to rely on your page URLs for context. Include useful information to help users orient themselves.

**8. Portrait screens**

Most desktop displays have a landscape orientation; however, the opposite is true of mobile devices- most users hold their device in a portrait orientation.

This is fueling a trend toward mobile sites that are well-suited to a vertical orientation, resulting in design decisions, such as:

* Fewer columns of content (a single column is ideal)
* No overly-wide elements: This includes large multi-column tables, as well as extra-wide images, slideshows, Flash movies, and iframes
* Navigation along the top rather than down the side

**9. People use mobile devices differently**

This is perhaps one of the most important overall differences between the mobile and desktop web experience. Much of the time, people use their mobile device in a different way than their computer, and for different types of tasks.

At their computers, users are:

* Sitting at a desk
* Frequently in an office environment
* Often working
* Sometimes randomly surfing the web
* Often creating content
* Focused on the computer, not so much on their environment

On a mobile device, however, users tend to be:

* Sitting on the couch at home
* Walking around, inside or outside
* Queuing for something
* Waiting for a bus, train, or plane, or travelling
* Looking for a specific piece of information
* Mostly consuming content
* Easily distracted by their environment

These differences affect the types of sites that work well on mobile devices. For example, websites and web apps that are designed to help people create content will generally be more popular on desktop computers, while sites that let users consume content easily, or find a specific piece of information while they're on the move, will attract mobile users more.

**10. Failing Flash support**

On the desktop, Flash is almost ubiquitous, with over 90% of browsers having the Flash player installed. With the mobile web, however, it's a different story.

No iOS devices run Flash. Android devices using version 2.2 or later can run Flash, although many users choose to turn it off since it can cause performance and stability problems. In all, well under half of the world's mobile devices are Flash-capable, and even when a device can run Flash, it's usually not a pleasant experience.

Therefore, a website relying on Flash is not going to be popular on mobile devices. Fortunately, there are now ways that you can create a Flash-like experience for both mobile and desktop users without needing to use Flash itself. Modern web standards like HTML5, CSS3 and SVG allow you to create vectors, animations, games, interactive elements, and embedded videos that will run well on all modern mobile devices, Flash-enabled or not.

When designing any website that will be viewed on mobile devices, it's important to keep these differences in mind. By accommodating both desktop and mobile in your website strategy, you can provide a great experience for both of these audiences.