

# Smoke-testing your apps 101: A guide for the non-techie co-founder

By **DENIS DUVAUCHELLE**, Sunday, 9 Feb '14, 01:00pm



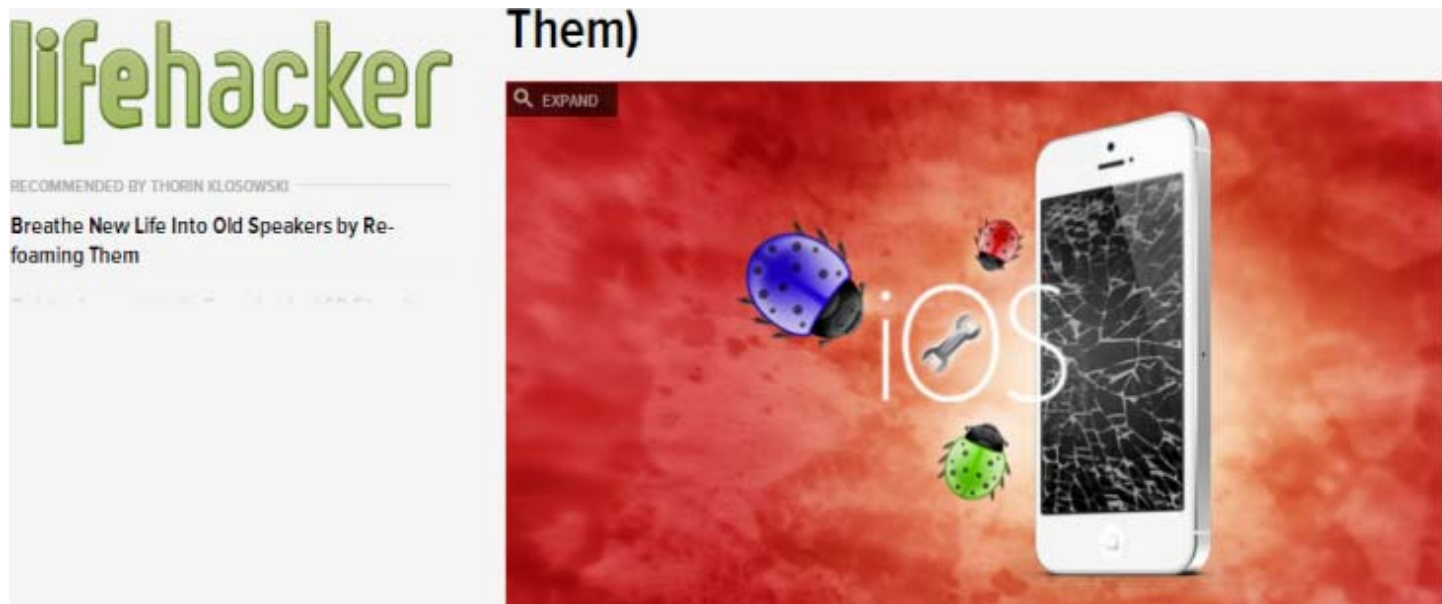
---

**D**enis Duvauchelle is CEO and co-founder of **Twoodo**, helping you organize everything and anything using #hashtags.

---

It's doubtful that if our **team collaboration startup** had as many bugs as iOS7 did that the reaction would be to **write a nice article on Lifehacker** explaining how to solve the problems yourself.

What an amazing achievement for a company, when you think about it! Any lesser technology company would have customers, dare I say, dropping like flies...



Simple bugs can be fatal to your website – especially if you are a SaaS company like we are. If a user arrives on your website and can't carry out a simple task like log in or reset his forgotten password, you could lose that user forever.

We learned this the hard way. Sure, having people from your team test the tool and look for bugs is important, but it's not consistent or thorough enough. Here, we are going to introduce all you non-techies to the world of smoke tests.

Smoke-testing was originally coined to explain how electronic engineers checked to see if their gadget worked – plug it in, and if smoke comes out...

## **Wait – so how does this apply to apps?**

The importance (and cost effectiveness) of the smoke test is typically unknown to non-techie managers and co-founders. Regular smoke tests can be considered an integral part of growth hacking. It minimizes the chance that your web app or phone app will fail – and as we all know, it it only takes one fault to lose a customer forever.

This is an introductory guide to what it is, how it can be done, resources for conducting it and examples to guide readers.

Smoke tests exist to check basic functionalities and should be a consistent part of your testing process. This can include something as simple as “can I log in?”

A smoke test will ensure that none of the most basic and obvious failings are left to chance. A deeper test should not be performed until you have cleared a smoke test 100 percent so that it clears the software of fundamental flaws.

## **Step 1: Decide what to test**

Look at what your application is trying to achieve. What are the most obvious baby steps that are required to get there? What are the minimum vital requirements, and in what logical order can you put them in?

Create a test suite. A test suite is when you group a set of test cases that are linked in some way (eg. link by function).

A smoke test will not include variables, or “what ifs.” They are about yes/no answers that critically need a “yes” before you start more advanced testing (also known as “test cases”).

Let’s take the example of creating an online forum. In order for this to work I need to:

- a) log in
- b) create a username
- c) have an avatar

d) post a message

e) reply to a message

## Step 2: Record in a spreadsheet

	A	B	C	D	E	F	G
1	<b>Test suite</b>			<b>Total tests =</b>	<b>70</b>	<b>Success rate</b>	<b>Result - 1</b>
2							Guillaume
3							2013-11-26
4							1.5.5.08
5							75%
6	<b>Topic</b>	<b>Step</b>	<b>Page link</b>	<b>Action</b>	<b>Expected</b>		<b>Status</b>
7		Registration	<a href="#">dev.twodo.com</a>	Register using various wrong inputs.	Check for warning messages on wrong or already used login/username	83%	PASSED
8				Register using valid inputs with a @mailinator.com email login.	Check for redirection to the walkthrough.	100%	PASSED
9		Walkthrough	<a href="#">dev.twodo.com</a>	View all steps.	Check if images and text is up-to-date	100%	PASSED
10				Upload avatar and set first/last name.	Check last step redirects to feed.	83%	PASSED

The image above is an example from our team. You can [find the template here](#). It's essential to keep a record of what has and hasn't worked – basic organization saves tons of time later on. We divided our success into pass, partial and fail.

- Pass: everything functioned perfectly
- Partial: you may not initially realize that an action could be subdivided even more, and part will work, part not.
- Fail: did not work

We described the precise action we wanted to test, and then added a brief description of how it should happen in the next column. Example:

Action	Expected
--------	----------

Post using a new team name	Check post is added with expected headers
----------------------------	---

## Step 3: Automate the smoke test

It is super important not to assume that once an action has passed a smoke

test that it will always pass. Smoke tests allow you to constantly check that these basic functions have not been affected over time or broken down over time.

Do not stop smoke-testing. Ever.

When your test suite of smoke tests are at a 100 percent clearance rate, think of automating it, but do not get complacent. The recommended frequency of smoke tests is daily test if your company does development every day.

At the absolute minimum, conduct a smoke test before every release and after every patch.

A rule of thumb for smoke tests:

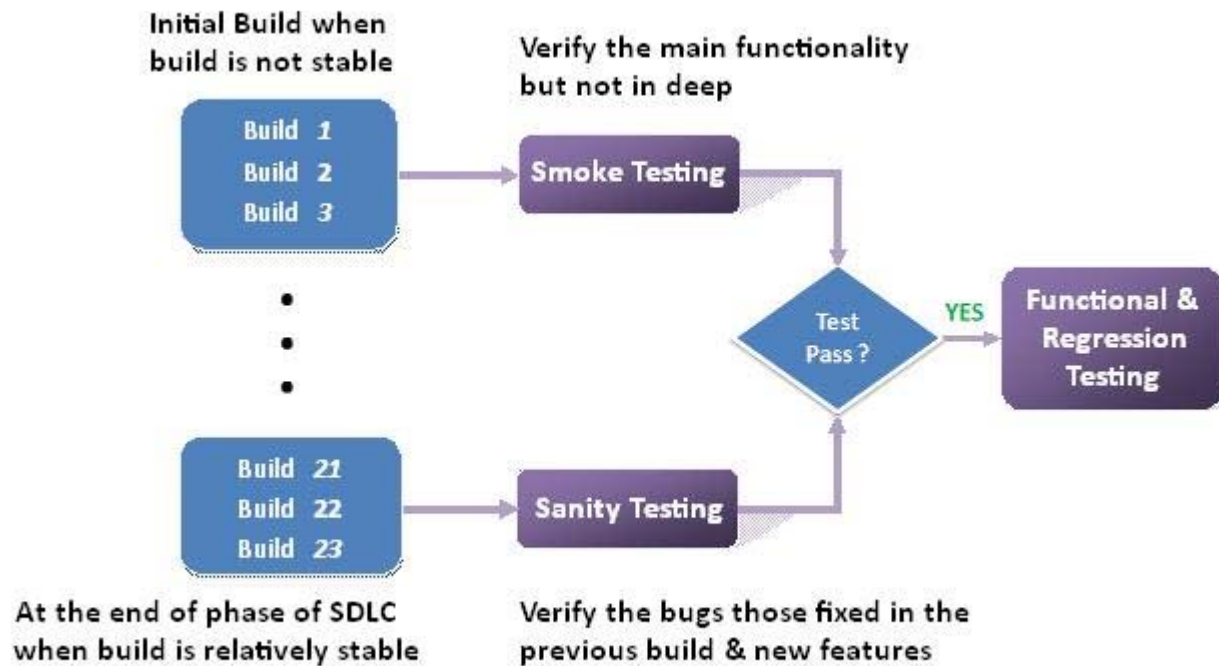
- Minimum time: 30mins
- Maximum time: 60mins

In the long run, automating a smoke test is time saving, but do do it manually every now and again because the human eye can pick up on details that machines can't.

## Resources for testing:

- [TestManagement](#) (when your company grows bigger)
- [Inflectra](#)
- [SeleniumHQ](#) (Web apps)
- [Our template](#)

After the smoke test? The Sanity Test!



Do you have other smoke testing tips that works for you? Share them below.

---

---

 SHOW 4 COMMENTS

