Quickstart on web testing on 2019

Juan M. Fonseca-Solís

juan.fonsecasolis.github.com

March 1, 2020

Outline

Introduction

CI & CD

Manual testing experiences

- Functional testing
- Non-functional testing

Automated testing experiences

Conclusions

Why do we test web application?

- Some examples of software development in Costa Rica: web applications, big data and embedded systems
- New trends for 2019 regarding web applications are [7, 8, 9]:
  - Progressive web apps
  - Single page applications
  - Chat bots
  - Blockchain

Scope of this presentation

By application: desktop, embedded, mobile, **web (responsive, desktop)**

Methodology: **agile**, cascade, XP

Market: Asia, Europe, **US**

Development cycle: **legacy**, new

Role: architect, **analyst**, **automation engineer**, **designer**, **tester**

CI & CD

- Before talking about web testing we should understand how software is released
- There are two extended practices:
  - Continuous integration (CI)
  - Continuous delivery (CD)
- Testing is performed at different levels on CI & CD as we will see in short

CI & CD

Continuous integration [1]: [1]

- *Is a development practice that requires developers to integrate code into a shared repository several times a day*.
- *Each check-in is then verified by an automated build, allowing teams to detect problems early.* - Thoughworks

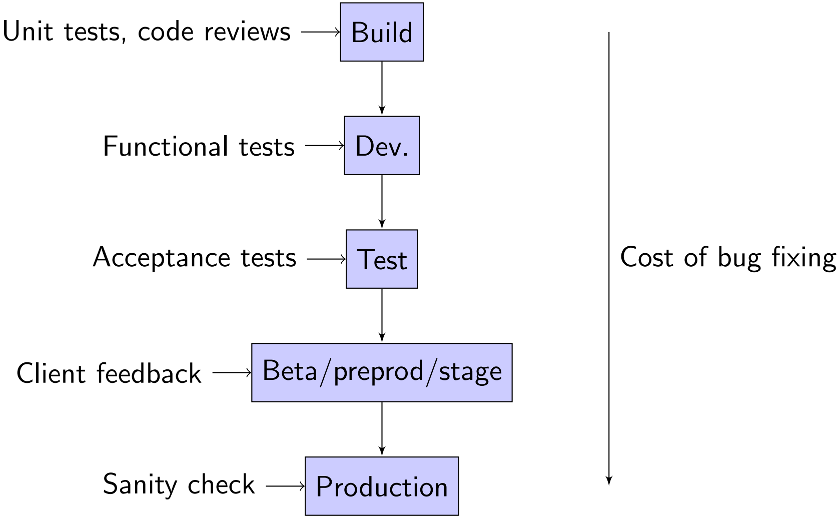
[1] Have you hear about Mr. hit n‘ run build breaker? `https://youtu.be/fuPFz5deXOw`

Continuous delivery (CD)

Continuous delivery [2]:

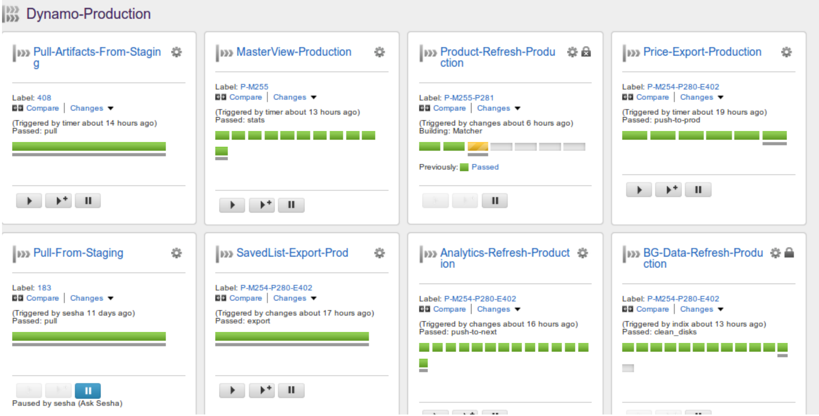
- *CD focuses an organization on building a streamlined, automated software release process.*
- *The feedback loop revolves around delivery of software to the end user as quickly as possible.* - Atlassian

Continuous delivery (CD) (cont.)



Tools used in CI/CD (cont.)

Here is an example of a pipeline implemented in GO:



Dynamo-Production

Pull-Artifacts-From-Staging
Label: 408
Compare | Changes
(Triggered by timer about 14 hours ago)
Passed: pull

MasterView-Production
Label: P-M255
Compare | Changes
(Triggered by timer about 13 hours ago)
Passed: stats

Product-Refresh-Production
Label: P-M255-P281
Compare | Changes
(Triggered by changes about 6 hours ago)
Building: Matcher
Previously: 🟩 Passed

Price-Export-Production
Label: P-M254-P280-E402
Compare | Changes
(Triggered by timer about 19 hours ago)
Passed: push-to-prod

Pull-From-Staging
Label: 183
Compare | Changes
(Triggered by sesha 11 days ago)
Passed: pull
Paused by sesha (Ask Sesha)

SavedList-Export-Prod
Label: P-M254-P280-E402
Compare | Changes
(Triggered by changes about 17 hours ago)
Passed: export

Analytics-Refresh-Production
Label: P-M254-P280-E402
Compare | Changes
(Triggered by changes about 16 hours ago)
Passed: push-to-next

BG-Data-Refresh-Production
Label: P-M254-P280-E402
Compare | Changes
(Triggered by indix about 13 hours ago)
Passed: clean_disks

`http://codingnirvana.github.io/pune-scala/pictures/go_pipelines.png`

Manual testing

- As you might already know, testing can be classified in two categories: manual and automated
- One complements the other:
    - We cannot ask a machine to test what a person can't verify itself!

Manual testing

- Some must-know concepts on manual testing are [4]:
  - TCs estimation techniques
  - Testing levels
  - Quality control vs. quality assurance
  - Verify vs. validate
  - Test life cycle

Manual testing (cont.)

- Test case estimation techniques [4]:
  - Boundary values
  - Equivalent classes
  - Cause-effect diagrams
  - Transition diagram (finite state machine)
- No one will ask you to use them, but everyone would be mad if you don't find the bugs!

Manual testing (cont.)

- **The art of software testing:**
  - Chapter 4: Test-Case Design (p. 35-69)
    - Boundary value analysis (p. 46)
    - Equivalent classes
    - Cause-effect diagrams (p. 50)
  - `http://barbie.uta.edu/~mehra/Book1_The%20Art%20of%20Software%20Testing.pdf`
- **Requirements-Based Testing- Cause-Effect Graphing:**
  - `http://barbie.uta.edu/~mehra/59_RBT_Cause-Effect_Graphing2.pdf`

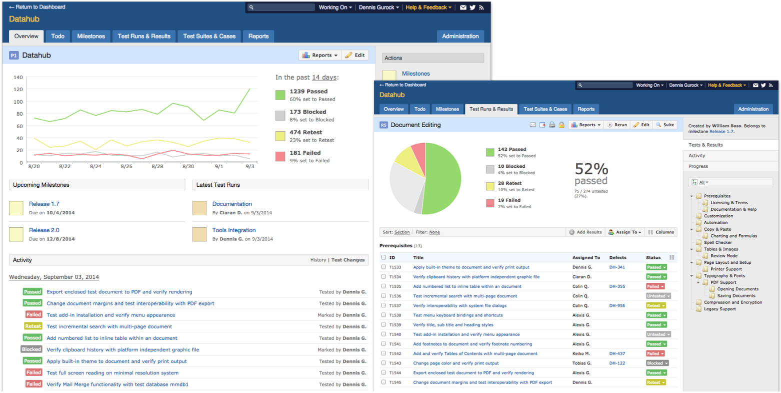
Manual testing (cont.)

You should also know about test case management systems (TCMS), which allow to:

- Maintain test cases better than using a spreadsheet
- Log test runs
- Create test reports in a visually appealing way
- Calculate quality metrics (like the number of bugs found per TC)

Manual testing (cont.)

Here is an example of a TCMS called Testrails:



In the past 14 days:

- 🟩 **1239 Passed** 60% set to Passed
- ⬜ **173 Blocked** 8% set to Blocked
- 🟨 **474 Retest** 23% set to Retest
- 🟥 **181 Failed** 9% set to Failed

Upcoming Milestones

- Release 1.7 Due on **10/4/2014**
- Release 2.0 Due on **12/8/2014**

Latest Test Runs

- Documentation By **Ciaran D.** on 9/3/2014
- Tools Integration By **Dennis G.** on 9/3/2014

Activity — History | **Test Changes**

Wednesday, September 03, 2014

Passed	Export enclosed test document to PDF and verify rendering	Tested by **Dennis G.**
Passed	Change document margins and test interoperability with PDF export	Tested by **Dennis G.**
Failed	Test add-in installation and verify menu appearance	Marked by **Dennis G.**
Retest	Test incremental search with multi-page document	Tested by **Dennis G.**
Passed	Add numbered list to inline table within an document	Tested by **Dennis G.**
Blocked	Verify clipboard history with platform independent graphic file	Marked by **Dennis G.**
Passed	Apply built-in theme to document and verify print output	Tested by **Dennis G.**
Failed	Test full screen reading on minimal resolution system	Tested by **Dennis G.**
Failed	Verify Mail Merge functionality with test database mmdb1	Tested by **Dennis G.**

Document Editing

- 🟩 **142 Passed** 52% set to Passed
- ⬜ **10 Blocked** 4% set to Blocked
- 🟨 **28 Retest** 10% set to Retest
- 🟥 **19 Failed** 7% set to Failed

52% passed

75 / 274 untested (27%).

Created by William Bass. Belongs to milestone Release 1.7.

Prerequisites (13)

ID	Title	Assigned To	Defects	Status
T1533	Apply built-in theme to document and verify print output	Dennis G.	DH-341	Passed
T1534	Verify clipboard history with platform independent graphic file	Ciaran D.		Passed
T1535	Add numbered list to inline table within an document	Colin Q.	DH-355	Failed
T1536	Test incremental search with multi-page document	Colin Q.		Untested
T1537	Verify interoperability with system file dialogs	Colin Q.	DH-956	Retest
T1538	Test menu keyboard bindings and shortcuts	Alexis G.		Passed
T1539	Verify title, sub title and heading styles	Alexis G.		Passed
T1540	Test add-in installation and verify menu appearance	Alexis G.		Untested
T1541	Add footnotes to document and verify footnote numbering	Alexis G.		Passed
T1542	Add and verify Tables of Contents with multi-page document	Keiko M.	DH-437	Failed
T1543	Change page color and verify print output	Tobias G.	DH-122	Blocked
T1544	Export enclosed test document to PDF and verify rendering	Alexis G.		Passed
T1545	Change document margins and test interoperability with PDF export	Dennis G.		Retest

- Prerequisites
  - Licensing & Terms
  - Documentation & Help
- Customization
- Automation
- Copy & Paste
  - Charting and Formulas
- Spell Checker
- Tables & Images
  - Review Mode
- Page Layout and Setup
  - Printer Support
- Typography & Fonts
  - PDF Support
    - Opening Documents
    - Saving Documents
- Compression and Encryption
- Legacy Support

https://marketplace-cdn.atlassian.com/files/images/e9a091f7-2f37-4956-870c-558f4ac3efba.png

Manual testing (cont.)

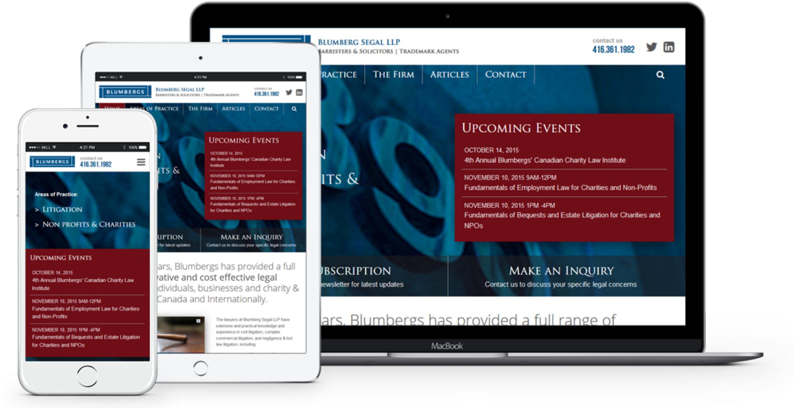
- Now let's talk about different kinds of manual testing:
    - Web responsive testing
    - Accessibility testing
    - Performance testing
    - Security testing
    - A/B testing
    - (to mention a few...)

Web responsive testing

- **Purpose:** to render the same webpage in different devices
- Must-know concepts:
  - Responsive design
  - XHTML, RHTML, HTML
- Tools:
  - Vyzor, Reflector / Airserver
  - Charles Proxy, Fiddler
  - Browser stack

Web responsive testing (cont.)

Test is performed at least in three formats:



UPCOMING EVENTS

OCTOBER 14, 2015
4th Annual Blumbergs' Canadian Charity Law Institute

NOVEMBER 10, 2015 9AM-12PM
Fundamentals of Employment Law for Charities and Non-Profits

NOVEMBER 10, 2015 1PM -4PM
Fundamentals of Bequests and Estate Litigation for Charities and NPOs

newsletter for latest updates

Contact us to discuss your specific legal concerns

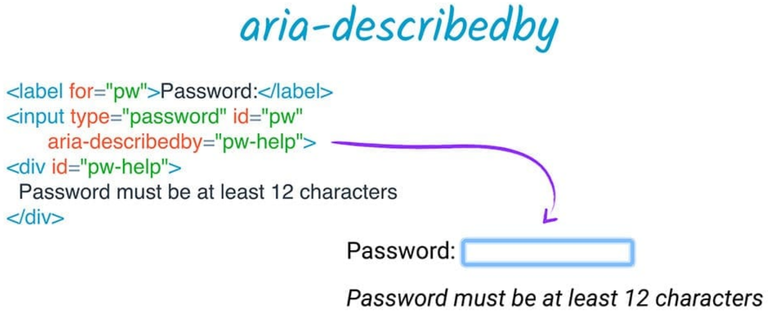
ars, Blumbergs has provided a full range of

https://www.thewire.ca/media/blumbergs-hero.png

Accessibility testing

- **Purpose:** make content available for a broader range of users
- Must-know concepts [5, 6]:
  - POUR principle (perceivable, observable, understandable, robust)
  - WCAG 2.0 guidelines
  - Web Aria
- Tools:
  - Paciello toolbar, Sortsite, Webaim
  - Google Accessibility toolbar
  - Screen readers (VoiceOver, Jaws)
  - Windows high contrast mode

Accessibility testing (cont.)



aria-describedby

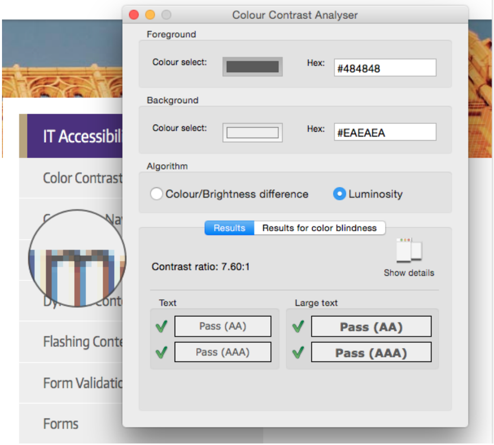
```
<label for="pw">Password:</label>
<input type="password" id="pw"
    aria-describedby="pw-help">
<div id="pw-help">
  Password must be at least 12 characters
</div>
```

Password:

Password must be at least 12 characters

https://developers.google.com/web/fundamentals/accessibility/semantics-aria/imgs/aria-describedby.jpg

Accessibility testing (cont.)



Colour Contrast Analyser

Foreground

Colour select: ▇ Hex: #484848

Background

Colour select: □ Hex: #EAEAEA

Algorithm

○ Colour/Brightness difference ● Luminosity

Results | Results for color blindness

Contrast ratio: 7.60:1

Show details

Text	Large text
✓ Pass (AA)	✓ Pass (AA)
✓ Pass (AAA)	✓ Pass (AAA)

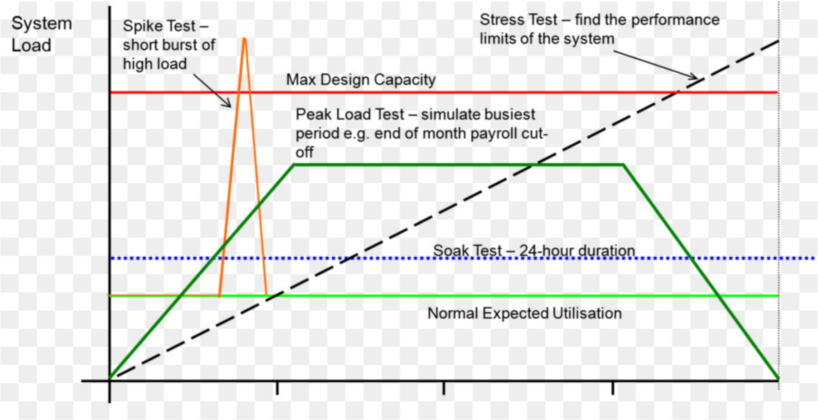
https://s3-us-west-2.amazonaws.com/uw-s3-cdn/wp-content/uploads/sites/54/2014/10/06144407/ColourContrastChecker.png

Performance testing

- **Purpose:** know how the webpage behaves on different work loads.[2]
- Must-know concepts:
  - Stress, load, performance, soak
  - Ramp-up period, throughput
- Tools:
  - JMeter, Blazemeter, Taurus

[2] Here is an example of why this is useful: `https://juanfonsecasolis.github.io/blog/JFonseca.suavizadoTraficoServidorWeb.html`.

Performance testing (cont.)



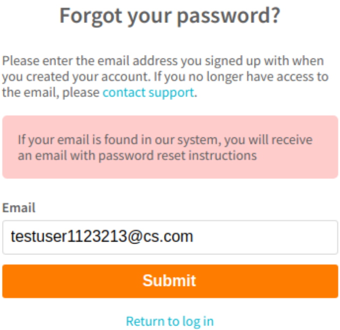
https://banner2.kisspng.com/20180420/btq/kisspng-load-testing-stress-testing-software-performance-t-peak-5ada50b21f6611.0507167015242569461286.jpg

Security testing

- **Purpose:** protect data integrity and privacy
- Must-know concepts:
  - OWASP top 10
  - XSS, SQL injection, LDAP injection
- Tools:
  - BURP
  - Kali Linux and tools installed

Security testing (cont.)

Here are some examples of a fixed data leaking and XSS:



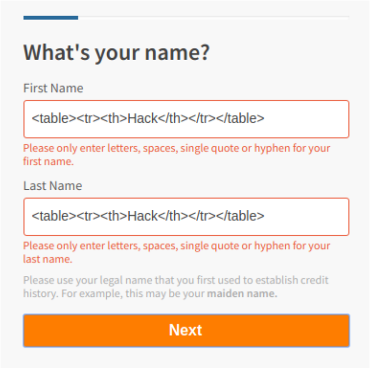
Forgot your password?

Please enter the email address you signed up with when you created your account. If you no longer have access to the email, please contact support.

If your email is found in our system, you will receive an email with password reset instructions

Email

testuser1123213@cs.com



What's your name?

First Name

<table><tr><th>Hack</th></tr></table>

Please only enter letters, spaces, single quote or hyphen for your first name.

Last Name

<table><tr><th>Hack</th></tr></table>

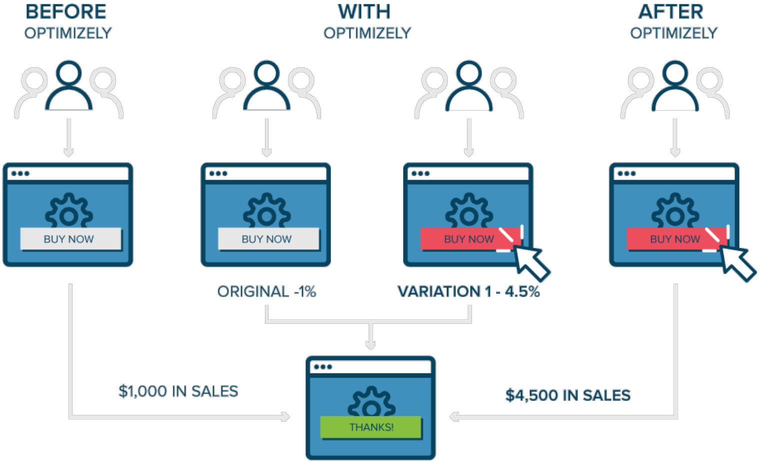
Please only enter letters, spaces, single quote or hyphen for your last name.

Please use your legal name that you first used to establish credit history. For example, this may be your **maiden name.**

A/B testing

- **Purpose:** increase revenue by getting feedback from users
- Must-know concepts [3]:
  - Hypothesis testing (null, alternative)
    - Is the revenue increased with any variation?
  - Statistical significance
    - T-student, normal distribution
- Tools:
  - Optimizely
  - Mixpanel
  - Splunk Insights
  - Adobe Omniture

A/B testing (cont.)



https://images.ctfassets.net/zw48pl1isxmc/3lRlyAZISsG0iWyiO8Ug84/7a32f963dd0fdc3ccb4483fc56aa55d2/ab-testing-optimizely-2.png

Automated testing

- To finish, a quick overview on automated testing:
  - UI test automation
  - Web service test automation

Pros and cons of automation

- Pros:
  - Input combinations can be tested exhaustively
  - Repetitive and time consuming flows can be automated
- Cons:
  - Periodic maintenance on test suites can be time consuming and complex
  - Tests are only able to detect bugs for which they were programmed to

UI test automation

- **Purpose:** catch bugs by reproducing a flow in the web browsers
- Must-know concepts:
  - Page-object model
  - Page factory
  - Locators (XPath, CS)
  - Coding standards (.Net, Java)
- Tools:
  - Selenium (Webdriver, IDE, grid)
  - Cucumber

Web service test automation

- **Purpose:** catch bugs by checking data and operations
- Must-know concepts:
  - SOAP, RESTFul
  - XML, JSON
- Tools:
  - JUnit
  - Postman
  - SOAP UI
  - RestAssure
  - CURL, JQ

Conclusions

- CD/CI allows to serve software more frequently and with less risk.
- Proper functional testing requires knowledge on test case estimation techniques and TCMS.
- Non-functional testing covers areas like: responsive design, accessibility, performance, security, and A/B testing.
- Test automation makes testing activities easier, but at cost of maintenance and visibility.

References I

- Thoughworks.
  Continuous integration.
  https://www.thoughtworks.com/continuous-integration.

- Attlasian
  Continuous Delivery Principles.
  https://www.atlassian.com/continuous-delivery/principles.

- Optimizely
  A/B Testing.
  https://www.optimizely.com/optimization-glossary/ab-testing/.

References II

- Glenford J. Myers, Corey Sandler, Tom Badgett
  The art of software testing.
  ISBN: 978-1-118-03196-4, Wiley, November 2011.

- WebAIM
  Introduction to Web Accessibility.
  `https://webaim.org/intro/`.

- W3C Web Accessibility Initiative (WAI)
  WAI-ARIA Overview.
  `https://www.w3.org/WAI/standards-guidelines/aria/`.

- Uriah Solution Pvt. Ltd.
  Top 10 Web Development Trends in 2019.
  `https://www.uriahsolution.com/blog/website-development-trends-2018/`.

References III

Noemi Balog
Top 8 Web Development Trends of 2019.
https://www.codingdojo.com/blog/web-development-trends-of-2019.

Jake Prins, Free Code Camp 2019
Interesting Web Technologies to follow in 2019.
https://medium.freecodecamp.org/trending-web-technologies-to-follow-in-2019.

License

Click on the license icon to get more information.