



A TECHWELL EVENT

W17

Compatibility Testing

Wednesday, October 3rd, 2018 3:00 PM

Engineering for Compatibility

Presented by:

Melissa Benua

mParticle

Brought to you by:



350 Corporate Way, Suite 400, Orange Park, FL 32073
888-268-8770 · 904-278-0524 - info@techwell.com - <http://www.starwest.techwell.com/>

Melissa Benua

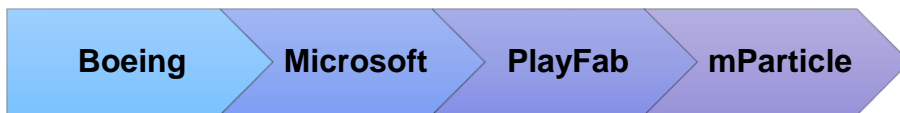
In her career at companies from large to small, Melissa Benua has worked in nearly every software development role-engineer, test, DevOps, and program management. She's created and run high-availability, high-quality services at Boeing and Microsoft on products such as Bing, Cortana, and Xbox One. Melissa discovered her love of massively-scaled systems while working on the Bing backend, where she honed the art of keeping highly-available complex systems up while undergoing massive code churn. Now a senior engineer and manager at the disruptive gaming startup PlayFab, Melissa isn't afraid to mix traditional approaches with bold new ideas to make her products better, faster, and more reliable. She's passionate not only about maximizing efficiency both in her product code and in her developer tools but also about sharing best practices among colleagues and the tech world at large!

Test Design for Fully Automated Build Architectures

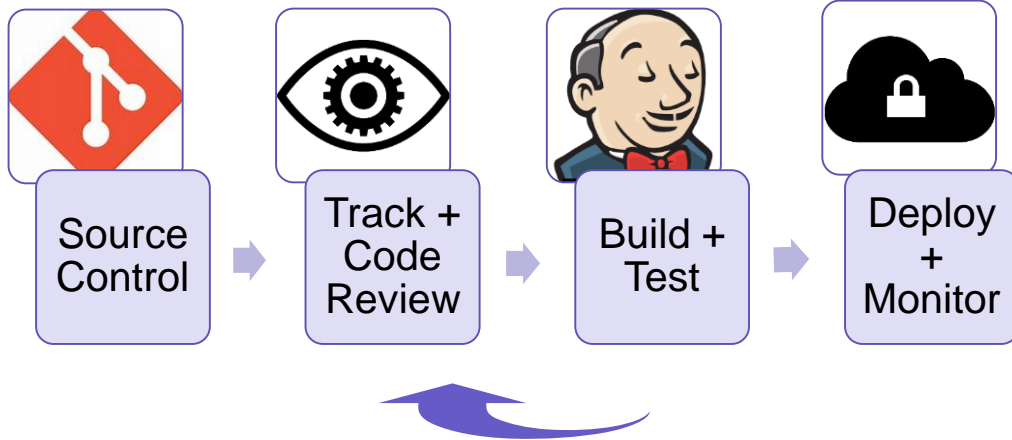
Melissa Benua
Senior Technical Lead, mParticle
mbenua@gmail.com
@queenofcode
STARWEST 2018



About Me



About The Tutorial



3



Guiding Principles

What are we doing here?

4



GUIDING PRINCIPLES

Key Test Features

Important

Reliable

Specific

5



GUIDING PRINCIPLES

Importance

Triage scenarios based on priority

IMPOSSIBLE to cover every scenario

Understand what failures can be tolerated

Run the most important tests first

6



GUIDING PRINCIPLES

Importance

Build Failure?

```

Directory: C:\Users\Wlissa\Documents\Github\ci\Samplewebsite

mode                LastWriteTime         Length Name
----                -
d-----            8/19/2017   18:18      packages
d-----            8/19/2017   18:18      webapp\integration_tests
d-----            8/19/2017   18:18      3011 Samplewebsite.sln

C:\Users\Wlissa\Documents\Github\ci\Samplewebsite [master a]> msbuild .\Samplewebsite.sln
Microsoft (C) Build Engine version 14.0.25420.1
Copyright (C) Microsoft Corporation. All rights reserved.

building the projects in this solution one at a time. To enable parallel build, please add the "/m" switch.
Build started 8/20/2017 12:31:26.
Project "C:\Users\Wlissa\Documents\Github\ci\Samplewebsite\Samplewebsite.sln" on node 1 (default targets).
  Waftecs\lnt\lntconf\lntoperation
    building solution configuration "Debug|Any CPU"
  Project "C:\Users\Wlissa\Documents\Github\ci\Samplewebsite\Samplewebsite.sln" (1) is building "C:\Users\Wlissa\Documents\Github\ci\Samplewebsite\Samplewebsite.metaproj" (2) on node 1 (default targets).
  Build:
    C:\Windows\Microsoft.NET\Framework\v4.0.30319\aspnet_compiler.exe -v /localhost_0841 -p .\..\websites\Samplewebsite\ -u -f -d /precompiledweb/localhost_0841
  Waftecs\lnt\lntconf\lntoperation
    The directory "C:\Users\Wlissa\Documents\Github\websites\Samplewebsite\" doesn't exist. (C:\Users\Wlissa\Documents\Github\ci\Samplewebsite\Samplewebsite.metaproj) (default target) (2) =>
  Build FAILED.
    ASPNETCOMPILER : error 1001: The directory "C:\Users\Wlissa\Documents\Github\websites\Samplewebsite\" doesn't exist. (C:\Users\Wlissa\Documents\Github\ci\Samplewebsite\Samplewebsite.metaproj)
    0 warning(s)
  Time Elapsed 00:00:00.30
  
```

UI Test Failure?

The screenshot shows a web browser window with the URL 'Public/Tutorial/Welcome.html'. The page content displays a success message: 'Congratulation! You are successfully logged in!'. Below the message, there is a yellow box with the text: 'During developing your site will change. Buttons, colors, aligns and more. Some changes will be correct. You can press "Approve step to baseline" and all expected changes will be approved to baseline'. A yellow 'NEXT' button is visible at the bottom of the message box. The browser's developer tools are open, showing the 'Test commands' section with a list of commands including 'span', 'T Me', 'T me@foo.com', and 'span'.

GUIDING PRINCIPLES

Reliability

No flakiness

No false-negatives or false-positives

Repeatable without human intervention

Cleans up after itself

GUIDING PRINCIPLES

Reliability

TeamCity > Trunk > Server Code Analysis > IntegrationBuild (HSQLDB + Jacoco Coverage) > #944 (30 Jul 16 03:02)

Run ... Actions Edit Configuration Settings |

Overview Changes 13 Tests Build Log Parameters Dependencies Issues Artifacts #943 | All history | #945 »

Code Coverage

Result: ✔ Tests failed: 13 (2 new), passed: 16706, ignored: 138, muted: 21 Agent: W7-esxi-13 (TeamCity pool)
 Time: 30 Jul 16 03:02 - 10:02 (6h:59m) Triggered by: Schedule Trigger on 30 Jul 16 03:00

Code coverage summary View full report >

Classes: 70.1% 10466/14920	Methods: 61.8% 50631/81864	Blocks: 35.2% 1016105/2882385	Lines: 61.9% 194560/314148
Diff: -0.06%	Diff: -0.02%	Diff: -0.01%	Diff: -0.02%

13 tests failed (2 new)

Group by: package/suite

- All tests
- Integration tests: jetbrains.buildServer.agent.impl (1)
 - * UpdateSourcesAutoTest.should_do_server_checkout_if_agent_cannot_checkout_from_repository |
 - This test looks flaky:**
 - Different test status of build configurations with the same VCS change
 - Frequent test status changes: 2 changes out of 38 invocations
 - [View test history >](#)
- Maintenance Suite: jetbrains.buildServer.serverSide.maintenance (1)

9

GUIDING PRINCIPLES

Specificity

Clear answer to a clear question

Have one main goal

Don't 'boil the ocean'

Minimal overlapping coverage

10

GUIDING PRINCIPLES

Specificity

Test Explorer | Run All | Run... | Playlist : All Tests

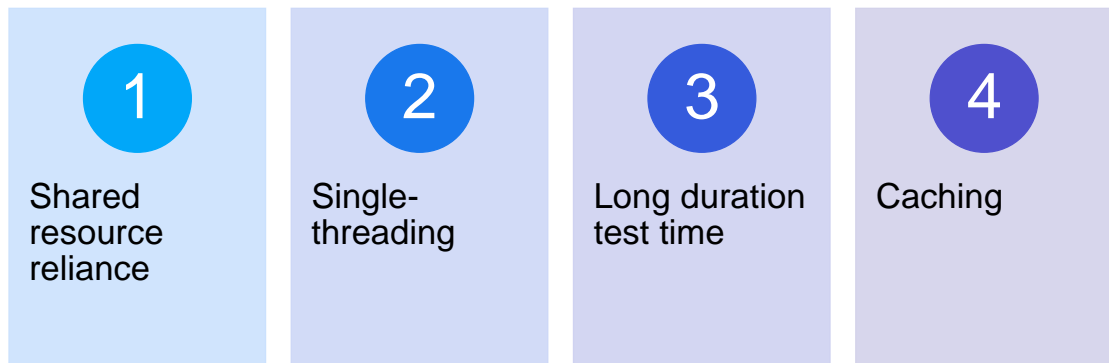
- ▲ **HomeControllerTest (7)**
 - ✓ Index_Success 38 ms
 - ✓ Login_BadPassword < 1 ms
 - ✓ Login_BadUsername < 1 ms
 - ✓ Login_EmptyUsername < 1 ms
 - ✓ Login_Success < 1 ms
 - ✓ Register_AlreadyExists < 1 ms
 - ✓ Register_Success < 1 ms
- ▶ **ValuesControllerTest (5)**

11



GUIDING PRINCIPLES

Proceed With Caution



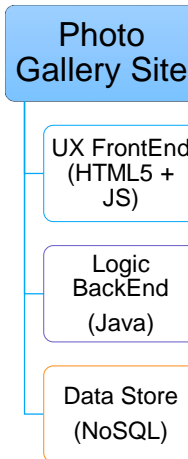
12



GUIDING PRINCIPLES

Exercise: Test Cases

- Most important?
- Easiest?
- Most reliable?



13



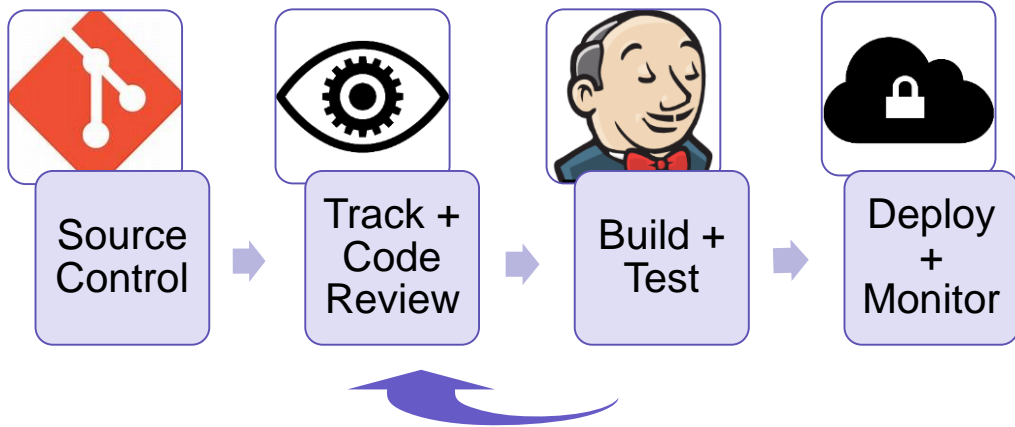
Automated Pipeline Structure

What does it even look like?

14



AUTOMATED PIPELINE STRUCTURE

CI + CD Pipeline

15



AUTOMATED PIPELINE STRUCTURE

Beyond Unit Tests

- Project structure?
- Moving parts?
- Functional boundaries?
- Shared resources?
- Mocking potential?

Develop: Diff Build

- Compile change against mainline
- Execute unit tests

Build: Continuous Integration

- Compile change as a part of mainline submit
- Execute functional tests

Deploy: Continuous Deployment

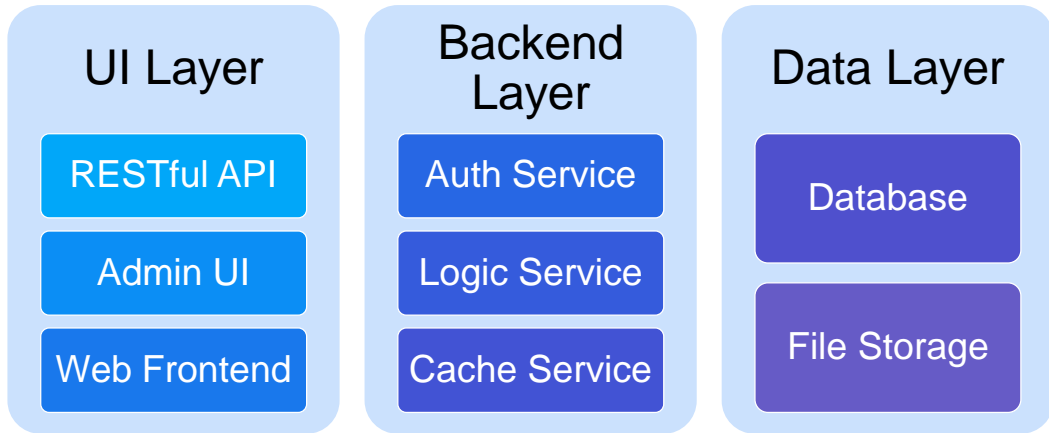
- Start staging environment
- Deploy staging environment
- Execute UI + load tests

16



AUTOMATED PIPELINE STRUCTURE

Example Service Architecture



17

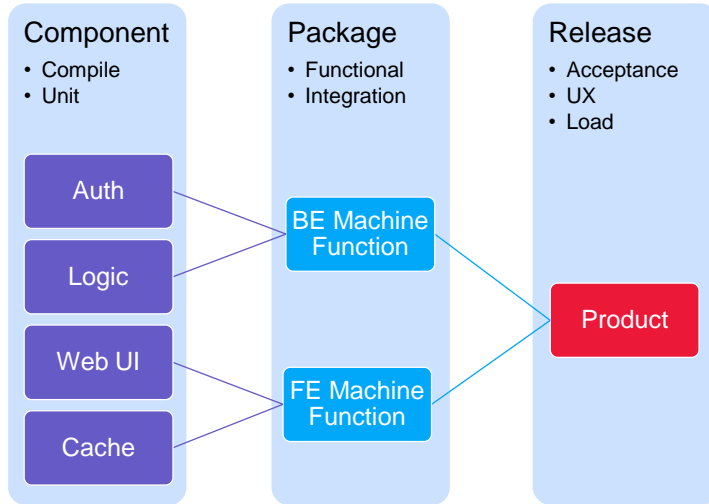


18



AUTOMATED PIPELINE STRUCTURE

The Matrix

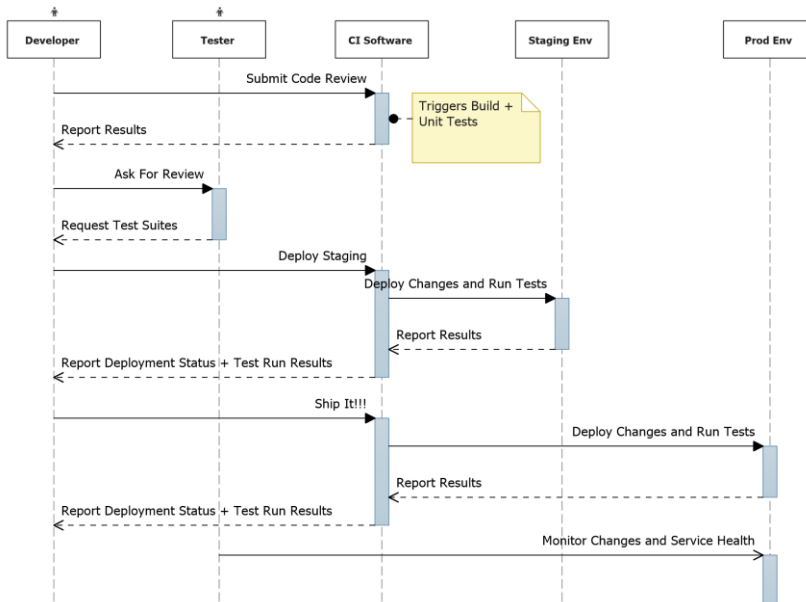


19



AUTOMATED

End to End



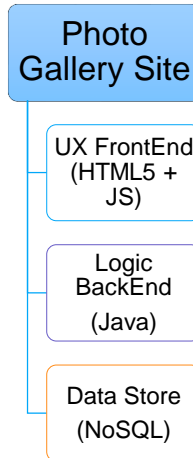
20



AUTOMATED PIPELINE STRUCTURE

Exercise: Mapping Categories to Stages

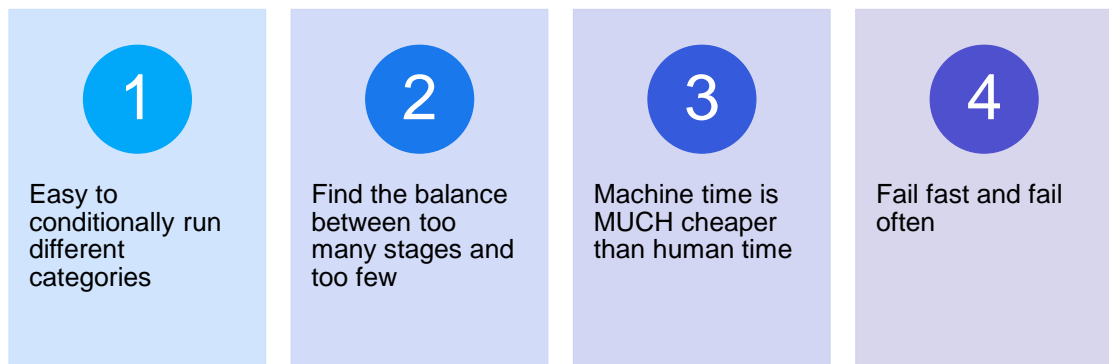
- What should we run?
- When should we run it?
- How long should we wait?



21



AUTOMATED PIPELINE STRUCTURE

Key Takeaways

22



Monitoring and Reporting

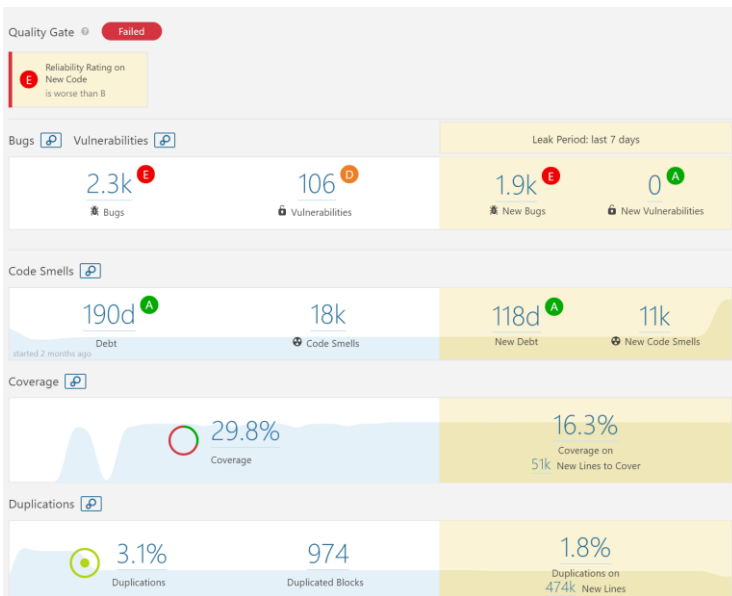
What just happened?

23



MONITORING AND REPORTING

Code Metrics

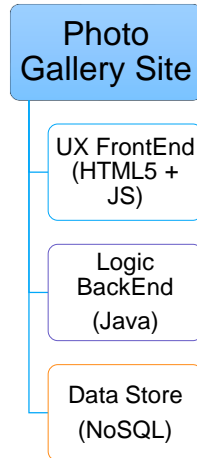


24



MONITORING AND REPORTING

Exercise: What Goes Where?



27



PUTTING IT TOGETHER

Overall Summary



28



—
Thank you!

Melissa Benua
mbenua@gmail.com
@queenofcode
<http://www.queenofcode.net>

