BT8

Session
6/6/2013 2:15 PM

"How to Survive the Coming Test Automation Zombie Apocalypse"

Presented by:

Dale Emery
DHE

Brought to you by:

340 Corporate Way, Suite 300, Orange Park, FL 32073
888-268-8770 · 904-278-0524 · sqeinfo@sqe.com · www.sqe.com

Dale Emery

DHE

Since 1980, **Dale Emery** has worked in both IT organizations and software product development companies as a developer, manager, process steward, trainer, and consultant. He helps people apply the agile values of communication, feedback, simplicity, courage, and respect to software development. Dale's combination of deep technical expertise and extensive organizational development experience makes him particularly effective in working with software teams. In 2007 Dale received the Ward Cunningham Gentle Voice of Reason Award, which the Agile Alliance created to recognize Dale’s unique contribution to the agile community. Dale's personal mission is to help people create joy, value, and meaning in their work. Learn more about Dale at dhemery.com.

How to Survive the Coming Test Automation Zombie Apocalypse

Dale Emery
http://dhemery.com
@dhemery

Too gruesome to display

The Trouble With Zombies

Legion

Relentless

Infectious

Eat your brains

Usually not as pretty
as the one
on the previous slide

Test Automation Zombies

Legion

Relentless

Infectious

Eat your brains

Look just like ideas

Oddly attractive

The Trouble with Test Automation Zombies

Sometimes the best you can do

They are easy

Lull you into doing them by habit

Focus on the short term

At the expense of the (not-so-)long term

Pound Foolish Zombies

The Record and Playback Zombie

Appeal

Lots of tests quickly, (almost) for free

Danger

Recorded tests are
unexpressive, brittle, unmaintainable

Can't record tests
until the software is done

The Automation is Easy Zombie

Appeal

Inexpensive automators

Danger

Tests written by unskilled programmers are unexpressive, brittle, unmaintainable

Surviving Pound Foolish Zombies

Remember:

Test automation is real software development

Test automation takes time

Big cost is maintenance

Writing maintainable code is hard

More Surviving Pound Foolish Zombies

Stop record and playback

Refactor recorded tests to make them expressive, resilient, maintainable

Use real programmers...

with the experience to value maintainability...

and the programming skill
to write maintainable test code

Displacement Zombies

The Automate Last Zombie

Appeal

"Can't automate tests before system is written"

Danger

Feedback is delayed

The system is harder to test
(not designed for testability)

Missed opportunity
to use tests as guidance

The Dedicated Test Automator Zombie

Appeal

Keep developers focused on developing

Danger

Test automation lags development

Feedback is delayed

Typically uses less-skilled programmers

The Test Automation Group Zombie

Appeal

Keep entire development team focused on developing

Danger

Same as dedicated test automators

Organizational boundaries make delays even worse

Surviving Displacement Zombies

Create automatable examples
before development begins

Developers automate tests

Done includes test automation

Math Zombies

The Test Case Count Zombie

Appeal

Confidence that the system is tested

Danger

Focuses on ease of automating each test

Instead of on the value of the tests

The Code Coverage Zombie

Appeal

Confidence that the system is tested

Danger

Confidence is unwarranted
(executed does not mean tested)

Surviving Math Zombies

Focus on automating tests
that provide value

Use code coverage tools
only to identify what is not tested

Stealth Zombies

The Dedicated First Responder Zombie

Appeal

Relieves the development team from having to run, maintain, respond to tests

Danger

Delays feedback
(through first responders)

The Hide the Broken Feature Zombie

(Disable tests for known-broken features)

Appeal

Reduce distraction by known failures

Danger

Reduces urgency to fix the features

The Hide the Flaky Test Zombie

(Disable or ignore flaky tests)

Appeal

Reduce wasted effort diagnosing tests

Danger

Discards potentially useful test results

Reduces urgency to make tests more robust

The Automatic Rerun Zombie

Appeal

Reduce wasted effort diagnosing tests

Danger

Discards potentially useful test results

Reduces motivation to make tests more robust

Increases suite run time

Surviving Stealth Zombies

Fix flaky tests now

Fix broken features now

Bypass flaky technology

Analyze every failure

Include automated tests
in code promotion procedures

Reserve automatic reruns for exceptional cases

Hacker Zombies

The Fixed Wait Zombie

Thread.sleep(8000);

Appeal

Easy way to deal with variable response times

Danger

Fixed waits always increase

Test suite execution time
becomes maximally pessimistic

Surviving The Fixed Wait Zombie

Bypass slow, variable technologies

Poll for the relevant condition

Arrange to be notified
when the condition becomes satisfied

Create a utility for fixed waits

Instrument the utility
to gather information

The Dependency Chain Zombie

Appeal

Speed up test suites

Danger

Prevents running tests independently

Skips subsequent tests
even if they would provide value

Harder to understand
what responsibility each test is testing

Surviving The Dependency Chain Zombie

Make tests independent

Bypass slow technologies
for setup and verification

Let each test
establish its own preconditions

The Predefined Test Database Zombie

Appeal

Speed up test suites

Often already exists for manual testing

Danger

Hides essential details from test code

Harder to understand what responsibility each test is testing

Surviving The Predefined Test Database Zombie

Express every essential detail
directly in the text of the test code

Let each test
create its own data

Use the builder pattern
to create non-trivial test data

The Chewy GUI Zombie

(Testing only through the GUI)

Appeal

Available and accessible to automators

Tests whole system

Danger

Makes tests dependent on most volatile interface

Increases test execution time

Harder to diagnose failures

Surviving
The Chewy GUI Zombie

Bypass the GUI whenever it is
not essential
for the test

Now You Know How to Survive the Coming Test Automation Zombie Apocalypse

Dale Emery
http://dhemery.com
@dhemery