Test Session: Using ContextQA to test website with problematic identifiers

I wanted to check ContextQA against a website utilizing markup that makes it difficult to robustly test. Such problems as ambiguous control identifiers, lack of identifiers, and multiple siblings under the same parents. I wanted to see how this impacts ContextQA's recorder and self-healing capabilities, and especially wanted to try a common "fragile automation script" scenario by changing the website after a script was created.

Summary

A familiar scenario that haunts automation – scripting against a poorly architected web application that is still changing. ContextQA's recorded scripts and attempts at self-healing mostly handle the scenario well, but issues arose that created fragile scripts as the application morphed. Some of the issues are intrinsic to the problem, at least one seemed a bug in how ContextQA handles exceptions from its own libraries.

Other issues were visual, usability behaviors tangential to the main question in the use case.

Method

Using https://datamakerjs-f3b6b7d13de0.herokuapp.com/ record a session with ContextQA's Chrome extension. Try to record the presentation and interaction with graphical data on the page (represented as an SVG object, so items with the data are navigable). Run the recorded session, and then try to augment the resulting script to add verification steps to see if a viable automated check could be created.

After recording, modify the application and run the above again to see how robust the AI script might be.

Input data:

The web page has the following problematic content:

- Some of the page content is random by design
- Some of the page content, but textual and graphical, presents HTML elements that do not have an id tag. Some of the graphical objects that have id tags use a meaningless form such as "node1", "node2" that are difficult to relate back to the actual data structure which generates it.

Version of the DataMaker web page at the beginning of test, pre modification:

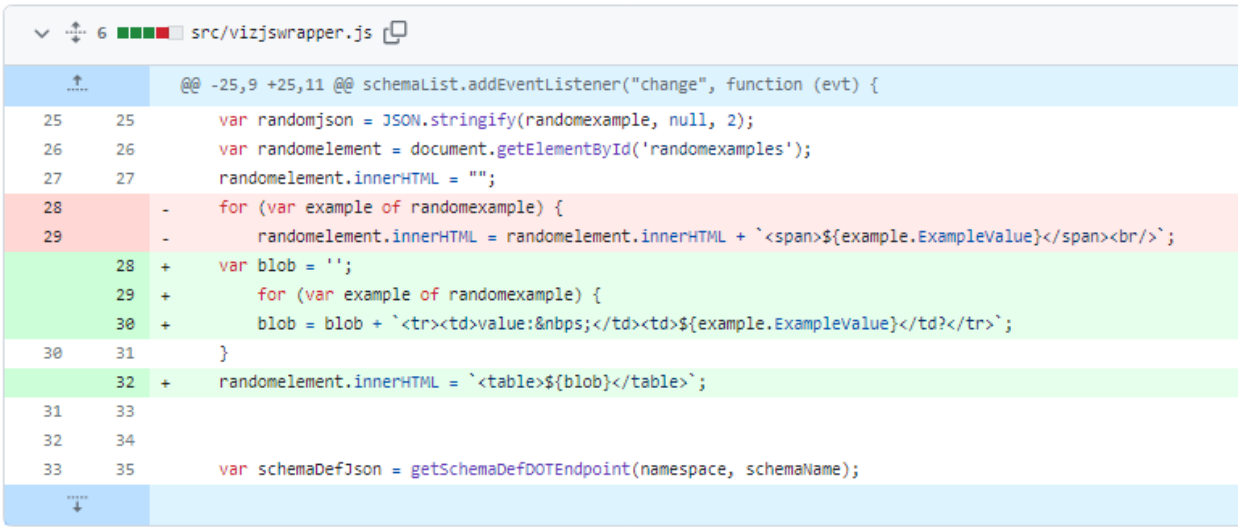
> The main page is "mainindex.html"

datatoolsjs/pages at b1b7ad9a5dec6099962e4b000fc7406510b74819 · WayneMRoseberry/datatoolsjs (github.com)

Final version of the DataMaker web page as per this test, post-modification:

https://github.com/WayneMRoseberry/datatoolsjs/tree/ceb2e052c262aee435cac0de26ad8550ffca3275/pages

The change is to the randomExamples rendering code, shown below. The change imitates a developer making visual changes to a page but having similar information with similar relationship as the user would perceive it.:



6 src/vizjswrapper.js

```
@@ -25,9 +25,11 @@ schemaList.addEventListener("change", function (evt) {
25 25      var randomjson = JSON.stringify(randomexample, null, 2);
26 26      var randomelement = document.getElementById('randomexamples');
27 27      randomelement.innerHTML = "";
28    -     for (var example of randomexample) {
29    -         randomelement.innerHTML = randomelement.innerHTML + `<span>${example.ExampleValue}</span><br/>`;
   28 +     var blob = '';
   29 +         for (var example of randomexample) {
   30 +         blob = blob + `<tr><td>value:&nbps;</td><td>${example.ExampleValue}</td?</tr>`;
30 31      }
   32 +     randomelement.innerHTML = `<table>${blob}</table>`;
31 33
32 34
33 35      var schemaDefJson = getSchemaDefDOTEndpoint(namespace, schemaName);
```

All activity in ContextQA was performed against the account Wayne_roseberry@hotmail.com under the wayneroseberry company name.

The workflow I created is located here:

https://wayneroseberry.contextqa.com/td/cases/34/steps

The results for the workflow are located here:

https://wayneroseberry.contextqa.com/td/cases/34/dry_runs

Session Notes:

Start time: 5:30

Preparing workspace, updating to most recent ContextQA chrome extension, getting signed in.

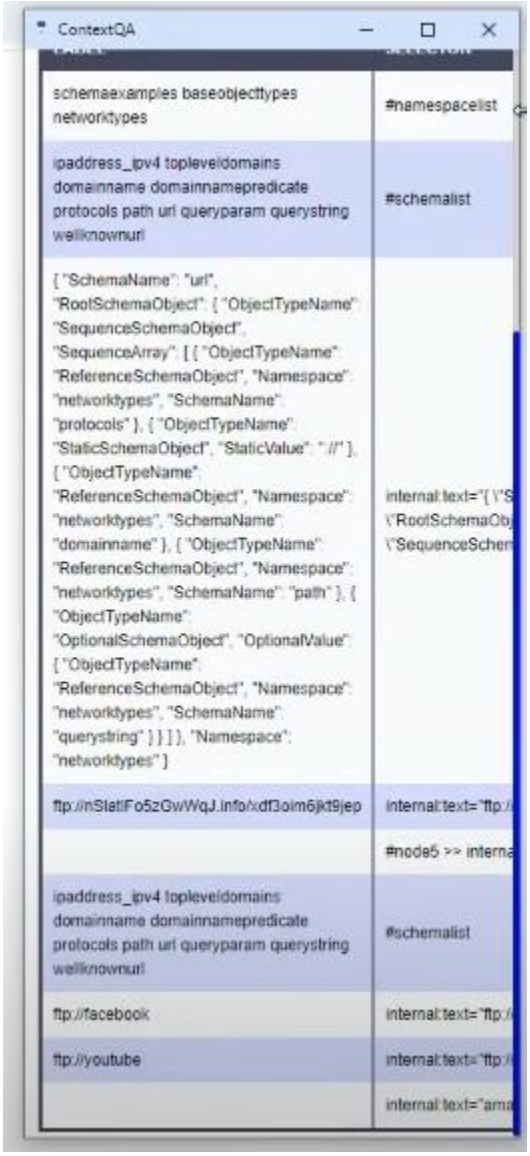
5:36 am – started writing this document.

5:54 am – started recording in ContextQA

Issue: 1 how to select workspace from ContextQA Chrome Extension? – resulting recording saved to a different workspace than I wanted. I created a new workspace before testing (https://wayneroseberry.contextqa.com/td/3/cases/filter/2) specifically to capture this session. It saved the test case to the first workspace (https://wayneroseberry.contextqa.com/td/1/cases/filter/2) in my list of workspace, although I don't know what it's true order of selection is. Feels out of my control.

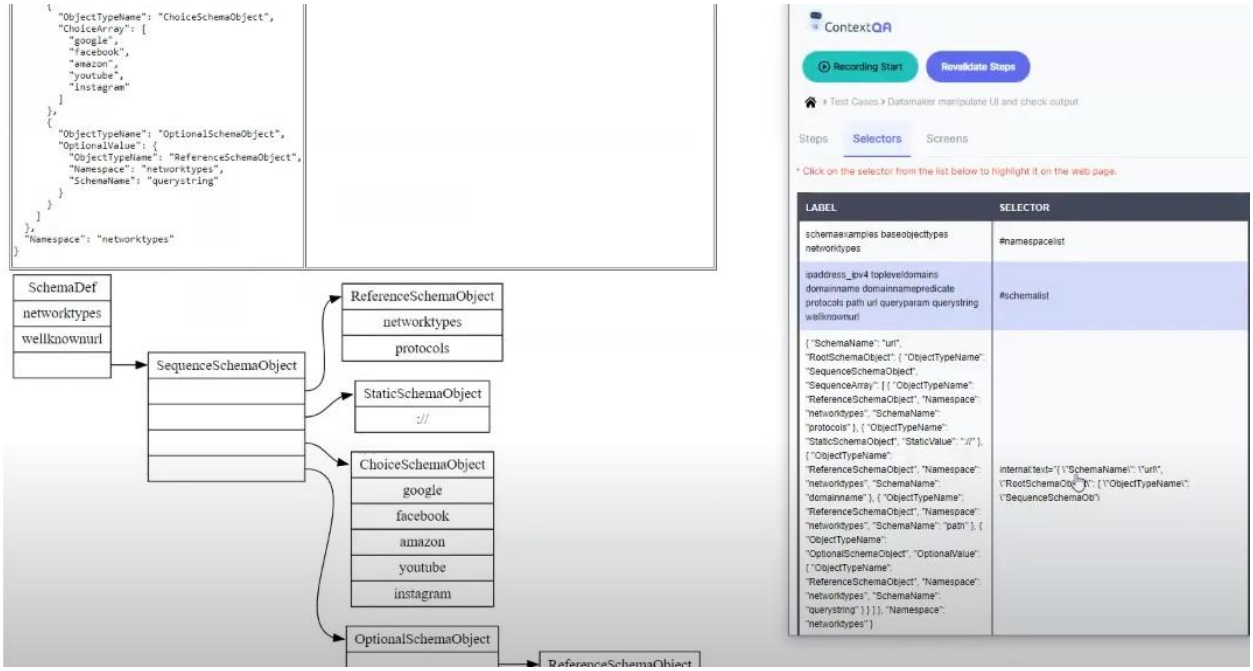
Issue: 2 ISSUE: ContextQA chrome extension screen is horizontally resizable, but the content inside (selectors tab) does not resize and does not present a scrollbar

In the screenshot below, I have resized the Selectors tab in the Chrome Extension, and the table does not resize with it, but also does not present a scroll bar on the bottom. Likewise, if I resize to make it wider, the table does not resize permitting easier viewing of the data on the page.



Issue: 3 ISSUE: ContextQA chrome extension, Selectors tab post recording, post save, "* Click on the selector from the list below to highlight it on the web page." – did not actually highlight the object on the web page when selected

Screenshot below from video of session, click on the selectors is not highlighting the relevant object on the page recorded from.



Issue: 4 Issue: the "details" link on recorded test case has narrow right column and visually hard to read

Test Step(4) Details

Title	Click on { "SchemaName": "url", "RootSchemaObject": { "ObjectTypeName": "SequenceSchemaObject", "SequenceArray": [{ "ObjectTypeName": "ReferenceSchemaObject", "Namespace": "networktypes", "SchemaName": "protocols" }, { "ObjectTypeName": "StaticSchemaObject", "StaticValue": "://" }, { "ObjectTypeName": "ReferenceSchemaObject", "Namespace": "networktypes", "SchemaName": "domainname" }, { "ObjectTypeName": "ReferenceSchemaObject", "Namespace": "networktypes", "SchemaName": "path" }, { "ObjectTypeName": "OptionalSchemaObject", "OptionalValue": { "ObjectTypeName": "ReferenceSchemaObject", "Namespace": "networktypes", "SchemaName": "querystring" } }] }, "Namespace": "networktypes" }
Metadata	
Specs	
Description	Click on { "SchemaName": "url", "RootSchemaObject": { "ObjectTypeName": "SequenceSchemaObject", "SequenceArray": [{ "ObjectTypeName": "ReferenceSchemaObject", "Namespace": "networktypes", "SchemaName": "protocols" }, { "ObjectTypeName": "StaticSchemaObject", "StaticValue": "://" }, { "ObjectTypeName": "ReferenceSchemaObject", "Namespace": "networktypes", "SchemaName": "domainname" }, { "ObjectTypeName": "ReferenceSchemaObject", "Namespace": "networktypes", "SchemaName": "path" }, { "ObjectTypeName": "OptionalSchemaObject", "OptionalValue": { "ObjectTypeName": "ReferenceSchemaObject", "Namespace": "networktypes", "SchemaName": "querystring" } }] }, "Namespace": "networktypes" }

Other PW Locator

- internal:text="{ \"SchemaName\": \"url\", \"RootSchemaObject\": { \"ObjectTypeName\": \"SequenceSchemaOb"i
- tbody > tr > td > #schemablob > pre
- text={ "SchemaName": "url", "RootSchemaObject": { "ObjectTypeName": "SequenceSchemaObject", "SequenceArray": [{ "ObjectTypeName": "ReferenceSchemaObject", "Namespace": "networktypes", "SchemaName": "protocols" }, { "ObjectTypeName": "StaticSchemaObject", "StaticValue": "://" }, { "ObjectTypeName": "ReferenceSchemaObject", "Namespace": "networktypes", "SchemaName": "domainname" }, { "ObjectTypeName": "ReferenceSchemaObject", "Namespace": "networktypes", "SchemaName": "path" }, { "ObjectTypeName": "OptionalSchemaObject", "OptionalValue": { "ObjectTypeName": "ReferenceSchemaObject", "Namespace": "networktypes", "SchemaName": "querystring" } }] }, "Namespace": "networktypes" }

Pause: 6:27 am

Return: 8:16 am

Issue: 5 ISSUE: step name is occluding screenshot on very long step name

In the step below, the screen shot is on the left, underneath the name of the element that was auto-generated by the recorder.

Issue: 6 ISSUE: edits on steps are lost if enter text into “Search” box

- Select a step, click edit pencil icon
- Change the action to something other than what it is
- In “Search” box, type some text which does not match item…
- Clear search box

RESULT: edits which were pending “update” are now lost – NOTE: this seems to generalize with anything that goes away from the editor view… edits lost, no prompt

Issue: 7 ISSUE: text typed into “Search” box does not respect unsaved changes to steps

- Select a step (use one that is not “Press Enter”, click edit pencil icon
- Change the action to “Press Enter”
- In “Search” box, type “press”

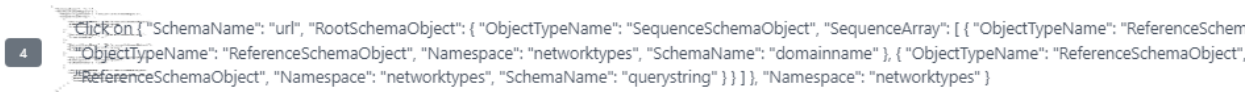
RESULT: empty search list… or.. at least the one you were editing is not in the view

EXPECTED: the one I was editing, because “Press” is in the control now… it isn’t made clear what the search box is searching on – current state, or last saved state – and the edits on the control are gone now.

Issue: 8 How do I create an element based on the recorded session?

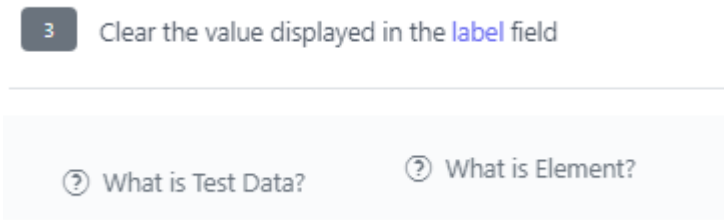
This might be something I am missing, but assuming the Chrome Extension recorded a session, and that some of the identifiers/locators it constructs are useful, how do I create an element in ContextQA based on that recorded element in the step? I may want to reuse that element for other workflows, but the only thing I could find was creating a new element from scratch.

Issue: 9 the vertical gap when you click a step and hit “Edit” seems excessive



Issue: 10 The help text available does not match content of the task selected – just hard coded to "test data" and "element" – also the layout is vertically misaligned

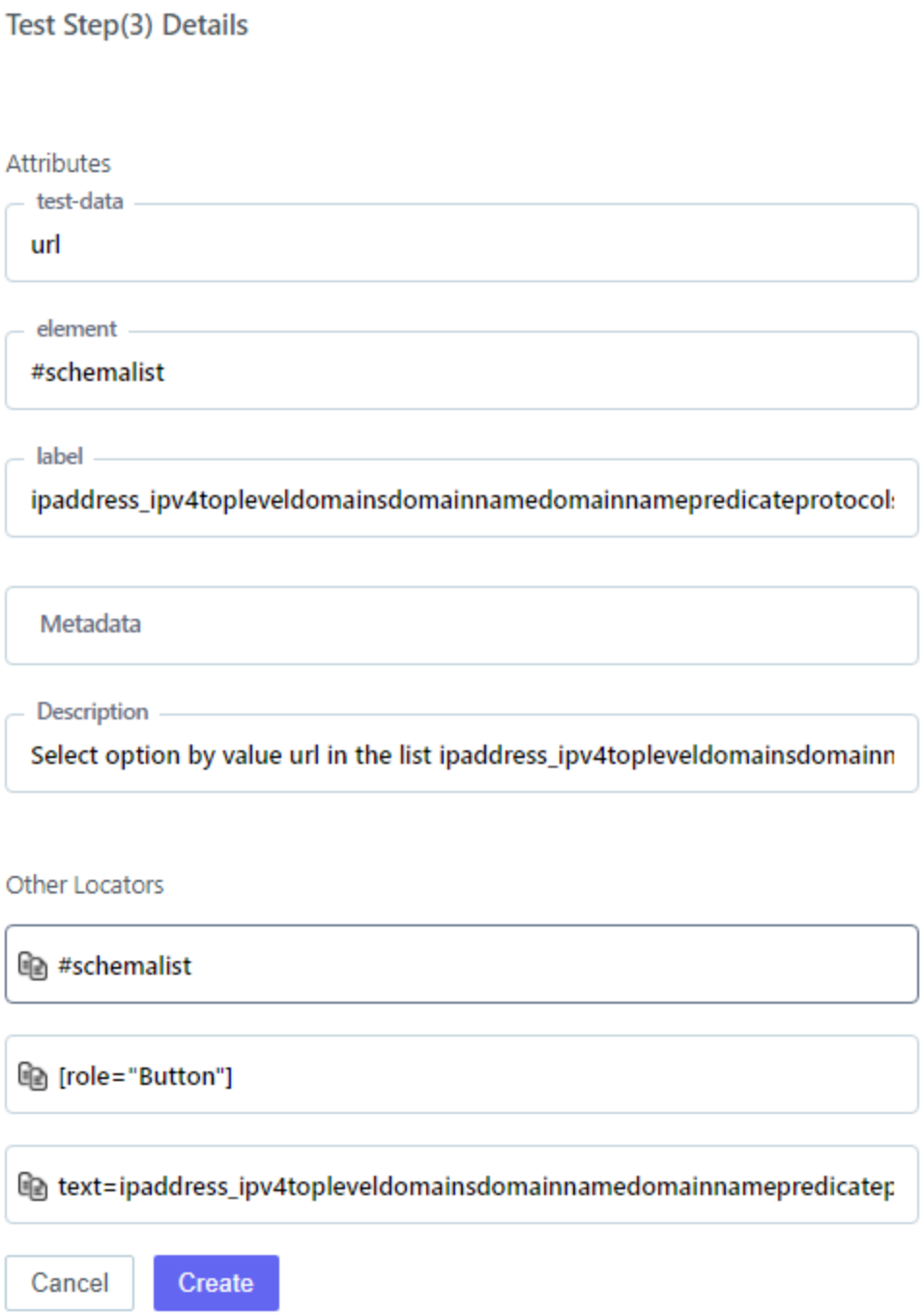
In the example below, the step template I have selected has a "label" field, but the help only addresses "Test Data" and "Element"



Issue: 11 "Other locators" on recorded step details are read-only – what if I want to remove one?

The items below "Other locators" on the screenshot below do not respond to attempts to edit them. They do not have a delete option. What if I determine they are problematic and want to redefine them? What if I want to add more robust locators to the list?

As my experience has shown during this test, and as you probably know far better than I do, recording only gets a person started, and trying to create robust automation against a changing product is challenging. It seems to me that tweaking and adjusting scripts and their locator logic is a core scenario, yet as far as I can tell it is not permitted.



Issue: 12 ISSUE: changed a recorded step to use a different selector (select 19th child or #randomexamples instead of ftp://facebook) – but it found ftp://facebook in a "GetByText()" call where the full string was much longer (same string with random query string appended) – not guaranteed to be the 19th child of the element named. The fallback self-healing winds up being a bit random

This is on the following run:
https://wayneroseberry.contextqa.com/td/test_case_results/47/step_results/397

The step in question is #8, “Click on click on 19th child”

The “Golden HTML (Before Code)” is this: `<span>ftp://facebook</span>`

The “Difference HTML (After Code)” is this:
`<span>ftp://facebook?rt3lUjBxTi1fmu=htVze5x9pjlC</span>`

The sequence of location checks it tried are the following:

> checking visibility of getByText('ftp://facebook')
>
> checking visibility of locator('tbody > tr > td > #randomexamples > span:nth-child(19)')
>
> checking visibility of locator('text=ftp://facebook')
>
> waiting for getByText('ftp://facebook')
>
> 2mslocator resolved to ftp://facebook?rt3lUjBxTi1fmu=htVze5x9pjlC
>
> 0msattempting click action
>
> 28mswaiting for element to be visible, enabled and stable
>
> 0mselement is visible, enabled and stable
>
> 2msscrolling into view if needed
>
> 7msdone scrolling
>
> 2msperforming click action
>
> 0msclick action done
>
> 0mswaiting for scheduled navigations to finish
>
> 0msnavigations have finished

This seems to have created a misleading behavior, because the thing it matched against is certainly different than what it was recorded against.

This scenario is one where as the end user I recorded something that isn’t fair to match against, as the application creates random data, but it still seems problematic to use a matching technique which can fallback to partial matches. I feel like there are a lot of other ways the above could miss the desired control because something else matched on the page.

Pause: 8:55 am

Return: 10:00 am

Modified Datamaker to change layout of RandomExamples table

Issue: 13 playback of “Page.Isvisible(locator('text=ftp://nSlatlFo5zGwWqJ.info/xdf3oim6jkt9jep'))” yielded error

The run in question is here: https://wayneroseberry.contextqa.com/td/test_case_results/48

The failed step is this one:
https://wayneroseberry.contextqa.com/td/test_case_results/48/step_results/404

The error generated feels as if ContextQA has lost control of what is going on. It appears selector it used caused the underlying libraries to fault processing the return value. On the one hand, I would not expect the script to succeed, because the script is poorly constructed and the web application changed in a way that broke the fragile script. But as a testing tool, I expect ContextQA to handle its exceptions in its utilities in a way that helps the script author more.

"Cannot set properties of null (setting 'innerText')

TypeError: Cannot set properties of null (setting 'innerText') at HTMLSelectElement.<anonymous> (https://datamakerjs-f3b6b7d13de0.herokuapp.com/pages/vizjswrapper.js:2:1365342)
"

Also... "locator:Error: This model's maximum context length is 4097 tokens, however you requested 4586 tokens (4486 in your prompt; 100 for the completion). Please reduce your prompt; or completion length."

Issue: 14 The credibility of ContextQA's claims seem threatened by the experience demonstrated in this scenario test.

From the ContextQA website:

> *"ContextQA's user-friendly automation platform empowers your team, regardless of coding skills, to expand test coverage with ease, even in the cloud or CI/CD pipelines."*

ContextQA promotes the notion that even with little to no coding skills, team members will be able to build automation suitable for CI/CD pipelines.

The trouble with this claim is that the problem of creating non-fragile automation demands coding expertise and algorithmic thinking. People need to understand how locators work to recognize why a locator failed or caused a behavior different than anticipated, or why seemingly cosmetic changes to a web page break the scripts that try to utilize the page. As demonstrated here, a script can suddenly start selecting a different element than it was built to select, or the entire script can become confused and abort, even if the page functionality is essentially the same.

The self-healing behaviors will sometimes exacerbate such problems, as the attempt to workaround poor web-application page architecture can create the very confusion one is trying to avoid.

What follows is just my opinion, but to me it seems there is a far more credible story if ContextQA would drop the "regardless of coding skills" aspect of their pitch and stick to the idea that automation engineers are engaging in a skilled discipline. What ContextQA provides is a set of tools that are more powerful, easier to use, but do not diminish the skill and expertise. You are making great engineers better, not tossing them to the curb in a race to drive the skilled labor market to the

floor. Sell excellence, sell pride in craftsmanship. It is a story you can sell without contradicting it via a problem that winds up being intractable.

Issue: 15 Is there a way ContextQA could suggest or promote better web application design?

The most powerful step to stabilize testing of a web application is to make the web application intrinsically testable. Better markup, better information architecture, better availability of control points.

What an engineer might not know is what to do to make the web application more testable.

It seems ContextQA is in a position to guide developers and test engineers to better architecture. Is there a gamification feature? Some kind of rating? A reliability risk assessment? Your customers are running scripts against your engine. It seems you could measure flake rates, failure rates, regression rates and categorize them against patterns in the script, locator strategy, web application structure. It seems like you have opportunities to inject ideas to the product team to guide them to building a better application that will run much more smoothly.