Supporting Multiple Versions of DANY with GOPHER – Option 3

Issue – Support multiple versions of DANY with GOPHER, for example Production vs QA. Currently the only way we can do this (Option 1) is to run a separate instance of GOPHER for each version of DANY.

This option would require duplicate test cases to be maintained for DANY versions where the features to be tested do not change. Hopefully the number of changes per version would be small creating a situation where most of the test cases would be duplicated between versions.

One simple approach to address this issue would be to change the framework such that each test case would be supplied a DANY website URL instead of having one URL per GOPHER instance.

Advantages:

* Allows multiple websites to be tested within one GOPHER instance, i.e. Production, QA etc.
* Allows flexibility in organizing test suites – by feature, version or combination.
* Allows test case reuse between DANY versions. Where DANY features vary between versions separate test cases would be developed.
* Better support for parallel runs of test cases (see how too below.)

Disadvantages:

* Not in harmony with original concept of framework – one application under test per GOPHER instance.
* Concept needs prototype to make sure there are no show-stopping issues with Selenium RemoteWebDriver .go (URL) (see how too below.)

How to:

* For each ‘TestCase’.xls add a new column ‘URL.’
* For each row in ‘TestCase’.xls add a value for URL.
* Change the DataProvider for each test case to supply this value to the test case along with platform, browser and test data.
* Create a factory utility to provide a RemoteWebDriver instance to GOPHER. With a factory, we can provide the most efficient pattern of reuse for RemoteWebDriver – one driver instance per GOPHER , one instance per URL or a new instance per test case. If we consider parallel test cases per GOPHER, it may be better to have multiple drivers.
* Have each test case call the factory (perhaps from the base class) to get its driver instance based on the URL.
* Remove single URL from config.properties and existing code to create driver.
* As it is now, each test case would have its own .xls file. Each run of a test case would be driven by a line in its .xls file.