



# SOA Testing with AxeHarness

## THE SOA TESTING CHALLENGE

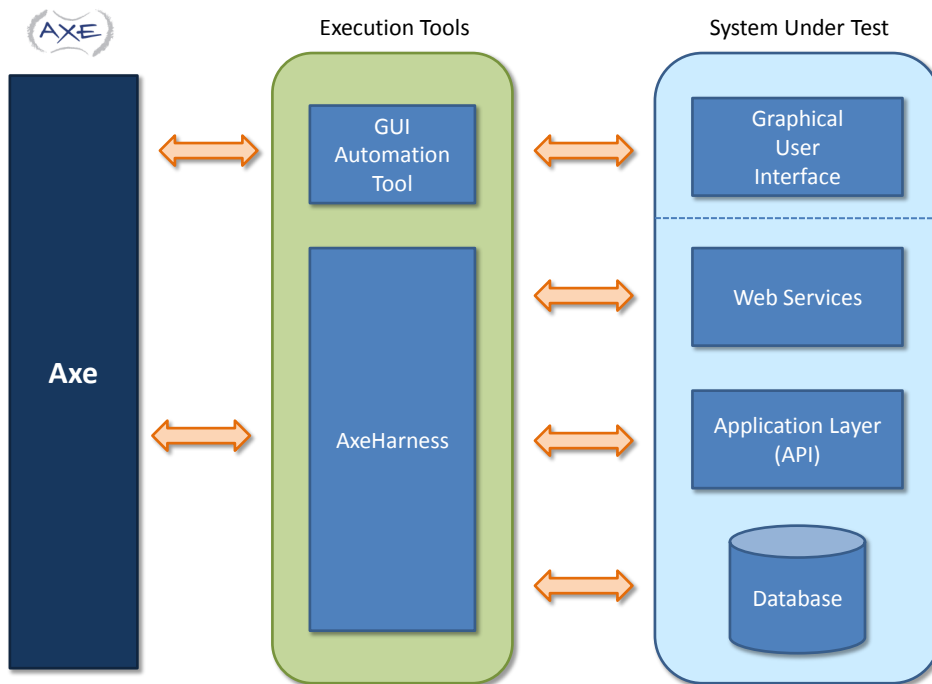
Advances in IT Infrastructure architectures have not made the tester's life easy. The adoption throughout the industry of SOA (Service Orientated Architectures) has meant that traditional testing via GUI based applications now needs a different level of testing, different test tools and languages. SOA based architectures are not "human friendly" but why should they be? The end-user of most SOA components is other software components. Messages are sent in low-level protocols, such as SOAP & XML, between Applications, so how exactly do you test something that has a complicated application architecture?

## ABOUT AXEHARNES

At Odin, we realise that two levels of testing, the UI level and the SOA level need to be tested either together or separately as required.

At the lower "depths" testing becomes more complex as web services and databases may also need to be tested and validated. Overall the business requires that the end-to-end business process is tested, this obviously needs a common holistic approach which Odin provides.

So how do you test this complex architecture without Testers developing code? (which once again needs to be tested!)



AxeHarness is an automation tool for executing tests against non-UI application interfaces and SOA applications. It has the following features:

- It can be used independently or in conjunction with UI automation tools.
- Test Scenarios and data are defined in Axe Spreadsheets.
- Component Interfaces are described in Axe Object Maps.

AxeHarness is consistent with the standard AXE UI interface in that it uses Excel spreadsheet to define tests. Unlike UI testing AxeHarness carries out the execution itself without the reliance on other vendor's tools to execute tests. This allows testers to be conversant with the Axe toolset irrespective of whether they are testing GUI or SOA interfaces to the application.