Gold Test Automation Framework
==========================

Release v0.0.1 - 08/06/2009

This software constitutes of the Gold Test Autmation Framework - gtaf. The goal of gtaf is to provide a clean Object Oriented Design of Domain Specific Language for easy and readable automation of products based on Gold platform. To meet this end, gtaf makes use of Selenium as the underlying test automation tool and FitNesse/Fit as the underlying test description and management interface.

Design Rationale
============

The framework follows a strict object oriented design. Each of the Gold platform based products will extend the `BaseTestCase` which in turn extends the `SeleneseTestCase`. The `BaseTestCase` encapsulates all tasks that need to be done to setup the execution of a test case on the Gold platform. The product specific test case e.g. `PowerSearchTestCase` encapsulates everything that is specific to that product, e.g. product URLs, opening of default product page/sub-page etc. The product test cases in turn are extended by specific module test cases e.g. `AdvancedSearchTestCase`. The module test cases in turn make use page specific UI Locators and module specific Action Dispatchers. The UI Locators, e.g. `AdvanedSearchUiLocator` encapsulate the UI objects on the pages and hence make the framework de-coupled with changes on the UI end (renaming of controls, position changes of input boxes etc.). The Action Dispatcher, e.g. `AdvanedSearchActionDispatcher` encapsulate all the actions that can be carried out in that module, e.g. Search. Other than this module specific objects such as `SearchResult` encapsulate data that is relevant to the particular module.

The framework can be driven either through JUnit tests or FitNesse/Fit fixtures. The JUnit tests are written in the tests package. The `framework.fixtures` package contains the Do fixture code that helps in writing tests in simple English like use cases.

Please refer to index.html for the complete javadoc and class diagram.

Folder Structure
============

```
gtaf ---+
     |
     |- build +
     |     |- lib
     |     |- classes
     |
     |- doc
     |- src +
            |- framework
            |- tests
```

`/gtaf/build/lib` - contains the JARs that gtaf is dependent on
`/gtaf/build/classes` - contains compiled class files
`/gtaf/doc` - cotains the complete java doc
`/gtaf/src/framework` - contains the framework java code
`/gtaf/src/tests` - contains JUnit tests that make use of the framework

Importing Project into Eclipse
=====================

1. File > Import and select General > Existing Projects into Workspace
2. Choose the gtaf root directory to import the complete project into workspace

Description of Demo Test
===================

`com.qait.gold.tests.powersearch.search.advanced.SimpleAuthorSearchTestCase` is a simple *JUnit* test that demonstrates the use of the gtaf framework. Note that gtaf can be driven through JUnit or FitNesse.

- This test case opens the power search default page
- Navigates to the advanced search page
- Searches on Author for author William
- Verifies that search results are returned
- Opens the first result in detail view
- Verifies that William appears in author field

Steps to run the Demo Test
====================

Software needed to run this demo:

- jre 1.6
- Selenium RC
- Firefox

1. Start the Selenium RC Server on localhost and default port 4444. Ensure that the server starts successfully
2. Open a command prompt and navigate to the gtaf project root
3. Run the testSimpleAuthorSearch.bat (present in root folder)

Note - The test by default runs on *Firefox* browser and would require one on your system. The test host, port, browser etc will be configurable once integrated with FitNesse