

## The Great Mind Challenge - Project Scenario Template

*Note: Already filled information should not be changed*

1.	Name of the Project	Detection of duplicate test cases/scripts of a test suite
2.	Objective/ Vision	<ol style="list-style-type: none"> <li>1. In an organization, the testing teams create lot of test cases/ scripts and over the years and many of the scripts become duplicate. Testing tools such as Rational Functional Tester (RFT) creates test scripts and assets for playing back the scripts. The objective of this project idea is to find out the probable duplicate scripts.</li> <li>2. A report should be generated to report duplicate scripts.</li> <li>3. A report should be generated to report probable script that can be merged.</li> </ol>
3.	Users of the System	A. Any customers who used testing tools such as RFT to test their application
4.	Functional Requirements (Atleast Eight)	<ol style="list-style-type: none"> <li>i. There should be a mechanism to find out duplicate test cases/ scripts.</li> <li>ii. There should be mechanism to find out similar test scripts, but not exactly same.</li> <li>iii. A filter criteria for comparison</li> <li>iv. A comparator to show the difference for similar test scripts.</li> <li>v. Optionally provide a manual merge for similar test script from comparator window.</li> <li>vi. The idea can be generic to any testing tools, but optional can be implemented for RFT.</li> <li>vii. Testing tools can be installed on the machine and recorded scripts and assets can be used for implementing the algorithm if required.</li> <li>viii. There should be report that can be viewed from tools IDE or from a browser.</li> </ol>
5.	Non-functional requirements (Atleast Four)	<ol style="list-style-type: none"> <li>i. Secure access of confidential data (user's details). SSL can be used.</li> <li>ii. 24 X 7 availability</li> <li>iii. Better component design to get better performance at peak time</li> <li>iv. Flexible service based architecture will be highly desirable for future extension</li> </ol>
6.	Optional features	a. Help page for user

7.	User interface priorities	<p>A. Professional look and feel</p> <p>B. Use of AJAX at least with all registration forms</p> <p>C. Browser testing and support for IE, Opera, Mozilla, and Firefox.</p> <p>D. Use of Graphical tool like JASPER to show strategic data to admin</p> <p>E. Reports exportable in .XLS, .PDF or any other desirable format</p>
8.	Reports	<p>1) Report scripts that are exactly matching</p> <p>2) Report scripts that are not exactly matching but can be merged to create one script.</p> <p>3) Comparator for comparing similar scripts.</p>
	Other important issues	
9.	Team Size	2 – 4 members
10.	Technologies to be used	Java, Dojo (for GUI for browser front end), UML, J2EE, XML, e-Forms, AJAX, Web 2.0, Web-services, SOA
11.	Tools to be Used	<ul style="list-style-type: none"> <li>• Rational Software Architect – For Design</li> <li>• Rational Team Concert – for source code management</li> <li>• Rational Quality Manager – for test management.</li> <li>• ROSE/RSA / WebSphere Modeler</li> <li>• Eclipse/ RAD / Lotus Forms Designer / Portlet Factory</li> <li>• WebSphere Portal/ WAS/ WAS CE / WPS</li> <li>• DB2 Express – 'C' or DB2 UDB</li> <li>• Tivoli CDP/TSM / Tivoli Directory Server</li> <li>• Linux will be the preferred OS.</li> </ul>
12.	Final Deliverable must include	<p>A. Help documentation for users.</p> <p>B. Application archive with source code</p> <p>C. Complete Source code</p>