

1: Learn SAP Testing: Create your First SAP Test Case

Since test scenarios offer little information about how to complete the testing, they offer the maximum amount of flexibility to the tester responsible for them. Like test cases, the flexibility that comes with using test scenarios creates similar benefits and drawbacks.

Visual Studio Load tests contain scenarios, which contain Web performance tests or unit tests. A scenario is the container within a load test where you specify load pattern, test mix, browser mix, and network mix. Scenarios are important because they give you flexibility in configuring test characteristics that allow for simulation of complex, realistic workloads. For example, you might be testing an e-commerce site that has an Internet front-end used by hundreds of concurrent customers coming in over many connection speeds and using different browsers. The same site might also have an administration function that is used by internal employees to update products and to view statistics. These internal users would typically access the site by using the same browser and a high-speed LAN connection. You would want to encapsulate the properties of these two different groups of users in different scenarios. Each scenario can contain a virtual user type. In this case, a load test scenario can be made to represent virtual customers and another scenario can be made to represent virtual internal users of a Web site. The Load Test Editor lets you modify the properties of an existing scenario or add new scenarios to the load test. Additionally, you can also add more counter sets and run settings. Scenarios contain the following components: Load Pattern Specifies the number of virtual users active during a load test and the rate at which new users are started. Test Mix Model Specifies the probability of a virtual user running a given test in a load test scenario. The test mix model should reflect the objectives of your test for a particular scenario. Test Mix The test mix is the selection of Web performance and unit tests that are contained within the scenario and the distribution of those tests within the scenario. Network Mix Simulates that virtual users examine a Web site through a variety of network connections. These offer you a set of predefined and important counter sets for your load test. You manage your counters in the Load Test Editor. Load tests provide named counter sets that are useful when you analyze performance counter data. Counter sets can be selected for the computers that you specify for inclusion in the load test scenario. Load Test Scenarios In addition, a scenario has several other properties that you can edit by using the Load Test Editor. For more information, see Load Test Scenario Properties. Tasks Associated Topics Create and edit load test scenarios: Load tests contain one or more scenarios, which are used to model how a group of users interacts with a server application. An individual scenario consists of a load pattern, a test mix, a browser mix, and a network mix. Each of these settings corresponds to a page in the New Load Test Wizard. See Create and run a load test. Think times are used to simulate human behavior that causes people to wait between interactions with a Web site. Think times occur between requests in a Web performance test and between test iterations in a load test scenario. Using think times in a load test can be useful in creating more accurate load simulations. You can configure the load pattern properties to specify how the simulated user load is adjusted during a load test. You get three built-in load patterns: You choose the load pattern and adjust the properties to appropriate levels for your load test goals. You can use the test mix, which specifies the probability of a virtual user running a given test in a load test scenario. This lets you simulate load more realistically. Instead of having just one workflow through your applications, you can have several workflows, which is a closer approximation of how end-users interact with your applications. You can add or remove a Web performance or unit test from a load test in a scenario. A load test contains one or more scenarios, each of which contains one or more Web performance or unit tests. Using the network mix, you can simulate network load more realistically in a load test scenario. Load is generated by using a heterogeneous mix of network types instead of one single network type. You create a closer approximation of how end-users interact with your applications. The network mix model should reflect the objectives of that scenario. Using the browser mix, you can simulate Web load more realistically in a load test scenario. Load is generated by using a heterogeneous mix of browsers instead of one single browser. You create a closer approximation of the browsers that will be used with your applications. You can edit a load test scenario to configure test iteration

settings using the Load Test Editor and the Properties window. By default, a scenario is set up with no maximum test iterations. You can optionally configure the maximum number of iterations in the scenario and how long to pause between them. Using the Load Test Editor and the Properties window, you can specify a delay before starting a scenario in a load test. An example of when you might want to use the Delay Start Time property is if you need one scenario to start producing items that another scenario consumes. You can delay the consuming scenario to enable the producing scenario to populate some data. After you create a load test, you can edit the properties of your load test scenario to indicate which test agents you want to include.

2: Setting Up a Scenario in HP ALM – 1 | Intrepid Testing

Performance Testing Requires the Right Set of Tools As you can see, there are a number of tests you can run to ensure the performance of your API. While load testing is essential, your performance strategy should extend throughout the lifecycle of your API.

How do I monitor the system under load? Introducing the LoadRunner Controller Load testing means testing your system under typical working conditions. For example, you may test the system while many travel agents simultaneously reserve flights on the same flight reservation system. You design the test to emulate real-life situations. To do this, you need to be able to generate a load on an application and schedule when the load is applied because users do not log on and off the system at precisely the same time. You also need to emulate different types of user activity and behavior. For example, some users may use Firefox to access the system, whereas other users use Internet Explorer. Users may also employ different network connections to access the system, such as modem, DSL, or cable. You create and save these settings in a scenario. The Controller provides you with all the tools you need to help you build and run tests to accurately emulate your working environment. How do I start the Controller? To begin developing a scenario, you open the LoadRunner Controller. Select a scenario type. There are two scenario types: A Manual Scenario gives you control over the number of running Vusers and the times at which they run, and lets you test how many Vusers your application can run simultaneously. You can use Percentage Mode to distribute the total number of Vusers between the scripts based on a percentage specified by your business analyst. The Percentage Mode check box is selected by default the first time you start LoadRunner following installation. If it is selected, clear it. A Goal-Oriented Scenario is used to determine if your system can achieve a particular goal. Add a Vuser script to the load test. In this tutorial, you will use only one Vuser script to model a single group of users performing identical actions. To more accurately emulate a real-world scenario with more versatile user profiles, you would create a number of different Vuser groups, with each group running several scripts with different user settings. The script that you previously recorded in VuGen contains the business processes that you want to test. They include logging on, searching for a flight, buying a ticket, checking the flight itinerary, and then logging off the site. You will add a similar script to the scenario, and configure the scenario to emulate eight travel agents simultaneously performing these actions on the flight reservation system. You will add two more Vusers during the test. For this purpose, a sample script is provided that is similar to the one you created. We recommend that you use the sample script. In the New Scenario dialog box, click the Browse button. The LoadRunner Controller opens and displays the Design tab of your new scenario. Note that the control for the Design tab is in the lower left corner of the Controller. The Controller at a Glance The Design tab of the Controller is the main interface for designing your load test. The Design tab is divided into three panes: You configure the Vuser groups in the Scenario Scripts pane. You create different groups to represent typical users of your system and specify the number of Vusers that will run, and the machine that they will run on. Service Level Agreement pane. When you design a load test scenario, you can define goals or SLAs Service Level Agreements for the performance metrics. When you run the scenario, LoadRunner gathers and stores performance-related data. In the Scenario Schedule pane, you set the load behavior to accurately portray real-world user behavior. You define actions according to which the Vusers will run, the rates at which load is applied to the application, the load test duration, and how the load is terminated. How do I modify the script details? You modify the script details as follows: Change the Group Name. The Group Information dialog box opens. The new name is displayed in the Scenario Groups pane of the Design tab. How do I generate a load on the system? After you have added your Vuser scripts to the scenario, you configure the load generators, the computers that generate the load on the system. A load generator is a computer that runs multiple Vusers in order to generate a load on the system. You can use a number of load generators, each generator hosting multiple Vusers. In this section, you will learn about adding load generators to the scenario, and testing the load generator connections. Add a load generator. Click the Load Generators button on the Controller toolbar. The Load Generators dialog box opens. The Load Generators dialog box enables you to view and configure

the load generators that are defined in the scenario. The Load Generators dialog box shows details for the load generator called localhost. The status of the localhost load generator is Down. This indicates that the Controller is not connected to the localhost load generator. In this tutorial, you will use your local computer as the load generator. Note that in a typical operational system, you would have several load generators, each hosting multiple Vusers. Test the load generator connection. When you run a scenario, the Controller connects to the load generators automatically. However, you can test the connections before trying to run the scenario. In the Load Generators dialog box, select localhost and click Connect. The Controller attempts to connect to the load generator machine. When a connection has been made, the Status of the load generator changes from Down to Ready. How do I emulate real load behavior? After you have added your load generators, you are ready to configure load behavior. Typical users do not log on and off the system at precisely the same time. LoadRunner allows users to gradually log on to and off the system. It also lets you determine the duration of the scenario, and the way in which the scenario terminates. The scenario that you will configure below will be relatively simple. However, when designing a scenario that more accurately reflects a real life scenario, you can define more true-to-life Vuser activity. You configure the load behavior for a manual scenario in the Scenario Schedule pane of the Controller. The Scenario Schedule pane is divided into three sections: You will now change the default load settings and configure a scenario schedule. Select schedule type and run mode. In the Scenario Schedule pane, make sure that Schedule by: Scenario and Run Mode: Real-world schedule are selected. Set the scheduled action definitions. You can set the Start Vusers, Duration, and Stop Vusers actions for the scenario schedules from either the Global Schedule grid, or by manipulating the Interactive Schedule graph. When you set the definitions from the graph, the Global Schedule grid properties adjust accordingly. You will now set the definitions so that the Global Schedule grid will look as below. Set up the Vuser Initialization. Depending on your system configuration, initializing Vusers before they start running may provide more realistic results. Double-click Initialize in the Global Schedule grid. The Edit Action dialog box opens displaying the Initialize action. Select Initialize all Vusers simultaneously. Specify a gradual start for the Vusers Starting Vusers at intervals allows you to examine the gradual increase of Vuser load on the application under test over time, and helps you pinpoint exactly when the system response time slows down. Double-click Start Vusers in the Global Schedule grid. The Edit Action dialog box opens displaying the Start Vusers action. In the Start X Vusers box, enter 8, and select the second option "2 Vusers every You specify a duration to make sure that the Vusers continue performing the schedule action for a specific period so you can measure continuous load on the server. In the Interactive Schedule Graph, click the horizontal line which represents Duration. The line is highlighted and a dot and a diamond are displayed at its endpoints. Note that if the Legend is displayed on top of the diamond, click the Hide Legend button on the Interactive Schedule Graph toolbar to show the diamond. Drag the diamond shaped endpoint to the right until the time in brackets reads You have just set the Vusers to run for a period of 10 minutes. Schedule a gradual closure.

3: Lesson 5: Creating a Load Testing Scenario

A scenario test is motivating if a stakeholder with influence wants the program to pass the test. A dry recital of steps to replicate a problem doesn't provide information that stirs emotions in.

If you are using an earlier version, some or all of the information does not apply. When writing tests for your application it is often desirable to avoid hitting the database. Entity Framework allows you to achieve this by creating a context with behavior defined by your tests that makes use of in-memory data. Options for creating test doubles There are two different approaches that can be used to create an in-memory version of your context. Create your own test doubles This approach involves writing your own in-memory implementation of your context and DbSet. This gives you a lot of control over how the classes behave but can involve writing and owning a reasonable amount of code. Use a mocking framework to create test doubles Using a mocking framework such as Moq you can have the in-memory implementations of you context and sets created dynamically at runtime for you. This article will deal with using a mocking framework. To demonstrate using EF with a mocking framework we are going to use Moq. The easiest way to get Moq is to install the Moq package from NuGet. Limitations of EF in-memory test doubles In-memory test doubles can be a good way to provide unit test level coverage of bits of your application that use EF. One example of such a difference is loading related data. If you create a series of Blogs that each have related Posts, then when using in-memory data the related Posts will always be loaded for each Blog. However, when running against a database the data will only be loaded if you use the Include method. For this reason, it is recommended to always include some level of end-to-end testing in addition to your unit tests to ensure your application works correctly against a database. Following along with this article This article gives complete code listings that you can copy into Visual Studio to follow along if you wish. This will allow the mocking framework to derive from our context and overriding these properties with a mocked implementation. If you are using Code First then you can edit your classes directly. Name select b; return query. Name select b; return await query. The following test uses Moq to create a context. Next, the context is used to create a new BlogService which is then used to create a new blog using the AddBlog method. Finally, the test verifies that the service added a new Blog and called SaveChanges on the context. UnitTesting; using Moq; using System. We can then create a BlogService based on our test doubles and ensure that the data we get back from GetAllBlogs is ordered by name. For more details see <http://> Whilst the async methods are only supported when running against an EF query, you may want to use them in your unit test when running against an in-memory test double of a DbSet. In order to use the async methods we need to create an in-memory DbAsyncQueryProvider to process the async query. Whilst it would be possible to setup a query provider using Moq, it is much easier to create a test double implementation in code. The code for this implementation is as follows:

4: How to install the SIPp testing tool on Ubuntu Server - TechRepublic

After moving your event to Test Mode, you can create unique testing scenarios to quickly evaluate different aspects of your www.enganchecubano.com the invitation that exhibitors receive, view your website through the eyes of a speaker, or experience registration as someone not in your Address Book.

Principle[edit] Scenario-building is designed to allow improved decision-making by allowing deep consideration of outcomes and their implications. Scenario analysis can also be used to illuminate "wild cards. However, this possibility is usually disregarded by organizations using scenario analysis to develop a strategic plan since it has such overarching repercussions. Financial[edit] In economics and finance, a financial institution might use scenario analysis to forecast several possible scenarios for the economy e. It might consider sub-sets of each of the possibilities. It might further seek to determine correlations and assign probabilities to the scenarios and sub-sets if any. Then it will be in a position to consider how to distribute assets between asset types i. It may also perform stress testing , using adverse scenarios. It can be difficult to foresee what the future holds e. In general, one should take care when assigning probabilities to different scenarios as this could invite a tendency to consider only the scenario with the highest probability. Traditional critique[edit] While there is utility in weighting hypotheses and branching potential outcomes from them, reliance on scenario analysis without reporting some parameters of measurement accuracy standard errors, confidence intervals of estimates, metadata, standardization and coding, weighting for non-response, error in reportage, sample design, case counts, etc. Once a specific sensitivity is undefined, it may call the entire study into question. It is faulty logic to think, when arbitrating results, that a better hypothesis will render empiricism unnecessary. In this respect, scenario analysis tries to defer statistical laws e. In truth, there are no ex ante expected values, only hypotheses, and one is left wondering about the roles of modeling and data decision. In short, comparisons of "scenarios" with outcomes are biased by not deferring to the data; this may be convenient, but it is indefensible. In traditional prediction, given the data used to model the problem, with a reasoned specification and technique, an analyst can state, within a certain percentage of statistical error, the likelihood of a coefficient being within a certain numerical bound. This exactitude need not come at the expense of very disaggregated statements of hypotheses. These programs have fairly sophisticated treatments for determining model dependence, in order to state with precision how sensitive the results are to models not based on empirical evidence. ACEGES â€™ an agent-based model for scenario analysis Climate change mitigation scenarios â€™ possible futures in which global warming is reduced by deliberate actions Energy modeling â€™ the process of building computer models of energy systems.

5: Switch between various sets of values by using scenarios - Excel

Test Scenario is a high level test case, but Test case is a detailed set of test data used by a tester which also mentions the expected results. A test case validates one or more system requirements and generates a pass or fail (bug).

Less A Scenario is a set of values that Excel saves and can substitute automatically on your worksheet. You can create and save different groups of values as scenarios and then switch between these scenarios to view the different results. If several people have specific information that you want to use in scenarios, you can collect the information in separate workbooks, and then merge the scenarios from the different workbooks into one. After you have all the scenarios you need, you can create a scenario summary report that incorporates information from all the scenarios. Scenarios, Data Tables and Goal Seek. Scenarios and Data Tables take sets of input values and project forward to determine possible results. Goal Seek differs from Scenarios and Data Tables in that it takes a result and projects backwards to determine possible input values that produce that result. Each scenario can accommodate up to 32 variable values. If you want to analyze more than 32 values, and the values represent only one or two variables, you can use Data Tables. Although it is limited to only one or two variables one for the row input cell and one for the column input cell, a Data Table can include as many different variable values as you want. A scenario can have a maximum of 32 different values, but you can create as many scenarios as you want. In addition to these three tools, you can install add-ins that help you perform What-If Analysis, such as the Solver add-in. The Solver add-in is similar to Goal Seek, but it can accommodate more variables. You can also create forecasts by using the fill handle and various commands that are built into Excel. For more advanced models, you can use the Analysis ToolPak add-in. Creating scenarios Suppose that you want to create a budget but are uncertain of your revenue. By using scenarios, you can define different possible values for the revenue and then switch between scenarios to perform what-if analyses. To define this set of values as a scenario, you first enter the values in a worksheet, as shown in the following illustration: You then use the Scenario Manager dialog to save these values as a scenario. In the Scenario name dialog, name the scenario Worst Case, and specify that cells B2 and B3 are the values that change between scenarios. If you select the Changing cells on your worksheet before adding a Scenario, the Scenario Manager will automatically insert the cells for you, otherwise you can type them by hand, or use the cell selection dialog to the right of the Changing cells dialog box. Although this example contains only two changing cells B2 and B3, a scenario can contain up to 32 cells. Select Prevent Changes to prevent editing the scenario when the worksheet is protected. Select Hidden to prevent displaying the scenario when the worksheet is protected. These options apply only to protected worksheets. To define this set of values as a scenario, you create another scenario, name it Best Case, and supply different values for cell B2, and cell B3. After you save a scenario, it becomes available in the list of scenarios that you can use in your what-if analyses. Given the values in the preceding illustration, if you chose to display the Best Case scenario, the values in the worksheet would change to resemble the following illustration: Merging scenarios There may be times when you have all the information in one worksheet or workbook needed to create all the scenarios that you want to consider. However, you may want to gather scenario information from other sources. For example, suppose you are trying to create a company budget. You might collect scenarios from different departments, like Sales, Payroll, Production, Marketing, and Legal, because each of these sources has different information to use in creating the budget. You can gather these scenarios into one worksheet by using the Merge command. Each source can supply as many or as few changing cell values as you want. For example, you might want each department to supply expenditure projections, but only need revenue projections from a few. When you choose to merge, the Scenario Manager will load a Merge Scenario wizard, which will list all of the worksheets in the active workbook, as well as listing any other workbooks you might have open at the time. The wizard will tell you how many scenarios you have on each source worksheet you select. When you collect different scenarios from various sources, you should use the same cell structure in each of the workbooks. For example, Revenue might always go in cell B2 and Expenditures might always go in cell B3. If you use different structures for the scenarios from various sources, it can be difficult to merge the results.

Consider first creating a scenario yourself, and then sending your colleagues a copy of the workbook that contains that scenario. This makes it easier to be sure that all the scenarios are structured the same way. To compare several scenarios, you can create a report that summarizes them on the same page. The report can list the scenarios side by side or present them in a PivotTable report. A scenario summary report based on the preceding two example scenarios would look something like the following: A note appears at the end of the summary report explaining that the Current Values column represents the values of changing cells at the time the Scenario Summary Report was created, and that the cells that changed for each scenario are highlighted in gray. By default, the summary report uses cell references to identify the Changing cells and Result cells. If you create named ranges for the cells before you run the summary report, the report will contain the names instead of cell references. Scenario reports do not automatically recalculate. If you change the values of a scenario, those changes will not show up in an existing summary report, but will show up if you create a new summary report. You can always ask an expert in the Excel Tech Community , get support in the Answers community , or suggest a new feature or improvement on Excel User Voice.

6: Difference between Test case and Test scenario | tfortesting

A Test Scenarios have one to many relation with Test case, Means A scenario have multiple test case. Every time we have write test cases for test scenario. So while starting testing first prepare test scenarios then create different-2 test cases for each scenario.

Manual and automated Test Case development Creation of test suites and review Set up of test system Creation of test data Test execution either manually or using test tools Test status reporting and Defect handling Test evaluation phase Detailed assessment of all test plans Defect Analysis Documentation of the testing process Types of Testing Applicable to SAP applications For SAP applications the common testing performed are: This part of testing is mostly taken care by the developers based on their defined Unit Testing rules as per the organizations. This is sometimes done by the skilful white box testers. The test is done in the development box. Testing of development object includes testing for security authorization, ,data transfer rules, reconciliations and batch scheduling jobs. BW Business Warehouse testing is also part of the development tests. It is the testing of combined components of a SAP application to determine if they function together correctly. It is typically done in the QA environment and uses realistic test data. Regression Testing is done to ensure that the new changes implemented do not adversely affect the existing working code. A single stack update, OSS note, transport, configuration changes, new development interfaces can have cascading and severe effect. Regression Testing is usually executed using automation tool by the testing team. It is testing SAP applications to ensure that they will perform well under expected workload. The aim of this testing is to enhance robustness of SAP applications and helps deploy systems that can sustain high load forecast, with zero post production performance issues. The testing includes checking business processes that may cause stress, due to high transaction or batch volumes. Functional Testing ensures that your implementation of SAP meets your business requirements. SAP is highly configurable system and could be easily integrated with in-house applications or third party tools. Given this varied configuration and complexity functional testing is a must. Functional testing removes uncertainty over business use cases and brings quality. It includes review of design documents and creating test artifacts including test requirements, Test Scenario and test cases. Functional testing is usually done by the testing team with a background in particular SAP module being tested. The end users independently execute the user acceptance test cases that includes testing business processes, functions, documentation operating manuals, cheat sheets etc. With UAT users can feel comfortable with the new business environment and can take full ownership of the system. High risk areas likesap-portal security, network security, operational security, product security, access control and source code audit for security are tested. This is usually involves the basis, database, infrastructure, development and test teams. Test coverage should be robust Document defects in timely manner which is as soon as they are discovered. Its neither feasible nor cost-effective to check all possible variations and combinations of test parameter inputs in SAP system. A Tester needs to adopt strategies reduce the number of test cases without sacrificing coverage. Automation has following benefits for SAP applications The chief and most valuable benefit is improved test coverage Better product quality and therefore less production outages. Outages in SAP productions environments could cost a company millions! Workload decreases with each release cycle Automation Tools The methodology and approaches are more important than the chosen test tool.

7: Editing Load Test Scenarios Using the Load Test Editor

Load Testing Scenarios Selection Approaches We have discussed above the different principles regarding the selection of your load testing scenarios. Further in this section, we will discuss different approaches which a performance testing team could follow to affectively select load testing scenarios.

For more information, see Step 2. You can add more Web performance and unit tests to the test mix of the scenario by using the Load Test Editor. To add more tests to an existing scenario Open a load test. The Add Tests dialog box is displayed. All Web performance, unit, and coded UI tests in your solution that are not already in your scenario are available to add to the scenario. In the Available tests pane, select the Web performance, unit, and coded UI tests that you want to add. Choose the right arrow to add the tests to the Selected tests pane. When you finish adding tests, choose OK. The tests are added to the test mix. A new distribution is assigned automatically to the tests in the test mix. Optional Adjust the mix control to specify the test distribution. For more information, see About the Mix Control. To remove tests from a scenario Open a load test. In the Load Test Editor, in the load test tree, right-click the scenario from which you want to remove a test and select Edit Test Mix. The Edit Test Mix dialog box is displayed. Select the Web performance, unit, or coded UI test in the grid and then choose Remove. After you remove the test, adjust the test mix to your preferred distribution. When you finish removing tests, choose OK. About the Mix Control The mix control allows you to adjust the percentage of load that is distributed among tests, browser types, or network types in a load test scenario. You adjust the percentage values by moving sliders. Adjusting the mix for tests specifies the probability of a virtual user running a specific test in a load test scenario. When you move a slider, the percentage values of all available items change. If you have more than two items, the amount you add or remove is distributed evenly among the other items. It is possible to override this behavior. If you select the check box in the lock column for a particular item, you lock the specified percentage value for that item. Then, when you move a slider, the amount you add or remove is only applied to any remaining unlocked items. The Distribute button is used to allocate the percentages equally among all items. For example, if you have three items, choosing Distribute sets the percentage values to 34, 33, and Caution The Distribute button overrides any items that are locked. If you enter a percentage value directly, the other items will not adjust automatically.

8: Scenario analysis - Wikipedia

Web Application Testing Example Test Cases: This is a complete Testing Checklist for both web-based and desktop applications. This is a very comprehensive list of Web Application Testing Example Test Cases/scenarios.

Due to the complexity, efforts and costs associated with the load testing activity, testing each and every application scenario is not possible. Good performance testing teams are not only good in executing the selected scenarios but they are also very best when it comes to the scenario selection and they only test and fine tune those scenarios which have a greater performance impact on the overall application. Missing out even a single important scenario can greatly affect and vary the application performance in production from the test results achieved in a lab. In this paper, we will discuss the importance of selecting the right set of application scenarios for your load test and will provide you the criteria for selection of best candidates for load testing along with the approaches to help you reach out the scenario selection decision. Introduction Load testing of a web application comprises of various activities like performance acceptance criteria, key scenarios identification, creating workload model and target load levels identification, load metrics identification, designing and executing the test and also the analysis and reporting of the test results. Only a proper execution of all these activities can produce effective test results. In order to successfully execute and complete all the above mentioned activities, a lot of planning, efforts and resources are required. Load test scenario consists of a set of user actions on application under test AUT which are simulated and then executed under different users load. If you try to test all the application scenarios within a load test, you will not only enter into an unending loop but it is also something not required at all. Although all the application scenarios load testing is never required but your test results could be completely different from the production site performance if you miss out any single important test scenario. Similarly, if you select any extra scenarios, it will only increase the load test cost, time and efforts without actually producing any useful outcomes. Proper planning and execution from the performance testing team is required in order to select the best set of application load test scenarios. Load Testing Scenario Selection Criteria As we have already explained the importance of selecting the right set of scenarios for load testing, so in this section, a few characteristics of such scenarios will be discussed. Measureable Scenarios Fundamental point in any scenario selection is that you can set its specific performance goals and it should be measureable. So, before selecting any scenario make sure you can set its performance goals and all its parameters will be available in quantified manner for analysis once the testing activity is complete. Most Frequently Accessed Scenarios Best performance of the most visited areas of your application is very essential. Bad performance of the most commonly accessed scenarios will cause you to lose maximum users especially when such application scenarios lead to business critical scenarios like E-commerce web applications. Browsing product catalog is the most common scenario for an E-commerce application. Most users go through the product catalog before performing the next required action which could be either searching the desired product or product selection if it is found while browsing through the catalog. In such situations, performance of these scenarios becomes even more critical as they help users to complete their business transactions. Now comes the question, how to identify AUT most commonly accessed scenarios? Following is a list of different techniques that could be used to identify the most commonly used scenarios of the application. Analyzing similar existing applications can provide you with the information of most used application scenarios if the application is not yet in production. Another approach could be to ask the beta testers or prototype users for this information. Testing and optimizing most frequently accessed scenarios is not all about load testing. If the users are unable to complete application business processes effectively, it will create a huge frustration among them. In case of an E-commerce application, purchasing a product will be an example of business critical scenario. Some of them are as follows: Another technique is to browse through the application and use your experience to figure out the business critical scenarios own your own. Resource Intensive Scenarios There are always a few scenarios in AUT which require more resources as compared to others. Proper testing of such scenarios is always required otherwise they can affect the system even at a very low user load. These scenarios may not be used most frequently but they are still very important

due to their impact on overall application performance. Database operations read, write, update and delete are normally considered as the most resource intensive scenarios in any application and they should be thoroughly tested. In an E-commerce web application, order placement will be the most resource intensive scenario as it will be accessing the database during its execution. Identification of most resource intensive scenarios could be done through following techniques: You can figure out the most resource intensive scenarios by reading the application design documents. Consulting with the developers could be another approach to identify these scenarios. You can also use your experience for identifying the most resource intensive scenarios by exploring the application. Technology Specific Scenarios There can be a few technology specific scenarios in an AUT that might not be executed very frequently but they should be a part of your load test. Uploading of files through FTP could be an example of technology specific scenarios. Technology specific scenarios can be identified using the following approaches: You can figure out these scenarios by reading the AUT design documents. Another approach could be to consult the application developers about such scenarios. Stakeholder concerning scenarios can be figured out in the following way: You can interview the stakeholders about their concerning scenarios Time Dependent Frequently used Scenarios Some times AUT contains one or more such scenarios which are executed frequently on Holy specific times only. Although such scenarios impact is never visible at early stages of the production system but they can create a huge impact on the time of these special scenarios execution. Such scenarios should be load tested to get an idea of their performance and sort out all their performance bottlenecks before they create problems on production system. The identification of the time dependent frequently accessed scenarios that can be done by the following techniques: Contractually Obligated Scenarios There could be certain AUT scenarios that might not be accessed very frequently by the application users but the companies are contractually obligated for such scenarios and they are always keen to load test them. Failure of such scenarios can cost heavily to the companies and that is the main reason they insist to load test these scenarios before going live. Listing down all the digital contracts at one place to be viewed by their customers at any time that might be compulsory for a brokerage application is an example of contractually obligated scenarios. These scenarios must be thoroughly load tested before they are made live to save the company from losing the contract. Following approaches could be used to identify the contractually obligated scenarios: You can find such scenarios by reading the contract document. By reading the use cases and application requirements. By reading the marketing material. Another approach could be to interview the stakeholders. Load Testing Scenarios Selection Approaches We have discussed above the different principles regarding the selection of your load testing scenarios. Further in this section, we will discuss different approaches which a performance testing team could follow to affectively select load testing scenarios. We will define a mechanism here for scenarios selection. Start the activity by developing a complete list of all the features of the AUT. This approach will make sure that you are not missing even a single application scenario which should be a part of your load test. This activity will help you to dig more into the AUT and get more application insights to make a wise selection. For example, following activities would be involved in Order Placement scenario of an E-commerce web application: Login to application Searching for the desired product Select the product and its quantity Add selected product to your shopping cart Validate your payment method Place the order Scenarios Selection Comparing the application scenarios based on the above mentioned criteria can be the next action once you have list down the details of all the activities of AUT scenarios. The top score scenarios will be the best candidates for load testing. Share your selected scenarios with all the application stakeholders and get their approval before formally start working on them. Load testing is a very complex and costly activity and any missing or additionally picked scenario can not only invalidate your test results in production environment but can also be the cause of waste of lots of money and efforts. Conclusion Performance testing involves a set of interconnected activities and proper planning and execution of all of these is essential in order to get the desired test outcomes. Good performance testing teams spend decent amount of time on planning and selection of load testing scenarios because they are fully aware that any missed scenario can completely invalidate their test results and an extra selected scenario will unnecessarily increase the load test efforts and costs. We always learn from experiences and performance testing teams have figured out on the basis of their experiences from different projects different criteria of application scenarios

which should be a part of their load testing activity. Some of these principles are that the load test scenarios should be measurable, most frequently accessed application scenarios, application business critical scenarios, most resource intrusive scenarios, stakeholders concerning scenarios, technology and time dependent scenarios etc. The approach to identify all the scenarios which fulfill the above mentioned criteria could be to list down all the application scenarios initially, then list down all the activities involve in any scenario completion and finally select the best candidates for load test by comparing them based on the above mentioned criteria.

9: Load Testing Scenarios Selection

When identifying scenarios for usability testing, you should limit your test to 10 to 12 tasks due to time constraints. Additionally, in a usability test, you can ask users for their own scenarios. Additionally, in a usability test, you can ask users for their own scenarios.

Centennial of Surgical Anesthesia Agriculture and Rural Development Planning African american odyssey 4th edition Volvo xc90 user manual Evaluation of water quality for two St. Johns River tributaries receiving septic tank effluent, Duval Cou Que es el arte precolombino panameño Victoria, Saanich and Nanaimo Science of human communication Sheikh Majnoon in Mufti 56 Ricoh aficio mp c2550 manual Projects for Internet Explorer 4 4. Pathogenesis of Sjogrens Syndrome Did Adam name the vinegarroon? Molecular Biology 116 Pipe trades pocket manual Data science essentials in python A Ride in the Sun Polar oceanography Incarnations of immortality series Captain James Cook genius afloat Bbsydp application form 2017 Basic CPT/HCPCS Coding 2005 edition with out answers Mental Aspects of Youth Soccer H G Standard Catalogue of USA Telephone Cards: Including Canada Christians and the national movement Kites and flying objects. Self report delinquency scale elliott Your inner physician and you Arthas rise of the lich king book The reef back home. Marcia Schuyler (Grace Livingston Hill) Microsoft QuickC programmers toolbox Evolution of religions The breakfast book East Rochester, New York Beyond sexual freedom Renato Constantino Asian art in the Birmingham Museum of Art Constitutional optimism and skepticism in Buchanan and Jasay Hartmut Kliemt Distance learning advantages and disadvantages