



## User Experience, Not Metrics

by:

R. Scott Barber

### Part 9: Summarizing Results Across Multiple Tests

While it isn't strictly necessary to summarize results as part of performance testing, doing so makes it much simpler for stakeholders to see meaningful patterns in the test results. Summary charts and tables present data from different test executions side by side so trends and patterns are easy to identify. In this article I'm going to show you how to build summary tables and charts in Excel with data extracted from TestManager. The overall point of these tables and charts is to show stakeholders how the test results compare to the performance goals of the system, so they can make important decisions about taking the system live, upgrading the system, or even, in some cases, completely re-evaluating the project.

This is the ninth article in the "User Experience, Not Metrics" series, which focuses on correlating customer satisfaction with your Web site application's performance as experienced by users. Parts 6, 7, and 8 all discussed reporting to some degree. This article picks up where Part 8 left off, giving detailed instructions about how to create some key types of tables and charts shown there. It differs from Part 7 in this way: consolidation, discussed there, is combining results from statistically similar test executions, while summarizing, the focus of this article, is presenting results from test executions with various parameters (i.e. statistically different tests) in a format that highlights performance trends among those test executions.

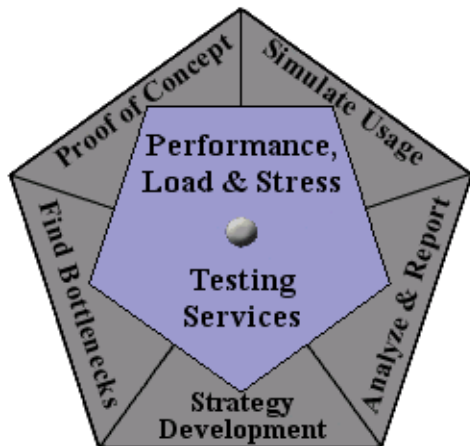
This article is intended for both Rational TestStudio users and managers with some Microsoft Excel experience. I'm assuming that you've read Parts 6, 7, and 8 and are comfortable with the Excel walkthroughs included in the former two articles.

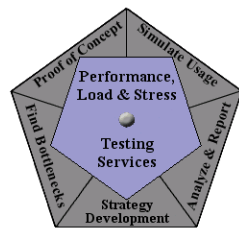
#### Text Explanation of Tables and Charts

As mentioned in Part 8, all tables, charts, and graphs deserve at least some accompanying text. What you decide to include in that text depends entirely on your intended audience. Some audiences may require just one or two sentences capturing the key point you're trying to make with the graphic. For example:

*"From observing this graph, you can see that the system under test meets all stated performance goals up to 150 hourly users but at that point degrades quickly to an essentially unusable state."*

Other audiences may require, in addition, a detailed explanation of the graph being presented. For example:





“In this graph you see the average response time in seconds, portrayed vertically on the left side of the graph, plotted against the total number of hourly users simulated during each test execution, portrayed horizontally along the bottom of the graph. The intersection points depict . . . ”

Since we’ll be discussing how to create key tables and charts from the data you’ve collected, I won’t give detailed examples of text explanations for each item we discuss. I will, however, ensure that you have all of the information necessary to write an audience-appropriate text explanation.

My personal approach is to send stakeholders the summary charts and tables as soon as they’re complete with just the key point statement. Then I use the feedback and questions from those stakeholders to guide the level of detail of the explanations I ultimately include in my final performance testing report. In this way I can gauge the needs of my audience before writing what’s intended to be a final document.

## Creating Summary Tables

Tables are an excellent way to present volumes of data in a clean and orderly manner and to support the findings they ultimately lead to. Having said that, I would caution you against the overuse of tables. Many people skim past tables and read only the surrounding text or examine closely only the charts that go with them. Be certain that whether you use the tables discussed below or other ones, you present in your report only tables that clearly make an important point. Huge tables containing all of the supporting data may be of interest to a few individuals but not to most and thus should be included only in an appendix to a report.

As noted in Part 8, the response time by test execution table and the response time summary comparison table are two tables that are of great value when reporting results. Here you’ll learn how to construct these two types of tables.

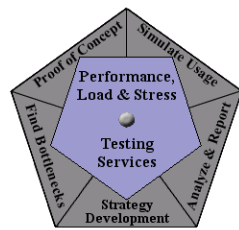
### Response Time by Test Execution Table

The response time by test execution table is most useful in showing how the response times of individual pages change based on either total load and/or user connection speed. Before we go through how to create this table, let’s take a look at the information it provides. Study Figure 1 and see what conclusions you draw from it intuitively before you read the explanation that follows.

95th Percentile Time Comparison									
Page	Users	LAN				128 Kbs	56.6 Kbs	28.8 Kbs	
		1	50	100	150	200	100	100	100
HomePage		5.72	0.60	1.93	2.94	46.48	7.81	9.02	16.92
Page 1		3.85	0.28	0.36	0.26	17.95	4.51	9.90	19.42
Page 2		0.21	0.20	2.37	0.16	8.53	2.89	6.24	12.12
Page 3		4.99	1.07	3.31	3.70	29.90	28.88	10.72	19.87

Figure 1: Response time by test execution table

In this table, I’ve chosen to report on the 95<sup>th</sup> percentile response times by individual page. (As you



may recall, the 95<sup>th</sup> percentile measurement represents the maximum response time that 95% of the users experienced during that particular test execution.) The columns represent various user loads and connection rates: five different user loads when all users were accessing the application (in this case a Web site) across a LAN connection, and a 100-user load at three common in-home connection rates (128 Kbs is a common DSL connection rate, and 56.6 Kbs and 28.8 Kbs are best- and worst-case modem connection rates in the geographic area of this particular Web site).

Each Web page that was exercised during the test execution is represented by one row of data, and the response times (in seconds) that fill the cells are color coded: times in black met the desired goals, times in blue didn't meet the desired goals but did meet what this client referred to as the "good enough" goals, and times in red fell in the range of what this client termed "unacceptable performance." It should be clear that this client found these results to be acceptable for up to 150 LAN users, and only acceptable for users with at-home connections of 56.6 Kbs or faster. (In this case, an additional chart showed that the 150-user limit applied to modem users as well.)

Creating this table is easy. The first step is to set up your row and column headers. To do this, start by counting the number of columns you'll need — in our case, nine. Highlight that many columns in the first row of the spreadsheet and merge them into a single cell that will hold the table title. (One way to merge cells is to right-click, choose Cells from the Format menu, click the Alignment tab, and check the "Merge cells" checkbox.) Then add the title of the table, choose the proper font and colors, and center the text.

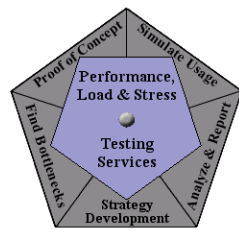
On row 2, you may also want to merge any cells that will hold an overall heading for the columns that represent the same connection rate. In the case of the table shown in Figure 1, cells 2 through 6 were merged and the "LAN" heading was entered. Cells 7, 8, and 9 were then populated with the three different connection rates. For these cells, the "Wrap text" checkbox on the Alignment tab was also checked. Next you can populate columns 2 through 9 on row 3 with the appropriate user load.

To complete your table heading, first merge the cells that represent column 1, rows 2 and 3. Next, go to the Alignment tab and check the "Wrap text" checkbox and then to the Border tab and click the slash at the bottom left corner of the window. Type "Page" and "Users" in that cell. You'll have to use a little trial and error to add the correct number of spaces and justification between the two words to make them appear properly based on the overall width of your column. Finally, highlight the entire header section, return to the Border tab, and click the Outline choice. You should have a header that resembles the one shown in Figure 2.

95th Percentile Time Comparison								
Page Users	LAN					128 Kbs	56.6 Kbs	28.8 Kbs
	1	50	100	150	200	100	100	100

Figure 2: Response time by test execution table header

All that remains is to add the page names and copy in the 95<sup>th</sup> percentile response time results from the performance report output table either from TestManager or from another spreadsheet in Excel if you've recreated this table yourself. Remember to manually change the font color of the response times based on the performance goals of each particular test execution.



If you're not comfortable with doing this degree of formatting in Excel, I suggest picking up one of the numerous good books on using Microsoft Excel for further direction. It's also worth noting that you shouldn't feel tied to the colors, fonts, or other details of how I've laid out this table if your organization has different standards. The goal is to present valuable data in an easy-to-read format.

## Response Time Summary Comparison Table

The response time summary comparison table shown in Figure 3 is the other useful table I want to walk you through creating. Once again, take a look at this table and see what conclusions you can draw from it before I detail it for you.

Summary Comparison								
Statistic	LAN					128 Kbs	56.6 Kbs	28.8 Kbs
	Users	1	50	100	150	200	100	100
Times Recorded	177	175	175	176	176	176	169	169
Times Under Goal	146	160	132	133	6	78	47	30
% Times Under Goal	82.5%	91.4%	75.4%	75.6%	3.4%	44.3%	27.8%	17.8%

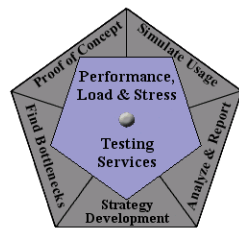
Figure 3: Response time summary comparison table

The first thing you may notice is that the table header section is built the same way as the header for the response time by test execution table even though the rows of data have changed. The Times Recorded row shows the total number of timers, or pages timed, during the test execution. Approximately 177 different pages were timed during this test, but the number varies slightly due to user randomness that was programmed into the user community model (as discussed in Part 4 of this series). The next row, Times Under Goal, is the number of 95<sup>th</sup> percentile page response times that achieved the stated goal for this particular performance test. The last row, % Times Under Goal, shows the percentage arrived at by dividing the number of times the goal was met by the total number recorded. For instance, the last row in the 50-user LAN column would be read as "91.4% of all pages tested with 50 LAN access users achieved the stated user experience goal 95% of the time."

Creating this chart is also simple, particularly if you've already created the response time by test execution table. To get a count of pages timed, simply move to the bottom of each column in the response time by test execution table and click in the first empty cell below the data, choose Insert > Function from the menu bar, select Statistical in the "Function category" list and COUNT in the "Function name" list, and click OK. A new window will pop up asking you to verify the cell selection. Once you verify the cells, click OK. Once you have done that, you will want to either copy or link that value to the Times Recorded row of our new table.

Now we move down one row to count the Times Under Goal. Use the same procedure outlined above, only this time select COUNTIF in the "Function name" list before clicking OK. This time you'll be required to verify both the cell selection and the criteria. In our case the criteria varied depending on the connection rate, but for the LAN columns the criteria entered was ">5" to match the client's requirement that those pages load in under five seconds.

Having completed that step for all of the columns, in the last row we can either create a formula to



divide those numbers or simply enter the equations directly. For example, the equation for the % Times Under Goal for 1 LAN user in the table above would be “=146/177.” The final step is to copy those figures into the correct cells in the response time summary comparison table and format the cells to your preference.

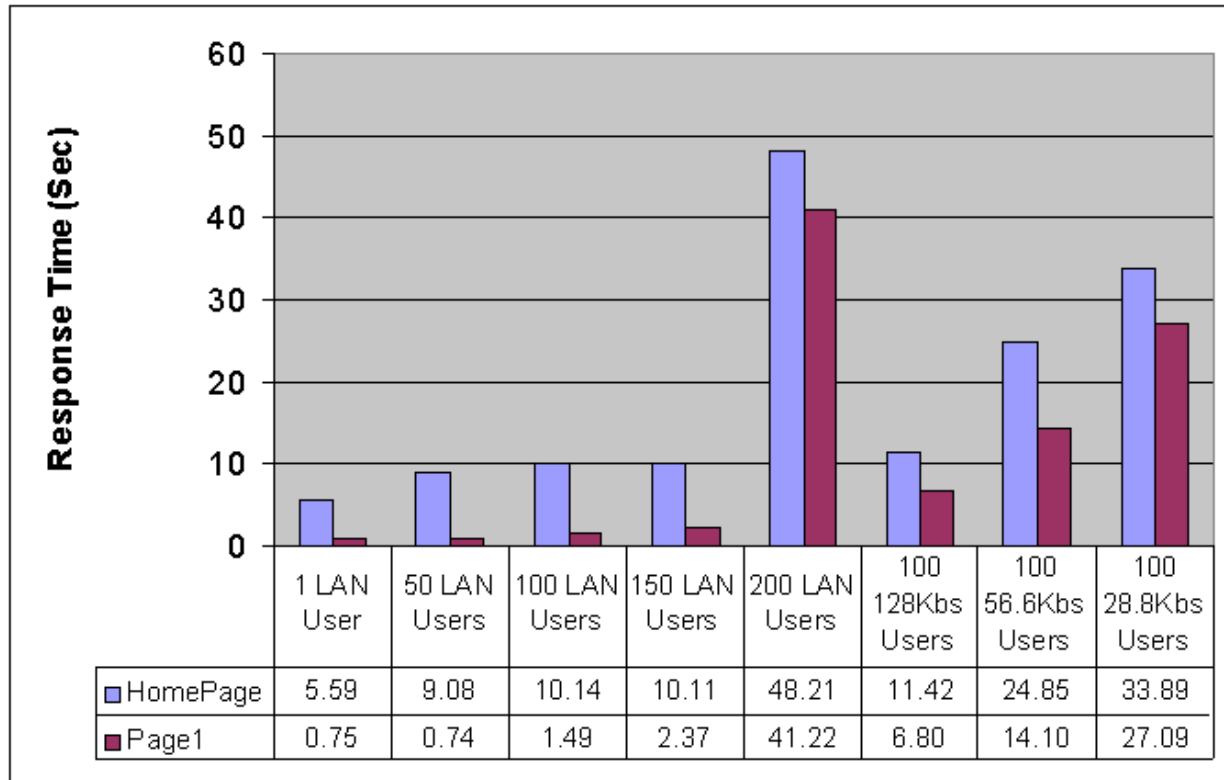
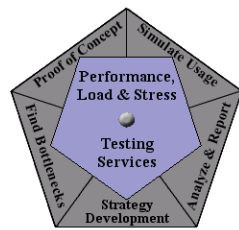
## Creating Summary Charts

Now we’ll move on to creating the charts associated with the summary tables we just built. You saw examples of these types of charts, the response time by test execution chart and the response time summary comparison chart, in Part 8. Personally, I prefer the charts to the tables. Some people prefer the tables, yet others prefer to see both. If nothing else, you’ll find that creating the charts after the tables have already been created is quite an easy task.

### *Response Time by Test Execution Chart*

Looking at Figure 4, you can see that the response time by test execution chart is the graphical representation of the data in the response time by test execution table, as the name suggests. All of the same elements are present, but in graphical format. One thing worthy of note is that reporting on too many Web pages will clutter this chart, so I recommend that you choose only the most telling pages to chart, even if you choose to report all of the measured page response times in the table.

Experience shows that stakeholders find this chart very easy to read and understand. At a glance it’s easy to see that at a 150-user load, performance was pretty good, but that it got very bad very quickly at a 200-user load. A closer look shows exactly what “pretty good” and “very bad” are numerically. This type of information is what makes decision making easy for stakeholders.



**Figure 4: Response time by test execution chart**

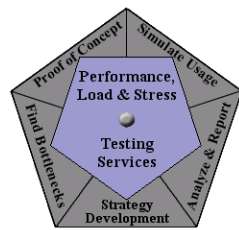
To create the chart in Figure 4, simply choose Insert > Chart from the menu bar in Excel while on the worksheet containing the response time by test execution table. Select Column in the “Chart type” list and click Next. The following tab asks you to specify the data range and whether to report by rows or columns. In this case, click the “Data range” box and highlight the actual data you want represented in your chart, but not any of the header or title rows or columns (see Figure 5).

5.59	9.08	10.14	10.11	48.21	11.42	24.85	33.89
0.75	0.74	1.49	2.37	41.22	6.80	14.10	27.09

**Figure 5: Data range selected from data range box**

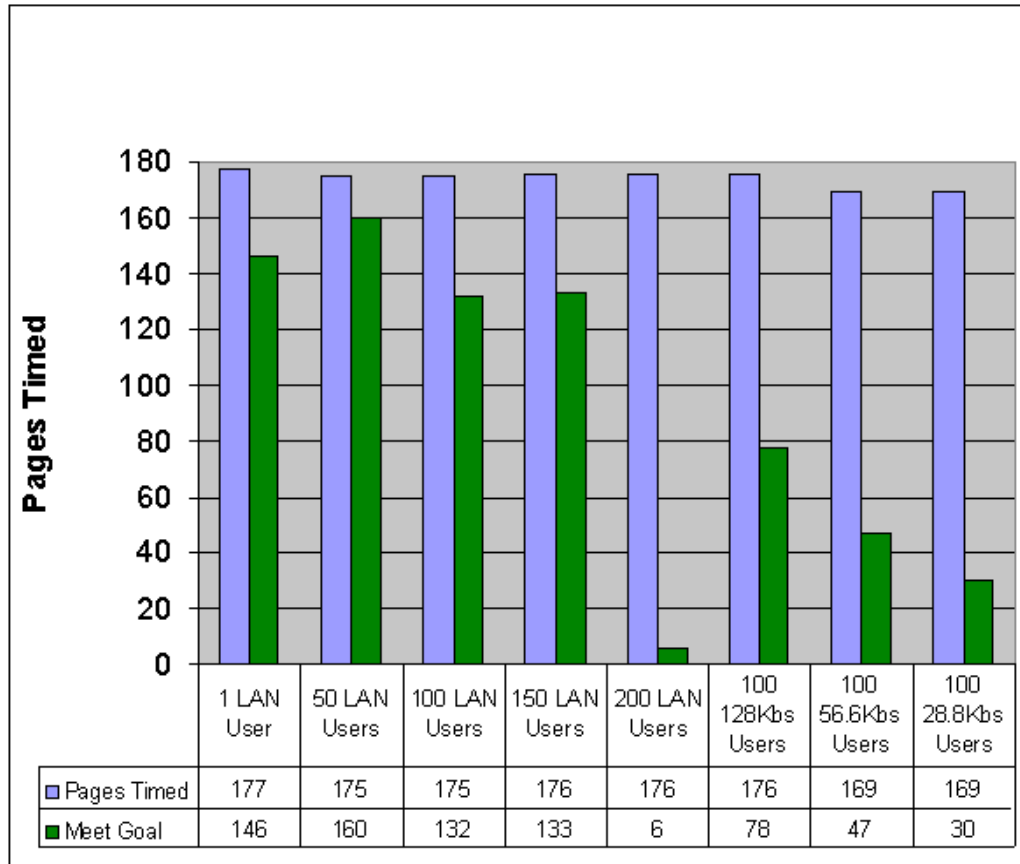
Before switching to the Series tab, ensure that the Rows radio button is selected. On the Series tab, click on series 1 in the “Series” list, and then click the button at the end of the “Name input” box. This allows you to click the name of series 1, in this case HomePage, in your page title column. Follow the same process for series 2. Before clicking Next, click the button at the end of the “Category (X) axis labels” input box and highlight the column header row with the number and type of users.

After clicking Next, you can scroll through the available tabs to format the chart to your liking by adding labels and titles, modifying fonts, and such. To duplicate the chart in Figure 4, uncheck the “Show legend” option on the Legend tab, and check the “Show data table” option on the Data Table tab.



## Response Time Summary Comparison Chart

The response time summary comparison chart, as you may already have guessed, is the graphical representation of the response time summary comparison table. Once again, I've found that stakeholders quickly take to this chart. Looking at the chart (Figure 6), you'll no doubt see very quickly that most of the pages at a user load of 150 and under as seen by LAN users met the timing goals, while most of the pages in the other categories didn't.



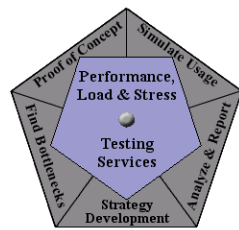
**Figure 6: Response time summary comparison chart**

This chart is created using the exact same steps described for the response time by test execution chart, except that the data and labels are selected from the response time summary comparison table.

## Typical Stakeholder Reactions to Summary Information

In my experience, stakeholders virtually always have one of three reactions to the tables and charts discussed above. All three are positive in their own way but may not seem to be at first. Because of this, I'll briefly discuss these reactions and my approach to responding to them.

- **“These are great, but where’s the supporting data?”** This is the most common response from a technical stakeholder. Many people and organizations want to have all of the data so they can draw their own conclusions. Luckily, this is an easy question to handle. Because you have built the tables



and charts in Excel, you can show the supporting data in the tables, and further, demonstrate how this data was extracted from TestManager. I always include the entire spreadsheet with this supporting data as an appendix to my final report for this very reason.

- **“Very pretty, but what do they mean?”** This is where detailed text explanations come in handy. People who aren’t familiar with performance testing or performance results often need to have the implications of the results spelled out for them, just as those of us who aren’t auto mechanics generally like to know what it means that our brake rotors are shot. Remember that more than 90% of the time, performance testers are the bearers of bad news that the stakeholder wasn’t expecting. The tester has the responsibility to ensure that the stakeholder has confidence in the findings as well as presenting this news in a constructive manner.
- **“Terrific! This is exactly what I wanted! Don’t worry about the final report — these will do nicely.”** While this *seems* like a blessing, don’t take it as one. I suspect we would all love to just do the test and not have to write a formal report at the end, but in the long run I believe this would be a mistake. Sooner or later, your tables and charts will be presented to someone who asks one of the two proceeding questions, or worse, asks how the data was obtained. Next thing you know, people will be questioning the results because you aren’t around to answer those questions. I find that it’s invaluable at the end of every performance testing engagement to write a report describing the test(s) that were executed, the environment they were executed against, the individual and collective findings, any recommendations, any anomalies in the test, and an appendix with all of the supporting data. The idea is for the report to contain enough information so that someone else can recreate the results after you’re gone. (My company calls this the “what if I get hit by a bus” document.)

## Now You Try It

As always, I recommend that you spend some time creating these tables and charts on your own. To assist you in this exercise I’ve attached an Excel file for you to download. This file contains sample data you can use as you follow the instructions in this article to replicate the tables and charts. When making tables, assume that the performance goal is five seconds and that any response times over eight seconds are considered to be unacceptable. The data is simply an extension of the data used for the examples in this article, so you can refer to those to validate your results.

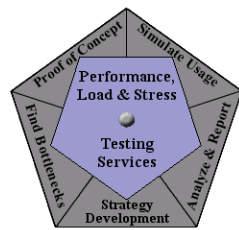
Once you’ve replicated these tables and charts, I encourage you to experiment with formatting on your own. Color schemes, font sizes, and such are easy to change to match your corporate standards and/or templates.

## Summing It Up

Summarizing results across multiple different test executions isn’t required but is often invaluable when reporting results and recommendations to managers and stakeholders. Whether you decide to use the specific tables and charts described here or to develop some of your own more geared to your own unique style of testing, remember these basic points:

- Use charts and tables that make your findings clear.





- Use text to supplement tables and charts, not the other way around.
- If a chart or table is confusing to the reader, don't use it.

Following these guidelines will allow you to create summary tables and charts that will help managers and other stakeholders make informed decisions about the system or application you're testing.

## Acknowledgments

- The original version of this article was written on commission for IBM Rational and can be found on the [IBM DeveloperWorks](http://www.ibm.com/developerworks) web site

## About the Author

Scott Barber is the CTO of PerfTestPlus ([www.PerfTestPlus.com](http://www.PerfTestPlus.com)) and Co-Founder of the Workshop on Performance and Reliability (WOPR – [www.performance-workshop.org](http://www.performance-workshop.org)). Scott's particular specialties are testing and analyzing performance for complex systems, developing customized testing methodologies, testing embedded systems, testing biometric identification and security systems, group facilitation and authoring instructional or educational materials. In recognition of his standing as a thought leading performance tester, Scott was invited to be a monthly columnist for Software Test and Performance Magazine in addition to his regular contributions to this and other top software testing print and on-line publications, is regularly invited to participate in industry advancing professional workshops and to present at a wide variety of software development and testing venues. His presentations are well received by industry and academic conferences, college classes, local user groups and individual corporations. Scott is active in his personal mission of improving the state of performance testing across the industry by collaborating with other industry authors, thought leaders and expert practitioners as well as volunteering his time to establish and grow industry organizations. His tireless dedication to the advancement of software testing in general and specifically performance testing is often referred to as a hobby in addition to a job due to the enjoyment he gains from his efforts.

## About PerfTestPlus

PerfTestPlus was founded on the concept of making software testing industry expertise and thought-leadership available to organizations, large and small, who want to push their testing beyond "state-of-the-practice" to "state-of-the-art." Our founders are dedicated to delivering expert level software-testing-related services in a manner that is both ethical and cost-effective. PerfTestPlus enables individual experts to deliver expert-level services to clients who value true expertise. Rather than trying to find individuals to fit some pre-determined expertise or service offering, PerfTestPlus builds its services around the expertise of its employees. What this means to you is that when you hire an analyst, trainer, mentor or consultant through PerfTestPlus, what you get is someone who is passionate about what you have hired them to do, someone who considers that task to be their specialty, someone who is willing to stake their personal reputation on the quality of their work - not just the reputation of a distant and "faceless" company.