

Story Board - Interviews

“As per NASSCOM, the market for software services is growing at an average of about 10-12 per cent while testing is growing at more than 50 per cent per annum,” Arun Rao, VP-HR, AppLabs

Sheetal Srivastava



1. Software testing is a field that has always been regarded as not an exciting option by IT professionals. Do you think this perception has changed lately?

This may have been true a few years back. But I genuinely believe that this perception has changed in the last couple of years. With the maturing of the outsourcing industry, the end-user of IT services and with business applications becoming more mission critical, the need for specialised software testing has become apparent. This means that it is no longer an insignificant check-it box when it comes to projects; it becomes an integral part of quality software being supplied to a discerning customer.

2. What is the size and scope of the Indian software testing market?

Software testing is evolving into a promising market for Indian players. With the prominence of IT as a business enabler, testing has become both, a specialised as well as a critical service. Gartner predicts the worldwide software testing market is at \$13 billion and the global market for outsourced testing services is around \$6.1 billion. Out of this, India is expected to corner 70 per cent share. The market opportunity for Indian offshore testing companies is seen at around \$8 billion by the end of 2008; from \$2-3 billion a year ago. Leading IT companies in India today derive up to 10 per cent of their revenues from their independent software testing businesses

As per NASSCOM, the market for software services is growing at an average of about 10-12 per cent while testing is growing at more than 50 per cent per annum.

3. Do you think Indian software testers are competent as compared to their global counterparts?

I have no reason not to believe that Indian software testers aren't as competent as their global counterparts. This is especially true when it comes to technology testing and usage of automated tools, etc. However, when one looks at the global scenario, I have found people adopt a career in testing after working in the industry for some time. These people bring their functional expertise while testing business applications and surely score over their Indian counterparts in terms of their effectiveness when it comes to functional testing of business applications.

4. Do you feel academia is providing sufficient knowledge to people or do they have to be trained to get them up to the mark? Is there any formal training/programme offered?

The formal education curriculum that most universities follow has only a representative mention of software testing and probably very little time is devoted to this field of study.

However, with the software testing industry looking up, there is some professional training that is available in terms of finishing schools. While such schools may not always have the best of content delivered, they are at least bridging a bit of the industry- academia gap that exists today. They help the organisations reduce the learning curve and hence help jumpstart a career in software testing.

5. What are the essential traits for a software tester?

I think the biggest quality that a person who is opting for a career in testing should have is the ability to critique dispassionately. An ability to “break the application” (figuratively speaking) to evaluate its robustness, an ability to have a constructive approach to fault-finding. Needless to mention, such a person should have perseverance, an aptitude towards problem identification, teaming capabilities because it is all about team-work, analytical thinking etc.

6. Is software testing an 'art' or 'science' or for that matter is it 'engineering'? How far can a career in software testing take you?

To me software testing is both art and science. Art, because it involves a fair amount of creative and out-of-the-box thinking. It is also an issue of representation of ones views.

Science, because we pursue a charter of either proving or disapproving a hypothesis that the software program is error-free. We then do exactly what we do in a scientific laboratory – experiment on the product and observe the impact.

Therefore, it is both a form of art and science. When we bring in concepts of frameworks and automation, it transcends from being a science to becoming a field akin to software engineering.

(Please send in your comments/queries to sheetal.srivastava@timesgroup.com)