

Databases selected: ProQuest Newspapers, ProQuest Central

E-WORLD: 4 threats to Indian software industry

Anonymous. **Businessline.** Chennai: Apr 7, 2008.

Abstract (Summary)

When the complexity of the required software product is known and well understood, these attributes are likely to result," he adds, in an e-mail interaction with eWorld. "Since software quality is hidden from apparent view most of the time, there is very little chance in ensuring software quality in a product, as an add-on activity with software quality assurance practices of reviews, inspection or testing. Software quality has to be built into the very guts of the product at each stage in the product evolution." Excerpts from the interview, in which, apart from sharing his insights on software quality, he also discusses passionately the four threats that the Indian IT industry faces. Can you trace the evolution of software quality assurance over the years? The evolution of software quality assurance can probably be traced back to the very first software project. However, for purposes of defining a timeline for software quality assurance, it is best to trace the evolution of software itself. Software was an accident that happened during the era of building business computers in the 1960s. It is believed that software built for early computers, which predominantly emphasised hardware solutions, was comparatively simple by today's standards. Software in the 1960s was a disposable asset since it was custom-made. An application only worked on the kind of computer it was written for. Software had to be replaced and rewritten if the underlying hardware changed. The first known software ever to have been built is the operating system. IBM's OS/360 dating back to 1968 is probably the first recorded origin for software applications with unexpectedly long life expectancy. It is, therefore, natural to conclude that software quality assurance, much like hardware quality assurance, took roots in the early 1970s. It is believed that the early software quality assurance groups were research-oriented, and through trial-and-error established the fact that testing alone was not sufficient to guarantee high-quality software solutions. So they tried software inspections. When inspections were combined with known forms of software testing, a cumulative defect removal efficiency of 99 per cent could be approached. Early research results reported on software inspections by Ron Radice and Mike Fagan while at IBM are the only known oldest recorded history to which software quality assurance dates back. Does our IT education system have to be modified to cater to software quality needs?

Full Text (1261 words)

(Copyright 2008. Financial Times Information Limited - Asia Africa Intelligence Wire. All Material Subject to Copyright .)

from BUSINESS LINE, April 07, 2008 What is software quality? How can it be ensured? "Quality is a hidden attribute or virtue that defines 'worthiness for use' of any product or service. It comes to light only under detailed scrutiny

Databases selected: ProQuest Newspapers, ProQuest Central

Software quality is this dimension of the hidden worthiness, of software," answers Raghav S. Nandyal, author of Making Sense of Software Quality Assurance. The best way to ensure software quality is to build it into the software product from the very beginning, he avers. "The earliest opportunity to build software quality into a product is at the time of requirements definition and elicitation." Some of the quality attributes Nandyal mentions are clarity, consistency, correctness, completeness and testability

"When the complexity of the required software product is known and well understood, these attributes are likely to result," he adds, in an e-mail interaction with eWorld. "Since software quality is hidden from apparent view most of the time, there is very little chance in ensuring software quality in a product, as an add-on activity with software quality assurance practices of reviews, inspection or testing. Software quality has to be built into the very guts of the product at each stage in the product evolution." Excerpts from the interview, in which, apart from sharing his insights on software quality, he also discusses passionately the four threats that the Indian IT industry faces. Can you trace the evolution of software quality assurance over the years? The evolution of software quality assurance can probably be traced back to the very first software project. However, for purposes of defining a timeline for software quality assurance, it is best to trace the evolution of software itself. Software was an accident that happened during the era of building business computers in the 1960s. It is believed that software built for early computers, which predominantly emphasised hardware solutions, was comparatively simple by today's standards. Software in the 1960s was a disposable asset since it was custom-made. An application only worked on the kind of computer it was written for. Software had to be replaced and rewritten if the underlying hardware changed. The first known software ever to have been built is the operating system. IBM's OS/360 dating back to 1968 is probably the first recorded origin for software applications with unexpectedly long life expectancy. It is, therefore, natural to conclude that software quality assurance, much like hardware quality assurance, took roots in the early 1970s. It is believed that the early software quality assurance groups were research-oriented, and through trial-and-error established the fact that testing alone was not sufficient to guarantee high-quality software solutions. So they tried software inspections. When inspections were combined with known forms of software testing, a cumulative defect removal efficiency of 99 per cent could be approached. Early research results reported on software inspections by Ron Radice and Mike Fagan while at IBM are the only known oldest recorded history to which software quality assurance dates back. Does our IT education system have to be modified to cater to software quality needs? Having integrated Peter Senge's seminal work on 'The Fifth Discipline,' into my understanding of process management, I can say that most issues are complex systemic issues that deal with dynamic complexity. To address them effectively, one needs to have systems thinking - the Fifth Discipline - instead of linear thinking. This shift requires a holistic approach to improving all aspects that bear upon the final determinant - good quality software. This includes improving the basic education system with a

Databases selected: ProQuest Newspapers, ProQuest Central

curriculum that is kept current and fresh catering to the needs of one of India's cash-cows: the IT industry. I would not be surprised if the current software engineering curriculum of most universities would not even make a footnote reference to the CMMI or the People CMM! Yet, a well-grounded understanding of articulating process models and an integration of such practices to a competency framework is what the IT industry needs. In terms of software quality, how does India fare, globally? In a very popular TV broadcast in the US, called '60 minutes,' the presenter is quoted to say: "The United States imports oil from Saudi Arabia, cars from Japan, TVs from Korea and whisky from Scotland. So, what do we import from India? We import people! Really smart people". This sums up the trust and importance the world has placed on the Indian software professional. They trust the quality of the work that is produced; for, the highest number of high process maturity software development organisations found anywhere in the world are still in India. With strong middle-class values and an emphasis on good quality education, our ethos is still very much learning-oriented and geared towards improving the intellectual faculties. Infrastructure provided by most software development organisations in India is world-class. At times, they even beat the best-of-the-best in the world! The logical and analytical abilities of the Indian software professional are unrivalled. Costs to produce good quality software using Indian software organisations are still within acceptable parameters of economic value. Given the nature of geographically distributed software development, assigning a unique ownership to the goof-ups is difficult. Often times, the customer is not sure of what they want until they see a prototype or some abstraction of what is in their reality. This is one opportunity where Indian software houses have to improve a great deal. We are very good, I believe in giving what the customer wants - and not leading him, literally, by the nose. We have to make entrepreneurial leadership across the rank-and-file in an organisation, a core competency. Developing individuals with excellent communication and presentation skills is still a big opportunity to improve. This gap is significant and I believe is the primary reason why we still don't have 'great software companies'. We might have many 'good software companies'

Again, good is the enemy of great! Consider this: who is the closest to the customer? The humble software engineer found in the trenches! The CEO of an Indian software organisation can do precious little marketing of his talent base given the fact that all such follow-up business opportunities get defined in the trenches - by the one who is dirtying his hands - the software developer or the tester. The CEO of an Indian software organisation is seldom in the trenches! I find another aberration, which is hard to digest. The Indian culture unfortunately has still not produced or created enough numbers of what is called the Level 5 leadership by Jim Collins. Level 5 leaders are modest and wilful, humble and fearless. From the People CMM perspective, it is very surprising to note that despite all notions that the Indian IT industry is informal with boss-subordinate relationships and so on which humanises the IT work culture, the most intimidating and hostile relationship is still the same - boss-subordinate

Databases selected: ProQuest Newspapers, ProQuest Central

relationships! I find this duality very difficult to digest. A third critical factor is - we have not learned to walk away from bad business practices

Indian software organisations are still mostly bottom line driven

When businesses function purely from a bottom line perspective, values and principles get thrown out of the window. The biggest threat, which established Indian software organisations must be sensitive about, is mostly internal. One need not look at other lower-cost countries as the threat. I am sure that the dominance of these established businesses can be easily challenged by a few upstart companies driven by entrepreneurial leadership that is highly ethical, people oriented, competency focused and quality driven

D. Murali and G. Padmanaban Copyright 2008 Business Line

Indexing (document details)

Author(s): Anonymous

Document types: News

Publication title: Businessline. Chennai: Apr 7, 2008.

Source type: Periodical

ProQuest document ID: 1458177881

Text Word Count 1261

Document URL: <http://proquest.umi.com/pqdweb?did=1458177881&sid=1&Fmt=3&clientId=53837&RQT=309&VName=PQD>

Copyright © 2009 ProQuest LLC. All rights reserved. [Terms and Conditions](#)
[Text-only interface](#)