

Deficiencies lurking behind the “Code Kings”

(Original creator: bolarotibi)

Guest contributor, Bola Rotibi from analyst firm Creative Intellect Consulting Acquiring the very best development talent goes beyond having the best code developer. What skills should underpin this broadly applied term and how should one go about supporting the creative process of development. Read part 1 of this series [here](#)

Deficiencies lurking behind the “Code Kings”

A good developer needs more than coding skills. But wait a minute; hasn't that always been the case? Yes, but in the past, coding prowess played higher in the job requirement. Prior to the Rapid Application Development movement, developers needed to be more adept at handling the code to deliver both the functional and performance requirements of a business application. Coding proficiency was also highly prized with the hero worshiping of fire fighting hackers on hand to fix an application, rather than address deficiencies in the development process and project management. Such deficiencies included robust quality processes and testing adherence, strong project management, good tooling support, clear and succinct handover policies (at all stages), and lack of sufficient investment in developer research, training and education. Despite the maturity of the industry and market, many of the stated deficiencies still remain within organizations today.

Developer spotlights

One doesn't need to look far to understand what makes a developer tick and what support is required. Zereturnaround (a hub of Java technology experts) has released its 2012 developer productivity study (<http://zereturnaround.com/labs/devprod-report-redux-ides-and-preparation-for-2013/>) and it makes for illuminating reading as to the tools and frameworks popular amongst the Java community, what stresses them out, the amount of code that actually gets written in a working week and their efficiency profile. There is also good guidance out there to help managers better support developers in general. For example: ensuring that developers feel valued, have enthusiasm for their work, have the right tools, and are encouraged to challenge constructively. Unfortunately few employers invest in training developers to update their skills or give them sufficient time within their work schedule for knowledge research. Instead the onus is put on the developer to use self help books, online training or book their own courses.

Today's skill conundrum

The skills required of today's developers read like the perfect employee wish list: Creative, but with engineering like process application and discipline; business savvy, open minded team player; strong estimating skills and excellent time management; domain expertise with wide ranging up-to-date technology and programming knowledge; skillful and good looking. Okay, strike that last feature, but you get the gist. Developers have always needed to be more worldly and current with the languages and scripts being widely used in the market place. They have always needed to be in-tune to common practices and cognizant of key development issues, application architectures and coding strategies. In the case of some of the current coding knowledge required, one software industry insider sums it up well: *“The modern day savvy developer has to be a polymath. They can't be pigeonholed into a platform or language. Mobile is everything right now (the numbers for mobile usage are staggering...kinda like the early days of the web). The desktop is dead. In mobile, there are three platforms that every developer has to be proficient in: iOS, Android, and HTML5/mobile web. The backend systems serving these apps has also changed radically...and developers need to be proficient in Rails, PHP, and ASP.NET, of course, but also in new stuff like Meteor and Parse. It can be daunting, but for any real developer, it's tremendously exciting.”* Whilst there are new software technologies enticing the developer audience, there are the legacy ones that continue to run business critical services and processes. Cobol applications and mainframe platforms alike, are still well deployed within many organizations. In fact, mainframes not only continue to serve a vital role in delivering highly utilized transactional systems they have evolved to deliver new levels of efficiencies and to offer new capabilities and services. The technology platform's applicability for a broad range of business services, offers opportunities for those mindful of the aging skill set and the hole it will leave behind in many businesses when those with the skills exit the workforce. But whether it's a mistaken perception or not: it is not sufficiently viable for a developer to stay in a bubble of coding knowledge.