

A culture of curiosity: how to grow, nurture talent in the post-PC generation

 By [Anice Hassim](#)

30 May 2012

We have become disconnected from the "why" of things. Today, everybody seems to be pursuing the making of money and, if we are all trying to produce money, you tend to wonder who is producing anything useful.

You can see this attitude at play all over South Africa as politicians, businessmen and community alike squander precious time and resources that will ill-serve the next generation.

The fact that Juju is a national joke and not in jail, speaks volumes.

Now what has a political rant got to do with Dev Culture?

Well, it's got to do with principles that somehow have been lost over the last couple of generations. Dev Culture reinforces meritocracy and relevancy... it accommodates both skill and experience - after all, code doesn't lie.

Dev Culture has less to do with manufactured short-term goals and more to do with a desired end state over a known arc of time. When commissioned to undertake a new project, the next thought that pops into a developer's mind after "Cool" is "When do I need to do this by?"

Innovation, insight, knowledge

Innovation requires insight, and insight comes from knowledge - a thorough understanding of the objective of the product. The further the layers of separation between the executives who direct the project and the hackers whose job it is to create it, the longer the development cycle and the less effective the product.

We have all heard the tech legends: how the core principal of Facebook was coded in a weekend, or how Woz wrote circuit board magic in four days for Jobs to create Atari's Breakout. Maybe, those overnight 'flukes' aren't exceptions. Maybe, when you invite hackers to the table, you create opportunities for insight to be translated into action.

Dev Culture is entrenched in the solving of a problem. You need them in the room, because their essential animating force is a *curiosity* that will empower them with enough data to hack the world around them.

Change agents looking for a mission

In effect, they are change agents looking for a mission. The more you keep them away from your organisation's centre of power, the more you are discarding a strategic company resource.

Yet, all too often, when it comes time to flesh out strategic projects, developers are invited in when the discussion is all over. In effect, they are handed the spec sheet and told, "Go and build this."

Of course, technical innovation doesn't track executive-suite comfort zones. So the scenario where the developers chuckle over the choices of approach and technology, shake their heads and build it anyway is an oft-repeated one.

Now I can imagine many eyebrows being raised and thoughts of "He wants me to invite Kenny from Dev to offer his opinion! Nuts!"

Studio boy bands vs indie bands

And if Kenny is anything like the typical developer in SA, you'd be right. SA doesn't really encourage a developer culture and, unfortunately, the BSc graduates churning Java code for Nedbank or Absa aren't really living in a Dev Culture environment. They are more akin to studio boy bands, whereas Dev Culture needs the indie bands, of which there are woefully too few in SA.

But we can create many. We just need to understand geeks a little better and, in the application of some basic principles from yesteryear, you can achieve some interesting results.

I don't want to suggest that somehow building an app dev studio is a simple formula that you can apply to every situation. Human beings are all different and each tribe of coders may gel together in different ways. But in the main, encouraging curiosity and innovation has some standard requirements.

So how do we get some respect for developers?

1. Encourage a culture of listening

Nurturing talent means allowing your coders to question and explore every angle - even if it's not entirely relevant. You can't pay lip service to allowing freedom of thought, just like you can't pay lip service to innovation... and listening implies taking action.

2. Change the concept of *reward* away from money to opportunity

Better opportunities will automatically bring more money, but money should never be the goal. Developers who create the most innovative products tend not to be animated by money, in any event. Legend has it that they are too hairy, smelly or silent to ever have much need of it anyway, other than in the occasional stripper denominations of local currency.

3. Reward asymmetrically

If your approach is grounded, you will find that your studio will be a mix of young and old, self-taught and tertiary-schooled, all of whom will ripen at different rates. Don't just give seasoned developers the plum projects - aerate them with enough "safe" opportunities for juniors to stretch themselves. You will be surprised at the results.

4. Create a playpen

This could be the same physical space as your office, but let Cinders out to play at the ball after dark. Without too much prescription of what goes on, allow the space and context for your developers to explore and teach each other. It doesn't look like it, but developers are natural teachers and they love to share. They tend to not do that in the

workday due to the pressures of client work, but if you create zones of time and space that allows them, they will discover and innovate together.

5. Keep the suits far (if you have a continent between you, that's a start), far away

Never let developers talk directly to a suit. Always have an interpreter, a Geek Whisperer, if you will. This is also the most important job in the entire mess you call your dev team. For example, when I suggest that you have the developers offer an opinion on the strategic imperative you are investigating, I don't mean that you should ask the ex-developer executive (you know, the one who develops the tech spec) what the geeks would think. I mean, have him take a walk down to the zoo, brief the team properly on how to behave, and then invite them (or a duly appointed team representative) into a brainstorm with everyone else. So the geeks become first-tier observers of all of the facts at hand.

6. Beware a geek with religion

Any person you appoint to overseeing any element of your development requirement should be asked to point out the good in another hot platform. If he insists that all problems can be addressed by Java 1.6, he's not a geek. No real geek has religion. How do I know? Look at all the new Apple developers! :-)

7. Don't just say you are transparent, *be* transparent

Schedule regular navel gazing to understand failures AND successes. All too often, we never spend enough time on questioning, "Well, why *did* that work?" And remember to tell Kenny why he didn't get to work on that project. It helps him understand what to do to get there.

8. Hire diversely

Having a good gender and cultural balance in your team gives you a full spectrum perspective and helps you formulate a thoroughly interrogated approach to issues that arise in the creative cycle. App dev is the creation of a human experience, and therefore requires imagination and personality. A mix of personalities, backgrounds and sensitivities in the studio allows people to contribute authentically.

ABOUT ANICE HASSIM

Anice Hassim is the CEO, Founder and majority shareholder of the immedia ecosystem. He is a speaker and educator around digital trends, strategy and marketing, and helps brands better understand and negotiate the digital space. Contact Anice on tel +27 (0)31 566 8000, email him at anice@immedia.co.za and follow him on Twitter at [@anicehassim](https://twitter.com/anicehassim)

▪ [Design Indaba 2016] An Indaba dabba doo - 18 Feb 2016

▪ [BizTrends 2016] Heading for a 'privacy apocalypse' - 2 Feb 2016

▪ A culture of curiosity: how to grow, nurture talent in the post-PC generation - 30 May 2012

▪ [2012 trends] Your friend is literally the trend - 19 Jan 2012

▪ Is your digital agency about thought creation or regurgitation? - 18 Oct 2011

[View my profile and articles...](#)

For more, visit: <https://www.bizcommunity.com>