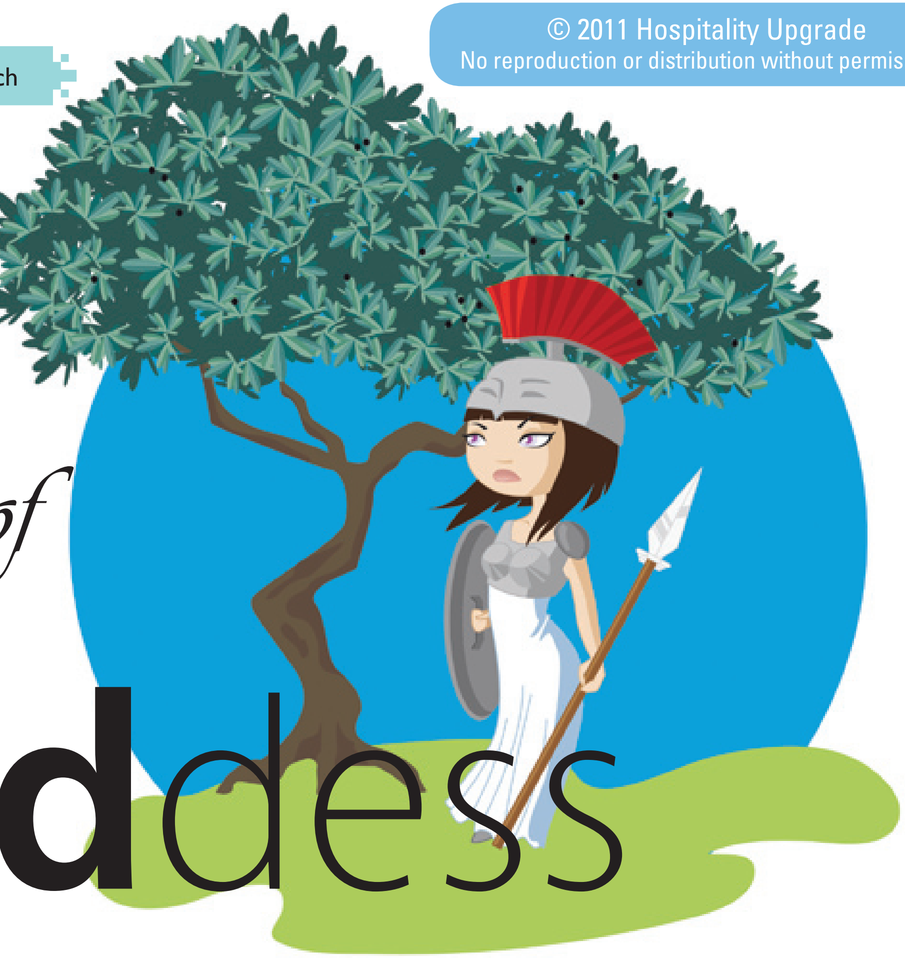
By Michael Schubach

In search of a missing Goddess



Those of you not all brushed up on your classical mythology may not remember some of the salient characteristics associated with the goddess Athena. Well, there's no time like the present for correcting these little oversights, so here we go: First on the list of fun facts about this particular deity was her residence. She and the rest of the family called the summit of Mount Olympus home. Olympus is the highest mountain in Greece, some 9,570 feet tall. We're not talking about some rolling grassy highlands here – we're talking a location way up in the sky – essentially on a cloud, as it were. (See if you can guess where this is going…)

The second interesting point on the resume is her area of influence: she is the goddess of wisdom. (Perhaps I should say that she was the goddess of wisdom. In what can only be described as an early Christian popularity downturn, the entire Olympian pantheon went into recession, downsized and the officers were all laid off.) Finally, and central to my point (and I actually do have one) is that Athena wasn't born in the typical fashion – in fact, her mother wasn't even present at the event. Details are a little sketchy here but in what had to have been an amazing finale to a whopper of a headache, the goddess sprang fully formed from her father's head. (Kids, Athena was a professional goddess; please don't try this at home.)

It strikes me that those who manufacture software for a living are currently searching for a new Athena. They're looking on behalf of customers who seek the telltale attributes of divinity: they want a great deal of wisdom concentrated into a single cloud-based apparition that emerges fully formed at initial delivery. Developers would love to accommodate this search; they realize that time is their enemy when customers are poised to purchase. Being late to market can be like not coming to market at all, so all this Athenian lust is easy to understand. Despite that wistful appeal, a next generation of fully formed apps with full legacy feature sets fully intact is a tall – perhaps even mythic – order.

Is there anything wrong with conventional children as opposed to full-grown skull-leapers? Nothing really, except that for the first few years your typical mother-borne offspring are a little slow on the uptake. They have no fashion sense. The most interesting place they go is the bathroom. They don't add much to the conversational landscape. Their parents and grandparents are the only ones to see their diamond-in-the-rough potential, overlooking their temporary status as extremely short people with serious ambulatory and drool production issues. Despite their drawbacks, I really liked the appreciative comment made by a child development specialist who noted that the educational process is not about upgrading children from dumb to smart (his concept, my word choice). He opined that children are essentially what they are from the very beginning and that education is the act of investing tiny tots with a vocabulary and relevant life experiences.

It strikes me that those who manufacture software for a living are currently searching for a new Athena.

A similar process and reality govern software development. There is no leaping fully formed from old legacy applications, resplendently cloaked in dramatic new technology, born and feeling completely at home in the cloud. Even the very best of new cloud-based applications lack vocabulary and relevant life experiences until they've seen a bit of the real world. Over time and with exposure, they become more informed, more practical and better able to respond to a wider variety of demands, perhaps even wiser. Don't misunderstand me: I believe that software engineers can invest many appealing attributes into any new product offering as it comes into the market, but maturity comes with time and experience. I personally don't believe there is any way to invest the attributes of having been around the block a couple of times without having been around the block a couple of times.

GMS/PMS software could be front and center in the nature versus nurture battle. Its nature is what the development team imbues into the application – the genetic material, as it were. Nurturing produces the wisdom that comes from the various installations that the application goes through – life experiences, as it were. The application grows and takes on new perspectives, specialized interests and a certain expertise within chosen areas of concentration. I have always contended that it is fairly easy to spot GMS/PMS heritage, to determine the background of the application. You can spot those who grew up in a luxury environment and easily differentiate them from those that came up through the motel ranks. Some apps appeal to the small and boutique-ish where others were born or bred to serve the supertankers of our industry.

But heritage isn't always a virtue: no one invests in old technology. Consumers want the state of the art – applications so new that they're barely out of the box. Applications that are being re-envisioned and reinvented to leverage all that newness of the cloud won't be successful in version 1.0 trying simultaneously to leverage the full content of their predecessors. Nevermind if sometimes the buyer isn't really certain if or why cloud-based computing is an appropriate solution to his or her business challenge; cloud readiness is the new baseline of technology. (One Big Blue executive who I am quite certain doesn't wish to be cited in this article told me that "the cloud" – in quotes – is "a marketing ploy." This wasn't in 2006, it was last month. Nonetheless, they clearly recognize the need to be dead center in that marketing ploy along with the rest of us.)

Developers may have to settle for bringing up baby the old fashioned way, investing in the little one with as much experience and exposure as possible.

New technology usually demands a new methodology and sometimes those paradigm shifts can be so profound as to require developers to start over at the bottom to work themselves heavenward. Athena may have climbed out of Zeus' skull but developers may have to settle for bringing up baby the old fashioned way, investing in the little one with as much experience and exposure as possible – as fast as possible. It's a tough race but it goes to the swift, so get out there and procreate… even if it means employing the more conventional method of delivery.

MICHAEL SCHUBACH *is a long time contributor to Hospitality Upgrade. He can be reached at michaelschubach@mac.com.*