



Capital Area Cloud Computing User Group
<http://www.capcloud.org>

Cloud Innovations

Issue #3



Thoughts from a Cloudy Mind:

My Pair beats your Ace

Our last user group meeting way back in August featured Kevin and Jimmy from Thycotic Software giving a presentation on the advantages of Pair Programming.

Pair programming is essentially 2 programmers operating side by side for 8 hours on one computer. They really try to live up to the adage that two heads are better than one. While working in this paradigm you will share ideas with your coding partner and they in turn will challenge your assumptions. One way to look at this is Master and Apprentice coding. However I kind of envision it like each partner will take turns driving (coding) and the other will watch over their shoulder like a back seat driver waiting to offer suggestions or point out mistakes.

Some of the advantages put forth by the Pair Programming Paradigm are:

- a. Shared Code Ownership. There isn't just one expert on this section of code but at least two.
- b. Deficit Reduction: Bugs will be caught sooner as the person looking over the coders shoulder can help catch syntax and simple logic errors, or even interject and propose a cleaner way to do the same thing. Kevin and Jimmy say that Pair programming reduces defects by 15%.
- c. Pair programming limits breaks in the coding day (no tracking the Dow, checking your latest IPO stock offering price, checking LinkedIn.com or your private email for job offers) as you have someone sitting right beside you all day. This reduces coding time and with the reduction of bugs you can get your product into production faster.
- d. Pair programming fosters stronger team environment as programmers are forced to interact with each other.

- e. They claim that over time developers prefer pairing than going solo.
- f. That programming becomes more enjoyable because you are sharing.
- g. Better evolution of coding standards.
- h. Pair Programming offers constant code review. While every company sings the praises of and promotes code reviews, come crunch time code reviews are the first thing to fall by the wayside. It is estimated that 80% of developers don't perform code reviews and I suspect the percentage is even higher.
- i. Knowledge Sharing – This to me would be a big plus for pair programming. You end up sharing the knowledge of the task you are working on and share monotonous tasks.
- j. Disagreements can lead to clarity of purpose

There are downsides to Pair programming

- a. Two cost more than one. You are going to have to sell the client on the benefits of pair programming as no matter how you cut it, two programmers doing the same task will be perceived to cost more than one. To convince the client you will need to show that pair programming takes 15% less time, has fewer bugs and rewrites, enforces coding standards and shares knowledge which is invaluable for future upgrades.
- b. Developers are opinionated and some never get along. Gee as a developer for 20 years I have never come across an opinionated developer (gag – am choking on that lie as I wrote it),
- c. Skill differences in the pair. If both programmers don't bring something to the table then one will just be riding the coat-tails of the other.
- d. Experience mismatches and age mismatches – some dinosaurs refuse to learn anything new or do something different – some whippersnappers straight out of school think they know it all but real life programming and the realities of business are very different than school projects.
- e. Good Hygiene – need I say more? Nobody wants to sit next to a caveman who hasn't seen a bar of soap or a tooth brush since the last ice age for 8 hours.

What makes up a good pairing?

- a. Overlapping skill sets. Don't pair the UNIX guy with the Visual Studio guy. Oranges and Apples don't mix well.
- b. Have somewhat different expertise. Don't put two database guys together to design a page, instead have a database guy work with a GUI programmer.
- c. Encourage different point of view – but don't put two zealots together, need interaction and agreement not a religious coding war.
- d. Each person must be flexible and open to new ideas
- e. Willing to allow their partner to explore or propose a different solution – there are often many ways to get to the same place.
- f. Need the ability to adapt to different personalities

The most important part of Pair Programming, the so-called “Corner Stone” of Pair Programming is **COMMUNICATION**. As with any couple, each person must be able to listen objectively, must be able to articulate (no grunting), and must be able to debate without casting judgment or offense.

Communication is the problem that broke up my first marriage (I grunted and she was a break wall – as in unmovable) and I can see the need for counseling or a mediator for failed Pair programming couples. Hmmmm – if pair programming becomes the norm maybe I could explore becoming a Pair Programmer mediator. At a \$120 bucks an hour it could be pretty lucrative.

After listening to the presentation I came away not sure I could make it as a Pair Programmer. I don’t have the stamina to code 8 hours straight anymore unless maybe the company provided cases of 5-hour energy drink. Do they still make Jolt Cola?



THE INAGURAL TECHGATE 2011

Did you go – if not you missed a well-run and informative event championed by our own user-group president Andy Zhang. Well done Andy!

I won’t review all the sessions I went to but want to mention the first one I attended first thing in the morning with eyes half closed and coffee in hand. The presenter was a very energetic Yung Chou, a Microsoft cloud computing evangelist, who woke up a filled to capacity room by dropping cloud sound bites left and right claiming he is dreaming of a Clear Cloudy Day. His mantra was “Capacity on Demand” and don’t waste or lose money by not meeting demand or over committing resources. He claimed that for every 100\$ in computer resources, 92\$ is wasted as only 8\$ are needed to handle normal traffic. This is where the cost effectiveness of the cloud with its Capacity on Demand shines brightly.

Yung went on to explain 5-3-2 which he uses to define what Cloud Computing is

5 Essential Characteristics of the Cloud

- a. Self Service
- b. Global Access
- c. Resource Pooling
- d. Elasticity (ability to shrink and expand based on demand)
- e. Consumption based Business Model

3 Delivery Methods

- a. SaaS {Software as a Service}
- b. PaaS {Platform as a Service}
- c. IaaS {Infrastructure as a Service}

2 Deployments

- a. Public Cloud
- b. Private Cloud

If interested explore this link: <http://blogs.technet.com/b/yungchou/archive/2011/03/03/chou-s-theories-of-cloud-computing-the-5-3-2-principle.aspx> for more information on 5-3-2.

I would strongly suggest you go and visit Yung's blog which is stuffed full of useful Cloud information at this link: <http://blogs.technet.com/yungchou/> Yes Yung is Microsoft but the blog has tons of useful information and links and well worth the time exploring.

You might also want to circle the date of November 29 where he will be at TechNet 'Accelerating Your IT Pro Career with Private Cloud' in Chevy Chase MD. <http://technetevents.com/> Looks like a good one.

Other sessions of interest for the cloud were done by some upcoming presenters for our users group. Steve Bobrowski of Salesforce gave a presentation on 'Multitenancy Magic: What Powers the Database.com Cloud Database Service'. Dan Harding of RightScale gave a presentation titled 'Managing servers in the Cloud with RightScale'.

All in all it was an enjoyable conference on a rainy Saturday and I look forward to the next one – especially if Andy plans it as I know it will be well organized, informative, and therefore very successful.

So in ending this newsletter let me leave you with these two thoughts:

Computing should be a utility like electricity or water - You should only pay for what you use.

Computing is going full circle. First there were mainframes where you paid for data storage and retrieval costs and processor time, then to PC's where you paid for everything, and now to the Cloud where you pay for data storage and retrieval costs and services used.



Storm Clouds:

Steve Jobs stepping down from Apple. Here is an interesting read about one of the most successful CEO's ever. Not so much on the cloud but a good worthwhile article.

<http://www.forbes.com/sites/ericjackson/2011/09/19/the-top-ten-lessons-steve-jobs-can-teach-us-if-we-well-listen/>

Cloud Computing – The Risks and Benefits of Cloud Storage: {Pertains to Small Businesses}
<http://smallbiztechnology.com/archive/2011/09/cloud-computing-the-risks-and-benefits-of-cloud-storage.html/>

Google Docs Outage: Offline Editing Needed, Now:
<http://www.informationweek.com/articles/231600978>



Clouds on the Horizon:

Because I like Green:

Google Says Gmail is Eco-Friendly, Greens Your IT: <http://www.tomsguide.com/us/google-gmail-datacenter-enterprise-email,news-12461.html>

Google Details, and Defends, Its Use of Electricity:
<http://www.nytimes.com/2011/09/09/technology/google-details-electricity-output-of-its-data-centers.html>

Member Benefits

You can receive discounts for the following great conferences as our group member. The priority code for our group is: CapCloud11. This will give you a **25% discount** off the prevailing rate (so registering during the early bird discount will yield an even better discount):

Enterprise Architecture Conference, Nov 7 - 8, McLean, VA
<http://goveaconference.com/events/2011/information/saag.aspx>

Cybersecurity, Dec 8 - 9, Washington, DC
<https://www.eiseverywhere.com/ereg/newreg.php?eventid=28420&>

Secure Enterprise Mobility, Dec 8, Washington, DC
<https://www.eiseverywhere.com/ereg/newreg.php?eventid=23963&>

Until next time:

Mike Maddox, {The Head in the Cloud Guy}

I can be reached at CapitolCloud@gmail.com