GET GLOBAL ENTERPRISE TECHNOLOGY

(http://get.syr.edu)

GET (HTTP://GET.SYR.EDU/)   IMMERSION EXPERIENCE (HTTP://GET-IMMERSION.SYR.EDU/)

EUROTECH/ASIATECH (HTTP://GET-EUROTECH.SYR.EDU/)   GET @ SU (HTTP://GET-SU.SYR.EDU/)

HOME (/)   WHAT IS GET? (/WHATISGET/DEFAULT.ASPX)   IS GET RIGHT FOR ME? (/ISGETRIGHT/DEFAULT.ASPX)

GET PROGRAMS (/PROGRAMS/DEFAULT.ASPX)   TESTIMONIALS (/TESTIMONIALS/DEFAULT.ASPX)   COMPANIES (/COMPANIES/DEFAULT.ASPX)

UNIVERSITIES (/UNIVERSITIES/DEFAULT.ASPX)   CONTACT (/CONTACT/DEFAULT.ASPX)

SEARCH

>

GLOBAL ENTERPRISE TECHNOLOGY

ENTERPRISE THINKING
Read posts from GET Program Director Professor David Dischave.

A WATERFALL SYSTEMS DEVELOPMENT METHODOLOGY … SERIOUSLY?

9/17/2012

As many of you know I teach a systems analysis and design course, IST352, Analysis of Organizational Systems and a Systems Analyst for a number of years so I'm in familiar territory on the subject of systems development methodologies. What I find interesting is the pervasiveness of myths that don't seem to ever die of old age or that can't seem to be demystified by using empirical reasoning; especially in light of the overwhelming number of these little things we call facts. Every semester I cover systems development theories that include the guiding principles supporting repeatable systems development methods framed by the Systems Development Lifecycle. Like clockwork, early in the semester, there will be at least one student that will ask if I plan to teach the "Waterfall" model.

As an IT professional and academic for about 40+ years I have heard many myths about the IT industry. So the student's use of the word "waterfall" wasn't a surprise and it certainly wasn't the first time I heard the word used in the systems development context.

But what continues to surprise me is why the word "waterfall" is still being used to describe a systems methodology that doesn't exist and why the creators of systems development methods use it as a source of comparison.

My first encounter with the mythical "Waterfall" methodology was in the early 1980's. As the director of a systems development department at a fortune 100 company I received a visit from a salesman from Coopers & Lybrand (C&L.) As the director of a multi-million dollar IT department you can imagine how many people wanted to sell me stuff. The C&L sales rep, attempting to sell me a methodology called Summit-D, asked me what systems development method we used. So I shared with him that our shop standard was the Systems Development Lifecycle which entails doing a bunch of tasks and activities and we grouped in five phases: planning, analysis, design, implementation and maintenance & support. I went on to say; you know, we subscribe to Winston Royce's work. Before I could finish my description, the C&L sales rep quipped right back with "ahhh, you are using that old obsolete Waterfall model. Oh, by the way who is Winston Royce?" Not waiting for an answer, he added, nobody should use the Waterfall model anymore. "You see, Dave, once you complete a phase it is frozen. I asked, what is this "waterfall" thing that you keep referencing? The C&L sales rep said it is the method almost every company uses, where the phases are worked sequentially i.e. in lock step and no phase can start until the previous one finishes. He went on to say that all deliverables produced in a phase were frozen once that phase ended. He said each phase cascades down into the next, you know, like a waterfall. See, it is called waterfall because water just can't flow up hill."

You can imagine what I must have been thinking. At that time, I'd been in IT application development for 20 years with five different major corporations and I was introduced to waterfall - by a salesman. I had never heard of a waterfall method. In all of the years and all of the places I worked and all of the conferences I had been to, I didn't know of any organization that built systems this way. Yes, I did try to dissuade the sales rep that we were not using any waterfall, watering hole, water table, water can or water cooler methods but as you can suspect sales folks can't sell you a solution if you don't have a problem and he was really trying to create a problem.

Well of course we iterated and revisited tasks in previous phases. There many good reasons to revisit completed artifacts from any systems development phase. And yes, even in the early 1970's, we built prototypes, skipped steps and did what had to do to deliver a solution. What are you supposed to do when you learn that you are not producing a quality product, one laden with defects, or your solution doesn't solve the business problem, move the initiative forward anyway? That would be unprofessional, but I suppose it is possible that a few have tried such an approach; but I suspect those responsible didn't get to keep their jobs. I have read Royce's work many times. What I find fascinating is Royce never used the word waterfall nor did he support the notion of freezing deliverables at the end of each phase. Yet, to this day many people keep referring to his systems development lifecycle as a "waterfall" method, one that is rigid, burdensome, laden with documentation, slow and inefficient. In fact, just the contrary, Royce advocated that the systems development process be fluid, flexible and iterative,

where the tasks and activities of any phase can be done in any time increment and in any order; essentially, systems development work can be scheduled and delivered commensurate with the problem being solved. I will also say that Royce and I agree that systems should have sufficient documentation. My argument on this is, how do the systems designers communicate complex processing and data relationships to the systems builders? Use extra sensory perception (ESP) or verbally … seriously? I think you get the idea.

For those information technology professionals and other interested parties I encourage you to do some research. Look for the seminal work that introduces a "waterfall" methodology to the discipline. If you find it please share it with this community. I would like to read it, but in the meantime please read Winston Royce's, "Managing the Development of Large Software Systems."[i] I think you will find Royce does a nice job proving that the "Waterfall" systems development methodology is indeed a myth.

---

[i] Royce, Winston W, "Managing the Development of Large Software Systems" Proceeding of the IEEE, TRW, August 1970, Pg 1 – 9

---

6 Comments   **GET Site**   **1** **Login** ▾

♥ **Recommend** **2**   ↱ **Share**   Sort by Best ▾

Join the discussion…

Charles T. Betz · 6 months ago
Dave, when I was working at Andersen Consulting, we were very deliberately schooled in waterfall. "Stage containment" was a performance factor; having to revisit a previous stage resulted in demerits.

More to the point, if the customer made you revisit a stage because of "changing requirements," you could charge them more. So, if you are looking for a historical reference, try to get ahold of the Andersen Consulting "Business Integration Methodology" circa 1998.

I of course concur with the industry consensus that this is an utterly dysfunctional approach. But many thousands were schooled in it.
∧ | ∨ · Reply · Share ›

Carbonspeed · 7 months ago
Thanks for the information.
∧ | ∨ · Reply · Share ›

Drew · 7 months ago
Bravo. To the authors of the comments, read and absorb before writing lazy replies. Prof Dischave was an IT professional, that link doesn't work, quoting wiki as a credible source- are you serious.
∧ | ∨ · Reply · Share ›