WHY THE WATERFALL DEVELOPMENT METHODOLOGY HAS A VERY APT NAME

How often do you use terms or phrases without really examining what they mean? I thought of this recently when considering what it means to develop and deliver applications using the "Waterfall" development methodology.

If you work in mainframe IT, you've probably been familiar with the Waterfall development methodology for a long time because it's what your team has always worked with. In examining the terminology deeper, however, I realized just how great a name Waterfall is for what it represents—though what it represents really isn't that great at all.

The Risks of Dropping Down a Waterfall

Consider a natural waterfall. Being from Michigan, I picture the Tahquamenon Falls (above), but you probably picture another large waterfall or group of them, like Niagara Falls.

What you have is water heading inevitably towards a cliff. If you were to think about going down the waterfall, you would want to plan and prepare. This has been attempted with mixed success using barrels to go over Niagara Falls.

You have to plan for all possibilities when you're dropping 165 feet in a barrel: it's a project with a definite deadline—once you're in the water moving towards the drop-off, that's it—and potentially serious consequences. What if your planning doesn't support the outcome you expect? That is, to live. You only have one shot.

This is why we call the Waterfall development methodology what we call it. It's trying to plan and

prepare for all the possibilities that could occur once you drop your project over the cliff a year or more out from the project start date, hoping it makes a big splash and keeps floating—hoping your barrel doesn't splinter into pieces.

And when things do work out, you can't escape the reality of having spent countless hours planning for consequences that never occurred—time you could have invested in development and innovation instead.

The Benefits of Moving to Agile Rapids

What if you could break your year-long project with the Waterfall development methodology into a series of 12 shorter, monthly drops? This is what we call [Agile development](), and it would probably make you feel better about your project thanks to its shorter periods of planning that mitigate risk because they let you see ahead and plan for upcoming obstacles or the next drop.

When you're rafting down rapids as opposed to dropping down a massive waterfall in a barrel, even if you make a mistake and get dumped out, you have a life jacket to keep you afloat and you can get back in. You will have learned your lesson, you will be able to move on and move forward, and the consequences will be much less painful and impactful on your project.

Taking this further, what if you made the drops even more frequent but shorter? This would be like going from Class III to Class I rapids, where the water is even smoother, where you can really move around, be nimble, find mistakes sooner and cut risk. Rather than dropping month-to-month, you would be developing in Agile two-week sprints, where the flow is much smoother, much faster, and the drops shallower.

The benefits of [ditching the Waterfall development methodology]() and making more frequent, iterative code drops using Agile Development on the mainframe are easy to realize once you're doing it. But getting there is harder than it looks.

When you use the Waterfall development methodology, try to really consider what it represents and not just go through the motions because it's the status quo. Do you really want to continue to plunge over dangerous cliffs, or would you rather manage more frequent but far-shorter drops?