

LIVING AN AGILE LIFESTYLE—FROM SOFTWARE TO HOME IMPROVEMENT



After you engage in the Agile process, over time you realize the productivity benefits. After doing Agile for some time on the job, it hit me: I've already been doing Agile for quite a while outside of work.

When [Compuware](#) announced it was switching from [Waterfall to Agile](#), I didn't know at the time how it would transform the company. It wasn't a gradual process; it was a conversion that occurred before our eyes, and we faced many trials and tribulations as we adjusted.

Shifting to Agile was a change the entire company endured, and some refused to embrace it. Change makes people uncomfortable, but sometimes you just have to bite the bullet and charge ahead. That's exactly what happened.

Software engineers, QA, Scrum masters and product owners began to participate in [daily Scrums](#) and grooming and planning meetings. They began trying to understand story points and how to make stories (tasks) small enough to fit into two-week sprints—incremental and iterative work cadences.

After the grooming and planning meetings, the [two-week sprints](#) began and Scrum teams worked on the stories they groomed and planned for. At the end, the review and retrospective meetings took place. Since our first run over two years ago, this process has been recurring every two weeks.

In the beginning, it seemed like there were too many meetings, and I felt unproductive. This couldn't have been further from the truth.

Unlike Agile, Waterfall development treats analysis, design, coding and testing as discrete phases in

a software project that typically take months for each phase. In Agile, these processes are all completed in each sprint.

After you engage in the Agile process, over time you realize the [productivity benefits](#). After doing Agile for some time on the job, it hit me: I've already been doing Agile for quite a while outside of work. I then realized Agile was the right direction.

Making Sense of Agile at Home

Before I became an avid home fixer upper, I didn't even want to think about home projects. They seemed like huge endeavors that would take too much time to complete, until one day I realized I shouldn't look at the big picture. I needed to break down the job into smaller tasks.

My house was over 20 years old and its four bathrooms were outdated, but I didn't want the expense of hiring a contractor to complete the job. When I viewed the house project through the scope of lots of little tasks that could be tackled separately, I took my first step to start the project.

Sounds pretty Agile, right?

My first task was to purchase the sink, toilet, light fixture and towel bar for the smallest bathroom, which didn't need new flooring. The next five tasks were to demolish the bathroom, paint the walls, replace the trim, install the light fixture and handle all other bathroom necessities.

It was a huge sense of accomplishment once this project was done. I completed the other three bathrooms in a similar manner, which involved even more tasks, like replacing the subfloor and re-tiling the floor and walls. Unbelievably, I finished the last bathroom in two weeks—just like a sprint.

I never would have started the project if I didn't break it down into smaller parts and concentrate on completing the task at hand. Any other way would have overwhelmed me.

Agile Makes Sense

Waterfall reminds me of how I viewed home projects in the past: [one huge endeavor](#) that would take too long to complete. This thought process easily makes one stressed, overwhelmed by a sense it could never be accomplished. Thus, most people wouldn't even attempt to start the project.

Agile makes sense to me. Adopting an Agile process allowed me to complete the projects by breaking them down into manageable tasks. I had a long term vision but a short term view on the details. With Waterfall, you just can't get as much work done.

The common wisdom in software development is to avoid tackling a big problem and instead view the same project through the scope of lots of little problems tackled separately. That is, doing Agile!