

# Counting change

## A perspective on why the Waterfall process continues to dominate software development projects

While running some errands I stopped at a convenience store and bought a few items which totaled \$11.33. Once the purchase was rung-up by the cashier, I tendered a twenty-dollar bill. My change: \$8.67. I was delighted when the cashier began counting change back to me starting with two pennies, then a nickle, a dime, two quarters, three one-dollar bills and finally a five dollar bill, calling out the total already returned to me while handing me each coin and bill: "... eleven dollars thirty four ... thirty five ... forty ... fifty ... seventy five ... twelve dollars ... thirteen ... fourteen ... fifteen dollars ... and five is twenty dollars." I complimented the cashier on the good customer service for using this method of returning my change since we both could see instantly that I was getting the correct change.

With the proliferation of electronic cash registers in the United States over the past few decades, we have become accustomed to having our change counted back to us starting with the largest denomination, if it is counted to us at all. Since the amount of change appears on both the cash register display and the receipt, perhaps it is second nature to reach first for the larger denominations. However with mechanical cash registers, or for those establishments without one, counting change back to the customer starting with the smallest denomination is commonplace. More significantly, counting change starting with small denominations allows the customer not only to confirm they are getting the correct change, but also to perceive the value they are receiving from the transaction. In the transaction described above, I tendered twenty dollars and, starting with the value of the merchandise, twenty dollars of value was counted back to me.

The process of counting change represents a useful analogy for comparing the Agile and Waterfall software development methodologies largely due to its inherent aspect of customer service. Counting change starting with larger denominations seems to be comparable to the Waterfall methodology for these reasons:

- It is the more prevalent method in use today and has been for decades
- Its focus is on the remainder instead of the total
- Its effectiveness for engaging the customer and delivering high quality customer service is rarely considered

Conversely, counting change starting with smaller denominations seems to be comparable to Agile methodologies for these reasons:

- It is not the most prevalent method in use today
- Its focus is on the total
- Its effectiveness for engaging the customer and delivering high quality customer service arguably rivals that of its more commonplace alternative

A more significant aspect of this comparison lies in the compatibility of dissimilar processes and methods. Counting change where the smallest denominations are returned first does not have much in common with the method where the smallest denominations are returned last. It is difficult to conceive using both methods in a single transaction. Likewise, the Agile and Waterfall processes use such fundamentally opposite models that there is very little common between the two methodologies.

This leads us to one of the reasons why the Waterfall methodology, despite its weaknesses, continues to be the predominant process used within the software development industry: There is no easy way to apply small and incremental changes in Waterfall to facilitate a gradual and steady migration toward Agile.

To move from using Waterfall to using Agile generally requires a sudden shift from a place of relative comfort and familiarity to a place where most things are immediately and strikingly different, similar to the feeling we might

experience when, after decades of becoming acclimated to a process where change is returned to us starting with the largest bills, suddenly one day change is returned to us with pennies first. There seems to be no way to mitigate the abrupt change associated with moving from one method to the other. Though we eventually find that it works, we are unprepared for an immediate and strikingly different method of counting change, at first finding “largest bills last” unfamiliar and uncomfortable compared to the method we believe had come to be universally accepted and practiced.

This type of change requires a shift of mind to fully embrace the transformation. It may explain why many managers at corporations now using Waterfall hesitate to entertain the prospect of making a move toward Agile, as well as explain why some valiant efforts to incorporate only small portions of Agile elements into a largely Waterfall process fail to produce the expected benefits. Indeed, failure with the attempt to incorporate only selected Agile elements into a mostly Waterfall process may cause those involved to conclude that Agile does not work for their organization. According to Mary and Tom Poppendieck, authors of books on Lean Software Development:

Trust is not enough if teams and their leaders do not understand how to go about successful agile development. Starting agile development without the necessary technical discipline and tools in place is dangerous, and no manner of goodwill and trust can save a bad implementation of agile.<sup>1</sup>

The transformation of a software development organization from using Waterfall to using Agile requires effort and discipline. Though it may not be easy, it has its rewards, not the least of which is the ability for the organization to more easily adapt to and capitalize on a rapidly changing software universe. Once the transition to Agile is complete a software development organization can begin to define their contributions in terms both they and their customers can recognize jointly, counting value to the customer rather than simply counting change.

Jim McDonough – March 14, 2011

---

<sup>1</sup> Poppendieck, *Leading Lean Software Development – Results Are Not the Point*, 2010, p. 224.