Some Assembly Required: The Pitfalls of Using an Assembly Line Method to Create Software

Yunus Burhani, X by 2

In 1913 Ford Motor Company turned on the world's first moving assembly line. The idea was to speed up the time in which a Model T could be made, while lowering its production costs at the same time. That way Ford could, and did, charge less for their cars than other manufacturers, while paying their employees more on average than other factory workers of the day. Likewise, most companies' IT departments think that creating an assembly line for software makes writing and assembling software more efficient. Just like Ford did with auto parts, they divide various functions of IT into their own groups, such as architects, designers, developers and testers.

IT leaders often think that once the business requirements are captured, the technical team can create almost any software easily and without complications. They believe that the architects working with the business analysts can come up with a good conceptual solution. Once that conceptual solution is completed to everybody's satisfaction the work can move to the next "assembly" station for designers to design. After the design is completed, work moves to the developer's shop where they develop the software, and send it on for testing. It all seems very logical and efficient, at least until you take a closer look.

In the real world, it's not quite this simple, but that doesn't prevent some IT departments from deploying an assembly line approach. As noted, the idea of the assembly line came from the automobile industry. However, in the auto industry, the assembly line is only used to make copies of a concept car that has been fully vetted and is ready for production. The concept car is created by a group of engineers working very closely with many other distinct functional groups, including potential consumers, or in our IT department example, business people. It takes several iterations and prototyping before a concept car can be created. After its creation, it is thoroughly tested in a safety laboratory before it can be approved for production. Throughout the testing, the engineers and business people work closely to monitor and gauge the car's quality. Once it is complete, it can then, and only then, be sent to the assembly line for mass production.

Most of the time, when the business needs IT, it is for custom software development, just like creating a concept car. Therefore, the IT team and business team need to work closely together to achieve their goal. There are times business buys the software from off the shelves vendor but it always needs some customization or integration with rest of the systems in the enterprise.

A recent example of a project comes to mind. The goal of the project was to build data mart for an analytics team that included data scientists. The IT team worked very closely with the analytics team to understand their needs, and to create various prototypes along the way. Following this iterative approach, these prototypes were turned into a final product and now the analytical team uses a data mart custom built for their needs. They have also used this experience to begin to fulfill the needs of the other business areas, such as the actuarial and the wellness care departments.

As a contrasting example, in this same company there is an enterprise data warehouse that was built using the assembly line approach. It might in helping to understand the frustration of the business areas with this approach by looking at a very small enhancement. The Claims team wanted to add one new field that was added in the claims system. The business team forgot to put the word “add” in their requirement document and the data modeler, not fully knowing the purpose of the field, decided to put the data in one of existing fields in the warehouse. Since there was no collaboration with the business and everyone in the assembly line works on their own part of the line, no one questioned the decision until the customer reviewed the finished work and found it unworkable. So even though this is a very small change and the business was expecting it to be done quickly, because of the assembly line approach it caused rework and delayed the production timelines.

Auto assembly lines have been around a long time, but the approaches and processes involved in creating a concept car are a newer development. Though not as long, software development has been around for some time and has borrowed some aspects of the assembly line approach from the auto industry. Software developers would now do well to borrow the auto industry approach for concept car development – software development should be done in a tight team with full business collaboration along the way.

Yunus Burhani is a Senior Architect at X by 2 in metro Detroit, a technology consultancy focused on the practice of architecture for the insurance and healthcare industries.