



## A New Era of Software Development

There are incredible pressures on the software industry to produce better software at less cost. Today software literally runs the world. Yet software is still created using manufacturing techniques that were developed hundreds of years ago.

The 1990 book *The Machine that Changed the World*, by James Womack, Daniel Jones, and Daniel Roos, describes how automobiles were built in the 1890s:

Different contractors used slightly different gages to make their parts. Then they ran their parts through an oven which frequently warped the parts; thus requiring more machining. When the parts arrived in the final assembly hall, their specifications could be best described as approximate. Workers took the first two parts and filed them to make them fit perfectly. Then they filed a third part to make it fit the first two. And so on, until the entire vehicle --- with its hundreds of parts --- was completed.

Anyone in the software industry should see some striking similarities. The way an automobile was built in 1890 is still how software is built today. Manufacturing has since evolved. Is it time for the software industry to follow?

What was known as craftsman production was dominant in manufacturing until the early 1900s. The following are some of the characteristics of craftsman production. Compare these to the software industry today.

- Expensive, highly skilled workers progress through an apprenticeship
- Decentralized control with various skill sets operating independent of each other
- The use of general purpose tools to help accomplish the tasks
- No standards of how to do a task. No two tasks done exactly the same, by the same person or by two different people.
- Career path was to own your own business

Parts made in craftsman production are difficult to integrate, hard to test, contain unknown defects, take a lot of time to build, and are impossible to maintain. But once all of the bugs were worked out, customers received the customized product they asked for. Sound like software?

*AGILEAN shows organizations how to use lean office techniques to produce greater customer value in less time with fewer errors*

**AGILEAN CORPORATION**  
3325 126TH Avenue NE  
Bellevue, WA 98005-1363

425/877 881-0108 Office  
service@agilean.com

Visit us at [www.agilean.com](http://www.agilean.com)

## Mass Production

The early 1900s saw a new era in manufacturing. Henry Ford found a way to overcome the problems associated with craftsman production. Ford's techniques reduced manufacturing costs dramatically, while increasing quality and throughput. Ford called his innovative system *mass production*.

As noted in *The Machine that Changed the World*: "the key to mass production wasn't --- as many people then and now believe --- the moving, or continuous, assembly line. Rather it was the complete and consistent interchangeability of parts and the simplicity of attaching them together. These were the manufacturing innovations that made the assembly line possible."

Some characteristics of mass production:

- The work is defined so each person does one simple job, with no knowledge of how that step fits in the big picture
- Centralized control of all activities
- Required the creation of new overhead jobs (indirect workers) to *manage* the efforts of the *direct* worker
- The use of generalized tools to replace as much labor as possible.
- Disciplines and standards for the direct workers, but not the indirect workers.
- No real career path for the direct worker; the indirect worker's was to reach higher management positions

Parts made in mass production required less integration, testing, labor, time, and were much easier to maintain. The tradeoff was the lack of customization. You could have a car in any color you wanted, as long as it was black.

Software development does not lend itself to mass production. Software components (parts) are highly customized and built on demand. Mass production is more suited towards creating inventory by stamping out a lot of the same parts. Some software shops (especially larger ones) added the overhead of mass production without the ability to create interchangeable components that attached simply together --- requiring the continued use of craftsman production; with all of its challenges.

Software construction has more in common with manufacturers who produce low numbers of highly customized parts. These manufacturers were not able to take full advantage of mass production. They, like software shops, received limited benefits from part standardization and had to continue to use craftsman methods.

## Lean Production

The most recent era of manufacturing started in the 1950s. Just after World War II, Eiji Toyoda and Taiichi Ohno at the Toyota Motor Company, pioneered the concepts of what is generically known in the U.S. as *lean production*. Lean production is more about standardizing the process than the parts.

- Workers are trained to perform several tasks to maintain a continuous and regulated flow of work
- Workers are assigned to self-sufficient cells, with each cell responsible its internal work processes and the external processes that impact them

- Decentralized control of the activities within a cell, but centrally established timing, throughput, quality, and measurement requirements between cells
- Specialized tools for individual tasks within a cell to increase labor productivity
- Disciplines, standards, and metrics are established to govern all activities within and between cells

Lean production allows manufacturers to make fewer parts at the same time, and at less cost, than either mass or craftsman production. Lean production methods are used by both high and low volume manufacturers to speed throughput, reduce inventory, increase customization, and improve quality.

## Building a Software Factory

Lean production is the start of a new era in software development. Not every aspect of lean production applies directly to software. The following describes some of the steps to establish a lean software factory:

- Software development efforts are assigned to self-sufficient cells/teams of employees
- Employees are trained to perform more than one activity within a cell, with employees jointly responsible for all cell activities
- Cells form, change size, and disband as different assignments are made
- All software design, code, documentation, and test tasks within a cell are established such that the start/end development cycles for each cell are the same
- The deliverables at the end of each cell's development cycle are 100% complete, requiring no additional work
- Development cycle times are set relatively short; e.g., 30 days
- Disciplines, standards, and metrics are established to govern all tasks in and between cells

The software industry can achieve the same dramatic results from lean production that are being realized today by manufacturers of all shapes and sizes. The benefits of a lean software factory combine to reinforce each other, with the total being greater than the sum of the parts:

- **Employees** are happier because they are given succinct tasks, without interruption, with more control over the outcomes, and more responsibility to get the job done
- **Managers** are happier because software development is more deterministic, with greater visibility, tighter schedules, and higher productivity from all development staff
- **Customers** are happier because lean software is delivered quicker, it better meets their needs, contains fewer defects, and is less costly to maintain

###