



## **The Zen of Agile Management**

***By David J. Anderson with Mike Watson (contributor)***

### **Introduction**

We wanted to get this column started properly by introducing the background and concepts which we will be using over the coming months to present agile management for software engineering. This article is intended to boil agile management down to its bare essentials – just the principles and theory.

Future articles will be based on our experience implementing Feature Driven Development along with some Scrum and Extreme Programming practices with two teams in the United States over the last 3 years. Both these teams worked in the wireless telecommunications sector and both are part of Fortune 100 companies – Sprint and Motorola. Both of us work as managers overseeing software development on a daily basis. The articles we will be presenting in this column are written by active practitioners, managing software development, for actively practicing managers with real problems to solve.

So here it is, one article of theory, to be followed by many offering real hands-on experience and advice.

### ***Management Science***

Management science started with the British economist Adam Smith who worked during the start of the industrial revolution. He wrote in his seminal work, “The Wealth of Nations” that what separated advanced industrial economies from backward craft-based economies was “specialization”.

What really got management science into full swing was the work of Frederick Winslow Taylor, who worked for Bethlehem Steel around the turn of the 20<sup>th</sup> century. He created “time in motion” studies which sought to analyze the efficiency of specialist workers. Henry Ford was a fan of Taylor’s and he used his ideas to create the assembly line (in 1917) – which minimized inefficient time-in-motion and maximized local throughput for individual workers. Extreme

specialization was a boon for Ford as it minimized training and allowed new (often illiterate) employees to become effective very quickly.

The next major step in management science came in the 1950s when Peter Drucker taught the world that there was only one value for a product – the price which the consumer was prepared (or able) to pay. The idea of cost based pricing was dead. 20 years later, Michael Porter refined this idea and introduced the notion of a value-chain. The idea that a chain of workers or a chain of suppliers each added value to a raw material until it was eventually delivered to a consumer. The ideas of Drucker and Porter are essential to the notion of “value creation”.

Arguably the next giant leap in management science came in 1990 with the publication of Peter Senge’s “The Fifth Discipline”. In this Senge, teaches the world that businesses (and value chains) are complex systems. To maximize the effectiveness of a business, the system must be considered as a whole rather than the sum of its individual parts. This was the critical break from Taylor’s theories which existed under the assumption that optimizing locally (time-in-motion) would lead to a global optimum. All mass manufacturing was grounded in Taylor’s theory. Mass manufacturing was losing in the marketplace to an alternative approach from Japan – known in the West as “Lean”. Senge’s “Fifth Discipline” was to put the nail in coffin for the local optimization approach of mass manufacturing.

### ***The Theory of Constraints***

Senge wasn’t the first management guru to advocate systems thinking. In 1984, Eli Goldratt published “The Goal” which made just such an argument for manufacturing production. Senge describes a system archetype which he calls “limits to growth”. All “limits to growth” systems are eventually limited by some constraining force. Goldratt’s argument was that management should focus on removing that constraining force and growth would continue until another constraint started to limit it. Again management should focus on removing the constraint and productivity growth would continue. And so on for ever – a process of continuous improvement focused on the constraint which was limiting the growth of productivity.

## **The Essence of Agile Management**

Agile management is about management science applied to software engineering. Agile management is a minimal, lightweight approach to management. It uses a hands-off style. Agile management is highly delegated and provides empowerment at all levels. An agile manager should be empowered, his team leads should be empowered but above all developers and testers doing the real work should be empowered.

### ***Empowerment***

What does empowerment mean? Empowerment means that an individual takes responsibility for delivery of value-added output. That responsibility means that the individual must determine what is needed and working with others devise the

best means to deliver that output with the minimum of effort. Delegation and empowerment mean that fundamentally, individual contributors are responsible for the financial performance of a business.

### ***Value Creation***

For empowered employees to be effective, they must understand the concept of value creation. They must understand what their organization does to add value to the overall business. If the development group is part of a corporate IT division then the developers must understand what business value their applications development will deliver. They must then focus on delivering that value as effectively as possible.

Value creation is at the heart of management science. Effective empowered employees do not need to understand management science but they do need to understand the notion of value creation. Agile managers, however, do need to understand the fundamentals of management science and hence, it is worthwhile summarizing it here.

### ***Objectivity and Metrics***

Good management is objective. It is based in blunt reality. Good agile managers must know and accept reality. In order to make objective decisions, based in reality, a manager must know the current reality of a business. In order to do this, system metrics must be gathered. Without facts and data, there can be no management.

In the software world, people are skeptical of metrics. For years many metrics have been gathered. Two favorites are number of hours worked on an activity – the weekly timesheet – and number of lines of code written. Another popular one is number of defects per thousand lines of code. Many agile methodologists are openly critical of metrics gathering initiatives and put scorn on the activity.

Agile management will show that metrics can be effective. This is not so much to say that 30 years of management theory in software engineering is wrong but rather the project management aspects of software development have been misguided. They failed to focus on the most important thing – value creation. If software methodologists had been better versed in established management science then methods which focused on value creation rather than costs spent, e.g. hours worked, would have emerged earlier.

Donald Reinertsen in “Managing the Design Factory” describes ideal metrics as simple and relevant to the goal. That means they must measure value created. They should also be self-generating and provide a predictive (or leading) indication of performance. This can be achieved in software by tracking progress of fine-grained client-valued functionality.

Agile software engineering methods focus heavily on results or more specifically the delivery of valuable software as frequently as possible. In doing so, agile methods better reflect the teachings of established management science.

## **Focus**

If a manager is to deliver results, he must focus. The Theory of Constraints teaches managers to focus on the constraining factor which is limiting productivity. This is very powerful because it does two things. It focuses management time and attention where it is most urgently needed – and we've all worked for managers who didn't have enough time on their hands – and it focuses investment on the area which will generate the greatest return through the greatest increase in productivity.

By focusing on the system constraint, the overall productivity of the system is increased. The Theory of Constraints offers a method to achieve global effectiveness (value efficiency) rather than the local cost efficiency of earlier Taylor inspired management methods.

It is our belief that many early attempts at management metrics for software development were based in the local efficiency and optimizations inspired by Taylor. Agile management is about value efficiency inspired by thinkers such as Peter Senge and Eli Goldratt.

## **People**

Good management understands that the service model goes down rather than up. A good manager is there to serve his workforce. Bad managers believe that the employees are working for them. They have it backward. The employees are creating the value. The managers merely steer. Managers do not directly create value. Hence, the rightful role of the manager is to serve the staff, to get them what they need to be successful, by removing the constraints limiting their productivity.

## **Why go Agile?**

### **Agile Methods encourage Agile Management**

Agile development methods such as Extreme Programming (XP), Scrum and Feature Driven Development (FDD) have their priorities right. They follow principles from established management science focused on value creation. They focus on effective delivery of valuable software (value efficiency). They offer delegation and empowerment (often referred to as 'self organization') and encourage a hands-off management style. They lend themselves to the right type of metrics - metrics which measure value and are both leading and self-generating. Finally, and above all, they value people and appreciate that people make the biggest difference. In agile software development, the service model must work downwards – developers are left free to create great code at optimal speed.

### ***Agile Management is simply good practice***

Agile management is not new. The principles of agile software development are not new either. Agile management applies established management science to software engineering. Agile development processes are fully compatible with

these ideas. Management and agile development are definitely not mutually exclusive and don't let anyone tell you otherwise.

### **About the author**

**David J. Anderson** is the author of the recent book, “Agile Management for Software Engineering – Applying the Theory of Constraints for Business Results” published in Peter Coad’s series by Prentice Hall PTR in September 2003. He is Principal Consultant with VA Systems Professional Services. David was one of the team which created the popular agile development method, Feature Driven Development. He has introduced FDD at two Fortune 100 companies Sprint (a telecommunications operator in the United States) and Motorola. He writes the regular *Agile Management* column at the Borland Developer Network website and publishes his weblog at <http://www.agilemanagement.net/>. He holds a degree in Computer Science and Electronics from the University of Strathclyde.

Email: [dja@agilemanagement.net](mailto:dja@agilemanagement.net)

### **References**

- Anderson, David J., *Agile Management for Software Engineering – Applying the Theory of Constraints for Business Results*, Prentice Hall, Upper Saddle River, NJ, 2003
- Goldratt, Eliyahu M., *The Goal*, The North River Press, Great Barrington, MA 1984
- Magretta, Joan, *What Management Is – How it Works and Why it’s Everyone’s Business*, Free Press, New York, NY 2002
- Reinertsen, Donald G., *Managing the Design Factory – A Product Developer’s Toolkit*, Free Press, New York, NY 1997
- Senge, Peter M., *The Fifth Discipline – The Art and Practice of the Learning Organization*, Currency Doubleday, New York, NY 1990