

IDENTIFYING THE VALUES OF CUSTOMERS IN SOFTWARE DEVELOPMENT PROCESS

Jeff Widman, Western Washington University, jeff@jeffwidman.com
Stella Y. Hua, Western Washington University, stella.hua@wwu.edu
Steven C. Ross, Western Washington University, steve.ross@wwu.edu

ABSTRACT

Lean manufacturing begins with the first and highest priority principle - defining values in the eyes of the customer. This paper describes the process of applying this lean principle in software development processes with the example of a start up software company Palantir Technologies.

Keywords: Lean Principles, Customer Values, Software Development, Case Study.

INTRODUCTION

Lean manufacturing follows five basic lean principles [1]: specify value in the eyes of the customer; identify value stream and eliminate waste; make value flow at the pull of the customer; involve & empower employees; continuously improve in pursuit of perfection. Compared to mass production, lean manufacturing requires “½ the human effort in the factory, ½ the manufacturing space, ½ the investment tools, ½ the engineering hours, ½ the time to development new products” [1, p.13; 2, p.6]. To achieve such results, lean manufacturing has to start with the first and highest priority principle, which focuses on defining “what a customer values”, and “effectively communicating and operationalizing customer-defined value throughout the product development process” [3, p. 19]. In order to truly understand the values defined by customers, companies often use the Kano Model to prioritize product attributes into different groups, ranging from basic or must be factors to excitement or delighter factors that customers do not expect but will be delighted to have [4]. Toyota went as far as requiring design engineers to establish emotional connection with target customers. One Chief Engineer even “moved in with a young target family in southern California to enhance his understanding” of their lifestyle [3, p. 30].

Lean principles have been applied in software development to eliminate errors, reduce untested codes, and reduce cycle time [5, 6]. Lean implementation can help turn a chaotic, constantly changing development process, into a much more predictable, fast moving, and streamlined process [7]. In addition, lean implementation allows software development teams to reduce technical and market risks while achieving a faster delivery with a focus on creating the highest value to the customer [8]. This paper discusses the process of applying the first lean principle in software development processes. Understanding what customers want is especially important in software development. A Standish study found that 45% of features were never used and only 20% of features were used often or always [9]. However, in an analysis 30 software development experience reports published online or at agile software related conferences, Wang et al. found only 2 of them documented the process of identifying the value of customers [10]. Microsoft’s Windows 8 slogan “Reimagining Windows from chips to experience” reflects their efforts in engaging customers in the development process. Windows 8 team has an open dialog with end-users, developers, and information professionals about features and pre-release versions through online blogs, build conferences, and previews [11]. Both Windows 7 and Windows 8 teams are organized into feature teams which are responsible for the entire development process of features from design to specification to testing.

Traditional product development process begins with market data, surveys and focus groups. These tools often “do not provide the deep knowledge that establishes a concrete understanding of customer-defined value.” [3, p28] They are of little use in identifying performance or delighter attributes based on the Kano Model [12]. Hence, the market research outcomes lead to non-value added activities during the development process. In addition, customers do not always know what they want or how to articulate their values. Software companies go through extensive analysis to find out what all similar customers want or what values are shared among customers, and extracting

values from specific customer requirement [2]. What do they do when customers do not know what they want or how to articulate their values? Five Whys have been recognized as an important technique to “uncover a root desire or root need that is more important than the one with which the customer began” [2, p. 155; 5]. This technique begins asking “why?” when customer states the value, and follows up each answer with “why?” for four more times. The front-load question and answer process helps reduce the muda (waste) in the product design.

In the next section, we present Palantir Technologies and discuss how they identify the values of customers in the software development and testing processes.

CASE STUDY

Palantir Technologies begins with a well-known technology startup: Paypal. All of the online payment companies were losing massive amount of money to stolen credit card transactions. Paypal was the first company to use advanced heuristics to detect fraudulent transactions across hundreds of thousands of transactions. In 2004, several engineers who built this anti-fraudulent heuristics system decided to leave Paypal and start another company. Together with a few Stanford computer scientists, they founded Palantir Technologies - a company that built advanced data analysis software for government agencies. The initial brainstorming began in June 2004, and the first lines of code were written in January 2005. While the first product focused on government applications, they also added a second product line, “Palantir Finance”, to focus on hedge fund analysis and financial fraud. Palantir Government offers a platform for information analysis, such as detecting prescription drug adverse events and information security fraud within an enterprise. Their customers fall into three categories: Intelligence & Defense, Regulation & Oversight, and Cybersecurity. Figure 1 shows the screen shot of the flows tool that allows intuitive visualization of resources flowing between entities. Palantir Finance offers a comprehensive analysis platform for leading financial institutions. Figure 2 shows the the financial dashboard that allows a financial analyst to follow a configurable set of widgets delivering real-time financial data. According to CrunchBase, Palantir Technologies currently employs approximately 550 employees at their headquarters in Palo Alto, CA, and have received a total of \$301 million in venture funding since November 2009 [13].

Customer Development

Palantir began as a group of brilliant software engineers who understood how to manipulate large chunks of data, but zero knowledge of government intelligence agencies. They had to quickly learn what this new type of client wanted from the software. They began by working closely with target customers to rapidly iterate the feature set. According to Bob McGrew, their VP of Engineering:

We literally flew to a prospective client, showed them ten features in a live demo, and they generally liked one or two, so we'd toss out the rest. Painful, but we became very skilled at building just enough of a feature for them to decide whether they wanted it.

We also found that live demo's were worth the extra work over screenshots because it created a more transparent experience for our customers—which in turn allowed them to provide better feedback.

This was our primitive version of the Customer Development Methodology that Steve Blank created and Eric Ries expanded upon. While our process worked, we've greatly improved it after finding Eric Ries' blog. (Eric's blog has grown to 7000+ readers in the seven months since he started writing.)

This process of constantly talking with the customer not only resulted in a successful product, but also molded Palantir Technologies into a learning organization, which allowed Palantir to efficiently navigate the unique world of government contracts. Palantir also uses their own software internally to improve their recruiting, administrative, and development processes.

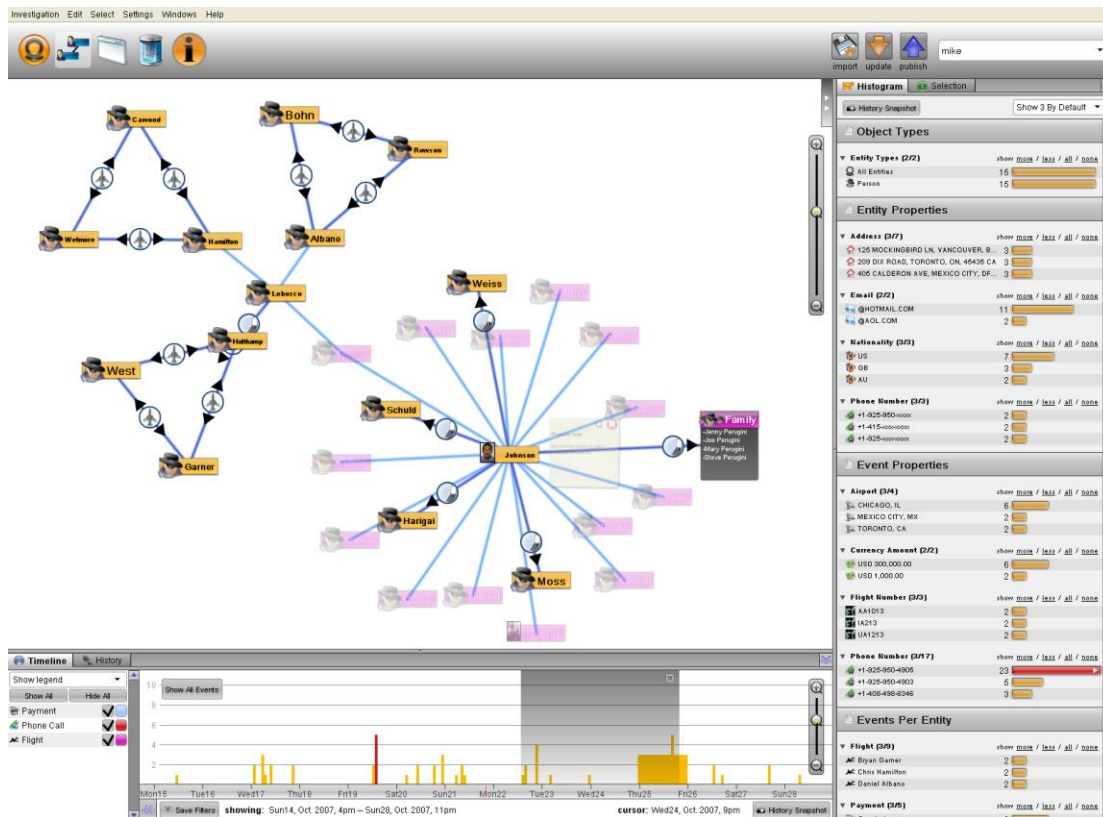


Figure 1: Visual Flow Screen Shot (source: blog.palantirtech.com [14])

None of the founders knew the intelligence business. Their first business contacts came from a VC firm and random friends. But after a while, this constantly changing product was something everyone wanted. After the feature set was starting to solidify, Palantir began offering free training before any contracts were signed. It allowed them to stay close to the customer, continually identify bugs, and also served as a great sales engine. When they finally began selling product, it wasn't long before everyone in the intelligence community knew of them simply through word-of-mouth.

In addition, Palantir doesn't have any soft-skills marketing or sales people. All their customer-facing employees have degrees in computer science or symbolic systems—generally from a top-ten university—and all can walk a customer through installing software or debugging something. It's helped them identify problems faster and build tighter relationships with customers, and they do pay attention to soft-skills marketing:

*Our marketing is mostly having really smart CS folks represent us in the field. And then we worked hard on the *feel* of our software. For example, most folks who work in intelligence agencies want to *feel* like James Bond. So we made our avatars look like a guy wearing a hat and a trenchcoat. [Bob McGrew]*

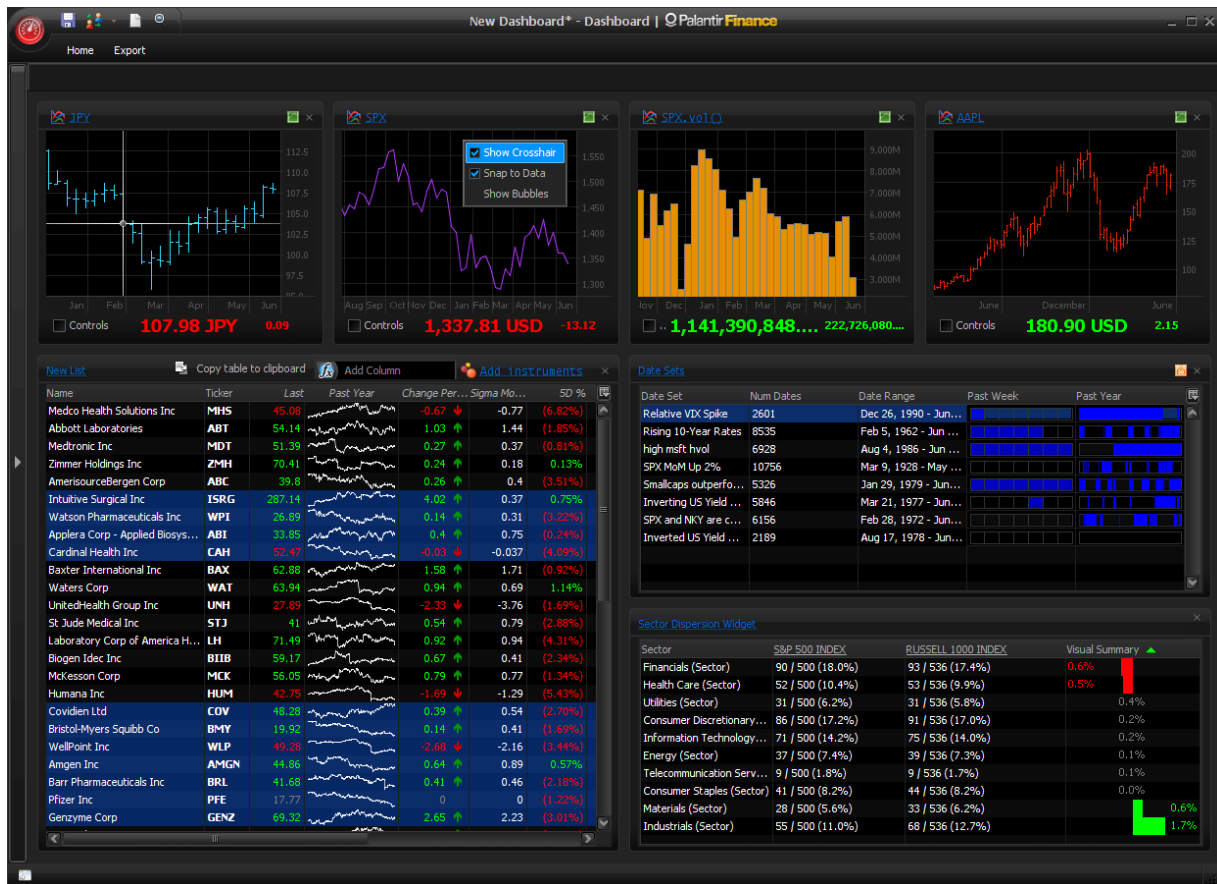


Figure 2: Financial Dashboard Screen Shot (source: blog.palantirtech.com [14])

5 Why's

The engineering team at Palantir initiated a Five Why's after reading Eric Ries's blog post on applying 5 Why's to software programming [15]. Since then, every department has adopted the process—even facility operations. An unexpected side benefit from implementing Five Why's is an increase in company cohesion. When field staff submits a bug report, they now receive a summary of the five why's analysis including details on how the bug was resolved. Similarly, when a customer states a need for certain feature, five why's analysis is done to identify the customer's root desire and the value proposition of the desired feature. Palantir lists all their processes on an internal wiki—here's a brief summary of their five whys process:

1. Schedule a meeting with all the involved parties.
 - a. Part of the goal of Five Whys is to get people understand the root of the issue and get them involved in creating the solution.
 - b. If you are doing a Five Whys report on a product failure that occurred in the field, be sure to involve at least one representative from both the development and the implementation teams; preferably QA as well.

2. Appoint a Five Whys master to lead the discussion – asking “Why?” five times to uncover the root or reason, writing down the results, and making sure action items are assigned.
3. During the discussion, try to keep the Five Whys in a public place, like a whiteboard - it helps prevent people from talking in circles.
4. Finish the Five Whys step before creating action items - it helps keep the discussion focused.
5. Make sure each action item is assigned to a specific person to do. The action item should be possible to verify that it is done within the week. "Lessons learned" should be incorporated into the best practices page for the particular task.
6. Within 24 hours, the Five Whys master should write up the report on the internal wiki using the Five Whys Template and mail out the full text plus a wiki link. Each report should be added to the Five Whys Archive.
7. Participants have one week to echo back to the Five Whys master that the action items have been completed. The Five Whys master should check back 7 days after the fact.
8. As a best practice, assign Jira (project management tracking software) issues where possible as a simple way to close out action items.

Small Batch Sizes

The theme of a constantly iterating, constantly learning organization is manifested throughout the series of interviews one of the authors had with Palantir employees. When Palantir first built demo's for customer validation, they operated on a two-week release cycle. As time passed and the product became more complex, cycle time stretched out longer and longer—from two weeks to a month, to three months...

Eventually, we had one cycle that took six months. That's when we decided it was getting out of hand. A short release cycle forces us to maintain high quality code and quickly illuminates when we're building features that customers don't care about. [Bob McGrew]

Since their customers' internal networks are entirely disconnected from the rest of the world, Palantir must ship a physical product to them. Currently, Palantir pushes a new release at the end of every month to their internal instances. Their stated goal is a three-week internal release cycle. They only push a new release to customers every two to four months. But still, in an industry dominated by ponderously slow-moving enterprise software companies, even four months seems incredibly fast.

CONCLUSIONS

To truly define values in the eyes of the customers, Palantir Technologies emphasizes the need to talk with customers throughout the development process, keep asking “why?” but never ask “why?” without implementing the answers, and be very disciplined at every step of the process. Their success in both the government and financial sectors began with the very first lean principle – identifying the values of customers. Of course, overly emphasizing the voice of customers could be detrimental – it may lead to vulnerability such as delayed product releases and the loss of priority in meeting the most important customers' needs [2]. Clear strategic directions at Palantir helps their employees stay focused while listening and responding to customers' needs.

REFERENCES

- [1] Womack, James; Jones, Daniel; and Roos, Daniel. (1990). *The Machine That Changed the World*. New York: Simon and Schuster.
- [2] Middleton, Peter and James Sutton (2005). *Lean Software Strategies*. New York, NY: Productivity Press.
- [3] Morgan, James M. and Jeffrey K. Liker (2006). *The Toyota Product Development System*. New York, NY: Productivity Press.
- [4] Kano, N., Seraku, N., Takahashi, F., & Tsuji, S. (1984). Attractive quality and must-be quality. *Journal of Japanese Society for Quality Control*, 14(2), 39-48.
- [5] Middleton, P. (2001). Lean software process. *Journal of Computer Information Systems*, 42(1), 21-25.
- [6] Poppendieck, M. & Poppendieck T. (2003). *Lean Software Development: An Agile Toolkit*. Addison-Wesley Professional.
- [7] Widman, J., Hua, S.Y., & Ross, S.C. (2010) Applying lean principles in software development process – a case study. *Issues in Information Systems*, XI(1), 635-639.
- [8] Peter Middleton, P. & Joyce D. (2012). Lean Software Management: BBC Worldwide Case Study. *IEEE Transactions on Engineering Management*, 59 (1), 20-32.
- [9] Fowler, M. (2002). Building only the features you need. *The XP 2002 Conference*, <http://www.martinfowler.com/articles/xp2002.html>.
- [10] Wang, X, Conboy, K., Cawley, O. (2012). Leagile software development: an experience report analysis of the application of lean approaches in agile software development. *The Journal of Systems and Software*, 85, 1287-1299.
- [11] Microsoft Windows 8 Online Blog <http://blogs.msdn.com/b/b8/>
- [12] Wood, N. (2004). Customer value: applying the first principle of lean. *Management Services*, 48(3), 14-17.
- [13] Palantir Technologies Profile on CrunchBase. <http://www.crunchbase.com/company/palantir-technologies>
- [14] Palantir Online Blog www.blog.palantirtech.com
- [15] Ries, Eric (2009). Online blog <http://www.startuplessonslearned.com/2009/07/how-to-conduct-five-whys-root-cause.html>.