

Westerly Consulting, LLC

Back to Basics Solve the customers' problems

The Problem

Too many products do not achieve the needed market success. The strategy is fine and the market is growing but the product is not being adopted as quickly as needed. Why?

A recent Agile Product Management discussion elicited the following summaries to "Why Products Puck":

- "...you have to validate the product with customers...not 'Is it what they want' but 'will it answer problems they have'."
- "Insufficient end-user research, poor cross-functional management ..., and a general lack of leadership decisiveness."
- "Those that remain successful never lose sight of their product's purpose, namely the business problem(s) they are focused on solving."
- "...designed without considering the user and its uses."

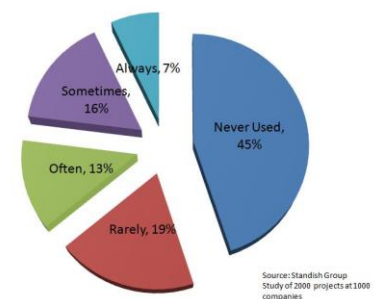
Products are poor because the focus moves *from solving a customer problem* to *debating feature lists* and never goes back. Feature lists take on a life of their own. Everyone "knows" what the features should be so the debate focuses on them, not on the problems. The actual Problem Statement is assumed, implicitly. Features need to solve problems.

45% of all software features are never used, never! Another 19% are seldom used. That means over 50% of product development investment is often wasted. The effort to verify and test the solutions with the customer and their problems is often short changed. People are just too busy to go back and verify. They are driven to get things out, not stop and make sure it is correct. The problem lies not in the building of the software but in the steps before the building process begins.

Are you reaching out to your customers properly? Do you understand their needs better than they do? Do you understand the problems they are willing to pay money to solve, in detail? Do you know the range of solutions they are willing to accept? Are you testing each feature against these criteria?

Getting this right is essential and a matter of good leadership, process and habits. Assuming people are doing it right is high risk. Creating a process that is accessible to the whole team is much less risk. Having a solid definition of the customer problem and a high level description of the solution usually

Usage of Features and Functions in Typical System



saves time and effort in getting to the right solution quickly. It allows the whole team to iterate and drive to a solid definition of the needed features. This is a solvable problem.

Going back to basics

- What is the problem being solved by this development cycle?
 - Can we articulate it?
 - Does the customer agree?
 - How often do we go back and verify?
 - How much is it worth to the customer?
- What are reasonable solutions to the problems we are solving?
 - Can we articulate it?
 - Does the customer agree?
 - How often do we go back and verify?
 - What is the revenue attached?
- Are the User Stories in line with the problem statement, the solution, and the business timing needed?
 - Who checks this?
 - How do we check this? How often?
 - Are we doing solid acceptance testing after each iteration?

Westerly Consulting can help. We have a methodology for writing a Problem Statement, a Solution Statement, and iteration prioritization. We help in the discipline of verifying User Stories, creating User Story Acceptance tests. We help change the process to accommodate verification, reaching out to customers, and getting team buy-in. We can get things back on track by teaching and coaching those responsible.

Our Methodology

There are three basic steps in our methodology. We start with a high level **Problem Statement** that outlines what the product needs to do. It is followed by a **Solution Statement** that describes the broad guidelines of what is essential to deliver to the customer. Lastly, it organizes the solution into **steps of multiple iterations** that can then be used to convert to User Stories. These steps are designed to work well with an Agile approach to building software.

The **Problem Statement** describes the problems that people are willing to spend money to have changed. It is the foundation of all that follows in building a product. It is not a detailed specification, but a high level reviewable document that is accessible to the customer and all the members of the team. Basic questions it answers are:

- Who is it that has this problem?

Fred Engel fred@westerlyconsulting.com

© 2013 Westerly Consulting LLC. www.westerlyconsulting.com.

- What is the problem?
- When does the customer encounter this problem?
- What are some scenarios of the customer encountering this problem?
- What is the relevant customer environment?
- Why is this important to the customer?
- How would solving this problem change the customers' behavior?
- Is the customer ready to buy a solution?
- Do we understand the economics of this problem?
- Is cost in-line with benefit?
- What homework have you done to be certain of the above?

The **Solution Statement** discusses the solutions to the problems in a broad and complete manner. It is the document from which User Stories are created and guides the detailed feature creation process that comes after it is complete. Again, it is intended to be consumed by the customer and all the members of the team. It covers:

- Who is the user and what is their context?
- What are the core mandatory capabilities of the solution?
- How will the user behave with these new capabilities?
- What are some scenarios of the customer using the product successfully?
- What are some scenarios of what the product should not do?
- What are the environmental and integration needs of the product?
- What are the "nice to have" capabilities?
- When do various capabilities need to be delivered?
- What is the pricing of the product? What are the cost constraints?
- How will the product be licensed?
- What standards does the product need to adhere to?
- What are the performance, and availability objectives?
- What is the risk mitigation plan?

Given the iteration methodology of today's software development, the next step is to define the **building steps for the iterations** involved in the release. Iterations allow for incremental deliverables that can be used to verify assumptions and provide customer feedback and buy-in. Planning the iterations can lead to a better achievement of the product goals, and better guidance to the ordering of the User Stories. This planning takes the following items into account:

- How do we get the riskier features far enough along to reduce risk? When?
- How much do we need to build in order to get customer feedback in time to make changes?

- Can we build the broad components in a way that allows the whole system to be functional early?
- Can we verify basic business assumptions better by changing the order in which we build capabilities?

Westerly Consulting coaches the key team members in the process, facilitates meetings, reviews the documents, and generally helps bring focus and discipline to the product definition process.

Westerly Consulting