

SUCCESSFUL NEW PRODUCT DEVELOPMENT: GOOD DEVELOPMENT TEAM MIXED WITH PROJECT MANAGEMENT, ADD SOME (NOT SO) COMMON SENSE AND STIR¹

Mike Venter, RF Ops LLC

ABSTRACT

We were rushing deliveries to an important client in time for his financial year-end. Approximately 25% of the deliveries were late and incurred penalties. It was an unexpected outcome for our world-class team that has been developing similar modules, products and systems for many years. With over 30% of profits lost in penalties, a detailed post-mortem process identified the problem with our previously successful recipe for new product development.

The size of the development group has more than doubled to facilitate the new development. The roadmap required multiple product ranges sharing core functional modules. We were using centralized on-line management. We found that project scope and complexity have increased to the point where working sixteen hours a day at the end of the project could not overcome the problems caused by the large multi-project development teams and the inter-project dependencies.

Good (not common for us at the time) sense dictated the following corrective actions:

1. Formally managing dependencies
2. First develop high-risk and client-critical items
3. Early customer demonstrations of functionality
4. Improved formal project communication.

After the first year, project on-time delivery improved to over 85% with penalties approximately 3% of profits. The next year on-time delivery increased to over 95% with insignificant penalties and remained at the same levels for subsequent years.

INTRODUCTION:

I am a PhD qualified Electrical Engineer. For most of my career I have been involved in the development of new products. After graduation I spent two years in the South African Military Corps of Signals ending up as a Project Officer managing the implementation of a Mobile Electronic Warfare System.

Thereafter I joined a small startup company as a design engineer involved in the development of telecommunication products. I designed and developed custom hardware and software for modules, products and systems. As I gained experience I

¹ This paper was originally presented at the 8th Annual UT Dallas Project Management Symposium in Richardson, Texas, USA in August 2014. It is republished here with permission of the author and symposium organizers. For more about the annual UT Dallas PM Symposium, [click here](#).

progressed through positions of greater responsibility until I became the CTO responsible for all development in the company. Later as CEO, I led the effort to optimize the company product portfolio to facilitate business growth and the expansion into new markets.

Over the years I have gained experience in all aspects of new product development, from mobile products and large systems to products selling in volumes of tens of thousands.

I have learnt that projects for the development of new products can be delivered on time, within specification and on budget. The only requirement is that a good development team, proper project management and some (supposed to be) common sense be combined for the implementation of the project.

The following case study illustrates the point.

BACKGROUND

The Company

The company is a small high tech product development house doing customized product development for large corporations. The company is involved from the original idea through design and implementation to manufacture and supply of the required products.

The company specialized in the design, development and supply of custom telecommunication and network modules, products and systems. Most of the products were unique and therefore had to be developed from scratch. The shipping volumes varied from less than ten for large systems to tens of thousands for some modules.

The development skill sets required covered the whole spectrum from real-time embedded modules through desktop and mobile products to high speed networking equipment and centralized management centers.

To grow the business and streamline the development effort, a roadmap for multiple product ranges sharing core functional modules and a centralized management system was devised. Two separate development teams were combined and expanded to provide resources for the implementation of the roadmap.

Development Team

The world-class development team consisted of engineers and computer scientists, most of who have been with the company for multiple years. Even though team members were regularly headhunted for their skills, we had a low personnel turnover (hopefully a result of the time and effort invested in the hiring process and the developer friendly company culture).

In each project the project management was performed by the technical manager who also contributed to the development effort. The projects were planned with input

and buy-in from the responsible team members with oversight from the executive management team.

WHAT HAPPENED?

We were rushing deliveries to an important client in time for his financial year-end. It started with the typical symptom of missing a delivery with the sincere excuse from the responsible manager and developer: We just need another week to finish the work. The week then became a month and even three to four months in some cases. Based on the lack of accuracy in their delivery estimates, it looked as if the responsible people had no idea what they were doing.

In the end approximately 25% of the deliveries were very late and incurred penalties. The penalties accumulated until the last deliverable was out of the door. At that stage more than 30% of the profits were lost to penalties and the client was, to put it mildly, not a joy to work with.

This was an unexpected outcome from our world-class development team that has been successfully developing similar modules, products and systems for many years.

HOW DID WE REACT?

Needless to say, with over 30% of profits lost in penalties the next steps were obvious: Search for the guilty, punish the innocent and reward the non-participants...

Fortunately, sanity prevailed and the mother of all post-mortems (at least for the company) was started to find out what went wrong. The process involved everybody: From the most junior to the most senior, technical, client-facing and even the client. With everybody involved, they became part of the solution. The results were an eye-opener.

WHAT DID WE FIND?

It was found that in the preceding twelve to eighteen months a number of important changes occurred. Most of it was by design, with the aim of growing the business and streamlining the development processes.

As a result of a strategic decision to grow the business by developing additional business in offshore markets, new product ranges were identified to enable the planned growth. The development roadmap for these new products required that core functional modules be shared between different products and that a single centralized on-line management system be developed for all product ranges.

To have the resources to implement the development roadmap, the two semi-independent development teams in the company were combined and new developers were hired to fill some of the skills gaps. All the projects were initiated, planned and executed by this "new" group.

The new development roadmap with:

- Multiple new product ranges
- Shared core modules
- Centralized on-line management system

Implemented by an expanded development group to form:

- Multi-project teams
- Different skill sets
- Multiple inter-project dependencies

Translated into

- Increased project complexity
- Very high implementation risk

With hindsight, common sense indicates that these indicators should have been recognized and taken into account in the planning.

WHAT CHANGES DID WE IMPLEMENT?

To accommodate the increased project complexity and risk, good practice dictated that we implement some (what is supposed to be) common sense actions w.r.t. to project management and the technical involvement from the development team.

We started the changes at the beginning with the breakdown of the project into tasks and deliverables. The following principles were used:

1. The order of execution was prioritized based on the implementation risk of the item and the importance of the item or functionality to the client. High risk items implementing functionality that was important for the client, were done first. This ensures that any risks that materialize can be addressed without delaying the project (i.e. the sooner you fall behind, the sooner you can start catching up).
2. Schedule early and frequent tangible deliverables or demonstrations that need to be witnessed by the client. This has two advantages:
 - a. It keeps the client in the loop and makes him part of the process. If required during the project, this will make it easier to negotiate for concessions.
 - b. It forces the developers to start as soon as possible and let them give their best effort, since they will be doing the demonstrations.

3. Ensure that the developers know how all items will be implemented and tested and that all test resources are provided for.

To pro-actively manage the risk introduced by the complexity of the product roadmap implementation, the inter-project dependencies were catered for as follows:

1. Fast tracked the roll-out of the centralized project management system and expansion of the project office to allow inter-project co-operation as part of the process.
2. The different tasks were defined such that each one could be implemented as far as possible independently from other tasks.
3. Managers and developers involved in the different dependent tasks formally managed the specific dependencies between themselves.
4. Inter-project communication was facilitated by regular formal meetings and linking the inter-dependent tasks on the project management system.

As with all initiatives involving change of this magnitude, there were some teething problems. However, with everyone part of the process and pursuing the same goals, these initial problems could be managed over time without derailing the primary goals of achieving and in some cases even bettering our previous performance levels.

WAS IT SUCCESSFUL?

In the end all the investment of effort paid off. One year into the process our on-time deliveries improved to 85%, but even better was the fact that only 3% of profits were lost to penalties. The year after that (two years into the process) the on-time deliveries were at more than 95% and the penalties became insignificant. From then on the performance stayed at the same levels.

An additional return on the investment was that the development roadmap was implemented within the required time and that the new product ranges proved to be the engine for business growth and offshore expansion.

WHAT DID WE LEARN?

According to Van Holthoon and Olson in their book "Common Sense: The Foundations for Social Science" (University Press of America, 1987, p.9) "common sense consists of knowledge, judgment, and taste which is more or less universal and which is held more or less without reflection or argument".

In this case what should have been common sense in the company was forgotten and not available when the environment changed. We had to remind ourselves of what should be common sense in implementing new products. Only then could we achieve the required levels of performance in the more complex environment with

multiple product ranges and complex projects consisting of multiple inter-dependent sub-projects.

In the process of getting accustomed to the new more complex development environment, we gained additional information that should be included in the "common sense" body of knowledge. This additional common sense is given below as a few "Rules of Thumb".

Rules of Thumb:

1. Rule of thirds:

- a. Address risky items in the first third of the project. If any of the risks materialize (and they normally do) there is enough time to address it without delaying the project delivery. Options are adding resources, interacting with the client to modify requirements and / or change functionality.
- b. Addressing risks in the middle third of the project requires more resources and approval for changes to the requirements and functionality is more difficult to get and more expensive to implement.
- c. When risks materialize in the last third of the project, all you can do is brace yourself for the fallout (i.e. late project, unhappy client, unhappy management, etc.). At this late stage additional resources are going to be very expensive and will only delay the project even more. It is late in the project to tell the client his project is going to be late and too late for any changes in requirements or functionality that will make a difference.

2. Address human nature in the execution of the project:

Developers often have a high level of confidence in their own capabilities. They want to give the client the best (in their opinion) solution possible and are worried that they will not get the opportunity to implement these cool new features they thought of during the design of the project (i.e. galloping excellence):

- a. Walk on water syndrome: Sometimes experienced specialist developers think that they can do everything better than everybody else, forgetting that they are better than everybody else in their field of expertise because of their specialist knowledge and years of experience in their field (forgetting that they create the impression of walking on water because they know where all the rocks just below the surface are). Give them so many challenges in their field of expertise that they do not have time to show their "capabilities" in other of their non-specialty fields.

- b. Start with difficult tasks first: Starting with easy tasks create the impression that the project is easy and that more than enough time is available. The developers then tend to waste time on irrelevant features, thereby using most of all the contingency time available before starting with the risky tasks. Such projects can only be delivered late.
 - c. Deliver only the functionality and performance required by the client: This way the required functionality and performance can be implemented and stabilized before any new functionality is added and optimizations made. When 80% is good enough, do not promise to deliver 100%. If the 80% can be implemented in X time, the extra 20% optimization will also take X time. Delivering the 80% will allow you to implement the other 20% for the client while getting paid for it.
 - d. "Always wanted to" is not a good design driver: Always re-use existing and well-known modules as far as possible. Any new technology brings a new set of risks. If it makes sense to use new technology, introduce it in the company on your own time to allow for the learning curve and to address any latent risks.
3. Confidence in the proposed solution: Spend sufficient time in analyzing the proposed solution and work breakdown until there is a high level of confidence in the accuracy of the solution. Before the implementation of a solution is completed, a number of important decisions must be made. By checking the validity of those decisions as a "thought exercise" before implementation is started, major risks can be mitigated. If one or more of those decisions must be changed due to some project or performance related reason, it is by far easier and cheaper to do in our minds and design documents than it is to change an existing implementation.

On a lighter note:

We all know the engineer Dilbert, the star of the comic strip by Scott Adams. There is also an animated television series. Those of you that have watched the series will remember the opening sequence where Dilbert evolves from a cell to a fish that grows legs to venture on land to get up on two legs to evolve into Dilbert the engineer.

I think that if Dilbert evolves further he would evolve into a Project Manager. It is the logical next step.

About the Author



Mike Venter

RF Ops, LLC
Dallas, TX, USA



Mike Venter is the CEO of RF Ops LLC, a Dallas based electronic engineering design house specializing in the design and development of turnkey modules, products and systems where RF wireless operation is an important part of the functionality. He was previously CEO and CTO of Nanoteq (Pty) Ltd, a South African company specializing in the design, development and supply of custom cryptographic security modules, products and systems.

Over the last ten plus years, Mike Venter became a specialist in building and managing strong technical development teams to facilitate the efficient design, development and supply of world-class custom embedded modules, products and systems from original idea to manufacture and operational deployment.

During his career he gained experience in the following roles:

- Project officer in the Signal Corps of the South African National Defense Force, managing the development of a mobile Electronic Warfare system
- Investigating the viability of video conferencing systems for Telkom, the telecommunication service provider in South Africa
- Created and lectured a post graduate course in Image Processing at the Department of Electronic Engineering at the University of Pretoria
- While working for Siemens in Munich, Germany developed software for Digital Telephone Exchanges
- In various development positions were involved in the development of a data modem for HF radio channels and various cryptographic security products and systems for both commercial and government clients

Mike has a BSEE, MSEE (majoring in Telecommunication and Cryptography) and a PhD in Image Processing from the University of Pretoria in South Africa. He can be contacted at mike.venter@rfops.com.