Speedbumps and Potholes on the Road from Projects to Products

MIKE GRIFFITHS, Leading Answers Inc.

Transitioning from projects to products made perfect sense for my client. Much of their business was digital and their websites / online-services would not be "completing" or going away soon. Development was deliberately continuous, and executives embraced this as both inevitable and desirable. However, just because it was the logical thing to do, did not mean it was easy.

1. INTRODUCTION

Digital products and services are designed to evolve over their lifespans. Unlike, say, a bridge that is constructed and then used, we expect significant ongoing development of digital products. This development goes beyond simple maintenance as products continue to grow. This is the new world of product vs project development.

Projects are the default way of working for most organizations. Projects get approved, then assigned budgets, teams, and timelines. Hopefully projects deliver what was asked for on time, then transition to support and are closed.

Products however are not temporary endeavors, we expect (and hope) for them to continue indefinitely since that demonstrates they are still evolving and delivering value. Switching from projects to products changes the way we approach planning, funding, staffing, and many other elements of work. This experience report recounts the journey of one organization as it transitioned from projects to products. It is not a success story.

2. BACKGROUND

The client undertaking the transformation is a telecommunications company who offers mobile phone, internet, and cable TV services. The bulk of their new services are digital, but they also have considerable investments in physical infrastructure (hardware boxes, cabling, networks) to integrate with, manage and maintain.

My involvement with the client dates back on-and-off almost 20 years. I have been invited in to consult and train various groups and departments since 2000. My most recent engagement was to assist one department troubleshoot their agile scaling challenges and it was this work that led to the transition from projects to products.

I worked with them for a couple of days a week providing strategic direction, consulting and some coaching services. While this was all the time that I could commit, it would have been better to have had more face time with the client. I also helped recruit an agile coach to work full time with the client, working with the teams directly.

Mike Griffiths, Leading Answers, Inc.; email: mikegriffiths100@gmail.com

3. THE ISSUES AND CHALLENGES

3.1 The Initial Diagnosis and Plan

The department I was called in to work with had a very challenging mandate. With over 100 in-flight projects, many competing to integrate with the same websites and legacy systems there appeared to be far too much work in progress.

In addition, the staffing structures in place created many handoffs between groups. Opportunities were identified and defined by business groups then assigned to in-house development teams that typically worked with external vendor teams experienced in the domain or technology involved.

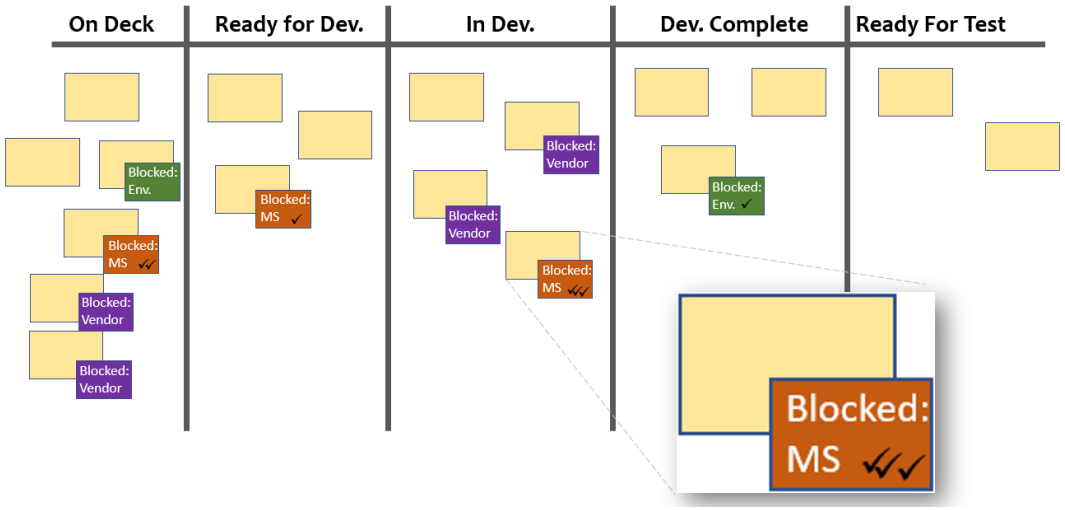
Most applications and services being developed needed to integrate with existing systems that were managed by a legacy systems management group. Infrastructure (provisioning technical environments) was supported by a separate department located in the US. Finally support and sustainment of applications once developed were handled by a separate support group.

This collection of teams and groups created many dependencies, handoffs and delays as teams tried to get their work done. When we mapped the flow and transfer of work between groups there seemed many opportunities for optimization. Reorganizing around products with long-running product teams responsible for creation, development and sustainment of a single service should eliminate many of the handoffs.

This is what we set off to do. Restructure to eliminate handovers and bring people from supporting groups into the product teams to reduce the dependencies between groups. It seemed the logical thing to do, but unearthed deeper, harder to solve problems.

3.2 Early Stages and Discovery of Further Issues

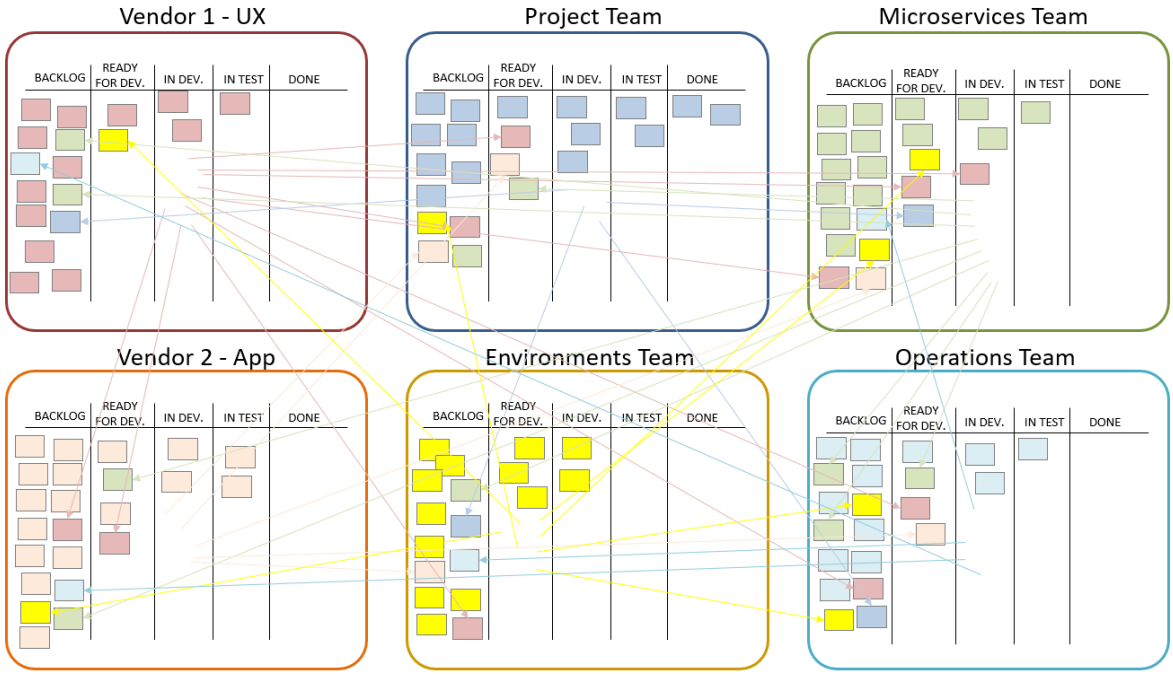
First, we needed to illustrate the extent and impact of the inter project team dependencies. This allowed us to build support for the remediation activities. Using techniques described in the book, *Making Work Visible: Exposing Time Theft to Optimize Work & Flow*, by Dominica DeGrandis, we showed the impacts of the delays. We made the dependencies, blocked items and queues visible by flagging items on the project Kanban boards. A simplified model is shown below:



Team members added blocked sticky notes to items waiting for other groups. The sticky note showed the initials of which group the task was waiting for. In the example above "MS" stood for the MicroServices group. Each day at standup team members added a tick to the sticky if the item was still blocked. This way we quickly generated an easy to interpret score of the most impacted work items and the source of delays.

Scrum masters and program managers worked to remove the blocked items and follow up with the groups creating the delays. Working with these groups we discovered they too had many dependencies and delays that were preventing them from doing their work (or our requests) as quickly as they would like to.

Creating dependency maps for the impacted team illustrated what we suspected. The web of project teams, vendors, and departments meant everyone was waiting on everyone else in an ensnarled mess. A simplified dependency map for one small project is show below.



The lines on the dependency map indicate a task on one team task board that is dependent and waiting on a task in another team. Analysis of how people spent their day revealed that many people spent as much time, if not more time, following up on work and trying to get things expedited than undertaking development.

Following ideas shared by Troy Magennis in his 2015 Agile Alliance conference presentation "Entangled: Solving the Hairy Problem of Team Dependencies" we analyzed and investigated the impacts of having so many dependent teams. Potential solutions included forming larger teams if it meant fewer handoffs and dependencies. While larger teams may be slightly less efficient due to increased communication channels, if they result in fewer dependencies this penalty might be well worth the effort.

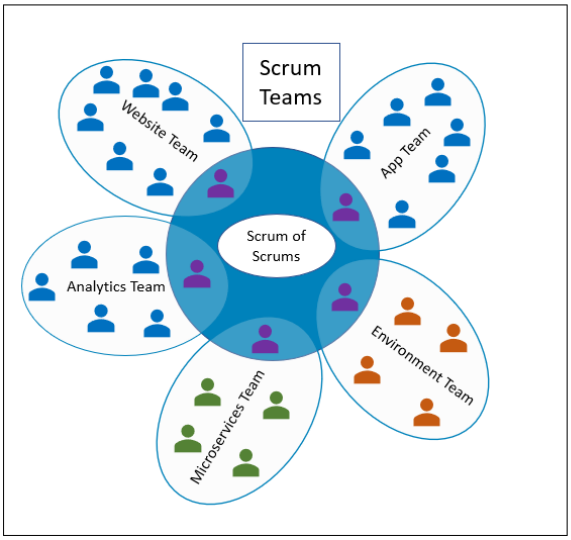
It is an example of global vs local optimization. Yes, small teams can operate the most efficiently internally, but if they are spending a large proportion of their time waiting for or escalating work within other teams then they are no longer efficient.

Before making the switch to product-based development, the client had also recently adopted a microservices architecture and Amazon Web Services (AWS) cloud-based development. The two largest product development streams were using microservices and AWS for the first time. The AWS environments were provisioned and managed by a separate team located at an office in the US. A dedicated team was formed to create and manage microservices across all projects and products. The creation of these groups made sense from a grouping of specialized knowledge perspective. However, it created more silos and dependencies for the teams. Now whenever a team wanted an AWS environment created or modified, they needed to ask the AWS Environments group. Worse still, all microservice changes needed to be performed by the dedicated microservices team. Quickly, both teams became bottle necks to the product teams.

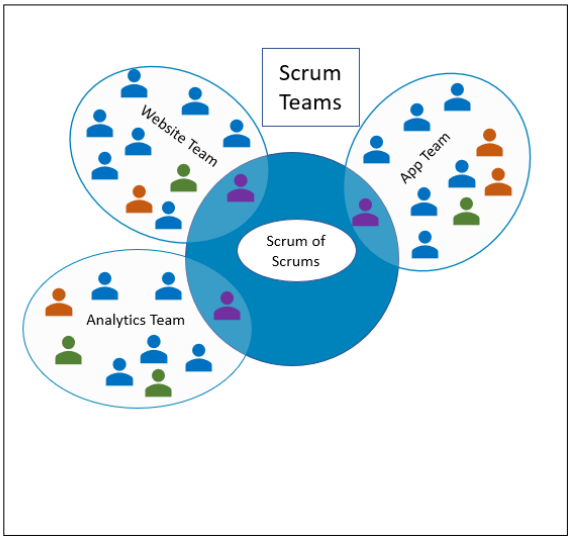
Product teams would need environment and microservice work done to complete their stories. These tasks would have to be passed to the relevant group. However, these groups were busy satisfying their own work items and requests from other teams. Often it would come down to who created the most fuss or found the most pressing justification for an escalation that got served next.

By breaking up the environment and microservices groups and embedding their members within individual teams we created larger teams but were able to reduce dependencies and waits. These larger but less dependent teams did help to reduce hand-offs and wait times. Their throughput increased and WIP reduced.

Before Team Consolidation: Many small Teams



After Team Consolidation: Fewer, slightly larger Teams



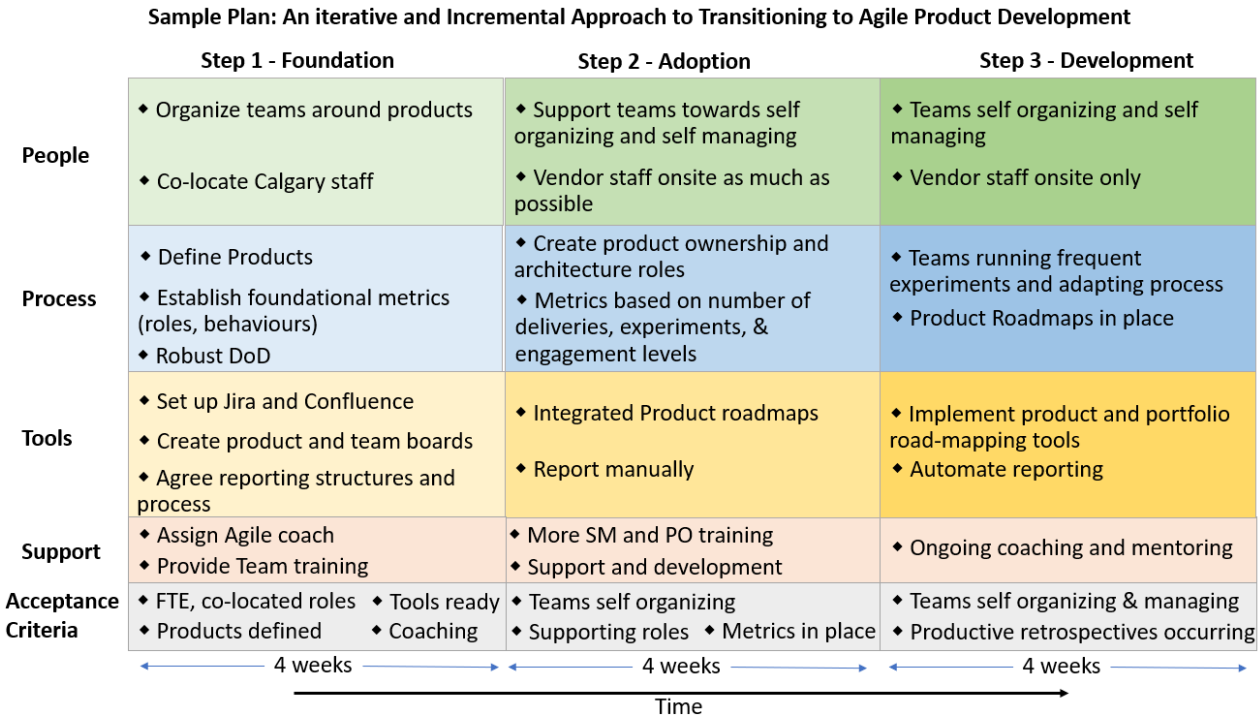
Along a similar vein we asked that the Product Owners who had previously worked from a head office downtown to come out to the industrial park to work embedded within the development teams. Previously they frequently reviewed and commented on the emerging product remotely, but always shared more information and provided deeper insights when onsite for weekly meetings. Of course, the downtown office location was more convenient for most people and had better lunch options so the move was not universally welcomed or embraced. Luckily the product director saw the benefits and encouraged the Product Owners to comply.

Forming dedicated, co-located, long-lived teams was the main organization strategy for reducing handoffs. Embedding the support staff within the development team removed a handover to another group at go-live time. So long lasting development and sustainment teams were formed around the main websites and major applications. These websites and applications became the new product streams we organized around. These product teams would not be created just for delivery and then disbanded, instead they would remain intact and undertake all subsequent updates and integrations.

The final team structure change would be vendor staffing models. The client used a variety of outsourced vendors to undertake development work. This was either because the vendor possessed expertise in a software package or technology or because there were no people available internally to work on the initiative. Unfortunately, analysis of project throughput and delays showed that back-end integrations to legacy systems were often a bottleneck.

So, while initial development for vendor teams might go well, getting the system integrated and operational was very difficult and competed for the same people and physical resources as in-house projects. The result was that, despite vendor promises, outsourcing a project did not result in it being completed when desired. Also, all these additional projects and integrations further hampered the in-house project delivery plans since the in-house projects were also competing for these integration services.

A simplified schematic of the plan to transition from projects to products is shown below:



The general plan was to create a three-step transition, each step taking two team sprints or four weeks in total. The primary goals of the three steps are outlined below:

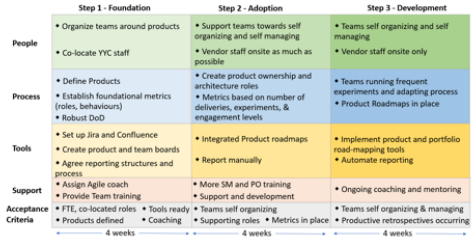
- **Step 1 – Foundation** – Get everyone to a common starting point. Form the dedicated product teams and co-locate them in the same workspace. Define the products. Make sure each team has a definition of done and some basic metrics based on checking if the necessary roles are in place and people are functioning as a single team. Make sure teams have the tools and training they need. Also, provide support in the form of coaching and mentoring.
- **Step 2 – Adoption** – Try to ensure the desired behaviors are being instilled. During the adoption step we wanted to promote self-organizing teams. Most teams were operating this way but some vendor teams were used to having tasks distributed to them. We also wanted to make sure the vendors were transitioning staff to be onsite. We wanted to start measuring throughput and engagement metrics. In addition, develop integrated product roadmaps that show dependencies to other product teams at integration points.
- **Step 3 – Development** – Encourage ongoing product-based development. By the third four-week productization step we wanted to make sure vendor staff were onsite full time. Also, that the Product Owners were now embedded with the teams permanently and the teams were familiar with throughput based reporting, plus running more experiments as the result of their retrospectives. We hoped to see mature product roadmaps, burn rate based budgeting and ongoing coaching and mentoring in place.

The idea was to take a single product team through this process and review how it went before rolling it out to the other product teams. The executive wanted to accelerate the move to product teams and so we were asked to overlap the transition for subsequent product teams.

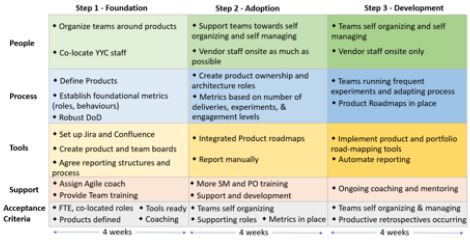
The overall plan, for all product teams was a nesting of the 3-step process just described and is depicted below.

Sample: Incremental Rollout – One Product First, then a couple more, then the remainder

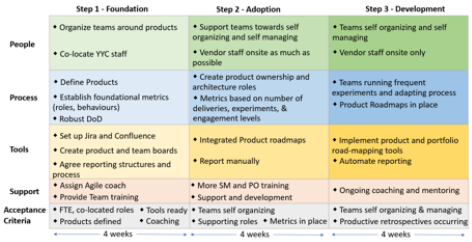
Product 1



Products 2 & 3



Products 4-end



Following this approach, we would start with forming and executing one product team, then transition two more, and finally all the remaining product teams. It would have been nice to start products 2 and 3 after product 1 had completed all three steps so that lessons learned could be applied. However, the entangled, interdependent teams were struggling to deliver any systems to production and so the decision was made to overlap the product transitions by one step.

4. IMPLEMENTATION PROBLEMS

The initiative was not without its challenges, here are some that were encountered and eventually stalled the transition.

4.1 Staffing Changes

Around the time of planning the transition from projects to products the organization announced a voluntary redundancy program. The idea was to reduce overall operating expenses and so staff were offered the opportunity to take a financial package based on the number of years of service and voluntarily leave/retire.

The goal was to reduce headcount by a couple of hundred people. Ideally senior managers who had been with the company for many years, had high salaries, and had been managing large groups of service representatives whose roles were getting replaced with online self-service offerings.

However, more than two thousand people elected to take the voluntary redundancy option. This far exceeded initial estimates and also depleted the development staff who would be needed to create the new digital products that facilitated the anticipated cost and staff savings.

The organization honored all the voluntary redundancy requests and so backfilling critical roles and knowledge transfer activities diverted significant time from the in-flight projects. Deadlines were only adjusted when they slipped and it created more pressure for teams that were now attempting to achieve their original project goals, replace departing team members, and transition to a product centric way of working, all at the same time.

This also caused a distracting preoccupation of who-is-staying, who-is-going conversations. Large-scale change and job departures threaten people's sense of security, corporate identity and personal happiness. In

some groups 50% of the people took voluntary redundancy creating difficulties in maintaining momentum of improvement initiatives, as survival and core functions became the new priority.

4.2 Funding Challenges

Previously projects were funded based on their business case. If a sufficient return on investment could be presented to senior management then funds were allocated for the project. Project and program managers then tracked burn rates and reported on overall spend along with requests for additional funds if necessary. It was an inexact process that lacked good follow through on delivered benefits, but everyone understood it.

Switching to a continuous product development stream presented some challenges. Predicting the expenditure was quite easy, just total the anticipated run rates for the stable teams, add vendor costs, software licenses and some contingency for additional items. The problem was estimating the anticipated financial benefits for the investment. Without the normal return on investment analysis for the backlog of planned features there was less justification to authorize the expenditure.

The issue was partially solved by requesting quarterly tranches of budget to fund the product teams for three months at a time. Each quarterly budget request was also accompanied by a list of planned features and enhancements provided by the product owner. The product owners also included business justification and return on investment predictions. It was essentially a process similar to getting large projects approved. However, because the periods were shorter and so the dollar amounts smaller, it attracted less scrutiny.

These more frequent reviews with decisions to invest, keep-stable or scale-back product teams are in line with recommendations from product development books like Allan Kelly's *Continuous Digital* and Evan Leybourn and Shane Hastie's *#noprojects: A Culture of Continuous Value*.

5. COACHING AND CONSULTING LIMITATIONS

Before accepting the assignment I explained I was only available a day or two per week. The client agreed to this arrangement and we also hired a full-time coach to work with the teams. The initial coach recommended was too expensive for the client and they selected a cheaper, less experienced candidate. While they understood Scrum and agile principles well they did not have experience of product development or director level consulting.

In hindsight I should have flagged this as a risk. In such a dynamic environment many decisions and changes were often made between one of my visits on, say, a Wednesday and a subsequent visit the following Monday. While I posted my schedule online and at my desk, people were too busy to check and typically unaware of when I would be in the office so I missed many meetings and opportunities to provide guidance. I was available via phone and email, answering many questions that way, but when the voluntary redundancy program was announced the rates of change increased. If doing this again I would likely recommend 75% - 100% availability to better handle the volume of questions, requests for review, etc.

6. SPONSOR CHANGE

As we were rolling out the first product teams the director who had brought me in to help with the transition resigned. The interim replacement director for the department was not familiar with agile and so naturally had lots of questions about our work practices. They wanted to know why we organized into small teams, co-located with the business, and did development in-house when there were plenty of consulting companies that specialized in software development.

I worked with other people in the department to provide answers, overviews of our development approach and education presentations. We hosted an open house, took executive to visit teams in their workplace and did some public sprint demos, show-and-tells and business walkthroughs.

There was more staff turnover as several people who had worked at the client championing the agile rollout for many years felt the agile rationale was being questioned. I continued to help provide additional information about why we were using agile but then one day when trying to log into email found my account had been suspended. A telephone call confirmed my suspicion that my contract, like others, had been terminated immediately. So I wrapped up my current work and sent it in via my personal email.

I had worked with people from the client on and off for 20 years so I had a good relationship with many of them. I heard from a couple of people afterwards that two products had reversed the decision to host vendor staff on-site. Instead they moved the majority of the work out to be done remotely by vendors and a shift back to project structures was adopted.

It was a very disappointing outcome since a lot of people worked hard to overcome many technical, organizational and process challenges to align with a product based approach to development. It felt like much of the difficult problems had been solved, but, as they say, culture eats strategy from breakfast. The client has a culture of making sweeping changes. I still believe orienting around product delivery would suit the organization well. They are a technology company; websites and apps are their core competency and market differentiator. Adopting a product-based model would support their digital-first strategy very well.

Policy and staffing changes are just part of real world business. I hope I get an opportunity to work there again and help them in their journey. After leaving the client Mik Kersten published his book, *Project to Product: How to Survive and Thrive in the Age of Digital Disruption with the Flow Framework*. I believe it offers some great metrics and strategies for organizations embarking on a similar exercise. I recommend people undertaking similar initiatives read it along with the other books listed in the references.

7. ACKNOWLEDGEMENTS

I would like to thank all the people I worked with on this attempt to transition to product-based development. Given it was ultimately dropped, postponed, repurposed or however you want to describe it, I chose not to name the client or provide the full names of everyone involved. However, they know who they are.

As often occurs, the people working in the trenches had all or nearly all the answers before I set foot in the company. A big part of my role was to be the credible-outsider who proposes what internal staff have been suggesting for years and is then heralded as insightful.

Everyone I dealt with was open with their suggestions, transparent with their challenge areas, and willing to try new approaches. This was amazing, you are a credit to your professionalism and it was a privilege for me to work with you. Thank you so much – next time we will nail it.

REFERENCES

Dominica DeGrandis, *Making Work Visible: Exposing Time Theft to Optimize Work & Flow*; IT Revolution Press; 1 edition, Nov. 14 2017.

Troy Magennis, talk "Entangled: Solving the Hairy Problem of Team Dependencies" Agile 2015 Conference; Video: For Agile Alliance members https://bit.ly/2ELC66G , PDF Slides https://bit.ly/2HOXIB5

Allan Kelly, *Continuous Digital: An agile alternative to projects for digital business*; Software Strategy Ltd.; Oct. 2018.

Evan Leybourn, Shane Hastie, *#noprojects: A Culture of Continuous Value,* lulu.com; Jul 18 2018.

Mik Kersten, *Project to Product: How to Survive and Thrive in the Age of Digital Disruption with the Flow Framework*; IT Revolution Press; Nov 20 2018.