

# **Managing a cross-company software integration**

**A case study**

**By Noah Lloyd-Edelman**

## **Introduction**

My relationship with E2F began when I was asked to manage a mysterious company partnership. That was back in October 2013 and I had no idea what to expect. There was no agenda, no paperwork or notes, just a few hours blocked off for a chat. There wasn't even an official name for it yet. For now, it was simply TTS.

For a good portion of our first meeting, E2F talked about what they did, how they did it, their company structure, who they worked with, and what projects they were working on. For the final twenty minutes we focused on what E2F needed and wanted to include in the integration. The full picture wasn't clear to me just yet, but it would be soon enough.

## **What is Transifex?**

Transifex was created to help internationalize digital content by enabling any company to present content in different languages and locales. It does this by creating a way to easily manage, collaborate, and of course translate digital content with one platform.

As a developer, you can quickly hook your product into Transifex, upload your content, collaboratively translate it, then export the finished files as quickly as you can translate them. There's no need to depend on a translation agency because experienced third-party translation vendors are built into the platform. With a slew of translation utilities and supported file formats, most content can be directly imported and exported from the user's platform. Transifex can even collect and localize websites with a single JavaScript snippet.

Transifex was formed in 2007, and has offices in both Menlo Park, California and Athens, Greece.

## **Transifex's translation ecosystem**

As the popularity of Transifex grew, a series of small translation communities formed within the product over the years. Unfortunately, they weren't a viable business solution, so third-party translation companies were integrated into the platform. These companies offered inexpensive, crowd-sourced translations by a community which was paid by the word. This created additional revenue and a marketplace for translators, but, this arrangement wasn't enough.

Reoccurring complaints focused on the quality and consistency of the translations. As a result, E2F was brought on board to help provide our customers with a higher-quality (but more expensive) translations. We hoped this would breathe the much requested professional quality to the platform, while still providing other more cost-effective and faster alternatives.

## **Understanding the problem**

The closed beta of TTS was supposed to launch in a few months, but Transifex was developing in so many different directions, the integration kept being pushed back. I was recruited, with two other people, to get the project on track and to meet its launch deadline.

After my discussion with E2F, there were some disagreements between our companies, and a lot of last minute engineering work that needed to be implemented. With a looming deadline and

limited resources, our goal was to create a minimum viable product and take whatever actions were necessary to have it ready by the deadline.

Our plan was to work quickly and efficiently by:

1. Learn how E2F handles orders
2. Refining and agreeing on the integration experience for the translators, managers, and customers
3. Create a viable product with the time, budgetary, and engineering constraints we faced
4. Frequently running sample orders to uncover bugs and discrepancies (while keeping scalability in mind)
5. Creating intuitive on-boarding and training material for E2F employees and translators
6. Creating a cross-company support network for quickly communicating, issue tracking, and business insight

### **Agreeing on a common product**

To make certain both companies were on the same page, we used behavior flow charts (then working prototypes) to test and demo any refinements we made to the system. The ordering process was supposed to be as automated as possible. This meant delivering enough information to E2F about the expected translation, while not overloading the customer with questions or requiring person-to-person interaction with them.

The solution manifested as forms which clarified the tone, industry, and personality of the desired translations. This baseline allowed us to finalize our engineering work for the beta testing and allowed us for focus on our ultimate goal of a seamless ordering process and a workflow which was was scalable and adaptable.

Since I joined the integration half-way through the process, much of my time was spent playing catch-up. With TTS being partly integrated into Transifex, many engineering and interface decisions had already been made and coded in. During the entire process I had to keep reminding myself to see if the system worked, not redesign it.

### **The teams and their roles**

I struggled with the telling of this story because it isn't just about Transifex. When we say "we", it consisted of both teams at E2F and Transifex. I have tried my best to show what we at Transifex were heavily involved in and what we influenced, but in each and every case I need to acknowledge that the E2F staff were right beside us.

The management and design was mostly between me, a Transifex manager, and two product managers at E2F. During the information gathering phase, we consulted with both companies' CEOs and CTOs, as they had the final word on the design and final product. Most of the initial discussions were done in-person. Then, we switched to Skype for conference calls and Google Drive for online document sharing.

### **Deciphering the TTS black box**

When I began this project, I saw it as a black box. Our goal was to design a seamless ordering and delivery system between Transifex, E2F, and the customer, but there was no supporting

material. The order and file management system was mostly in place, but this told me very little — I needed more information.

I interviewed the CEO and CTO of Transifex to find out what had been discussed and envisioned for the final product. There was no paper trail I could follow, so I had to construct a mental model as I went along. Aside from handling bugs and support tickets, Transifex wanted the entire ordering and delivery process to work without human interaction. To make certain we were on the same page, we outlined what had been engineered to that point and what was yet to come.

That same day, a product manager and I drove to E2F to get their input. I wanted to learn how E2F worked internally, how they were planned to handle Transifex orders, and what they needed from us. After our visit, we consolidated our notes and began the process of brainstorming a workflow for the order process. Our goal was to create a prototype which both companies agreed upon. Using a Visio-like program, we created a flowchart of the order process and eventually built it into a wireframe. After several revisions, both companies agreed on the system and it was sent to the engineers.

When the engineers completed their work, we scheduled a closed-beta test. For our test subjects, we chose clients who placed medium-to-large orders with a mix of common and rare languages. The content of the orders was real, so we checked them daily to make certain things were going smoothly. A few bugs were found (which were quickly patched) and as E2F started receiving more and more orders, they began to refine their handling processes.

At the same time, I began designing a curriculum to teach product managers and translators how to use Transifex. This was mainly a distillation of the documentation and common questions I had seen while working in the support department.

I divided the content into two types (based on the sample personas). The first was manager material which focused on the platform, team management, communication features, utilities, Translation Memory, and so on. The translator content was focused on interacting with the application and the practical functionality of the platform. In both cases, we tried to do a top-down set of concepts with bottom-up set of explanations and examples.

During this time we also created a ticket-based support system for them, which allowed us to seamlessly move issues between companies and track issues for partnership performance analysis. In case a major issue arose, a point-of-contact chart was made, which based on the issue, provided the name and contact information for the appropriate Transifex employee.

For weeks after the closed-beta test, we continued to closely monitor the orders and even re-engineered parts of the system that were causing difficulties for customers or E2F staff.

## **Solved Problems**

### *Unifying product vision*

After gathering the breadcrumbs and goals of both companies, we were able to create a single product experience which had the personality, content, and experience both companies agreed upon.

### *Giving the orders personality*

By adding additional pages and fields to the order form, we were able to collect the information E2F needed to personalize an order to the customer's tastes, while requiring no human intervention. This allowed Transifex to run the integration with very little manpower while generating extra revenue.

### *Issue handling and support*

By creating cross-company support and a point-of-contact systems, problems were quickly answered by the appropriate staff member. Although this was a small addition, it made communication significantly more efficient.

### *Handling language capacities*

To make sure we didn't overload E2F's translator pool, we created an internal algorithm which used E2F's translator pool sizes (for each language), a calendar with international holidays, and the amount of content that could be translated per hour. This allowed us to monitor outgoing orders, provide requests (in advance) for increasing translators for a language, and to give accurate estimates for delivery dates.

### *On-boarding material*

The content was delivered to translators and managers in a PDF packet. After testing the material on the product managers and a several translators, I felt I no longer had to teach the staff how to use the product. — Teach a man to fish, right?