

Software Modernization For an Ultimate Market Leader Among Personalization Engines

How switching from deprecated AngularJS to up-to-date React and crafting a custom UI kit helped a global SaaS customer experience (CX) company **facilitate maintenance of their personalization engine** and **speed up design-related tasks**.

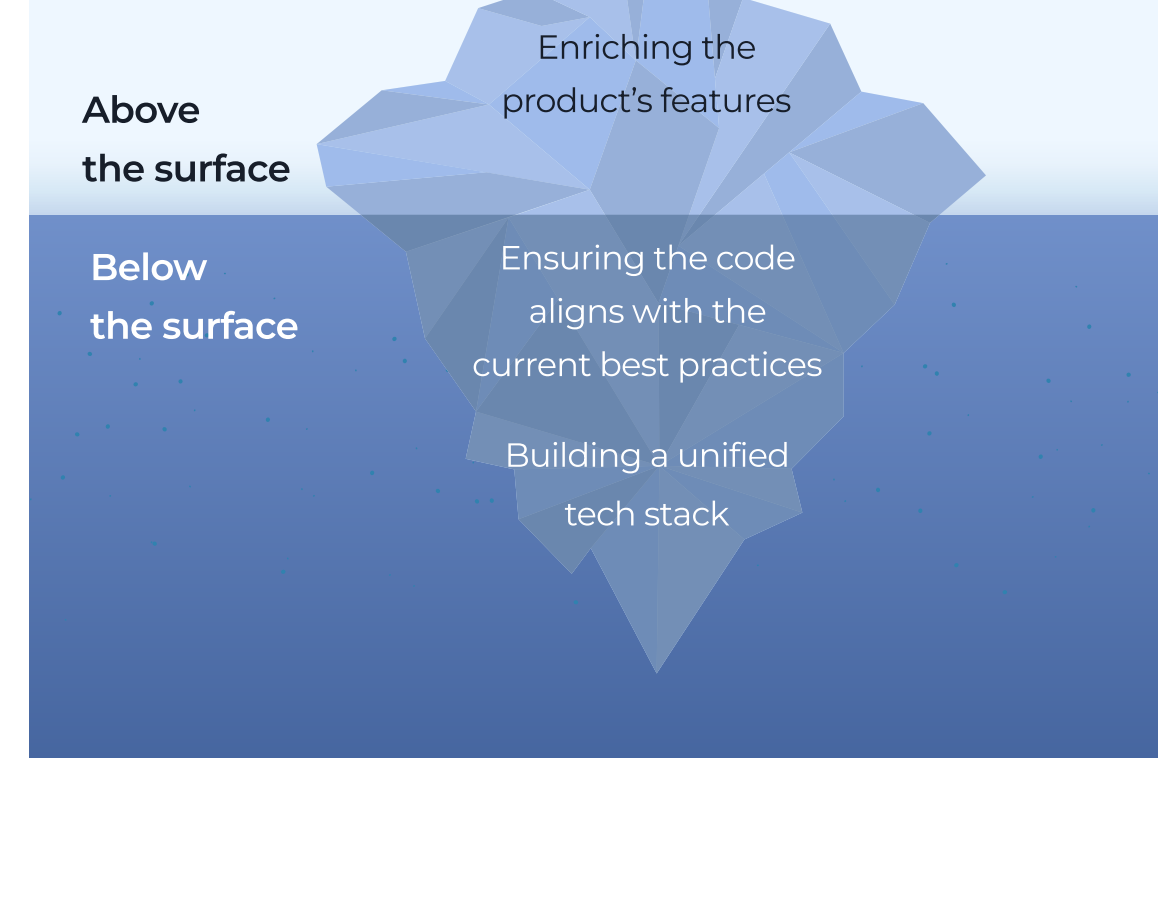
Industry:
Technology

LEGACY SOFTWARE MODERNIZATION

Business challenge

Understanding that nothing lasts forever helps ambitious SaaS companies truly excel. Organizations pushing their field's boundaries can't peacefully rest on laurels. They have to put a premium on constant product evolution, spanning three areas:

Focusing on the above-the-surface part, or implementing new features, to win and retain customers, companies tend to neglect the technical foundation that can spell significant trouble without essential upkeep.



The aging heterogeneous tech stack was the only vulnerable spot of our client – an **unbeatable market leader among SaaS personalization engines**. They have already blazed a trail in customer experience (CX) solutions with their cutting-edge personalization engine, which empowers B2B and B2C companies to reach their potential customers across any channel and device with highly targeted offerings. However, their **core products' front end in AngularJS**:

- Didn't match the client's current tech stack in React
- Muddled software maintenance as support from Google ceased in 2018

To eliminate these hindrances, the client decided to **migrate legacy AngularJS code to cutting-edge React**.

Additional benefits of switching from AngularJS to React included:

- **Flexibility in organizing the software code** as React is a UI-focused library, while AngularJS is a model-view-controller (MVC) framework with a set of predefined design patterns.
- **Accelerated performance** thanks to React's reliance on the virtual document object model (DOM), which allows changes without updating the entire page.

The client has already started code migration from AngularJS to React but soon hit a roadblock: their **in-house team has neither time nor capacity to juggle both feature development and legacy modernization**. Therefore, the client decided to hand over the migration part to a tech partner.

However, finding an ally they could trust wasn't a walk in the park, as the client previously had a negative experience with an outsourcing company, making them cautious this time around. Determined to avoid the same pitfalls, they came up with **strict technical and soft skills-related requirements for the future partner** so that only the best would make the cut:

Battle-proven expertise

in both AngularJS and React

Accurate planning

and sticking to the outlined timeline

Proactiveness

in decision-making to strengthen their core team

Top-level soft skills

for swift communication and synergy with their in-house team

According to the client, "instinctools turned out to be the perfect fit. How did the migration go and what were the extra efforts we put in?"

Solution

The client had already outlined an overall migration strategy and prepared the infrastructure, so our team picked up the **code migration while perfecting existing functionality and fixing bugs**.

01 Replacing a rigid legacy tech stack

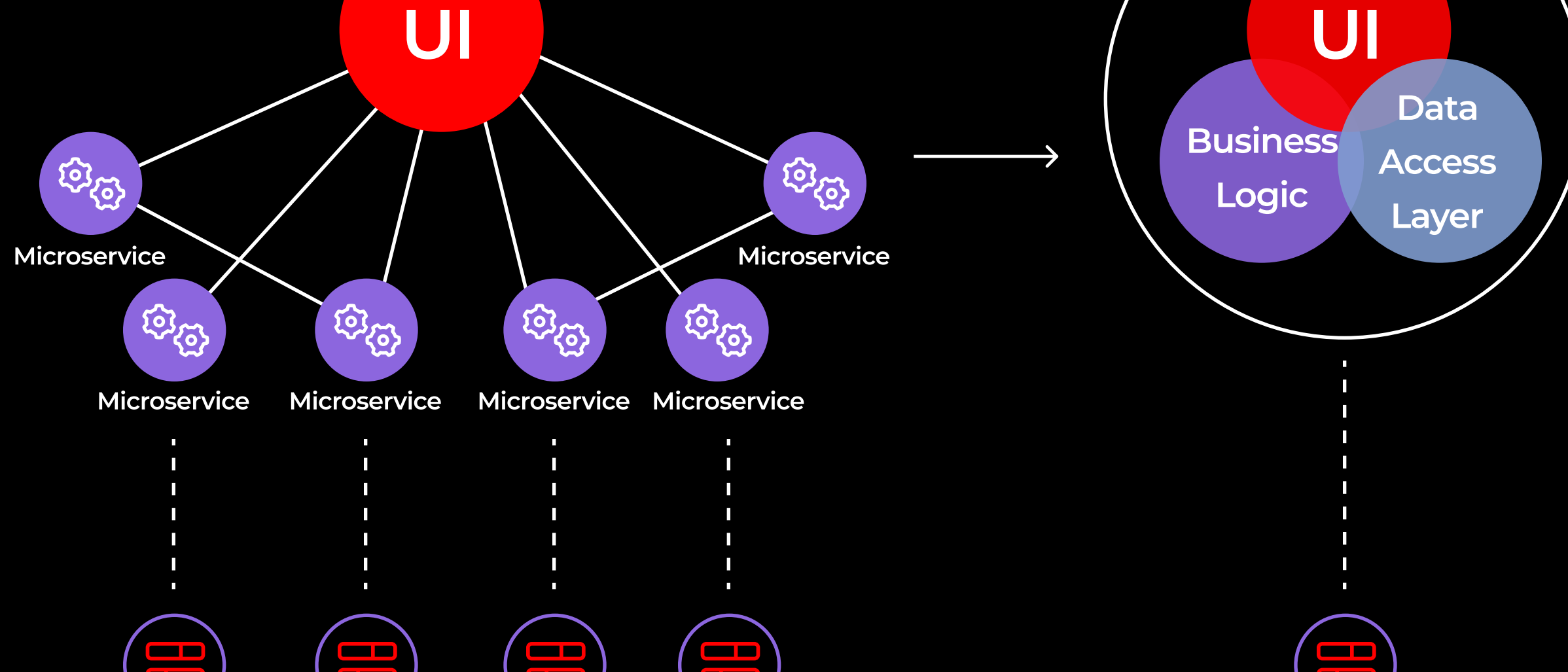
We were responsible for rewriting several solutions within the client's software ecosystem. Some of them raised notable challenges, such as their **legacy customer journey mapping platform**.

The client's system was a sore spot because of its:

- **Fragmentation**. The platform consisted of six sub-systems, each written by different developers at different times.
- **Documentation void**. Without system documentation or knowledge keepers left in the company, rewriting the solution became a daunting task.

We had to replace this ticking time bomb with a **single user-friendly React solution**. To achieve this goal, our team:

- Analyzed the platform's code to figure out the best way to rewrite it
- Prepared updated software architecture from scratch
- Selected Nx library for collecting code from different microservices and a Vite bundler to set up a development environment for React
- Replaced six disconnected microservices with a monolith
- Migrated libraries from AngularJS to React



02 On to new challenges

Six months after the start of our cooperation, the client trusted our team enough to delegate the tasks beyond the migration. That's how we started **working through their overall backlog and participating in R&D activities**.

Taking on the primary stage of their research efforts

The client wanted our team to **join their ongoing in-house research dedicated to creating a custom UI kit**:

- Explore and examine current best practices in crafting compilations of UI components
- Draw up suggestions for their custom UI kit

Impressed with the proactiveness of our team, the client asked us to join the UI kit development.

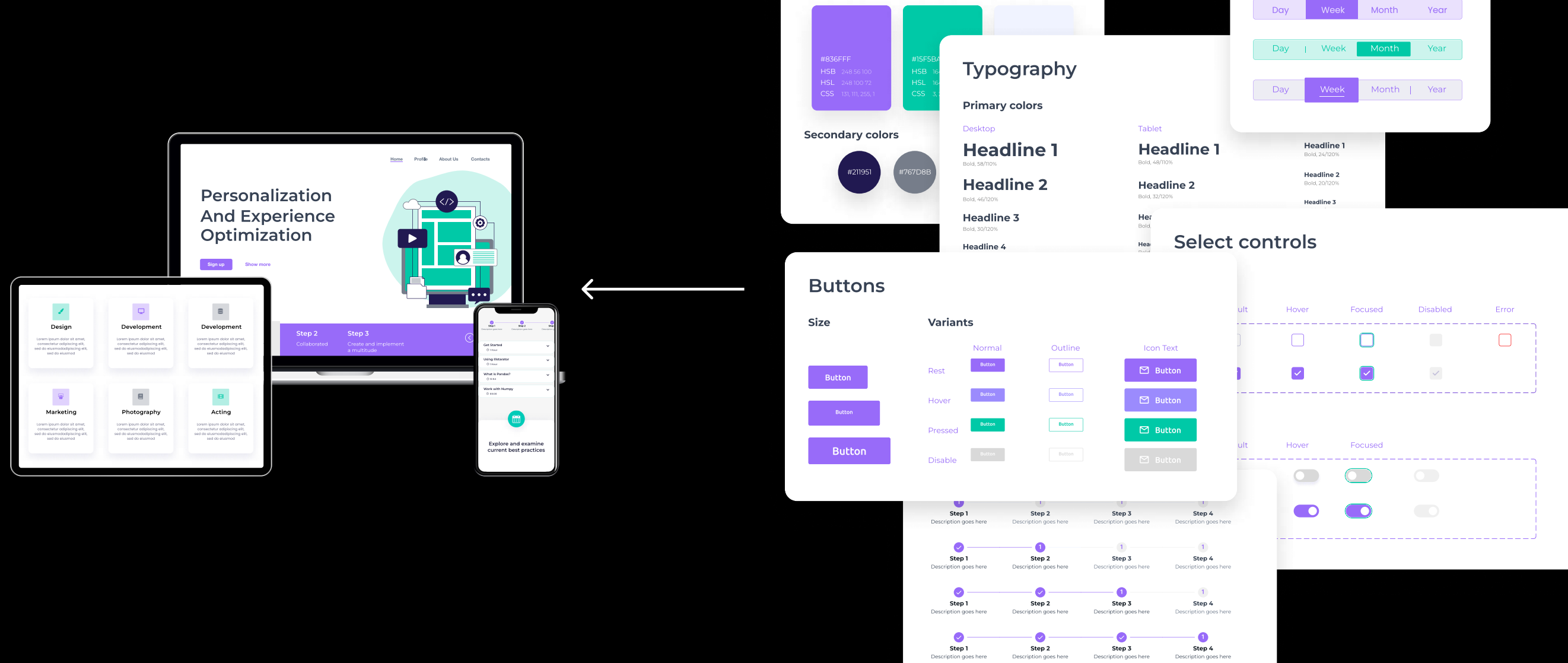
Assisting in crafting a custom UI kit

Just as with the AngularJS to React migration, the client had already invested efforts in building UI components in line with the company's brand book. And much like switching to the more up-to-date tech stack, they also needed **reliable, high-qualified partners to carry out the UI kit development**.

Before our team knuckled down, the client got tangled up in their existing visual components, such as buttons, checkboxes, widgets, etc., scattered across various sources.

Here's what we did to help the company stop wasting their time and effort:

- **Created a single library for all UI components** to enable simple reuse across all the client's current and future products
- **Collaborated with their in-house design team** to create and implement a multitude of UI elements, enhancing consistency and efficiency



Before

- **Heterogeneous** tech stack of their SaaS CX solution
- **Challenging** product maintenance because of six siloed microservices
- **Absence** of a robust UI kit
- **Massive** backlog
- **Lack** of experts to run technical research

After

- **Unified** tech stack for the client's SaaS personalization engine
- **Simplified** product maintenance thanks to monolith architecture
- **Solid** custom UI kit
- **Reduced** backlog
- **Support** of dedicated experts in R&D activities

In our client's words

Frontend Guild Manager on the client's side outlines their impressions of cooperation with *instinctools:

"*Instinctools' team hits the ground running to seamlessly integrate into our internal development team and deliver high-quality work. They exceed our expectations in both the quality of work, as well as hitting delivery milestones. We were also impressed with the team's effort to be involved more, and to reach beyond the areas they worked on.*"

Do you have a **similar project idea?**