Rebuilding For Scale

East Coast Product's Collaboration with Vsnap

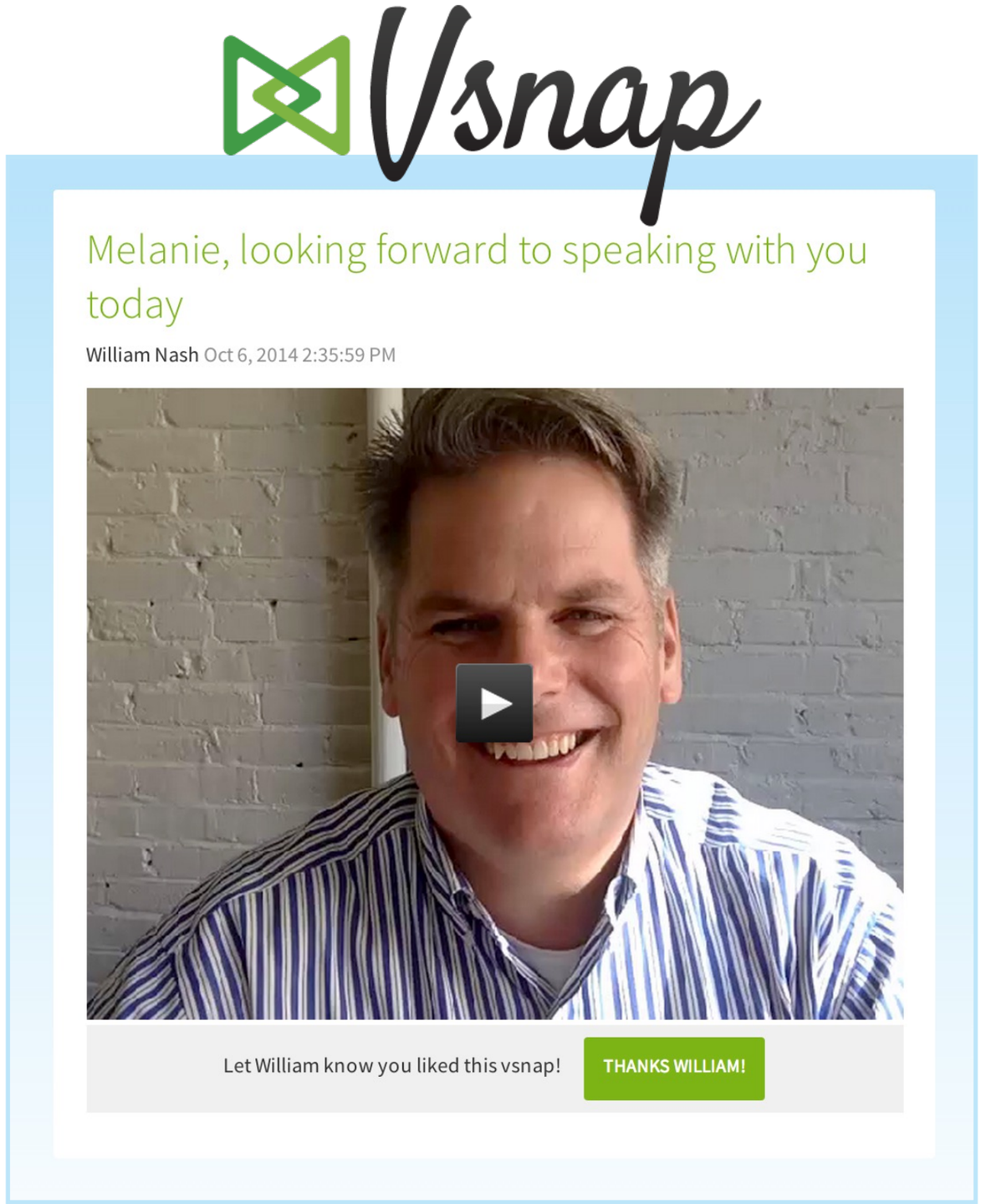
EXECUTIVE SUMMARY

Vsnap is a SaaS video messaging platform for inside sales. The app humanizes sales interactions to build sustained business relationships.

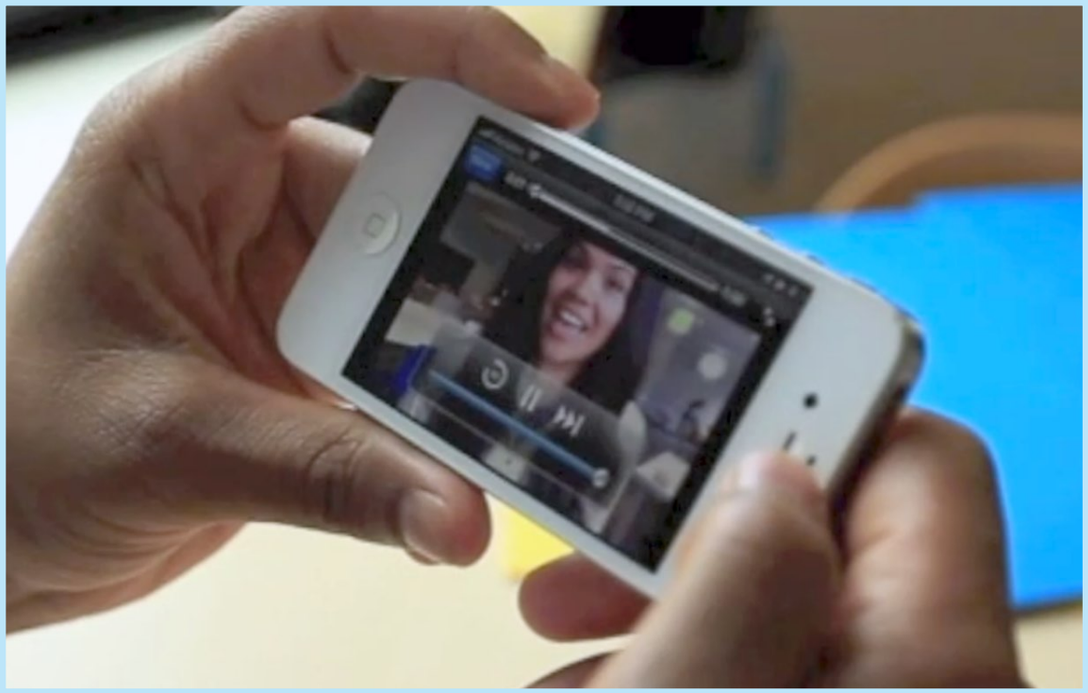
As Vsnap grew and the user-base was more defined, it became clearer to CEO, Dave McLaughlin, that the prototype application in its current state was not scalable. In December 2013, he brought in Chris Swenor as CTO and the East Coast Product development team to rethink the app.

After three months, the application was completely rebuilt in JavaScript and successfully re-launched on schedule.

The rebuild resulted in the successful acquisition of Vsnap in April 2015.



ABOUT VSNAP



Vsnap is a video messaging platform for inside sales. The app's mission is to humanize sales interactions to build sustained business relationships. The easy-to-use app allows a sales rep (the sender) to record a 60-second video message and add attachments or a note.

Next, the sender indicates the recipient(s) who receive a notification to view the video message via email.

The recipient does not have to download anything or sign up for the app to view the message. Once received, the recipient has the option to "Thank" the sender or reply via email.

As Vsnap grew and the user-base became more defined, it became clearer to CEO, Dave McLaughlin, that the prototype application in its current state was not scalable. In December 2013, he brought in Chris Swenor as CTO and the East Coast Product development team to rethink the app.

"I spent two and a half years trying to get a clear path on tech – all of which was wrong. Chris got his head around the product quickly and gave me a roadmap to where we wanted to be."

- Dave McLaughlin

THE REQUIREMENTS

The original Vsnap application was intended to be a prototype only, and it was primarily developed by a team of outsourced developers responding to the needs of the sales and customer management teams. Because Vsnap was a prototype, the application became outdated and monolithic very quickly. Code updates required careful planning because it was unclear how changes would reverberate in other parts of the application. Updates took several hours and caused significant downtime that inconvenienced customers. Leaving things as-is with the Vsnap prototype was not sustainable for the long-term.

Chris worked with the development team to evaluate the best course of action: repair and update the existing application or rebuild the application from scratch.

Key Challenges

Rebuilding an application that is currently in use presents its own set of unique challenges. As the new development team planned for the build, they anticipated the following hurdles:

- Maintaining two separate applications for the three months allotted to the project
- Accounting for emergency patches and pressing feature requests from the customer-facing teams that might disrupt development
- Choosing an appropriate coding framework
- Hiring the right developers to build
- Quickly onboarding a new team that had not worked together previously

OUR PROCESS

Phase 1: Discovery

McLaughlin had a hunch that something was amiss with the fragile application so Swenor and the newly assembled development team decided to rethink the product altogether. The team needed to determine whether the existing application could be salvaged or if a fresh start was the way to go. They had three months to produce the next version of the Vsnap application.

The team spent the first month of Discovery familiarizing themselves with the existing product and understanding how it worked. While brainstorming improvements, they focused a lot of their research around one key area: how to make the application scalable and cloud-deployable for new users.

Development objectives:

- Work in a future-proof coding language
- Make it easy to add new features to meet customer needs
- Make it easy to deploy changes to production
- Create a scalable application

After careful research and deliberation, the development team unilaterally decided that the best approach was to rewrite the entire application. The Vsnap product in its current state didn't have code that was worth recycling and was written in a language that was losing popularity -- Grails.

Phase 2: Design & Architecture

Swenor and his team didn't want to disrupt service to current Vsnap customers during the rebuild. Their goal was to make the current application as usable as possible for existing customers without a large investment of developer time, freeing the team up to focus on the build. To achieve their goal, Swenor and his team worked with Vsnap's Customer Success team to define what bugs and features needed fixing to get the current application in its best functioning form. The team decided to take an API-first, modularized approach to development.

> ***What is API-first development?***
>
> API-first development includes the business logic in the API. This allows developers to easily create mobile and web apps connected to the same logic. The API can eventually be made public.

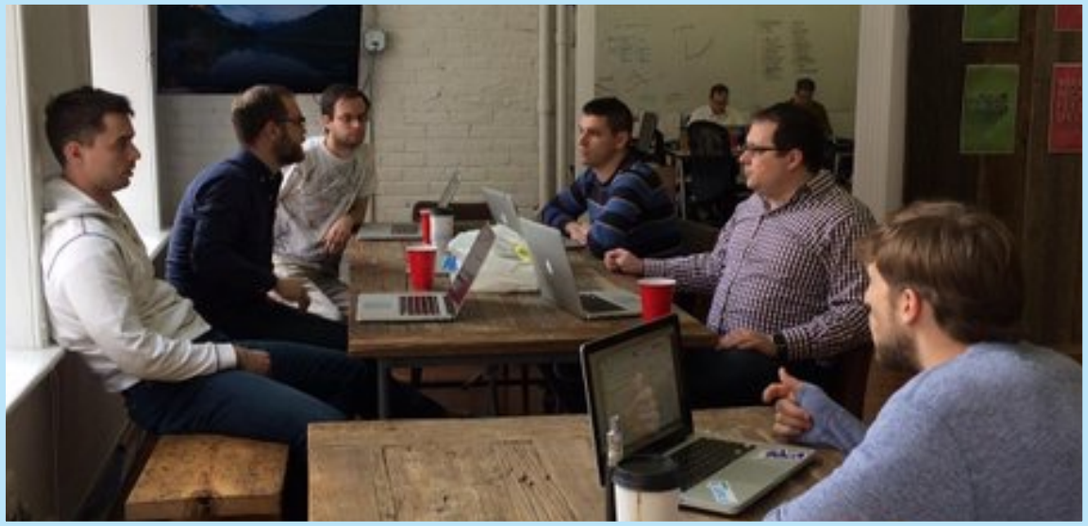
The team evaluated various languages and decided that the most performant choice was JavaScript. Selecting a popular, future-proof language, meant that it would be easy to hire developers down the line and ensure an abundant community of resources and support for further product development. The frontend was built in Angular.js and the backend was built in Node.js.

The biggest challenge going into the project was that no one on the development team had previous experience with a working on a Javascript-only application. Working in Javascript for the new version of Vsnap meant the team would spend additional time researching practices for the rebuild.

Phase 3: Development

As development began, the original plan was to turn the existing Vsnap application code into an API that could be consumed by the front end. However, this plan soon proved unviable because the app was so large and monolithic. There were performance concerns across the board. With the application in its current state, it could only handle four people recording video at the same time. Even adding a Node Abstraction API in between the old app and the front end wouldn't help because there would still be performance issues. It was clear that the entire backend would have to be rebuilt from scratch.

The Vsnap application consisted of multiple parts, including video recording and streaming using a Wowza server, which added to the overall complexity of the project. This meant the team had to leverage distributed computing and micro service architecture to split the application up into chunks and not have to rely on a large, singular app. In the future, individual parts could be scaled as needed.



The team ensured that they were only pushing out high quality code that followed best practices through peer review, thus preventing technical defects from being deployed. The process cut down on bugs and poorly architected code. Even the most senior developer on the team is required to have their code reviewed because this is a learning opportunity (at the very least) for more junior developers.

THE RESULTS

The development team invested three months into the Vsnap application rebuild and successfully re-launched the product on schedule.

For Swenor and the rest of the team at East Coast Product, this project converted them into JavaScript evangelists. Why? If you ask Swenor, it comes down to three main reasons:

1. JavaScript is the only language that exists in browsers. It isn't going away, and has the eyes of the best and brightest developers in the world.
2. Utilizing a single language allows the team to focus on becoming true full-stack engineers and experts in JavaScript itself.
3. JavaScript requires high attention to detail and has enormous potential.

Perhaps more importantly, the three-month rebuild also cemented the importance of technical autonomy and a learning company culture. Swenor says, “A huge contributor to the success of the rebuild was that Dave gave me 100% trust and control of the redevelopment. This made for better, faster development. Also, Vsnap’s existing learning culture made it safe for the developers to stretch past their comfort zones to create something in a language that was totally new to them.”

The relaunch of the application eventually resulted in the successful acquisition of Vsnap in April 2015.

“After the rebuild, our technology was acquired by an awesome company which folded it right into their product. The way Chris's team did the work is what allowed for that.”

- Dave McLaughlin