

Industry: Full-service HVAC&R equipment distributor

Size:

- 500+ employees with 16 locations in North America
- Represents 100+ HVAC suppliers in the US and Canada

Location: Toronto, Ontario, Canada

Solution Goal: Replace a customer ERP system that's more than 20 years old.

Solution Selected: Cincom Smalltalk



"Cincom Smalltalk just works. Period."

– Bob Nemec
Solution Architect

HTS Delivers a Custom-Built, Web-Based ERP Application with Cincom Smalltalk™

Up and running in less time than it would have taken to implement many commercial solutions



HTS is one of the largest independent, built-to-order, commercial and industrial, full-service HVAC&R manufacturer's representatives in North America. With 16 locations in Canada and the United States, the company has grown significantly since its 1992 founding.

Architects, engineers, building owners, general contractors, mechanical contractors and facilities maintenance teams—as well as the leading manufacturers they represent—routinely turn to HTS for solutions to issues ranging from energy-efficiency and environmental issues to better building economics and comfort for building occupants. Their industry experience spans industrial, institutional, commercial and residential projects of all sizes throughout the US and Canada.

Solution architect, Bob Nemec, had been using Cincom Smalltalk for over 20 years and knew it worked. So when the team and Bob began working on an ERP system for HTS in 2012 ... well, let's have Bob tell the story.

According to Bob:

The project the team has been working on for the past couple of years just went live. So with the benefit of hindsight, here's what we have learned.

Have a Project Champion

The HTS project champion/sme was one of the founders of the company. Without him, risks would not have been taken and the project would not have happened. We replaced a custom system that was more than 20 years old and also championed by the same person. He and the company believe that a custom ERP provides a competitive advantage. The old software proved it, which made selling the idea of a new system easier.

Smalltalk Productivity Rocks

The total effort was about 16 to 18 person years (our team size varied from 3 to 5 over 3.5 years). Compare that with the effort to deploy something like SAP, and we look good. Our team's productivity will really shine as new features and customizations are rolled out over the next couple of years.

Expect a Long Trail of Trivial Things

The beginning of the project, which consisted of figuring out how to use thousands of Cincom® VisualWorks® window specs in a Seaside application, including modal dialogs and dynamically morphing views, was fun. Other tasks included finding ways to hold complex updates prior to a save/cancel decision, building a new report framework that allows for edits and generates PDF content, adding application permissions, implementing a RESTful web-to-GS mechanism and more. All good stuff, but mostly done.

What really stands out is how much time we spent (and continue to spend) on the little things. It tends to be boring, almost clerical work, but it's what users notice—font sizes, colors, navigation sequences, default values, business rule adjustment, etc.

Pay Your Technical Debt Early

Looking back, we would have been better off not trying to preserve the ecosystem of the old, fat client framework (the idea was to keep most of the domain code as is). Instead, we should have started from scratch, using the old system as a spec.

The old framework was poorly architected. We knew it, but thought the technical debt could be managed. It was, but at a cost. We now know where we spent our time. It's evident that switching to new code earlier would have allowed us to be deployed earlier. If you see garbage code, be ruthless and get rid of it. Bad code is like a bed bug; it will keep biting until properly exterminated.

Show Progress

Users need to see progress, and developers need feedback. We hit the jackpot with our beta users; they were willing to put up with a lot of early unfinished code. It gave them a view of what was to come, and they communicated that to the rest of the company.

Have Clear Metrics

Make long-term metrics just as visible as short-term ones. Break them down, and make them part of each iteration, even if they are obscure and of no interest to the end-users. It will be frustrating. You'll be asked, "Why are you working on that and not the feature I'm waiting for?" But they'll be far more aggravated if the application is not reliable. It's like backups—you don't notice their absence until you need them. Be sure they see the value of the boring internal tech stuff.

Use Agile Development

We release a new production version every two weeks, with minor changes published twice each week. Developers merge their code every couple of hours. All new code is expected to have an SUnit test. The full test suite is run each night with Jenkins, and keeping tests green is the first developer priority. We pair up for tricky problems. Refactoring is considered to be a "technical investment." All changes are tracked (we have a nifty issue management tool). Reviewing the process is part of the process, and we adjust things almost every week. It's not easy, but getting to a smooth productive rhythm is so worth it.

Overall

It is nice to announce the deployment of a 100% Smalltalk application, built with VisualWorks, GemStone and Seaside. Our users are mostly happy. They want more features, and they want them sooner than later—not a bad place to be.

