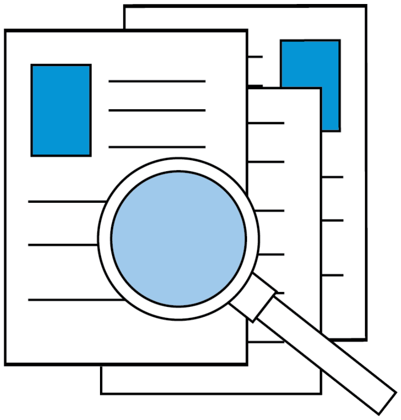
A Case Study on an Enterprise Software & Computer Hardware Agile Transformation



CPRIME RESOURCE CENTER

www.cprime.com | learn@cprime.com | (877) 753-2760

Company Details

- **Industry:** Computer Technology; Enterprise Software & Computer Hardware
- **Company Size:** 122,830 employees
- **Interviewee:** Sr. Architect for Customer-Support Portal

At a Glance:

- cPrime sends Agile Practice Lead across the country and then the world to train **6 teams from one of the largest computer software companies**.
- Company **increases release time by 780%,** improves transparency & visibility and increased customer feedback loop after engagement with cPrime.
- They give praise cPrime for being easy to work with, having **coaches with high expertise in Agile** and creating an overall great experience.

Overview:

In an effort to understand our customer's experience with cPrime, I've set out to conduct several interviews with key clients to understand the reasons for why they chose to go Agile, why they selected cPrime as a vendor and what they have noticed as the biggest benefit from performance improvement since the engagements.

The interview below is with a Senior Architect, from one of the world's largest computer technology firms, which develops Enterprise Software and Computer Hardware. The Senior Architect works for the Customer Support Portal (CSP) and leads all user experience for desktop and mobile support solutions.

They knew very little about Agile except for what they had read and needed to have their entire distributed team trained. The Sr. Architect spoke with friends at other large software companies who spoke highly of cPrime so they reached out for help.

Challenges:

One of the main reasons the CSP team wanted to go Agile was because **they simply wanted to be able to deliver things to their customers quicker**. They also wanted to be able to get feedback from their customers faster to improve their product.

"My role as an architect is mostly focused on user experience design. One of the things that drives me nuts would be that we would come up with a solution to a problem and then it would take us 18 months to deliver it. That's not acceptable." …

…"We wanted to get stuff to customers quickly and try it out and get feedback and then improve the product."

The Plan:

The other challenge was the fact that the **team members were located all across the world** – United States, India, Romania, etc. They were concerned that if they simply told each team to go out and get local Agile training that they would lose consistency in the messaging.

cPrime assessed their situation, challenges, environment and together, they came up with a 3 day course (2 days of Agile team training and 1 day of Product Owner training). cPrime sent their Agile practice lead, Kevin Thompson, around the world to teach the same custom course to all of their CSP team members.

- **2 Day Agile for Teams:** Agile Basics, Basic Requirements, Sprint Planning, Scrum Ceremonies, Advanced Requirements.
- **1 Day Product Owner Training:** Product Vision, User stories, Product Backlog, Story Mapping, Scaling Agile Planning, and also included simulations.

Outcome:

Increased their Release Time by 780%

"In the first 18 months, we delivered 8 releases which is 7 more releases than any other product our company delivers. That's 2 in a quarter month release cycle."

Improved Transparency & Visibility

"Transparency is one of those foundational things that is transformative."… "From a visibility perspective, our upper management can also have visibility into our work. If they wanted to come in

and say, "Show me where you are today for something that's delivered in 8 weeks." I will have something to show them as opposed to I'm still writing the speck."

Increased Feedback Loop from Customers

"It's also being able to see stuff come out the other side and be able to get quick feedback on it whether it's from customers; in my case, I work with customers and demo working code to somebody before I deliver it to production so I can continue to refine it. We do this all the time. We did it last week. We did it the week before that. I have things coming out the other side and I'm not just dealing with prototypes. I'm dealing with actual working code that I can test."

Evaluation of cPrime:

Coaches with Agile Expertise

"The reason why we went back to Kevin and cPrime repeatedly in multiple engagements was because we felt like we got exactly what we wanted. He really understands his stuff. He delivered a consistent message. He's a nice guy. He is very sublime. It worked fine for us."

Easy to work with & logistics taken care of without flaw

"It was very easy. It really didn't require a lot of effort. All the materials were delivered, the rooms booked, facility, all that stuff."

Overall a great experience

"We had a great overall experience and I've recommended cPrime in multiple occasions already, to other teams her as well as other organizations in the Bay area."

The Interview:

Sr. Architect: *This is the first team that was trained at our headquarters (Points to a photo on the wall). Kevin Thompson trained that team in Redwood City in December 2010.*

cPrime: *Was that one done in India.*

Sr. Architect: *No, that was here. This is the first team that cPrime trained on Agile.*

cPrime: *Very cool.*

Sr. Architect: *It says December 2010.*

cPrime: *2010, yeah. A while ago and you've come a long way. Let's get some background. Initially, what made you reach out to cPrime? What was your initial contact? What struggles were you guys going through on your team?*

Sr. Architect*: There's a saying that Donald Rumsfeld has said about known knowns, known unknowns and unknown unknowns. Unknown unknowns are things you don't know that you don't know. You can look it up. Just look up Donald Rumsfeld, unknown unknowns. It's a great quote. You can quote that I quoted it.*
It's a really good one. When we starting off with Agile, I didn't know anything about Agile. We kind of knew about it and we started to read about it and research it. I'm like, "I don't think we can switch the way we think just by reading." I reached out to some of my friends at other organizations that I knew were more advanced software development companies like Cisco, for example. I reached out some people at Cisco and I asked around. It's like, "How do I get better at this?" That's how we got a list of consultancies in the area and one of those was cPrime. I reached to a few different companies and I think we talked with Zubin actually at the time. I was able to successfully pitch consulting which here is not normally … It's hard to get that kind of thing. I argued it as training, because we know what that means and we understand that. We worked with Kevin and maybe someone else at the time.

cPrime: *Can you describe the adoption "plan" that was prescribed for you by cPrime?*

Sr. Architect: *We came up with a game plan but we realized that over time, we'd probably grow and we'd need scale our training exercises, and what we found was we were concerned that we weren't going to be able to give a consistent message if I went to the Romania team and said, "Just go get some Agile training." Go to the India, "Hey, go get some Agile training." We were concerned that we wouldn't be communicating the same level of quality level message. Of course when we got Kevin as our trainer, his dry humor standing, he knows his stuff really really well and he communicates it really really well.*
We felt that we basically could offer a package, like here's what we want trained, here's how we want it, leave this in and add this out and here's how we do things. Kevin was able to convey that message throughout the world and it made more sense for us to send, and Kevin

was willing thank God, but made more sense to send Kevin out to the field rather than try to bring people in because it's much easier even as a consultant to send 1 person to India than it would be to bring 5 people here. We were able to create a little tour, if you will. We'll hit the east coast, we'll hit Romania, we'll hit India.

cPrime: *Yes, I remember seeing pictures of that trip from Kevin. Pretty amazing. He went to India, Romania, Colorado…*

Sr. Architect: *Yes. From that perspective, that was the starting point. Or course the challenge with Agile for us is you can train somebody but if they don't actually use it, they're going to lose it. We definitely had some hiccups in trying to get everybody fully engaged in Agile methodology and then even go a step further now; we're very interested in Scout Agile, getting everybody stepped up to Scout Agile. We're a little worried because we're not really very … To the outside world, we're probably not very agile. But on the inside, we're way more agile than everybody else. It's just a level of quality to the process.*

Now we're at the point where, granted people have come and gone, in the last few years but we're mostly trained up and we're mostly following some of Agile but we're not really good at evangelizing it and the constant improvement part. For example, retrospectives, all of many consider that to be one of the more important tasks. Almost nobody does those here. In terms of visibility across the organization, we still have teams using spreadsheets, which is absolutely absurd for a company with 29 sprint teams.

cPrime: *When you say "we", are you just meaning your group, the Customer-Support Portal team?*

Sr. Architect: *Yes.*

cPrime: *Okay. And now it's spreading towards other groups?*

Sr. Architect: *We're a large software organization and we tend to acquire lots of other software organizations and in general, given the propensity of software organizations to do Agile, it's more likely than not. Smaller companies, and when we mean smaller we mean billion dollar companies, that are going to come with Agile already. They're going to come using Jira, GreenHopper, Pivotal Tracker, whatever, VersionOne. Many do come with Jira, but they're also small companies already and they're small teams. They may be a billion dollar company but they may only have 30 developers or 50 developers or 100, not very many. They come with those tools and expertise already and so partially you've got probably dozens, if not*

100 I don't even know, organizations that are running Agile, but you've also got hundreds and tens of thousands of people who are not in our large organization. It's not my problem. I've got enough. We've got enough on our plate. We run a $20 billion service. It's bigger than most everybody's companies. That's just for us with 300 people.

cPrime: *So, when you say in the beginning you didn't know much about Agile but you talked to friends at Cisco. What was the main reason behind that? What were challenges were you facing and wanting to achieve? Why go there?*

Sr. Architect: *My role as an architect is mostly focused on user experience design. One of the things that drives me nuts would be that we come up with a solution to a problem and then it takes us 18 months to deliver it. That's not acceptable. We need a method to be able to deliver customer value sooner which is exactly what Agile does. As long as we can come up with a method for prioritizing things, which by the way, Scout Agile plus some of the methods I introduced do, we can figure out what's important and get it to customers as soon as possible. From a design perspective, we were interested in basically just delivering quickly. In the first 18 months, we delivered 8 releases which is 7 more releases than any other product would've delivered here because other products deliver typically in an 18 month cycle, and we delivered 8 full releases.That's 2 in a quarter month release cycle. Right. Now we're on a quarterly schedule which I base on Scout Agile. I prefer to go to a 5 time a year schedule is what I think would work best for us in terms of partnering sprints and timing and such. We were really just trying to get more … They were trying to be fast. We wanted to get stuff to customers quickly and try it out and get feedback and then improve the product. You can't do that on an 18 month schedule. In fact, services just … You can't run a service that way. No website gets released every 18 months.*

cPrime: *Now, how soon did you feel that after you initially said okay, we want to get training or consulting? How soon did you feel that impact?*

Sr. Architect: *That was December of 2010. That would've been 2 years after we originally founded the project. If you think about it, we're 300 people now. When we started, we were 2. Don't exactly need Agile training for 2 people to work together.*

cPrime: *You started doing it back then, with only 2 people?*

Sr. Architect: *We started pretty soon. For example, when you look at the Agile artifacts, did we have burn down at the beginning? No. Did we have standup meetings or retrospectives? No. What I refer to us as, I refer to us as lower case agile. What I mean by that is we were agile-like. The word agile, not Agile Manifesto-agile. I called us agile-like or I called us "lower case agile". That is we did deliver quickly. We repeated execution. We tried to take things off in small bites, but we didn't know about stories or story points or breaking down ethics into stories. We intrinsically knew to do the most important things first because I'm a designer and that's what I would've done anyway. We didn't know it from an upper case agile perspective. Once we got past, it was the VP and I and some contractors which lasted more than a year and we started to actually start to acquire internal company development resources. Then we started realizing we should do Agile. When we had a few dozen people, then it became like we should've done Agile, not Agile at scale of course because it didn't really exist then, but just simply Agile. Then we started to work on indoctrination of actually having a 2 week cadence. We had continuous builds back then. We've moved away from it. We're trying to move back but the technology made a bid difference back then. We were using Flex and everybody could do builds in real time. We really didn't have QA integrated with the team. Wanting to get value out to our customers quickly was our biggest concern and because we had pressure from our executive management as well. It's like, okay, I want something. They want it sooner rather than later. I'm like how can we get something of good quality out sooner than later. That's where we felt Agile so is not going to work.*

cPrime: *You didn't get any pushback from that decision?*

Sr. Architect: *No. I don't think there even was any discussion of that decision. That was an internal decision.*

cPrime: *When you were describing the plan that you came up with, Kevin did an assessment for you guys in the beginning to customize training for all of your teams that was somewhat customized to what you guys needed to learn.*

Sr. Architect: *It's on a Wiki page so I can show it to you and I can share it and print it, whatever.*

cPrime: *It was a 3 day..*

Sr. Architect: *Basically what we did is we looked through the agendas for the 3 day and we figured out what are we going to do to get. We took the 3 day class and we figured out what would we need for us. We scratched some things out basically to create the 2 day Agile class* *plus the 1 day product owner class. Most people, and these names match those pictures, but most people would go to the 2 day class. People who weren't doing story building, released manager. These are released manager, released manager QA didn't have to go. Developer didn't have to go. Developer didn't have to go. That of course allowed us to be more effective in how we were trained. We created what we called a cookie cutter Agile proposal. We basically worked with cPrime at the time to basically say okay here's this standard proposal, here's the standard schedule, here's what's in it, here's how much it costs. We could just keep turning the same item over and over again. I could make the argument for, especially at headquarters given that Kevin is here and there's no travel cost, I can argue wow, it's really cost effective way of doing business as opposed to sending a bunch of people to individual training.*

cPrime: *After the training, did you feel like that was efficient for you guys to start and run with it or did you feel like there needed to be follow-up coaching?*

Sr. Architect: *Absolutely, we should've had follow-up coaching. Absolutely. It's one thing to lead a horse to water. It's another to make him drink. If you can constantly engage people in the skills they were just learning, over time they're going to learn it a lot better than walking to a classroom, taking 2 days of class, getting to know the basics but now coming back. Are you really applying it correctly? What happens is it's like a control system. It's like a helicopter. A helicopter, traditionally, you have to control a helicopter with these 6 degrees of freedom. It can go in all directions plus speed. If you let go, it's going to fall. You need to keep that system in balance to make it move forward or make it move in the direction you want.*

For us, in my opinion, absolutely like there should be basically some form of Agile evangelists coach part of the engagement to continue to work with us. Even I think if you did the big bang which is this approach for Agile up scale, who there knows how to keep the ship going in the right direction? Granted you may have somebody on staff who is an Agilist, certified Agilist or whatever, an expert and they may or may not contribute to the team. cPrime coming in and giving training or something like that. That's there. That's great. We certainly had none of that. We had one certified Scrum Master in Denver who is no longer with us and that was it. No, absolutely I think that if you want to keep the ship moving forward, you have to have at least the harbor pilot on board. Your captain is going to be the leader of your internal system, but for that while you're still in the harbor trying to get out to the open sea, you need the harbor pilot and that I think is the coach or evangelist. Boy that's a good quote.

cPrime: *Good quote, I like it.*

Sr. Architect: *You need that harbor pilot.*

Most the people won't understand why a helicopter flies. The reason why a helicopter can fly *here is because a human can actually deal with all these degrees and keep the system in balance, in check because it's just chaos. A helicopter by definition is chaos to try to fly one.*

cPrime: *Good point.*

Sr. Architect: *It's going in all directions. One, 2, 3, 4, 5, 6, and forward and backwards, 7, 8. It's pretty complicated. To be able to control that stick, you have to be able to realize that it can get out of whack and you can get into these oscillations that you won't be able to recover from, but humans can be trained to recover from them. Otherwise you wind up with … You know how you drive one of these video game cars for the first time and you're always driving it into the one wall and then you drive it right into the one wall.*

cPrime: *The sensitivity.*

Sr. Architect: *It's so sensitive. You need to understand that when you're starting to come back to center, it's already time to try to get it back into the middle, not boom and boom. That's what I think Agile is for a lot of people. They keep running into the left side and then the right side. They're getting bruised. If you have somebody coaching you, you can not only start to move towards the center but move faster. If you play any video game, you can't go faster by keep running into the walls. You have to keep going down the middle and then you can go faster. I think that's what the coaching does.*

cPrime: *I love that. Do you think you have that now?*

Sr. Architect: *No I don't think we have. I think we've … Here's another analogy. Actually I have a chart for this but I have to go dig it out. The understanding of Agile has grown in the last few years, actually very very quickly. What you knew about Agile 3 years ago is nowhere near the corpus of information you now know about Agile because of Agile and skip pretty much. That's where all the evolution has come up. Somebody's expertise in Agile, the world's expertise in Agile has grown. Where has our expertise grown? Even if ours has stayed the same, we're now that much farther behind the curve because the curve has grown up. I actually think we've teetered down a little bit or maybe at best stayed even. That's because we haven't done the constant improvement.*

There's no mechanism. We're not having any mechanisms do that. I'm trying to fight that battle. This just is not good enough to go out there and say we're Agile. We had a team, the head of that team is no longer here but, she came to the meeting, I swear to God, she came to *the meeting and it was like, I don't know. It was a Tuesday or Wednesday meeting and she referred to it as a beating, not a meeting. She comes in and says, "We're going to be Agile on Monday." What? This is nonsensical. This is nonsensical. It's like telling somebody you're going to speak Chinese on Monday. How is that going to happen? Where's the infrastructure? Where's the process? Where's the methodology? You can't do that. It's not meaningful. For us, most of us know what it should be but there's no one leading the ship, the Agile charge. This is all that managers [inaudible 00:23:17] but who really owns Agile? I went to get the certified Agilist training class.*

I mean SAFe training. When I went for that training for SAFe, it was just great. I met a lot of people from other companies and they all were the Agile champions. They were the people who were-

***cPrime:** Those are the harbor masters.*

***Sr. Architect:** They're leading the charge. They're not in charge of the team, but they're coaching the captain. They're taking control of the process to get them out into the open ocean. A lot of those people there at that class; Bank of America or all these other companies, Frontier; they all had that. I look at our organization; I'm like I could do that. I have a day job unfortunately and expect for me to do that. If I had my way, that would be a role. Somebody would be that role. At least 1 person. We have 300 people with 20 sprint teams.*
At least 1 person would have that full time role and they could go every month and sit in on somebody … They'd be sitting on people's teams every week and they're coaching the coaches, coaching the product owners, helping us make sure that we're doing the right thing at the product owner level, working product management, and helping just improve oil the machine. I think that's where consultants can do a great job because the great thing about that is you can come and go and you really are leaving intellectual property with the company as opposed to me having a developer and then I have to train them on our infrastructure for 6 months and then they write the backlog and then they leave and all that infrastructure training went away. Whereas the coaching I think works great as a consultancy because when you're coaching them, that's what you're leaving on the ground is their experience and then hopefully they can train other people.

***cPrime:** Long term coaching.*

***Sr. Architect:** When you are off trying to maybe get some, which I think you guys should cut a deal, you should write a proposal and they should accept it. Let us help you. We'll put some* *numbers up. Not an hour of your guys' time and actually everything is done in an hour. It takes an hour to solve this.*

cPrime: *Overall, how would you evaluate what cPrime did? How would you evaluate Kevin as an instructor? The responsiveness of our people?*

Sr. Architect: *The reason why we went back to Kevin repeatedly in multiple engagements was because we felt like we got exactly what we wanted. He's very logical. He really understands his stuff. He delivered a consistent message. He's a nice guy. He's very low key of course. He is very sublime. It worked fine for us.*

cPrime: *He knows his stuff.*

Sr. Architect: *He knows his stuff. It was very easy to work with cPrime. It really didn't require a lot of effort on our side. All the materials were delivered. Usually room, facility, all that stuff. We had a great overall experience and I've recommended cPrime in multiple occasions already, off the record if you will, to other teams here as well as other organizations in the Bay area.*

cPrime: *When you were first evaluating, I guess, other training providers or other consulting providers, what stuck out to you about cPrime in the beginning? Did you meet Kevin right away?*

Sr. Architect: *I think the biggest thing was the referrals. That meant a lot. Of course I was doing them, not through cPrime asking for referrals but I was looking through my friends and that's where that came from.*

As long as the trainer was a good trainer, the rest of it is infrastructure and that it mattered because I only really had to interact with 1 person and that was Kevin. Granted logistics or given the India trip and getting Romania, it got a little complicated and I always try to be efficient. I found from that perspective, we're very fortunate and cPrime and Kevin were extremely flexible in letting us basically do an around the world round trip. Let's go to India, let's get that done.

Fly to Romania, get that done and make it a single thing that we can deliver an efficient timeline. It's not something that you can usually ask of a contractor. You came to me and said *you got to go to Romania and then you got to go to India and then you're going to go to China. Maybe the family, but I never felt at all that there was any issue. cPrime was very flexible and they saved me a lot of money doing the big thing approach.*

cPrime: *You said now you're hoping you can get approval on some more coaching or some follow-up stuff, but from your perspective now, what do you see as the number 1 benefit that this Agile and cPrime has brought to your work and your teams?*

Sr. Architect: *To date? What is the most valuable?*

cPrime: *Yeah.*

Sr. Architect: *Just being agile. Just the basics here. Being agile is a huge improvement and I don't think that so much of the selling point these days because almost everybody should be agile at this point. Those processes that are appropriate for Agile should be agile at this point. You're not going to build, like Kevin says, you're not going to build a nuclear power plant agile. Once you pour 100 million tons of concrete, you're not exactly going to go redesign it.*

cPrime: *Hardware, yeah.*

Sr. Architect: *Hardware. I think by far that was the biggest incentive for us is just to be agile.*

cPrime: *What one of the benefits from Agile do you appreciate the most? Your VP has said transparency in the past, do you agree?*

Sr. Architect: *Certainly I absolutely agree that transparency is one of those foundational things that is transformative. If you're a VP, you should be able to go within basically a 2 week window and see exactly where we are and using an Agile tool like version 1, for example, you can get a good sense of how your teams are progressing and where you're at. If you really do need to dig down into the mud which here is very common, you can. In Waterfall, there's really no visibility at all because frankly teams come up and say, "We don't have code yet. It will be another 3 months." Then 3 months come along and they're like, "We're still going to be another 2 months." Then they go, "Okay here's our first piece of code," and then your guy is, "What the hell is that? That's not what I asked for." It's not just the transparency from the perspective of the life cycle and the process.*

It's also being able to see stuff come out the other side and be able to get quick feedback on it whether it's from customers; in my case, I work with customers and I try to demo and I have working code that I can demo to somebody before I deliver it to production so I can continue to *refine it. We do this all the time. We did it last week. We did it the week before that. We did it last month. I have things coming out the other side and I'm not just dealing with prototypes. I'm dealing with actual working code that I can test. From a visibility perspective, not a transparency, a visibility perspective, our upper management can also see that stuff. If they wanted to come in and say, "Show me where you are today for something that's delivered in 8 weeks." I will have something to show them as opposed to I'm still writing the speck.*

***cPrime**: Do you think that with distributed teams, because Kevin went out to India and Romania, do you think that those distributed teams are working well now or better than they did before?*

***Sr. Architect:** Better than before because they're agile but not nearly as good as they should be if we were to organize around the Agile principles. It's really important with distributed teams that the distributed team be fully …*

I mean that if we have a team in India, they should be fully self-sustaining. They should have their own QA, their own product owner, their own development team, and everybody should be in India. That product owner is the only person who really needs to coordinate back with other product owners across the world. We're not quite there yet so we may still have QA in India but the development team is in Ireland or the development team is somewhere in North Carolina. It's hard for those teams that are so time zone challenged to work as efficiently together as they should. I think Agile in general, even when you're distributed, works better when the distributed team is fully self-contained. That's just common sense I guess. We're not where we need to be on that yet.

***cPrime:** Going back to tools, I know you touched on this and I've heard a few tools being thrown around. I know you just started. I know you worked with VersionOne.*

***Sr. Architect:** Yeah, we started on VersionOne which is a great tool and they're very good about updates and the quality of the products really really high. Very happy with it.*

***cPrime:** Now you've moved to Jira?*

***Sr. Architect:** I haven't moved to Jira. We haven't evolved yet to Jira but Jira is the corporate platform, standard platform for our organization, for our company for Agile, but not necessarily Agile at scale. Jira is a bug tool. It's a bug system. We us it as a bug tool and then we use GreenHopper on top of it for Agile planning. Out of the box, it's more like a tool kit. We are working on that to see whether that is a viable option for our team, but many many other teams may already be on it or are fine on it.*

cPrime: *That makes sense.*

Sr. Architect: *We need something that runs on a stack that makes sense for us. It runs on a software technology stack that's viable for our organization and it certainly has most of the Agile tools we had expected for a scaled Agile. If I was 1 team doing Agile, Rally there's probably overkill whereas Jira is simple, cheap, and it's probably fine. When you start talking about distributed teams that have a lot of relationships, blocking stories, dependencies, and then the visibility transparency- we need a tool to support that. I want visibility across that spread schedule.*

cPrime: *How are you on time?*

Sr. Architect: *I have an 11:00 and I'm presenting.*

cPrime: *Okay. Well I think that you have answered all of my questions. Thank you so much for doing this.*

Sr. Architect: *Good. Send me the webpage where you're going to post it so I can see my quote. Senior architect at one of the largest software development companies in the world.*

cPrime: *Sure will. Thank you so much.*