EPISODE 1104

[INTRODUCTION]

[00:00:00] JM: ADP has been around for more than 70 years fulfilling payroll and other human resources services. Payroll processing is a complex business involving the movement of money in accordance with regulatory and legal strictures. From an engineering point of view, ADP has decades of software behind it and a bright future of a platform company used by thousands of companies ahead of it. Balancing the maintenance of old code while charting a course with the new projects is not a simple task.

Tim Halbur is the CTO of ADP and he joins the show to talk through how engineering works at the company and how the organization builds for the future of the company while maintaining of the code of the past. I hope you enjoy today's episode.

[SPONSOR MESSAGE]

[00:00:48] JM: Your code is going to have errors, even code written by an amazing developer. And when bad things happen, it's nice to know that Honeybadger has your back. Honeybadger combines error monitoring, uptime monitoring and cron monitoring into a single easy to use monitoring platform for less cost than you're probably paying right now. Honeybadger monitors and sends error alerts in real-time with all the context needed to see what's causing the error and where it's hiding in your code so that you can quickly fix it and get on with your day.

The included uptime monitoring and cron monitoring also lets you know when your external services are having issues or your background jobs are going AWOL or silently failing. Honeybadger.io is 100% bootstrapped. It's a monitoring solution that's bootstrapped, and this is important, because a self-funded business means that they're priced simply. They operate the business simply and they answer to you, not investors.

Software Engineering Daily listeners can get 30% off for 6 months of Honeybadger, and you can simply mention Software Engineering Daily when signing up. They'll apply the discount to your account and you don’t need a credit card. Thanks for listening, and thanks to Honeybadger for being a sponsor.

[INTERVIEW]

[00:02:09] JM: Tim, welcome to the show.

[00:02:10] TH: Thank you.

[00:02:11] JM: You work at ADP, which does a lot of different things. Let’s start off with just the overview of what the company does. What are the most important services that ADP provides?

[00:02:23] TH: ADP is a human capital management company. Big words, but stands for basically doing a lot around the employee and their actions throughout the day. It includes time management. It includes benefit, and it includes probably the most important thing, the payroll. We run payroll for one in six Americans. We are in 140 different countries. So we run payroll in many countries.

So the employee is going to get a lot of value out of the services that ADP can provide around managing their time, their benefits so they can add beneficiaries, remove beneficiaries, understand what plans. We have a 401 (k) option. It goes on and on. So a lot of benefits and efforts around the employee and the actions that they would take.

[00:03:21] JM: Describe the canonical technical challenges of building ADP.

[00:03:26] TH: I think the number one issue is scale. Going back to what I said, one in six Americans, 140 countries, 9,000 developers. Anytime you’re talking this kind of scale, then you’re talking having to deal with complex calculations, regulations across countries. The privacy rules are different, where in one country you can expose what religion somebody is. In another country, you absolutely can’t. Various things like that really make it difficult to work for ADP, but make it difficult to understand how we can create applications for a wide variety of users. We have companies that are as small as one employee themselves. They run payroll for themselves. We have companies as large as 300,000. We have companies across, as I said, 140 countries. So all the rules and everything associated with that, the scale of that is incredibly challenging.

Another area where we are large and deals with complexities is around the breadth. As I mentioned, we have 9,000 technologists. But beyond technologists, we also have – We started hiring people with unique skillsets. We hire anthropologists now. We hire behavioral psychologists. We hire people that have skillsets in areas that one normally might not think of around a technology company, but we found that by hiring these types of people, then the skillsets that they provide, we can understand what our users are looking to do better and we can actually help our users.

In our financial tools, for example, knowing what the right nudges are to move a person from poor saving behavior to good saving behavior is a value to them and it's a value to ADP. So dealing with anything that large is probably the number one issue that we deal with.

[00:05:48] JM: You've been with the company for 15 years. How has your perspective on engineering changed over that period of time?

[00:05:55] TH: It's interesting where ideas that were valid 15 years ago kind of fade away for a while and then seem to come back. So it's interesting, when I started SOA, service oriented architecture, was the buzz word, and that was the focus that people were trying to do. And it kind of fell out of favor for a while, and now it's back again. It's just called different. Now it's called microservices.

I think that the thing that the thing that I takeaway over the number of years is the dealing with parts and pieces of applications is you deal with them on a much smaller and simpler basis. People building large monolithic centralized applications, we just don't see that anymore, and there's a good reason for that, because when you have a lot of different bits and pieces that you can plug together, you can find the best one of them and the cream rises to the top. And you can take the right parts and pieces and plug them together.

So, again, back when I started and we all started in 15 years ago, in 30 years ago and whatever, it was all about one big monolithic system. Now I don't see that as much anymore. I don’t know that we’ll ever get back to that simply because we have the knowledge of how to get the best out of individual parts and pieces.

[00:07:28] JM: How do you keep the full software architecture of such a large company in your head? Because it’s a lot of legacy components. There’re a lot of different divisions of the organization. There’re probably a lot of applications throughout the company where very few people even know what is going on in that black box legacy piece of code over there. How do you keep it all in your head?

[00:07:54] TH: I honestly don’t know if you can. I mean, I know I can’t. I tried and I cover a gamut when you talk about those business units. I was the chief architect for one of those business units for a period of time. Yeah, for that particular business unit, I knew a lot. But we have 5 to 7 different major business units. So trying to keep that all straight is basically I think almost impossible.

But what you can do is you can have layers of people, and that’s the way we work it, is that we have – And just from the architecture point of view, we have technical architects. We have principal architects. We have enterprise architects and we have chief architects. I think you need that layer of people. Again, we have developers as well and product owners and such. But from an architecture point of view, I think you need the layers of those people so that you can go find an architect or a developer who is working on a line of code. But you can also go find an architect or a developer that’s working on an application or a chief architect that’s working to try to make sure that they modernize or simplify a larger application.

I think it really requires that you depend upon the team that you’re working with and you run the gamut. So as I worked with teams, I would ensure that I talk to the one to the architects, not just the ones right below me and whatnot, but the ones below them and the ones below them. So you work that whole chain to try to keep it all straight.

You can talk about documenting systems and you can talk about automated applications that will map-out your APIs and your firewalls and all, and we run them just like everybody else does. But at the end of the day, we’ve got applications out there with 2,000 tables in one application. There’s no way for one person to keep that all straight. And along when there’s another 25 applications that they're also working with. So it has to be the dependency on your people that you work with.

[00:10:10] JM: Very simple question. How does money make it from one place to another? For example, you've got lots of payroll applications. I use ADP for payroll. How does money make it from a bank account to the bank accounts of the people that are being paid in the company?

[00:10:34] TH: So ADP works off of – It depends upon which approach you take. But most common one, a company will contract with ADP and they will have a bank account where they will deposit the money into an ADP account based on the payroll that we run. So we run payroll for them on say a Tuesday. They would deposit the money on say Wednesday based on what that payroll will be running. And then on Thursday or Friday, depending upon the bank and the company and the schedule, we would then work with the NACHA system, North American – N-A-C-H-A. I'm not sure exactly what the acronym is. We would work with that system to actually move the money from the ADP account into the 5,000 different employee accounts.

Now, the reason we do it this way is that the company then would give the money to ADP so that they did not have to deal with the actual moving of that money. We would take the responsibility of moving it from a centralized location to the 5,000 distributed locations.

[00:11:48] JM: What are some challenges of dealing with all the integration points around the ADP infrastructure and you have to integrate with banks. I assume there's other kinds of companies that you need to integrate with. What is your strategy for managing lots of third-party integrations?

[00:12:08] TH: ADP was one of the first companies to come out with what's called the ADP marketplace. We have the ADP marketplace where companies can – Both companies and partners can register to integrate with ADP. So an individual mom and pop accounting firm could register with ADP. And something as large as Chase Bank could integrate with ADP. With the idea being is that with the definition of the APIs and unfortunately file movement as well. That's prevalent in the financial industry. You can register to both move money to register, to get account information, to get employee information, etc. If the company okays it, so if the shoe repair company down the street okays it, they would say that mom and pop accounting firm can access my applications and my data so that they can provide a service to me and/or maybe they say you can access the employee data, but not the employee's financial data, for example. So there are different levels of authorization. But that's how we – Using the ADP marketplace, we allow developers, partners and companies to access our data with the right authorizations to allow them to pull and push data in an out of ADP systems.

[00:13:39] JM: What about tax reporting infrastructure? There are lots of minutia that needs to be handled in calculating taxes that would be paid on employee salaries. What's the – One thing that you do is basically codify taxation infrastructure or accounting rules. What's the process for codifying that and checking it?

[00:14:04] TH: So that's actually one of the primary reasons a large number of companies sign up with ADP, because ADP will stand behind you when we do those calculations. Many payroll companies will do payroll for you and they will give you the numbers and you will then start paying your people and you'll pay the government and will work out. But will end up happening is that if the IRS comes calling, you may be standing on your own.

ADP doesn't do that. ADP stands behind the calculations they do. We actually have an entire group called the stat group, statutory regulations group whose job it is is to manage and to find out all the tax implications and changes that happen if a city council pass – And this happens all the time. If a city council passes a new regulation that says 1% of everybody who lives in this city and works for companies greater than a size of 10,000 needs to pay extra money. And then our statutory group will be responsible for finding out about those changes, documenting those changes and getting those into the programmers so that they can make the appropriate changes to the application systems. So it is an entire group, an entire statutory group that is responsible for managing those regulations.

Along with that group, we also have a group that is responsible for dealing with the federal government. We have a group that spends their time talking to the senators and the representatives about IRS policy. Recently, there was a change to the W-2 form. The ADP was one of the companies that was selected to pilot this change to work with the IRS to design and change, to pilot the change, and to then roll the change out. So now there's a bar code that will be on W-s. That type of information, because of the size of ADP, because of the scope of our government and statutory groups, we're actually able to understand, recognize and make those changes.

[00:16:41] JM: Tell me more about the legacy application stack. So the older applications, how do you ensure that they get managed properly and that you have proper coverage as the different people in the organization who might have had their dominion over the applications. It might age out. They might rotate to different parts of the company. How do you make sure you have the right areas of the application covered?

[00:17:12] TH: So we've been doing just like everybody else. We have next-generation platforms. I shouldn't say everybody else, but most everybody else. We're developing next-generation platforms in tax, in payroll, in benefits, etc. So we have our legacy applications.

That said, we're spending a fair amount of time working with those legacy applications to modernize those applications. So we certainly have the next generation of our applications, but we're actually spending a significant amount of time working within those applications to develop new and improved ways of those applications working. For example, we're moving to containers. You can take a legacy application, wrap it in a container and now you're getting a lot of the benefit of what containerization can provide and the modern application stack can provide and yet still some of the legacy application.

Along with that, while we're doing that, going back to the microservices, we're actually evaluating our applications for what parts of our applications should be re-architected into a microservices organization application. So we take applications that are monoliths that were based on, as you said, legacy programmers, legacy skillsets, and we both modernize the application and we both – And we look to convert the application into the services that it's made up of so that we can take those bit piece parts and combine them together along with other applications.

One of the things, for example, are payroll engines. Payroll, for many people, they think of it as hours' work times your rate, and you take out a little bit for the government, and it's all good. But the reality is, is that it's not anywhere near that simple as much as we'd like it to be. And so there's a lot of bits and pieces of the payroll applications that we believe very strongly is the secret sauce that ADP has and the benefit that ADP can provide to companies. So we want to make sure that we don't lose that. So we are not just abandoning those legacy platforms. We're actually evaluating them and repositioning them.

[SPONSOR MESSAGE]

[INTERVIEW]

[00:21:32] JM: Are there are pain points around the deployment process for legacy applications?

[00:21:39] TH: There's always going to be pain points around legacy applications, but there's also opportunities for improvement, and that's the way we're kind of approaching it. As I mentioned, one of the things that we're talking about is wrapping applications within containers. Once you wrap it within a container, now we have the opportunity to apply automation to that deployment of that container. Whereas in the past, you might have a COBOL application or a Java application deployed into CICS around to the mainframe into DB2. Now what we're able to actually do is take those applications, again, break them into smaller parts and pieces, wrap them into current technologies and use automation to do deployments of them. So really, we found that when we started trying to work towards consistent services for deployments that we were able to actually wrap our legacy applications into some of those same processes.

In fact, one of our legacy applications is probably one of our furthest along with doing that because they saw the benefit. Going back to that aging workforce, they saw the benefit of having this available to them so they didn't have those concerns, and they grabbed it and ran with it. And so we now have one of our legacy applications probably the furthest along in our automated deployment processes simply because of that reason.

[00:23:16] JM: I think technical debt is just unavoidable as an application gets older, and the older an application stack gets, the more the technical debt and just legacy applications become an issue. I want to talk about the brighter sides of having a large platform, a large well-developed application. But I would like to focus a little bit more on technical debt and how to relieve or have a constantly in place relief process for assessing technical debt. Do you have any additional strategies for how to re-factor these problematic black box areas? I just remember, I've worked at a few different companies, and there're a lot of different strategies for dealing with these legacy applications. Sometimes you put a shim over it and you kind of like build an entirely new API because you have some part of a legacy application that is just – It can't be really understood. And so you have to sort of refactor it and reframe what this thing does. Do you have any stories or lessons for dealing with that kind of technical debt?

[00:24:20] TH: So we do, and we started – This started because of AWS and movement to the cloud, but it really came about, we were able to take that movement and that effort and apply it, as you said, to legacy applications as well that may or may not be moving to the cloud. And that is around the well-architected framework.

So AWS publishes a well architected framework that allows you to evaluate an application and determine some of its bit pieces and parts and to determine where there are some weaknesses and some areas of concern, etc. We were able to take that well-architected framework documents and process and we actually applied it to our applications as well even if they may or may not be moving to the cloud. We had to modify it because it's obviously focused for applications and their movement to the cloud and we weren't necessarily looking to move all of our applications there. But it did give us a consistent platform on which to evaluate applications and determine where, what application we should do, what with?

For example, should we just re-architect this application? Should we retain it? But we should containerize it? Should we rewrite it? Should we just re-platform it? Should we take it and should we break out different parts and pieces? So taking the well-architected framework approach that AWS provided, modifying it for ADP specific benefits really allowed us to evaluate our legacy applications and determine how to handle some of that technical debt and eliminate it.

[00:26:16] JM: So there're a lot of different surfaces and a lot of different application points within ADP, and you might want to prioritize which of these to modernize overtime. So maybe you have one application over here that you would like to see in the cloud. You have another application over here that you would like to move to continuous delivery. What's the process of prioritizing what areas of an application should be modernized?

[00:26:42] TH: We have a center of excellence, and that's one of the things that I think a lot of people benefit from, is to have centers of excellence and architectural review board type groups, depending upon different companies, call it different things and do different things with it. But to have a group of folks that are able to evaluate some of these platforms and services, going back to we take all of our applications and we have them go through our low architecture framework review. We then come away with some findings as far as what the direction this application should take on various parts and pieces.

But as you mentioned, there's always a prioritization, because with unlimited money, we could do unlimited things. But we don't have that. And so then the center of excellence would get together and review that and help the product owners and the strategy team make a decision as far as which ones of these we should focus on.

For example, ADP, I'm talking about things like cloud, which is very good around the scaling up and scaling down. We run a lot of activities that take place at a specific time of the day, of the weak, or of the month. So, for example, at 9 o'clock Eastern Time, or pretty much all, 9, 10, 11 o'clock Eastern Time on the hour, we have a lot of people that look to clock-in. And at the 5 o'clock Eastern and so on, we have a lot of people look to clock-out. So we need extra bandwidth and extra processing power at those periods of time, but we don't need it the rest of the periods of time.

So somebody would take a look at this and say, "Is this a good application based on the needs that it has based on the review of its scope to understand which ones of these applications might benefit from moving to the cloud?"

20 years ago, we weren't running in a cloud. We're running in our own data center. It was a private cloud, but it was fully funded by ADP. So you are paying for all the hardware. So you needed the full breadth no matter the burst capability. So by modernizing it, by evaluating which applications could take advantage of things like scalability of the cloud, we're able to decide which application we should focus on first, second and third of that modernization effort.

[00:29:18] JM: Tell me more about how you choose which public cloud services to use. I've heard different companies will have strategies like they'll say, "Okay. We're only going to use this particular cloud database. We're only going to use this particular relational database or keyvalue store. How do you prioritize which areas of a public cloud to use or how do you – Sorry. Narrow your focus to which areas of public cloud to choose?

[00:29:44] TH: So ADP started with a pretty broad brush. We started taking advantage of a lot of those different services across a lot of our organization, and there wasn't a cohesive picture necessarily. But we took a step back and evaluated that and said, "We're gaining great experience and knowledge of how these services work and the benefits of the various services." And so it was okay, but we did start to then – After a period of time, start to take a second look at this and say, "Okay. Now that we've had time, we've had a number of different applications, try a number of different services, let's take a step back and evaluate which one of those services is actually adding us the most value.”

If you take a very simple one like container management, and we are primarily an AWS shop. If you take a look at it, they have – You can run Kubernetes on EC2 instances. You can run their ECS service. You can run their EKS service or you could run their Fargate service. And so they have a number of different approaches just to solving the one problem, and we had a number of teams evaluate and look at and actually try to roll applications out on a number of those different options. Then we took a step back and said, “Okay. Now that we've done that, we've gained the experiences to which ones are helpful and which ones are hurting us and which ones are more beneficial than others, we’re able to take a look at that, evaluate which is the best of the 3, 4, 2 options and then start to focus our efforts on that particular option. And so we've done that across APIs, across networking, across storage, across containers, etc. So it was a process by which we generated approaches at a number of services, sort of the shotgun approach, and then evaluated it and then came away with just focusing on the individual ones that made the most sense.

[00:31:56] JM: What has the data platform evolution look like from your point of view? So I did many shows about data platforms and the relationships between transactional data stores and analytical data stores and processing. So you often have these transactional data stores where it’s like storing the user data or storing the different records of transactions. Then you might have analytical jobs that you might want to run to aggregate large sums of data and use different databases, different processing systems for this. What does your data platform look like today?

[00:32:34] TH: So data, as stated, is our middle name, automatic Data Processing. And we feel very strongly about data. We feel that the value that the data can provide is something that can benefit obviously ADP, but can benefit our clients as well. It can benefit the industry as a whole, and not put too fine a point on it, it can help the consumer in general. So we take a look at this, and one of the things, for example, that ADP does is publish a jobs report. And we do that and we provide that and make that data available to anyone, including our competitors and including the governments and whatnot else for the purpose of trying to make data more important and valuable to everyone.

So we've gone and we created out of our labs environment, we created a product called data cloud that we have rolled out for – It's probably been three years now that we have rolled out and made available to our clients and at times made available to the general public for the purpose of letting companies understand their employee base, helping them to make decisions on where to build new locations, where to hire more staff, where to focus some of their attentions based on their employees, their locations, their behaviors, their attitudes, etc.

Because ADP does have this wide breath of data, we take our transactional information, as you mentioned, and roll it up into an analytical format into data lakes and make it available so that people and companies can make decisions that will help them. So it's incredibly valuable to ADP and incredibly useful to the industry as a whole.

[00:34:43] JM: What has the term digital transformation meant from your perspective?

[00:34:49] TH: ADP, and many people, myself included, believe that digital transformation is around taking the manual effort out of human task, automating them and therefore giving the people actually more responsibility. One of the things people often look at is, "Hey, if you automate a task, can you then eliminate a position?" And that's not the way ADP looks at it. ADP looks at it, can you automate a task so you can give that person more responsibility. Taking our developers as an example, taking away, providing automation to the development pipeline allows a developer now to not just focus on development, but they can also focus on the building, the shipping and the sporting of that same application.

So now we've taken people that were focused in a specific area doing a specific manual effort and we're actually able to turn them into individuals with a wider breadth of responsibility across both larger number of applications or deeper within their application and make the job actually more interesting than it was before.

So a lot of people worry about digital transformation. They feel that it's a risk to jobs. ADP is approaching it actually as a benefit. It obviously is there to remove manual effort. There's no question about that. But it's there to do it for the right reasons, which is actually adding value rather than what people worry about with the elimination of jobs.

[00:36:43] JM: Does that actually play a role? Do you actually talk to people in the company who are afraid that their jobs is going to be eliminated by, for example, the addition of infrastructure as code?

[00:36:55] TH: Yes, there are people – And this is common throughout. We have people whose job it is, is to evaluate software license compliance. And one approach to doing that is to manually review software license compliance. But a better way to do that is to have tooling go through and evaluate the software licensing and highlight the issues. Now we can have a person instead of manually scanning applications and manually compiling results, we can have people work towards identifying what are the software licenses we should be working towards and getting those rules applied to all applications stead of one or two applications. So we're actually taking these people and increasing the breadth of their scope by doing it this way.

So certainly when people first hear about digital transformation and we hear about customer service reps wanting to use chat bots instead of having people answer phones, having a chat bot handle this, people worry that what that means is we want to get rid of customer service representatives. And what that really means is we want customer service representatives focused on the task that chat bots can't do. Because going back to what I mentioned, payroll is complex. Benefits is incredibly complex. And in order to understand and help our customers with that, we want our people to be focused on the right set of problems. And by taking away some of these lower-level problems, having the automation with the chat bot with an automatic scan, with infrastructure as code so deployments go smoother. Having those pieces in place allows us to actually have the people focus on the higher-order issues. So we do have people that are concerned about that. But I honestly feel that that we are giving them a better solution rather than focusing on elimination of jobs.

[SPONSOR MESSAGE]

[00:39:21] JM: Errors, and bugs, and crashes happen all the time across my software. Most often, these crashes have to do with obscure exceptions that come from React components, failing to render on the client device. Source maps and stack traces would be useful, but in many cases, I'm not able to identify the root cause because the error is occurring on a client device. It’s not on my infrastructure. And what can I do about that? I can use Sentry. Sentry.io can quickly triage and resolve issues in real-time.

Visit sentry.io/signup and use code SEDAILY to sign up for two free months. You can simply choose the platform that you’re integrating with. Sentry works for every major language and framework, from Rails, to C#, to Java, or React Native, and you just install the SDKs. You integrate it with your application. After that, errors will be caught by Sentry and you’ll be alerted the moment that they occur so that you can triage, and resolve, and keep your users happy with functional applications. If you’re building an application and you want to monitor your errors and your performance, try Sentry by visiting sentry.io/signup. And new users can use code SEDAILY for two free months.

Thank you for being a sponsor of Software Engineering Daily, Sentry, and you can go to sentry.io/signup and use code SEDAILY if you’re curious about it.

[INTERVIEW CONTINUED]

[00:40:53] JM: And what about hiring? Hiring at scale? What is the hiring process for bringing on more engineers at ADP?

[00:41:03] TH: ADP has started – First off, just let me preface this by I am not responsible for a significant amount of hiring. I contribute, but I do not hire. I’m a sole contributor in my role. What ADP has been doing more of is focusing on some of our internship and college hire opportunities. Trying to look for people that are interested in innovation, interested in new approaches, because we want to make sure that we continue to get a talent base that has a breadth of skills that is not looking at doing it the way it's been done just because it's the way it's been done, but looking at doing it what is the right way to do that.

One of the things that we did over the last five years is we started opening up innovation centers in different parts of the country. We have one in Pasadena, one in New York City, one down in Brazil, etc., where we actually put these innovation centers right amidst some of the top names. In our New York office, it is surrounded by Google, by Microsoft, by other well-established innovative companies as well as some of the new companies on the marketplace, Spotify, etc.

It's in what's called what many people call – Instead of calling it Silicon Valley, they call it Silicon Alley. And we do that so we could start to attract a different type of talent. It isn’t that it’s better or worse, but it's different, and get some of the skillsets that’s associated with some of those locations into the ADP environment. Having people like that and skillsets, as I mentioned, behavioral psychologist, an anthropologist, having those types of roles in ADP has allowed us to really broaden the scope of talent throughout the entire company.

[00:43:14] JM: And when you have acute engineering issues, do you ever hire contractors? How do you utilize contractors at ADP?

[00:43:24] TH: ADP does hire a fair amount of contractors. In fact, I have one of the projects I'm working with is our ventures group. Our ventures group is, just as it sounds, ADP spun up a group to evaluate new ideas within ADP and we didn't want to necessarily commit large number of people to those efforts until we find out how successful they are going to be. So there is a case where we absolutely use contractors to come in and help us design, build and execute a system that will then be available and sold as an ADP product and determine whether or not there is a market for this type of application.

It could be – It's usually within the scope of what the rest of the human capital management portfolio is, but it doesn't have to be. And that's where we found that using things like contractors to come in and help us augment staff in particular locations like that allows us to quickly and rapidly produce an application, evaluate the success and value of that application without having a high cost of employee staff.

[00:44:45] JM: How do you orchestrate communications between different teams?

[00:44:51] TH: So one of the things, COVID was an interesting – Obviously, it’s a phenomenon that is happening to us and it's directly affecting everyone. But one of the things that ADP has always done is have a work-at-home and collaborative approach. So we've always offered and allowed people to work from home. So you've always had a mix of some people working at home, some people working in the office, some people working part time both places alternating between the two. So we've always had to deal with collaboration and communication.

So we use a variety of toolsets. We use WebEx teams. There's a fair amount of cloud software, cloud infrastructure, applications that we use to communicate and collaborate between our application dev teams to ensure that everybody stays on the right path and stays together in alignment. So ADP adapted Agile a number of years back. So all 9,000 of our technologists are in a set of Agile teams. Some of those are scrums, some of those are Kanban, etc., but they each have – Many of them have standup meetings every day, etc. So we do – Because of COVID, it has pushed us to do more of it, but we were always doing a fair amount of that collaboration and communication, because with distributed teams it’s required.

[00:46:34] JM: You mentioned the ventures efforts a little bit earlier, and when you have an application that’s on the critical path of something like payroll, then doing experimentation is always kind of tricky. How do you provide experimentation and freshness in a company that is on such a critical path?

[00:46:59] TH: ADP has always been working towards innovation. We were some of the first to deal with, if not the first, to have a mobile HCM application. We were one of the first, if not the first, to have a marketplace application, human capital management marketplace application. We recently won an award from New York City dealing with startups. We were voted one of the top startups in New York City. And you would think about that and you think about a company the size of ADP that had around for 70 years and has got close to 60,000 employees, and yet we still were considered and voted on to be a startup. And that's because we realize that the core technologies and core values that require ADP to focus on our payroll, our human capital management, etc., those are critical and we never lose sight of them.

That said, we do understand that it does, as you mentioned, have to stay fresh. So for example, we recently started looking at payroll on demand. I mentioned a while back ago, many companies run payroll on a weekly basis, or biweekly, or monthly basis. They might kick it off on a Tuesday. They might finalize it on Wednesday. They actually run the reports on a Thursday. And now that we've changed to is to have a model where you can run payroll at 2 o'clock on a Monday and you could run it for one person or you could run it for the entire company, or you could not. You could run it Monday, Tuesday and Wednesday. You could run it all three days and pay people each day.

We really feel that while payroll, while benefits, while retirement services, etc., time management, are all critical functions. We also always recognize that we've got a be looking for innovation and new approaches even within that and look to evaluate things like payroll on demand or the use of pay cards. Especially with COVID going on, people are less interested to get actual paper checks. They're not as able to get to their banks. So what we found is we're able to deliver money. We've always had the ability to deliver money to pay cards instead. More people are starting to take advantage of it. So we're looking for ways to how to enhance their value to the employees so that they can get more out of those pay cards.

[00:49:47] JM: Just to wrap up, there were multiple companies that you've spent more than 15 years at, which is amazing. That is a lot of dedication to spend 15 years at a company, much more so to do it at two different companies. How do you stick around for so long?

[00:50:06] TH: Darn! Good question. I enjoy my job. So back to my history, yes, I worked 15 years for one large firm, Prudential, and that I work – Now I did a startup in the middle. I did a couple other jobs in the middle. And now I work 15 years for ADP. And I think the key to everything that I've done is having flexibility to work and do a wide variety of jobs. I know many people give large companies a bad rap. They think, "Well, once you get in there, you're just a small cog in a big machine and nobody's going to listen to you."

Again, I did a startup as well. And so when you do start up, everybody thinks, "Oh! You've got complete control over everything and it's so exciting and new and fresh." And what I actually found was I enjoyed working for a large company, not because of the security of a large company or being able to get lost among so many other people, but instead of the opportunities that it gave me. So I've worked in development. I've worked in support. I've worked in strategy. I've worked in multiple different business units. I've had done architecture as well as development, and I did that in my previous 15 years as well. I've moved across different business units. I've moved across different application teams. And having that ability to move around within a large company, I've actually felt that I've turned my job into multiple different jobs.

My current job is chief architect strategy, I'm focused now on strategy. It's not something that I thought I would be doing three years ago, but I'm starting to do it and I enjoyed it tremendously.

Where I go from here, well only time will tell. But I think that the key is looking beyond the boundaries of just what's in front of you and what's the job right in front of you, but what's out there throughout the company. What's out there throughout the industry?

So one of the things that I do is I try to make sure that I stay curious about what else is going on in the industry. What else is going on both with our competitors and outside our competitors, and then try to look to apply some of those new ideas into my current job so my job always stays fresh.

[00:52:44] JM: Tim, thanks so much for coming on the show. It's been a real pleasure talking to you.

[00:52:47] TH: Thank you very much, and I look forward to chatting with you again, Jeffrey.

[END OF INTERVIEW]

[00:52:59] JM: You probably do not enjoy searching for a job. Engineers don't like sacrificing their time to do phone screens, and we don't like doing whiteboard problems and working on tedious take home projects. Everyone knows the software hiring process is not perfect. But what's the alternative? Triplebyte is the alternative.

Triplebyte is a platform for finding a great software job faster. Triplebyte works with 400+ tech companies, including Dropbox, Adobe, Coursera and Cruise Automation. Triplebyte improves the hiring process by saving you time and fast-tracking you to final interviews. At triplebyte.com/sedaily, you can start your process by taking a quiz, and after the quiz you get interviewed by Triplebyte if you pass that quiz. If you pass that interview, you make it straight to multiple onsite interviews. If you take a job, you get an additional $1,000 signing bonus from Triplebyte because you use the link triplebyte.com/sedaily.

That $1,000 is nice, but you might be making much more since those multiple onsite interviews would put you in a great position to potentially get multiple offers, and then you could figure out what your salary actually should be. Triplebyte does not look at candidate's backgrounds, like resumes and where they've worked and where they went to school. Triplebyte only cares about whether someone can code. So I'm a huge fan of that aspect of their model. This means that they work with lots of people from nontraditional and unusual backgrounds.

To get started, just go to triplebyte.com/sedaily and take a quiz to get started. There's very little risk and you might find yourself in a great position getting multiple onsite interviews from just one quiz and a Triplebyte interview. Go to triplebyte.com/sedaily to try it out.

Thank you to Triplebyte.

[END]