



Improving Human Efficiency with Hyperautomation



CAPTECH TRENDS PODCAST | EPISODE 26



Vinnie:

Hello everyone and welcome back to CapTech Trends. Today, I've got again, Bill Klos, a CapTech fellow out of our Columbus office. He's been on the podcast a few times before. Welcome back, Bill.

Bill:

Thanks, Vinnie. Thanks for having me.

Vinnie:

Yeah. So, today we're going to talk about automation. The term hyperautomation has become trendy, I guess. Talk about how we see that in the marketplace right now, what companies are doing, how people can get started, the value of it, et cetera. So, as a way to kick this off, what do you think of when people talk about automation and hyperautomation? Where does it live? What is it?

Bill:

Being a technology person that always goes to technology first for me, but based upon some of the definitions that we've seen and talked about before, there's two parts to it. I was just doing a little bit of research before this podcast and from a Harvard Business Review perspective, it was very much a business-driven process. Identifying where the decisions are and then automating down to that level. And from an IBM perspective, it's more from the bottom up. It seems like it's just automate all the things. And so, anything that can be automated from an electronic standpoint, IOT, however you want to do it, that automation would occur. And it sounds like what they're both trying to do is meet in the middle for how they're actually going to join that.

Vinnie:

Right. So, let's do some examples here. Because I get a little bit confused when I think of automation versus what's a set of web services. A lot of the stuff that we write in our coding is a type of automation because we're having computers do things on our behalf, but we don't consider that automation. Right? So, if I take a picture of a check and have it get deposited in my account without having to leave my couch, that's an automating a process. Even though it's just going through well-orchestrated APIs.

Bill:

Right. Right. But, it's also not very flexible, right? There's my check... Is the picture clear? Is the account number on? All those things have to take into account. There's still a lot of human interaction in that. So, if you depend upon your bank, I imagine, but mine basically says, here's what you need to put on the check on the front and the back. And when you send it up, there is a delay. There's a delay as far as that check being accepted by the bank to do. Is it legible? Is the number correct? I don't think you're checking for the balance and so forth, but is there an account number on there and are there other things on there? Is there a date and so forth?

And some of those things can be automated, but some of those things are not. So, if you look at it, typically a check is pretty much set as far as where the fields are, what information goes in them. So, there's a little bit of standardization that goes along with that, but in a full automated process, it would basically go through and do everything soup to nuts to use that old term and essentially put that money in my account if that money exists in the check that's being given to me.



Vinnie:

But, I guess partial automation is still automation.

Bill:

Yeah.

Vinnie:

From a user experience perspective, I didn't leave my house. I didn't have to type any information in the fields. Right? I just took a picture.

Bill:

It is, unless there's something wrong with it because then you have to do it again.

Vinnie:

Correct. And so, the difference in my mind between automation and hyperautomation, and again, I'm kind of like you, I go a little too techy first. I think of automation as a well scripted decision tree where we can automate obvious decisions and tasks based on concrete data. Where I think of hyperautomation is on one hand to your point, automating everything, but also incorporating some machine learning and artificial intelligence to infer some things and recommend some things that aren't as discreet as a sequel query, for instance.

Bill:

Yeah. I could get behind that. I would add something to it. I was thinking about the services aspect again from a technical perspective and it comes down to how micro you get with them. And if you have a function out there that is say a web service, for example, and it's got all sorts of if statements in it, what kind of assumptions is that service making on behalf of whatever rules are being done? Or even if there are no rules, there are some defaults that are being done as well.

Hyperautomation to me would be actually removing those or breaking it down to the level where even those are being brought in by data or rules or AI. Those types of things. So, the actual code is making very few, if any, decisions at all. And those are all being done from external forces that are being controlled in real time possibly or based upon market effects or things that AI did or feature flags or something from an external perspective that the business does. So, that the code is not really making any decisions at all, specifically in the service, but it's being done from a different layer that's handling all the decisioning.

Vinnie:

Can you define feature flags for those who may not know what that is?

Bill:

Feature flags are used by a lot of companies now, especially in the development process, where there may be some new functionality and new version, a new type of function or feature they want to test, but they don't necessarily want to turn it on right away. Or they want to be able to turn it on, see how it works and if it behaves correctly, they leave it on. If it doesn't, they can turn it on very quick and not have to necessarily go through the entire process of pushing software through the SDLC in order to get it up there.



But, it also could be used for, I've got all these functions, say for example, a short term sale or some other type of thing that only exists for the period of a marketing push that you're trying to do that they can basically say, I don't want to change the behavior of the software, but for this time period, I'm going to put out an extra percentage off on sale for these types of things if you meet these other criteria. And you would go through and basically say, turn it on, runs through the process for the amount of time period, and then turns it off and then it just goes without having to make modifications to the software.

Vinnie:

Right. So, it's remote configuration?

Bill:

Essentially so, yes.

Vinnie:

Yeah. And right. And so, in layman's terms that all we're saying is that the company who owns the app that you downloaded is able to turn things on and off when they want to.

Bill:

Right. Exactly.

Vinnie:

Yeah. So, going back to the definition of automation and hyperautomation, it's not easy, right? Because you and I are talking through this because you can think of individual, let's say microservices. We've got some microservices in the cloud. Any one of those microservices maybe is automating something, but typically not. It's typically an orchestration of many of those microservices. So, I think of orchestration architecturally as a means by which you can achieve automating a process or set of tasks. Does that make sense?

Bill:

That makes sense. And then as you change the process, and this is where I guess I was talking about decomposing those down to relatively more granular levels, is that if you want to change the behavior of it, can you change code or do you just change which other services are being called by that one particular service? So, if you do it in detail enough, then you can do it from that way, change it by the calls as opposed to the code.

Vinnie:

Yeah. So, you're getting into well architect... If we're going to say hyperautomation... Now we're getting what does hyper mean? Right? Because you're getting into well structured automation, right? Because look, we've had rules engines for a long time, but no one claimed a rules engine was automation. Right?

Bill:

True. But, it is. It's funny because they don't really call them rules engines as much. They call them decisioning platforms at this point because it's also... It's a rules engine.

Vinnie:



I dated myself, didn't I?

Bill:

Only because I happened to be on a project a year ago that dealt with that. But it becomes a suite of tools now that is wrapped around decisioning. And that's just one of the components.

Vinnie:

And I like that idea of rules based engines and orchestration as being part of automation. Because I think, unfortunately, a lot of the people in our line of work, when they think of automation, they think of outside the architecture. So, for instance, RPA. I have these two systems. I've got to go to system A, pull some information, put it in system B. I can integrate that at the glass using RPA. And they think of that as automation. As opposed to saying, well we could have a couple well structured API calls and have it do it in the cloud as opposed to doing it at the glass.

Bill:

Yeah. That's fair. And it is, but I think it's a very brittle type of automation. And you're also potentially automating bad processes simply because, I'm going to use the term even though it's not very popular in the RPA group, of screen scraping. So, I can go through there. And as long as everything stays the same and things are filled in a certain way, I can blow through and I can actually do this automation, but if something changes and we've done projects like this before. If the front end changes for whatever reason, all of a sudden my process is broken, but if you've got a bad process, then you're now beholden to the RPA. And if you need to change it, that becomes a critical piece that you have to modify in order to make sure that automation continues to work. Right?

So, it's more like a surface layer of our automation as opposed to, I would consider, bottom layer or lower level where you actually have access to data and rules and all the things that go along with it.

Vinnie:

Sure. So, this is a public service announcement, right? RPA has some really good use cases, but ultimately it stands for really poor architecture. Right? If you grew up learning about enterprise and application architecture, one of the big things you stay away from is tight coupling, especially at the view tier. And that is by definition what RPA does.

Why would people care about that? Well, one point you made, which is anything changes on the presentation of one app, it can break the system and be difficult to repair.

But, the second I like to think about too, is to the extent that you're applying RPA to legacy systems and you've got 20 or 30 of these things. Now you can't sunset your legacy systems without breaking important processes and workflows so you become handcuffed to legacy and out of date systems unintentionally.

So, probably spending more time on RPA than I wanted to, but when I do talk about RPA with clients, it's debt. It is technical debt.

Bill:

Mm-hmm (affirmative).

Vinnie:

It can be well structured technical debt. And as long as you go into it knowing that, and maybe I



wouldn't use it for a production system, but maybe something back office or something. If you know how long you're going to use it for and how you're going to get back out of that debt, then sure it can help. Maybe it can help you get past a 6 or 9 month hurdle before the next upgrade for something you want to do, et cetera. But, if you look at it as a silver bullet, as opposed to technical debt, then you're going to paint yourself into a corner.

Bill:

Right. I totally agree.

Vinnie:

Good. So, we've beat up what automation is. One other thing I wanted to talk about it though, is I did mention machine learning and AI, as part of this. I want to get into that a bit more, but the second part of this is decisioning above and beyond what a human can do. So, sometimes it's not about just replicating a human task, but it's doing something superhuman or at a superhuman speed. For instance, a Tesla recognizing a stop sign or that cars two or three ahead of have brake lights on and applying the brakes. Seeing it, responding and doing it faster than you could or even the cars talking together in a mesh network and some sort of future implementation.

Bill:

By the way, Vinnie. I'm sorry to interrupt. I took my first ride in a Tesla a few weeks ago. I was riding with Christine and it is the most... The thing you said is amazingly true. There's a stop sign up there. A stop sign shows up on the little thing. If there's a person walking by, it's like little ghosts of cars and ghosts of people walking by. It is amazing and somewhat mystifying at the same time. It's just a little bit weird.

Vinnie:

It is weird and it can do it at a speed... Now what's interesting with this type of decisioning is computers are really good at making decisions that humans can make in less than a second.

Bill:

Mm-hmm (affirmative).

Vinnie:

So, I know what a stop sign is in less than a second. I know what a brake light is in less than a second. Once you start getting up into four or five seconds, computers aren't very good at that. And the example I give is, and a Tesla would fail this test. It's a two lane road, yellow line down the middle. And I pulled up behind a trash truck. I think I told the story on an earlier podcast. And actually it was a tree service truck and it was stopped. And I had to go out of my lane and around and a woman on the other side had stopped. And so, we didn't know who was going to go first. And the tree service guy was using hand signals that were not common to anything. One was a movie projector, moving his hand in a circle. And the other one was side to side. It made no sense.

So, both of us looked at him for direction. And then we looked at each other and we made that eye contact. This guy doesn't know what he's doing. He's not making any sense. It's up to us. And then we went through the hand gestures of who was going to go first and go around.

Bill:



Right.

Vinnie:

Tesla would fail that test. Now, what it could do is obviously talk to the cars and if they knew what was up, they could orchestrate some sort of moving around, but all that human interaction of facial recognition and hand gestures and stuff would be foreign.

Bill:

Right. Even intuition. You know intrinsically when a person... They keep looking over their shoulder and they keep looking over their shoulder, they're going to come into your lane, whether they signal or not. They're coming over. You'll see that and you can adjust accordingly because you have that, but the computer is going to wait and react to them actually coming over before that happens. You're right.

Vinnie:

Yeah. Another part of the automation is computer vision, IOT. So, I think of manufacturing a lot with IOT and having heat sensors, vibration sensors, motion sensors within large expensive pieces of equipment that can not just determine this situation is bad, but based on other things that have gone bad, we can predict something is going to happen. And then we can automate scheduling of a repair person to come out and do preventative maintenance as opposed to waiting for it to break and do repair work.

And the reason I'm bringing all this up is I want to make the point that there's direct structured automation of if then else, easy decisions. There's machine learning AI of predicting behavior and making recommendations, but there's also this superhuman measurement of information that we don't get that better informed decision making.

Bill:

So, let me ask you a question based upon on what you're saying there, because one of the things that we have not talked about is in order to facilitate the things you're discussing is a dumbification or a simplification of the components that have to be automated. If we go back to the manufacturing side and the cars, for example. I just bought a new car. Relatively advanced from a technology perspective, compared to any other car I've been involved with to the point where they had delays because of chips and so forth. But, when you get into it at the end of the day and you watch the manufacturing process, there's still people that have to get underneath there. Robots can do certain things, but they're still wiring, there's still junctions, there's still things that are obviously designed later after a problem showed up.

But, at what point, if you go back to the movie Minority Report when the Audis were being built on the line. Basically it's a block that's got plugins. It goes in, it's got another block with plugins, it goes in. And essentially all comes together to the point where, yeah, maybe you can't work on it because it's so high tech, but everything's a box and it's so simple. Then it's just plug, plug, plug. It's not a matter of me getting under the car, having to route wires, use zip ties and those types of things. And until you get to that simplification of all the components in a process, can you automate it fully?

Vinnie:

Yeah. Yeah. You can automate parts of it, which is happening and what I was thinking, as you were talking about that, that's what we did with software. We used to hand wire everything, we used to get up under there. And now we do design by contract, right? When I can write two different pieces of code



and they have connectors and they plug right in and they work. We're not hand wiring that stuff anymore. Now, we have to build the boxes that are going to connect to each other, of course, but any sufficiently, well architected solution will allow for the commoditization of the components.

Bill:

So, where does it stop then? Because if I can do all the components that we talk about and we talk about the plumbing and the technical aspect of things, the mechanical portions of whatever it is you're doing. What about that next layer? Because we discussed this before, as far as here's where the decisions start to come into play, and which ones are the ones you automate versus which ones are the ones you don't? And are there complexities involved with within... For example, if I'm going to buy a new company. All right. It's probably going to be a lot of work that has to be done that's not automatable in that, but once you get that company or you buy that company, then a lot of the things are automated as far as bringing them into your system, integrating them in and those types of things. Where do we stop with that hyperautomation?

Vinnie:

Yeah. I'm seeing it a little differently in that I don't think everything is... Automation doesn't replace all of these things. A lot of times it's making recommendations on where and how to spend your time. So, if we think about politics and different presidential campaigns that have been using big data and machine learning. It's not automating going door to door and having personal conversations, but it's making incredibly precise recommendations on which homes and which neighborhoods to go have those conversations. So, it's focusing the human work, not replacing it, right?

Bill:

That's fair.

Vinnie:

And that exists in a lot of other places, too. And this is where you don't want to rely on machine learning and artificial intelligence to make loan approvals or college application approvals or rejections, I should say, because it's a negative workflow. But, if you can use machine learning and AI to automate which need further review and which do not need further review or make predictions, then the humans can know which ones to apply their expertise to and process many, many, many more of them than they would otherwise because the signal to noise ratio has been improved.

Bill:

Okay. And that's fair. From a political standpoint, making sure you do the handshakes and especially for people that are on the bubble about things, obviously you've got people that are so far one way or so far the other way that you don't need to talk to them. And that's a separate conversation, but there's still situations where decisions are being made that are not as critical, but they're still being manually made. And how do we determine which ones of those are looked at, which ones are not? And I think one of the things I saw from the HBR review that I was looking at is it's very much like a grid like you would normally do. And what I was thinking was, let me check my notes here. It was essentially, they talked about decision impact as a unit of measure. So, basically you're saying how valuable is this particular decision to be made? And then based upon that value, where it falls within yes, let's automate it versus no, let's not automate it type thing.



Vinnie:

Yeah. So, that's the second thing I wanted to talk about post definition, which I think we beat up, which is how are companies using this? Are they being successful using it? How are they getting into using it? And I think you raise a good point, which is if you're going to have a culture that promotes automation, then what is the mechanism by which you identify candidates? How do you prioritize those candidates? How do you measure the accuracy and results of those candidates such that you can refine them over time or replace them over time?

So, what are you seeing in the field as it relates to an organization's appetite for really embracing automation and where they typically start and get traction?

Bill:

Most of the ones I have seen have are not even looking at it from this perspective. It's more of a, hey, there's a process that just needs to be automated. And what they don't necessarily look at is the outcome of that process. Does it warrant that automation? So, for example, if it's a high dollar process, it's very expensive to do and so forth, but the outcome of that decision or decision of that outcome is somewhat meaningless or not really worth a lot. A) do you need to have that process at all? But, do you automate it just to get rid of the high dollar process that goes along with it? And the very simplistic example is how much time do companies spend with people between 11:15 and 11:45, trying to figure out where to go lunch together? Well, back in the day when everybody worked in the same office... Is you spend maybe four or five people together, spend a half hour trying to figure out where to go to lunch. And the outcome of that is because completely useless, but you spent collectively a salary of that half hour.

That's a very simplistic and rudimentary example, but that's not even a level that most companies that I've seen are even thinking about. It just doesn't enter the equation.

Vinnie:

Yeah. And it reminds me of why we started talking about automation and hyperautomation back in October. And it really is the intersection, I believe, of this great resignation that we've gone through, now turning into the great regret, where people are leaving and they're leaving some jobs where there's some information archeology, right? The broken processes that you really have to dig through to find out how they work. And as opposed to just replacing people to continue on with those bad processes, they're automating them and they're modernizing them. And I think there's... For two reasons. One, like I said, people are leaving, but the second is I think the automation tools are getting so much better that the cost to automate, the cost to integrate is going way down. So, we don't have to say, okay, we can pay Jack to sit here for the next three years and just do this same thing. That's going to cost a whole lot less than trying to integrate it.

I think that math has changed. And I also think the pressure to be incredibly responsive to business demand and need has changed. So, the cost factor is different and the demand on being very flexible and rapid in response has changed and people are leaving those jobs. So, I think those are some big factors that are causing companies to really take automation more seriously.

Bill:

And maybe this is a good or a bad thing. I don't know. It depends on, I guess, perspective, but if you go through and you automate the process and the decision making to the level that you need to and people leave, they come and go, they come and go so that the personality of a company was maybe based upon people making those decisions at a point. They're approving some loans that normally maybe would not



necessarily follow the rules or doing those types of things. Once you get to the point where you standardize all this, does it take away personality of a company? Because a lot of these things are now being... No matter who happens to push the button, the answer's going to be the same.

Vinnie:

So, I hope not. And this is something that we'd spoke about, gosh, a couple years ago. Earlier with machine learning, our point was if everyone starts following the same machine learning models, is there a lack of diversity in how these decisions are made? Right? And that can be problematic. I don't think that's going to happen. I think what's happening is that these are tools that are tweaked and used in subtly different ways, nuanced ways, that for the most part are making recommendations on how to behave and what to do. And it still comes down to... People in the company are still the ones that are programming in the parameters and ultimately still making final decisions.

So, I guess I would go back to the difference between automating and scripting, something that is deterministic, which I don't think has any personality to it from a culture perspective, versus automating something that is basically using machine learning to categorize or make recommendations or inferences on large sets of data that a person then would have to respond to. And I think because of that second part, you keep that personality in the process.

Bill:

That's what you need to do anyway if you're doing AI for all these things is constantly feeding, was this a good decision or bad decision back into the model. So, that becomes part of the responsibility you have is did we do the right thing at this point?

Vinnie:

Right. And what is a good or bad decision, right? That depends on the perspective of the company evaluating that. And I think that puts that personality back in. So, it's a tool and it's how you use it that defines it. But, yeah, I do worry about a commoditization of some of the technology that would basically create... If there's bias and there will be bias to the extent that we've commoditized these, then we've commoditized the bias as well as opposed to having a lot of variability.

Bill:

Right. And I guess the thing with the bias aspect of it is sometimes you need it in order to be able to make exceptions.

Vinnie:

Explain that a little bit.

Bill:

If you say, for example, I'm looking at ACT scores or SAT scores, and I'm making my decisions based upon this and maybe a couple other things too, but there's other things that go into a college application. Like you said earlier, why you'd want to have a manual process for it is sometimes that bias towards somebody that's underprivileged or underserved, you have to be able to say, all right, I'm going to bend the rules in this particular case in order to get this person in or to allow this particular situation to happen. So, from that perspective, bias is required.



Vinnie:

Right. But, that can be part of your model, right? And I think there's some interesting things. One of the great things about machine learning is you don't have to be declarative in what fields to look at. You can say here's 3,000 applications of good students and 3,000 applications of bad students. Find things we wouldn't normally see. Right? And by the way, don't look at these things because they're demographic information and what it can come up with are patterns that are beyond what a human would've found.

Bill:

Agreed.

Vinnie:

Yeah. So, I know we're pushing on time. So, I want to talk about building blocks now and...

Bill:

Sure.

Vinnie:

How companies can get started. What are some good recommendations for getting their feet wet? You started talking a bit about that about first identifying, basically putting up some guardrails as to what can and can't, or should or shouldn't be, or high value, low value. Basically prioritizing opportunities. But, what would your recommendations be for people listening to this podcast, wanting to do more with this type of technology?

Bill:

It's funny because I was thinking about this as we were researching this. It's almost to the point though where some of these things that we would normally talk to a client about is, all right, look at your code. How can you do these to decompose this and where your decisions are being made and how do you make it less complex and more simple and those types of things. But you almost have to start from the top. What are the decisions that are being made? And look at it from that perspective. And I think we were talking the other day about a decision driven design type thing.

And it happens again, like I said, two ways. From the top down, as far as here are the decisions we're making and what are the components of those decisions? And from the bottom up, what are the components of all the things that we're doing and which ones factor into which decisions? And you figure out which decisions are valuable, which ones are not, what's the cost of making those decisions, what's the cost of being them wrong and those types of things. And you figure out where, from that perspective, where you want to apply your time and your effort, but that's a little bit turning things on your head from most assessments that we do now. It's still the same methods as far as current state to be those types of things in your roadmap to get you there. But you're looking at it from a different set of metrics. It's the value of a decision as opposed to the cost of a process.

Vinnie:

Yeah. As you were talking, I was thinking of a partial or a level zero or high-level canonical model. What do we do? How do we do it? What are the major attributes of our audience that we're addressing? And look at those flows and those workflows at a canonical level before you start decomposing it, right?



Bill:

Yeah. So, the people that you have and the amount of effort. I worked for a company once that they basically said, if I want to make 10% more money, I hire 10% more people. Because everything that they had was completely manual. They had runners to go down and get policies, for example, from the archives whenever somebody called in. So I say, hey, my name's Vinnie. I'm calling about my homeowner's policy. And they say, well, Vinnie, I will call you back in a half hour. And they would put something in the outbox. A runner would go to grab it and bring it back. And that was 2002. Those are the types of things you would look at first. Where's the biggest pain? And that's what we typically have in every single client we have is where's the biggest pain?

Vinnie:

Right. So, I think it does start with the business domain and the business problems or the business aspirations. I think the next level down is decomposition of those. Where do they live? Who owns them? And I think of things like data. If you're going to be doing data driven decision making, whether it be machine learning or otherwise, you better have good data, it better be clean, you better have enough of it, enough variety of it, etc. So, it becomes a real data engineering, data science process also, which I think is the first mile for some of that stuff, as opposed to the last mile.

Bill:

And then I wonder if one of the new things to think about, or maybe it's not a new thing, is it to require information that I have readily at hand or do I have to get IT involved? So, we didn't talk about low code and no code solutions after Microsoft Power Tools and those types of things. But is that a situation where can I solve it in and amongst myself? And that brings in different problems. Now, am I solving problems that somebody else is solving somewhere else and now we're committing different solutions for them? But that would be one of the things I would look at too, is do I have to go through the whole process of getting IT involved in the evaluations and assessments and so forth? Or is it something that I can do with my Excel and my access to the data that I have and the tools that I have at my disposal? So, it's a whole new level of shadow IT potentially.

Vinnie:

I could see it rolling up to a single person too, that has some ownership over this across the organization.

Bill:

Yeah.

Vinnie:

And that person does have to have a unicorn set of experiences of understanding the business and the business intent, but also understanding architecture because one of the potential downfalls of automation is getting into technical... Unintentionally wrapping yourself around technical debt.

Bill:

Right.

Vinnie:



And then getting yourself into bigger problems.

Bill:

Yeah.

Vinnie:

Great. Well Bill, thank you so much for joining me again.

Bill:

Thanks, Vinnie. A pleasure, as always.

Vinnie:

Hopefully I'll have you on again soon and everyone, thanks for listening and stay tuned for more.

The entire contents in designing this podcast are the property of CapTech or used by CapTech with permission and are protected under U.S. and International copyright and trademark laws. Users of this podcast may save and use information contained in it only for personal or other non-commercial educational purposes. No other uses of this podcast may be made without CapTech's prior written permission. CapTech makes no warranty, guarantee, or representation as to the accuracy or sufficiency of the information featured in this podcast. The information opinions and recommendations presented in it are for general information only. And any reliance on the information provided in it is done at your own risk. CapTech. makes no warranty that this podcast or the server that makes it available is free of viruses, worms, or other elements or codes that manifest contaminating or destructive properties. CapTech expressly disclaims any and all liability or responsibility for any direct, indirect, incidental, or any other damages arising out of any use of, or reference to, reliance on, or inability to use this podcast or the information presented in it.