# EPISODE 1398

[INTRODUCTION]

**[0:00:00.3] JM:** Robotic Process Automation, or RPA refers to software robots constructed to automate some business process. Perhaps, the most ubiquitous example is adding filters to your email inbox. I've worked with a lot of salespeople that use tools to configure complex email follow-up campaigns when inbound emails come in. Even that is a fairly basic example, compared to what's becoming possible.

UiPath is an automation platform. They offer a suite of solutions that empower developers and non-developers to construct effective software robots that can measurably improve business efficiency. In this interview, I speak with Boris Krumrey, Global VP of Automation and Innovations at UiPath. We overview the platform and discuss the current and future state of RPA.

[INTERVIEW]

**[00:00:49] KP:** Boris, welcome to Software Engineering Daily.

**[00:00:53] BK:** Well, great to have me here. Thank you.

**[00:00:57] KP:** To kick things off, can you tell listeners what your role is at UiPath?

**[00:01:01] BK:** I look after automation innovations, and I run the, we call the UiPath Immersion Labs. The UiPath Immersion Labs was an invention, actually from my end, because I felt that automation is something abstract and doesn't really make it easy for people to understand what it means and how it changes your daily operations and how to apply it and so forth. We basically, created showcases that bring that in the whole context of business processes, automate processes, and turn it into a real experience.

We have a hands-on lab, so people could log in on a virtual desktop, basically, experience automation and what it means, and so forth. That's in a moment what I'm doing. When I joined

UiPath, I was actually a customer before, and we have run large automation programs. Through these topic around automation and robotics, I discovered UiPath. It was one of the first softwares that was very easy to use, and very powerful, because it was very accurate in the using User Interface automation. That's where I started.

I came from the consulting and outsourcing space from Accenture, Infosys and Atos for many years. Originally, I'm a computer scientist, so I programmed in my past. Anything that is in a turning a business problems, or into software in a very easy way has always fascinated me.

**[00:02:37] KP:** It seems to me that every listener has to be thinking of at least one time in their life when they had to do some very manual, tedious thing on a computer. That could have been a good use case for this. Maybe they messed around with Excel macros, or I had a browser plug in for a while and did some fancy things. How does one engage with UiPath? Where does it stick itself into the ways I might want to automate things?

**[00:03:01] BK:** That's a very good question. That's the key point where it all starts. Imagine if you want to onboard a new employee in your business, or in your company, there's so many things that needs to happen, right? A candidate would fill out whole form with all those details, you set up the payroll system, you will create a batch, you order a laptop, you order a mobile phone, and the new manager has to approve these.

There's so many steps, and there's so many different systems involved. Imagine now with one workflow, you can basically, tether these all together and turn it into an application. One application, one UI, where you enter all the data, and it will then do the entire workflow, sending the email, getting the approval, understanding natural language, even extracting if necessary, if there's some particular forms that are necessary, I don't know, for tax reasons, or whatever.

Extracting forms, of data from forms as well, so doing like an OCR. Putting that data into the backend system, wherever HR system it may be. Then going into whatever the internal IT purchasing system is to order a laptop, and so forth. So many different interfaces, from APIs to UI interfaces and so forth. UiPath can combine this whole, and combine it all in a very easy way that you just record yourself doing that action. There you go. There's your code that you can then, like a workflow format that you can just edit and adjust and work with.

**[00:04:41] KP:** Even for the seemingly simple task of onboarding a new employee, I think I counted five or six independent systems in your description. I can't imagine, many, if any of them have nice, RESTful APIs and that thing. Does UiPath have to go out and engage with all those systems, or am I bound by a fixed list of a catalog of things? What's the opportunity for people who want to build stuff?

**[00:05:05] BK:** This is the great thing about UiPath as a toolkit, is you can build an API on anything. Just on the user interface, for example. If it's a mainframe application that doesn't even have an API at all. I can just record myself extracting the particular information from the screen, and then take that information and provide it to an API, where you have maybe already made RESTful API that I can use.

That makes it really, really flexible of connecting to anything. Plus, we also extended our platform with a whole API management capability. We acquired a company called Cloud Elements. That was exactly what they provided, similar to – probably, people know, Zapier or so forth. You just have a ready-made API. You just drag and drop it into your workflow. Then there you go. You can right away, connect to it, you can connect to all the objects. You can even define triggers and say, “Okay, if this SAP system table here is being updated, then I want you to update another table in another system.” You can just connect those systems in this way.

Or, if there's not an API at all, and it's some kind of a mainframe or whatever, and you have to go to a different user interface, then you can combine here API with a UI automation in any form of workflow.

**[00:06:42] KP:** I'm wondering if we can stick with that mainframe use case. I'm picturing some system, probably a legacy system, the software was all written custom by someone who hasn't been at the company for 15 years. It's in a language nobody knows. We don't want to touch it. Do I install a client on the mainframe, that then controls, or what's the actual technical integration point?

**[00:07:03] BK:** Oh, yeah. You don't need a client at all. See, you may not have the people who have programmed it to operate, but you still would have people who know how to enter data and get data out of the system, right?

**[00:07:16] KP:** Yeah, for sure. They rely on it, usually.

**[00:07:18] BK:** Yeah. All you need to do is actually, you're not even talking to any technical person. You're talking to the user, saying, "Okay, how do you get this certain information out of this legacy system." I type in this and I type in that, and so forth. They would just perform this activity. Once they're performing it, we are generating the automated workflow. Then, we see it as in the WF format, and can just then modify it, adjust it, make it flexible, put parameters, variables, and whatever around it as we want, and then turn it basically, into a full application.

**[00:07:57] KP:** That application then, does that run onto data entry person's machine? Or how does that go?

**[00:08:02] BK:** Yeah. You could either have it on the data entry machine, as what we call attended automation. The robot would be working right side by side with me. Or, you can have a separate VM, virtual machine with a robot configured in its own separate user ID, and it would just perform this in that environment. That's what we then call unattended.

**[00:08:25] KP:** There's certain tasks that just seem like, obvious wins with a tool like this, where maybe I go to a website, I get some data, I need to enter it in some form somewhere. A lot of heavy work to do. I don't want to undersell what the software accomplishment is, but I believe that's implementable. Then you get to cases where there are more and more human judgment calls. Maybe a text paragraph has to be read and interpreted. How far can you get with natural language processing and computer vision and things like that?

**[00:08:53] BK:** We have embedded in our platform, computer vision. We have a whole AI engine, which means you can take any machine learning model that you've written in Python code, for example. So long as it's serialized, you can just upload it into our engine, and then you can execute it within the workflow. That's immensely powerful. You could use it for fraud detection. You could use it for any prediction you want.

We have a whole set of ready-made solutions that you can also use for whether it's particular models for predictions of defaulting loans, or for example, we implemented solutions in our lab for customers, where we did the prototype of for the cash collection process. Basically, able to predict how many customers and which customers will probably not pay in time, their bills and so forth.

That mechanism then helps to trigger certain actions to remind people and so forth. It can help to control your cash flow of a company. That's one we have also ready-made models for document understanding. Any forms, receipts, invoices, you can just send to our robots these invoices, and we would be able to extract the information. We have them this whole mechanism within, what we call human in the loop. Whatever the system is able to recognize, if there's anything that the system is not very confident to have recognized, it just then puts it into the queue of the human to revise it.

When they basically revise it, or make any correction, we would then track that information, and then retrain the model automatically. Our engine also does auto ML on automatic machine learning, basically. That's all part of our platform.

**[00:10:49] KP:** Do you see any rules of thumb for – I'm thinking, any organization of a sufficient size, they've got to have some, if they're not already doing RPA, there's got to be a good opportunity for them to find improvements. Do you have any rule of thumb for how big a company is before they need to explore and where to find the low-hanging fruit?

**[00:11:09] BK:** If you think of a small business, or a big business, they all have to use certain same key functions. Yes, to a different scale. Still, automation, there's always swivel-chair type of activities everywhere. Whether I have to process thousands of invoices, or whether I have to process in a very small company, hundreds of invoices every month, it's still a lot of work. If I can automate it and save that time, it is as impactful for small businesses, as for large businesses.

**[00:11:42] KP:** Is RPA purely a cost-saving effort, or are there side benefits as well?

**[00:11:48] BK:** Well, that's an interesting one, because I think in the very early beginning, when people first looked at RPA, they were all triggered by the vision of, “Oh, this must be cost-saving activity. Great. I can save, and many FTEs, and whatsoever.” It turns out, it's not so much just cost-saving, it's just increasing productivity. It's really helping to – that you can actually do more. It gives you more capacity to do work that humans should be doing, and not the work that robots should be doing. Whether it's taking data from your emails into some backend system, or whatever you may be doing.

That's the key proof point, that productivity, capacity increase, that you reduce the amount of hours that we would be doing as humans. As we are humans, we would forget some data, or mistype something, and then correcting that would take even more time. All those things, you can really eliminate.

**[00:12:53] KP:** Do I need a background in software engineering in order to utilize UiPath?

**[00:12:58] BK:** I think, it would be ideal if you implement large projects, because UiPath is now grown to the capability of being a fully application development platform. You can combine all these automation flows into a real application that you can just trigger within. It's just a URL, and then, there you go. You have a user interface on top of all these automations. When you do larger projects, then yeah, it makes a lot of sense to have that.

We are also very much targeting the non-technical user, the users who are operating a process, an accountant, or a sales operator, a customer service agent, and so forth. If they have the ability to just say, play around with some – creating some process or so forth, or they have a bit of an affinity for being interested in building something logically, then they are very much able to create in whole automation themselves. We have a special product for that. We call it Studio X. It makes it really easy.

It's also designed for the purpose, what a typical user would need. A typical user needs to put – extract data from place A, and put that into the system B and then extract maybe another data from system C and combine that, do something with that data. Then, put that into a system D, or something like that. Ideally, for all those things, so anything you do around Microsoft productivity tools, or Google productivity tools, you can then basically, combine and turn into a fully-fledged automation.

**[00:14:51] KP:** Well, the bigger the automation gets, the nervous I become that something could go wrong under the hood. Like onboarding a new employee. What if the laptop that we typically order is out of stock? I don't want to just set it and forget it. Can you talk a little bit through exception handling and how someone can monitor and resolve issues when they arise?

**[00:15:12] BK:** That's a very good point. First of all, as part of our platform, we also offer process mining. A tool that links into all these systems and gives you a log of what people are doing. What that helps with is visualizing an entire flow, shows you how many deviations of this end-to-end process do really exist? With that, you could identify whether a process is highly standardized, or not. If it's not, then you know that your automation journey is going to take much longer. That's the first thing. Tools that help you to discover and also, to assess how you're doing certain things.

Secondly, to pick up exceptions. The robot is connected to a management console. That management console is called the orchestrator. In there, you can see any errors and so forth. You have a very good interface that would alert you if anything goes wrong. If you come across this exception, and then you could pick that up fairly quickly. Now, in order to keep – I'm going to say, to make the maintenance challenge that you would have as you're turning manual process in software, into software; to keep that very low, we actually also entered into the test automation space. You can use UiPath to test any kind and automate the tests for any application; web applications, mobile applications, whatever your software development target is.

You can also do it for RPA. In this sense, we are applying the best method that you could do in continuous integration. Ensuring that the same skill that you have for automating processes, you can also use the same skill in applying test automation to it, so you're reducing the amount of maintenance challenge that you would have with the software you're creating, with the RPA software you're creating. That's another very important aspect. That's one reason why we've also went into test automation.

Now, typically, when you're deploying automations, the way I would always suggest to operate is say, it's a business process. That business owner that knows the process in and out, you're basically creating that automation. You're giving it to that business owner. The business person, or subject matter expert can then trigger this process, and can really make sure that all the business logic are a 100%. Because it's attended, it can really follow what the process is doing, or whether there's any issue around it. Before then, you put it maybe into the larger production environment, and even decide maybe to put it as an unattended robot workflow into the backend, into a virtual machine.

There's also a method around how you deploy automation in a most controlled and effect foray. Probably, when I explained that, you can already sense that software engineering, all the best practices of continuous integration, if you do automation at scale, do apply here as well.

**[00:18:34] KP:** Could you compare and contrast that further? What are some of the distinctions between the standard vanilla software release lifecycle we're familiar with, and the release lifecycle of an automation application?

**[00:18:47] BK:** I think, if you look in many areas, so companies and advisors, or consulting companies looking into RPA deployments, they all look at it like, software development for web environments. Very agile. In sprint and so forth. That works. That's fine. I think, that the approach should rather be combined with something that we used to do in the outsourcing space, where we actually look at the full scope of the activities that people are doing, that you basically, you do a knowledge transfer about.

That means that you are actually looking at an entire stack of activities you're going to automate. Your knowledge transfer approach is a bit different. Meaning, you are first, when you create the – document the process, which you can do here with pretty effect fully, because we have a tool called Task Capture. Even if you're not a programmer, I can just perform the process. Besides of just creating the skeleton code, it also generates the entire process definition document in Word, in the template I want.

I already got a piece of documentation that then can be taken to the developer, RPA developer, who then can read the process immediately, with all the screenshots, and so forth. Based on that, you're basically creating then, the automation. Of course, you're receiving also the skeleton code of that recording. That cross-tracks this implementation for customers significantly, about to 60%.

When you take this, and this is where I'm coming close to the outsourcing analogy, because when you do knowledge transfer, you do the same thing. You capture, basically, a process. When you capture that process and you have it documented, you then start, basically, taking that as your theory to learn. In the same sense, you're now taking that document as a theory of the process and then turning it into software. You then, basically, giving that software to the person that actually be using, or has done it before.

With that phase that you use, also, with a knowledge transfer and outsourcing is your – they're probing already how reliable is this whole process and works in the business context, before you actually then scale up the transaction of exactly that automation. It's a bit of a different approach, but it's a very effective approach if you want to run a very large transformation, or turn a lot of manual processes into software. That's a bit new and different than the traditional software development approach, where you either follow an agile principle, or waterfall principle, or whatever you want to follow here. That's what I'm seeing when you're looking at a large-scale implementation.

**[00:21:52] KP:** From what I've learned about UiPath, it seems to me, a professional could pick this up, teach it to themselves and start using it inside their organization. They don't necessarily need your help. Maybe at the other extreme, you offer full service, or some hybrid model. What is a typical engagement like?

**[00:22:09] BK:** What's interesting, it started that people who are not technical people, just in the business itself; downloaded the software and started learning it. Then pretty quickly, we're able to create their first automations. Then, they show that their management, they said, "Wow, this is easy. This is great." This is the beauty of it, and it's also one of the reasons why it grew so fast. That you could pick that up relatively quickly.

I would say, for technical people, it's even much quicker to pick up, but to do much bigger and complex processes. Most like, the low code of the future, of turning business processes into very quickly into software. That's where it's very strong about.

**[00:22:55] KP:** When I think about some of the historical wins from RPA, it's event-driven reactive kinds of things, like email comes into the recruiting inbox. We pull the resume out, put it in the recruiting portal. These kinds of steps that yes, it's good to automate them, but there isn't a whole lot of decision-making going on. Although, I feel that's one of the major features UiPath has been introducing, that these can be more intelligent systems. Can you talk a little bit about that transition from these rote RPA processes to more AI-driven decision theoretic systems?

**[00:23:31] BK:** Yeah, sure. Because for us, automation is intelligent. We don't even differentiate anymore and saying, speaking about intelligent automation and just automation. Because it's such an integral fabric and such an integral part to be able to apply decision models into the entire workflow. For us, it's an integral part. The nice thing is, when you think about from a development perspective, the guys who are developing machine learning models, a bit of a different type of people, than the ones who are programming – programs an RPA.

I think, when I spoke to some of our technology partners who were very early on in the AI space, they said, “Oh, yeah. My data science guys always looked down upon the programmers, and the programmers looked down upon data scientist guys.” It's somehow, it's just like, it's two different teams. The nice thing about our platform is the data scientists, they can really evolve and create their models and upload those and build any custom-built model for the particular purpose, and train it and have auto ML and everything integrated.

Then on the other side, the RPA developers, they can just use that decision model as a blackbox, input, output and just combine that with any workflow and combine it with also, with the entire human supervision and the auto ML functionality. In that regard, it's pretty neat. What we see in business very much as when it comes to AI and machine learning, most of the next level uptake from basic automations, or rule-based automations, is actually all-around extracting information from documents, and doing that in a very reliable way.

Basically, all the cognitive part, where information is provided for other humans, are meant for other humans to process. I think, that part we have fully integrated. We can process voice. We can have full natural language conversations, link it to our robot automations. All that is very easily possible and integratable with our platform.

**[00:25:49] KP:** What's the current state of the virtual assistants' involvement in the process?

**[00:25:55] BK:** Yeah, the virtual assistants, you basically hook that to our system. For the actual fulfillment of the transaction, say you want to change your address, you want to find out what the tracking of your packages you ship. The conversational part is basically handled by the conversational engines. The fulfillment, request fulfillment part is done then through RPA. RPA also helps to create new conversational knowledge, or even trees, or semantical changes into the conversational AI systems. That's also something. It's not just fulfilling, but it's also helping to add, improve the program and make the system smarter.

We can provide both ways. We have integrated many of the big existing ones from Google, from Microsoft, from AWS, and so forth. There's also many other providers that offer this full capability and with the integration of speech-to-text, so that you can have almost a natural conversation in any language, by the way, with the robot. It's because, it's so smoothly integrated.

**[00:27:11] KP:** Well, if we take that to its logical extreme, it's like interacting with the computer on Star Trek The Next Generation, where you ask it to do anything, and it's the operating system. Do you see that as an ultimate direction that UiPath will take things?

**[00:27:26] BK:** It's direction that's not just happening through UiPath. Generally, there's a tendency that it will go this way. Yeah, we can already realize it today. It's quite astonishing that nowadays, where we have all these technologies, we still have big call centers, where you're calling and you're waiting in a line for, I don't know, half an hour just to get someone to talk to. So much could be already automated today, just with what we have today. UiPath definitely could enable that fairly quickly within a reasonable time to configure.

**[00:28:04] KP:** I can see a lot of large enterprise. We've talked about medium-small business opportunities. Is there an angle for the indie developer to be leveraging UiPath?

**[00:28:13] BK:** Actually, I think, for developers, now an ideal time. Say, if I'm a developer, and I'm always interested to work very close to the business needs, or business process, and I want to solve a business problem, the good thing about the RPA and about UiPath is that you can turn it around much faster than before. You can work very closely with the subject matter expert from the business than before. Because now, we have tools where the subject matter expert can just create a simple recording, generates an entire document, generates the skeleton code, whatever it needs to be done. Immediately, you can turn that into a practical application that then, this same person, subject matter expert, can then test and verify and use right away.

I mean, one of the beauties that RPA was giving is that it really made it much quicker of implementing a complex process and bit by bit immediately, and giving that back to the users, where immediately saw the benefit. As you went ahead, you could solve entire complex processes. Because we have automated bits of this entire big process, you have basically, immediately turned around and improved an existing process and made a digital, where it wasn't before.

If when you consider the current need and the moment to put everything into digital, to turn business processes into a digital operation, and create new customer experiences, and so forth, then it's really essential to have a tool that helps you to do that very quickly. It's like, you have a new tool that just with your programmer skills, just gets you much faster and turning a manual business process into software.

I think, it's actually a special tool that many programmers, if they would pick that up, they will immediately get the attention of companies and your – we have always seen that people have automation RPA skills, they immediately get picked predominantly for any job interviews, and so forth. Because that's a real need in the business today. Today, you could say, a CIO has to become a proxy COO. That comes from Gartner.

That just shows that if we are staying just tech people, and you're just staying in our tech world, you’re not moving along with the current demand. The current demand is that everything needs to be more digitized. We work from home. We operate online, and so forth. Everything we want to be doing through online services, and so forth. In order to comply to that, we need better tools to digitize. UiPath is exactly that tool that allows that. For programmer, it's like, wow, that's the new case tool of software development to turn those business processes into a digital operation, and connect with all other systems, because there's so much that we offer around this whole software stack. In that regard, yeah, it's a whole new powerful business development platform for digital transformations.

**[00:31:47] KP:** Are there any interesting use cases you're able to talk through from customers?

**[00:31:51] BK:** There are quite a number of them. Well, I gave you examples for HR onboarding. Just think about the recruiting process, how you could optimize the recruiting process, and combining it, also with machine learning. It would be able to scan millions of candidates with opportunities. In this sense, recruiting companies could far quicker find those fitting opportunities and find the right matches. The other thing is, they can then, and this is where the effect what I mean, with productivity increase comes. If you think about, a recruiter would have to update so many different recruiting websites. Not that the own system, but also, all the other systems that people would be using, looking in social media and so forth. Syncing up that data with all these other systems, you need a robot for doing that, right?

That's where we have use cases, where people, or customers that are specialized in this business are using it, are using AI for the scanning of resumes, finding the best fit and so forth. You can even go further, because he could make it for the individual recruiter, that with this automation running from its own social network, tracking its own social network of candidates against the positions and forth, that’s immensely powerful. That's an example you could apply to. There's many examples when it comes to supply chain, when it comes to call center operations.

Basically, creating a robotic call center agent that I can talk to, when I do all these standard transactions, maybe wanting to change an address, or track a delivery, or I want to have a guided path of submitting a claim, and all those kinds of insurances, all those kinds of activities you can – our customers, basically can take and automate.

**[00:33:48] KP:** Well, we've covered a little bit about how you've leveraged natural language processing and computer vision in certain cases. I'm curious, if you have any thoughts on the most impactful technologies, or methods that have come out of the ML space that you've been able to leverage, or maybe research you're closely following that you're excited, maybe leverageable in the near future?

**[00:34:07] BK:** Well, I think in the near future, so the things that are really exciting is when you look at from the narrow AI automation space that we are in now, that we can apply in many ways, going more into the semantic and automation space, where we combine a particular semantic knowledge of having a particular ontology, or so forth. Then, using that as a basis, as a reference model to extract and understand the world.

To give you an example, think about, you have an SAP system. SAP is pretty straightforward, because it has dedicated transactions for any business process, and it's very well documented. In that case, you would know when you enter a particular transaction SAP, what are the data fields and so forth. Now, imagine you would have fed all this information into a knowledge tree that is able to know, if you want to do a purchase order, or set up a purchase order request, the robot would already know what data you need to feed into the SAP system, and would already retrieve and know where to take certain data already from some other, maybe master data system, and combines it. Then just ask you that data points that are still needed, that it can't retrieve from anywhere else.

Does everything else itself, because it understands what it needs to be doing. It’s almost like, we do with conversational AI, where you understand the intent, but understanding and translating the intent already to the underlying application. That's where it becomes very powerful. That's where we see semantic automation.

That also goes to the next level, which is generative AI. I call it generative, because that's where you start doing something that you're basically creating. The example is with AlphaGo, where you basically, you find new ways to compete in the game of Go, or you go into the area of Alpha Fold, where they predict help, certain proteins will fold and so forth, which then is impactful to tell you whether certain Pharmaco will actually be – how they would react and so forth. You don't even need to experiment anymore. You can really use AI.

That's the part where say, it's generative AI, or producing narrative content. GPT3 gears a little bit into that area, and so forth. I'm seeing that is a semantic AI, or semantic automation is going into that direction, because that's the precursor for the generative bot. That's something we are very interested in, particularly in the semantic automation part, as we see that as a realistic next step.

Then further along, when I say generative AI, further along, yes, there will be at some stage, maybe robots that do self-automations, or things that self-program to a certain degree. Again, we need to be careful, to what extent do we allow an autonomy of these things, and there's always a human in the loop to validate and control. It could help for solving very big, complex problems and turning them into software. Semantic automation is basically, our big focus in the future. It more relates to the example that I was getting.

**[00:37:41] KP:** It's going to be exciting to follow. Boris, for listeners who want to plunge right in, where's the best place to see if they can get their hands dirty with UiPath?

**[00:37:49] BK:** We have a free version on the automation cloud. You can sign up and then join. This automation cloud is Community Edition; has all the full functionality. All our trainings are free. You can sign up for the UiPath Academy and get started right away. I mean, that's the nice thing about it. From whatever background, if you like tinkering around with those solutions, and you have an affiliation for technology and trying out things, and so forth, this is your time. It can really just make use of. It's all there. Yeah, even for small businesses, they could start. That's a very easy way to get started.

**[00:38:34] KP:** Well, very exciting time indeed. Boris, thank you so much for coming on Software Engineering Daily.

**[00:38:39] BK:** Thank you for having me.

[END]