Leandro Melendez: Automating the automatic automation

[00:00:01] Hello and welcome to this Test Automation Guild presentation that we are so happy to start and present for you on this new 2021. Let me introduce today we are going to talk about automatic automations, where we automize automatically and everything that goes around how to streamline pipeline or whichever you want to call it, ways to automate whatever is already automated. We're going to go into some details. But first, before I begin with this presentation, I would like very much to thank my amigo Joe Colantonio. First of all, inviting me here, allowing me to be sharing the knowledge here with all of you and giving some of these key elements on how to get better automations and to integrate them within your pipelines. And second, I want to thank the amigo Joe for putting all of these together, enabling these set of sessions that help us, the ones who want to share the knowledge, share it but we have going around too much. Thank you, amigo Joe. Thank you for inviting me, giving me the opportunity to be here and spoiling your people and let's get into it. But again, before that, I for the ones that do not know me, I want to do a small quick introduction for again, if you haven't heard of me, my name is Leandro Melendez. I am a performance manager in Qualitest group where we provide multiple services around testing everything testing again, not only performance. And what you are going to learn today is not only about performance, anything, but there are as I said, I'm a manager. I have been working around IT for over three years now and as well in the testing and performance realm for over 10 years now as well. So there are some experience here that I would like to share and I actually do like to share it. That's why I have these persona in the Internets, also known as Senor Performo, where I started off as a blog where you can read some things around performance testing best practices, how things work. I have been also very active on social networks as Senor Performo as well, where you can find Twitter, Facebook, Instagram, LinkedIn and lately even YouTube, where you can look for Senor Performo content in English, content in Spanish around all testing topics, but especially performance testing. And as well, I'm the host part of the PerfBytes family, hosts of the PerfBytes Spanish version PerfBytes in Espanol, the podcast where I try to share most of the knowledge in all Latin America and Spanish speaking amigos, where some of the content that we have in PerfBytes in English is not very well understandable we try to share it. And well, I join always and often my friends James, Mark and Brian not anymore on their shows from the PerfBytes family. So you'll hear around about me. I'm not only here as you're seeing me right now on this presentation, you can see us well, some of the other, as I mentioned earlier, the YouTube channels where I started to pioneer or try to get my feet deep in the YouTube waters. Let's see. I hope you like it. If you haven't heard of it, go there, subscribe, hit the like, and all that. And last but not least, I have done public speaking. I have been doing it for about three years now. And some of the things that I talk about are similar to what you are about to learn here.

[00:03:54] So try not to go longer here and get ramping up, let's jump into the topic, which is automations, automating them, making that automatic and everything out of something. But before we dive into that and of course, we're going to be talking mostly about test automations, but we need to define what is an automation. And mostly well, I went, as you could see, to Technopedia and grabbed the official definition of what is an automation. It's the creation and application of technologies to produce and deliver good services with minimal human intervention. This is one that I want for you to land on your selves, minimal

human intervention that's super important. And automation technologies help to process and improve the efficiency, reliability, speed of the tasks that again were previously done by humans. But where machines are better to do definitely than us, that's where we want to leverage them. And don't be afraid, machines are not here to take over us. They are here to help us. So and that's very much. Machines do or automate processes that we generally used to do ourselves.

[00:05:06] Okay, but why do we need the automation? I mean, it's a bit self evident, but as I mentioned, when these machines do the things that we need or that we want them to do automatically automate it they generally do it faster. Since as well they are machines they do things like a machine repeatable in the same way. They follow a process specifically. They don't need to start to breath and think and to scratch your head while you're saying like, "Am I doing well, this presentation?" They just do it. They follow the instructions that you gave them at the rule of the step specifically. Well, another advantage of automating something is that it gets cheaper. It gets because at the beginning, generally, it's a big investment. You need to buy something to produce a machine, to program some process. And after a few runs, in the long run, it's cheaper. It gets to be like, "Hey, I bought these machines that now makes everything easier. I did a huge expense at the beginning, but now I'm getting all the results." And as well automation, while you have a process automated, it's supposed to be easier, supposedly if you did it well. If you didn't, it gets more complicated. We will get into that. But most important for us human beings, people and engineers that want to do other things, it frees up time for us. So many benefits. Some examples of these and many think of it like Richie Rich in the future that had so many cool modern machines that could do everything for us, but more or less in theory, that's what we are looking for with these automations, but not as sci fi or futuristic, which many of those things already exist as Richie Rich.

[00:06:52] Let's go over some regular examples of how automations in our daily lives. As first few examples would be like a blender, where, believe it or not, a blender is an automation machine that helps us in doing some manual processes that we used to do before. Literally manually we used to power and mortar our fruits with milk and tried to do some sort of smoothie and now the blender can do it easily. Same with a coffee machine, you would say, "Well, I still have to work a lot on that." But originally without a coffee machine, you would have to grind your beans, not cook them, though most machines already get them cooked. We will get into that as well with a pre steps. That's important. But you just put beans, it grinds them, it puts water on them, filter the water, puts everything even a little bit of air to create that delicious foam. And you get an automatic coffee. Same with a shower. You would say, "What? A shower?". Yeah, the shower is automating many processes that we used to do before manually. Heat up the water, transport it, have it up so that you can get the water flowing through everywhere and draining. The drain part of the shower it's an automated process that takes away the dirty water, all of that. Even a light bulb. That's an automation. We don't have to build candles, put the thread on top of it, get some light, be there and again generating it, automatic process that brings us light. All of those are mundane, I would say examples of automation, but very good examples and automations in the end.

[00:08:25] So test automations, there are many activities that in testing we get benefits of automating those processes. We can do multiple automatic things while we are testing

things like clicks, scrolls, typing, sending messages, doing a verification, doing a validation, even visual automations. We can nowadays program things that visually will say, "Hey, that's a box. That's a button. No, they are different. That's a checkbox. That's a checkmark. That's a radio button." And they can tell what are the differences.

[00:08:57] So nowadays, test automations can do many, many things that can aid us in our testing efforts. But what are the goals? What are we trying to achieve through test automations? As I said, there are many benefits similar to just generally the automation automating a process. But here some of those benefits are speed. An automated test process as well can go faster than a person clicking through it, thinking about it. If we just create the steps, it's going to go machine or the machine like possible speed that a machine can do, usually way, way faster than a human being. It's repeatable as I mentioned earlier. It's exactly the same set of steps and we can use it over and over and over again as long as the circumstances are still the same. Again, less cost. We will have savings while executing those automatic testing processes. But it's going to be as well high cost at the beginning and eventually will ease off. And it will be just like, "I don't pay for this anymore and I'm getting automatic tests. We can also increase coverage because the these automations allow us to do many, many more testing tasks that we used to do before and many other things. But test automations are here to stay. We have had them for, I don't know, 20, 30 years for sure, even more. But they're here to stay and we need to step them up. Why? There are points where we automate the tests, as I said, where we have these boring processes that are automatable. I don't want to call anyone's job boring, but that we can automate it, that we can put it simple. Just click here, just click here. And the second step that we did an automated test is to evaluate, to say it passes, it fails. It's more or less in this area of acceptable results or it's totally out of whack and we don't want it. Those two things are the main items that we will get through an automation for testing. Validate and verify and execute actions that are, let's call it executable, could be boring, like just clicking the button as you are seeing here. But the goal is to make something easy, even easier, not repeatable, not boring.

[00:11:11] Now, the automation process could be even better. I mean, right now, as I mentioned earlier, with the coffee machine, we still we do have to push a button to start it or to even schedule it. There are some coffee machines. I had one a while ago that you could just program and at 7:00 a.m. it will just turn on and start preparing coffee. But for that, you have to previously put the beans or the ground coffee, put water, prepare everything for it. The same even as today we can microwave popcorn. Just calling Alexa. "Alexa microwave the popcorn, please." Well, the popcorn still needs to get into the microwave. And the same with a shower. There are even some that are a little bit advanced that you just walk in and starts pouring our water into everywhere. And even nowadays with bulbs that you just walk near by them and you just trigger them. Something that is already automatic is triggered automatically, but could be way even better. As I said, the initial step where you prepare everything for it just to trigger automatically those preparations and then the post steps again with the coffee machine. I said earlier you could automate the process of putting the beans into the coffee machine, putting water just connect it to the water faucet so that it always gets all what it needs and automatically runs without you intervening. But after the steps, whatever is the output, you can as well automate it so that the coffee that is ready you can get, this is an extended example that you can get a robotic arm to put it in your car, in your table, ready for you to just sip. It

could even add sugar. Prepare it. There even been some coffee baristas, I've seen them. Japan has all sorts of crazy things, right?

[00:13:02] But as well, with the blender, your smoothie, after you got all the ingredients ready and automatically through, I don't know, an automatic warehouse through a pipe that puts the fruit and yogurt and everything that you need. Again, the robotic arm could just power it, put it in your car, ready for you to go to the office when we could go to offices. And back to Richie Rich, after the shower process that started automatically you could have some post-process. You, the clean person that's the output of the shower process and you are massaged, you are calmed, and shaved not that I'm too much. And same with the light. Afterwards you are done with it. Most of them have a timer to automatically turn off after the output that it received. And you could say there's no output. I just left. That's an output. The lack of information could say, it could indicate I'm done with the light. You can turn it off now.

[00:13:57] So these are some examples, silly examples how we can extend the automation of already automated processes. We're going to get into some of those examples, why it is important. But the automating those automations and sorry if this sounds too redundant, too much like Inception it's a dream inside of a dream, but that is the goal. We want processes that are already automated to automatically start. This helps us to streamline the process, not to think about it. And you save even more time. You can iterate faster, just throw all the stuff or do other things. It frees up time for many other tasks that you may want to do. But this is the key of automating the automation when you don't have to intervene at all with your coffee machine and you can just run out and grab a coffee ready for it and be into the office earlier, that saves you some time for sleeping for, I don't know, reading a book in the morning or doing some more productive stuff than just using that time to pound the beans or put the whole beans into the machine. You get the point. You get many benefits when you automate the automation. You streamline it. And this is crucial, especially in this Agile days when we have CI/CD DevOps all the sprints where we need many of these automations to be automatically triggered. This test automations and as well, we want those outputs to automatically get again and take decisions based on them and to be even constantly executing again, like the light bulb it will be on all the time on, but as soon as we walk in, we will trigger it. So there are some tricks, but get this, in Agile days it is uber, super important that you automate the automation so that everything goes and flows easily.

[00:15:50] So how can we get this? We get this through pipelines, continuous engines. They have multiple names. These set of tools, frameworks whichever you want to call them, they enable us to put in a pipeline these types of processes. I walk in, the coffee is grounded, water is poured. It gets hit from something that is already automated and ready, gets a coffee hot, put some sugar ready for me, milk, whatever you need, if it's a frappuccino or whatever, automatically. These solutions in IT and testing worlds are such as Travis, Jenkins, Code CI, Amazon, there are multiple Azure and many, many more but these ones are more or less the main ones, the most relevant ones that you can find. That's most of what they are doing, triggering automating these processes. But for them to work, there are some key elements that you will need to have at hand with both your automations, your test automations or whatever other process and your pipeline systems like Jenkins and Travis that I mentioned earlier. The first of all is that there must be some

sort of interconnectivity to them. They need to be able to trigger each other. Same for the automation and same for the pipeline or continuous framework that we have the Jenkins and our, let's say, Selenium. They need to be able to interconnect and talk to each other. That's the key. Triggering, that's another key component. They need to be able to trigger stuff on the CI side and be triggered on the automation side. Very important. Second, communication. They need to be able to send messages to each other through a myriad of different channels. We will get into that in a moment. And last, multichannel. They need to be able to communicate through almost any thing that you can think of but we'll get into that.

[00:17:50] Other key elements for this, as I mentioned, the results. Once the communication channel sends information, it must be able to send it. It must be able to receive it, to hear it, to gather it. Once the CI system has that information, it must be able to process it, take decisions based on that, decide, evaluate and know what to do with it. And after that, do something with it being it rolling back, sending back code, letting some code pass and be checked in and start to compile many, many things that can be done. That's another key element. They need to be able to take decisions based or generate reactions based on what happens with our automations.

[00:18:32] So there are ways to trigger our processes or our automations from this pipeline or CI environments available nowadays. There are multiple ways I won't go into all of them. But some of the most important are scheduled where you call this automation, "Hey, I'm going to wake up every day weekday. At six thirty I want coffee to be ready at seven. So make sure that every weekday at seven there's coffee ready." So they will be triggering every time that that action happens. Well, time action, because there can be other triggering actions that can happen. I'm stepping ahead. The other one is manual. We can trigger these animations manually, as we have done over most of the testing automation time that we go into our test tool and click play and we are triggering them, but as well we can trigger it internally in the server that has the CI. We can go into that server, into that Jenkins, into Travis and say, "Start this process, this job." That's another one that we can do. Or remotely, these applications allow us to trigger those processes, those automations remotely from my home, from my browser, from another application through API, through many, many channels we can tell not being locked inside of that machine, "Hey, you run something" and it will start it. That's another key element and a way that we can trigger those processes. Another one that I mentioned earlier was when something happen automatically to trigger something remotely through a service. We have an API, we have a service call that initiates that. So when I walk into the room, my sensor triggers the API to turn on the light bulb. That's more or less how it works, and that's more or less how it works as well in these CI environments. The other one that is available is to chain them. When this happens, then this happens, then this happens. I'm going to use again the example for the coffee machine. When it's 7:00 a.m. first start grounding the beans and start filling up the water tank. And after you're done with those two start the brewing process, I don't know, put the ground coffee inside of the container, add some water, get it hot. Those are chained. After one finishes, the other is triggered, the other is triggered. And where it is triggered and how it happens, it's a crazy combination but we will get into that. A warning, word of advice here. All your automations need to be triggerable. I mentioned that earlier. They need to have some response channels, ways to send back the information to our CI tool and they need to be easy to be set up for executions are key

elements for our automations. An added value is that the automated script can be moved or updated and not dependent on boxes, we will get a mention of that. But you should be able to move them freely.

[00:21:38] The ways to trigger our automations as I mentioned earlier, I gate into my CI environment, into my pipeline, my Jenkins, my Travis, and I am doing that through the GUI. That's the way that I can trigger some of the processes. Get in the GUI and click run. Then as well have a URL where we can initiate a service. The service gets like, "Hey, start this process. Yeah, sure." That's more or less why the name of Jenkins like you get orders and it's like a butler that will do your commands. The common way to invoke these from the other side is through your cURL command that just triggers as many things as you need or a batch command. You can also get into the system and just in a terminal or MSDOS or batch or whatever, you can just input the command.

[00:22:30] Now, the ways to trigger this communication, because when we trigger something, we need to send some information to many of these processes. Generally, it can be through a JSON and input output schema from the command line. We can send XMLs or get where to pick it up from a database and many, many other ways that this should communicate. Again, as I was saying, it needs to be varied and available all over the place. And as well same with the response paths we can do many mixed schemas with a response pads where it can be an output from our command. It just responds with that. It can be a response as a service through a JSON, an XML, many other formats that we can get from our triggering or sending. We can get a response in an HTML. Whatever you can think of can be a response. And on the case where we initiate something asynchronously and just start it and let me know when you're done, there are some other ways where the trigger response calling another process inside of our CI because our Jenkins I'll keep calling it Jenkins, our Jenkins can receive or trigger something else when our automation is done. The very last step could be trigger the Java and Jenkins, sends my results and that's it. I'm done.

[00:23:53] So that's one way. Again, there's another schema where the automation, our test automation can put the results elsewhere and just say, "Hey, Jenkins, I'm done. You can go and check it in a database, in a file repository, in Fox TV or something elsewhere. And the last is that, again, it puts the results, our test automation puts their results somewhere and doesn't even notify. Whenever the CI, the Jenkins needs that information we'll go and check it and expect it to be available there and have prepared actions to do. And we can go all sorts of crazy paths here, whatever you need, whatever you want to do. With all these results as I mentioned, we need to be able to take actions on to those responses, validate and verify, do a pass or fail, roll back processes, start a build or a release, send notifications, call the servers, activate or deactivate with toggles many, many, many things.

[00:24:58] And as well, another element is where are we going to run our test automations? All these mix scanners will happen. Those automations can be inside of the CI box where we have our Travis server and that those automations may be already sitting inside of that server and we just trigger them. The other one is externally placed. We have wherever I created my automation system, that machine is just listening to calls and is able to initiate it or placing them on the mend. Whenever I need to start a process, my Travis

will say, "Download it from GitHub or GitLab, wherever we have it. Download it, execute it, and be done with it." This can be as well in other machines that I need the software to run our test automation and could be as well local in the cloud. Many, many mixes. There are no rules for this. We can do all sorts of mixes. Where are our test automations? Where are we going to place it? What is the communication channel? How are we going to respond? Whatever fits us the best as long as we pursue the goal of continuous automated flows that it doesn't stop, it keeps flowing, whatever channel, whatever way, only enabling for it to keep working. As well, that we can automatically trigger those automations and that those keep communicating with each other. As long as we can guarantee that that happens, that keeps flowing we are good. Whatever channel, whatever mix or specifics that we choose, we're good to go. As long as you achieve them, go nuts, integrate everything and do everything that you need.

[00:26:42] So it's time for an example. I'm going to, I was going to do a live demo of some of these things, but I don't want to run into a Microsoft Windows launch event where something might fail. And I need to be careful with time here. Here, quickly I just want to show you Jenkins. I am inside of the web interface and showing you how you can start these processes that automate all the processes. In Jenkins they are called jobs. We can start building them, create a job. There's a big button that you can see where we can indicate that we click it and we will get to some steps that we need to define first. There's one that is configured to integrate with the pipeline. Many have different names, many different have many ways to integrate. I'm just going to create freestyle project, whatever I want to integrate into it, connect and do things just for this example, because, again, this is not running. I just got some screenshots from environment that I had just to record this. And it's going to be quasi code. It's not going to be code that runs that does specifically something just to show you what I'm talking about.

[00:27:55] So I'm going to create a freestyle project and here I'm going to get many actions or options on what to do on my Jenkins job. A few of the ones that I like the most I mean, we can put a description, we can discard old builds, connect to a GitHub project, lock resources while we are running it. We can just block them. No one touches them. I'm working with them. One that I like a lot is to parameterize it. If we need to input different information for it to run differently here and there, we can do that - number of test, number of in my case, for performance we use it a lot for a number of virtual users and anything that you need a parameter input you can configure it here. Then we have some other source code management. As I mentioned, you can grab everything from elsewhere. The default is Git. Jenkins have lots of plug ins so you can get from anywhere that you can find a plugin for. I think that they have a plugin for everything like the Apple there's an app for it. And another very important one is the build triggers. This is as I mentioned, it can be remotely triggered. It will give you a URL, how to run it, a token for initiate it. It can be started after something else finishes, as I said, streamlined like dominoes. Something finishes and then it's my turn. Then it's someone else's turn or periodically. Here I am telling it how often in an hour, ten times per hour to be executed where I like some processes just to run every ten minutes to say a number. And that's how I am scheduling it or triggering it. GitHub we can also get a hook for a trigger, polls through SEM. There are many, many ways that you can initiate this and ensure that it automatically automates the automation. Okay?

[00:29:48] Other elements that we will have here in Jenkins, the build environment if it needs to delete everything before it starts working, probably will copy new stuff. I will get into that. Secret text, files, I mean, you can go nuts with all the things that you can select from the build environment, what you will need before you start running. When you start running the build, that's how it's called in Jenkins you can do all sorts of things, execute Windows batch commands, say shell, invoke an ant and cradle, run with a timeout. I mean, you can do many, many things. The most frequent that I do honestly are batch commands, but you can invoke many things. And again, there's always a plugin for that if you want to do something else. Here in this semi code that doesn't probably doesn't run, I'm showing you some of the steps that I like to do.

[00:30:42] In the first command through a Windows batch or bash, whatever you want to do I go to a specific folder that I have previously created. I can clone my GitHub repository, grab all my test scripts. Then in the second step that you see at the middle, the ScriptRunner whatever tool is that Selenium, Tosca, Jmeter, LoadRunner used to be, you can just invoke that command and trigger your automation, trigger your test automation, and let it run. If it has outputs, you can here also program what you are going to do with those outputs or if you just want to invoke them, probably as I said, because they put the results elsewhere. And in the end I can run the validator process that checks what is the results, what takes decisions or some other actions or even after that, you can put a bunch of post actions that you can do depending on what happens or just send the notifications, you can get publisher to start the build. I mean, you can do all sorts of things after this happens. So as you can see, all these streamlines and even gives you the opportunity to trigger another process after it finishes. So the possibilities here are endless. But the point is that, as I said earlier, you guarantee that everything is streamlined to keep working, keep making everything easy and automatically run always.

[00:32:07] So with that, I want to thank you all for enduring this presentation to hoping that you have learned a lot. If you have more questions, please let me know. I'm all in the social networks. Email, Twitter, YouTube, under Senor Performo LinkedIn and Facebook. On Instagram and Twitter it's srperf, YouTube senorperformo eng for English, preguntas@perfbytes.com if you want to send any question. With that, I want to thank you all for having endured all this time with me. And let's get on with the Q&A. Thank you, amigo Joe. Adios!