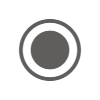
**ACCELQ Demo with ADF data science-20240709\_123216-Meeting Recording**

July 9, 2024, 7:02AM

48m 56s

 **Speaker 2** started transcription

 **Speaker 1** 0:04  
I don't what the name.

 **Speaker 2** 0:07  
Got it, got it. OK.

 **Speaker 1** 0:09  
Yeah, but like, I honestly made the statement that like we are not actively looking for any product at the moment, but like we want to understand how the product you have probably like if any kind of requirement suits us in the future, definitely we will go for it.

 **Speaker 2** 0:17  
Umm.  
OK.  
OK.  
Sure.

 **Speaker 1** 0:29  
But there is no real like exploratory at the moment.

 **Speaker 2** 0:29  
Certainly, certainly.  
Got it, Vijay.  
So so how is your automation done?  
How is your testing done?  
Basically, for your platform as of now.

 **Speaker 1** 0:42  
I don't.  
Do you want to give you a quick short summary?

 **Speaker 3** 0:44  
Yelling.  
So we have 3 multiple components.  
So for our application, so we are using selenium and testng framework and we are testing the entire application through UI.

 **Speaker 2** 0:49  
Mm-hmm.  
OK.

 **Speaker 3** 0:57  
So meanwhile we are doing it on multi level testing like unit testing.

 **Speaker 2** 0:57  
OK.

 **Speaker 3** 1:02  
Also, we are having a wide range of test cases and the API level we are having the next level of discussion with integration as well as individually testing and the third layer comes to you automation with cell name and test engine.

 **Speaker 2** 1:11  
Mm-hmm.  
OK.

 **Speaker 3** 1:14  
So this is part of our application and for role management we are using Salesforce where we are here.

 **Speaker 2** 1:19  
OK.

 **Speaker 3** 1:20  
Also, we are doing the multiple level testing and for Salesforce we are using a different component like play rate as well as connective framework for our automation.

 **Speaker 2** 1:23  
OK.

 **Speaker 3** 1:31  
And apart from this, we have other back end systems which completely tested through API.

 **Speaker 2** 1:31  
OK.  
OK, OK.

 **Speaker 3** 1:39  
So these are the major components which we are currently testing, yeah.

 **Speaker 2** 1:41  
But for API you are using which tool as of now.

 **Speaker 3** 1:44  
So we are using Java language register issue.  
So for some packages we are using Python request packages as well and the for sales force application we are using currently karate, yeah.

 **Speaker 2** 1:47  
OK.  
OK.  
Correct it.  
Clear it, OK.  
OK, great.  
So so how large is your testing team overall here?

 **Speaker 3** 2:04  
You'll be around 30 members approximately, yeah.

 **Speaker 2** 2:06  
30 Members OK, both the manual and then automation put together.

 **Speaker 3** 2:11  
We have very less number of manual resource around five.

 **Speaker 2** 2:12  
Uh manual.

 **Speaker 3** 2:14  
Yeah, five or six and all others are autumn.

 **Speaker 2** 2:14  
OK, OK.

 **Speaker 3** 2:17  
We we both both, I mean we just both.  
So whenever Nano is required so people switch chats to do the manual, and mostly they tend towards the automation.

 **Speaker 2** 2:22  
Mm-hmm.  
OK.

 **Speaker 3** 2:29  
And only when it is required, they will do the manual as well.

 **Speaker 2** 2:33  
Got it.  
Got it.  
So so with the with the selenium on test Ng framework and even with the playwright while you're testing Salesforce.  
So do you, do you see any gaps as of now?

 **Speaker 3** 2:42  
Yeah.

 **Speaker 2** 2:44  
Like while I know that you have definitely mentioned you covered automation largely, but then are there any area where you still see things are not really going fine or where it is taking too much of a time or some challenges you see here anywhere in the overall thing?

 **Speaker 3** 3:01  
Ohh so only thing is we ohm but I as of now we are doing a flow testing which have some kind of disliking us both in Salesforce as well as our framework.

 **Speaker 2** 3:09  
Mm-hmm.

 **Speaker 3** 3:13  
But there are two things to look into that one us with the design of the framework as well as the tests backness because of the environment which we are using, which we are currently working on.

 **Speaker 2** 3:19  
Umm.  
Umm.  
OK.  
OK, OK, sure.

 **Speaker 3** 3:31  
And for Salesforce? Yeah.  
So as Salesforce system itself doesn't have a direct components right?  
So the stability of those things is.

 **Speaker 2** 3:38  
Umm.

 **Speaker 3** 3:42  
Slightly uh, we are trying to manage as of now, but yeah, we are looking into.

 **Speaker 2** 3:47  
OK.

 **Speaker 1** 3:47  
So uh, it's just to give like, some, uh points about Salesforce.  
We are not using the vanilla Salesforce systems like we are also made lot of customization.  
I think that is the reasons we need, like a lot of testing, otherwise they will Salesforce like comes with like a lot of testing from that side.

 **Speaker 2** 3:58  
Personalization.

 **Speaker 1** 4:06  
But like we when we roll out like new features, all the custom features, whatever we introduce, we want to make sure that we test it for all the releases.

 **Speaker 2** 4:07  
Correct.  
Umm.  
OK.  
No.  
Fair enough.  
Fair enough.  
Yeah.  
So thanks for those insights.  
Certainly they they'll help us.  
So what we'll do is today I'll give you a little high level overview of what actual queue is.  
I'll take only two to three minutes and then after that, you know Vishveshwar either senior architect and then you know AUTOMATION export.  
He will show you how actually ACCELQ platform looks like and how it works.  
So it's a completely a codeless automation tool and then we're also very well known for all the enterprise apps automation which includes sales source.  
So I'm sure you'll find a, you know, some of some interesting points there.  
When you look at the demo, let me know if you're able to see my screen as of now.

 **Speaker 1** 5:00  
Yes.

 **Speaker 3** 5:01  
Yeah.

 **Speaker 2** 5:02  
OK, great.  
So we are a 8 year old company based in Dallas, TX.  
That's where the headquarters is.  
But then we do have a large R&D center in Hyderabad where me and Vishu work from.  
So axial queue is a unified platform today where you can you be able to test all kind of an applications you know under the same hood.  
So you can test your web apps API mobile mainframes, you know, databases, desktop, you know an email validations, PDF, M queues, it covers everything, and then travels across a, you know, a complex end to end automation journey is where you know you move from one system to another and then back to the original system.  
And in between you have so many other things coming up.  
So actually it's been designed to handle the complex systems of today's world where there are too many components and that's what actually made as a leader in the forest wave where they, you know, Forrester does an analysis of, you know, who who is doing best in the continuous automation platform.  
And then axial case been declared a leader within the three years of our starting one because of the unified nature.  
And 2nd is also the codeless capabilities that we bring in and then also some of the AML capabilities that we also you know have in the system which makes life easier for automation teams.  
So this you see this table.  
You know, Forrester picked around some 30 plus different tools sometime back and then did a detailed evaluation of them on multiple parameters.  
And then declared, you know the top 15 here in the table and yes, you see Axial queue has been given the highest rating.  
So we have done better than a lot of other tools.  
I'm sure you will be able to identify most of them in the table, like tricentis microfocus Parekh OFT IBM you know they have been there from longer time and then you know what Axel QS been done given a higher rating than all of this on multiple parameters.  
So what makes U.S.  
Special and what makes us better than anyone else?  
One, it say entirely.  
A cloud native SAS platform which is very easy to deploy, easy to scale up, easy to add new functionality where you can write your own libraries that suits your business under the same time, uh, you know, design to manage, end to end complex scenarios that is on the application side.  
But if you come to the end user side, you know basically the tester side.  
It's been developed on a low code.  
It's a development kind of a model, so anybody who don't have any much of a programming knowledge can also use this tool very, very well and then can learn it very quickly.  
So a typical learning curve is like three to five weeks for anybody.  
So that's what actually makes it.  
Uh, you know?  
Uh.  
A champion in the overall world because as you see, you know, always most of the teams will have a mix of automation and then manual testers and then they would like to migrate the manual testing teams into automation as soon as possible.  
And Axel Queue enables that very easily and we have all the integrations with the you know the Ajay and DevOps tools.  
So are you using something like Jira, Jenkins or your DevOps?  
Kind of tools here ohh.

 **Speaker 3** 8:27  
Yeah, but we are using JIRA.

 **Speaker 2** 8:29  
Geethu.  
OK.

 **Speaker 3** 8:29  
They did a Jenkins.

 **Speaker 2** 8:30  
So yeah, so so there are out of box integrations with Jira, Jenkins, you know you can directly log your bug from Axel community that and then you know you can see the traceable destablished and multiple AML capabilities.  
We use a smart locator approach.  
You know where system can identify all elements automatically which you know ensures that you're automation doesn't break in spite of your uh changes.  
Very strong test data generation capabilities that saves time and then there are auto read and facilities and then there is self feeling that kicks in.  
You know during runtime where system can intuitively understand what it was before and what kind of properties they've changed, and then the tags which makes the reconciliation a lot more easier and faster.  
And there is a for every application under test, we'll have a blueprint, you know, which you will be able to see in a design studio, which we call it as an universe.  
As you develop your test scenarios you the universe get developed and then you will have a visual appeal to your overall automation.  
Coming to technologies, there are no limitations.  
You know, we can test all kind of web technologies when it comes to mobile.  
You know, we can test your native with Android apps, browser driven Applications API.  
We can cover your risk so you know microservices.  
Any desktop windows legal simplifications can be managed.  
Mainframes can be managed and sales force dynamics Oracle, ServiceNow, Pega, SAP.  
We are extremely strong in testing all this.  
We actually started with sales source as the first application where we call it Axel Queue live.  
So other problem normally you see is you know there are at least three liters in a year from Salesforce and whenever there is a new release, your transition in automation need some kind of a tweaks every time.  
We were moving from one to the other, but we have a collaboration with Salesforce where we patch all those changes into axial queue much in advance so that even the release changes also will not affect your overall fails for automation.  
And we have many many pre built assets in the Salesforce so that a lot of you are automation testing is easier and then you know you can also uh you know man over your iframes problems and all with with the Axel we'll show more in the on the Salesforce front in the demo so yeah yeah.

 **Speaker 3** 10:59  
But do you also, sorry Internet.  
So do we also have GRPC?  
Just ability function from here, protonation.

 **Speaker 2** 11:08  
What is it, George GRPC?

 **Speaker 3** 11:12  
Yeah, but then we can test the GRPC from the Excel queue.  
Uh, any services with GRPC?

 **Speaker 4** 11:18  
Yeah, I think we can do that or yes, possible.

 **Speaker 3** 11:21  
OK.

 **Speaker 2** 11:24  
Thank you, Kishore.

 **Speaker 4** 11:24  
It's really added that recently but.

 **Speaker 2** 11:27  
Ohh, OK OK great.

 **Speaker 4** 11:28  
Capability, yeah.

 **Speaker 2** 11:31  
Yeah.  
So yeah, we also have integrations with the browser stack Lambda test, you know, headspin sauce labs kind of companies so that you know, even if you want to run multiple cloud executions or want to test it on different emulated devices on mobile, you know you kill, you be able to do all that easily from this.  
So compared to you know any traditional development like you know working on selenium test Ng, you know there is no lead time for any framework development as you load your application, your automation starts in 5 minutes.  
OK, with a click of your mouse you will see so many commands available.  
All you need to do is select the right one and then put them together so you are framework developmentally.  
Time is very minimal.  
Your script development is a lot more faster because you're just writing English like commands and then you know, yeah, your scripts are ready.  
You know the element identification is a lot more easier, which ensures that your doesn't break, you know, and then integrations with all these upstream downstream applications ensure that they change.  
Management is very strong and then the test executions you can run on multiple staging environments.  
You can create your own staging environments.  
There are auto read and facilities, you know beyond a certain percentage of bugs.  
If you find system which is, you know and an anomaly you know system can also run it by its own which you can define all these parameters you know you can get your reports and email teams and slack.  
You can get your reports, so all this will make your overall productivity much higher and then your test creation, maintenance efforts much lower.  
So majority of our customers who were running large teams before they got ACCELQ have, you know became a lot more cleaner.  
But at the same time, a lot more efficient.  
OK so here are some of our you know customers, you know the small list, we have multiple hundreds of them and because you know you are coming from a financial services background.  
But of course, though you are IT company.  
But then in the same domain, so we are working with many banks today, a lot of non banking financial corporations, credit unions, you know, large companies like Wells Fargo apps or Comerica, you know in fact we work with the Federal Reserve Bank of USA also under which there are 12 states you know which are testing their banking applications using Accel queue and coming to you know system integrators The Who is who of of you know say like in fee Accenture Cap Gemini they came they all work with the Axel Key today and then we work on.  
Multiple pursuits together, we help them with POC's proposals and then you know they're actively selling on taking Axel queue to their end clients because you know it's a lot more easier.  
We do a lot of partner trainings for them, you know, give them, make them make their engineers certified.  
I'll do it all day long.  
Workshops for this large companies.  
OK, so that's the journey so far.  
Uh, so I'll take a pause.  
There you have any more questions, please ask me.

 **Speaker 3** 14:48  
No, but.

 **Speaker 2** 14:48  
OK.  
OK, great.  
So.  
So we will go to the demo, so we should will run over the.  
Demo and then please stop him.  
Wherever you have any questions.

 **Speaker 4** 15:05  
Yeah, hope you are able to see my screen.

 **Speaker 1** 15:11  
Yes.

 **Speaker 4** 15:12  
Yeah. So.

 **Speaker 3** 15:12  
But.

 **Speaker 4** 15:15  
You can see my screen right?

 **Speaker 3** 15:16  
Yeah.  
Yeah, this is, yeah.

 **Speaker 4** 15:18  
Yeah.  
OK, so this is how our platform would look like as Bharat was mentioning Accel Tuesday SAS based test automation platform and then it is accessed via Web URL doesn't need any separate ID where you need to do the development.  
Everything would happen on this a the the web application itself.  
OK, so I will explain how easy it would be to do the automation on the platform.  
So let me create one small flow.  
So and then I'll explain the different concepts involved.  
And this is single single uh.  
The framework using which you can do the variety of automation, so be it the web automation, the mobile, the desktop, the you want to do the database testing the API.  
So you would be doing everything from one single roof.  
So say for example I want to like do some kind of interactions on a web application.  
I see.  
I want to log in and then do some kind of the assertions over here.  
OK.  
I'll I'll explain how easy it would be.  
So one thing we need to remember is the Axel queue, the kind of well I should be.  
There you are doing the automation on a custom web application or the ERP applications like the Salesforce SAP.  
The kind of it remains same, basically like a the framework is technology agnostic.  
So the.  
Usually for any kind of about the mission, you start with the launching the browser.  
OK, so you can so the left side where you can interact and then right side there will be a live browser and then you can build your interaction.  
So using by interacting with the live browser and you can get the feedback on what is happening at the bottom of your screen.  
So this is my say the application say want to and if it is turned off or then you can use this as a regular browser.  
So now I want to create a login action.  
So I want to enter the username I want to enter the password and click on this button.  
So just to mouse over and then you can say I want to enter the text.  
So what I want to enter so I want to enter the user name.  
Uh, good.  
So let me pick those details.  
And then I want to enter the password here and it knows that it is a encrypted field.  
It's a password field.  
It is suggesting to utilize the encrypted encryption commander.  
So then finally I want to click on the sign in button, so I'll do a mouse over and then click on the stilmant.  
And so you can select all these statements and I can do your playback.  
Then whatever interaction you're making, you are doing, so those would be happening live on the browser.  
So basically the advantage is you are creating this script and it the same time you're validating that it is working good as well.  
OK.  
And then say now I want to perform some kind of assertions, he said.  
I want to verify that OK these fund transfer meaning item exists OK, I can just simply say that OK verify the element exists.  
And then they can perform those assertion and we were that assertion is there.  
OK, you will see that the green color tick mark indicating that it is passed OK.  
In case if the assertion fails, what do you want to do?  
OK, you want to report that as an error and then continue or it is a critical element you don't want to continue further, so you didn't know about the test case also.  
So you similar to these interactions like the entering the text, clicking on a button.  
OK, you would have the different the the commands for different the the element types.  
Say you want to select a value from the drop down, OK or you want to do some kind of operations on a the checkbox.  
So you would see all those operations.  
Uh, OK and say for Salesforce you would have the controls where some lookup kind of where where you enter the text, OK and then it shows a list of the available values with that search.  
Then you want to pick up something from the list, so you would have the commands to, uh, search and the select from the drop down Azure to to that easy it would be to create a the interaction.  
So for example, you have created one business function automation for the business function you have done and you want to utilize them in different other the flows.  
So you can create some kind of reusable artifacts.  
I can call this as perform login.  
Yeah, I can give some kind of the page where I'm performing this operation and you can parameterize the values for the username and then the password so that you can utilize the same script for the different between different credentials.  
So now this has become your reasonable entity, like the perform login and then whether you are doing a fund transfer or you are doing some kind of verification of the few transactions so you can utilize this.  
So the log in the function.  
OK, so this is how you kind of build your the business flows.  
It's once you create this business flows, OK, you kind of finalize then it would become a a the scenarios like this.  
OK, so you launch the browser.  
So for example, you're creating and converting the lead, so you launch the browser, login personate, create, delete, verify the lead details, convert, and then verify the opportunity.  
So these all you kind of create individual interactions and the make a business flow and you can vary that the the flow with different permutations and combination using the test cases.  
So basically you provide the different values for those parameters, so that makes it is the data driven.  
So and that makes the different test cases for a business flow.  
So I'll I'll just take a pause here.  
Any questions so far?

 **Speaker 1** 22:30  
One person from my side, like how in what are the cases you explain how are you using a here.

 **Speaker 4** 22:37  
Umm.  
So Jane used it multiple places.  
One, I think it knows that.  
OK.  
What type of element you are interacting with?

 **Speaker 1** 22:49  
OK.

 **Speaker 4** 22:49  
OK.  
And then when you are say for example, instead of the text based approach, OK, say you can use the locator free approach.  
Also, whatever interaction we are made, we are not using any Dom structure of the page.  
We are utilizing the the locator free approach and in case if it is a smart locator and all OK.

 **Speaker 1** 23:03  
OK.

 **Speaker 4** 23:11  
So the selection of those attributes.  
Uh, you you would be like a the A would be coming in there and then?  
When you are creating the test cases.  
The like you can auto generate the test cases.  
So there a comes into picture.  
So there are multiple places.  
It's not just one place.  
The algorithm is self feeling is another classic example where we use the AL Thems in identifying the changes to the UI and then continuing the executions.

 **Speaker 1** 23:51  
OK. Yeah.

 **Speaker 4** 23:52  
Yeah, makes sense.  
OK, so you create the multiple test cases and then you kind of do the executions like you can schedule.  
You can run right away.  
You can integrate with your CI tool.  
Say you have the Jenkins Cir.  
The Bam Azure DevOps so you can link which were CA tools and then kick off the pipeline from the CA tools as well.  
So same script do you can run it against the different environments you want to run against test staging, production.  
You can do that.  
You can choose which browser you want to run it against.  
The Chrome, Firefox, Edge you can do the task well.  
You can, but it against the different team you laters as well.  
You can control the browser behavior using the profiles like you want to run it in headless mode.  
You want to turn off the notifications so you can start using the driver profiles.  
So we also have this the app testing.  
It can be native app or it can be hybrid app.  
So you can OK, we we we kind of support the mobile automation as well.  
You can control the level of screenshots.  
I'll show you like how these screenshots would look like for a result for a test execution result, and these are the default page and then element load times, but you can control the synchronization conditions.  
So between the action logic as well, so some of the elements or some of the pages may take more than these times and you can do the execution in there like multiple test cases.  
You can run them in parallel as well, and then you can do the notifications through the email.  
The teams are the slack channel as well, and we talked about the cell feeling you can set up some kind of the auto rerun strategies like I want to rerun the execution if the percentage of the failure sees some experts and and or but the there are like more number of failures.  
OK, you can kind of set up your the rerun.  
Uh strategies.  
OK, so these are the different options you can use for doing the executions.

 **Speaker 3** 26:09  
Uh.

 **Speaker 4** 26:11  
Yeah, good.

 **Speaker 3** 26:11  
She's in.  
So you were saying about the browser execution, right?  
So how do we test the different version?

 **Speaker 4** 26:15  
No.

 **Speaker 3** 26:17  
So it is taking up the system version.  
What I have in selling my local or?

 **Speaker 4** 26:23  
Yeah.  
It so whichever agent you are picking up, say for example, uh, if I'm picking correct, if I'm picking my agent, it will pick the browser which is available on this agent machine.

 **Speaker 3** 26:29  
And let's assume we are taking Chrome so.  
OK.

 **Speaker 4** 26:39  
So if I pick a, say, the cloud agent, it will pick the browsers that are available on this agent machine.  
So in case if the requirement is to test against the different versions of a browser, OK, one of the option you need to keep the respect to versions on that agent machine or we also have this integrations with the the the different the the device forms.  
OK, so you can integrate with the the browser stack.  
You can integrate with the this all slabs.  
You can do the attachment, say for example this browsers to.  
I can do it with browser stack sauce labs perfect to Lambda test.  
You can integrate quite easily.  
Once that integration is set up the all the devices, all the, all the browsers available against the device farms would appear over here.

 **Speaker 3** 27:33  
OK.  
I thank you.

 **Speaker 1** 27:38  
And then we talked about CA integration, right?  
Let's say if we have multiple pipelines, can we group uh some test cases like towards like 1 pipeline will be able to trigger from the CA site rather than configure it from here?

 **Speaker 4** 27:42  
Ohh.  
Yeah.  
Actually we we triggered from the CI itself.  
I'll, I'll tell you how it is done.  
So I think you have different test cases.  
OK, we have the concept of called the test suites.  
So you can combine the different the test cases based on some criteria into the test fields.  
So you want to combine all your regression test cases.  
You want to combine all your test cases that belongs to a particular release into your test suite.  
OK, So what you do here actually is when you create a job, you would get some kind of the job ID.  
OK, say for example, I'll I'll create this job.  
To I'll pick up the test suite.  
And then I'll create a job.  
You get a number here.  
OK.  
And this is a static.  
OK, it doesn't change for every execution, so you have set up this test suite.  
OK, say and you have created this test feed based on some kind of conditions.  
Say for example, uh this test suite contains all the test cases that has the tags as regression.  
OK, so as and when you add the other test cases with that tags, those would automatically get added into your test suite.  
So you need not change the the criteria for creating creation of your work.  
That is sweet.  
OK.  
And we were talking about how you trigger it.  
So basically like you got these, uh, the job ID, the CA job ID, that job ID, you kind of configure on the respective the CA tool say for example.  
These are our just opening the Knowledge article related to Jenkins.  
So we have a the CI connect the axle, QC connect is present on the plugin is present on the Jenkins the marketplace.  
So you kind of download that the plug in on the Jenkins and then provide this some kind of configurations.  
OK, so one of the configuration item is the job ID.  
OK, so whatever job ID we have created here so that is what you give over here.  
And then this pipeline would be triggered on the the Jenkins itself, but the execution, the triggering point would be your junkins, but the execution would happen on the axial platform and you would also get the results link on the the respective CI, the console. OK.  
So this results link will route it back to the actual true results over here.  
The does it answer your question or or I have confused you. Yeah.

 **Speaker 1** 30:43  
Yeah.  
Thanks.  
Thanks.  
You too, handsome.

 **Speaker 4** 30:47  
Sure.  
OK, so that is how you kind of do the execution of your scenarios or your test suite and all the results would be stored into the results section.  
So let me show you how the results would look like.  
So these are test suite and then that contains 9 different test cases say how many past, how many field you would get that and then each test case you would see the results console.  
OK, it will be on the same line.  
The way the script is written, you launch the browser OK, you enter the in username.  
OK, you enter the encrypted password.  
OK.  
You clicked on this login button at each of these interactions, so you would see the screenshot of the application also like.  
Uh, say it is a synchronizing well on the page and it clicked on this the search button.  
OK.  
So you would kind of the navigate between the previous and then the next buttons to understand how the execution has happened and the wherever that interaction is happening you would see that portion is gets highlighted as well.  
It is now waiting for that user detail to appear.  
And then this is a clicking on this button.  
OK, so you can navigate between the previous and next, or you can look at the details in the slide show as well.  
So that is how the results would look like.  
And if you want to look at the different scales you can, there is a the the format will be the same thing in case if there is any issue.  
OK, you can create a defect.  
You can integrate with your defect tracking system and create the defect right from the results itself.  
You can attach the screenshots while creating the defect, so that would be beneficial.  
Who is going to work on the defect resolution?  
So you can also like, yeah, go ahead please.

 **Speaker 3** 33:01  
But we shouldn't.  
Uh, so two questions here.  
So one is related to how the failure looks like.  
Uh.  
With each and every step and each and every page has some page load time, right?  
So each and every page wait item is directly handled by the system itself or we have to give wait times for page loading and proceeding further, because based on the test environment or other environment, the loading time of the applications might differ, right?

 **Speaker 4** 33:16  
Umm.

 **Speaker 3** 33:32  
And the production segment, yeah.

 **Speaker 4** 33:32  
Correct.  
To the page load times that you that would be controlled the the at the time of setting up the A, the execution model itself.  
OK, so here you see that page load time is 60 seconds and then element locked them in 30 seconds.  
So if you want to change it in case if you're running running it up against your test environment, the page would take more time.

 **Speaker 3** 33:52  
Umm.

 **Speaker 4** 33:59  
So you can give you can increase as well.  
OK, in any case the these are like kind of the maximum time it would wait and even after 2 minutes if the page doesn't get loaded it will fail.  
OK, but in case if the page gets loaded within two seconds 3 seconds then it will continue the operations from that point itself.  
OK, it doesn't wait 120 seconds every time.  
It's a kind of dynamic wait.  
So that is how you kind of handle the dynamic, the loading of the elements and then the page between your the action logic.  
Uh.  
And then what was the other question, sorry.

 **Speaker 3** 34:41  
How how does the failure exactly looks like?

 **Speaker 4** 34:44  
Look right.

 **Speaker 3** 34:44  
Will you be?

 **Speaker 4** 34:45  
OK.  
So let me show you the things some there could be some failures.  
We'll just open and then show you.  
So there is a.  
So you see that there is one total and then one is failed over here and zero passed that that may open that and then see.  
There was some issue with the APM.  
OK, so the basically it looks like this.  
Whenever there is any failure, you get the cross with mark over here and then you get some kind of more details by clicking on this more link.  
Uh, but.  
True is the API, so API you were expecting the status code, the response code to be 200 but it is 400, so it's a kind of the field.  
So this is another way other failure.  
How it looks OK this statement level you see where it failed and the uh the action level, the step level also you see there is a failure indication and in case if everything is the successful you would see the green colour tick mark for those executions.  
You would see that all these steps are like kind of marked green and all these the page loadings.  
OK, so the synchronization it is verified, the page contains the text OK those would be like with the the green color tick mark and wherever some other the assertions are made.  
OK, say I'm expecting the the name.  
Uh, they leave name should contain this detail.  
OK, which is passed as an input parameter, so do those are like kind of validated that also would have the green color tick mark, but these are the different ways.  
So you can look at to understand what the issue is and then how you can go about the resolve.

 **Speaker 3** 36:49  
Yeah.  
Thank you.

 **Speaker 4** 36:56  
Then we also have, I think we talked about the CI integration, we talked about the the defect integration.  
You can also like establish the traceability with the requirements tool as well.  
OK, say you you have some kind of 00 stories against which you are creating these the scripts, so you can kind of establish that mapping between your the requirements and the distance scripts.  
And this is the kind of the bidirectional, the traceability.  
So you click here and then it will take you to the JIRA and then this state is whatever is coming from coming showing up here that is directly coming from the JIRA.  
OK.  
And it's a kind of the real time, the integration as and when you change that the status or the values over there that would be changed over here, that helps the automation engineer to understand the status of ohh the either the requirement or the kind of the defects associated against this ohh scenario.  
Just take a couple of things, OK?  
And one of the thing OK you would see that in many kind of automation.  
So the test data management is one of the critical items, OK?  
You don't want to like, parameterize too much.  
That will become.  
That will increase the maintainability perspective.  
So axial Queue provides a lot of random generation of the test data, so you can generate the random first name, last name, the email address, physical address, so and then you can store it into some kind of variable and then utilize them for entering it on to the UI R validating against the API.  
You can do that.  
You can also generate the data of the random dates, so in a specific format, the past in the future, all those you would be able to generate.  
And I was, as I was telling API's also like quite easily.  
You can do it and we provide a kind of the wizard that will help in easily creating the API calls.  
So you provide the endpoint, you give a a.  
What type of API call you're making provide the headers and then the next screen there'll be a placeholder for providing the the payload.  
You get the response the dot.  
You can use wrapping trap that ohh the response whether it is in Jason or it is an XML.  
You can do that and then you can use that for the further processing or comparing it against the UI RF the database.  
Any any other questions?

 **Speaker 3** 40:00  
OK, I also know database connection OK yeah.

 **Speaker 4** 40:03  
Yeah, we we can do.  
Yeah, that is why I wanted to show so we can connect to different kind of the database beat the SQL Server, Oracle, my SQL or it is the the Mongo DB.  
So all those database can be connected and no separate plugins are required.  
These are all baked into the framework, so you just utilize these connection and either you provide the credentials and then or use JDBC connection string to connect to the database, retrieve the information and then update.  
And did whatever operation you want to do that, and even for the best sales force, you can make this equal connection and then perform the queries as well.  
You can do the PDF validation so you can connect to the emails.  
You can perform the operation on the Excel the text files, so you can do some kind of the operations on the, uh, the windows you want to do some kind of the the operations on the SSL.  
OK, you can do variety of operations from the platform.  
Any any other specific questions?

 **Speaker 3** 41:36  
No, she said.

 **Speaker 4** 41:41  
That's all I wanted to cover.  
And there are any specific questions so.

 **Speaker 2** 41:54  
So how do you find these Arun and Vijay like compared to what you've been doing now?

 **Speaker 1** 42:07  
I think it provides like lot of.  
Uh efficiency improvement in terms of?  
Multi Tools connections jira.  
Let's see a plus the automations.

 **Speaker 2** 42:20  
Yeah.

 **Speaker 1** 42:24  
Also, you're talking about building it once and like passing it to the data in the configurable way.  
I think all these subprocessor like also a lot of things are like configuration based right and just in the right browsers and everythings as part of automations.

 **Speaker 2** 42:37  
Correct.

 **Speaker 1** 42:44  
I think that is really helpful, yeah.

 **Speaker 2** 42:49  
Yeah, absolutely.  
Yeah, yeah.

 **Speaker 1** 42:53  
I think overall it looks good as it supports for mobile desktops mobile app also as part of the flow you can do the various connections, all the elements plus database connections.

 **Speaker 2** 43:01  
Umm.

 **Speaker 1** 43:08  
Uh, I I think this cells also the scenario based grouping into multiple scenarios, bidirectional connection between JIRA and all other things.  
I think it will be really helpful, yeah.

 **Speaker 2** 43:23  
Sure, sure.  
And and I know you Salesforce testing can also be improved in a very big way.  
Ohh rather than because today we are we are largely you know known across the Salesforce ecosystem for automation.  
While we have capability to all others also, but then we've been part of multiple Dream force and then we go to a lot of these world tour essential tools that happen in multiple countries and then we get many, many leads for Salesforce and in fact even the all the large system integrators also do reach out to us at least you know 50% of the leads that come for Salesforce automation big because they find it very challenging to do with the with other tools.  
OK, so so you it is like you know one stop shop for all your needs and I'm sure today you're running a you know definitely a larger team of around 2430 people.  
But then I think you should be able to, yeah, do automation with a much leaner team with Axel queue and then maybe you know you can you'll be able to do a lot more coverage with the same number of people.  
So think about it.  
I think you should you.  
You should maybe start with the with small number of licenses also and then maybe start your sales or automation and then think about doing some kind of a bring in the platform.  
Later, I'll.  
I'll send you more information also anyway on how exactly we are aligned with Salesforce and send you some kind of a comparison with selenium.  
How?  
How are we better than that?

 **Speaker 1** 45:01  
I think that is helpful.  
Also you can how your pricing model here.

 **Speaker 2** 45:06  
So the the pricing model is basically either user based pricing model so named user licenses we sell and then it can come with.

 **Speaker 1** 45:17  
So when you say user base OK, sorry, go ahead.

 **Speaker 2** 45:20  
Yeah.  
Yeah, so so every user will have one license and then there is no concurrency as of now.  
So.  
So you particular that you know you need 5 licenses, 5 users will be able to use those and they are, each license will be assigned to 1 user and of course let us, yeah.

 **Speaker 1** 45:34  
OK, that means that they can run with like a number of iterations, but all those things will go sequentially.

 **Speaker 2** 45:41  
Uh, yeah, correct.  
So.  
So these five, let us say, you know they can run any number of applications, there are no limitations on number of executions, that number of applications you want to test, but only when one user is logged in, the other cannot log in with the same user ID.  
That's the only limitation.

 **Speaker 1** 46:02  
OK, got it.

 **Speaker 2** 46:04  
Yeah, and it works on a hybrid model also.  
So you know, while you the the application, the whole everything, the test assets all around cloud, you can run your executions from your local machine.  
So there is a small local region that we install and then the executions can happen from your local PC.

 **Speaker 1** 46:25  
OK, got it.

 **Speaker 2** 46:26  
Yeah.  
So you have native mobile apps also correct?  
Uh, Android I OS both.

 **Speaker 1** 46:34  
But we don't know any mobile app.  
We have only responsible, yes, yeah.

 **Speaker 2** 46:36  
OK.  
No, no, only web and API OK.  
I don't know.  
Sure.  
So so I think you we have we know we sell licenses based on a unified license where everything is covered or you know just Web API is also possible.  
And you can always upgrade when you have some mobile component at later.  
Let's say you build a mobile app.  
After 1-2 years, you can always upgrade it to a mobile license also.  
So which can cover everything, but you can start with Web and API.

 **Speaker 1** 47:08  
Yeah.  
OK, maybe you can share like all the recording and your presentation.  
Any other additional details I as I mentioned like a start of the presentations like no plans, nothing budgeted, but if if anything we feel interested or some things we will contact you this. Yeah.

 **Speaker 2** 47:14  
Sure.  
I know when I completely agree.  
Sure.  
Definitely, definitely.  
See, this is definitely a better way to do automation.  
I'm sure it will.  
What you call give you a lot more coverage and then you'll be able to do with with lesser cost also for sure because while there is a license cost to it.  
But then, as you know, especially the selenium resources are very expensive nowadays.  
So we don't get them cheaper.  
So with the cost of 1 selenium resource, you can actually buy file licenses and then maybe you know give it to 5 manual testers and then convert them into automation engineers and then we'll be able to do better larger automation, a lot more simpler and then regression will be a lot more easier.  
It will take less time than what it does in a in an open source, and it doesn't break all that.  
OK, a lot of advantages and I'll send you more information and we'll stay in touch.  
And in case you are really ready to explore it and then you know, start at a small scale week.  
We can also do a small no cost POC for you, so if you can bring us a complex use case, we can always uh uh, you know, do a POC and then show the demo to you.

 **Speaker 1** 48:44  
Sure.  
Thanks for your support.

 **Speaker 2** 48:47  
Thank you, Vijay.  
Thank you, Arun.  
Nice talking to you.

 **Speaker 3** 48:50  
Thank you.

 **Speaker 2** 48:50  
Good day mobile.

 **Speaker 3** 48:50  
Thank you.  
Thank you.

 **Speaker 1** 48:52  
Thank you.

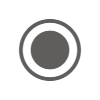
 **Speaker 2** 48:52  
Thanks.

 **Speaker 1** 48:52  
Thank you.

 **Speaker 2** 48:53  
By right.

 **Speaker 4** 48:54  
Thanks, bye.

 **Speaker 3** 48:54  
God.

 **Speaker 2** stopped transcription