

Greg Lotko: Hey, folks. Welcome back to the next episode of The Main Scoop. I'm Greg Lotko, joined by my co-host, Dan Newman. Hey, welcome back.

Daniel Newman: Yeah, Greg, it's good to be here.

Greg Lotko: Absolutely.

Daniel Newman: I am always in the mood to scoop with my good friend, Greg Lotko. How's it going?

Greg Lotko: Today we're going to be talking about application testing and development. I mean, the reality is everyone wants to go off and write code. They want to create capability, they want to deploy it out there. But we all want to make sure that that works right and that it doesn't interfere with stuff that's already out there or go fluid. It's a technical term.

Daniel Newman: Yeah. You want to make sure that when things get updated and changed that nothing breaks, and there is a lot to it. And across the entire tech landscape, the integrity of QA and of making sure that software is properly tested, that app developers are innovating intelligently and safely, it's not negligible. It's pretty darn important.

Greg Lotko: And you want some consistency to that, right? You don't want a free for all in your development teams, but I mean, we could go on and on about this.

Daniel Newman: We could talk about it.

Greg Lotko: Well, let's bring in our guest. With us today is Bertrand Cardinal from Desjardins.

Daniel Newman: Yeah.

Greg Lotko: Tell us about yourself, what you do, and then jump right in here, talk about the subject.

Bertrand Cardinal: All right. Bertrand Cardinal. I'm working on the biggest credit union in Canada, which is Desjardins. Some people saying banking, but it's close. But-

Greg Lotko: Lots of money either way.

Bertrand Cardinal: Yeah, both. Yeah. Lots of money, but it's the biggest credit union in Canada. I'm responsible of the banking solution because of course it's only just not doing just banking, but insurance and a lot of topic. But I'm especially responsible for modernization of the banking solution that reside on the ZUS.

Greg Lotko: Very cool. All right. What do you think about when you think about application development and testing this stuff? Do you just hack away and throw it over the transom?

Bertrand Cardinal: Well, no, for sure. No.

Greg Lotko: Good.

Bertrand Cardinal: That's a first good answer. Of course, you can test and test any new development with unit tests, but there's more just than the quality of the product that you need to deliver. Unit tests can be useful for documentation, especially for senior that's retiring.

Greg Lotko: Mm-hmm.

Bertrand Cardinal: A lot of people leaving Desjardins because of their age, 20, 30 years that they're working with us. Of course when there's a really complex module, we want to document those modules and unit test is a way to document. That's first thing for sure. Second thing, of course, unit tests can trap bugs before going to prod for sure, especially automated test. But also if you can trap it earlier, it's going to cost less.

Greg Lotko: And I think that's the thing we more often hear people talk about. We think about quality, we think about improving it, we want to catch defects earlier in the process. But the other stuff was really interesting. And on top of it, the idea that you're documenting what's going on in these applications or in these programs, preparing or thinking about that, whether it be because of retirement or somebody moves on or they're on vacation, that the initial developer might not be the person who has to do an enhancement or God forbid, resolve an issue.

Bertrand Cardinal: Agree. And especially since we have a legacy of old development that is not really well documented, there's a lot of reverse engineering that we need to do.

Greg Lotko: Not any of the code I wrote.

Bertrand Cardinal: No, no, for sure. I'm pretty sure it's pretty well documented. But let's say seriously it's code from the seventies, eighties, and nineties that we need to do reverse engineering to make sure we understand business rules and everything.

Greg Lotko: We don't want to add to that problem, right? The reality is the age of the code doesn't matter if it's effective, it's efficient if it's still doing what you wanted it to do. But the problem is when you look at that code and you're like, I don't remember exactly why this is doing this and there's nothing outlining it, certainly going forward, we don't want to create new code that's undocumented, but we want to go back and look at the other stuff.

Bertrand Cardinal: And for the engineer, it's how do you say, it's useful, but it's always make sure that he's doing the right thing.

Greg Lotko: mMm-hmm.

Bertrand Cardinal: It's kind of a filet to make sure that he doesn't do any additional bug that we covered with those unit tests prior to his changes.

Greg Lotko: Très bien.

Daniel Newman: How are you thinking about the process of empowering developer teams? In the work that you're doing for Desjardins and also just more broadly across the industry, how are you thinking about a developer centric approach?

Bertrand Cardinal: And we didn't need to discuss for Desjardins, I don't want to talk, but-

Daniel Newman: Broadly. Broadly a little bit.

Bertrand Cardinal: Yeah. Well, a lot of investment was done on the distributed side. A lot of the new stuff, the cold coverage, the unit testing, everything. There was a lot of effort with the new tools on the distributed side. But of course there was less, let's say investment done on the mainframe side. But for the, let's say four or five years, we changed that. And we try to, not mimic, but try to get closer to what the distributed side is doing.

Greg Lotko: Give the developer a common experience across platforms.

Bertrand Cardinal: Common experience. Exactly. Common UI, common unit test and coverage and unit coverage and everything like that.

Greg Lotko: Leverage the power and the strength of the platform, but make the interaction consistent.

Bertrand Cardinal: Yeah. Of course, it's a goal. We're not there yet, but one step at a time.

Greg Lotko: Getting closer every day.

Bertrand Cardinal: Exactly. Yeah.

Daniel Newman: How much are you seeing things being sped up and automated? I mean, one of the things I know that we're super focused on societally, this can be on the mainframe, off the mainframe, right? You have the power of AI of course is very centralized.

Daniel Newman: Yes. But I mean I do think that there's a lot of discussion right now, whether it's the actual process of testing, you're going to see some automation done there, whether it's the actual code creation, there's going to be some automation there. But of course you want to be developer-centric, which means you want to do this with the developers, not in contrast to. You're not replacing and displacing, you're augmenting and adding. Is the process as you see it moving forward while still maintaining quality, is automation a big part of it?

Bertrand Cardinal: Well, especially on the mainframe side for us, we at the beginning for sure.

Every small step that we going forward on automating test, adding or make sure that new code has that unit test integrated at the same time. But definitely we need to empower that way to make sure that we have a full set of unit tests to make sure that we have a nice, I don't know, safe file.

Greg Lotko: Nice combination, correlation.

Bertrand Cardinal: Combination or safety. It's a safety. Unit test is make sure that it's a safe net that we make sure before going to prod.

Greg Lotko: But then you keep it together with the code going forward, right?

Bertrand Cardinal: Of course.

Greg Lotko: That as you do enhancements, as you do changes, you can have the confidence that you haven't changed anything you didn't intend to.

Bertrand Cardinal: Yeah. And especially unit test, the main goal of that is trapping the bug as soon as it is written and not once it's in the production. The cost is really triple or quadruple as each step, the code is moving forward, close to the prod, of course it's going to cost more to correct it. As soon as we trap it, of course it's really easier to correct it and make sure we have a quality.

Greg Lotko: I've got an invention idea. Maybe we need to develop a productivity tool so that as the programmer is about to write that defect, it just zaps their fingers and they can't type the bug. It might be hard to develop though. Somebody would have to code that.

Daniel Newman: We need to locks your fingers on the keyboard, doesn't let you remove them. Well, you need that negative reinforcement sometimes. Zap, zap. Well, another thing though that you would think, Bertrand, for developer centric, it's also going to be very much about bringing the next generation of developers along. We've spent some time over the years talking about new skills. We talk about training. People can go out, we'll link you back to those episodes. But we also talk about enticing people into the space, having developer centric platforms that encourage. It seems like that's something that would be pretty important in any developer approach.

Bertrand Cardinal: And cross-platform no matter what. Developer that has the skills and knowledge of the developer, but cross-platform so he can do the end to end of what's requested by the business side for sure.

Greg Lotko: Whether that be searching for an issue when it happens or making an enhancement. Somebody who can use common tools and doesn't have to be as aware of what platform they're actually implementing on it. And then when they make the change, they can kick off the automated test and get through unit tests.

Bertrand Cardinal: Prior to deliver something, for sure.

Greg Lotko: Yeah. It makes your developers way more portable versus platform specific.

Daniel Newman: Yep. It seems like the goal of reducing manual tasks, of increasing automation, of making it more cross-platform friendly, it just seems like these are kind of the steps. And by the way, I don't know about you, obviously I know that I'm kind of the all of tech guy sometimes, and when we play our parts, we sit in our parts and you play the I'm the mainframer guy, I'm joking, but this doesn't sound that different than any other developer.

Greg Lotko: It's not. And I haven't been mainframe my whole life. It's what I'm known for now.

Daniel Newman: It's just your part.

Greg Lotko: But yes, I mean there's commonality of what-

Daniel Newman: I'm a Sundance kid.

Greg Lotko: Every developer does regardless of the platform. There's a process, and I don't mean not agile, I mean agile is a way of development, but there's a process of what we go through. The creativity, but wanting to of course assure that we're not introducing change that's unintended, right? And you want that to be a consistent experience.

Bertrand Cardinal: Yeah, the reflex, the thinking.

Greg Lotko: Yeah.

Bertrand Cardinal: Everything's the same between cross-platforms.

Greg Lotko: Absolutely.

Bertrand Cardinal: For sure.

Greg Lotko: Absolutely.

Daniel Newman: What about bringing developers in that were developers outside of the mainframe though? Does this kind of tools, does this advance the capabilities? Because obviously I don't know how much compatibility there is. I don't know if you can speak to that.

Bertrand Cardinal: We're starting same ID, there's a couple of ID on the market, but we try to concentrate cross-platform the same ID. We have a coverage, security coverage for sure. Cucumber, I'm not saying Sonocube. All those kind of tools are really the same between cross-platform. And that's our goal now.

Greg Lotko: The reality is it's here, it's evolving. The idea that you can take a developer that doesn't have to know the specifics of the platform, but can leverage the characteristics or seek out the qualities of the platform that they want it to be on, but use the same types of tools that they interact with on other platforms. And then that's just one aspect of opening up the platform, opening it up to a broader set of developers. But then of course, obviously if you think about the workloads, no workload sits in isolation. Of course you would want to have a developer that understands how it flows through.

Daniel Newman: That's really interesting. You would think there'd be a lot of innovation that would come from the diversity that you get.

Greg Lotko: Always.

Daniel Newman: Developer types that come from developing on other platforms outside of the mainframe, bring them in, they bring new ideas, they bring curiosity and of course the experience of the mainframe developers can give them the education and capabilities.

Greg Lotko: Yep.

Bertrand Cardinal: And same thing, we talk about developer, but AQ was also QA analysis. You have to use the same kind of tool because we don't know where do we need them. Do we need on the distributed side, do we need on mainframe or the whole solution? That's why we need to not forget them also. There's developer, there's AQ, there's analysis, bunch of role that needs to be let's say cross-platform regardless.

Daniel Newman: What do you think are some of the most important characteristics to make sure that in the end, that these application development platforms and the developer ecosystem that we have today, Bertrand, what do you think are some of the most important characteristics and capabilities and what are you personally doing to ensure this in your work to have success as we move forward, to speed things up to drive more automation, to bring more talent to the space? Are there some specific guidelines that you can offer?

Bertrand Cardinal: Yeah, we try to accomplish first make sure the knowledge is still there. For sure we really have an issue for that because a lot of people is going to retirement. After that, quality first. Quality first for sure, no matter what. And speed of development for sure. That's the main three topic. Speed, quality, and of course documentation to make sure that we are not losing those kind of knowledge.

Greg Lotko: You also want the characteristic of consistency, right? When you're looking across your development teams and want to be able to inspect, are we following good practices? If you have teams using a similar process across platforms, it makes it easier for you.

Bertrand Cardinal: For sure, for sure.

Greg Lotko: Yeah.

Bertrand Cardinal: And it's the language, not language of programming, but the link between manager that manage distributed team versus manager that does managing some mainframe team, let's say. Often the dialogue is not really fluid.

Greg Lotko: Mm-hmm.

Bertrand Cardinal: And when you have the same kind of process now it's easier to grow and make sure that we go to the right path.

Daniel Newman: Yeah, it's a really important topic I think sometimes, Greg, that it's easy to take for granted the importance of good QA management, of having the right cross-platform and the right set of developers that have time to value realized. And something I think that Bertrand brought up was also strong collaboration that exists between business and developers. Remember in the end, I mean that's why we're doing it is we're developing usually an application.

Greg Lotko: It's not code for code's sake, it's to perform a business function.

Bertrand Cardinal: Yeah, a business requirement for sure.

Greg Lotko: Right.

Bertrand Cardinal: Yep. If we have a unique language in IT, of course business will understand more instead of knowing bunch of different technology. If they understand the comment of each platform, it will be easier to deal with them for sure.

Daniel Newman: Well, I think this is one of those topics that as we sort of try to wrap this up and conclude it, Greg, we have to have the cross-platform. We need to be bringing new developers into the fold that can connect to the mainframe space, that can also bring a lot of innovation and new ideas. I like a lot what he talked about, like you said, when you talked about QA, he talked about making sure the applications are running and performing and of course bringing together the type of environment that people can really focus on building the most important apps to run the business, run their business, whatever it is.

Greg Lotko: And paired with that, it's consistency of processes and making sure that you're documenting or capturing the capability of things as you are developing them, that you're testing it and documenting and keeping it together going forward. A lot of good stuff.

Daniel Newman: Yeah, there is a lot of goodness in there. And you keep mentioning the documenting and I keep forgetting to say it, but yeah, he definitely reiterated that one, didn't he?

Greg Lotko: Absolutely. Absolutely.

Daniel Newman: All right. Bertrand, we want to thank you so much for joining us here on the Main Scoop, telling us a little bit about what you're doing at Desjardins and of course talking a little bit about overall how app-centric development can really help and can drive forward business and can encourage a community of developers and can encourage innovation.

Bertrand Cardinal: All right, thanks for having me.

Daniel Newman: Well, Greg, that was a good one. I'm going to go ahead and take us on out of here. Everyone hit that subscribe button, join us for all of our episodes of The Main Scoop. We appreciate you being part of our community, but for this show, it's time to say goodbye.

Greg Lotko: Thanks for joining us.

Daniel Newman: I knew you'd do that.

Greg Lotko: See you later. Bye-bye.