

EPISODE 868

[INTRODUCTION]

[00:00:00] JM: Creating a software company has never been easier. Software engineers are increasingly seeing entrepreneurship as a viable career path, but the path to becoming an independent software developer is not always clear. Most engineers spend some of their career working at a software company. Even an engineer who intends to build a company on their own someday can thrive within the right environment.

Lior Kanfi is a software engineer and the founder of Tikal. A company that he started around the height of the .com bubble in 1999. Lior's original vision for the company was to build a product around managing knowledge and people within a company. When the .com bubble burst in the year 2000, it became much harder to run a product-focused business. Companies were not buying lots of experimental software and investors were not aggressively funding new software companies. After the market collapsed, Lior shifted Tikal from a product-focused company into consulting in order to have a more reliable income stream.

Today, Tikal is a successful software consultancy based in Israel and the Bay Area. Lior has built this business over the last 20 years, and in today's episode he describes the engineers within Tikal as free radical software developers. Independent people who want to learn about new technologies and build experience interacting with clients.

Lior joins the show to talk about his 20-year journey building Tikal and the differences between engineers in the Bay Area and those in Israel. Lior also hosted the Fullstack Tech Radar day in Tel Aviv, which was a great conference that I attended and I hope to perhaps attend next year. I encourage anybody who is in Israel or who wants an excuse to travel to Israel to check out that conference, because it was a great multi-faceted array of different subjects and very technical content.

[SPONSOR MESSAGE]

[00:02:11] JM: You probably do not enjoy searching for a job. Engineers don't like sacrificing their time to do phone screens, and we don't like doing whiteboard problems and working on tedious take home projects. Everyone knows the software hiring process is not perfect. But what's the alternative? Triplebyte is the alternative.

Triplebyte is a platform for finding a great software job faster. Triplebyte works with 400+ tech companies, including Dropbox, Adobe, Coursera and Cruise Automation. Triplebyte improves the hiring process by saving you time and fast-tracking you to final interviews. At triplebyte.com/sedaily, you can start your process by taking a quiz, and after the quiz you get interviewed by Triplebyte if you pass that quiz. If you pass that interview, you make it straight to multiple onsite interviews. If you take a job, you get an additional \$1,000 signing bonus from Triplebyte because you use the link triplebyte.com/sedaily.

That \$1,000 is nice, but you might be making much more since those multiple onsite interviews would put you in a great position to potentially get multiple offers, and then you could figure out what your salary actually should be. Triplebyte does not look at candidate's backgrounds, like resumes and where they've worked and where they went to school. Triplebyte only cares about whether someone can code. So I'm a huge fan of that aspect of their model. This means that they work with lots of people from nontraditional and unusual backgrounds.

To get started, just go to triplebyte.com/sedaily and take a quiz to get started. There's very little risk and you might find yourself in a great position getting multiple onsite interviews from just one quiz and a Triplebyte interview. Go to triplebyte.com/sedaily to try it out.

Thank you to Triplebyte.

[INTERVIEW]

[00:04:31] JM: Lior Kanfi, welcome to Software Engineering Daily.

[00:04:34] LK: Thank you. It's a pleasure to be here.

[00:04:38] JM: Yes. You and I have gotten acquainted since you invited me to your conference, and we'll talk a little bit about that. But I want to start off by talking about Israel, because you are a software developer – Well, you run a software development consultancy in Israel, and I want to better understand how Israeli software developers operate. How do Israeli developers differ from the United States developers?

[00:05:07] LK: I think it's related to most of us that we grew up in an army sense of authority. So, basically, most of Israelis are working on small teams. They improvise a lot. They like to take initiatives. They love challenge, and sometimes they break the rules.

[00:05:31] JM: What kinds of startups get started in Israel?

[00:05:35] LK: It depends. At the start of Israel, one of the most known startup started around 1990. It's checkpoint, as you know, and it was a part of experiencing the army and the security sector is what used to be. Other than that, later on, if you heard about ICQ, which was a messenger. On the internet boom, a lot of startups used to do communication or messengers. Let's say after the .com boom, a lot of startups start to go to the enterprise and you can find either security, solutions, or software enterprises. If I fast-forward today, I think that we can find healthcare, a lot of healthcare. Security is a big sector. Developer tools is a new emerged sector in Israel.

[00:06:40] JM: Developer tools is a thing that I have seen the most. There's a bunch of companies that have created developer tools in Israel. Is there something about the Israeli culture that leads to creation of developer tools companies?

[00:06:55] LK: I think that most of us are in technology background, and the first thing that we see is our self as users, and we not business oriented or marketing oriented. So we see our self as a user. So, as you start working on a startup, you try to find tools which will help you. Later on, you try to boost it or use it as a product. I think it's also related to the open source culture. We like to use open source and we like to publish it. Suddenly, we shall see that developers has opportunity.

I remember on 2000 when I tried to raise some money around open source developer tools, most of them said that you should go to the enterprise market. But now, we see a very – Let's say [inaudible 00:07:57] or Bloom of developer tools.

[00:08:02] JM: Well, it's funny that used to be two separate ideas. The idea that you're selling to developers versus you're selling to the enterprise. Now it's the same thing. Now you sell to the enterprise by selling to the developers.

[00:08:17] LK: Right. This is the grass root approach. It's kind of the big win of open source. It used to be top-down marketing. Meaning, you try to persuade the manager. After persuading the manager, you needed to train the developers to use it. Now, you spread your words, you put your product in the internet and they are coming, and they are talking inside the organization and asking the manager to use the tool. So it's kind of a vice versa approach.

[00:08:51] JM: You started Tikal, your company, 20 years ago. What was your original vision for the company?

[00:08:59] LK: It's a nice thing that you put it. I'm just finishing as presentation to all of employees about the story of Tikal. Basically, it was a spin from [inaudible 00:09:10] company, which was called Tikal Software, and we kind of caught in the startup boom and we tried to create a startup. It was on knowledge management area. We had idea, which called it K Sphere. It was kind of a knowledge messenger. It was trying to find people instead of documents inside the enterprise.

So, we created a POC and we managed to raise \$1 million and we started developing. While developing, we decided to go with what was the most common software stack at that time. It was Oracle, Sun Solaris, and there was a big gap between the brochures of the marketing and the actual releases. So, we found a lot of bugs in the product. It was the Oracle application server. It was very edgy and we didn't be able to continue.

So we decided to go open source with J-Bus, which was it didn't have any release. It was releasing – I remember the first time that I used it. It was release of [inaudible 00:10:30], and then we used Linux and we started to use Apache, Apache Ant, and all kind of libraries. It's kind

of related to our goals, because it was – We liked to contribute to it. We like to use it. We like to report to the developer that they are using it.

So, after using it for let's say half a year, the 2001 9/11 came and all the startup where kind of freeze. So instead of closing the company, I said this is an opportunity. Let's try to sell the infrastructure that we developed in-house as a consultancy today, Israeli RND market. Because most of them didn't have budget at the time. So, we started to create integration between several tools. We had the build system. We had issue tracker. We had a version control. We also have an I.D. A distro of Eclipse. So we started to approach RND manager and tried to sell them and say, "Instead of using a Jbuilder [inaudible 00:11:47] whatever, IntelliJ, why not start using open source tools? And the only things that we will help you is to feel trust." So that's how we started and spin off the company from a product company to consultancy of professional service company.

[00:12:07] JM: It sounds like there was this issue with closed source software, specifically at least with the Oracle application server that you're using, where you would try to deploy your application to the software and you would encounter a bug that would reduce your ability to go to market or reduce your ability to actually deploy. Then you would be stuck. You couldn't go into the code and like solve the problem yourself or like find – Because it's like you just sent a binary, right? They're giving you a binary.

If your code throws the exception, you can't really look into it. No. I mean, I've never done this. I've never had an issue where I like use a piece of software and then there's an issue with the software and I go and fix the bug. But I can imagine that at the core of it, this is one of the issues with closed source software, is people can't fix it themselves.

[00:13:03] LK: Yeah, I have a story around it. While using the software, although we used all the open source stack. We stayed with Oracle Text, which was kind of Elasticsearch at that time, and we had a demo at Intel Sacramento where we presented the product. While presenting the product, we encounter a bug in Oracle Text, and I tried all their support. Between two days I had said, "Let's try to solve it and go to the class and try to make them see that it's working."

The support guy didn't know the product. So I needed to explain him, and it was very long cycle. At the end, when I arrived to Israel, back to Israel, that's the only way that I got some reply to my mail at that time. So it's kind of – You have a short cycle. You're working explicitly with the developers. When I had a bug in J-Bus, I was opening an issue and I could chat around the issue with the developer, with the core developer. At that time, I was familiar with them. I was chatting with them.

[00:14:21] JM: Tell me about the .com boom and bust from your point of view.

[00:14:29] LK: It's kind of – Most of us in Israel are relating it to the 9/11 issue, which the terror attack. It's kind of – Israel, there was a lot of VCs in Israel, but they were mostly funded from U.S. So, when the U.S. got into this big security issue and the big terror attack, it's kind of the flow of money was stopping.

So, it's kind of got us, everyone into some kind of freeze. A lot of companies were closed. You can see it on the flip-side, because before that, you would see that every two young people that's got out of the army and had an idea, the VCs were spreading money on them and they didn't know to manage things. So they start to buy hardware.

There was a lot of hardware garage sales at that time, at the end, after the bust. So, it's kind of from an inexperience market, you started to get into a more – I would call it more aligned or more rational market.

[00:15:52] JM: But what about you personally? I mean, you were a business person around this time, right? If I get the timeline right, about 20 years ago, that was 1999. This is when the market crashed, right?

[00:16:06] LK: Yeah. I was a business and a technology guy. Basically, I was visiting U.S. for several time and tried to secure some POCs. I had potential POC at Intel and I had some –

[00:16:22] JM: This is before the bust?

[00:16:23] LK: Yeah, before the bust. I had some potential leads in Israel too. At that time, the enterprise lifecycle was very – Sale cycle was very long. So you kind of try to persuade the investors to believe that you're going to secure some deals. But at the end – If I found myself without money at the end. Basically, I needed to change the approach and try to go on a self-sustained business model. That's why I used the professional service business model.

[00:17:05] JM: You basically shifted to becoming a consultancy, because in the 90s, or when the bust came, you had these potential clients that might have bought your software otherwise. But because the market was getting so cold, you needed to shift to a consulting model to build a business more consistently to have a sustainable flow of work. Because people are always willing to pay for a consulting.

[00:17:34] LK: Right. Basically, when there is a cold market, sometimes you go external, because you're kind of seeking for elastic. Meaning, you don't hire new employees yet, because you are not sure that you'll be able to hold them for a long time. So consultancy sometimes, it's a very short-term option. We were unique, because we were shouting or focusing on open source. So, instead of asking for RND manager to do capital, meaning invest in tools, it would invest in service.

So, it's kind of was a win-win situation, because we came with really find suite of tools that we used and he could get the boost. Instead of trusting in commercial closed source companies, he started to trust us as an evangelist of open source.

[SPONSOR MESSAGE]

[00:18:41] JM: DigitalOcean is a simple, developer friendly cloud platform. DigitalOcean is optimized to make managing and scaling applications easy with an intuitive API, multiple storage options, integrated firewalls, load balancers and more. With predictable pricing, and flexible configurations, and world-class customer support, you'll get access to all the infrastructure services you need to grow. DigitalOcean is simple. If you don't need the complexity of the complex cloud providers, try out DigitalOcean with their simple interface and their great customer support, plus they've got 2,000+ tutorials to help you stay up-to-date with

the latest open source software and languages and frameworks. You can get started on DigitalOcean for free at do.co/sedaily.

One thing that makes DigitalOcean special is they're really interested in long-term developer productivity, and I remember one particular example of this when I found a tutorial in DigitalOcean about how to get started on a different cloud provider. I thought that really stood for a sense of confidence, and an attention to just getting developers off the ground faster, and they've continued to do that with DigitalOcean today. All their services are easy to use and have simple interfaces.

Try it out at do.co/sedaily. That's the D-O.C-O/sedaily. You will get started for free with some free credits. Thanks to DigitalOcean for being a sponsor of Software Engineering Daily.

[INTERVIEW CONTINUED]

[00:20:42] JM: The idea of a software consultancy is almost too abstract. It's almost too broad, because there is a Accenture, which is arguably a software consultancy, where you hire a bunch of engineers from Accenture and they give you Accenture engineers and you can scale up your team that way. But then there're all these other models. There's ThoughtWorks. There's small consultancies you can find online.

Then more recently, you have this trend where companies that are building a product, often times an open source product, build what is essentially a consulting arm. So a company like Pivotal, for example, where they have a set of software solutions, Cloud Foundry, Greenplum, Pivotal Cloud or whatever. I mean, they have a bunch of different things. But then they also will parachute in a bunch of consultants that will help you deploy these solutions and maintain them.

So, even like Elastic, company like Elastic. You need consulting help to make the best use of Elasticsearch. Many people who are listening to this may wonder, "Wow! How would I need help with – Why would I need help with Elasticsearch? I just want to buy a search engine, internal search tool."

But if you think about a gigantic enterprise, it's a question of how do I index my data. How do I do permissioning? How do I do machine learning to build an even better indexing system? There's a whole lot of complex problems and you probably don't have in-house solution, in-house knowledge to help you with those problems.

Anyway, I'm just trying to paint a picture for the fact that the software consultancy is such an abstract concept, and I want to get your vision, because you started as a product company. You basically shifted at consulting, because you ran a cold market. Then your consulting business took off and here you are 20 years later. So, what is your current vision for Tikal?

[00:22:46] LK: Okay. Let's start with your original question about what's the business model for doing it. We export several business models till we used our current business model. But if we took a wider look, in professional service, you either have three types of service types, or vision. Either you are efficient. Meaning that it's better to take you because you are more efficient from your client. The second option is that you're experienced. Meaning, you've already done that and he's looking for your experience. At the end, you can find that you're kind of a brain surgeon, an expert. Meaning that you need someone specific to solve a very odd problem.

So as a consultancy, first decision that you need to take, where do you focus? At Tikal, we decided to focus on tier 2 and 3. Meaning, we don't do efficient work. We don't try to do code monkeys, QA or just calling us because we are more efficient or more cheaper or we don't use the offshore approach or whatever. Meaning that we will like be valued to our clients that they can get it. So we are approaching the market with experience and expertise options.

Second thing is how do you penetrate the market. So, basically, what we tried at the start is using the open source strategy and create some kind of integration between open source tools and try to sell subscription on top of that. We had some success around it. Meaning, we were able to sell a bunch of tools, similar – I don't know if you're familiar with Maven or other tools, and we sold subscription model to the RND managers.

But it kind of caught us in a situation that it wasn't enough sustainable. So we needed to work on a dual approach. Meaning, take some consultancy and try from the consultancy to build a product, or the suite of product, which was sold as a subscription model. So, it's kind of having a

dual vision inside your company and it's very hard to align both parties of the companies, because the people that are doing general consulting doesn't feel aligned with the general vision to create a suite of tools.

So, around 2008, I decided to focus only on expertise or expert. That's why we changed the vision from open source, which was already familiar with to the full-stack vision. Meaning, we'll give you a full cycle services around development and we'll have a lot of expertise and experience around them. So you will use us to boost your team, to help you boost your knowledge inside your organization. Basically, we decided to stop developing tools and only articulate our professional knowledge.

[00:26:05] JM: Was that a hard decision for you? So as somebody – Just from talking to you, I know you are interested in building products. You have ideas for products, and as software engineers or as entrepreneurs, we want to bring our creativity into the world. We want to these products. We want to make these inventions that are going to make the world operate more efficiently.

As you're doing consulting, I'm sure you see places all the time where you're like, "Wow! We could just build a tool that would solve that problem," and I bet a hundred other companies are having that problem too. But the distraction and the diversion that that would create to do two things at once, it can be really tough. So was it hard to cutoff that part of yourself or at least that part of your business at least for now and go straight down the road of consulting?

[00:27:07] LK: Yeah, it was hard, because once in a while, I'm trying to – Two years ago, I tried to open some kind of a lab with someone that is very entrepreneur, and we tried to develop some tools. Something like GitCoin, if you remember a thing like that. We called it –

[00:27:27] JM: Sure. Yeah.

[00:27:29] LK: Cold Monkey we called it. So, basically, it's kind of sometimes I find myself in asking why not gather all the information, gather all the macro-perspective that I have and develop something? But on the other hand, I'm very truly believe in focus in a company and in a specific vision. So, I believe that what our vision is to be a home for free radical developers.

Meaning, to be home for technologies. So, it's kind of something that take us a way from this vision. So, I only explore it on a personal basis. But don't align into the company.

[00:28:11] JM: That idea of the free radical software developer, you and I have talked about that a little bit, because as you know, I kind of have this view that there are a lot of forces that are pushing developers towards a commodification. That's what I call it. Just kind of, "I'm a React.js developer and I charge \$100 an hour, or I am an entry level React.js developer." There are these labels. These labels that we ascribe ourselves, basically, we have been taught that way, "I'm a devops engineer. I'm a database consultant."

What resonates with me about what you're doing is the idea of the free radical is the idea that we're capable of migrating from one domain to another. You can take your skills in React.js and almost all of those skills will map in some way or another to the world of data infrastructure. You want to go into data infrastructure and data engineering. You are a couple weekends of studying away from making that transition. That's the reality in today's world. So, I think your perspective on the free radical, by kind of capturing your perspective, or how would you define your free radical software idea?

[00:29:36] LK: I think that's the first thing that we need to explore is what is the workplace for us today? Because if you take my parents, for them, workplace was something that gave them honor and leaving. So, if they have a living and honor, they will stay for a long time.

When I got into workplaces, I try to succeed. Meaning it's a kind of a platform for success. So, that's why – Most of the Israeli high-tech was emerging on that area. After the next generation, said to the market, "It's not enough to succeed. I would like to put my values into the companies." Meaning, I won't go with companies which are doing evil to the world, or gambling, whatever my values are.

If I'm looking at my son, which is a YouTuber, with 3,000 subscribers, he says to me, "I'm in a workplace, as you are doing. I'm creative. I'm creating content. I adnate it. I market it. I negotiate with the advertisers." So I'm kind of a solo.

So, this shift in mindset, meaning that most of our needs are changing. But on the other hand, most of the workplace that we know still thinks on the old days. They are trying to persuade that you are kind of part of them. That it's better for you to be in that group, that you're going to conquer some kind of an objective together. If you are a developer, which has passionate for technology, you sometimes find yourself in a kind of non-challenge area inside the company.

So, if you are speaking about it with your manager, they say to you, "Why not you will develop, or your personal role will be a manager?" But you say that I like hands-on. I like to develop. You're getting in some kind of a plateau in the technology and you're asking us what to do. At the first acquaintance, you are going and you're trying to find another challenge. Suddenly, after you experience the same experience, you are asking yourself, "Okay. It's not stable. What do I do with that?"

You already have enough knowledge. You're already experienced. So you can go to the big companies. So you are exchanging from the mountain or the objective comparison. You're going to some kind of a cave or second war cave, because when a company, whatever big it is, it's already solved the problem. It's already have a market. So, you're kind of getting into their specific niche. Again, the tech challenge is – You don't beat them. So, it's kind of – You're looking into their workplace and they say, "Okay, what do I do?"

So, you either go and be in some kind of freelancer or you can do some job hopping, or you can be part of a community. That's what Tikal is offering for these developers, because they are solos. They can manage along. They can be freelancers. But if they like to be part of community, which share the same values and explores and share a lot of ideas, it's kind of more appealing. So that's what Tikal is trying to be.

[00:33:17] JM: So, these engineers that come and join Tikal, do you encourage them to work on side projects or to like start their own company eventually. What are their long-term visions?

[00:33:32] LK: Yeah. Everyone in Tikal has a roadmap, and the roadmap is personal development objective for – The annual personal objective, and he sits with leader and they explore together what will be his next move or next learning items or whatever he would like, or

his personal soft skill development, and they are creating objective [inaudible 00:34:01]. Together, the leader is helping to meet his objective as part of our business goals.

Basically, our vision is constructed on small steps, which we call roadmaps. At the end, we believe that each one of us can be a tech lead. That's what we are trying to make our people. So, on one hand, they're creating a recognition for themselves. On the other hand, if they are part of Tikal, they're creating recognition for Tikal.

[00:34:36] JM: Understood. So, have you read this book, *The Alliance*, by Reid Hoffman?

[00:34:40] LK: No. No. No, sorry.

[00:34:43] JM: There's a really good book called *The Alliance*. So, Reid Hoffman, the founder of LinkedIn, and *The Alliance* is about how a manager can establish a relationship with an employee that meets the employees' long-term goals. Oftentimes, in modern day software employment, engineers are employed at a company for 18 to 24 months, typically. In many cases, sometimes longer, but in many cases, these engineers they have a longer term vision for themselves. Whether it's like starting a business or becoming a chef or something. Engineers are creative people. They have long-term visions for how they want to turn that creativity into something that satisfies them. To what degree do you try to help work with your employees for them to fulfill their long-term vision for themselves that's agnostic of their time at Tikal?

[00:35:45] LK: Basically, what we hire is people that are passionate about technology. So, what's their long-term vision for my point of view is to be relevant in a market. To be in the radar. Meaning, all the time, to have a market value. To be the most experienced, the most rock star developer. To be a tech lead.

Basically, if someone would like to grow to be an entrepreneur, probably he will leave Tikal and try to be an entrepreneur. But most of people that stay in Tikal are people that believe in the developer mindset. Meaning, "I would like to stay a developer, a relevant developer, because I love – And I love to code, but I would like to be updated relevant and not stay behind the market. Because all the time I have a market value, which is to talk to you.

[SPONSOR MESSAGE]

[00:36:53] JM: As a software engineer, chances are you've crossed paths with MongoDB at some point. Whether you're building an app for millions of users or just figuring out a side business. As the most popular non-relational database, MongoDB is intuitive and incredibly easy for development teams to use.

Now with MongoDB Atlas, you can take advantage of MongoDB's flexible document data model as a fully automated cloud service. MongoDB Atlas handles all of the costly database operations and administration tasks that you'd rather not spend time on, like security, and high availability, and data recovery, and monitoring, and elastic scaling.

Try MongoDB Atlas today for free by going to mongodb.com/se to learn more. Go to mongodb.com/se and you can learn more about MongoDB Atlas as well as support Software Engineering Daily by checking out the new MongoDB Atlas serverless solution for MongoDB. That's mongodb.com/se.

Thank you to MongoDB for being a sponsor.

[INTERVIEW CONTINUED]

[00:38:16] JM: We met in person at this conference that you invited me to, the Fullstack Tech Radar Day. I enjoyed this conference, because it had a lot of different elements that were modern. When I say modern, I mean not kind of trapped in the paradigms of two years ago. Two years ago may seem like a short amount of time, but two years ago, is quite a long amount of time in the world of Software Engineering Daily. Describe to me the biggest changes, the recent changes to software engineering that you wanted to capture in your conference.

[00:38:56] LK: Okay. What we tried to do in the conference is to bring to the table our point of view of working with a lot of clients and try to bring up some kind of visualization of what we think are technologies that you need to start using. Technologies or topics, technology topics that you would like to keep using and technology topics that you would like to stop using. Meaning, that if you're using them, probably you have a technical depth.

We used the approach of the Tech Radar, which is a kind of thing that we use from ThoughtWorks that has the Tech Radar, but we modified it a little bit to our need using their open source license. What we tried achieve in this conference is to put on the table trends that we see. The first thing that we saw, which is also we spoke about it, is that we see evolution in the term of full-stack. Meaning, the original term of full-stack was around the softer layers. Meaning, that you can develop in the frontend and the backend, for instance.

But today, you see that a true backend developer needs to know Kubernetes, or Docker, whatever. So you kind of get into the depth of [inaudible 00:40:18]. So it's kind of changing to what we call full cycle developer. I know that you – We also invited Greg Burrell from Netflix, which is using this approach inside his edge team.

Basically, a developer needs to know all the development cycle and to be some kind of – Be able to be executer in all these lifecycle of development. The other trend that we saw is kind of – There is a lot of data and a lot of data pipelines and you start to see that you kind of need an orchestrator.

So, what we saw emerge is the Airflow kind of – Or Luigi kind of tools that's helped. If you're a data engineer, you kind of need to correlate and create dependencies between pipelines. So, Airflow and likewise tools helping you to achieve that. Other than that, it was around security. Meaning, security today is going to be the first.

You're not taking the security and doing a penetration test. You need to create a security culture, which is related to dev sec ops role. Meaning, at the start of the company, start of the development, you need to know how to keep secrets, how to be sure that no one will get them. How do you write secure code? All those kind of practical techniques needs to be injected inside to the start of the development cycle.

[00:41:58] JM: Were there any trends that came to mind during the conference that made you think like, "Maybe we should have included this trend or that trend, or maybe we should have resituated how we phrase this sort of thing." Was there any way you would refactor the conference in retrospect?

[00:42:18] LK: That's a good question. I didn't think about it yet, but let me do it real-time. First of all, the conference, a lot of people gave us very good feedback about the content. So, I basically believe that we kind of caught their attention.

But looking back, I think that automotive, all the things that's around things, meaning doing assembly lines with a lot of sensors and collecting all the data and to create an IoT or edge computing. I think that we didn't cover it enough. That's something that needs to be covered more recent.

[00:43:01] JM: Is that a trend that is actually making its way into implementation, or are people just kind of talking about that? To what degree are these data pipelines around real-time streaming data from automobiles or from IoT devices? To what degree is this a reality?

[00:43:18] LK: Let's see. We had some pilot project with an automotive company, which tried to help them to create a POC, which predict an accident between two cars. So it's kind of a POC that we used machine learning and GPS tracking and kind of send alert to the car driver to be careful when something happens. That's kind of – You can see it on the POC side.

On the other hand, we have a client that has a business around collecting a lot of sensors around assembly lines inside big factories and trying to bring a dashboard and trying to bring efficiency indicators to them to help them create more efficiency, to product their product more efficient. So, I find it as a kind of emerged in the recent – From my point of view. From the start of the iMeet.

[00:44:21] JM: What else have you learned in the last, let's say, 12 months, when you're working with these enterprises that engage with your consultancy. What are some newer changes that you're seeing in the clients and how were those experiences, those consulting engagements, reflected in how you crafted the content from the conference?

[00:44:50] LK: Okay. First of all, I think that we are aligned around it. Kubernetes is the standard de facto for orchestration. Basically, from Docker and Docker Swarm, which were the first offering around orchestration. You see that Kubernetes is the standard de facto.

Most of the companies are working with Kubernetes either as a [inaudible 00:45:12] or on-prem. So you see a lot of ecosystem around Kubernetes, and it's also extending Kubernetes. Meaning, the go language was catching and you see a lot of people from the DevOp world are using and extending Kubernetes using Go.

The other things in DevOp is the SRE role. Meaning, at the time, most of devops were most kind of ops guys or developers that working aside the infrastructure or using, creating pipelines or devops pipelines, CICD's pipelines. But now you see that because you are using a platform like Kubernetes, you need a lot of system services. So you need a developer, a true backend developer, which is around the infrastructure. So, a lot of companies are starting to define, in Israel, the SRE world, which is a developer with devop mindset, but is developing engineering task.

[00:46:20] JM: Are there any other downstream effects of Kubernetes that you've seen among enterprises that you deal with?

[00:46:28] LK: Yeah. Sometimes you – Because most of the people are using Kubernetes, it's not all the time the right tool for the right job. So you kind of see a very small problem and someone is taking a big hammer and using Kubernetes and hammer to work with it. But at the end, it's very small set up, which you can use Docker Compose or even Docker Swarm. So, sometimes you see it's overusing the technology, or kind of trying to think big before they are not agile enough. Meaning, I will start with Docker Swarm and go ahead to Kubernetes when I need it. So sometimes when you're seeing a standard, sometimes people are overusing it for all kind of purpose, because it's safe to use it or because they are passionate about using it.

[00:47:27] JM: To wrap up, I'd like to get your perspective on the global nature of software engineering. So, I attended your conference, the Full Stack Tech Radar Day in Tel Aviv, and then I went to KubeCon. Directly from Tel Aviv, I went to Spain. It's remarkable how much of a unifying force software engineering can be, because it's this common ground that we can all come to. It seems like the software that used in Tel Aviv is very similar to the software that's used in San Francisco, which is very similar to the software that's used in Europe.

We can all come to the table and we can have very impartial conversations around the best practices. How has the global market for software engineering? Since you're somebody who's been traveling between San Francisco and Tel Aviv for many, many years, what's the state of software engineering globalism? Is it any different than it was 20 years ago, or what are the dramatic shifts you're noticing?

[00:48:39] LK: I think there is a common problem around the decades, is talent shortage. Meaning, at the end, you find yourself as a software product, whatever, that you are very hard to hire a developer. Because you needed talented developer. You need experienced developers. There are a lot of programs that tries to get into accelerating the talent shortage.

But at the end, basically, for instance, at Silicon Valley, you see there is a talent shortage around job offering. Meaning, the big companies are doing some kind of – I don't know how to call it, but they are offering a big money for a lot of talented engineers and other startups find themselves that they can't compete with it.

So, the talent shortage brings to the table the ability, because there is standard. Because there is a lot of using open source tools. It's often the idea of remote work. Meaning, not to work on efficient way, to work as a distributed team.

Basically, what I see that we are able to offer our services to the Silicon Valley and help them work with Israel, with Israeli time, because they can't hire it in Silicon valley, because they have shortage. It's kind of how it was – I forgot the word. Forgot the article. Thomas from the New York Times, the world is flat. I forgot the – Never mind.

[00:50:20] JM: Yeah, I know who you're talking about.

[00:50:22] LK: Yeah. I forgot his name somehow. But, basically, what you see that if you are a software manager or a software businessman needs to find talent, you now can look in Europe, in Israel, all around the world, and you can work on a distributed team model. It brings you to the table the global talent. So you can compete on other ways.

So, basically, most of the developers are looking for flexibility. They don't like to commute around. I had a conversation today with someone that came to our office and he said that he's one hour and a half driving to work. So, even Google is seeing it and they are opening offices around Sunnyvale to help people commute less.

Basically, I think that two forces. One, that there is a standard, as you said. You could talk to Tel Aviv on Kubernetes and you can talk in Spain on KubeCon and you can have the same ideas. On the other hand, the world is becoming flat. Meaning, you can approach each one and you can assemble teams using the communication tools, the distributed tools together to create software development team.

[00:51:49] JM: It is quite beautiful. I am really enjoying remote work. I've collaborated with developers in four or five different countries, at least, and it's just very fun. I really like the cross-cultural melting pot that online software development is becoming.

[00:52:08] LK: The only thing I would like to add, the most challenging issue with software developers is soft skills. Meaning, because of what you said, the melting of cultures, you need to develop your soft skill more frequent. So you can't stay with a computer. You need to communicate a lot.

[00:52:28] JM: Completely agree. Soft skills, as they apply to Slack messages and video calls and email responsiveness, these things are not straightforward and they take some experience. So, people will have to develop these in the coming years.

Lior, thank you for coming on the show. Thanks for putting on an excellent conference and inviting me. I endorse the Full Stack Tech Radar Day. At least I can endorse the first one. It was great. I had a great time at your conference, and I hope that this episode illuminates the conference for more other people. Because, certainly, if you're looking for an excuse to go to Tel Aviv, which is a beautiful city, your conference is a great one.

[00:53:12] LK: Great. Thank you for hosting me. As I said on the conference, you are my favorite podcaster. So it's great to be here.

[00:53:22] JM: Thank you, Lior.

[END OF INTERVIEW]

[00:53:27] JM: GoCD is a continuous delivery tool from ThoughtWorks. If you have heard about continuous delivery, but you don't know what it looks like in action, try the GoCD Test Drive at gocd.org/sedaily. GoCD's Test Drive will set up example pipelines for you to see how GoCD manages your continuous delivery workflows. Visualize your deployment pipelines and understand which tests are passing in which tests are failing. Continuous delivery helps you release your software faster and more reliably. Check out GoCD by going to gocd.org/sedaily and try out GoCD to get continuous delivery for your next project.

[END]