**Part 1 - Your Startup**

1. What is Startup p's main product/service?

02:10 Startup 12 was born in 2012, the idea. Our service is accounting. We automated what many accountants do Today, only manually. To give you an idea, what an accountant does in a week, we do it with a click. Now our focus is to streamline. Our product which is to automate the service, is accounting. We carry out traditional accounting services, but we came with a lot of technology. We automated what many accountants do Today manually, and now they do it with a click. The accountant already has a balance in real-time. Tax calculation, the accountant enters government websites and such. We have a robot that does this. The human only enters when the robot is unable to calculate the tax. So, our product is automated accounting for companies. And the customer is small customers, small companies. So are those bit companies that issue 1, 2 to 10 invoices per month to receive his salary, or provides service paid before u m minimum wage (R $ 1017.00) and now pay R$ 99.00.

4:13 We can do this because we automated a lot. To give you a sense of what technology has delivered to this accounting market, the ratio of an additional quantity and customer that he can serve is 1 to 30 companies that perform very well with traditional accounting. Speeding up the balance is 1 to 400. So, 1 accountant from us manages to serve 400 companies because we provide a lot of technology.

2. When did your startup establish itself?

4:45 AM Startup 12 is eight years old. We were born with a client. We joined the accounting office, so we were born with a paying customer, with 30 customers, to validate the business model. But it took a long time for us to sell the business, the product, our service well. Because we didn't even know what it was. We were in doubt if we are software? Are we an accounting company? In the beginning, we were in doubt. And it was difficult for people to sell. And then, as time went by, other online accounting players emerged. Startup 12 was the first. And then, it started to talk about it in Brazil. Many people have begun to know what accounting was online, and we c omeçou to settle from the second year, 2015, 2016. That was where we started to sell well and have the predictability of Growth. Today we are in a phase of complete Growth , thus exponential Growth. We received a contribution last year, the company doubled in size for this one now. And the expectation is to grow at least 100% again.

3. Where is your startup located?

08:29 Today, we pay attention to the whole of Brazil. Our physical headquarters. It was in Salvador. I say it was because of the pandemic gent and is hiring across the country now. Our headquarters is Salvador. We also have an office in São Paulo. But, both headquarters Today are not occupied. We are currently working 100% remotely, and we believe that this is a reality and should not change. We are the only online accounting that serves all of Brazil.

4. What is your role?

9:19 AM I am CTO. I take care of the tech technician team.

5. What kind of ecosystem are you currently working on?

09:19 Startup 12 is a Fintech.

6. How many employees do you currently have?

09:26 We have 180 people in the company. Of these 180, 20 are engineers and software engineers with the product team because the design and other profiles come in. We have 30. We doubled in the number of customers, and in the number of employees we triple or. We spent 7 years is 5. When we received funding, we quadrupled the team, and there were 20 participants.

7. What is the composition of your team?

11:04 The software engineering guys, I have a team divided into three profiles: people who take care of the front-end, which is the people who take care of the face of the applications, who develops what the customer sees; there is a group that takes care of the back-end, it is a group that takes care of the other part that is behind the applications, so that takes care of the database, dates, APIs that we develop, we develop Crowley. So, front-end, back-end, and DevOps, which is the group that takes care of the infrastructure for the applications.

8. Do you proactively participate in product life cycles, such as production, testing, and launch? S and yes, to what extent?

11:49 I participate in everything. I started as a designer in my career 20 years ago. So, I stick with the team every day. I'm there for what I need, back and front.

13:06 I got out of tactics. I'm more in the strategy. We did it since I built the team in a year. And we have Squad systems, so we have teams that take care of certain products. And the teams are autonomous, they have the leaders, and these leaders report to me. We get together for a week to follow up. And your team has a problem, an impediment that the leader cannot solve, it escalates for me, and I have to solve it. Being myself the person I resolve, or else seek help. I take care more of the management of the team's people and the technical development of the group. I always bring training from outside, consulting, mentoring, to raise the bar for the team.

**Part 2 - Software Engineering Practices**

1. What software development practices, tools are you using? Briefly describe how?

17:32 Speaking a bit of practice. We use Agile. I use Kankan here, with a hint of Scrum, Scrumban. I use a mix of what I find exciting, and that makes sense to us. We use it as a tool for managing activities. Even the Kanban board view, we use Jira. I'll give you the link here from Atlassian (https://www.atlassian.com/br/software/jira). Management of the cards, activities, this tool. Methodology, a lot of Kanban, a little bit of Scrum, a few frameworks. We have a solid culture of agility. So, we deliver software that works every week. There is something every week. We have a reliable delivery. Per day, I don't know, an estimate of 50, 60 deploys per day. It is always incremental. I do not accumulate development. The group's focus is to understand the tools well and understand a little of the software-based, to build software and reusable b on. Software with a lot of testing. We automate. Today, part of our development cycle has implemented automated testing. We have a solid test culture. I have two customers—an end customer who pays Startup 12. And I have an internal customer. My technology also serves the Startup 12 operation team, accountants, customer service, marketing. The demands always come from them, in real pain. We prioritize requests according to volume. So Imagine who has many customers who need to issue a ticket. We map the volume of demands. In Jira, we create activities in the backlog and priority according to the importance of requests and criticality of the activity. That done, we have to do.

Our PO (Product Owner), they pass on what needs to be specified. In this specification, the responsibility is of the PO / PM. But who participates is the whole team, we involve in understanding what needs to be done, a multidisciplinary team, the entire squad, which has a designer, back-end and front-end, usually someone from the business, which is typically the one who requested it, and the PM / PO. It is this team that specifies what has to be done. From there, technical doubts arise, if it is possible, if not possible. Estimates are already out, obviously relatively crude. But we already know that one thing takes longer than another. And we already decide which way to go.

2. What are the most critical quality attributes (UX, performance, security, reusability) for your current products?

23:18   And this is the specification phase, the refinement of a card. And the artifact is a card or an app with several cards, an app composed of a group of cards, activities, features that need to be delivered for everything to be complete. And then, these sets of refined cards, specified together with the team, are placed on the board and go to To-do. They pass on to delivery. This indicates that they can be developed. If you need a design activity, we use a design sprint to involve the client in the process. Within a week, a navigable prototype comes out validated with the customer. The first step is to discover the problem; the second is prototyping; each step is a day—the third validation with a group of users, a maximum of five. On the fourth day, the structuring of this artifact according to the feedbacks. On the last day delivery. And then, we already have user flow screens to become cards for software engineering. We call this process discovery, which is the discovery of the artifact of discovering what needs to be done. So the discovery artifact can be screens and user flows, anyway. And then enter the delivery. And the software engineering team takes it to develop. And that stays in the cycle. We delivered the product team, the UX team, and the design team to evaluate the usage with the customer. We have tools to collect usage metrics and feedbacks through tickets that they create from demands. We consider, and that becomes a cycle where we are constantly evolving. Are there any new discoveries? Get in the flow, design sprint, go back to engineering, improve, go up to production. And this cycle remains until we decide that the feature is mature.

3. What testing practices do you adopt to deal with and check your product/use quality rice Here, we have a solid test culture. Nothing goes up for production, nothing exaggeration, because not everything needs to be tested. Nothing that is the core that needs to be tested is the heart of the feature. Nothing goes up to do production without testing. We have a code review policy. So, all code is reviewed by the team itself. So, one of the stages of our Kanban is review. The coder, the engineer, does the coding, opens the pull-request for the team, the team there, revises the code, or comments, improves. And no pull request makes sense to be tested. That is not tested. It is declined by the team itself. We have test coverage, I would say around 80%.

4. How much did you invest in testing activities?

27:38 We don't have a tester. What are the phases of the test here? First, the development itself has automated tests. So we do unit tests at the code level, load test, integration test. We implement testing. We have a test environment for homologation and which tests as a user ... we also have end-to-end testing, a simulation of a user test performed by a machine. In addition to the real user testing, the robot simulates the user and does the test. And after that, it goes up to the homologation environment. And then, who tests is PM / PO. Tests as a valid user, approved and such, we send it to the end customer.

5. How do you document your product at different stages of development and testing?

29:11 documentation. Ready. We don't create formal documentation. Our documentation is the API documentation. We document the API using the tool, one of which is Swegar (https://swagger.io/). We use StoreBooks (https: / /storybook.Startup 12.com.br/) to document front-end components. So, there is documentation, the documentation that we put on the card. So, you already have the specification documentation, and what is developed is documented as a worklog. Then the developers log into the activity, which was done in the activity. This is the documentation. And when we need to consult something, go to Jira, and the documentation is in that activity. The documentation is in the activity itself.

**Part 3 - Technical and Dynamic Debt**

32:04 First, let me give you a general overview of technical debt. For me, there is no software without technical debt. Proof of this is that Windows comes out with a blue screen still there. It is natural. For me, it is a natural process. The biggest question for me is: during the delivery process, how much technical debt we accept to buy. So, sometimes there are technical debts that we can't buy, and there are others that we can buy. So it's to calculate how much we are willing to have, to risk.

33:12 Both moments were fundamental. Because at the beginning of the startup, we had no validation at all. And we needed to deliver software soon, to validate it soon, to know if the business was viable or not. So, you end up accepting a lot more technical debts than the goodwill we are in Today. So much so that we have the first software that we developed. We have about six applications here. The first app that we developed until Today we have technical debt, but it took us here. So even Today, she has a problem. We didn't implement tests in the beginning. We wanted to deliver. It went up with a bug. The customer was testing. There was no one of ours to try. Yes, the moment allowed us to do that and asked us to do it. Because the manager was there to validate the business model. From the moment that we validated the business model and that the product was accepted by the market, it is already running. The customer base begins to grow, so we can start paying technical debts, paying those debts. Because now we are able, in a structured way, to pay these debts. Then, enter our backlog, a slot for technical debts.

1. How aware are you of TD in your startup?

34:37 How much are you aware of. Total. 100% aware. So, each squad knows the technical debts it has. When we leave a technical debt, we create a card and document that that debt exists. Someday, at some point, we will prioritize and resolve it.

3. How do you deal with DT? Do you ignore TD? Do you accept and manage TD? Do you avoid TD?

35:25 How do you cope? In a natural way. I do not ignore, accept and manage. How do we manage? Creating the cards. So, all technical debts, or a good part of them, are documented on our board. I do not avoid it. I stop to analyze how harmful that technical debt is. If it is something that impacts too much, we do not go up to production. We do not go up without resolving that debt. How we work with compliance. An error of the people can have excellent collateral damage. We work with customer taxes, and we assume all fines for our mistake. So there are debts that we can buy, others not.

2. What is your perception of TD?

34:59 This is my perception, just as I told you. It is highly natural. It is part of the process. For me, the main answer is how much debt the moment and the company can buy.

4. How did you deal with TD in the early stages compared to now that you are in the growth phase?

36:23 In the beginning, we had a lot of debt. We launched a new application with 15 days to 20 days, and HR (Human Resources). And this application has everything but technical debt. Pro va is that we invested a lot in testing. It supported a lot of software quality. And he also invested a lot in the purchase of debts. We raised the application to production with few debts. And the proof of this was reflected in the very few bugs reported in a month of boiling operation.

37:42 The main point for me there was the moment of the company. We could wait for the launch. We could actually invest in the quality of the software. We were in a situation where we were able to buy more term. For example, it is a product of ours, which we could launch later. Of course, we have a deadline. But we bought from investors, that we need more time, that we needed more months to improve the quality. It depends on the time, the product, the company, and it depends a lot on your software quality practices. And in my view, mainly testing and code review.

39:03 In the beginning, we were not proactive in documenting and managing. Things broke out for us when the client reported later, for example. In the front, we had no observability of things. We did not have tools that would automatically and proactively inform us of the problems that were happening in production. There was no monitoring. Today, I have monitored. If the customer lands on a 404 page, anything, we get warned on Slack, receive an email. So, we can proactively anticipate problems. We have observability. I have a board with several indicators that should be zeroed. If an activity appears, a number 1 appears on that board means that there is a problem. Before the client reports, we already know.

5. Did you change the startup's focus during the transition from the initial phases to the growth phases? Did any of the situations occur?

The. Zoom-in. Has a single characteristic of a product become the complete product itself? - If so, explain the role of DT?

B. Zoom-out. Has the entire product become a single feature of a much larger product, mainly because the original product is insufficient to meet the customer's needs? - If so, explain the role of DT?

ç. Customer segment. When trying to solve the right problem, did your startup discover a different customer segment than originally planned? - If so, explain the role of DT?

d. Customer need. Has your startup realized that the problem you are trying to solve is not very important for customers and has discovered other related problems that are more important? - If so, explain the role of DT?

And. Platform pivot. Has your app become your support platform or vice versa? - If so, explain the role of DT?

f. Business Architecture. Did your startup change the business architecture, for example, aiming at low volume, high margin, instead of focusing on the mass market? - If so, explain the role of DT?

g. Capture value. Has your startup changed the way/method of capturing value (monetizing)? - If so, explain the role of DT?

H. Growth engine. Has your startup made significant changes to its growth strategy to seek rapid and more profitable Growth? - If so, explain the role of DT?

i. Can al pivot. Has your startup identified a more effective way to reach your customers than the previous one? - If so, explain the role of DT?

j. Technology Pivot. Your startup delivered the same solution using completely different technology - If so, explain the DT function?

41:45 We have remained with the same focus since the beginning of the company. However, the focus was increasing. I like a metaphor very much: "don't try to boil the ocean, first get a pot of water, if you can increase the size of the pot, move to a basin, to something bigger until you boil the ocean." And sometimes you don't need to boil the whole ocean, sometimes a part of it is enough. We started with this profile and only served these types of companies. Then we began to attend, and companies with an employee, companies with more than five partners, companies that were not National Simple Companies were Presumed Profit. Then, we changed the service later. We started to attend Commerce. It left Salvador, went to São Paulo, and expanded. We did not pivot, but we increased the scope of our reach. The market has guided the expansion of the software.

6. How would you explain DT's role in your Startup pivot scenarios in one or two sentences?

44:37 I would explain it this way. For me, it is essential to buy technical debts. But it is critical also to have a roadmap to remedy technical debts. As I said, for me, any software will have technical debt. But it has to be a debt that the team is aware that needs to be paid someday. Do you know payment in installments? We have to pay, even if it is in installments.

47:13 And you need to put technical debts in the backlog. And we have a time slot dedicated during the week for paying technical debts.

57:20 When someone enters the company, there is a company on-board process and technical aspects. Besides, it has documentation of many team practices on a Wiki. The knowledge is there on the Wiki, and it is powered by the team. We also have a solid culture of weekly meetings. Two meetings are for passing on knowledge: an engineering talk and one from the company. A new person in the company participates in the negotiations from other areas as well. And we have something called Tech Hode, it is a meeting to present news that someone is working on, that some squad is working on. So, everything is shared. It is a dynamic of passing on very lively and organic knowledge. The team even passes on the inside and feeds on it. We have this space within the working hours for sharing knowledge. The teams are self-managing. I needed to spend a year right next to the team to make the team self-managing. Because we had a team of five people and went to 20. We formed sub-teams within that team. And then, I needed to form leaders, and I needed to make the culture of that five initially become 20. The culture of Startup 12 became the culture of 20. So, there was a process.