



# Bits & Bytes Association

## Newspaper

October 2021

---

### What is imposter syndrome?

Imposter syndrome is the belief that you are not as competent as others perceive you to be. In simpler terms, imposter syndrome is the perception you give yourself as a “phony”.



The feeling of doubting your own abilities and accomplishments.

The thoughts in your head stating that “you don’t deserve this” and “I’m not really good at what I do”. This is a common occurrence in information technology for many reasons. Being in a role

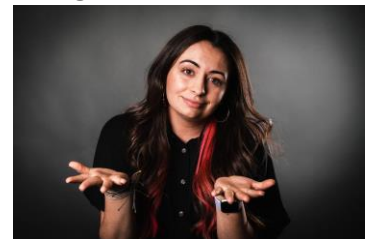
you feel that you shouldn’t be in or that you’re not qualified for. Being taken from a place of comfort to an area of discomfort.

In the tech industry, an informal study showed that 58% of tech employees from companies like Facebook, Amazon, Apple, and Microsoft feel like frauds. It just often happens in technology and can happen in different stages of your career.

With imposter syndrome there also comes a time where you’ll feel humble. You try to do better and prove to everyone else that you can do it. You’re also trying to prove to yourself that you can do the duties of that role. Trying not to be that imposter.

It can be a bad thing if you start to beat yourself up about it. You just need to say to yourself that you can do better. It all comes

down to how you manage those feelings and thoughts.



In this field you should always be pushing yourself to do better and knowing that you can/will do better.

Never feel like you’re not good enough.



This feeling will push you forward and make you work harder. Prove to yourself that you can do that role of your dreams.



# Bits & Bytes Association

## Newspaper

October 2021

---

### Tips for a software engineer

You must master a lot of different things and take a lot of different skills to be an amazing software engineer. Coding, different technologies, frameworks, code base architectures and delivery of all of this combined as a high-quality product. You also need to know how the products work and how the business works. If there's issues, you need to know how to debug a situation. Software engineers need to solve problems where you've probably never solved before. There isn't a way to prepare for it.

When you start to feel like an imposter, remember that everybody has their own beginning. When you look at other people, it seems like they've been doing it forever and you tell yourself that you'll never get to that level.



Everyone started from somewhere like how a baby had a day 1 of walking. You slowly gain the skills as time progresses. Whatever state you want to get to can be achievable with slow progress.

You can also check in with someone to talk about your feelings. Everyone is their harshest critic. Talking yourself down is one of those things.



Talk to a friend, trusted colleague, or manager. Someone who understands your work. Talking about

your problems makes the feelings a real thing. It makes you face your problem.

Figuring out your fear, and planning can also help you. Once you identify the fear you can address it. For example, not knowing a technology as well as others on a team.



I'm sure if you asked a team member they might point you to some resources, tell you about a project they did to help them get to where they are. If you feel like you're going to get fired, then talk to your manager about your feelings. Saying that you are not at the same level as everyone else. Hopefully they will reassure you on things you are doing well,



# Bits & Bytes Association

## Newspaper

October 2021

---

not doing well and address all the things in between.

### **Tips for a network engineer**

Being the new kid on the block can be scary and make you feel that you don't belong. When you're first entering a job, everyone around you will technically know more than you.



With technology you must always stay current to be relevant. This aspect of it can be difficult to manage.

A tip would be to constantly review the notes you've taken for certifications/learned from school. You don't want to forget what you already know. Making sure you remember all the things

you were taught. You also must remember that you won't know everything. Googling is a tool for reviewing what you already know about. Giving you reassurance that you need to perform. You have to rely on your ability to figure stuff out.

Remember that it's a normal feeling to have. Look back a year and you realize how much you've learned. The more difficult exams get, it never comes down to what you know. It comes down to how you apply what you know. You should always be learning and keep going. You can also join a bunch of user groups and see how others troubleshoot. Just stay involved.

### **Quick tips**

- Write down your projects, contributions, and accomplishments
- Networking at work and events, asking a lot of questions from other professionals in your field
- Do a pet project that you are invested in
- Mastermind groups
- Public speaking



# Bits & Bytes Association

# Newspaper

October 2021

---

## **Industry Event: Question and answer with Stephan Pawulski**

The idea behind industry events is where we get professionals in the field to talk to students. This article will be a quick rundown of what was discussed. Today we got the opportunity to talk to Stephen a previous bba president and talk about his experiences as a dev op developer.

## **Q: Who is Stephan Pawulski?**

Technology was his second career choice and came back to school at age 30 into the bit program. Like most he wanted the true college experience. He currently works at lotlinx and was taken on during his co op term. Learned a lot of things from the

ground up and has become an intermediate dev ops developer scratching senior level.

## **Q: What is dev ops?**

HUGE QUESTION. Its not a job its an ethos. The code we write needs to be translated to production. Code files turn into functional applications. Dev ops focuses on automation and tries to automate tasks throughout the application life cycle. Making it easy to integrate and test code. Once the requirements are met, it gets pushed to production. Its also about maintaining application. Continuous integration, continuous deployment, and continuous monitoring.

Its also about project planning. Its an iteration of the agile methodology. It takes the iterative

planning process and combines it with CICD.

## **Q: What are some tools used on your dev ops tool chain?**

Scripting languages are used rather than compiled programs to do their work. Picking the right tools is key. They are ruby focused but some more notable technologies are python and bash. There's also configuration management. They currently use Kubernetes on their journey. It's the new kid on the block.

They are migrating out of the one-to-one server topology. Where you have api/app and build out a cloud instance or server.

What they use for automated provisions are called terraforms. Its an industry standard in dev ops. It allows you to



# Bits & Bytes Association

# Newspaper

October 2021

---

provision things like instances, load balancers, and cloud resources with code. Infrastructure's code is the incentive behind all of this.

Chef is commonly used for the at Lotlinx. If you're writing cookbooks, you'll be writing ruby code. Its installing things to get the application to get to the state it wants to be in. If you're installing a webserver, you'll be writing a cookbook to automate that.

CICD pipelines used is Jenkins and new tech used is integrated with their Kubernetes clusters. They use GitLab. Stephen worked on the migration and with multinational companies. All of it is free doing all the deployment and CICD. Its an accomplishment he is proud of.

Advice he gives is to set up your own GitLab instances

and get a "dockerized" application going if you have cloud server like AWS or GCP.

For monitoring they use Azinga where you write checks with pager duty integrations. There's also Prometheus but its kind of hard to get into.

## **Time as a student**

Points of frustration where we think they're not teaching us the right things. The technology that they're teaching is old.

This industry is about learning and if there's one thing to take out of school, it's to learn how to learn.

Everything we need to know about programming is in term 2 of the BIT program. The hard part is how to integrate how things work all together.

## **Q: What is the work culture you've dealt with in the industry?**

When Stephan started at lotlinx they were on the tail end of the scrappy start up faze. They were very aware of burn out and making sure the developers were happy.

The reasons to using Ruby is because its enjoyable to work with. Easy to develop with and kind of fun.

He also had managers that were very accommodating for his time.

Advice for being a professional. Be an awesome co worker and go out of your way to get along with your peers. Do a good job but not everyone is fun to work with.





# Bits & Bytes Association

## Newspaper

October 2021

---

### **Q: Are your current soft skills developed something that happened over time?**

Stephan has always been good with people but like most there's always things that can be worked on.

He wanted a pay increase and he interviewed with other companies. Interviewing is a skill that can only be acquired by doing them. If you want to expand and grow into a position, it can take some time. You'll probably fail a couple times, but you'll eventually get one if you keep going at it.

### **Q: Is dev ops a 24/7 type of job?**

If he attends the first standup meeting but can pick any 8 hours of the day.

They do have an on-call rotation and are responsible for the infrastructure running smoothly. Three people on rotation where it lasts for seven days. Every third week he's on call 24 hours a week which is a component of the job.

Any good job won't ask you to be on call permanently. If there's no break from this, then it can be a red flag from an employer.

### **Q: Interviewing in this industry. How do you prepare for an interview as a junior?**

Stephan had rapport before interviewing at lotlinux and never actually had to sit through the "are you a human being" conversations.

He did have a take home project during term 3 of BIT and it was the time where he had to work with his first API call. He worked with an API dealing with artificial intelligence picture editing. Had to make it work the way they wanted using the Lotlinux technology at the time. You're not going to know every technology for jobs you'll be applying for.

He also interviewed for SportsNet. They had a very technical fun of the interview where he was given a cloud instance. He had to use ansible to provision a mini-Kubernetes cluster and server a web page that said the name the node it was running on.

He had to learn the whole ansible technology to complete the assignment. He had to work 8 hours a night on top of a job to



# Bits & Bytes Association

## Newspaper

*October 2021*

---

learn everything he needed to know to implement it. It was a valuable job process, even if he didn't get a job offer but he considered it part of the learning process.

### **Q: What does the company lotlinx do?**

They do business in the United States and are an advertising tech company. They find people who want to buy cars. All based on google ad profiles and 3<sup>rd</sup> party tracking cookies. Companies who do business with lotlinx get this information from them.

The job is composed of showing ads to the right people. They crawl every car dealer in the United States every 2 days. Looking at inventory, like every car color, for example. Keeping this information in a database.

### **Q: Did you find mentors or was it outside learning?**

The team he's on is the infrastructure team and composed of two people. Each person specializes in different things and divide work that way. VP of engineering manages him and every two weeks they have a work planning meeting on what's going to be worked on.

Day to day they use Jira as the ticketing software. Lots of freedom without a project manager. Usually, the first half of the day is the most productive for Stephan. Drinking coffee and no distractions. More exploratory things are done in the afternoon.

There's planned and unplanned work. If something breaks, then it can either be a quick fix or takes a little longer.

### **Q: How do you go about launching code into production? What's the next step from going from hello world to a production server?**

Learn how to containerize applications and run/build them. There can be challenges like networking layers. If you can understand docker and write a working application in there, It gives you a leg up on competition around you.

It's the industry standard now a to have things run in containers. Take this container and push it into a repo. Docker has a repo along with GitLab having its own project repo. Recommended because its all free.



# Bits & Bytes Association

## Newspaper

October 2021

---

There's also the concept of tagging containers like versions and commit hashes. A popular tag used is "latest" and untag the previous versions.

Pick a deployment tool and make sure that it has a docker host running on it. It's a cool way to use configuration automation tools like ansible. Ansible is a lot easier to set up than Chef. Using it to install docker and have the docker host pull down the repo host where its hosted and having it run the container. In a nutshell this is what deploying is.

### **Q: How does collaborative coding work from home?**

Git was built for collaborative code. Used to support an unlimited number of developers working on the same code

base. There are some processes that go along with this concept. People now a days have a micro service topology where they write bits of code that perform one thing. Writing an API that will just go to a database for example and get certain data.

This allows you to do is shrink code bases and making the point of collapse/breakage smaller. In a monolithic application, if one thing breaks then the whole application won't work. If a micro service is broken, then anything that isn't attach to it should work.

Git has the concept of branches, releases, and tags. People will create a branch from main for a future branch and create a merge request where a senior developer will review the code and approve the merges.

There's also the concept where people always deploy to main branch but hide newer features behind flags. You can push new code to production and the flags won't be released until you set it.

### **Q: What do you guys use for chatops?**

They use slack for day-to-day communication. Email is only used for formal purposes only.

Stephan has written different chat bots that integrate tools with slack. Making peoples lives easier.

One example of his work is making a bot with Jira where he can create a ticket in slack by messaging a bot and it will create a link to the ticket. We create chat bots to solve certain use cases with employers.





# Bits & Bytes Association

# Newspaper

October 2021

---

Commenting code is important and reading other peoples code is an even more important skill to learn in the industry. If you work at a company that moves quickly and is very agile, there won't be comments in the code. Don't expect comments as a gold standard.

## **Q: Any advice you would give to students?**

He got frustrated a lot thinking he was learning the wrong things.

Take your time to learn git and push all your code to your own repo. Get in the habit of getting to know the workflows. It's a good practice to back things ups.

Make friends while in college so you can ask questions to these people. Make sure you're getting paid properly and talking

about certain wage expectations. Now a days since everything is remote, companies are competing with Canada wide wages rather than just Winnipeg alone.

## **Q: How do you go about getting a raise?**

Having a good LinkedIn is part of playing the game. Being active on there and keeping your skills updated. Earlier on you can reach out to recruiters. The good recruiters you'll know right away by how professional they are.

Good recruiters will set you up for some insight on what to study or even what to wear. He also recommends to use hired.com and apply for jobs directly on there.

## **Link to Recording**

You can find the link to the recording in our discord channel under guest speakers. There are also previous industry events we've hosted you can check out!