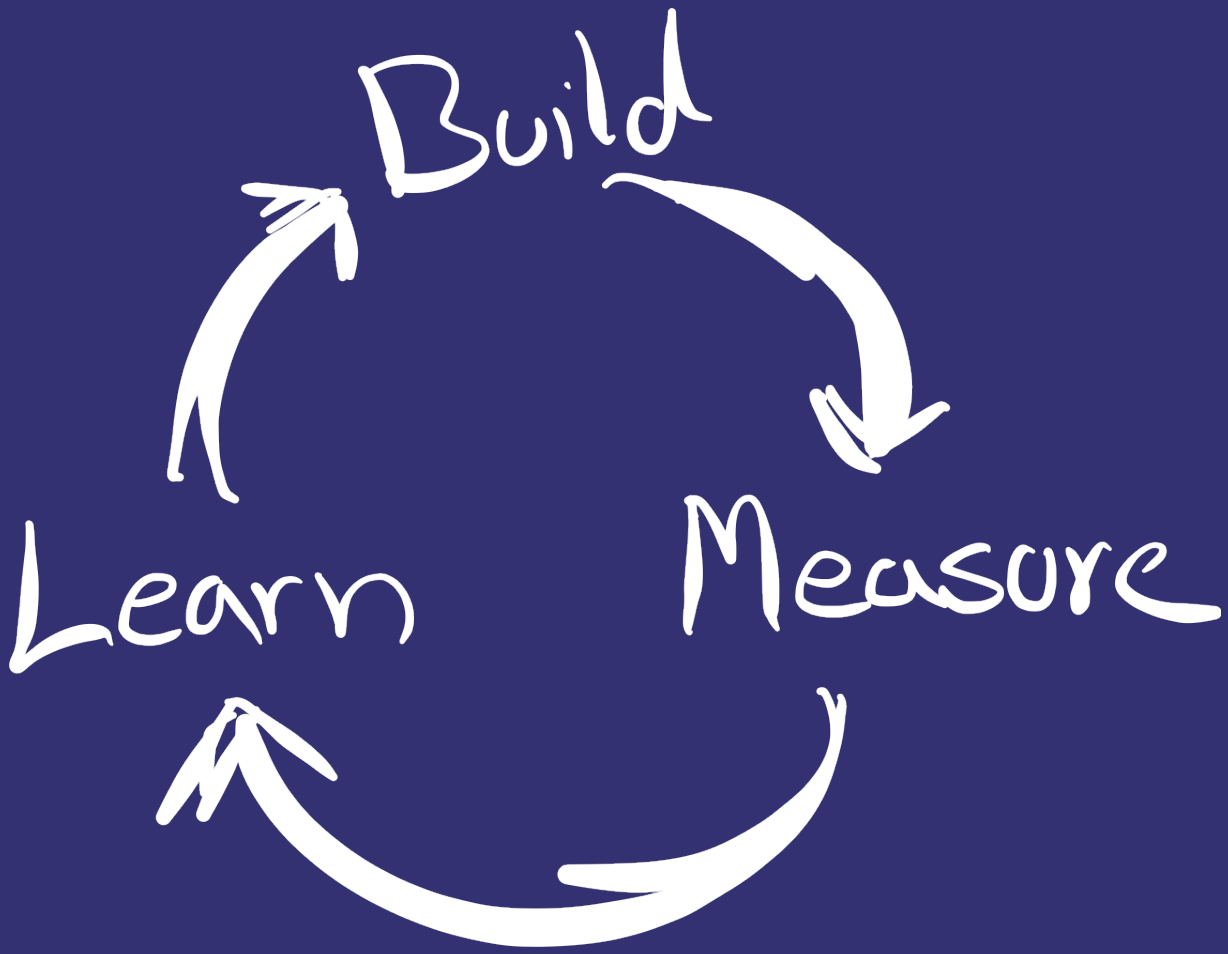
DevOps for SaaS startups



ISSUE 1- APRIL 24 2022

Inc-inc.dk

Words by Lars Kruse and Simon Grønborg

Illustrations by Simon Grønborg

“Think big, start small, scale fast”

The Startup Way is trending as an adventure everyone should try at least once, if for nothing else than the right to wear the “I Eat Failure For Breakfast” t-shirt.

The majority of startups are essentially software businesses, that is; the product is software. But typical entrepreneurs and innovators are first and foremost business developers and not necessarily ‘tech-savvy’, which puts a lot of startups in the awkward position that they are trying to get away with software, without knowing much about it.

Here’s what entrepreneurs need to know about DevOps if they are going to get away with software.

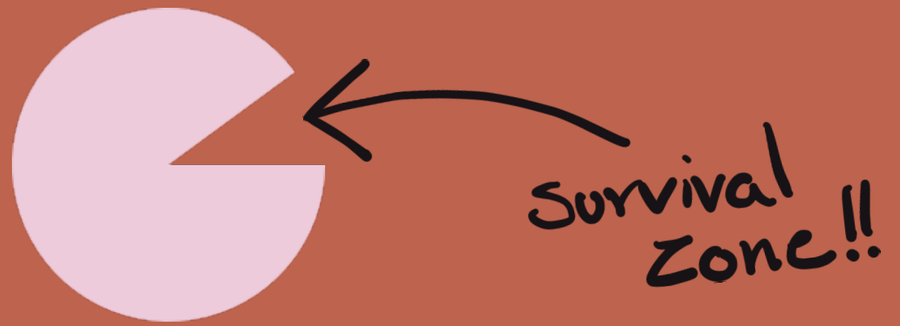
9 out of 10 startups fail

And it’s okay!

Some of them because the idea didn’t have the right market fit - no worries; these startups are meant to die.

However, a substantial number of startups fail because the software can’t carry the weight of the scale-up that follows a successful market fit or because the quest for the fabled market fit shoots in so many experimental directions that the software underneath can’t sustain the Design Thinking approach. This is worrying; these startups could live - if only the entrepreneurs understood the nature of software..

If you are an Entrepreneur in a startup, you’ll get insights into - just maybe - How To Get Away With Software.



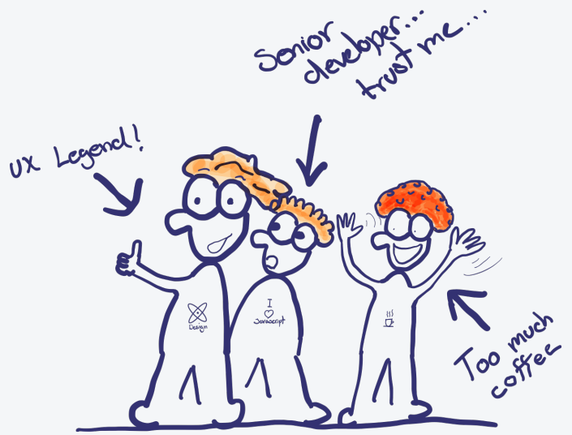
Tip 1

Build the right team

It’s an undisputed truth that software is eating the world. Your gaze has to wander far to find blue oceans on which to embark now disrupting startup adventures. SaaS products are popping up everywhere and the battles for customers are fought on the battlefields of the best user experience, the sleekest design, best communities, and most benefits.

Development and UX resources are scarce. They are hard to find, expensive to hire, volatile and notoriously hard to retain. Essentially they are like rockstars.

Have a strategy for how you attract the rockstars you need. Without a band, there is no music. No music; no concerts. No concerts; no sold tickets.
The core team should have at least one tech-savvy person onboard to take the lead on software and UX development.



Tip 2

Automate everything

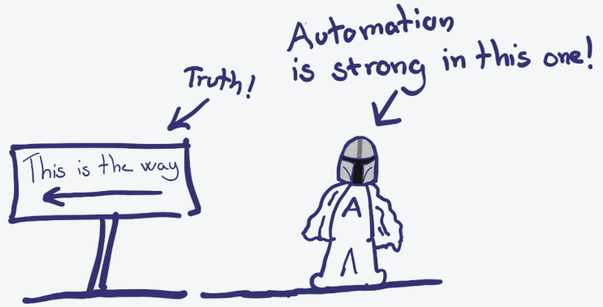
How many of your processes are automated? Build, Unit tests, E2E tests, code analysis, changelogs, release, deployment, verification steps?

Do it once; do it fast. Do it twice; automate it.

Manual processes are error-prone and time-consuming. Documentation is expensive to establish and maintain. Typically only a few individuals in the organisation know how to complete these processes and when they are otherwise-occupied panic starts. No one knows what to do.

Automate as much as possible; infrastructure, processes releases etc. Autonomize the rest; Build knowledge into the software so it can sustain itself without assistance from other resources like chat ops or admin consoles.

Turn *everything* into code and execute it - plain and simple.



Tip 3

Test in production

Can you run experiments in production? Create A/B tests? Can you segment your customers to offer a customized experience? - you should.

Your customers live in production, and so should you.

Production; people are afraid of it for all the things that can go wrong. Production should be a battle-hardened veteran able to survive a stampede of frenzied users treating the platform as a rental car. It should be resilient, unbreakable, and subject to Continuous Delivery.

User segmentation, Feature Toggling, A/B tests and tests in production are a must-have for any SaaS product. That's how you get to know the "what" in the "What do we have to change to build a great product".

Tip 4

Scalability

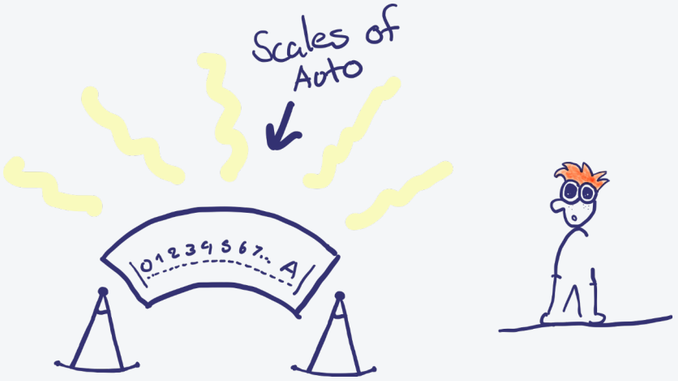
Is your infrastructure running on Physical machines? VMs? Cloud? Serverless? Is it managed?

Can you scale your business? Does your infrastructure follow?

"*How much time does it take for you to increase your capacity?*" That is a trick question. If you are doing it right then the answer is "*It's not something you should do and it takes no time at all!*"

When the marketing team is killing it and users are pouring onto your platform like an electronics shop on Black Friday, then the infrastructure should automatically scale to carry the weight. During the nighttime, when your users are sleeping soundly, your infrastructure winds down accordingly.

Make sure your infrastructure auto-scales.



Tip 5

Pick the right tool stack

What frameworks and technologies are you using? What processes are you using for managing work?

A cornerstone in “getting away with software” is the tool stack. It is our hammer and chisel, stone and mortar.

Progress over the last decade has left us with “everything as code”, serverless infrastructure, build-in analytics and a ton of plug and play services.

Pick a modern tool stack that can be configured as code, automated and require next to no maintenance, has a ton of plugins and can easily be extended. Development should only be on the core business logic, not on everything around it.

Most of the Millennials and all of Generation Z don’t get the Ops in DevOps; There are no Operations. To them, DevOps equals Programmable Immutable infrastructure, it’s all code. To them, the world is Serverless. In the future; be sure to pick a stack that can attract the talent you need.

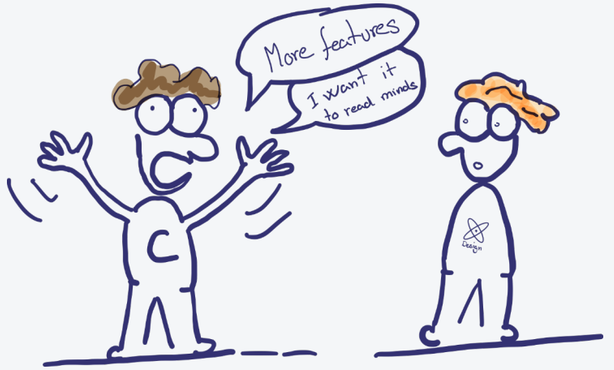
Tip 6

Listen to the customer

How do you collect feedback from your users and customers? What do you do with it?

Market fit is everything. This is where a large part of startups fail because they end up building the *wrong* thing. Establishing a customer's voice is an essential part of building the *right* thing.

Establish a customer voice through various channels. Customer support is a great place to get to know the needs of your users. Communities and social media are also two great places to learn from the people who are going to ensure that your product is going to achieve the mythical and well-reputed market fit. Don't fall for the misconception that it's only Marketing that needs to hear the customer's voice.



The customer's voice is a software enabler, so be sure to direct it to the software team as well.

Don't build *for* the customer, build *with* the customer.

Tip 7

Be data-driven

Do you collect data on user behavior automatically? How do you use it to drive product development?

Only Trust YODa - *Your Own Data*.

Don't ask questions; "*In thought-land, all you get is a bunch of opinions*". Instead, you should measure and collect data on user behavior. Know, rather than think or guess.

Collecting data is not a part of an evil business scheme of selling user secrets. That type of knowledge is much better gathered from a bush with a set of binoculars. Collecting data is how we learn what our users want and how we can give it to them.

Be data-driven. Learn from your data and use it to strengthen decisions in your product management process. Build experiments into the platform so you can learn what you don't know.

Tip 8

Easy onboarding for users

How easy is it to onboard your users? How long does it take for them to get started? What is your pricing model?

Months to build, months to sell.

Wrong pricing, slow onboarding, mediocre UX design, and difficult to use product, is how you spell CHURN with capital letters. Have a clear and well-formulated pricing model and make it clear what your product has to offer. Make onboarding easy, educational and entertaining. A good way to nail the right onboarding flow is to examine the Customer Journey.

Your product should first and foremost be delicious and easy.

Tip 9

Easy onboarding for developers

How easy is it to get up and running for new developers? Do they have unlimited access to production-like environments? Do they have a voice in the company?

In software development the workload varies. Early stages may require a lot of heavy lifting while later stages may require less. In time new features may call for upscaled effort and so the workforce must adjust. You need a small skilled core team, but you also need easy access to temporary help and sometimes even entire teams - for shorter periods of time.

Make sure developers can be onboarded easily. All the time spent getting a development environment up and running, enabling developers to contribute to your codebase is unproductive time, essentially time wasted. If it takes too long developers get frustrated and they jump ship.

Automated infrastructure and proper tooling are enablers of quickly getting access to production-like environments for developers.

Tip 10

Be design-led

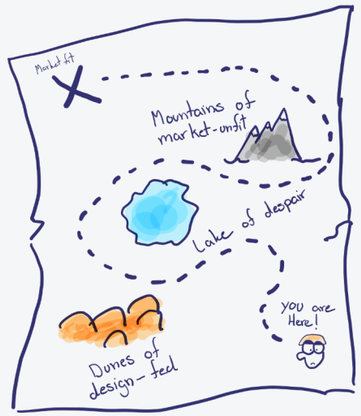
How do your designers and developers collaborate? Is your product design-led or design-fed? Who gets to decide the look and feel of your product?

All developers can hack a bit of front-end - but should they? We don't hack code, and we shouldn't hack the user experience either, we should use it as a rudder to our engine; it's steering.

The combination of developers knowing what is possible, combined with the insight that UX designers have about their users, is key. It's not something to split up into separate detached processes. It's teamwork.

UX and design is not "marketing" it's "software development". Consequently, your UX'er or Graphical designer is a software developer. If you don't have one, get one.

And be sure to make that person a first-class citizen in the development team.



This is the way!

No more talking, it's time to act. Every journey begins with the first step and fortunately this book have provided you with a map. Follow these 10 tips and experience first hand how your SaaS startup, maybe, just maybe, can get away with software and finally achieve the formidable market fit; becoming the most amazing product in the world.

Flip me over!

And read me like one, of your software books!