CATALYST FUND
POWERED BY BFAGLOBAL

Tech risk

What is tech risk?

Fintech startups are powered by tech, but that tech can either be a source of strength or a weakness. When the tech is built according to best practice and makes efficient use of other platforms, then it can provide a foundation for achieving product market fit and beyond. However, tech can also be a hamstring for startups if it does not allow for quick iterations or integrations.

A common situation (in maybe 1 out of every 5 startups we meet) is a distinct **lack of tech**. This can take place when none of the co-founders has a technological background. In these instances, the startup has been built on Excel sheets and relies on manual operations for data entry and analysis. Spreadsheets are great solutions, and they can take you farther than you think, but startups can outgrow them fairly quickly. They are ideal for early days, when the team is still validating the problem. However, once the team has a working solution, it will struggle to scale without upgrading the tech.

Lack of tech can also be the diagnosis if a startup's tech stack has largely been bought from software vendors or if the startup is using someone else's platform. In these instances, the tech solution tends to be very rigid, and is not customizable. Most often, the data is also inaccessible. Furthermore, such solutions have very low defensibility since the architecture isn't owned by the startup and can be sold to other providers.

On the other side of the spectrum, are startups that "**do it all**", meaning that the team develops code for everything and bypasses established tools like Hubspot (for customer relationships management) or Metabase (for dashboards). This means they will not have tech resources to develop solutions to users' problems or create valuable intellectual property for the startup. In many instances, such startups code more solutions than are needed, putting the tech before users' needs. As a result, they end up with a solution saturated with tech -- USSD, SMS, mobile apps, web app, and a Whatsapp bot all at once. The result is a constellation of solutions and channels without even knowing how the tech is being used.

In the middle of the spectrum are startups with **low quality tech** when the engineers do not know how to develop software according best practice using advanced tools, storage solutions, data privacy standards, and dashboards. High-quality tech teams should maintain clear documentation, version control, and avoid duplication.

In contrast to low quality, another tech risk is **over architecting** or trying to make use of advanced technologies before there is a real need. We understand this strategy is considered an investment for the future, intended to position the startup for scale, but in early days, simpler solutions that are easier to adopt and adjust will allow for greater speed towards product-market-fit. For example, we have recently been seeing more startups make use of NoSQL databases like MongoDB or Firebase, which give the tech team good flexibility for designing data schemes and also allows for extensive storage. This solution makes sense when your business requires massive amounts of data, but for early-stage startups, this type of database makes data difficult to access and extract. Startups often fall prey to this risk when startups are overly seduced by **hyped technologies and buzz words** like AI/ML or using an .ai domain.

Key factors of tech risk

- Database type
- Control over solution
- Extent of customization
- Hosting solution (PaaS vs IaaS)
- Approach to data analytics
- Solution ownership (proprietary vs. outsourced)
- Defensibility of solution v competitors
- Scalability of solution

Mitigation strategies

Depending on the diagnosis that best describes your startup's current status, you should adopt mitigation strategies that set your organization up for scale and success.

Still get the most out of excel sheets with off-the-shelf solutions

For example, if your startup relies heavily on spreadsheets and manual entry data, platforms like using AppSheets (as did Catalyst Fund portfolio company Spoon) or Glide can allow you to get the most out of them by creating custom apps on top of sheets. These apps could be used for internal teams, field teams, or agent networks as well as other stakeholders. These types of custom apps can also be connected to other platforms like accounting systems, customer relationship manager programs, payment gateways, and more via Zapier (see this example with Paygo Energy).

It will be important to consider the pricing models of these platforms to project future costs as your team and transactions growth. These solutions will allow you to grow your business to the next step, but costs may also go up. In this case, it is important to understand when it is right to invest in larger and more complex architectures.

One caveat is that these apps are not ideal for customer-facing interactions, particularly if your users are not very digitally savvy. To integrate digital methods into customer interactions, your team will need to consider the balance between tech and touch.

Build vs Buy

When considering the balance between what to build and what to buy, there isn't a straightforward approach. Some of the factors to consider in making the decision is (summarized from here):

	Build	Buy
Value proposition	Is the element core to your value proposition?	Is the element core to your value proposition?
Budget	Does your team have the requisite time and skill?	What is the cost of purchasing, especially as the business grows?
Timing	Do you have enough time to for team to build a solution?	How valuable is it to get the solution quickly?
ROI	Could you monetize the solution in some way to increase the ROI?	Could you invest the time and resources in a better way to improve returns?
Data integrity	Could you maintain better control and quality of data with a custom solution?	Would a bought solution be as helpful in maintaining data accessibility?
Integration	How extensively does the solution need to integrate with other, custom parts of your system?	Would off-the-shelf solutions easily integrate with your existing solution.
Level of control	Would a custom solution help you to access and analyze better?	Would a bought solution be as helpful in maintaining data accessibility and analysis?
Support	Is support for future customizations easily available?	Could an external contractor provide superior support for future needs?

Have a hiring strategy to form your tech team

If your business requires a considerable level of scalability, and there is a gap on the technical knowledge normally because none of the co-founders come from a technological background, you would need to form a hiring strategy. You could either start with a talented junior engineer and grow and skill them up, or you can look for a more senior position or even a co-founder. Your strategy would depend on the funding allocation for this. In the internship, you can also use a freelancer service and contract them as needed. It's important that you think and define a plan for this strategy ahead of time, before it's too late on the journey.

Have a hiring strategy to form your tech team

If your business requires a considerable level of scalability, and there is a gap on the technical knowledge normally because none of the co-founders come from a technological background, you would need to form a hiring strategy. You could either start with a talented junior engineer and grow and skill them up, or you can look for a more senior position or even a co-founder. Your strategy would depend on the funding allocation for this. In the internship, you can also use a freelancer service and contract them as needed. It's important that you think and define a plan for this strategy ahead of time, before it's too late on the journey.

Implement software development best practices

It is important to establish an internal culture of excelling on your tech team so that they care about the quality of the software before it's too late. We have seen many tech teams that have had to re-build their entire platform from scratch because they were more concerned about speed than quality during the early days. Implementing some key best practices can help you avoid this fate; they include:

- Have a member of the team responsible for quality assurance (QA). This could also be a rotational role.
- Have well-defined dev environments for development, testing, production, etc.
- Have a CI/CD (continuous integration/continuous development) processes and tools to be able to run tests and deploy new versions smoothly
- Implements code reviews and pair programming so that team members can support each other
- Develop unit test as well as integration testing
- Plan for the release of new app updates that includes a group of beta testers so you can gather early feedback and catch software bugs
- Test your app against a massive collection of physical devices using services like AWS Device Farm
- Monitor important technical metrics such as app crashes, server downtime, etc. and make your tech team accountable for these metrics

Scalable architecture design

Although startups start small, they need to build their tech architecture with scale in mind, upgrading on a just in time basis by anticipating business growth. One critical method for anticipating upgrades is to track service speed and to figure out what is slowing it down. The solution, described in depth here, probably includes:

- Horizontal Scaling
- Caching
- Duplication

- Create subsets
- Splitting functionality
- A combination of the above

This design mindset is needed when your application is hosted in a Infrastructure as a Service (IaaS) platform, meaning you ned to run load and performance testing to understand your systems breaking points and benchmark system performance.Alternatively, you can make use of a Platform as a Service (PaaS) that would "allow you to develop, run, and manage applications without the complexity of building and maintaining the infrastructure typically associated with developing and launching an app".

Additional resources

Google for startups

- Data insight tools
- Tools to build a product