

10 lessons startups can teach corporates

Varsity dropout and former super-yacht engineer, Roger Norton now spends his days creating support systems for startups as CEO of [Playlogix](#). He believes there's a lot Big Business can learn from how startups think.

He recently published *Start Here* - a guide to building and testing startups as well as creating the methodology behind the Lean Iterator that has helped coach over 40 entrepreneurs and been used by the likes of UCT GSB, Idea.org, Alan Gray Orbis Foundations and Standard Bank Innovation.



"Innovation is the new competitive advantage, and large companies are realising that it's hard to do when culture, processes and mindset don't support this new way of thinking. Startups, however, are increasingly proving to be great vehicles for creating innovative products as they continue to disrupt markets and outcompete the more entrenched larger and slower companies (until they get acquired at least...).

"Acquiring external innovations and merging them into a larger company is an approach that often fails. This is because the dynamics that drive a corporate for things like risk reduction and cost optimisation are totally at odds with the dynamics that have allowed the startup to thrive in the first place. Applying key practices to create the right environment could significantly increase the odds of success," says Norton.

Firstly, startups are small autonomous teams that work under conditions of extreme uncertainty, searching for a repeatable, scalable business model by being laser-focused on the value that they provide to their customers. There are also many

dynamics at play in a startup and an important number of constraints. For example, time and money, and the type of funding startups raise needs constant validation and proof that they're on a winning track to encourage them to keep experimenting until they are sure (in theory, but in practice, it's a lot messier...)

To recreate these startup constraints, while removing the big corporate ones, is no simple feat. Here are some tips on what helps that we've picked up along the way so far:

1. Run many small projects simultaneously, not a few large ones

You're not going to get the best ideas in the beginning (no matter how good you think it is now). Running lots of small experiments allows you to "learn how to learn" faster and increases your odds of finding amazing opportunities. It's a numbers game - ask any venture capitalist. This approach also allows you to focus your energy and capital on what really matters and leave off the 'nice to have' features.

2. Create a safe-to-fail environment

Running lots of small experiments is a great way to achieve a safe-to-fail environment, but an extra effort should be made to celebrate the failures as these indicate the things that you've learnt. It's also important not to overhype small experiments and create high external expectations. Every project you invalidate early saves you the money you would have previously spent trying to launch it. Fail fast and early.

3. Create cross-functional teams

Use multi-disciplinary teams from many different areas of expertise and various levels of management. Diverse teams not only bring very unique perspectives to each problem, but they also allow the space for the idea to morph into a bigger opportunity in an adjacent area. Hierarchy bridging teams help allow decisions to be made fast and implemented faster. You need to keep the feedback loops tight.

4. Have a single driver for each project

If you're trying to build a startup, you need an "entrepreneur". One person that is involved in every aspect, has all the context and can make decisions really quickly. The buck needs to stop somewhere, and at least one person needs to be 100% focused on making it work. This person also needs to document the project and decisions along the way, something that is critical when needing to report to the traditional business.

5. Have a clear validation path for each project with clear milestones that needs to be followed

Mapping out clear objectives, what is expected at each stage, what support is available, and what the team should be focusing on at any given time helps create the laser focus on what's most important. The objectives should also set out time and budget limitations throughout the process (we've put together the Lean Iterator process under a Creative Commons license to help with that).

6. Focus on solving a customer's problem, not on a particular solution

By trying to build a particular product, it's not complete until it is, and that means that you can't learn anything until the end. By focusing on a customer's problem, you will easily find ways to make improvements early on, and you will learn your way to the best solution. It also means you're more likely to build something people actually want.

7. Identify the business unit that will be the custodian if it works and engage them early on

If it works, then the project is going to need to move onto a department's balance sheet. Keeping them in the loop of the project from early on will help you build something that makes that process much easier. Find out what their KPIs are and how you might affect them. Understand the corporate governance restrictions that you're going to have to navigate. This aspect is one of the biggest failures of the "successful" projects that I've seen.

8. Define success before you run experiments and review regularly

Every experiment needs a hypothesis. You need to know what you're testing for before you start. Clear success criteria help you work out what is most important and is the easiest way to prevent getting distracted on things that don't matter. It helps you hear the signal in the noise. Research competitors as early as possible to make sure you're differentiated and

keep testing for feasible revenue streams early.

9. Allow anyone in the company the opportunity to try something

Innovation is not limited to an “innovation team” or a particular level of employee. To build an innovative culture and environment, you need to allow anyone in the organisation to try something, give them the time away from their normal responsibilities they need and not punish them when they go back to their role.

10. Have clear incentives for winners

Startups are hard. The risk, pressure and energy required to make them work need to be worth the reward. The type of reward will depend greatly on the project, but there should be a rewards framework defined up front. This compensation could be in bonuses, recognition, profit share or something similar.

The bottom line is that in a corporate environment that optimises for cost reduction, failure is seen as a waste. But failure is inevitable when you are trying something that hasn't been done before. It is better to optimise for “maximum learning” which is how you optimise to come up with new innovations fastest. And we believe that creating startups is just a more reliable way to do this.

For more, visit: <https://www.bizcommunity.com>