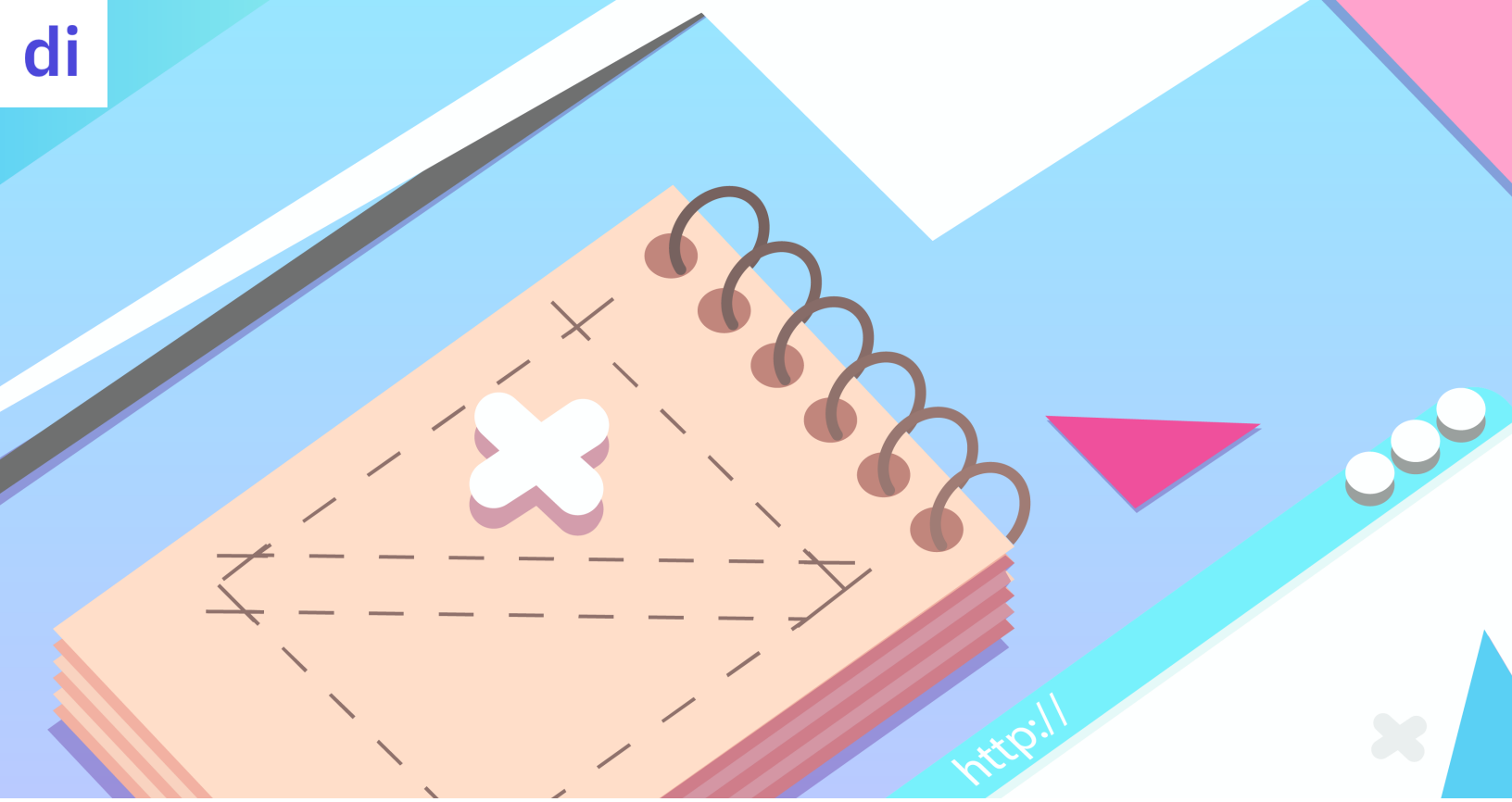
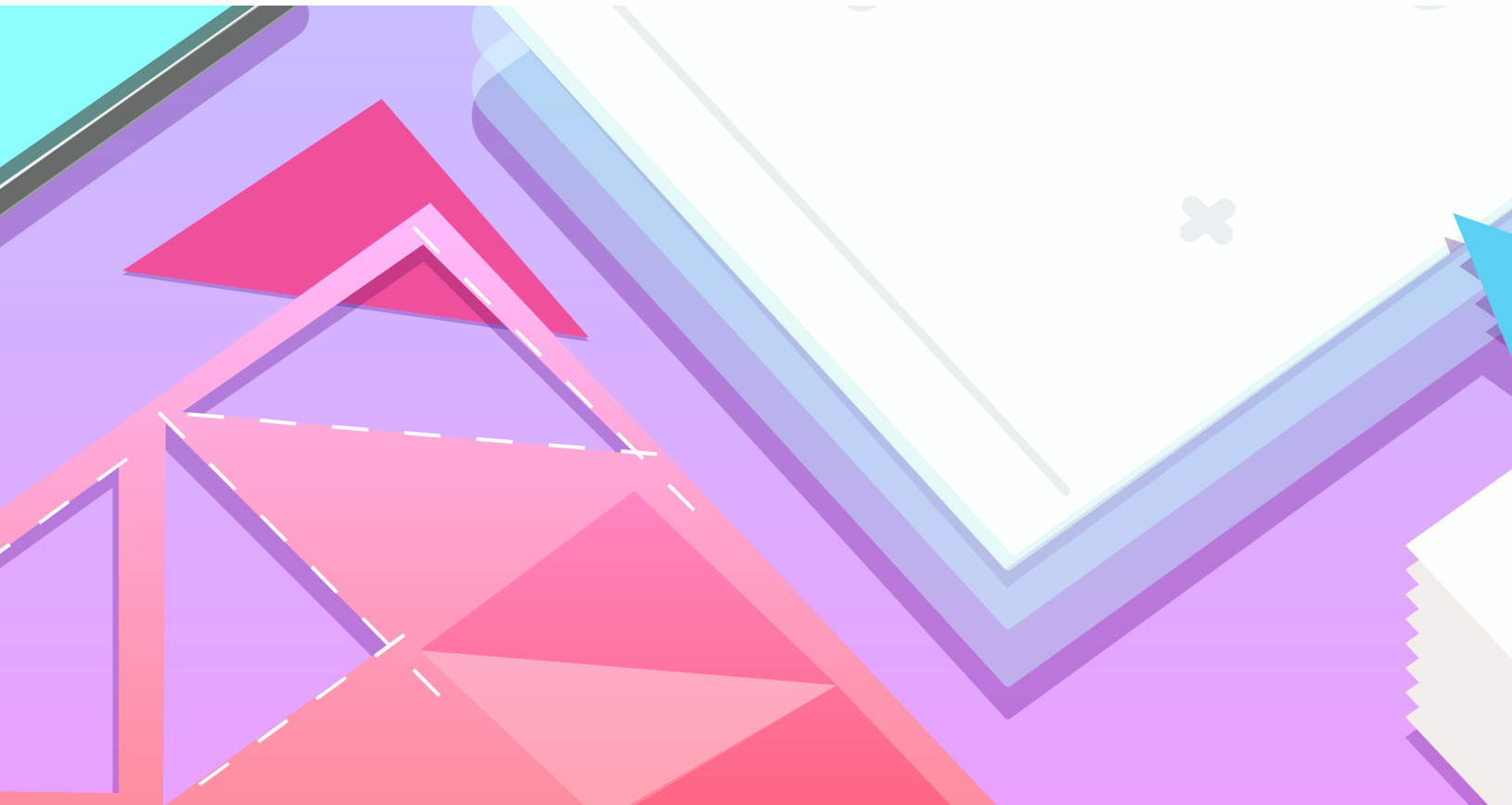
di



11 Laws of Product Development



DI helps amazing organizations unlock extraordinary value through digital innovation.

DI's unique combination of product strategy, user experience design, software engineering and digital marketing allows organizations to rapidly execute on their digital innovation efforts.

Whether you're building a new customer facing product, engaging in internal digital transformation efforts, or seeking to leverage emerging technology, DI can help.

About The Author

Sean Johnson is a product strategist, user experience designer, and marketing expert. He is a partner at Digital Intent and sister company Founder Equity, a fund that employs a novel approach to early-stage investing. He also teaches innovation and entrepreneurship at Kellogg School of Management.

He speaks all over the country on innovation and growth, and writes for Forbes, Inc, Mashable and more.

Creating a startup is extremely hard. There are dozens of barriers and pitfalls along the way. Success requires fanatical execution and a bit of luck.

But while there are hundreds of ways a startup can fail completely outside your control, the product is not one of them. There are dozens of ways of building a product or a particular feature. Not all of them are created equal.

We've built over 100 products at Digital Intent over the last 6 years. And while every product is different, all those reps have helped us discover some broadly applicable patterns that can increase your chances of success.

These are certainly not comprehensive – there are many types of applications and technologies that these might not apply to (IoT, AI, VR, etc). But if you're building a web or mobile app, hopefully at least one of these is helpful.

1. Minimize Risk with Real Customer Development.

The first and most important rule is to do everything you can to avoid building something people don't want. The number one reason new products fail is an inability to achieve product-market fit.

This is usually a result of insufficient customer development – people go from an idea on the back of a napkin to writing detailed functional requirements, without asking whether people truly want the product in the first place.

If you haven't started building yet, one of my first questions is likely to be "how many customers have you talked to?" The most common answer is zero. Often the answer is effectively something like "I talked to 5 people, talked the whole time, and only used their feedback to validate my initial hypothesis and make pretty quotes for my deck."

Given the proliferation of content around customer development and lean startup principles, this boggles my mind.

Wasting months of your life and thousands of dollars of other people's money to solve a problem you're not sure exists is insanity.

De-risk your problem hypothesis as soon as you can. Be rigorous about it. And be 100% honest with yourself as you receive feedback. Iterate on the problem and solution hypotheses until you've found something people get legitimately interested in.

2. Build an MVP. Don't Necessarily *Code* an MVP.

MVP does not mean software. The purpose of an MVP is to test the riskiest assumption of the underlying business. This can often be done without a line of code.

One of our favorite entrepreneurs used a Keynote on an iPad to test mockups of his loyalty platform with small businesses. He would go down the street along the busy shopping district, try to sell them the platform, find out why they weren't interested. He'd sit in Starbucks for an hour, make changes to his mockups, and head back out. As a result he was able to learn more in a day than many entrepreneurs learn in months.

One DI client built a startup focused on using machine learning for a tool targeting hospitals. While that was his final vision, his initial solution was CSV files and Excel spreadsheets. His clients didn't really care how the system worked – they just cared about the results. He acquired 30 customers and got to a $1mm run rate before investing in the automated system.

Code is always dramatically more expensive to build and change. While it might sound like lost time, it's usually more than made up for by the conserved burn rate and the confidence when it's time to build that the underlying assumptions have been sufficiently de-risked.

3. Incrementally Better Won't Work.

Behavior change is incredibly difficult. The average user has 26 apps on their phone, and only 5 apps see heavy use. There are well over 2 million apps in each app store. The math alone suggests integrating something new into people's lives is no small feat.

It is extremely rare for people to adopt a new solution to a problem they've already solved for themselves. I don't need a weather app that is 5% more accurate – the app I currently have is good enough.

More importantly, you're not just competing against other weather apps. You're competing

against literally every other demand on my time or attention.

New solutions have to either be dramatically better than the status quo, or have to completely reimagine the experience to dislodge an incumbent and carve out space.

4. Simple is better.

Most apps people love are obsessed with what we call the Core Experience. It is extremely rare for products to do 12 things simultaneously well.

This is particularly true for mobile apps. If the Core Experience isn't amazing, bolting additional features onto the product won't move the needle. If I don't think the content in your UCG app is interesting or useful, a favorite or sharing feature isn't going to make me use it. Be obsessed with making me fall in love with the content itself first.

Identify the 1 or 2 things your product needs to do to be better than everyone else.

Spend most of your energy there. Iterate on it until that functionality is world class. Only add ancillary functionality if your customers are yelling at you.

5. Nail Onboarding.

Even if you create a great core experience, it won't matter if people don't stick around long enough to experience it.

The first time user experience and onboarding process is essential to making the light bulb go on for customers. While it's true a user's motivation to use your product will never be higher than right after they sign up, their enthusiasm will wane quickly if the onboarding

process is cumbersome, confusing, or takes too long to create value.

For most, onboarding is treated like something you do at the end right before launching and given far too little thought. In most cases a simple card-based tour is simply lazy, and likely hurting effectiveness.

Treat the onboarding process as an essential part of the core experience.

Ideally have it assist in the user creating content or completing whatever activity maximizes their chance of adoption.

Don't neglect the role that email nurturing can play in your onboarding process. Particularly for SaaS products or products with a free trial component, educating your new user on the value of the product post-registration is critical.

6. Notifications Don't Have To Be Annoying. They Can Even *Be* The Product.

At a minimum, email and text notifications represent one of your primary tools to increase retention. Even if users understand the utility in the product, many will drop off simply because they haven't engaged in your core experience enough times for a habit to develop. Notifications can be the trigger to help them build that habit.

But you can use them for more than notifications – you can bake functionality directly in. Nobody really wants yet another site to log into. All they care about is the utility your product provides – if it's a medium that's easier they'll be all for it.

The Quora email digest is a fantastic example – it's the same content on the dashboard. But rather than relying on me to log back in, they send the content to me. Because it's continually interesting,

I engage with Quora much more frequently than if I were left to my own devices.

Email and text shouldn't be afterthoughts – they can be an essential part of the user experience.

7. Compete on Iteration Speed.

The first version of your product is going to be wrong.

Users won't understand the value proposition, or they will but will find it too hard to sign up, or they'll sign up but not engage further, or they'll engage further but not convert from free to paid. Expect it. Plan for iteration.

One benefit of focusing on the core experience is it can speed up your time through the Build-Measure-Learn loop.

All other things being equal, the company that can iterate on their product in response to customer feedback the fastest will usually win.

Of course, don't iterate for the sake of iterating. This assumes you have the proper analytics set up behind the scenes (hint – you want to be tracking what happens on and between pages, not just big events like registrations or orders).

Identify the actions that represent the "aha moment" for your app. Use your analytics to find identify bottlenecks preventing people from getting there. Use that analysis to prioritize your iterative work. Rinse and repeat.

8. Stay close to your users as long as you possibly can.

Analytics only tell you part of the story. Too many founders hide behind their computers, afraid to talk to their customers.

But analytics will only tell you what happened. They won't tell you why.

You can either conduct a ton of iteration based on analytics alone and stumble half in the dark, or you can talk to your customers and find out much faster how to improve things.

Have a plan for engaging with your early customers throughout their journey. Make sure you know exactly what they do, what they think, what they like and what they don't.

Not all feedback is equal. Your biggest fans matter – find out what they like about the product and everything else you can about them. There are probably more people like them.

Also pay attention to the people who are on the fence – they like the product but they don't love it. Find out why, and give them what they want.

The people who don't really like it? Unless that's everyone, don't focus your energy here. No product is perfect for everyone.

9. Seed and Curate.

A major flaw in many user generated content sites is the "if we build it they will come" fallacy. Too often companies wait for the users to dictate what the product should be.

Instead, leverage internal resources to create fantastic examples of the kinds of user generated content you hope the community eventually creates.

This solves multiple problems. It populates the app with content so it doesn't look like a ghost town. And it teaches users how the app works – the types of content that get created, what gets rewarded by any game mechanics, etc.

This is doubly important for any machine-learning based systems, where again sufficient data is necessary to effectively train a

system. People can very likely create solid lists of the best restaurants in a neighborhood (either themselves or informed by other existing lists).

Users likely won't question whether the data is good – their implicit assumption will be that it is by the way the app is positioned. In fact, they are much more likely to question it if the algorithm isn't sufficiently trained and suggests poor data as a result of a small number of inputs.

This is triply important for marketplace businesses, which often face the dreaded chicken and egg problem. The best solution is often to control one side of the equation as much as possible. Don't wait for restaurants to sign up for your Yelp killer. Add them yourself, for free. Drive traffic and see if your experience is superior. If it isn't, improve it. If it is, get the restaurants hooked on the value before trying to charge them money.

The tech is not your asset, the content is.

10. Identify Your Growth Loop

The fastest growing products have a growth loop built in – a self-reinforcing cycle that accelerates over time.

There are three primary growth loops most apps leverage:

- **Organic** – users generate content, some of which is publicly accessible. Google picks it up, driving organic visits, which turn into users creating more content for Google to pick up. Most common with UGC sites.
- **Paid** – users come in, make enough money to justify the cost spent to acquire them. That money is plowed into acquiring new users. Common with games and other freemium apps.
- **Referral** – users are incentivized to send invites to their friends. Some percentage sign up and invite their friends. This loop

works best when User B's existing in the platform makes it more valuable for User A. But this can also work for ecommerce or other sites, typically in the form of a symmetric bonus (i.e. invite a friend, both get store credit.)

It's rare none of these levers are available to you – sometimes you can leverage more than one. Identify which one would be most applicable and bake it in. Iterate on each until you get it right.

11. Don't Scale Too Early.

The Startup Genome Project has done autopsies of hundreds of failed startups. 70% of them end up failing because they scaled prematurely – meaning they hadn't reached product market fit.

Product-Market fit is admittedly a fuzzy term. However, there are some good proxies for determining if you're on the right track. Getting to a place where you acquire over 100 users per day organically (including referral) is a great sign. If 40% of your users say they'd be very disappointed if the product went away, odds are you're in good shape.

Product teams should be aggressively iterating on their product until P-M fit is achieved. They should certainly be testing acquisition channels, but emphasis should be on acquiring enough users to see what parts of the product are busted, so they can improve and get closer to P-M fit.

Only when they're confident P-M fit has been achieved should they aggressively focus on acquisition or monetization.

Start Building Better Products Today

While following these rules obviously doesn't guarantee success, we strongly believe they can increase your odds, sometimes considerably. The world is full of beautifully designed, well-engineered products nobody uses. Don't build one of them.

Are there other rules of thumb we left out? Any we included you strongly disagree with? I'd love to hear your thoughts!

And if your organization is looking to build something new, we'd love the chance to learn more and work with you. Contact us, or connect with me on LinkedIn. Let's build something great together!