Steve's Bizapps ISV Notepad

01 idea

You identified an opportunity. Maybe it's a gap in Microsoft's offering, maybe you are refreshing some current functionality, maybe you are repackaging to serve a particular industry, or maybe you are building something brand new on the platform. Great! But don't build anything yet.

The very first thing you should do is review the product roadmap to make sure Microsoft is not planning to release anything that would impact you. While you are at it, check AppSource to make sure it does not already exist, or that you have a unique angle if it does.

02 licensing

If you are planning on building on top of the Microsoft Business Applications platform, you need to have a clear understanding of what licensing will be required, under your application. While it may not impact you directly, it will impact your potential customers' cost. When possible, you would want your application to run on the lowest cost licensing model, even if it means re-thinking your approach.

Don't create a mountain between you and your potential customer. Get a deep understanding of the licensing, and step on top of the proposed changes to know how they may impact you.

03 pricing

The most important thing you will do, is set your price. What you would like to get, or what you need to get to recover your costs, is irrelevant. You do not determine "market value", your customers do. Do not launch with a price that has not been tested with potential customers first. Don't hide your price, but make sure customers understand your full value proposition before they ask it. Remember, you can always lower a price later, but it is very difficult to raise it.

Avoid offering anything for free, even just to prove viability. The people who react to free are not the same ones who would pay for something.

04 math

With a grasp of the licensing requirements, and a sense of a viable price, together with an estimate of your development and backend costs, it's time to do some math. Too many ISVs either skip this step, or they get unrealistically optimistic, and race out and build blindly.

How many potential customers are there for your app? How many of those might buy it? Take your most conservative estimate of actual potential buyers and cut it in half, and then cut it in half again. Is it still viable? If you have to start rationalizing, you are only fooling yourself. Best case thinking, leads to worst case outcomes.

05 mvp

Even if your math works, you would still be wise to not build your complete application, unless it is pretty basic. Instead, build a Minimum Viable Product version (see A skeleton of your application that maybe is missing most, or all of the second-level capabilities. This is one area where you may want to offer this skeleton for free to a few potential customers as a pilot. If they don't like your skeleton, you will be glad you did not invest more in putting meat on the bones. More likely, they will provide guidance that will significantly impact your original product-level plans.

Don't assume you know what the customer needs.

06 test

For your MVP pilot, try to get as broad a cross-section of your target customers as possible. As an incentive for their feedback, you may need to offer them a deal discount for the final version later. But the feedback at this step, is the most valuable that you can get.

Don't argue with the feedback, it is real. Agree with everything they say. If you find yourself having to explain the much-to-do your pilot group, this will be a much-bigger problem later.

Focus on the common issues, but don't ignore the outliers, they might be the best ideas.

07 build

Based on your pilot group feedback, you now have a good idea of what to actually build. If your application will have several levels, you may want to just build the second-level, and then re-pilot before building the third, and so on.

Make sure your development team understands the licensing lanes you are targeting and doesn't stray outside, bringing another license need. Spend as much time on the mundane, as the exciting parts of the build. Watch closely for scope-creep on the fun stuff. Don't pretend, just because you can.

Focus on Form over Function, 100% of the time.

08 test

Success is awarded to those with patience. Corners cut on future will come back to bite you every time. If you cannot afford to thoroughly test, both internally and externally, and that you must rush out and try to generate revenue to fund the development, you should not pursue the ISV route.

Another option is to reduce the functionality to fit into what you can afford to build, and properly test. Depending on what you eliminate, you may have to revisit your pricing plans.

Leave room for future versions of your application.

09 backend

As part of your Math analysis, you need to understand the backend requirements to be an ISV. These are costs that are not part of your application, but are necessary to succeed. If you are planning to include the Microsoft licensing, how will you provision that? How many manual steps are there for you, or your customer to get your app up and running. How will you manage recurring customer payments, and their adds, deletes or cancellations? An eCommerce system will be required. How will you provide support for your application customers?

Be sure that all of these things are scalable!

10 channel

An obviously critical component of your success is identifying the channel, or channels, you will use to distribute your application. The simplest is Direct, meaning you will engage directly with your potential end customers. AppSource is an option for that but a reseller channel or distributor could potentially expand your customer base exponentially. Both of these options require significant effort. If part of your revenue model is margin on the license, these options could be problematic. If you are thinking about a reseller channel, be aware that an entire new set of assets will be necessary for their engagement.

Know how you will support customers indirectly!

11 appsource

AppSource is an ISV best friend. Regardless of channel(s) chosen, AppSource plays a role. For direct, AppSource can provide the provisioning for you. But be aware, there is no de-provisioning capability. You need to know how you will restrict access to your app when a trial ends or a customer stops paying. Otherwise, you just built a free app.

For an indirect channel, an AppSource Test-Drive is your best bet. A customer can experience your app, and then be directed to your channel for purchase.

Protect your test-drive from bad-faith actors!

12 ocp

If your application has Enterprise Customer potential, you will want to go beyond AppSource alone. Our Commercial Partner (OCP) has a specific, "Build with", "GTM" and "Sell with" program you should explore.

The ultimate goal of this process is to get your application "On-Sell Ready" and listed in the OCP Catalogue. This is the source that Microsoft internal sellers use, to find applications that would help them to close Enterprise deals. It is a critical motion for Microsoft.

OCP requires a lot of effort from you, short cuts!

13 gtm

Even if you are in AppSource, and you are On-Sell ready in the OCP catalogue, these are just part of your Go-to-Market (GTM) strategy, and they are Plan "B". Do not stand on all of your popcorn coming from there. In fact, to be safe, assume you will not get a single opportunity from those avenues, because you may not. You need to have a Plan "A", which is your own GTM strategy. Web Assets, Social Marketing, cold calls, letter in a bottle, whatever it may be, you need to have it. All of this needs to be tested as well before you go wide.

BTW, if you have a B2 app that needs a one-hour sales call with each customer, you have a problem.

14 hope

Even if you followed every step above, there is no guarantee of success. A customer buying your app is great, but it could just as easily have been a mistake. Once you are that fifth or tenth customer, you can relax a little but, but then you need to watch the churn rate. Ten customers a day signing up is awesome, unless you also have ten prior customers a day cancelling. Analyze this like your app's life depends on it, because it does. Are you building your value proposition? Have you provided enough training? Did you "miss" on price? Test alternatives, and test again.

If you hit all the "sweet-spots, congratulations!