1.      How big do we want to get?

I would like to stay small. I think my preference would be to stay serving smaller businesses, (mom & pop shops). I think we would need to bring on lots of new systems, but feel like we would only need to have a few more pieces to our current team to make it work. I feel like if we keep going after and getting bigger, public, companies then we would need to get more developer and customer service help to keep up with the demand of having an enterprise level service.

I think no matter what we could use 1 more server guy to help Wayne out and we could probably use 1-2 more developers to help with bugs, tests, and Metrc stuff. I think if we are serious about trying to sell more systems we will need to invest in marketing and sales.

2.      What is our time frame to get there?

If we stay small I think we could get to as big as we need to in a couple years if we bring on a new system each week (4-5/mon).

3.      What do we want to do/have for our company/user culture?

I personally like our culture. There could be better defined channels of responsibility potentially to resolve and complete tasks. I like that things aren’t so rigid that you have to jump through a bunch of bureaucratic hoops.

4.      What are some of the things you like about what we are doing right now?

I have always really enjoyed the people I get to work with and the problems that we get to solve for our clients. I like the freedom to dream up solutions.

5.      What are some things you would want us to change?

I feel like we are teetering on the fence on where we want to go as a company. We were built more as a solution for the small guys, yet now we have demand from major companies that like our solution, but they expect and demand an enterprise level product. Are we going to define what we are (small vs big)? If we want to keep both, maybe have 2 entities, 1 focused on the small guys and one focused on the larger companies because their needs will be different. Which are we going after and which are we trying to live up to? I feel like if we don’t decide which to target, we might be stuck in no man’s land where we are thinking about the small guys with development, but think about the big companies with the server reliability and scalability.

I feel like our risk management is our largest vulnerability right now. I think we should have a hybrid of our old approach and the new approach. The old way was just to push things up and fix them as there are issues; our users are our testers. We have a lot of companies that don't mind that as they get access to the new features and don’t mind being the beta testers. Other companies really detest that model.

I think it might be worthwhile to create 2 groups of servers.

-       Group 1 would be our beta corps where we can push changes up to those whenever those changes are ready and part of their perk for being the beta testers is they get quicker access to new features.

-       Group 2 would only get new features after they have been beta tested and only get updates quarterly or bi-annually, so the risk of service interruption is decreased.

6.      As far as structure, what do you like? Do you like the independent model or would you like it more solid?

When I first started I would have preferred to be an “employee” and have the security of a steady paycheck, hours, benefits, job title and responsibility, etc. The longer that I have been self-employed, I have started to really enjoy the freedom and options that I have available to me with this model. It took years to change my mindset, but now it’s hard to think about going back to the other model.

7.      ![IMG_256](data:None;base64...)Do you like the jellyfish model? (Organized with a loose, flowing feeling. Adilas is the overarching umbrella and there are a number of sub departments or subsets.)

I don’t feel like this “jellyfish” model is all that different from a normal business model. I feel like it only really feels different because we aren’t large enough yet to have teams dedicated to the different sections and many are individually managing or providing several of these “departments” by themself. I would say the only thing that is different about our structure is the money generated in each department isn’t really managed the way a typical company model would handle it. A typical model would have all the money go to Adilas and then Adilas would budget how much would go to each department. I feel like our model is the money goes straight to the department and then Adilas sends a little extra to the department and rarely does Adilas get any type of money kickback.

8.      Would you want to see something different (company structure)?

See above

9.      What products and services do you think we should focus on?

I feel like we have a couple options we could look into.

-       Focus on improving something that we currently have but improve it/extend it (sub inventory/mini conversions).

-       Build out completely new vertical with custom interfaces that are industry specific (rentals/reservations, campgrounds, etc), our own white labels/themes.

-       User experience - Add more settings/interfaces to help new clients feel like they aren’t over paying or help them not be overwhelmed with all the options/features Adilas offers when they first come on the system. Could be limiting access to only specific main Adilas features, or creating a basic, intermediate and advanced page to limit functionality that is available for those pages.

10.   Do we want to continue to internally fund this or seek outside investments?

I think it depends on how quickly we want to be able to grow things. Getting outside funding allows for more hands-on deck to try and get a quicker growth. If time isn’t a major factor there isn’t really a rush for trying to move things along with that extra funding.

11.   What do we need on the management side? What do we need in leaders and managers? Traditional managers? Or more of a Board of Trustees or Advisors?

I think we need vision and direction. We need a group to push us to reach that vision with attainable goals. Those leaders would guide and motivate others and get buy in. I feel that the leaders/managers from the different departments should make up the admin board to help keep and adjust the vision as needed to allow Adilas to grow in a balanced fashion.

12.   Do we want some sort of internal loyalty system or revenue sharing plan?

I feel like the loyalty system/revenue sharing could be good as a bonus for any quarterly profits the company has made and hasn’t spent, but feel like it should only be on net profits.

13.   How do you play? What is your time commitment, scheduling, maxes and mins (hours per week/month)?

However I am most useful, 35-50 hrs/week.

14.   What business verticals are you interested in?

I’m game for anything.

15.   Where would you like to focus?

I want to break up the features in Adilas to offer clients smaller, bite size pieces of Adilas (mom & pop shops) up to the platinum level where they have access to all of Adilas’s features.

16.   What are your thoughts on look and feel for the application interface?

I love the direction things are going with having designated designers who think about and draw up the front end designs. I want to keep that going to make things simpler so we can hide features the users don’t need.

17.   Let’s talk pricing. What are your thoughts?

I would like to work toward a tiered pricing where they have the ability to select the base option where they would be able to have access to 3-5 of the basic players in the system, then they could add additional tiers to add more features at higher price points. I think this will allow the entry level companies to feel like they aren’t “over spending” for the basic stuff they need the system for and we can charge a bit more for what our software niche is, that others don’t offer, all the key players in a single system.

18.   What about marketing and sales?

I don’t have much experience in this area, but think they are important if we want to grow. I think that word of mouth is ok, but it will only get you so far and definitely is slower.

19.   What about code, databases, automation, testing, deployment, and other techy stuff? Back-ups, redundancy, mirrors, servers, clusters, etc.

Again, here I feel like based on pricing and their willingness to be beta testers or secured systems we could have 2 clusters. 1 that gets the beta stuff and maybe doesn’t have as intense a redundancy system, and a 2nd that has the complete redundancy and an older stable code base that only gets updated quarterly. Those in the 2nd group would pay extra for the added redundancy, support, etc.

20.   What are your thoughts on education, training, and add-on services?

There is a correlation between the amount of education/training needed and the user interface complexity. I feel like the more simple the UI is the less training/education is needed; the more complex, the more training is needed.

If we break things down into easier interfaces then the less training people will need when they join the system and their employees won't be as overwhelmed learning a new system. Then as they are comfortable, they may seek for more functionality, they move to the next level and they won't need as much training to get from A to B.

21.   What about custom and API sockets and integrating with other external products?

I think that allowing companies to customize things is a niche that is unique to us and something that we are known for. I think there are ways that we might be able to manage our custom code/projects a bit better. Perhaps we set some sort of contract that would mark the custom projects as legacy after a certain point and no longer maintained and they would need to pay to have it refreshed, that way we don’t have to keep supporting it on our own penny after say 2 years).

22.   If we do a trust what are the other rules or parameters, we want to set up? Is it just time, is it time and money, is it investment, what is necessary if you want to buy in? How do we determine who is active and not active? How many virtual family members do we want? Etc.

I feel like there should be some sort of board that would have a vote on business decisions, future board members, and determining who is still considered active. I feel like time and expertise should play a major role in determining who should be on the original board.

23.   Where do you see us (as a company) in 1 year, 5 years, 10 years?

Next several months – create a business plan of action/vision

1 year – practices and policies in place that will enable us to grow and reach that vision.

5 years – Expanded to a few verticals to bring on enough accounts to hit Steve’s goal of everyone making 10k/mon.

10 years – See Steve and Brandon partially retire.

24.   Where do you want to be in 1 year, 5 years, 10 years?

1 year – Trustee, being major contributor to coding in whichever direction we are moving.

5 years – Maybe move into more of a management role, overseeing the different developer departments (UI, Internal Dev, Custom Dev, Testing, Bugs, etc).

10 years – same.

25.   Other ideas and thoughts?

I feel like I tend to be against any type of a style/developer guide that everyone must follow in order to code in the Adilas system. I feel like there are benefits to doing that, but feel we aren’t ever going to be big enough for that to really matter. It would be useful if we are constantly hiring new developers and have a large number of developers and multiple teams (30+), but having under 15 developers and usually only bringing on a few developers a year, I feel like we can just defer to a cold fusion sites for that documentation. I recommend they refer to Adobe’s help docs, cfdocs, learn cf in a week, or reach out to one of us. I think it more effective to let them focus on writing code and then have them clean things up as needed during code review if it is confusing/hard to follow. It might take a few projects for them to get a feel for things but it give the developer the autonomy. I know things are a little more sensitive with the front end code (html/css/js/jquery), but as far as backend stuff goes I think it is a waste of time and money for building and implementing the style guide.