Lean Sales Up - Making value from product conception

JORGE SILVA, 10Pines

This experience report is oriented to all who are worried about the gap existing between sales and execution of an agile project. Bringing the concepts and principles of lean model to sales, we had created a practice called "Lean Sales Up", which integrates the pre-sales phase with the traditional agile development cycle.

Unlike the traditional pre-sales practices, ours helps to add value to the customer beginning at the conception stage; promoting closing the deal, while at the same time making the development team happier.

1. INTRODUCTION

The process of pre-sales can be painful, long and expensive. It’s the first negotiation stage. You have to convince sales prospects that you are the best option and help them to set the always-high expectations. Often, this leads to overestimation, unrealistic promises about scope and deadlines, and a lot of non-healthy behaviors that pollute the relationship at the beginning.

When I was working in traditional companies, every time someone talked about the presales stage, I used to think things like, "How many impossible commitments has this salesman promised?", and, "How much time has he trimmed out from our estimates?"

The problem arises when you separate sales from development. Every unrealistic commitment the salesman does becomes a weight over the developer's shoulders, and since the salesman needs to sell, he will use every single means to do it (e.g. wishful thinking and optimism).

But, what happened when I started my own company? I had to sell too. And there began the conflict between my presales prejudices and my new reality. A well-done presale is indispensable to keeping your company alive. In this experience report we explore how at 10Pines we redefined the presale process to blend it with our development process and company values.

After five years of continuous work and improvements, our new process ends up with those three heuristics:

- **Explain the agile way**. Not everyone knows about agile. Explain what it is to your customer!
- **Discover your product**. We created the "Product discovery", a powerful workshop used to identify what to build adapting tools like story mapping [1], elevator pitch [2], among others, to fit together in a smooth way, in harmony with pre-sales.
- **Know the Psychology of the Customer**. Challenge your customer’s ideas; understand how they think, and more importantly how they decide.

2. BACKGROUND

In 2009 we founded 10Pines looking for a new way to create software inspired by this song: “A house with ten pines, to the south there is a place. Right now I’m going there, cause I can’t take it anymore, to live in the city...” [3]. We were four people tired of working for companies who saw software only as a mean to get money. For us, software was the end/goal by itself. We were tired of living in the city of big companies and decided to create our own company based on two main principles:

1. Technical quality: People passionate about programming and software development. People who strive for excellence in his solutions, and who are always learning, improving and expanding his knowledge
2. Human quality: By this we mean honest, responsible and critical thinking people. People for who money is not their only purpose. People who strive to change the world into a fairer and better place.

Based on these principles, we created a company with these practices among others:

- We have an open book policy. Anyone can know how money is spent and what everyone’s salary is,
- We have implemented a decision process based on consensus,

Jorge Silva, Av. Alem 693 Piso 5 depto. B Buenos Aires, C1001AAB Argentina; jsilva@10pines.com
Copyright 2015 is held by the author(s).
• We reduce the difference between the founders and the employees so that every worker at 10Pines is treated like a partner,
• We share profit with all 10Pines employees, and
• Employees manage and coordinate their own schedule and workload, based on responsibility and trust.

As you might expect, the traditional pre-sales process was a huge conflict with our company model primarily because in traditional companies:
• Prices are only known by the salesperson and the client (and the company owners),
• Client expectations are often sweetened to close the deal,
• Project scopes are often defined without development team involvement,
• Deadlines are unrealistic,
• The development team is not committed to customer’s goals, and
• The gap between sales and development is such that conflicts are somehow unsolvable.

In the beginning we naively thought that the most important thing was development so specialized sales people were not necessary at all. In 10Pines, we were all software engineers so this misconception was reinforced even more by our dislike of traditional sales practices. We didn’t have and we didn’t want, either, any specialized sales people because none of us liked only selling and we didn’t want the tension between the sales department and the developers.

But, as time went by, we realized that effective selling was necessary to keep our company alive. Key to our success and business integrity was to sell in a way that supported our company values. So we came up with two ideas:
• Developers must sell. If you want to sell it right, sell it yourself.
• We need to integrate sales into development. So our development process will include a new presale step. Therefore every time we talk about development, we imply a pre-sales stage.

Based on this reasoning, over the past 5 years we have crafted a process with a series of concrete practices to achieve this integration between our sales and development process. Completing the inter-disciplinary ideal scrum team by adding the sales role.

3. OUR STORY

In 2009, one of our first leads was a medium-sized company with a lot of growth opportunity. In the beginning we wanted to be technically perfect, focusing very strongly on the tech side while ignoring some important business issues.

We wanted to convince our potential customer we were the best option. So after two months and a lot of meetings, talks, negotiations, etc., we came up with a big proposal: a document of 20 pages including not only the price, but a lot of technical stuff that we were trying to anticipate for the future. Later, we realized that most of the items in the proposal were useless to anticipate, but were helpful in making the client trust us since they exposed our analysis skills.

Consequently, we won the project but it was really hard and it took a lot of work.

3.1 Problems

After 8 months, we finished this project and, in a retrospective, found these items to work on:
• Profit. The project didn’t generate the profit we expected. In fact, we had to sacrifice our earnings in order to complete our initial commitment/engagement.
• Estimation. We realized that we invested a lot of unpaid time preparing the original estimation. Our proposal was detailed to excess.
• Scope. This is one of my favorites. The scope we finally developed was half the one initially defined in our proposal. The changes asked by the customer, the gap between our assumptions and the reality we discovered, and all the activities that we didn’t contemplate before we started, forced us to trim down the original scope.
• Satisfaction. I think as a result or as a consequence of profit we lost, we improved the customer’s satisfaction. We bet on a longer-term relationship, sacrificing our profit to make the customer happy.
All of these problems could have been avoided, if in the pre-sales stage we had been more aware. So, we learned. I call this first experience the “life before the Product Discovery” era. After this blind age, we started trying to improve our development process in general and our pre-sales process in particular.

To do this, we initiated a series of practices that helped us working on the items we uncovered in our retrospective.

Although our first experience looks avoidable, I feel it was necessary for us to go through this to get the actual state of knowledge or learning, in which we now understand business in a different way. Because of this first experience, we were better prepared with a lot of new validated knowledge. One wouldn’t be the same person without his past or skipping it.

3.2 What we did
To start facing our detected problems, and looking to master in what we do, we started to try a series of practices to improve the success of our projects.

One clarification I would like to point out is I will use the word “product” as a representation of the “thing” created after developing software. In terms of software, it could be actually what is called a product, or it could be an internal application for a company which is not going to commercialize the “product”. So product for my something wider, I will use this word as a meaning for result of something and not limited to something to commercialize.

Below I present a review of the ten most important practices we use and how are they related to the final redefinition of our process development.

1. Short sheet dilemma <explain the agile way>: the short sheet dilemma is a metaphor, which manifests a contradiction about resources. Imagine you are in a hotel, with a new bed for you, and it turns out that the sheet they gave you is kind of short. You can only cover your chest, leaving your feet exposed; or cover your feet leaving your chest out. But you can’t cover both. This dilemma was useful for me to explain the eternal battle between scope, time and money.

What I did was to present this idea to our customer, and explain that we can’t think about a project whose scope is defined 100% with a strong deadline and a tight budget. You have to choose one when you are on real software development, and this metaphor help me to communicate this. I also use the iron triangle to depict this concept:

<table>
<thead>
<tr>
<th>Scope/Scale (Features, Functionality)</th>
<th>Resources (Budget, Labor)</th>
<th>Schedule (Time)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quality</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. Marshmallow Challenge <explain the agile way>: One of the hardest things to do when selling software development is to explain why is it difficult to build software. People in general tend to think that software is easy to build, changes are easy to make, and the right and correct way to do it is in a production line paradigm (Fordism way).

After our first project, I realized our agile way of development wasn’t as clear or intuitive or familiar to the customer, as I thought. I felt the necessity to explain this a bit more. My idea was to illustrate why it’s important to quickly try things, and why it’s important to build iteratively and incrementally; especially when you don’t know exactly what to build.

After thinking a lot with my colleagues, as truly TED Talks fans, we remembered a really good talk, “Build a tower, build a team”. Here Tom Wujec introduces a game called “The Marshmallow Challenge”. The Marshmallow Challenge is a remarkably fun and instructive design exercise that encourages
teams to experience simple but profound lessons in collaboration, innovation and creativity. The task is simple: in eighteen minutes, teams must build the tallest free-standing structure out of 20 sticks of spaghetti, one yard of tape, one yard of string, and one marshmallow. The marshmallow needs to be on top. Tom Wujec surprisingly concludes that 5-year-old kids are the best builders. Why? Because they prototype a lot, plan less, and learn more.

We thought that it would be a really interesting exercise to use it as an educational tool to explain how to build software. And we tried it.

Our results were really interesting. The first time we did it, we involved all of the people who worked on the project and their bosses. Initially this caused some inconvenience, but as time went by during the exercise, people started feeling more free. Anyway the real goal of this exercise was to illustrate how software development works, and why we chose to work in an iterative/incremental way. It was a great success and it helped us to have a reference and illustrative point. After our first experience using this exercise, we started to use it at the beginning of each project.

3. **Business workflow <discover your product>:** One of the most important things to do when starting a project is to identify the problem to solve. A problem we faced using Jeff Patton’s Story Mapping technique (explained later on) was how to identify the main/first activities to start working on and how they where connected each other. To address this problem, we started using a simple business workflow. This is a simple graph. Each node represents an important activity in the business model we are going to work on. Connections represent dependency between activities. It’s important to keep in mind that this is a high level view. You have to be very focused on the business, leaving aside discussions about technological or incidental issues. Our goal is to clearly identify the business to work on and its core. This is in order to propose a solution to the problem without any vices.

   A common anti-pattern is trying to “solve the current solution” instead of thinking out of the box to create a new one.

   Once we’ve done this activity, the nodes will be used as the main activities for a story mapping activity. We find them a good and easy starting point no just for Story Mapping but to understanding the problem. So we use this as a preamble to the next step, Story Mapping and to get a big picture of the problem and start getting wet with it.

4. **Story mapping <discover your product>:** Although the story mapping is a well-known activity, we have redefined it to meet our needs and context. Our custom version of story mapping was created in several iterations across many projects over about 4 years. The first time we tried to do a user story mapping just like Jeff Patton recommends. But we had trouble getting all the right people involved (about 10 to 15 people) for a full day (about 8 hours). It was really hard for me to convince the customer to do this, besides the fact it would be my first story mapping experience. So after several talks trying to convince him, we came to this compromise: get all the people but for just 4 hours. My idea was that at the end of the 4 hours, they would hungrily ask for more. We started early in the morning. The first step in Jeff Patton’s story mapping recipe is to collect features, a.k.a. activities. But in our context it was really difficult to start thinking about user activities. So instead, we came with the idea of making a graph with the actual business process (see previous section “Business workflow”). Since we were about to develop an application that was going to go with an existing process, we wanted to know and agree upon that process. That’s why we made our first customization and added a first step: “business process graph” or “business workflow”. Each node in the graph would later be expanded into our features/activities.

   The workshop continues adding details to each activity and identifying user, frequency and value of each activity. Then we add more details, breaking each activity into tasks.
Everything went great at our first story mapping workshop, until we arrived at the last part, where we needed to slice our story map into releases. I thought that it was too early to define releases, so I preferred to break it down into three different levels of priority. And that was what we did. We defined 3 levels of priority for the business. The highest priority should be our MVP (minimum viable product) [4]. Since at this moment, everything seemed important and it was hard for the customer to think about the application without all these features we had identified, I made up a strategy to force people to truly consider each task priority. In that workshop I remembered some statistics I had heard about the features actually used in software; and it says something like, applying the Pareto principle, 20% of the features are the most frequently used, providing the 80% of business value. So I counted all the tasks we already had on the board and took 20% percent of it, and split the rest evenly into priority two and three. So the result was this message to the customer, “You guys have 20 tasks you can choose to put at priority 1; 40 tasks you can choose for priority 2; and the last 40 for priority 3.” This statement, followed by the statistics and the chart, was great way to align people, get them focused and be really critical about what should be built first (given the fact that 80% of the people believe any statement that is followed by any statistic).

Obviously, it was impossible to accomplish all of this work in four hours. But when we were close to the last 20 minutes, all together we agreed on the importance of continuing to work on this, the next day. This idea emerged without me bringing it up, so I was really pleased. It worked well for us to split the workshop in two days, because it was useful go home and think about all we were talking on that day, and let the entire learning soak in.

So, we ended up with our first story mapping experience customized with three tweaks: “business process graph”, prioritization instead of releases, and a two-day workshop.

After this project, in the customer retrospective, it came out that the story mapping was one of the best things we did. So after this experience, I no longer needed to convince them to do story mapping. From then on, that customer has been the first supporter of and evangelist for this practice.

6. Elevator pitch <discover your product>: Sometime after working with the story mapping workshop, we started feeling we were losing the “why” of the project. So we thought it would be good to have some kind of summary of why we were building what we were building. Brainstorming again, we remembered the typical business startup exercise: The Elevator Pitch.

The idea is pretty simple: image you have a great idea for a gadget; a computer with just a touchable screen which you can carry everywhere you want. Image you are in a hotel in California, floor 21, waiting for the elevator. You press the go-down button, ready to go outside to know the city. Suddenly the elevator doors open and you see in front of you Steve Jobs going down in the elevator. The only thing you think is, “I have to convince Steve to build my idea! This is my chance.” So you only have the time of an elevator trip to tell him your idea and convince him is a really good one to invest on it. Hence, the elevator pitch goal is basically to summarize your idea in a few lines, clearly pointing the value you add and the principal concerns that make your project a great project to work on.

So we came up with a template to summarize the idea of what we were about to build. This is helpful to quickly start being aligned.
I like to think of it like our manifesto for our project, and that means we cannot betray it. It's useful to:

- Rapidly identify what is in or out in our project
- Have a common criteria of what to include (or not) in the scope
- Use it as a north star for finding the minimum viable product (MVP)

7. Challenging customers’ ideas and pre-conceptions <Know the Psychology of the Customer>: One of the more valuable skills we have at 10pines is our analysis skill. Sometimes it can be hard at first to speak up and challenge the “big ideas” your customer has, but if she is a reasonable person, she will appreciate it since this means that you are involved in their problem, you understand it, you have experience and you can see problems.

In early 2014 a startup company contacted us. They wanted to build a really interesting social network site for entrepreneurs. But they did not have an idea how to build it in an agile way. Even they were really convinced that they should build the entire site before launching it publicly. So, one of the first thing we did was to start challenging this plan and pointing out from our perspective and experience where there were the weak points in the plan. At the beginning was a really tough discussion but while we were explaining our justifications, they saw we were really solid on what we were saying, so they began listening and changing their minds.

At this moment in a project you might think that is not the best to contradict your fresh new customer in case you can lose them (actually they were deciding which provider to work with at the time). But I think it's better to show that you know what are you talking and have strong convictions about how to work, than to flatter your potential customer.

8. Psychology of pricing <Know the Psychology of the Customer>: The price of the proposal is one of the most important concerns in the sales (if not the most). And this is something always make me doubt. I’m never sure how to show the price because I truly believe that the way you do so conditions perceptions. The customer could find my proposal more or less attractive based on how it’s presented.

Because of that, something really important and useful to understand is how the consumer’s psychology works.

One day in mid May 2012 one of the team members showed us a TED Talk called, “Are we in control of our own decisions?” by Dan Ariely [5], based on a book called Irrational Predictably by same author. This talk was inspiring. In that lecture he talks about the effect of how we present options. So, for example, a country could be one of the top 3 in organ donation or not, based on what the default option is in the donation form. And not just that, in some experiments he did, he found that having the choice of choosing between different options is better than just one option.

Another edge of psychology of sales is anchoring [6]. Anchoring is a cognitive bias [7] by which people select a small or one variable to compare, and usually a reference element. When you present your price, the customer is going to compare it with something else he knows better. You can help them decide by giving them a familiar point of comparison. You can even use this to close the deal. It's important to give them something from their business domain, for example if you are dealing with Starbucks, the unit to compare might be amount of “Alto Caffe Latte”

Below you can see a proposal to a company that sells coffee; we present 3 different options for our services, each one translated to how many lattes the work represents.
Some final thoughts on how to best present options taking into account psychology of customers:

1. Always give at least 2 options
2. Try to use some parameter known to the customer
3. Always recommend one option

9. On how to present the project development <Know the Psychology of the Customer>: Another important issue to address is any customer fears about the project ending up out of budget. The direct consequence of any such fear is that she will choose a fix priced contract every time she can over a time and material one if she’s worried about running out of budget. Obviously, the fixed price contract places all the risk on the development team. To get a balance, we found a mixed approach that works really well for us: we set a fixed amount of iterations defined by the development team, plus an amount of variable iterations, which the customer can define (even while the project is ongoing).

   This way we assure the customer gets to a MVP with a fixed price by the fixed iterations (so the fixed iterations should be to deliver the MVP which is easier to estimate than the complete product). But we can also continue working after that with mutually agreed upon variable iterations, thus avoiding the estimation error caused by uncertainties that existing at the beginning of a project. Also, the customer has the control over when to stop, removing their fear of run out of budget. And the development team has a fixed floor on which to build to.

10. The icing on the cake <Know the Psychology of the Customer>: Last but not least, be aware of aesthetics in your proposals. Be motivated to innovate and try new things when you present, for example use new tools like prezi instead of PowerPoint, or present videos, an make analogies to other disciplines, etc. If you are selling great software, you should be great about presenting your ideas.

3.3 Results

Thanks to all these experiences and knowledge learned, at 10Pines we have redefined the presale process to blend it with our development process. What this means is that there is no longer a presales stage separated from the development cycle; instead we just merged each one in a complete cycle, understanding that presale or sales is also part of software development. Doing this, both sides can be aware of the complexity of both activities, sales and development; and on the other hand we erase the border between the two areas, creating a team more complete, capable not only of developing software but also selling it and explaining to the customer why we are the best option.
Initially, before we understood a good pre-sales process, we started with this classical agile process:

![Classical Agile Process Diagram]

There was no trace of how we came to get the project; we just got to start developing something that has kind of fallen from the sky into our laps, skipping all the previous history, which is part of the essence of the product to build.

After we had learned how to do pre-sales, we ended with this:

![Pre-Sales Process Diagram]

In this process the presales stage, the phase where the product is conceived, is part of the development cycle, hence it’s a more holistic way of development with more chance for being successful.

I believe that this new/rewdefined cycle is pretty solid and works well for us. It has had big payoffs and has led to growing business opportunities.

For example, one of the accounts we had when we started to refine our pre-sales process was an Argentinian subsidiary of a multinational company. We started working with them in mid-May 2010. By 2011, they introduced us to work with their Chilean subsidiary. A year later, we move on to do work in Mexico, the location of their company headquarter. And now, as of May 2015, after 5 years of continuous and hard work, we have developed for them more than 10 internal products and worked on over 50 projects in 4 countries.

This gives us the confidence to apply these practices with new clients, from the start of any relationship, as a consequence of applying these pre-sales practices with continued success.
4. WHAT WE LEARNED

My lessons learned could be summarized by the following points, which are at the core of this experience. Each point is a conclusion, which has emerged from the experiences I’ve lived through over many bid requests. And these points have now turned into a successful path I use to win any bid:

1. **Explain the agile way.** The customer rarely knows about software development, and it is even worse when it comes about agile software development. And it’s perfectly fine, that’s why they call us. So, it’s important to explain how we are going to work together, what an iteration is, what we expect from them, etc. We have to explain to them how we build software, BUT (very important) in a simplified and understandable way.

2. **The power of the “product discovery”.** The “product discovery” is a tool that we created at 10Pines to quickly discover what has to be built. Focused on finding the MVP, this kind of project chartering, covers several activities we accomplish with the customer’s team. We combine tools such as User Story Mapping, elevator pitch and impact mapping, among others, to get a prioritized MVP which leads us to understand the business, get a common idea between all people involved, and have something reasonable to estimate.

3. **Know the psychology of your customer.** As we saw in the previous examples, to know how people think in general, and how a person in a customer role decides, is really important when it comes to propose how to work. Always keep in mind: challenge your customer, think how best to show prices, propose several options for prices, and take into account that having an innovative presentation as a proposal is a plus.

Indeed, this experience report represents learning from several projects and customers. The current process we now use took about 3 years to develop. But, like software, it is a process that is never over since we continue improving it with every new project we start working on.

Finally, my last message is to encourage you to improve every day, work on anything new every time you can, try and continue learning. All our learning was a process that emerged gradually and worked by taking baby steps.

5. ACKNOWLEDGEMENTS

I want to thank the following people:
Dario García, he was my mental guide. Thanks for all the time you dedicated to work with me on this experience report, also for reviewing and suggesting a lot of ideas and changes. Also, several workshops arose from a lot of talks we had together, thinking how to make a better world.

Hernán Wilkinson, for introducing me to Rebecca for an opportunity to write this report, also for reviewing this work and for always trusting me and helping me whenever I asked him to lend a hand.

Emilio Gutter, who revised this work and more important, who was the inspiration for the name of this experience: Lean Sales Up!

Alejandra Alfonso, who introduced me to the idea of User Story Mapping and helped me when we made the first Story Mapping workshop in Alsea. That was a big help.

Gisela Decuzzi, which was my teammate for several of these experiences and always was ready to help on these workshops.

All 10Pines crew, who allow me to take the time to write and who are my final supporters.

Also my shepherd, Rebecca Wirfs-Brock who was always willing to help, reviewing my work, suggesting changes and following up.

REFERENCES
[3] Original lyric: “Una casa con diez pisos, hacia el sur hay un lugar. Ahora mismo voy alla, por que ya no puedo mas, vivir en la ciudad…”
[5] https://www.ted.com/talks/dan_ariely_asks_are_we_in_control_of_our_own_decisions