

# Pretotype It

Make sure you are building the right *it*  
before you build *it* right



**Alberto Savoia**

*Second Pretotype Edition*

*October 2011*

Copyright (c) Alberto Savoia 2011

## This Is Embarrassing

This is not a “proper book.”

Writing and editing a proper book on the subject of pretotyping would take months. I would love to write that book, but at this time I have no indication that such a book would be worth writing. Most books fail in the market, and most of them fail not because they are poorly written or edited, but because there aren't enough people interested in them. They are not *the right it*.

What you are reading now is a *prototype edition* of the book. I wrote and “edited” it in days instead of months, just to test the level of interest in such a book. I had a few friends and colleagues review it, but don't be surprised if you find typos, misspellings, bad grammar, awkward formatting and all sorts of *mistakes*.

Releasing it in its present state is not easy for me.

The toughest thing about pretotyping is not developing prototypes, that's the fun part. The tough part is getting over our compulsion for premature perfectionism and our desire to add more features, or content, before releasing the first version. The tough part is getting our prototypes in front of people, where they will be judged, criticized and – possibly – rejected.

Reid Hoffman, founder of LinkedIn once said: “If you are not embarrassed by the first version of your product, you've launched too late.”

I *am* plenty embarrassed. I must be on the right track.

## Introduction

At this very moment, millions of people across the world are pouring their hearts, souls, hopes, dreams, time, money, and energy to develop new ideas that once launched will flop miserably.

At this very same moment, a *much* smaller number of people are developing new ideas that will turn out to be successful – and some of them will be extremely successful: the next iPod, the next Google, the next Twitter.

What group are you in?

Most people believe that they are working on a winning product, but we know that can't be true.

Most new ideas fail, and predicting the actual market success of a new idea with any degree of confidence is next to impossible. Some brilliant “*can't fail*” ideas turn out to be gigantic flops, while some crazy “*who'd want that?*” ideas turn out to be spectacular successes.

Some people and organizations may be better at predicting success than others, but even the best venture capitalists, investors or entrepreneurs regularly invest way too much on *the wrong ideas* and frequently choose to invest nothing on *the right ideas*.

If all we have is an *idea* for some new product (or service, or book, etc.), the best thing we can do with that idea is collect *opinions* about its usefulness or market potential. Ideas are fuzzy and abstract; opinions are subjective and even more abstract; when you combine the two you get a big fuzzy ball of abstractions and opinions. Not much to go on.

Traditional prototypes can help to test and validate the market potential of new ideas more concretely and objectively than ideas and opinions. In many cases however, the development of a “proper prototype” is too difficult, expensive and time consuming. It's normal to invest weeks, months or years, and hundreds of thousands or millions of dollars to develop prototypes. Furthermore, most prototypes are built to answer

questions such as, “Can we build it?” or “Will it work as expected?” instead of focusing on questions such as “Should we build it at all?” or “If we build it, will people buy it and use it?” Unless you can answer the latter questions positively, the former questions are of little importance.

Prototypes can help you fail faster, but often not fast enough or cheaply enough. The more you invest in something the harder it is to let it go and admit it was the wrong thing. Once you have a “proper prototype” working, it’s tempting to work on it a little longer and invest in it a little more: “If we add this one feature I am sure that people will finally use it.” Prototypes often turn into *productypes* – a prototype gone too far – and you can kiss fail-fast goodbye.

Between abstract ideas and “proper prototypes” there are *pretotypes*. Pretotypes make it possible to collect valuable usage and market data to make a *go/no-go* decision on a new idea at a fraction of the cost of prototypes: hours or days instead of weeks or months, and pennies instead of dollars. Pretotyping helps you fail fast, recover fast and leaves you plenty of time, money, energy and enthusiasm to explore new tweaks or ideas until you hit on something that people seem to want – the rare and wonderful *right it!*

A lot of what you will read in this book will seem obvious to you. But before dismissing it too quickly, look around at all the products, services, apps, books, etc., that are launched every day only to fail soon after. Most of these new products don’t fail because the people who worked on them were stupid, lazy or incompetent, nor because they were built or marketed poorly, but because they were not the right product to start with!

Chances are that, unless you are just starting in your business or industry, you can look back at some of the products you have worked on and identify quite a few that in retrospect were not the right product. That’s certainly the case for me; I’ve been lucky enough to work on products that turned a few months of work into millions (and eventually billions) as well as products that turned years of work and tens of millions into “dust.”

Even though this version of the book is itself a pretotype edition (I'll elaborate on that later), it should have enough “meat” to be worth your while. I sincerely appreciate the fact that you are reading it: please send me your feedback – I need data to decide if I should invest the time required to turn this *pretobook* it into a proper book.

Thank you,

Alberto Savoia (asavoia@gmail.com)

August 2011

Mountain View, CA

# **CHAPTER ONE**

## **The Right It**

The title of this book is “Pretotype ***It***” and the subtitle is “Make sure you are building *the right it* before you built *it* right.”

I will explain and define pretotyping very soon; before I do, however, we need to address the following question:

What is this *it* I speak of, and why is it so important to have ‘*the right it*’?

### **What is this *it* I speak of?**

In the context of this book, *it* can be a new product, a service, a book, a startup, a charitable organization, a video game, an innovative type of boat, a new musical instrument, a revolutionary genetically engineered hypoallergenic hamster, ...

*It* is something that does not exist yet, but you have been thinking about it and would like to – or have to – create it and bring it to life.

*It* is something important to you, and creating *it* will require a non-trivial combination of your time, effort and money, as well as a considerable amount of your energy, drive, enthusiasm and commitment.

Ideally, *it* is something that you are deeply passionate about, but it’s OK if *it* is just something you have to do as part of your job.

## What's so important about having 'the right it'?

The odds are heavily stacked against the success of your *it*. Hopefully this is not news to you. I'm sure you've heard statistics similar to the following many times:

- 90% of all mobile apps don't make any money
- Four startups out of five lose money for the investors
- 80% of new restaurants close within one year

Most new *its* fail. Unless you've been given some form of divine dispensation, you have the same odds as everybody else. Chances are the *it* you are currently thinking of will not succeed – unless your *it* happens to be the rare *right it*.

If you don't have *the right it* then, by definition, you must have *the wrong it*. One of the most wasteful and costly things you can do is to continue working on the *wrong it* hoping you will be able to make it a success through sheer willpower and effort. Unfortunately, this happens very rarely: generally speaking, there is no amount of time, effort or money that can make the *wrong it* succeed.

Movies are a good example of how next-to-impossible it is to turn the *wrong it* into a success at the box office. If the movie concept (the *it* in this case) is not right, no star directors, actors or \$100M+ budget is going to turn the movie into a success (e.g., "Ishtar", "Heaven's Gate", "Howard the Duck".)

On the other hand, if you have the *right it*, everything becomes much easier and the odds of success swing in your favor. In the case of movies, this would be a movie with little or no budget, an inexperienced director, no name actors and no expectations, that turns into a blockbuster (e.g., "The Blair Witch Project", "El Mariachi", "Paranormal Activity".)

Having *the right it* is essential. Most people or organizations don't have unlimited time, energy or money to sustain a long string of slow and ex-



pensive failures caused by chasing *wrong its*. The goal of prototyping is to weed out *wrong its* and find the elusive *right it* with the minimum investment of time, money and effort.

## Why do I keep writing ‘*it*’ in bold and italic?

The concept of prototyping is applicable to a wide range of ideas for products or services – software, hardware, websites, games, soft drinks, hard drinks, books, movies, etc. Since it’s cumbersome to write (and read) things like “If your product or service is ...”, I decided to simply refer to whatever your idea is as *it*.

Throughout this book, I write *it* in bold and italic to differentiate *it* (your idea) from the pronoun ‘it’. Since this book is – at least at this time – a prototype itself, I might have missed a few *its* here and there. Hopefully it will be clear from the context when I am referring to your *it*.

As you read on, some of good acronyms and mnemonics for *it* are:

- **idea on the table**
- **idea to test**
- **innovation to try**

# **CHAPTER TWO**

## **Pretotyping**

## What Is Pretotyping?

Now that you have a rough idea of what I mean by *the right it*, we can give pretotyping a proper introduction. The best way to do that is by sharing with you the two stories that got me thinking about this whole thing: the IBM speech-to-text “experiment” and the Palm Pilot “experiment”.

### The IBM Speech-to-Text Experiment

I first heard this story during a presentation at a software conference a few years ago. I am not sure how accurate my description of the events is; I probably got a few details wrong, but in this case the moral of the story is much more important than the details. With that caveat out of the way, here’s the story as I remember it.

A few decades ago, well before the age of the Internet and before the dawn of ubiquitous personal computing, IBM was best known for its mainframe computers and typewriters. In those days, typing was something that a small minority of people were good at – mostly secretaries, writers and some computer programmers. Most people typed with one finger – slowly and inefficiently.

IBM was ideally positioned to leverage its computer technology and typewriter business to develop a speech-to-text machine. This device would allow people to speak into a microphone and their words would “magically” appear on the screen with no need for typing. It had the potential for making a lot of money for IBM, and it made sense for the company to make a *big bet* on it.

However, there were a couple of major problems. Computers in those days were much less powerful and more expensive than today, and speech-to-text requires a lot of computing power. Furthermore, even with adequate processing power, speech-to-text translation was (and still is) a very difficult computer science problem. Tackling it would have re-

quired a massive investment – even for IBM – and many years of research. But everyone would have wanted such a device. It would be a sure-fire hit. Or would it?

Some folks at IBM were not convinced that all the people and companies who had said they “wanted and would definitely buy and use” speech-to-text machines would actually end up buying them. They feared the company would end up spending years in research and lots of money developing something that very few would actually buy: a business disaster. In prototyping lingo: they were not sure that speech-to-text was *the right it*. After all, people had never used a speech-to-text system, so how could they know for sure they would want one? IBM wanted to test the business viability of such a device, but since even a basic prototype was years away, they devised an ingenious experiment instead.

They put potential customers of the speech-to-text system, people who said they’d definitely buy it, in a room with a computer box, a screen and a microphone – but no keyboard. They told them they had built a working speech-to-text machine and wanted to test it to see if people liked using it. When the test subjects started to speak into the microphone their words appeared on the screen: almost immediately and with no mistakes! The users were impressed: it was too good to be true – which, as it turns out, it was.

What was actually happening, and what makes this such a clever experiment, is that there was no speech-to-text machine, not even a prototype. The computer box in the room was a dummy. In the room next door was a skilled typist listening to the user’s voice from the microphone and typing the spoken words and commands using a keyboard: the old-fashioned way. Whatever the typist entered on the keyboard showed up on the user’s screen; the setup convinced the user that what was appearing on the screen was the output of the speech-to-text machine.

So, what did IBM learn from this experiment?

Here's what I've heard: After being initially impressed by the "technology", most of the people who said they would buy and use a speech-to-text machine changed their mind after using the system for a few hours. Even with fast and near perfect translation simulated by the human typist, using speech to enter more than a few lines of text into a computer had too many problems, among them: People's throat would get sore by the end of the day, it created a noisy work environment, and it was not suitable for confidential material.

Based on the results of this experiment, IBM continued to invest in speech-to-text technology but on a much smaller scale – they did not *bet the company* on it.

As it turned out, that was the right business decision. Keyboards are proving hard to beat for most text entry tasks. Thirty years ago most people could not type; but look at any office (or coffee shop) today and you'll see people of all ages and professions typing away on their laptops. In devices where a full-size keyboard is not possible, such as mobile phones, speech-to-text can be the *right it*, but otherwise the keyboard is still the device to beat. The keyboard is definitely *the right it*.

The IBM approach was ingenious, but what would you call it? The speech-to-text setup with the typist was not what one would consider a "proper prototype" – not unless they were planning to actually hide living and breathing typists into computers. They didn't prototype a speech-to-text system, they *pretended* to have a speech-to-text prototype, and used it to test users' actual reaction to the product. This way they were able to collect valuable market data based on actual usage instead of opinions, and they did that with a very small investment of time and money.

I thought that this was a very interesting and valuable approach, and that it was different enough from prototyping to deserve its own name (more about that later) and more study. But first I set out to find similar stories and uncovered another brilliant example.

## The Palm Pilot Experiment

The IBM speech-to-text story got me thinking about the concept of pre-totyping, but this next example is the one that convinced me that it was worth pursuing further.

Introduced in 1996, the Palm Pilot was a palm-sized digital device with four basic functions: a calendar, an address book, a to-do list and a simple note taker. The Pilot was the first successful PDA (Personal Digital Assistant.) But Jeff Hawkins, Palm's co-founder and one of the inventors of the Pilot, did not take the eventual success of PDAs for granted. Quite the contrary. According to a March 1998 story on Time magazine [emphasis mine]:

*Hawkins, 40, Palm's chief technologist and Pilot's creator, designed one of the first handheld computers, the GRiDPad, a decade ago. It was **an engineering marvel but a market failure** because, he says, it was still too big. **Determined not to make the same mistake twice**, he had a ready answer when his colleagues asked him how small their new device should be: "**Let's try the shirt pocket.**"*

*Retreating to his garage, he cut a block of wood to fit his shirt pocket. Then he carried it around for months, **pretending** it was a computer. Was he free for lunch on Wednesday? Hawkins would haul out the block and tap on it as if he were checking his schedule. If he needed a phone number, he would **pretend** to look it up on the wood. Occasionally he would try out different design faces with various button configurations, using paper printouts glued to the block.*

Here's a photo of the prototype Jeff built (you can see this artifact at the Computer History Museum in Mountain View, CA):



I can only imagine people's reaction when Hawkins pulled out a block of wood out of his pocket and tapped on it, pretending it was a working device. They must have thought he was crazy. Yes, crazy like a fox! That piece of wood with paper printouts convinced Hawkins that he was on the right track. He had answered the first, and most important, question: "If I had a Pilot, would I actually carry it with me and use it?" And his answer was a definite "yes!" He knew he had *the right it*; now he could focus on the next set of questions, such as: Can we build it this small? How much would it cost to build? How long would the batteries last? It was time to invest in building a "proper prototype."

The Palm Pilot was not just successful, it was *hugely* successful and had a tremendous impact. The Pilot was the predecessor to today's smart phones, and it all started with a little piece of wood – just like Pinocchio.



## Fake It Before You Make It

The speech-to-text and Palm Pilot stories have several things in common.

Both teams had doubts about the eventual usefulness and adoption of their innovation. It was a cool idea. It made sense. It solved a problem. But was it *the right it*? Would people actually use it? Jeff Hawkins, in particular, had just been burned by investing years to develop a product, the GridPad, that was “an engineering marvel but a market failure” (i.e., *the wrong it*) and was “determined not to make the same mistake twice.”

Because of their doubts, both teams wanted to test the usefulness of their idea with a prototype and collect feedback *from actual usage of the product* (as opposed to *opinions about the product*) before committing to its development.

In both examples, however, even the development of a “proper prototype” (a crude but functional version of the final product) would have required a lot of upfront time and a significant investment in research and development.

Their solution to the “proper prototype” problem was to *pretend* that they had such a prototype. In the speech-to-text example, actual hardware and software was replaced with a bit of trickery, and in the Pilot example it was replaced by Hawkins’ imagination – *fake it before you make it*.

I found these two stories striking because they are so different from the typical approach people and companies take when they have an innovative idea they want to pursue. Most people fall in love with their idea (their *it*) and assume it will be successful (*the right it*) so they just start building it. They *jump the gun* – so to speak – and begin by focusing and investing in the wrong things at the wrong time. More precisely, they invest too much too soon to develop a first version of the product with too many features, too much functionality and too much “polish.” They presume to know what people will want. They assume that if they build it right, people will want it. In most cases, these presumptions and assumptions turn out to be both wrong and costly.

## Pretotyping: A Word Is Born

The more I thought about the speech-to-text and Palm Pilot experiments, the more I became convinced that what those teams did was not only clever, but a crucial step in the process of developing new and innovative products. A step that most people skip and often end up paying dearly for doing so.

Over a span of several months, I shared these two stories with dozens of colleagues, friends, entrepreneurs, venture capitalists, engineers and product managers. Surprisingly, none of them had heard of these examples before. All of them, however, were similarly impressed by the clever solution of “faking it before making it,” and a good number of them slapped their heads and said things like: “I wish I had done something similar before sinking years and millions in my last idea.”

I realized that I had stumbled into something valuable and important that, while not new or original, was neither well known nor widely practiced. Quite the contrary. But this something did not have a term to describe it, and I thought it deserved and needed a name in order to become better known, discussed and more widely adopted. So I started thinking of a possible term for this concept. (Note: At the time I started thinking about pretotyping I was not yet aware of Eric Ries’ great *Lean Startup Movement* or the term MVP – Minimum Viable Product. More about the relationship between pretotyping and MVP later.)

Since a core element in both examples is the act of *pretending* (the IBM folks pretended to have built a speech-to-text system and Jeff Hawkins pretended to have a Pilot in his shirt-pocket,) the first word that came to mind was *pretendotyping* – yikes! My second attempt at coining a word was even worse. Since the core idea is to quickly test an idea even *before* you invest in building a proper prototype, I came up with the word *preprototyping* – double yikes! Fortunately, these two horrific words contained the seed for a much better term. By dropping a few letters here and there, I came up with *pretotyping*. Much better. The artifacts

that are produced by the process of pretotyping (e.g., Hawkins' wood block) would be called pretotypes.

I liked the words pretotyping and pretotype, but was I the first to use them? Perhaps someone else had already been using them and had some kind of "rights" to their use and meaning. I ran to Google and typed "pretotyping" in the search box. To my delight, Google came back with "Did you mean *prototyping*?" The search engine assumed that I had misspelled the word so it gave me a bunch of results for prototyping instead – a good sign. When I insisted that *I did not mean prototyping* and to please give me results for pretotyping instead, Google returned a relatively small amount of pages where people had misspelled the word prototyping. Searching for pretotype yielded similar results. The coast was clear. I had stumbled onto a new word that nobody else was using yet.

Even better, the associated domain names pretotyping.[com, org] and pretotype.[com,org] were also available. My first instinct was to pull out my credit card and buy all of them, but I realized that by doing so I would be violating the core message of pretotyping: I would be investing in something before making sure that it was worth investing in. Even though it would only cost a few dollars to reserve the domain names, the principle was at stake. I thought that pretotyping and pretotype were great words to describe a great concept, but would other people feel the same? I had to pretotype pretotyping.

Fortunately, as part of my job at Google, I have the opportunity to talk to a lot of people about innovation and give a lot of presentations on the subject to both customers and to other Googlers. So, along with the speech-to-text and Palm Pilot examples, I started using the words pretotyping and pretotype in all of my presentations, meetings and discussions. In every single case, people responded very positively to both the concept and the words. They were sending me examples of their pretotypes, asking me for suggestions on how to pretotype their ideas and were even lobbying with their colleagues and managers to pretotype ideas before building them. It looked as if I was on the right track.

One day, I was forwarded an email from the head of one of the biggest advertising agencies in the world who had attended one of my presentations on innovation. He thanked the organizers for the presentation, said that he and his team loved the concept of pretotyping and that "... the word pretotyping has now entered our company's lexicon."

That day I knew I had enough evidence that pretotyping and pretotype were the right words for the right concept, and I felt confident taking the next step and investing a few dollars to buy the associated domain names.

## Pretotyping Defined

Although I believe it's more effective to use examples to explain pretotyping, I also think that it's worthwhile to try and define it.

Here's a somewhat formal definition – the dry and boring kind you'd find in a dictionary:

***Pretotyping** [pree-tow-tie-ping], verb: Testing the initial appeal and actual usage of a potential new product by simulating its core experience with the smallest possible investment of time and money.*

Here's a less formal definition:

*Pretotyping is a way to test an idea quickly and inexpensively by creating extremely simplified, mocked or virtual versions of that product to help validate the premise that "If we build it, they will use it."*

Here's a very informal definition:

*Pretotyping: Fake it and test it before you make it!*

My favorite definition of pretotyping, however, is based on the subtitle of this book:

*Make sure – as quickly and as cheaply as you can – that you are building the right **it** before you build **it** right.*

## Pretotyping and Prototyping

Some people might argue that pretotyping is too close to prototyping both in spirit and practice and, therefore, there is no need to differentiate between the two nor to invent a new word. I thought about this issue a lot. The problem I see is that the term prototyping covers a huge range of the spectrum between the abstract idea for a product and the final product.

A prototype for a speech-to-text computer, for example, could include an actual combination of hardware and software to digitize speech, break it down into phonemes, convert the phonemes to possible words and sentences, apply error corrections to those words and sentences, etc. Such a prototype would take months or years of development and cost millions of dollars. It would be a one-off and still be far from being a final product, so it would definitely be considered a prototype. It would be exactly what most people think of when they think of a prototype.

Mention the word prototype to someone, and they will imagine something primitive and with rough edges, but they would expect it to be somewhat functional and close to the final product. If Jeff Hawkins told people he had a prototype for the Palm Pilot, those people would expect to see something with batteries and an LCD screen, not a block of wood. If IBM told its potential customer that it had a prototype speech-to-text machine, they would not expect a human typist taking dictation in the room next door.

Besides functionality, a key difference between pretotypes and prototypes is that the cost and time-frame for pretotyping is at the lowest end of the spectrum that is usually covered by prototyping. It's acceptable for a prototype to take months or years of development and cost millions of dollars. In contrast, it's definitely not acceptable for a pretotype to take that long or cost that much.

Prototypes are a necessary and incredibly useful tool that can and should be used to answer many questions about a potential product, such as:

- Can we build it?
- Will it work at all?
- Will it work as intended?
- How small/big can we make it?
- How much would it cost to produce?
- How long will the batteries last?
- How will people use it?
- What will people use it for?

Pretotyping, on the other hand, focuses on answering one – very basic and very important – question: **Is this the right thing to build?** Once that question is answered positively, then it makes sense to move from pretotyping to prototyping.

The conclusion I reached is that the term and practice of pretotyping deserve to stand on their own. Just as a startup is a specific type of early-stage company, pretotyping can be viewed either as a specific subset of prototyping or a prelude to it.

# **CHAPTER THREE**

## **It Will Fail**



You now have a rough idea of what prototyping is about and we'll get into more details and examples a bit later, but before doing that I want to spend some time explaining why prototyping all your ideas is so important.

Do you remember this sobering collection of statistics from earlier on?

- 90% of all mobile apps don't make any money
- Four startups out of five lose money for the investors
- 80% of new restaurants close within one year

The actual numbers may vary, but the message is clear. Simply put: Most new *its* are destined to fail – yours included. Most *its* are destined to fail because they are *the wrong it*: ideas that may sound good in theory but once developed turn out to be nowhere near as interesting, compelling, or as useful as originally anticipated.

Prototyping does not have the power to turn a *wrong it* into a *right it* – nothing can do that. But prototyping will help you identify *wrong its* quickly and cheaply so you can keep trying new *its* (or variations on existing *its*) until you find the elusive *right it*.

Since failure is our enemy, and it's important to “know thy enemy,” let's look at failure a bit closer.

## The Law of Failure

The evidence for the really bad odds against new *its* is so compelling and reliable that we can decree it to be a law:

### *The Law of Failure*

*Most new its will fail – even if they are flawlessly executed.*

Where the word “most” stands for a disheartening high percentage (typically 70-80-90%) and *its* represents pretty much any category you can think of: startups, restaurants, movies, books, soft drinks, TV shows, etc. And, yes, your *it* is included in one of those categories and, yes, it has the same lousy odds as everybody else’s *its*.

I can hear some of you complaining: “But how does this law help us? It just tells us that we’ll probably fail, even if we do a great job with our *it*. It gives us crappy odds and leaves us hanging. All this law does is lower our morale and kill our enthusiasm.”

True, on the surface, *The Law of Failure* does not appear to be very helpful. Strictly speaking, it’s not even a proper law. Can you imagine if Newton had stated his observations on gravity as: “Most things will fall if dropped?” Isaac, however, had it relatively easy. He was dealing with an immutable and universal law of nature. The eventual market success of any new product, on the other hand, has to contend with highly fickle, mutable and (more often than not) irrational human behavior. In this context, the probabilistic formulation of *The Law of Failure* is as good as it gets.

While far from perfect, I believe *The Law of Failure* is extremely important. If you accept this law as true, or even mostly true, and that neither you nor your *it* are exempt from the law, your mindset should change

from: “Let’s go for *it!* Let’s just build *it*, and go for broke!” to a more cautious “Let’s try *it*. Let’s prototype *it!*”

I know that “Go for it!” and “Go for broke!” have great romantic and heroic appeal. “Jumping in with both feet”, “betting the farm” and “damning the torpedoes” is how many legends are born – but it’s also how catastrophic failures arise.

Having said that, there may be cases when you decide that you don’t care what the odds are and you just want to go ahead with your *it* regardless of the consequences. I am in no way discouraging that. At least a few times in our lives, we should take some crazy risks and just *go for it!* There should be times when you care more about creating your specific *it* than having *the right it*. If that’s the case, laugh at *The Law of Failure*, throw caution to the wind, throw this book in the wastepaper basket and throw your heart and soul into *it*. Godspeed! I am rooting for you and I wish you success.

If, on the other hand, you are in a situation where you are not 100% committed to a very specific *it* and maximizing the odds for success is critical, then give *The Law of Failure* the respect it deserves because ...

## ... Failure Is Not An Option

That's right. For any given *it*, failure is not an *option*, but it's the most likely outcome.

We can't get away from the *Law of Failure*. We can't change the odds for new *its*.

What we can do, however, is use *The Law of Failure* to our advantage, the same way accountants use tax laws and Lady Gaga uses paparazzi.

How do we do that?

We invite failure, we seek it, we hunt it down and get it to show us its ugly face as soon as possible so we can determine if we are on the wrong track and make the necessary adjustments early on.

We concoct some really cheap bait (*bait*) in the form of a pretotype. Something that looks and smells like our *it*; something we can use to try to trick *the beast of failure* into rearing its ugly little head. We trek to the entrance of the dark and musty hole in the ground where the beast dwells. We then dangle our pretotype *bait* at the entrance of the hole to see if the beast of failure emerges from the shadows and inches close to us, close enough that we can smell its fetid breath and catch a glimpse of its cruel mouth and beady eyes. Close enough to make sure that it is the real beast. Then we toss our cheap *bait* to the beast as a sacrificial offering and run like hell in the opposite direction – before the beast can sink its teeth into our flesh and drag us down to its miserable hole to feast on us.

The best thing you can do is feed the beast cheap little morsels of various *its*. The beast likes to eat wrong *its* but – given a chance – it would LOVE to eat *you*! You must be ready to toss morsels of your *its* to the beast and run away. If you don't, if you get too attached to your *it*, and invest too much time developing it before dealing with the beast, you will probably end up having all your time and effort devoured by the beast.

If we do this well, the only thing we'll lose is our *bait* (our pretotype), but we get to live another day and try a different *it* – and keep trying until we come up with a *bait* that fails to attract the beast of failure – a *bait* that might just turn out to be *the right it*.

Pursuing your idea to the end, even if it turns out to be the wrong idea may sound exciting and heroic, but pretotyping is no less exciting. In pretotyping, you are still on an epic and challenging quest – the quest for the *right it*. Between you and *the right it* stands the formidable beast of failure. You cannot avoid dealing with the beast; you still have to fight it – but with pretotyping your odds of success are much greater.

That's the essence of our strategy – the very essence of pretotyping. But playing this game with failure only makes sense if the *bait* we use is something cheap and inexpensive, a pretotype that we put together in a few hours or days and at a minimal cost – something we don't mind abandoning.

### Three Ways To Fail

Failure is the most likely outcome for any given *it*, but not all failures are created equal. There are three ways to proceed with your *it*; three ways to deal with the beast of failure:

- Do nothing with *it*
- Go for *it* (*productype it*)
- Give *it* a try (*pretotype it*)

The first way is to fail the is way of sloths and chicken: people or companies too lazy, insecure or cowardly to put forward any effort or risk anything. Dealing with failure by not trying at all is the surest way to always fail. If you've read this far, I'm sure you are not in that category. You are ready to build something.

The second way to fail, is the exact opposite of the first. Instead of laziness, insecurity and cowardice, we have excessive eagerness, confidence and hubris. Dealing with failure by underestimating it will, more often than not, lead to painful, costly and slow failure.

These first two types of failures are caused by too much thinking, too much talking and too little reality too late. All *its* are born as ideas, but if we don't quickly shift from thinking and talking to putting something concrete in front of our potential users and customers, our *its* run the real risk of spending too much time in a very dangerous place I call *Thoughtland*.

## **Thoughtland**

Thoughtland is a fictional place inhabited by two strange entities that float around and interact with each other: ideas and opinions. More precisely: *unrealized* ideas and opinions about those *unrealized* ideas.

Thoughtland is where every *it* begins life as a simple, pure, abstract idea. As they float about in Thoughtland, *its* attract opinions that stick to them, like barnacles to a ship.

Thoughtland is a very safe place for ideas because, until they are actually converted into something more tangible such as a rough prototype of an app or the first draft of a book or script, they cannot fail. The only thing an abstract idea can “produce” is an opinion, something even more abstract and of even more dubious value.

While safe for ideas, Thoughtland is a very dangerous place for creators, innovators, entrepreneurs and authors. The opinions that fester in Thoughtland and attach to our ideas can lead us to fail in two painful ways:

*False Negative* opinions about our *it* can scare us into abandoning our idea so we *do nothing with it*.

*False Positive* opinions about our *it* can blind us to *The Law of Failure* and cause us to prematurely overcommit and *go for it*.

Let's look at how these two scenarios can happen.

## The “Do Nothing” Scenario

Most *its* never leave Thoughtland. They remain forever in this limbo as unrealized ideas. This is the saddest form of failure. Sure, the odds were that that idea was a *wrong it*, but there was a small chance that this idea was the next Palm Pilot, or Google or Twitter, and someone gave up on it without even giving it a chance. Sad. Sad. Sad.

A good percentage of *its* fail to see the light of day simply because their *ideators* never get off their butts to do something with them. They believe the idea is a winner; other people tell them that the idea is a winner, but they are too lazy/tired/busy/broke/inexperienced/afraid/(insert your favorite excuse) to do something about *it*. As we'll see later, prototyping can help us deal with this particular situation.

The remaining percentage of *its* fail to see the light of day *not* because we are lazy/tired/busy/..., but because, while in Thoughtland, our *its* attracted enough negative opinions (possibly our own but mostly from others) that our belief in *it* first wavered and then collapsed altogether. This happens a lot and, unfortunately, it's the fate of many *right its*. How does this happen? Let me illustrate with an example:

Let's assume that Alice has an idea for a new mobile app, something that uses text messaging and allows people to create short messages – say 100-200 characters max – that will be automatically broadcasted to our friends, family or anyone who wants to follow us. Let's call this app *MultiTextBot*.

Alice takes her idea for MultiTextBot, her *it*, to Thoughtland. Let's see what happens:

*Alice describes MultiTextBot to a couple of dozen of her friends and solicits their opinions.*

*Almost all of her friends tell her that this is a lame idea and that they'd never use it:*

*“Who would care about what you are doing?”*

*“Why would I follow you?”*

*“I don't want to be followed.”*

*“What's with that stupid 100-200 character limit?”*

*The few friends that are too nice to be completely negative give helpful suggestions: “Perhaps you should get rid of that stupid character limit and add the ability to send photos and GPS coordinates before you even think about launching it.”*

*Silly friends. What do they know anyway? Alice decides to take her **it** to real professionals - to VCs. They'll see how great it is.*

*The VCs don't get **it** either. Some just pass: “Sorry kiddo, this is not big enough for us, but good luck!” Some ask for user data, but Alice has none: “Sorry, it's just an idea at this time, but look at my Power-Point projections ...” The VCs tell Alice to “... come back if and when you have a million users then we'll talk.”*

Wow. How could Alice even think that this was a good idea. Good thing she asked for opinions before quitting her job and started development on this silly app. She decides to forget about it. Phew! That was a close call!

This happens – a lot! Of course, since most **its** are not *the right it*, negative opinions kill a lot of bad ideas. But they also kill many innocent and very promising right **its**.



Most of you have probably realized that my example of Alice's MultiTextBot idea is a very thinly disguised description of Twitter – arguably one of the most successful and world-changing products in history.

And yet, before Twitter's usefulness and impact became obvious and irrefutable, the initial opinion and reaction of most people who heard the idea – including a lot of VCs and smart investors – was negative: they didn't *get it*. There are still people who don't get it, but that does not matter because there are tens of millions who do get it and use Twitter everyday. Twitter was *the right it* – but you wouldn't have known it by its reception in Thoughtland.

Opinions, bah!

## The “Go For It” Scenario

We’ve seen how negative opinions can kill many *right its* in Thoughtland. But that’s only half the story. Let’s look at flip side and how positive opinions can get us to overcommit to *wrong its*.

We need another example for our *it*.

How about this: Tom, a first-class software developer, has an idea for a mobile app to help romantically-challenged guys like himself. This app will automatically send thoughtful text messages to their significant other at random times during the day. Thanks to this app – let’s call it the *HoneyTextBot* – your significant other will receive text messages like: “Hi honey. I’m thinking of you. Love. Your little hamster.” or “Hey babe, I just text to say I LOVE YOU. XOXOXO”

Tom’s *HoneyTextBot* will make the significant other think that the romantically-challenged guy is thinking about them at that very moment – even though he may be out drinking beer with his pals and watching mud wrestling. How romantic!

This is Tom’s *it* – his new *idea* on the *table*.

Here’s what can happen to Tom’s idea in Thoughtland:

*Tom mentions his idea for the HoneyTextBot app to a couple of dozens of his friends and colleagues (all males) to solicit their opinion. He calls this his “market research.”*

*Most of Tom’s friends, let’s say 70%, think it’s a great idea and tell Tom they will definitely buy the app for \$1.99 and would use it regularly.*

*Tom extrapolates from his “market research” and reaches the conclusion that he could easily make millions with this app: “70% of guys with phones and a significant other times \* \$1.99 = ... dunno exactly ... but it’s got to be a lot of money!”*

*Bolstered by such favorable expert opinions and careful financial projections, Tom quits his job and spends 3 months and all his savings writing a full-featured and highly-polished version of HoneyTextBot. Tom is a great developer and has an excellent sense for design, so the app looks beautiful and operates flawlessly. The first version can send lovely little text messages in over 20 languages! To cover all bases and pre-empt any competition, Tom decides to develop and launch his app on all the major mobile platforms (Android, iPhone, Blackberry) simultaneously.*

*Tom launches HoneyTextBot and ...*

*... not much happens. Nobody seems interested in Tom's beautifully crafted app. Not even his friends. Of those two dozen friends – 70% of whom of told him they'd buy and use HoneyTextBot – only three of them actually buy it, and only after Tom reminded them several times. After a week, two of them uninstalled it from their phones and the third forgot it was even there.*

What happened?

How could it be that an *it* that elicited such positive opinions turned out to be such a flop? How did Tom's 70%-will-buy projection turn into 0.0002%-actually-bought? Well, that's what happens when you make your decisions based on what you "learn" in Thoughtland.

In this case, Tom's Thoughtland-based analysis gave him a **false positive**. While dwelling in Thoughtland Tom was misled into thinking that his *it* was *the right it*. Thinking he had *the right it*, Tom quit his job and spent three months developing the full-blown app – three versions of it. Tom didn't just skip the prototyping phase, he even skipped the prototyping stage. He went straight from idea to what I call a *productype*.

Productyping is the evil twin of prototyping. If prototyping can be summarized as: "make sure you are building the right *it* before you build *it* right," productyping can be summarized as: "build *it* right even if you are not sure you are building the *right it*."

What was Tom thinking? He is a smart guy. Why did he invest even one penny in creating more than one version of HoneyTextBot? Why did he even bother to internationalize it and provide support for multiple languages?

What happened is that, buoyed by such positive opinions, Tom ignored *The Law of Failure*. He assumed success and decided to go all out, all at once.

This, unfortunately, happens a lot. When our own infatuation with our *it* is combined with false positives from Thoughtland it's hard to resist *going for it*.

Besides doesn't "going for it!" sound good? Doesn't it feel good to say it and do it? Isn't that *The American Way*? Yes. Yes. Yes. It feels good – make that GREAT – at first.

Overly optimistic individuals, by the way, are not the only ones who fall for this trap. Seasoned professionals at major companies fall for it just as often; they go from Thoughtland to productype in one fell swoop.

**Productyping is the way most new products are developed.**

**Productyping is the reason why most failures are slow, painful and expensive failures.**

## Get It The Hell Out Of Thoughtland ASAP

All *its* – wrong *its* and right *its* – are born in Thoughtland. But, as we've seen, spending too much time in Thoughtland can often lead us into prematurely abandoning potentially good ideas or over-committing and over-investing in potentially bad ideas. In other words:

- Do nothing with *it*
- Go for **it** (producttyping)

As we know, chances are that our *it* is not *the right it*, but the place to make that determination is not in Thoughtland but in the real world where, instead of subjective opinions, we can collect actual usage and market data.

We must not let our *it* fester in Thoughtland, we have to get *it* out of there as soon and as cheaply as possible. And that's where pretotyping – the third and best way to deal with the beast of failure – comes in:

- Give *it* a try (pretotyping)

# **CHAPTER FOUR**

## **Pretotype It**

Enough preambles, justification, explanation and definitions. Time to get to the meat of this book – the actual creation and testing of prototypes.

First, I'll introduce you to some basic types of prototyping, then we'll look at ways to test them and finally I'll combine everything we've learned into a few complete examples.

## **A Hodgepodge of Prototyping Techniques**

Some day, if this book turns out to be *the right it*, I will invest time and effort to create an extensive, well-structured, formal and official-looking taxonomy of prototyping techniques. At that time, I will give each technique a fancy name, describe ideal usage scenarios and provide plenty of examples. But since this version of the book is still a prototype, what you'll be getting is more of a hodgepodge of just a few different ways you can prototype your ideas with a rough explanation of when and how to use them.

Here's a quick summary of the techniques we'll talk about:

**The Mechanical Turk** – Replace complex and expensive computers or machines with human beings.

**The Pinocchio** – Build a non-functional, “lifeless”, version of the product.

**The Minimum Viable Product (or Stripped Tease)** – Create a functional version of *it*, but stripped down to its most basic functionality.

**The Provincial** – Before launching *world-wide*, run a test on a very small sample.

**The Fake Door** – Create a fake “entry” for a product that doesn't yet exist in any form.

**The Pretend-to-Own** – Before investing in buying whatever you need for your *it*, rent or borrow it first.

**The Re-label** – Put a different label on an existing product that looks like the product you want to create.

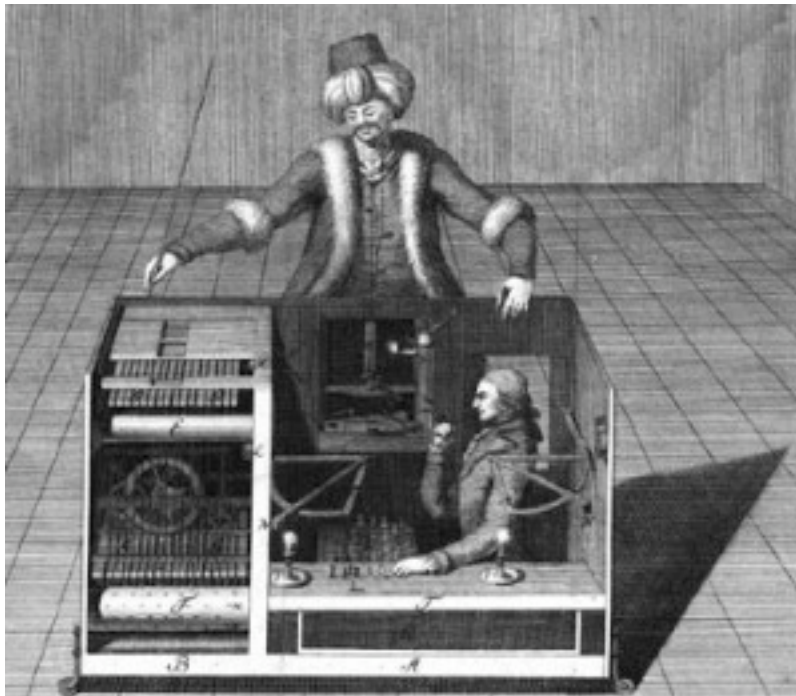
Feel free to use, abuse, misuse or confuse any of these techniques. Combine, refine, re-define and add to them to your heart's content. If you come up with an interesting prototyping technique or suggestion let me know about it ([asavoia@gmail.com](mailto:asavoia@gmail.com)); describe it and suggest a name for it, and I might just include it in future versions of the book or feature it on my blog ([pretotyping.blogspot.com](http://pretotyping.blogspot.com)).

Now a few more words on each technique.



## The Mechanical Turk

This prototyping technique borrows its name from the famous Mechanical Turk chess-playing “machine” that was touring the world in late 18<sup>th</sup> century. People were led to believe that the “Turk” was a mechanical contraption (an automaton) programmed to play chess. In reality, inside the box there was a small, but talented, chess player making the moves by manipulating the mannequin.



A Mechanical Turk prototype is ideal for situations where you can replace costly, complex or yet-to-be-developed technology with a hidden human being performing the functions of that technology.

The IBM speech-to-text experiment is a perfect example of this technique: Developing even a high-quality speech-to-text engine would have taken years and a huge investment, but a human typist, hidden in another room the same way the chess player was hidden inside the Mechanical Turk contraption, easily simulated that complex functionality.

## **The Pinocchio**

This prototyping technique was inspired by Jeff Hawkins' wood block Palm Pilot prototype and has been named after the wood puppet who, after being visited by the Blue Fairy, becomes a real boy.

A Pinocchio prototype is best suited for situations where things like size, shape, weight, portability, etc., are important and where one's imagination can be used to fill in the blanks – much the same way Hawkins' pretended that his wood block had the functionality required to schedule appointments, store phone numbers and keep notes.

## **The Minimum Viable Product (or The Stripped Tease)**

The term Minimum Viable Product (MVP) was introduced and popularized by Eric Ries, the creator of The Lean Startup movement and one of my personal heroes.

As the name suggests, this technique involves creating a working prototype – an actual product – but with features and functionality stripped down to the bare minimum in order to: “... *collect the maximum amount of validated learning about customers with the least effort.*”

Since they involve some actual, if basic, functionality, MVPs typically require more work than Mechanical Turk or Pinocchio prototypes. But an MVP can be developed much more quickly because it dispenses with all non-critical features. An MVP for an online family diary application, for example, should only support text entries (and perhaps uploading of pictures), but it should not bother to provide support for different text fonts, uploading of videos or different types of sharing. Such features may be nice, and even required, for the success of the final product but should only be added once initial testing indicates that the online family diary is *the right it*.

*Note: As I mentioned before, I learned about Lean Startups and MVPs a few months after I'd been talking about prototypes and building them. During a*

*workshop, I built a “Stripped Tease” (my name at the time) prototype for a mobile app and someone said to me: “Hey, how is this different from Eric Ries’ MVP concept?” I did not have a good answer at the time. But after learning more about MVPs and Eric Ries work, it’s clear that MVPs and prototypes (as well as the overall Lean Startup methodology) are all focused on helping creators, innovators and entrepreneurs avoid the same basic mistake: investing a lot of time and money to develop products for which there is no market – or not enough of a market to justify the investment.*

*If you are interested and intrigued by prototyping and this book, then you must buy, read and follow Eric Ries’ book The Lean Startup. It’s a proper book and something everyone should read, whether they work in a startup or a Fortune 500 company.*

## **The Provincial**

In many cases, the major costs associated with a product are not in developing the basic functionality, but in scaling the product to support and make it useful for a large number of users. A Provincial prototype provides the core features of the intended final product, but limits its scope (and scale) to support a small subset of the ultimate target market. As always, this is best explained with an example.

Let’s assume that Sandra has an idea for a mobile application that helps people find restaurants that serve only organic food. Let’s call Sandra’s *it* the *Organic Eater Helper*.

One of the most expensive and time consuming aspects of this app would be the creation and maintenance of a national database of restaurants that meet the requirements of serving only organic foods. There may be thousands of such restaurants across the country, and to include them all, and write the code to automatically keep the list up to date, Sandra would have to do a lot of work – unnecessary and wasted work if it turns out that the *Organic Eater Helper* app is not *the right it*.

A Provincial prototype would be developed as follows: Sandra should start by focusing on a particular city or county – ideally this is where she

already lives. Since there will probably be only a few organic restaurants in the area she selected, the development of the application is greatly simplified. Sandra can *hardwire* the names and location of the restaurant directly in the app instead of having to write code to poll a central database with thousands of restaurants and only return the ones closest to the user's location.

In addition to simplifying and accelerating the development of the prototype app, the provincial approach will also simplify and accelerate Sandra's marketing and testing effort. Instead of advertising the app nationally, she can focus on a smaller region and save a lot of money and still learn whether or not her app might be *the right it*.

## The Fake Door Pretotype

The name of this technique comes from a presentation by Jess Lee, co-founder and VP of Products for Polyvore. Great name. Thank you Jess!

With a Fake Door pretotype, the only requirement is to create an "entry" point for a potential product (or new feature). The product (or feature) does not have to exist at all. In Jess's words: "*In a web product, what this means is that you **pretend** that a feature exists and you see if anybody clicks on it.*"

Fake Door pretotypes are useful for determining the level of interest for an *it*.

On the Internet, a Fake Door can be implemented as a link, a button on a web page, or a web ad for your *it*.

Let's assume that Sandy is thinking about writing a book on *squirrel watching* (a worrisome variation on the already worrisome and mysteriously popular hobby of bird-watching.) Before she invests months of precious time away from her actual squirrel watching pursuit to write *The Complete Squirrel Watcher*, Sandy can use a Fake Door pretotype to

determine the level of interest in such a tome by creating a web ad – something like this:

The Complete Squirrel Watcher.  
The only book for serious squirrelers.  
Only \$9.98. [Click here for more information.](#)

She can then pay for Google AdWords to serve her ad on squirrel-related websites or whenever people search online for “squirrel watching.”

We’ll elaborate more on this particular example in the *Putting It All Together* chapter – I am sure that you, and the dozens of squirrel watchers out there can’t wait.

## **The Pretend-To-Own**

Some *its* may require a major upfront investments, in such cases, it’s critical that you prototype the idea by borrowing or renting those expensive items.

A new business that requires a physical store, for example, should not commit to a 5-year lease until they are sure that the idea is viable. Instead, they could try to get a 3-month deal on some un-leased space or – even better – arrange to squeeze their display inside another store that may attract the same type of buyers.

The idea for a new *green* car rental company that only rents electric cars should be tested by either renting or borrowing a few electric cars for a few weeks – not buying a fleet of them upfront.

You get the idea. Be a cheapskate until you know you have *the right it*.

## Ethical Considerations

Unless you are a borderline psychopath, some of these techniques may bother you from an ethical standpoint. Is it really right to create a “fake door” for example, just to see if people click on it?

I thought about this quite a bit and came to the following conclusion:

*Wrong its* are responsible for huge amount of waste. They waste the time of the smart people who develop them as well as the money and natural resources that should have been used to build something better and more useful. Time, money and resources invested in *wrong its* are time, money and resources stolen from *right its*.

Think of all the products you’ve bought and used only once or twice before throwing them away and regretting the purchase. Think of all the unsold products that end up in landfills.

Pretotyping can save you, and your potential customers, from wasting a lot of time and money on *wrong its*.

Use your judgment and sense of ethics when developing and testing pre-totypes and you should sleep well at night.

# **CHAPTER FIVE**

## **Test It**

Pretotypes are created for one reason, and one reason only – to help us determine the level of interest and people’s reaction to our *it*. The data we collect with the pretotypes will help us determine if our idea is *the right it*.

The only effective way to know if an *it* is *the right it* is to test it. Not in Thoughtland, parading an abstract idea and collecting subjective opinions, but in the real-world with a concrete pretotype used to collect data from actual users.

## Data Beats Opinions

At Google we have a couple of important core beliefs: “data beats opinions” and “say it with numbers.”

But what kind of data should we collect with our pretotypes and what numbers should we “say it” with?

It’s impossible to come up with a fixed set of metrics that will apply equally well to all *its*. The success of a book, for example, is typically measured by how many copies it has sold, and a movie by its box office receipts. The success of a web-based service such Google GMail, on the other hand, is best measured not by how many people signup a GMail account, but by how many people use their account *regularly* (e.g., 7-day active users.)

While there isn’t a universally applicable set of success metrics, there are some common guidelines that can, with some modifications, be applied to most *its*.

Since this is a pretotype version of the book (see section on MVP), I will only introduce two basic but important and useful metrics: *Initial Level of Interest* and *Ongoing Level of Interest*.



## Initial Level of Interest (ILI)

The first metric you should try to collect on any *it* is what I call the *Initial Level of Interest* or ILI.

The ILI metric is a simple ratio:

$$ILI = \text{number of actions taken} / \text{number of opportunities for action offered}$$

Where:

*number of opportunities for action offered* represents the number of people who have been offered an opportunity to take some positive action associated with your pretotype,

and

*number of action taken* represents the number of people who have actually taken you up on that opportunity.

As always, an example should help to make things clear.

Adam is a nudist and occasional skydiver and he is so passionate about his two “hobbies” that he’s thinking of quitting his job as an accountant (especially since they won’t let him work *in the buff*), buying a plane and starting the world’s first nude sky-diving business: *Birthsuit Skydiving*.

Before Adam resigns from his job and buys that Cessna, it would be a terrifically good idea (to put it mildly) for him to see what the level of interest in his idea is. Is nude skydiving *the right it*? We know that there are many nudists and many skydivers, but how many nudists would like to skydive and how many skydivers would like to jump from a plane with nothing but a parachute? Here’s what Adam should do to determine the level of interest.

There are online forums for both nudists and skydivers, and let’s assume that Adam is already a member of at least a couple of them.

Adam could write the following post in the local nudist forum:

*Fellow nudist, I am renting a charter flight for a nude skydive. The cost is \$100 per jump. No skydiving experience necessary, and I promise you won't land in a field of cactii. The first jump will be a month from now, Saturday, May 31<sup>st</sup> in Santa Barbara. To sign up please send me an email with the names and number of nudists in your party, and I'll respond with the details. Space is limited, so first-come, first-served will apply.*

*Adam*

A cool thing about online forums is that most of them show you how many people have read each post. This gives Adam the first number he needs (i.e., the number of people who have read his post and had an opportunity to act by sending Adam an email telling him that they were interested.)

Let's assume that a week after Adam posted his message, he sees that 1,490 people read his post (this is the *opportunity for action offered* number) and that he has received 2 emails replies saying that they wanted to sign up (this is the *number of actions taken*).

The ILI in this case and for this group would be:  $2 / 1490 = 0.0013$  or 0.13%.

Not very encouraging, but also not too surprising since most people (including nudists) are naturally reluctant to jump from a perfectly good plane. At this point Adam can send the two responders a message saying that he's sorry, but due to a lack of interest the nude parachuting event has been cancelled.

Before abandoning his idea, however, Adam should post a similar offer on the local skydiving forum. Something like this:

*Fellow skydivers, aren't you bored of the same old, same old, types of jumps? To make things interesting, I am renting a charter flight for a nude skydive. The cost is \$100 per jump. I promise you won't land in a field of cactii, but on a nude beach – imagine the surprise! The first jump will be a month from now, Saturday, May 31<sup>st</sup> in Santa Barbara. If you want to sign*

*up please send me an email with the number and names of people in your party, and I'll respond with the details. Space is limited so first-come, first-served will apply.*

*Adam*

Let's assume that after a week, 898 skydivers had read his post and 112 replied that they wanted to sign up.

The ILI in this case would be:  $112/898 = 12.5\%$  – much better. Now we are talking.

With this simple Fake Door prototype and ILI metric, and less than an hour worth of “work”, our nudist skydiving friend Adam has already collected some very valuable data:

Skydivers are a much better target market (by a factor of about 100) for his idea than nudists.

The ILI for skydivers is pretty high, over 10% with 10,000s of skydivers in the US this number is good enough to pursue the idea further.

A percentage of the skydivers who replied said that they were very interested and were ready and eager to sign up. This is a very strong *right it* signal.

ILI data is very powerful and easy to interpret and act upon when used for comparisons with similar ILI. In Adam's case, ILI data unambiguously indicates that skydivers are a much better target market than nudists. It's much harder to know, however, if a given ILI is good enough to proceed. For some *its*, an ILI of 12.5% might be considered great, for others it might not. While it's important and easy to collect data and calculate the ILI, interpreting it will usually require some judgment and domain/market knowledge.

Things are looking good for the *Birthsuit Skydiving* idea but, as we'll see ILI is just an early indicator of *right it* potential. Let's investigate what Adam should prototype and measure next.

*Note: I have a sneaky suspicion that there might be FAA regulations against nude skydiving. Since this book is a pretotype, I did not investigate this matter at length. And, just to be sure, I am neither endorsing nor suggesting that nude skydiving is a good idea – so don't try this **it** at home. But if you do, don't blame me for giving you the idea or send me photos of you doing it.*

## Ongoing Level of Interest (OLI)

For some *its*, where success does not necessarily depend on repeat business (e.g., a book or an arcade-style game app), a good result based on an *Initial* Level of Interest (ILI) may be enough to proceed to the next step. But there are many *its* where success does depend on repeat purchases, return visits, or ongoing usage by the same group of people who were initially interested in it. This is particularly important if the running of the business requires the upfront purchasing of some expensive equipment or committing to some significant recurring costs.

Unlike the ILI, the Ongoing Level of Interest (OLI) is best represented by a time-based graph (or table) rather than by a single number. Each point/entry in the graph/table represents the level of interest at a particular date. What you should be looking for in the OLI graph/table is a trend. Does interest fade to zero after a while? Does it drop a bit but then steadies at an acceptable rate? Does it go up? In the first case you probably have a *wrong it*, the second case could go either way and may deserve a bit more study, and the third case is a promising indication that you just might have a *right it*.

As always, this is much easier to explain with an example. Let's pick up where we left off with Adam and his nude skydiving business.

In the case of *Birthsuit Skydiving*, Adam would be foolish to quit his job and buy that Cessna airplane just based solely on his ILI numbers. Even if more than 10% of all skydivers were interested in a trying a nude jump, if none of them come back for more this would be a short-lived business.

Before making any major decisions (like quitting his job) or investments (like buying an airplane), Adam would be wise to check the Ongoing Level of Interest (OLI) in his idea.

Fake Door prototypes are great to test the ILI, but you need something more concrete and substantial to test the OLI. Most people will not continue to open Fake Doors. The Pretend-to-Own prototype would fit the bill quite well in this case.

Instead of buying a plane, Adam should just rent one on an as-needed basis. Renting a plane by the day may cost too much to make it a viable long-term business option for *Birthsuit Skydiving*, he may even be losing a few hundred dollars each time. But until Adam is convinced that his nude skydiving idea is going to fly, it's better for him to lose a few hundred dollars testing it, rather than drop tens of thousand of dollars upfront hoping that he has *the right it*. Remember *The Law of Failure*, even with a positive ILI result, the odds are still against Adam's *it*.

Let's assume that Adam follows the prototyping protocol, advertises the flights on his local skydiving forum every week and, over a period of two months, he runs 8 flights: one flight every Saturday.

Here's is OLI data after the two months:

<b>Flight #</b>	<b>Signups</b>	<b>Revenue</b>	<b>Cost</b>	<b>Profit/(Loss)</b>
1	21	\$210	\$250	-\$40
2	20	\$250	\$250	\$0
3	28	\$280	\$250	\$30
4	17	\$170	\$250	-\$80
5	7	\$70	\$250	-\$180
6	3	\$30	\$250	-\$220
7	0	\$0	\$0	\$0
8	0	\$0	\$0	\$0
<b>Total</b>	<b>101</b>	<b>\$1,100</b>	<b>\$1,500</b>	<b>-\$490</b>

Sorry Adam! Things were looking good for a while – you even managed to make a small profit on your third flight – but I’m afraid that this nude skydiving thing may not be *the right it*.

A high ILI is great, but if the success of your it depends on ongoing usage, you should test the ongoing level of interest if there are significant investments associated with your *it*. In Adam’s case, prototyping is suggesting that *Birthsuit Skydiving* may work well as a fun hobby or side activity, but at this time it would be unwise for him to quit his job, buy a plane and try to make a living with it. Prototyping saved the day – and saved us from the risk of having a nude parachutist landing in our backyard.

# **CHAPTER SIX**

## **Put It All Together**

Finally all the pieces are in place, so we can go through a couple of examples of creating and testing prototypes, and making decisions based on them. As you go through the examples, don't be surprised if you can come up with different ways to prototype and test these ideas, there is no single best approach. I'd be surprised if you can't think of other ways of approaching the same prototyping challenges.

### **Example 1: The Complete Squirrel Watcher**

Let's build on our example from the Fake Door prototype. As you might recall, Sandy is thinking about writing a book on squirrel watching. Since she would have to invest months of precious time away from her actual squirrel watching pursuit to write *The Complete Squirrel Watcher*, it would be a good idea for her to prototype the book.

In Sandy's case, since the success of a book is primarily determined by how many people buy it (i.e., it does not really depend on repeat purchases) all we need is a prototype to find out the Initial Level of Interest (ILI). Fake Door prototypes are ideal for this. And here's how Sandy could go about it:

For \$10, she can buy TheCompleteSquirrelWatcher.com domain and create a landing page that says:

*Fellow squirrel enthusiasts,*

*Thank you for your interest in "The Complete Squirrel Watcher".*

*I am hard at work on the book, but it's not quite ready for publication.*

*To reserve a copy at the special pre-order price of \$9.98 send an email to:*

*[iwantthebook@thecompletesquirrelwatcher.com](mailto:iwantthebook@thecompletesquirrelwatcher.com)*

*and I'll let you know as soon as the book is available.*

*The price will be \$9.98*

*In the meantime, happy squirrel watching and don't forget your rabies shots!*



*Sandy (Squirrelgirl) Watson*

Sandy can then craft a web ad, for example:

**Do you like stalking squirrels?**

[www.TheCompleteSquirrelWatcher.com](http://www.TheCompleteSquirrelWatcher.com)

The official book for serious squirrel watchers  
by Sandy Watson. Only \$9.98

For a few dollars, she can place the ad on websites dedicated to squirrels or have it show up when people use a search engine to search for anything related to squirrels. When people click on her ad, they are redirected to her website.

This Fake Door prototype would cost less than \$50 and take just a couple of hours of work requiring minimal technical skills.

Once this prototype is in place, Sandy can let the ad run for a month or so, after which she can analyze the data provided from the online ad service.

Assume this is the data generated by her prototype:

People who have seen the ad: 23,402

People who have clicked on the ad: 634

People who sent an email saying to buy the book: 230

There are a couple of interesting ILI ratios here.

The first is an indication of how many people who go to squirrel pages or search for squirrel are interested enough to click on an ad for a book on squirrel watching. This first ILI can be calculated as follows:

*ILI 1 = number of clicks on ad / number of ad impressions (i.e., how many people have seen the ad)*

In this case,  $ILI1 = 634 / 23,402 = 2.7\%$

This is not great, but not too bad either.

The second ILI ratio gives her the percentage of people who, after clicking on the ad, are interested enough in the book to send an email to Sandy:

$$ILI2 = \text{number of emails} / \text{number page visits to the landing page}$$

In this case,  $ILI2 = 36\%$  (230 / 634)

This is very encouraging, a whopping 36% of the people who visit Sandy's [TheCompleteSquirrelWatcher.com](http://TheCompleteSquirrelWatcher.com) web page send her an email to reserve a copy of the book. Of course not all of them will follow through, but this is still a very good number.

Now comes the difficult decision. Should Sandy go ahead and write her book based on this data?

That depends a lot on her expectations for the book. The data indicates that the book is unlikely to land a spot on The New York Times's best-seller list – not enough people seem *that* interested in squirrels. But that was never Sandy's expectation. For her, becoming an authority on the subject and selling a few hundred copies of her self-published book each year – enough to pay for her squirrel watching gear expeditions – would be good enough. In that case, the data from her prototype suggests that *The Complete Squirrel Watcher* will probably *the right it* for enough people to make Sandy happy.

## **Example 2: *Bob's Rate This Plate App***

For this example, let's assume that Bob is a nutritionist who wants to create a mobile app that analyzes a photo of a meal and returns a nutritional analysis and some kind of score from "A: Healthy and nutritious" to "F: Junk food"? Let's call this *it* the *Bob's Rate This Plate app*.

Bob talks to friends and many other people about this app, and most of them tell him that it's a great idea and that they would definitely use it. Fortunately, Bob has heard about *Thoughtland* and knows how misleading opinions can be. He does not know for sure how many people would use such an app or be willing to pay for it. Would users even remember to stop and take a photo of the food before they start digging in? Would they use it a few times – just for fun – and then never again?

Bob also realizes that developing an actual working software system to automatically analyze a meal based on a picture of it would definitely take a lot of work and money – and it may never get to the point where it's good or accurate enough to be useful (a problem similar to the one one faced by the IBM team with their speech-to-text idea.)

There are a lot of open questions that have to be answered and expensive technology to develop. This *it* definitely calls for some prototyping.

### **First step: Fake Door and Pinocchio prototypes**

By now, you should not be surprised that, as a first step, I would recommend for Bob to build some kind of Fake Door prototype to measure ILI (see previous example for how to do that.)

Let's assume that the ILI data is encouraging. However, Bob's vision for, and definition of, success for this app requires not only initial interest, but ongoing usage (i.e., a promising Ongoing Level of Interest). If what the app requires is cumbersome or awkward to do people may not stick with it. Heck, would Bob himself stick with it? Would he remember to take photos of his food before he starts eating it? Would he be embarrassed to do it in front of people, especially in a restaurant? Would he only take photos of his healthy meals and conveniently forget to record that banana split?

If we don't believe and don't use our *it* ourselves, how can we sincerely convince, or expect, other people to do that? To answer this question, Bob should follow Jeff Hawkins' Palm Pilot prototype example and develop a Pinocchio prototype to test the idea on himself. Since Bob al-

ready has a smartphone with a camera, he does not have to go out and build a wood-block like Hawkins did. He can simply pretend that his phone's camera app is the app he wants to build and fill the blanks with his imagination.

If Bob discovers that, after a few of days of using his Pinocchio prototype, his initial enthusiasm for the idea starts to wane and he takes fewer and fewer photos, then he might have a problem. Of course, he could try to explain the failure away, "this app is not for me, it's for my clients, I already know what I should eat, I don't need it." He might be right in this particular case, but he should still be concerned about it. The "I won't use it, but others will" argument is a giant red flag with "*wrong it*" written all over it: not something to be dismissed lightly.

However, to continue our example, let's assume that Bob quickly becomes so used to taking photos of his food before eating that it becomes a habit for him and he does it consistently and automatically. Not only that, but when he does it in front of other people, they ask him about it and say that they would love an app like that for themselves. He also starts to post his photos on an online album so he can keep track of everything he has eaten and mails them to a nutritionist colleague so she can give her feedback on his diet. This is a good sign. Bob now knows that he'd use the app himself on an ongoing basis and he found it useful enough to "implement" a couple of new "features" (i.e., posting the photos on an online album and sending them to a colleague.)

His first two prototypes tested well, the ILI was good and his *personal* OLI was also very good, now it's time to see if enough other people would use the app on an ongoing basis.

Bob needs to get an idea of what the OLI is and the Fake Door prototype won't work for that, nor would a simple Pinocchio prototype (the Pinocchio requires a lot of pretending and imagination about the intended features and functionality, they are great for convincing the creator of the idea who can fill those blanks, but not so good for collecting data from other users.) What Bob needs is a simple, but functional, prototype. Unfortunately, Bob is a nutritionist and not a programmer. Before investing

in hiring a programmer, is there a faster and cheaper pretotype he can create that will still give him some OLI data? You betcha!

### **Super-cheapo, low-tech, Mechanical Turk pretotype**

Since Bob is a nutritionist and he has over 500 clients, he can ask a few of his clients (say, 50 of them, about 10%) if they'd be interested in participating in a one month experiment. All they have to do is to take a photo of each of their meals before they start to eat and email that photo to him. In return, at the end of each day, Bob will send them an email with an nutritional grade along with some comments and suggestions on how to improve their diet. Nothing too fancy or time consuming; something along these lines:

*Dear Mary,*

*Thank you for helping me test Rate This Plate:*

*Here are your ratings for today:*

*Breakfast: F (eggs and bacon, c'mon you know better than that.)*

*Lunch: B (salad good, blue cheese dressing bad)*

*Dinner: A- (chicken and veggies looked healthy, but you get a minus for that buttered bread)*

*Please try to eat some fruits and veggies for the next few meals.*

*Sincerely,*

*Bob*

Let's say that 30 (out of 50) of Bob's clients agree to do the experiment (ILI = 30/50, or 60%). At first, Bob is disappointed, even though this ILI is high, he was hoping that all of his clients would agree to join – or at least 80-90%. After talking with the clients who declined the opportunity to join experiment, he learns a few things that he hadn't thought of. Many of his clients, for example, don't have a mobile phone with a data plan, so they can't email photos to him. And a few clients felt very uncomfortable sharing actual photos of their meals with him – or anyone

other *person* – but they’d be OK if their meals were analyzed by a computer. Good things to know and keep in mind as he progresses.

When the experiment starts, Bob sends his 30 volunteer clients instructions on what to do (i.e. Take a photo of your everything you eat and mail it to [bobthenutritionist@somedomain.com](mailto:bobthenutritionist@somedomain.com)) and, as the emails with photo start coming in (~80/day) he rates the meals and sends his emails with their grade and a nutritional analysis. A lot of work, but since he’s not a programmer, it was faster and cheaper for him to do it this way.

After a month of running the experiment, Bob has a pretty good OLI table:

<b>Week</b>	<b>Active (out of 30)</b>	<b>Photos received</b>
1	28	234
2	24	198
3	22	168
4	22	172

As it always happens, some people who said they would participate didn’t send even one photo and, as time passed, some other volunteers dropped off. By the end of then month, however, he still had over 2/3<sup>rds</sup> of volunteers actively submitting photos. This is encouraging.

Even more encouraging, a lot of the users are sending him requests for new features and functionality: “Hey Bob, can you send me my average GPA?” “If I forget to take a photo can I just send you a description of my meal?” “Can you send me a menu for each day that will guarantee me an ‘A’?”

Some, on the other hand, complain: “Bob, I don’t have good phone reception in our cafeteria, it sucks that I have to go outdoor to email you the photo – while my food gets cold.”

When you don’t hear from users, chances are that they are either not using your product, or don’t care enough to send feedback on how to en-

hance it or improve it. Getting feedback, good or bad, is a great sign. They care enough to suggest or complain.

Things are looking good for Bob: strong OLI and lots of user feedback. Bob's Rate This Plate app might just be a *right it*.

There is still a little issue of revenue and profitability. Bob wants to make a business out of it. Would the people who have been using the app for free be willing to pay for the service? How much would they be willing to pay: \$10/month – perhaps even \$30/month? By now, I am sure you know how he would answer that question. He still has 450 clients to experiment on. He can ask 100 if they would sign up for the service at \$10/month and another 100 to if they'd sign up at \$30/month and then measure the ILI and OLI for both.

Only a couple of clients signed up for the \$30/month service but, surprisingly, 42 of his clients signed up for the \$10/month service – more than he could handle manually. It was time to invest in automation. Unfortunately, he realized that the technology for automatically analyzing a meal based on just a photo was at least a few years away. But he found out that he could train college students who, for \$15/hr, would do almost as good a job as he did. He ran the numbers and realized that he could make a nice profit of \$4/patient each month.

After a few months running the service for his patients and making a profit, Bob decides to go big – his *it* was *the right it*. He hires a developer to create a custom app (instead of the clumsy email-based preto-type) and trains more students to handle the load.

*Bob's Rate This Plate* app was the *right it*, and because of it there are quite a few more healthy people out there.

Don't you love a happy ending?

# **CHAPTER SEVEN**

**Now Go Make It**



Even though we've gone through a lot of material very quickly, and I've subjected you to some rather *unusual* examples, I hope that I was successful in answering the following questions:

- What is prototyping?
- Why is it important?
- What are some of the prototyping techniques that you can use?
- What data to collect and what metrics to use with your prototypes?

## Now it's your turn!

I am sure you have quite a few *its* you want to try. Prototyping will help you in two ways:

- If your *it* has been held captive in *Thoughtland* for a while, prototyping should make it much easier for you to get started. Ignore the naysayers and get off your butt. Prototype it and see what happens.
- If you are getting ready to take a big risk or make a big investment in your *it*, prototyping will help you get started more quickly. It will also provide you with valuable data that will either give you more confidence that your it is *the right it*, or it will help you realize that you should make some changes to your *it*, or pursue a completely different it.

In all cases, feel free to keep in touch with me ([asavoia@gmail.com](mailto:asavoia@gmail.com)) to let me know how it's going for you and if I can help you in any way if you decide to give prototyping a try.

May you find your right *it*, and if you see the beast of failure, please tell it Alberto said "Hello!"

# **CHAPTER EIGHT**

## **Bonus Features**

## Is this book *the right it*?

Over the past two years, I have given dozens of presentations and demonstrations on pretotyping to thousands of people. I've practiced pretotyping for my job and I have started to help other people and organizations successfully pretotype their own ideas.

The overwhelmingly positive reaction to pretotyping has surprised me. People love the idea, they understand how and why it works, they want to know more about it, and – according to what many have told me – it has dramatically changed the way they think about pursuing and investing in new ideas and innovations. I have strong evidence that, when presented and explained live (usually with plenty of examples and hands-on demonstrations), pretotyping is *the right it*.

Many people, intrigued by the concept of pretotyping and wanting to learn more about it, have asked me to write a book on the subject. Writing a book, however, is no easy task (at least not for me) and one that requires a significant investment of time, energy and concentration. On top of that, most published books fail in the market – they are **wrong its**. That's why I decided to treat the idea this book as an *it* – an *idea to test* – and to pretotype this *it* (i.e., pretotype the idea of a book on pretotyping – a *pretobook*.)

Instead of spending months writing, editing, perfecting and polishing hundreds of pages (and sacrificing trees for a hefty book that few might read), I gave myself a few days to create a written version of my pretotyping presentations and workshops. The result is the slim book (or eBook) you are reading now.

Hopefully, the core idea, message and practice of pretotyping will be able to shine through my somewhat clumsy writing, poor organization of the material and painfully obvious lack of professional editing. If a book on pretotyping – at least one written by me – is *the right it*, even this version with a few rough edges, should achieve some level of success and popularity. Of course, I'd love to see *it* succeed wildly, but I know the odds are against *it*.

# The Prototyping Manifesto

Make sure you are building the right *it*  
before you build *it* right

**innovators** beat ideas  
**prototypes** beat productypes  
**data** beats opinions  
**now** beats later  
**doing** beats talking  
**simple** beats complex  
**commitment** beats committees

## About the Author

Alberto Savoia is Director of Engineering and Innovation Agitator in Google's Ads organization where, among other things, he led the development and launch of the original Google AdWords.

Prior to Google he was Director of Software Technology Research at Sun Microsystems Laboratories, and co-founder and CTO of two software development tools start-ups (Velogic Inc., acquired by Keynote Systems and Agitar Software, acquired by McCabe.)

Alberto's thought leadership in the area of software development tools and innovation has been recognized with numerous awards, including:

- *The Wall Street Journal Technology Innovation Award (2005)*
- *InfoWorld Top 25 CTOs Award (2005)*
- *InfoWorld Technology of the Year Award (2005, 2006)*
- *Software Development's Jolt Award (2005, 2006, 2007, 2008)*
- *Software Development Magazine Productivity Award (1998)*
- *Java Developer's Journal World Class Award (1998).*

Alberto is a frequent and sought after keynote speaker and author on the topics of innovation and software development tools. To date, his presentations on prototyping and innovation at Google have been seen by several thousand people.

You can contact Alberto at: [asavoia@gmail.com](mailto:asavoia@gmail.com)

## Acknowledgments

The concept of pretotyping as well as this book, would not have been possible without the encouragement and support of Patrick Copeland – my manager and mentor at Google. Patrick not only helped me to develop and refine these ideas, he makes sure that I practice what I preach – and launch early and often. He has also helped me to spread the word; he has given several extremely well-received keynote presentation on pretotyping at major conferences throughout the world.

I have been very fortunate to have two great innovators working with me at Google: Stephen Uhler and Bob Evans. Stephen is a born pretotyper who has helped me immensely by inventing and developing PretoGen, a tool that makes it possible to generate working prototypes in minutes. Bob, one of the smartest guys I know, has been a constant source of inspiration and lively debates during the earliest days of pretotyping. The idea for pretotyping was born from our discussions while we were sharing an office.

The other key person in the development, refinement and popularization of pretotyping is Jeremy Clark, a leading thinker and practitioner in the area of innovation and the founder FXX Inc. Jeremy and I continue to work together on pretotyping and we often give joint conference presentations on the subject.

Carlo Alberto Pratesi, professor of Marketing at Roma Tre University and founder of InnovAction Labs in Italy, has not only been a source of inspiration and examples but he has been very active in putting pretotyping in action in Europe.

Finally, I want to thank the hundreds of Googlers (and Google customers and visitors) who have come to my presentations and workshops. Their positive reaction to pretotyping, their own experiments with it and their ongoing suggestions and enthusiasm convinced me that pretotyping is *the right it*.

### **This book is dedicated to my family:**

To my father, who believed in me and made great sacrifices to invest in me when I first started and I had nothing but some crazy ideas. Thanks dad, you were my first VC.

To my mother, who let me leave Italy when I was seventeen to follow my dreams in Silicon Valley. Thanks Mom, I know that must have been hard.

To my wife, who has always supported my entrepreneurial risk-taking, makes our house a home and makes parenting seem easy.

To my children, who make me proud every day and who make the tough job of being a father almost too easy.

Follow Pretotyping:

[www.pretotyping.org](http://www.pretotyping.org)

<http://pretotyping.blogspot.com/>

[asavoia@gmail.com](mailto:asavoia@gmail.com)

Alberto Savoia on Google+

@pretotyping on Twitter