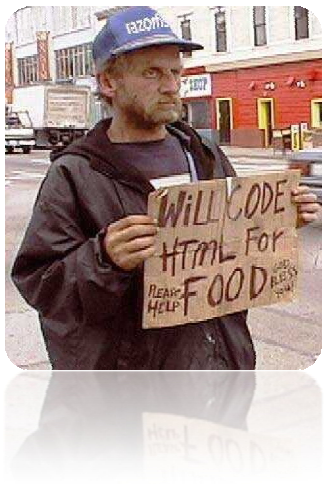
ZeMessenger

Project Report.

Benjamin Arnaud

Introduction



This document is supposed to be an internship report.
In Epitech, at the end of our studies, we spend 6 month in a company. Most of the time, you find your first job this way. But my story is different.

Here is a small recap of what I did in the past two years:

- In 3rd year we are allowed to work 3 days a week for any company.
- I chose to work for Zepeople, a community website.
- I had to develop an instant messaging application.
- During two years, I worked for them as an intern.
- But at the end, it was decided that my application wasn't good enough for being released.
- Our collaboration ended up just before my final 6 months.

At this point, I had two options:

- Throw away my code and find another internship in a completely different company.
- Follow my initial goal of making the vision we had a reality.

This document should be an internship report.
In Epitech, we spend five years devoted to coding, or at least that's what I did. From the first day until the end of studies, everyone got rid of their social life. You start up by one intensive month where you learn C programming language the accelerated way.

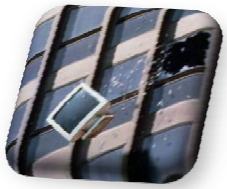
When I'm talking about C or C++ programming, I'm not talking about anything you've seen on the web like flash or facebook applications.
When I'm talking about C or C++, I'm talking about what made Windows, Firefox or Spore a reality.

Nowadays, everyone calls himself a programmer. That term is so wide it hardly defines anything. Being a C++ programmer is definitely a full-time job and even after 5 years, I consider myself as a beginner.

This document is not exactly an internship report.
During 2 years I spent hours after hours on that instant messaging application. I was at first working with a co-mate who eventually left the team to focus on his studies. On my own, I had to design, code and debug what would be the messaging platform of a full fledge community website.

At hard times I was watching the website photo section. Looking at all these students having fun, realizing I was working for them. I was sacrificing my whole social life trying to figure out how to code the most advanced yet simple user interface. Days of the release were approaching and I felt the excitement of their feedback on forums or blogs. My entire underground work would be recognized.

Days and nights of coding had passed, I felt the product was ready to be released or at least beta tested. The leads of Zepeople (~2 people) thought differently.
They made a proposition I couldn't accept. I decided to quit and keep the rights on my code.



This meant:

- I had no platform to host my application.
- I had no community to start from.
- A bunch of copyright changes were required in the application.
- Last but not least, I lost my end studies internship.

At this point, legally, I had no diploma.
So I did the following:

- I asked my uncle to take me as an intern in his company.

This way I could:

- Legally be in order due to the internship contract.
- Focus on releasing the applications to the users.
- Finish what had been started.

With that approach only one problem remains: the internship report.

For Google an internship is "an opportunity for students to gain supervised, practical work experience". If you dissect that definition:

- ✓ "is an opportunity": my program is an opportunity to develop something great and unique.
- ✓ "for students": technically even though I'm already working I'm still a student until I get my diploma.
- "to gain supervised, practical work experience":
  - ✗ "supervised": I'm not supervised.
  - ✓ "practical": Coding is always practical in both good and bad ways.
  - ✓ "work experience": 2 years on the same project is definitely an experience.

According to the above statements, the only problem with my approach is the “supervised” aspect. Since I’m the only one coding that project can you consider it as an internship?

This document is a project report.
When the team is no longer the key aspect of an experience, the only thing that matters is the project. Now that you know how things happened for me at Zepeople, I’ll let you be the only judge about the product. This document will take you back to the roots of the project, from early design to what it has become. I’ll try to be objective and mention both positive and negative sides. In the end you’ll decide whether you would have dropped it or not.

We will divide our report into 2 parts:

1. Birth of a project
2. Don’t dream it, code it ©

1. Birth of a project

1.1. From sunny Philadelphia to cold Paris

My previous experience was 6 months in the US. The last words of my report were the following: “It’s the end of the dream”.

During my oral exam, I remember someone asked me:
"You mentioned your dream; can you tell us what reality is now?"

That's when I realized I was back in France. Aside from his 35 hours a week and his 5+ weeks of holidays, the average French guy has no dream, no vision, so few ambitions. At this point I knew a lot of what made my experience so special would never cross the ocean. It was an odd feeling, being confronted to the French culture once again, but this time having a different view of what life could be. Should I keep what I learnt or get back to my old habits?

I had a lot of troubles "reporting" on that previous experience. It appears to me, someone was asking me to take the whole experience, remove its essence and keep a few insignificant details to make it fit with what an internship report should be.

My American dream got a 7 out of 20 grade.

My oral exam's very last question was:
"What are your plans now?"

This report is my answer.

1.2. What is ZeMessenger?

Paris, November, cloudy weather. I have an appointment at Zepeople with my friend Arthur M. to discuss the specifications of what will be my next milestone. This company is a community website for the most select clubbers. Their user database is very limited, but has a high purchasing power.

The concept of the website is quite close from Facebook. You have a profile page, friends, you interact with others and shine on your party pictures. The CEO was looking for some innovative concept to push his user experience forward. Just like MySpace did, it was determined that the website would make great use of a dedicated instant messaging application. ZeMessenger is that application.

Here are the original specifications of the application:

- Contact list:
  - Add a contact
  - Remove a contact
  - Grouped contacts
  - Respect Zepeople style
  - Avatar picture
  - Display contact's profile page
- Conversation:
  - Independent window
  - Conferences
  - History
  - Fonts
  - Smileys
  - File transfers
- Help:
  - User help located on the website

These specs were **not closed** and could evolve during the development process.

2. Don't dream it code it ©

2.1. Who would spend his days behind a computer screen?

I've been coding for five years, I've seen many projects. So far I would classify my programs in four categories:

- The programs you do for fun
- The programs you do to learn
- The programs you do for the others
- The programs you do for you and the others

Typically when you are:

- Learning a language: you fall into the first two categories.
- Developing for a company: you fall into the third category.
- Developing for the perfect company: you fall into the fourth category.

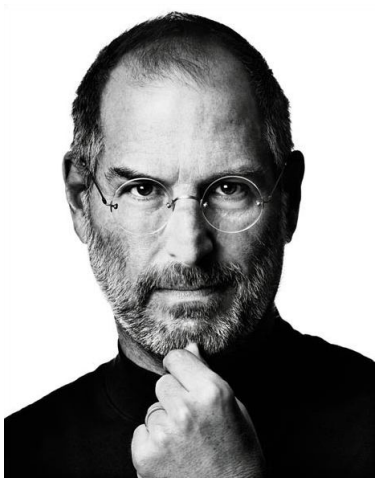
From what I've seen, results tend to be the best when you're on the fourth category.
Why?

Back in first year of studies, when we were doing our integration trip, the president of the Ionis group told us:

"I hope you like programming, because you're going to eat loads of code"

-- Fabrice Bardeche

From my perspective the biggest difference between my studies and marketing studies -for instance- is the passion. You don't spend nights behind a computer screen if you don't L.O.V.E what you're doing.



At times, you must feel like you have a holy mission of some sort that keeps you moving forward.
You need courage, you need to be a warrior and you need to be prepared for everything even starting from scratch.
Nothing's piece of cake when it comes to programming.

"If you don't love what you're doing, you'll give up"

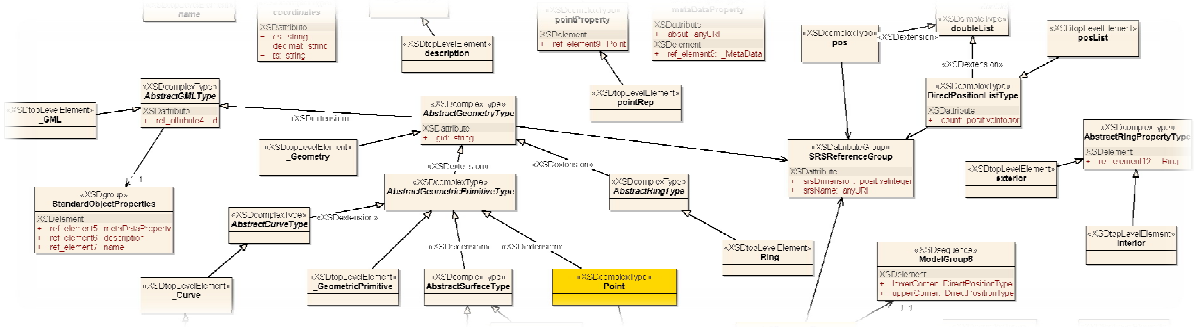
-- Steve Jobs

Now imagine you have to deploy all these efforts into an application you don't believe in.
You may end up doing something satisfying and usable, but you'll never spend those extra bits of efforts to make it awesome. You have to believe in what you're coding.

That might be one of the key reasons why Open source software works so well these days, because people are getting involved by passion and not by financial needs.

Coding is not a job, it's a passion.

2.2. Making specs a reality



What's the road from abstract specifications to a full implemented application?
One could think that a good programmer is someone who can reinvent everything.
From my point of view, a good programmer is someone that can reuse everything.

"Our strength is being able to reinvent, our talent is doing it when necessary"

-- Benjamin Arnaud

Early in the development I made the choice to use third party libraries:

- Qt GUI library
- Libpurple instant messaging library

- The Qt library is a cross platform application framework.

It means you can:

- Write the application once
- Deploy on three platforms (Windows, Mac, Linux)

- Libpurple is a cross platform instant messaging library. It is intended to be the core of an IM program.

Combining both libraries we get:

- A multi-platform GUI
- A multi-platform instant messaging core

At this point, what remains to be coded is the following:

- The best instant messaging GUI combining:
  - Ergonomic
  - Simplicity
  - Design
- Connecting that interface and libpurple together

Reusing code is crucial when you're developing a full scale product:

- It saves you a great amount of time
- It represents a solid base

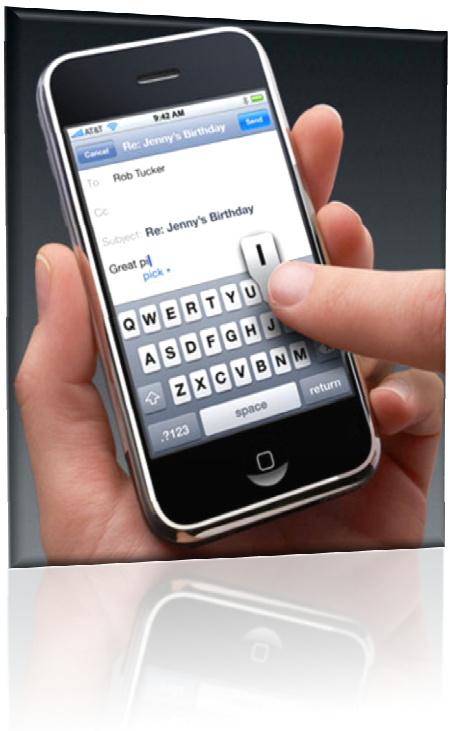
That said you have to choose a library very carefully, some might cause you a lot of troubles and conflicts. Being a software engineer is making the right choices.

2.3. Get some inspiration

Once you have defined the environment you're starting with and the tools you're working with, the journey begins.

Before coding anything you need to find inspiration and capture the essence of what makes an application great.
I found most of it in:

- Microsoft Live Messenger
- Apple iPhone



What's great about Live Messenger are the following:

- I'm using it (program for you)
- Most of my friends are using it (program for the others)
- It has a true identity
- It's one of the easiest to use (for now)

What's great about Apple iPhone are the following:

- It's very easy to use
- It has a "wow factor"
- It features great applications (multiple views and layers)
- It focuses on what's needed

"Perfection is achieved, not when there is nothing more to add, but when there is nothing left to take away"

-- Antoine de Saint-Exupery

2.4. Code it, until it works

Coding is not a scientific process. You can read as many design books you want. You can spend hours doing UML. In the end, I believe practice is what makes you good.

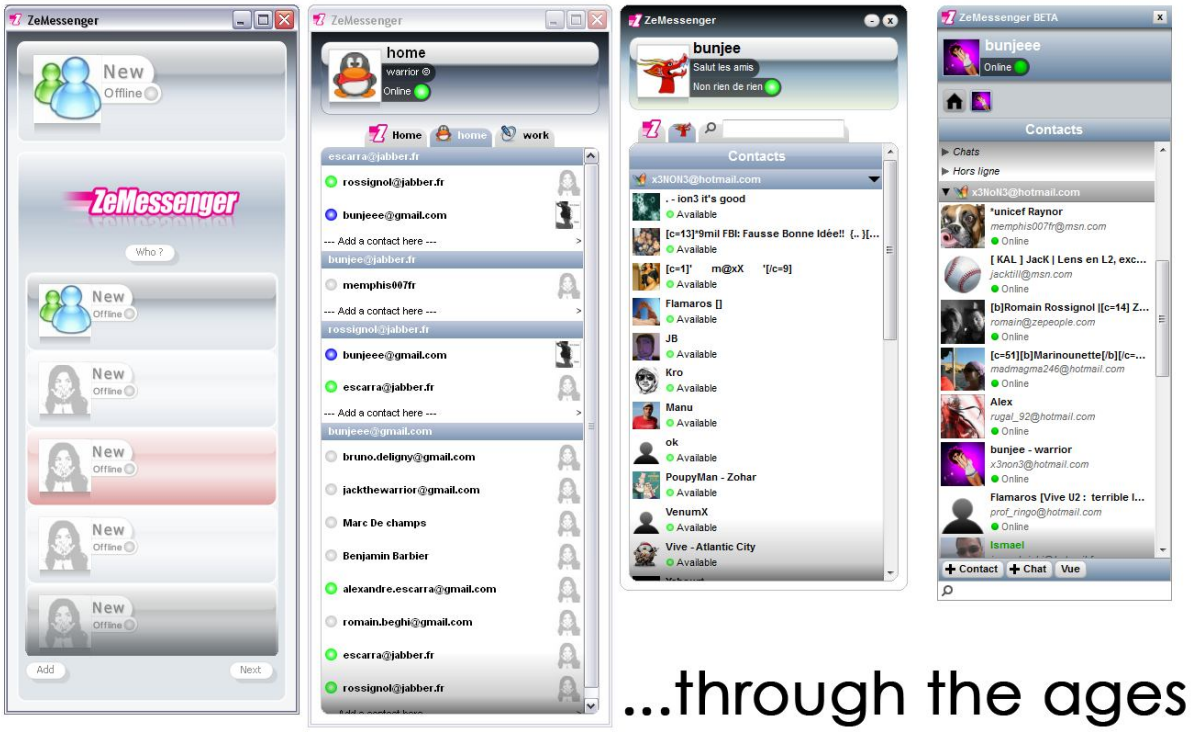
That said, you have to keep in mind some rules:

- Don't code something evolved before making it simple
- Think about scalability; don't code two times the same thing
- Divide your program into libraries
- Try to be generic, code as much reusable components as you can
- Pay attention to bugs, memory and performances on a daily basis
- Think about milestones, focus on what you're trying to achieve
- Don't move to the next step before being satisfied by what you've coded
- If you're using a framework (like Qt or Glib) stay close to their design or change it.
- Code like you would talk to a 3 year old baby; if you get smart, maintainers will hate you for that

"Success is a series of mistakes"

Instead of going through every step for making ZeMessenger I'm going to let it talk for himself.

Here is the program through the ages:



Here is a presentation of ZeMessenger:

- **An Instant messaging application**

ZeMessenger provides:

- A jabber IM account (like google talk).
- A Chatting platform for several users at the same time.
- A one-on-one chat based conversation to meet new people.
- A persistant contact list to keep track of your friends.
- A highly user friendly interface to make the experience fun.

- ***www.zepeople.com community***

ZeMessenger targets:

- Typical Zepeople user is:
  - A clubber, spending his time and money in clubs and bars with his friends.
  - Aged between 14 and 20.
  - A high purchasing power entity: “La jeunesse dorée”.

Zemessenger provides the community with:

- A way to communicate instantely with the whole community.
- A way to meet people from the same social class and not anybody.
- A powerful contact list to keep track of new friends.

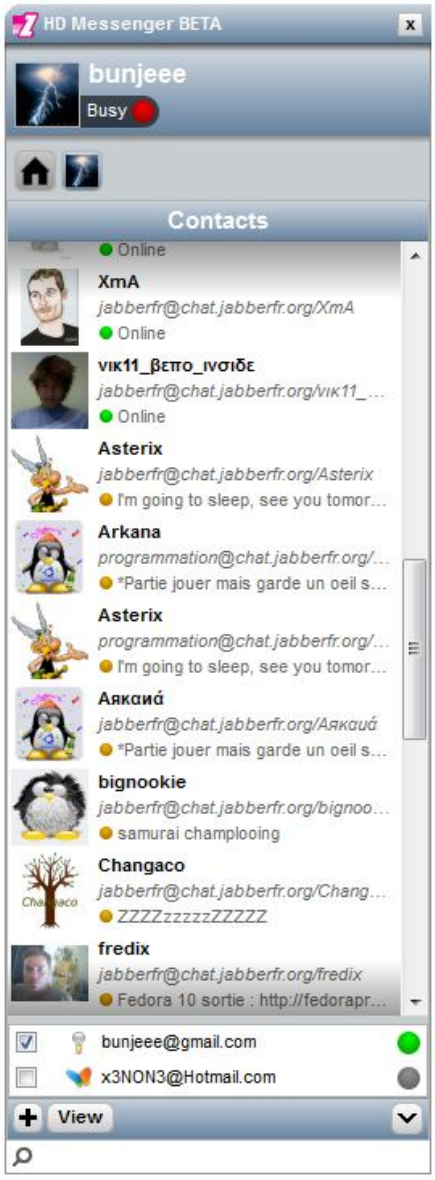
- ***www.zepeople.com community***

ZeMessenger will be out in March:

- 2 years of development.
- A premiere in french private communities.
- A brand new community friend making concept.

2.5. Features

Here is the marketing part.
Suppose for a second I had to sell you my product, what would be the killer features?



- Connect to several IM networks at the same time:
  - Live Messenger
  - Google Talk
  - XMPP
  - More to come…

- Smart interface
  - Auto-connect
  - Multi-page dynamic interface
  - All your conversation in the same window
  - Add a contact to your contact list in 2 clicks
  - Start a chat or a conversation in 2 clicks
  - Change account in 2 clicks

- Touch your contact
  - Cover flow contacts
  - Quick information on any contact
  - See your contacts in full screen

- Chat with millions of people:
  - Chat server list
  - Chat search server feature
  - Refresh all servers with 1 click
  - Chats directly in contact list
  - Add a contact to a chat in 2 clicks

- Attractive design:
  - Chat color bubbles
  - Font support
  - Font Color support
  - Smiley support
  - Multi theme support
  - Wallpaper support
  - Use your contact's picture as avatar

Here are potential ads:

2.6. What hasn't worked?

"If passion drives you, let reason hold the reins"
-- Benjamin Franklin

When we started developing with my co-mate we thought it would take us 6 month. In reality it took more than 2 years. Why such a gap between expectations and results?

- The **first** reason is what I call the unexpected. From my point of view 1 year was the time needed without the unexpected taken into account. You can try it with a small project, when you think: "it's going to take me 2 weeks to get this done" you should always add at least 50 to 100% to your estimation.

Unexpected factors are:

- **Bugs**: everything that makes a program crash
- **Memory leaks**: everything that makes a program crash after a while
- **Design flaws**: everything that implies starting over a component
- **Forgotten specs**: everything that is not in the initial design but needed

- The **second** reason would be what I call "not knowing what it takes before doing it". An IM software is a complicated piece of engineering. One couldn't think of all the details needed to make it work correctly before going through the creation process. I guess it's like being through a war, you only know what it really takes afterward. That said people just can't help criticizing:

"What is taking so long?"
"I would do the same in 1 month"
"That design sucks"
"You're a wanker"
"..."

Designing a good Instant Messaging application is a mammoth task.
Let me give you examples:

- Pidgin another open source client is a team of 20 coders
- Live Messenger as we know it today is not far away from the first version. It took 5+ years to have something convincing
- I don't have any example of IM apps developed by a single coder

- The **third** reason is doing more than your job. An application is not just lines of codes. A modern GUI is no longer a series of terminal queries. Now people want great design and good looking stuff. That's what "artists" are made for. I did everything on that side because I had nobody to do it for me.

- The **fourth** reason is bad communication. Like I said previously specs were not closed. When you keep that open during a development process you break the following pattern:

  Design > Implementation > Debugging

Say for instance you're on the debugging cycle and your boss comes back to you saying: "I need this and that". At this point, if the specs are not closed you need to go back all the way to "Design". When this happens you're extending dramatically the iteration process.

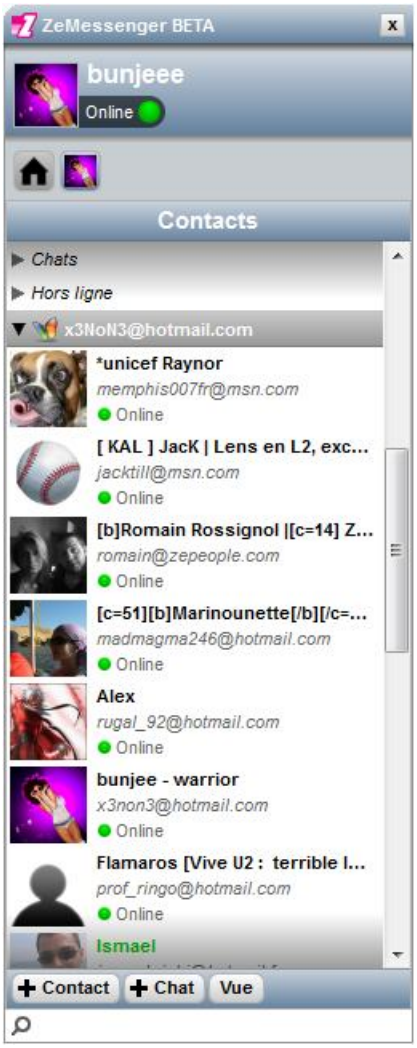
To sum up my advices are the following:

- Always take 50 to 100% more time than what you think you need
- If you've never developed a project like that, extend your time by 100%
- Define what your work is, don't forget you're a coder
- Define your specs for a version and then stick to it at any cost
- Make sure your boss believes in the project, if he doesn't you will fail
- Make sure you believe in the project, if you don't you will fail
- Show your advancement to others and ask their feedback
- Don't give up, code it until it works
- Treat your application like your baby, don't let people insult it
- Stay true to the project, focus on what you want to do
- Infuse some culture into what you're coding
- Only reinvent the wheel when it's needed
- Keep a high esteem of your work

All that said, I think having developed this on my own is a tour de force.

2.7. Results Vs Specs

Let's review features that **were** in the specs:



- Contact list:
  - o ✓ Add a contact
  - o ✓ Remove a contact
  - o × Grouped contacts
  - o ✓ Respect Zepeople style
  - o ✓ Avatar picture
  - o × Display contact's profile page
- Conversation:
  - o ✓ Independent window
  - o ✓Conferences
  - o × History
  - o ✓ Fonts
  - o ✓ Smileys
  - o × File transfers
- Help:
  - o × User help located on the website

Now let's analyze missing features:

- × Grouped contacts:
  Grouping contacts is a classical feature. The problem is we already have groups for the services the contacts belong to (if it's MSN or XMPP for instance). On top of that we've got a search contact list feature. That's why it was determined that feature was not needed for Beta 1.0.

- × Display contact's profile page :
  I didn't implement that feature for v1.0 due to the HTML and security requirements.

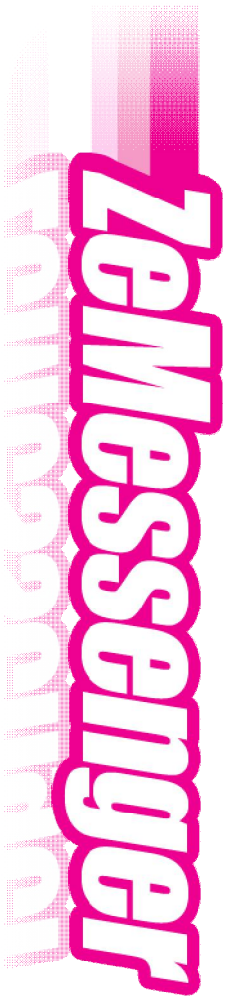
- × History:
  History is a tricky question because it changes from one computer to the other. Cloud computing might bring an answer to that.

- × File transfers:
  It was decided during the development that this feature was no longer needed for v1.0. I had plans to integrate a more sophisticated file transfer in the next version.

- × User help located on the website:
  Have you ever used MSN's help?

"Good software does not need documentation"
-- Benjamin Arnaud

Let's review additional features that **were not** in the specs:



- Contact list:
  - ✓ Search your contact by name on the fly
  - ✓ Notifications
  - ✓ Chats displayed in the list
  - ✓ Chat users displayed in the list
  - ✓ Disconnect an account directly from the list
  - ✓ Add a contact directly from the list
  - ✓ Add a chat directly from the list
  - ✓ Add a service account directly from the list
  - ✓ Change the view of the list:
    - ✓ Grouped
    - ✓ Global
  - ✓ Change the contact view
  - ✓ Change your status on the fly
  - ✓ Change your picture on the fly
  - ✓ Disconnect on the fly
  - ✓ Attach or detach contact and conversation window
  - ✓ Change language on the fly (English / French)
  - ✓ Change photo
  - ✓ Cycle through avatar photos
- Conversation:
  - ✓ Fade out on the contact's picture at opening
  - ✓ Displaying contact picture
  - ✓ Displaying your picture
  - ✓ Change your picture from the conversation
  - ✓ Get more info from the contact by clicking on it
  - ✓ Change font
  - ✓ Change font color
  - ✓ Start a conversation
  - ✓ Start a chat
  - ✓ Web view
    - ✓ Display any URL as a start page
  - ✓ Find a chat room:
    - ✓ Refresh every chat rooms at any time
    - ✓ Add a server to chat rooms list
    - ✓ Change the chat room view
    - ✓ Create a chat room
  - ✓ Touch your contacts in contact flow:
    - ✓ Start a conversation
    - ✓ Add a contact
    - ✓ Block a contact
- ✓ Design:
  - ✓ Original icons
  - ✓ Change your theme color on the fly
  - ✓ Change your theme background on the fly

As you can see we've got a bunch of additional features here.

2.8. ZeMessenger will never be

Now that you know ZeMessenger and hopefully like it 1% as much as I did, it's time for the sad part.

This program is supposed to be released
The Beta launch date was scheduled for July 2008. Everything was ready on my side.
I had:

- A release version
- A Windows installer
- A branded icon
- A Jabber server deployed
- A deployment plan for the website including:
  - Download page mockup
  - Front page advertisement

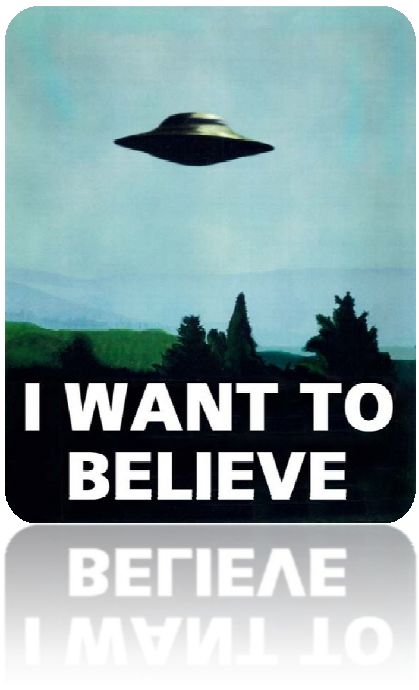
I sent a fully detailed procedure to the CEOs.
What remained to be done on their side was:

- Integrate my deployment plan on their website
- Upload my Windows installer
- Put the advertisement on the front page

"The worst thing for a programmer is not being criticized; it's to develop something that will never be used"

-- Benjamin Arnaud

This program should be released.
Remember my recommendation: "Make sure your boss believes in the project, if he doesn't you will fail"?
My boss told me that the simple fact of chatting with others community users would not be enough to launch the product.



Remember my recommendation: "Define your specs for a version and then stick to it at any cost"?
He added the application needed to have something special and different from other instant messaging applications. He had no idea what, but he wanted me to refine my design once again. Except this time, he would not pay me.

Remember my recommendation: "Keep a high esteem of your work"?
I decided to quit. I understood a little too late that:

- They would never pay someone to integrate my application into their website.
- They wanted me to code a "killer feature" they didn't know themselves
- They would no longer pay me for anything
- They were scared to lose their community
- They didn't believe in the project

"How people treat you is their karma; how you react is yours"
-- Wayne Dyer

This program is not exactly released.
Today is Sunday 31 November, what's left of ZeMessenger?

It's now called Codename Messenger; you can download an alpha version on my website.
But without the community aspect, it hardly means anything.

This project has attracted head hunters and I had interviews with:

- Microsoft
- Canonical

Both of these interviews were a failure.

This program will never be.
But remember that last recommendation: "Don't give up, code it until it works".

Benjamin Arnaud