Curriculum Vitæ

Facundo Batista

Personal Information

Name	***Facundo Batista***
ID	D.N.I. 24.379.404
E-Mail	facundo@taniquetil.com.ar
Date of birth	May 02, 1975
Nationality	Argentina

Resume

Formal Studies

Master in Innovation Engineering – *Bologna University, Italy*

Electronic Engineering – La Matanza National University*, Argentina*

English - *Certificate of Proficiency, English Culture Superior Institute*

Italian - *Bologna University, Italy*

Professional Development

Python - Python Software Foundation: *Core developer, PSF Member, Python Instructor and Conferencist*

Canonical: *Senior Software Developer (Oct-2008, ongoing)*

Compañía Ericsson SACI: *Solution Architect (Jun-2007 to Oct-2008)*

Cyclelogic: *Applications Developer Chief (Mar-2006 to Jun-2007)*

Telefónica Móviles de Argentina – Movistar: *Services Development (Apr-2004 to Mar-2006), Mobile Network Management (Jan-2000 to Mar-2004)*

Universidad Nacional de La Matanza: *Professor (Apr-1999 to Dec-2005), Technical Support (May-1998 to Nov-1999).*

CENCAP: *Informatics Teacher (Apr-1998 to Aug-1999)*

Consultant Profile

Summary

I'm a mature professional, with more than 8 years of mobile telephony related experience and 25 years of software development experience, mainly in Python.

Because of the different stages of my work and my educational path, I developed a balanced mixture of technical, business and human skills.

Some of my strengths include:
- experience in complex systems integration project
- focus on knowledge sharing and very experienced working in human networks
- management experience
- consultant approach maintaining a strong business focus and customer oriented, with strong business development interest in new opportunities
- good international experience and fluent in Spanish and English
- problem-solving attitude

I'm a specialist in the Python programming language. With an experience in it of two decades, I'm Core Developer of the language, and member by merit of the Python Software Foundation. I gave talks in the main Python conferences in Argentina and other countries (United States, Europe, and more). In general, I have a strong experience in distributed collaborative experience, being involved in FLOSS development, working with people around the globe, for more than 20 years.

I have an extensive career regarding services, because in all its stages I was involved in providing services to internal or external customers, in public or private services. In all the following ons I was involved in one or other way as Solution Architect, but as in the other Enterprises the role separation weren't as clear as in Ericsson, I normally worked also as Project Manager, and/or Solution Integrator:
- International roaming
- Network data and statistic services
- Mobile core services
- Value Added Services (SMS, WAP, MMS)
- Software development (in house and open source)

I have good leadership abilities, expressed mainly by managing groups, and persons separately. Specially when managing groups, I always try to involve each person in the team workflow, and encourage them to add their abilities to the task.

I'm very pedagogic in explaining things to other people, ability developed in all my years as teacher and professor, and exercised daily in mailing lists in the open source world. I'm also always wanting to help, and enforced by my technologic understanding of the problems, I develop a trust and respect relationship with colleagues and people in general, resulting in them providing me information and expecting me to analyze it and add value to existing solutions or develop new ideas.

Selection of assignments

Canonical cloud services

Two sets of services stand out in what my teams and I built in my last years of Canonical.

One was of a myriad of services for the Ubuntu Phone, mostly consuming backends services and readapting content, but also generating language aware information to enhance the user navigation experience.

The other is the Snap Store, a mix of old monolithic and new micro services, interacting to provide the whole snap experience: from developers creating, pushing and distributing code, to devices installing and refreshing snaps (being those supervised by human users, or everything automatically, representing the state of the art in the IoT world).

Ubuntu One File Synchronization

This was the core service of Canonical's Online Services infrastructure, and allows users to keep their files synchronized to the "cloud".

The service was splitted in two big components: the client, installed in user machines, and the server, hosted by Canonical's datacenter. Both components are developed in a fully asynchronized way, using Python's Twisted.

The client (open source software) is a big state machine that is always listening to disk events and working with the server to keep the directories trees fully synchronized (supporting recovering from events that happened while client was turned off) and also receives notifications from the server about events that happened in other user's machines, replicating those changes locally (of course, also supporting recovering from events that happened while client wasn't connected to the server). The main complexity of this software is the ability to handle all the combinations of actions that different programs do over the files, and the wild diversity of environments (millions of installed clients around the world across different versions of Linux, Ubuntu and Mac).

The server component is designed as a service of operations, getting commands and information from clients, keeping the state of user's files in databases, and notificating other clients (connected, or when they connect) about changes in that state. It's built in several layers, using a custom built RPC (using ZeroMQ) and eventually hitting the database shards (PostgreSQL) and file storage services (Amazon and others). It can scale horizontally, serving in production to dozens of thousands of connected client per server.

Fully Controlled Switches

The project was called, in Spanish, "Centrales Completamente Controladas". Its objective was to have an assurance that the internal data of the mobile switches (MSCs) was perfect, to be able to prove that the service quality, generated revenue, etc., were the best possible at that level.

This was an internal project of the Network Data Management department, with many internal clients of this Mobile Services company: Fraud department, Revenue assurance department, Quality department, Information Systems department, etc. However, this data affected directly to the final users service, so that brought a lot of complexities to the project (technical, commercial, etc.).

As this was my idea, from beginning to end, I have to write first the Scope and

Proposal, to "sell" the project to my manager. After the approval, I and other people started to work on it, according to the Matrix of responsibility and the design docs I wrote.

As it was a critical part of the delivered mobile services, a strict user and functional acceptance tests were developed, and only after they were satisfied the system got into production. As I later moved on to other tasks, I wrote system and operation guides to let other people to continue working with the system, and I trained them.

The biggest change this system brought to the department was that before it we received more than 150 complaints per day, and after the main part of the project was in production, only one or two per week.

Calls Statistics System

Its original title was "Sistema de Armado y Clasificación de Llamadas" (SACLLA). Its objective was to process the call data records that the mobile switches generate, in order to be able to generate high level, meaningful statistics.

As these statistics show a lot of useful information, there were a lot of internal clients of this Mobile Services company, and even external ones. As of internals, we can mention the following departments: Fraud, Mobile Network Access, Mobile Network Core, Marketing, Information Systems, Revenue Assurance, etc. In the external front, I must mention that results of this system were used to validate invoices from different providers, and to report monthly audits from the headquarters. This system was later sold as a turn-key solution to a brother company in other country.

This project was initiated because of very specific needs I had to satisfy. I came up with the solution, and after writing the Scope, Proposal and Costing (as it involved hardware) it was approved. As it would imply changes in auditing processes, I also had to write a detailed requirement specification, design docs and critical factor of success and then comply with them.

For the project to exceed the department and be integrated in the rest of the organization, I had to pass user and functional tests, and train other resources in the enterprise that would use and get useful information from the system.

When the project entered into its production phase, the Technology Direction was able to have detailed and critic information, not only to improve inner process and methodologies, but also to provide data and to others sectors of the Operator.

SMSC solution

The assignment was basically to bring enterprise-class SMS service to a Mobile Services company. The SMS solution existed at that time, but with a very small capacity, and it did not give all the services that are now standard.

As an Engineer in the Services Development department, I had to propose what the solution could be and how to integrate it with the mobile network and the rest of the Operator's systems. Detailed scopes of work and acceptance test criteria were developed, to be sure that the solution provided would satisfy the needs.

I had to coordinate not only the Provider resources, but also other internal departments whose services were essential to the project (Core Mobile Network, Infrastructure, Information Systems, etc.). For this I prepared a very detailed matrix of responsibility, which shaped the project flow and even the acceptance and pre-production tests.

As the project evolved during time, we had to handle a lot of changes from internal clients (Information Systems regarding billing, Marketing regarding new or modified services); some of these customizations were handled internally and some were solved through the Provider. I also had to train other departments of the organization that were involved directly with the product, most noticeable the Operation and Management Department.

In the two years I been on charge of the service, the quantity of SMSs per day multiplied 50 times, the International Roaming was activated to this service (more than 100 countries), interoperability was achieved with all the national competitors, and even was executed a merge with other Mobile Services provider.

Integrated Billing System

The Content Provider Company where this project was developed needed to interconnect to online billing systems of the Mobile Operators. Different operators in different countries had different billing systems, so an Integrated Billing System was needed to hide this complexity to the universe of already existing and future applications.

The company also sells, to 3pps, SMS connectivity to the Operators. Even as the billing system was developed at first for internal use, the company started to sell this service.

As a Chief Developer, scope and costing were critical documents I had to produce, but also needed to create the proposal document, as the original one was much too wide and general. As part of the project was to be implemented by people in my department, I developed detailed design documentation, and a matrix of responsibilities. The migration process was critical in the project, because of how it impacted in the billing of the enterprise, its clients and providers. That's why the acceptance tests were executed primarily against internal Administration department.

The project was implemented successfully, and handled seven Operators, with seven different on line billing systems, in five different countries, with an average load of 30 transactions per second.

PEP 327 – Python Decimal module

The Python programming language lacked for more than a decade of decimal floating point numbers. It always had binary floating point numbers (like "C", "C++", etc.), but this type had some intrinsic problems when you need to have accuracy to some decimal digits in your calculus (this problems are especially harmful when your system is dealing with money). So, as I needed the Decimal type, and Python is Open Source, I took the assignment to implement it, making first the community to agree in the project to be started, and then carrying it to the end.

My task is well separated in three parts. The first and essential part is to have an agreement of the community of what the Decimal module should have, how it should behave, etc. This involved hundreds of mails in the development mailing list, during several months, and its result is this formal document: the PEP 327 (http://www.python.org/dev/peps/pep-0327/).

When this document was stable, I took a partial implementation and finished (code, test cases and documentation), leaving it with the quality requirements to be included in the Standard Library of the programming language. Since version 2.4, Python brought this module to the world.

After it came out to the light, it entered in maintenance mode. It has a knowledge sharing part (answer questions in mailing lists, giving talks in conferences, etc.). Also, the code needs to be revisited, time to time. Bugs appear, of course, but most important is that the Decimal Arithmetic Specification get revisited by IBM, so this module evolve in time, and is my task to keep it updated (or to lead people to do it).