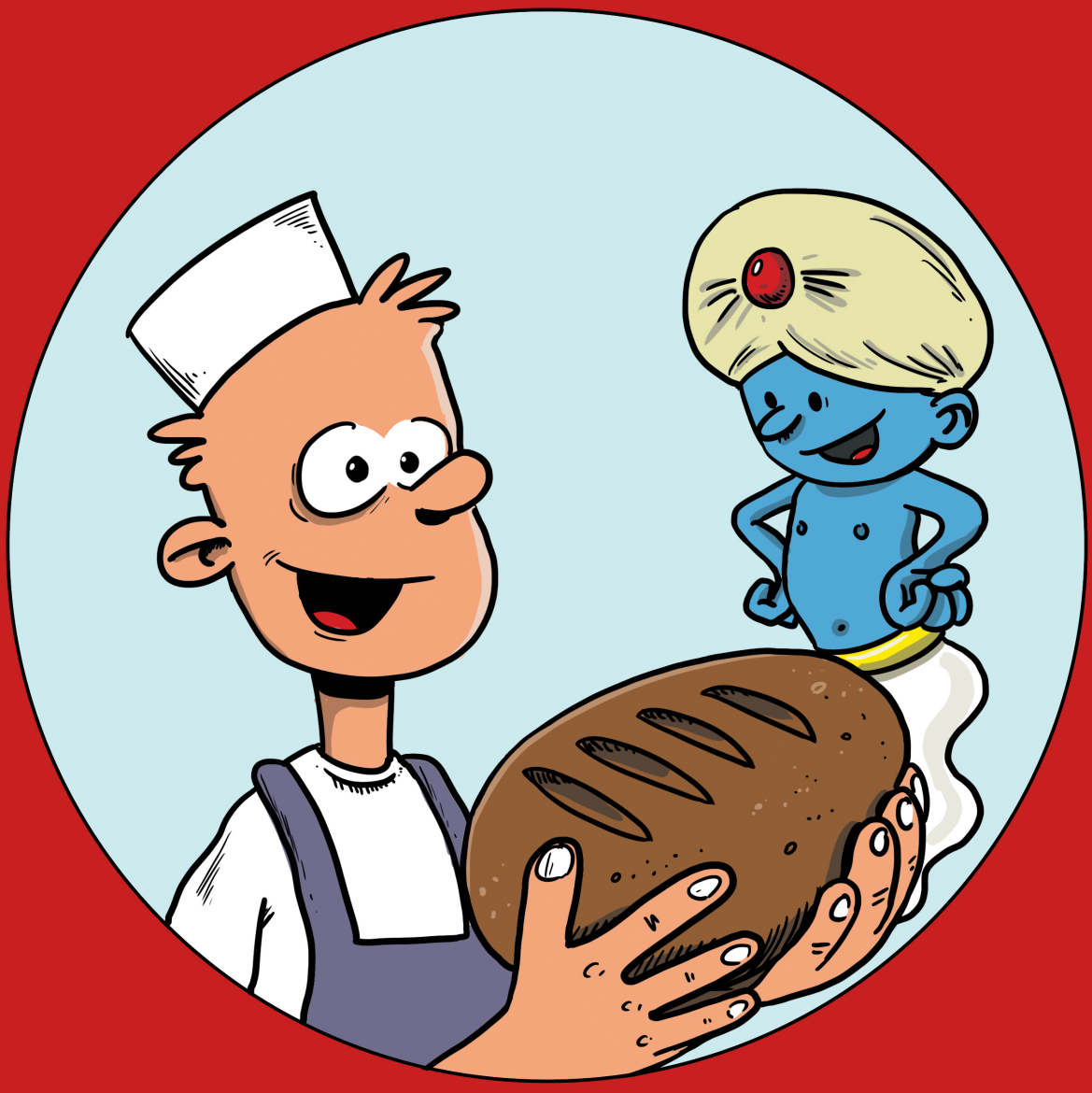
# GEORGE THE BAKER



# George The Baker

Interactive Version at http://www.GeorgeTheBaker.com

Center of Excellence in Enterprise Architecture
www.ceisar.org

Version 1.0
December 2013

# Introduction

## CEISAR has devoted itself for 6 years to Enterprise Architecture

CEISAR (Center of Excellence in Enterprise Architecture www.ceisar.org) has written white papers on the subject of Entreprise Architecture for 6 years.
Enterprise Architecture is the overall Model of the Solutions that an Enterprise chooses to Produce, Distribute, manage its Resources and its performance. Its global vision enables it to provide a coherent framework in which the different Enterprise Solutions are included. This framework enables us to gain in simplicity, agility and synergy and requires a structure to be set up that is independent of the Business side.

## But a broader framework is needed

But, the accelerating pace of Enterprise Transformations is changing the rules of the game: Enterprise Architecture can no longer be isolated as a separate discipline from the strategy definition, Product policy or change management.
We have therefore tried to define a more general framework which sets out why and how to lead Transformations. Of course, Enterprise Architecture is part of that, but it is only one of the themes needed for successful Transformations. We have had to present the workings and Transformation of the Enterprise in its entirety to help with a multidisciplinary approach to Transformation: Strategists, Product designers, marketing, Solution builders whether they are on the Business or IT side, Organizers, trainers, HR managers... have to be involved in the same process, within project teams organized around a Goal to be reached.

## A multidisciplinary Transformation language

There are many documents, works and articles that deal with these problems. Our objective is not to reinvent the wheel: we have tried to base ourselves on what is already available.

The biggest challenge when you cast your net wide is to be understandable by everyone. It is therefore no longer a question of using discipline-specific jargon: we have to share a rigorous and simple common language and the same vision of exactly what the Enterprise is.

Our only contribution is to have tried to present these themes using a common language where each concept is rigorously defined in a glossary.

## In the form of a comic strip

Regarding the format, we were extremely surprised by the fact that one of the most downloaded white papers from CEISAR was the story of a baker who illustrates, in comic-strip form, the development of a bakery Enterprise. We used this story to introduce the main themes of Enterprise Architecture. We tried to compensate for the aridity of the subject with a touch of imaginativeness. We therefore "listened to our Customers" and used the same Model again to illustrate the broader theme of Enterprise Transformation.

To build the scenario, we began by listing the 78 essential themes that concern Transformation to organize them into a story in 10 acts, divided up into as many scenes and there are themes.
We turned to the Belgian comic strip illustrator, Tonu, whom we would like to thank for his work, his pedagogic input and his humor.

In view of the number of themes, we haven't tried to detail each one of them but simply to illustrate what we consider to be the main ideas in a document entitled "learn more".

## George the Baker : Entrepreneur and Transformer

George the Baker is both an Entrepreneur and a Transformer : indeed, starting an Enterprise and carrying out large projects in existing Enterprises, use the same practices such as Modeling the Offer, Modeling the Operations, using new Transformation approaches... (subject covered here).

## Who can use this story?

Our objective is to help those whose ambition it is to rapidly Transform Enterprises or to start new ones.

Everyone can therefore use this story, which is available royalty free under a "Creative Commons" license; you can even use it in training courses that you charge for: simply indicate the source and make no modifications to the original (but do not hesitate to post comments).

The story can be used, for example:

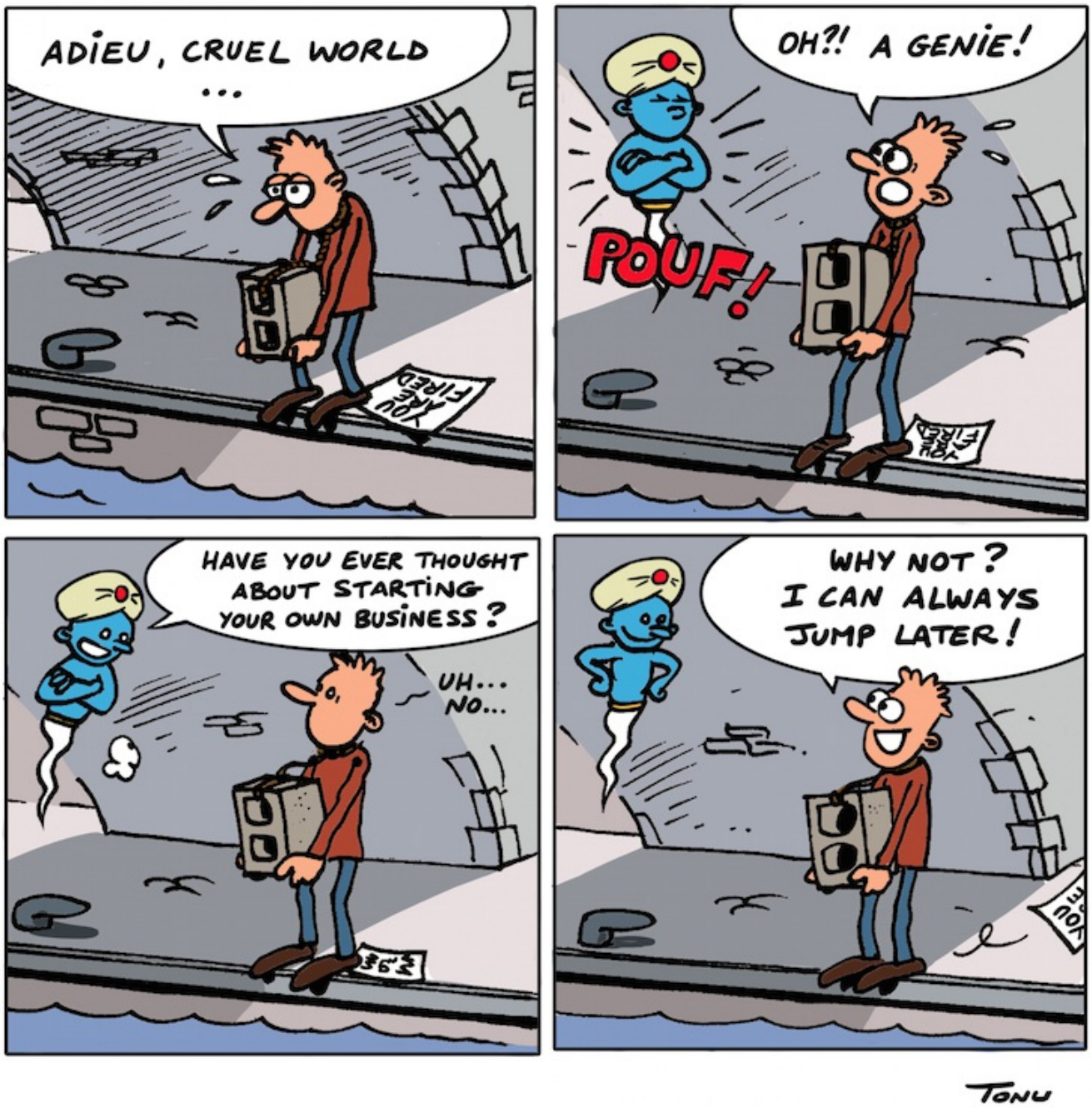
- By teachers wishing to present what an Enterprise is, its role, its start up and its transformation.
- By continuing professional development providers.
- By Enterprises, at the start of complex projects, to bring together a Transformation team.
- By those who define the new forms of organization in Enterprises.
- By those who define a multidisciplinary Transformation approach...

## How to improve this story

We have initiated this story, defined a glossary of key terms and written the ideas that seem the most pertinent to us. But this initial version is far from being perfect. In particular, certain themes are not yet very developed and we need expert help to give substance to them. We would like to see others contribute to its development. We can keep the thread of the story but go into more detail on the recommendations of each scene.
Anyone can, via the "Comments" and "Contributions" sections, participate in enriching the story: to integrate your contributions, we will use the same language so as to keep overall consistency.
A French version is circulated in priority; an English version will follow the month after. We will try to maintain both versions if we have enough means, or if some people would like to help us with the translation.

## Act 1: Start one's own enterprise

# Why not create one's own job?




## 1. Enterprises are replaced: so create yours

**Enterprise life cycles are shortening.**

The 100 largest American companies in 1966: what has become of them 40 years later?

- 19 are still in the top 100
- 15 still exist but are no longer in the top 100
- 66 have disappeared!

Globalization and technological innovation accelerate this movement. How many of today's Enterprises will still be here tomorrow? We can assume, given the current movement, that in less than 40 years it is likely that 2/3 of them will have disappeared.

General Electric believes that 30% of its current activity will naturally die off within 10 years.

Standard and Poor's have announced that enterprise life spans are inexorably decreasing:

- 75 years in 1957
- 25 years in 2003
- 10 years in 2013

**If many enterprises disappear, it is because the rapid renewal of Products is reshuffling the cards: more and more Enterprises are starting up**, replacing the existing ones...
Belonging to a "system" is no longer a guarantee of security for the individual; even job security in the Civil Service is threatened today in a growing number of countries. True **security** is now linked to one's own **Competence**: the ultimate competence being the ability to create one's own enterprise and job.

## 2. How can we know if we are capable of starting an enterprise?

But how can we know if we are capable of starting our own Enterprise?

Most Entrepreneurs had no experience of starting an enterprise before actually doing so, and yet, a large number of them have been successful. Not necessarily first time round, but during subsequent attempts: we have to remain confident about our ability to progress in the art of entrepreneurship.

In actual fact, it is impossible to know in advance if we will succeed; the only rule that we can set out is that we must not hesitate to try our luck if the desire is there so that, when we are 80, we have no regrets about not having tried.

Some state that they are not 20 any more and that they have passed the age for taking risks. In reality, most enterprise entrepreneurs have had initial experience as an employee in an existing enterprise. We start an Enterprise, on average, around 40.

# Start with Value


## 1. Define the Value

The *raison d'être* of any Enterprise is to bring **Value** to its Customers by delivering Products. Before describing the Product Offers, we therefore have to understand the Values that the potential Customers expect.

### 1.1 Values for the Individual Customer

We have tried to reclassify the different proposed Values in a single list, which is split into 3 categories: primary Values, social Values and personal Values. Each Value is defined by a verb.

- **Primary values**

        - To be alive: Eat, drink, breathe, sleep, heat, shelter, take care of our health
        - To be safe:
            - protect the person and his/her possessions
            - ensure moral safety
            - live in a stable and predictable environment
        - To optimize one's resources
            - Optimize the quality/price ratio of Products obtained
            - Invest one's financial Resources intelligently
    - **Social values**
        - To be accepted by others
            - Belong to a group
            - Communicate: language
        - To be recognized by others
            - Have accomplished a difficult task
            - Display the attributes of success or good taste: have important and visible Goods, be well-dressed (brands)
            - Be above others: glory, power
        - To be loved
    - **Personal Values**
        - To enjoy comfort
            - Ease of use of Products used (we speak therefore of the **Usability Value** of a Product)
        - To have self-esteem
            - Be useful
            - Be self-confident
            - Accomplish difficult tasks
        - To have competences
            - Know how to (read, write, count...)
            - Have access to knowledge
            - Understand
        - To have pleasure
            - Excel oneself
            - Shows, Arts
            - Games, sports, tourism
            - Sex
        - To be able to get around
        - To feel free
        - To believe in a future life; wisdom

## 1.2 Values for the Enterprise Customer

We propose the same split for the Enterprise Values:

- **Primary values**, those that enable the enterprise to live
    - To be profitable (or benefit from donations or subsidies and grants)
    - To have good Resources at their disposal: cash, personnel, premises
    - To grow
    - To respect legal rules: tax system, regulations
- **Social values**, linked to the relations that the Enterprise has with the outside world: its **image**
    - Offer quality Products: functionalities, robustness, ease of use
    - Create quality customer-oriented Processes: distribution, after-sales service
    - Act honestly
    - Respect the environment
- **Personal values**, linked to internal relations: its **culture**
    - Team spirit, commitment and primacy of the collective interest
    - Professionalism, responsibility
    - Ambition and innovation
    - Justice, ethics

- Conviviality
- Access to information

This list could certainly be improved, but it can already help Product designers to identify the Value that they would like to bring **before** designing the Product.
This approach enables Product managers to focus their attention on a specific objective, that of satisfying preselected Values, which should enable them to remain focused and not get sidetracked.

## 1.3 The Customer plays several Roles

The Customer is the natural Person or legal entity that the enterprise brings Value to through its Offer.
The Customer can be an individual, a group of individuals (like a family), or a legal entity (enterprise, association, public organization...)

**Several roles** are grouped together under the term Customer:

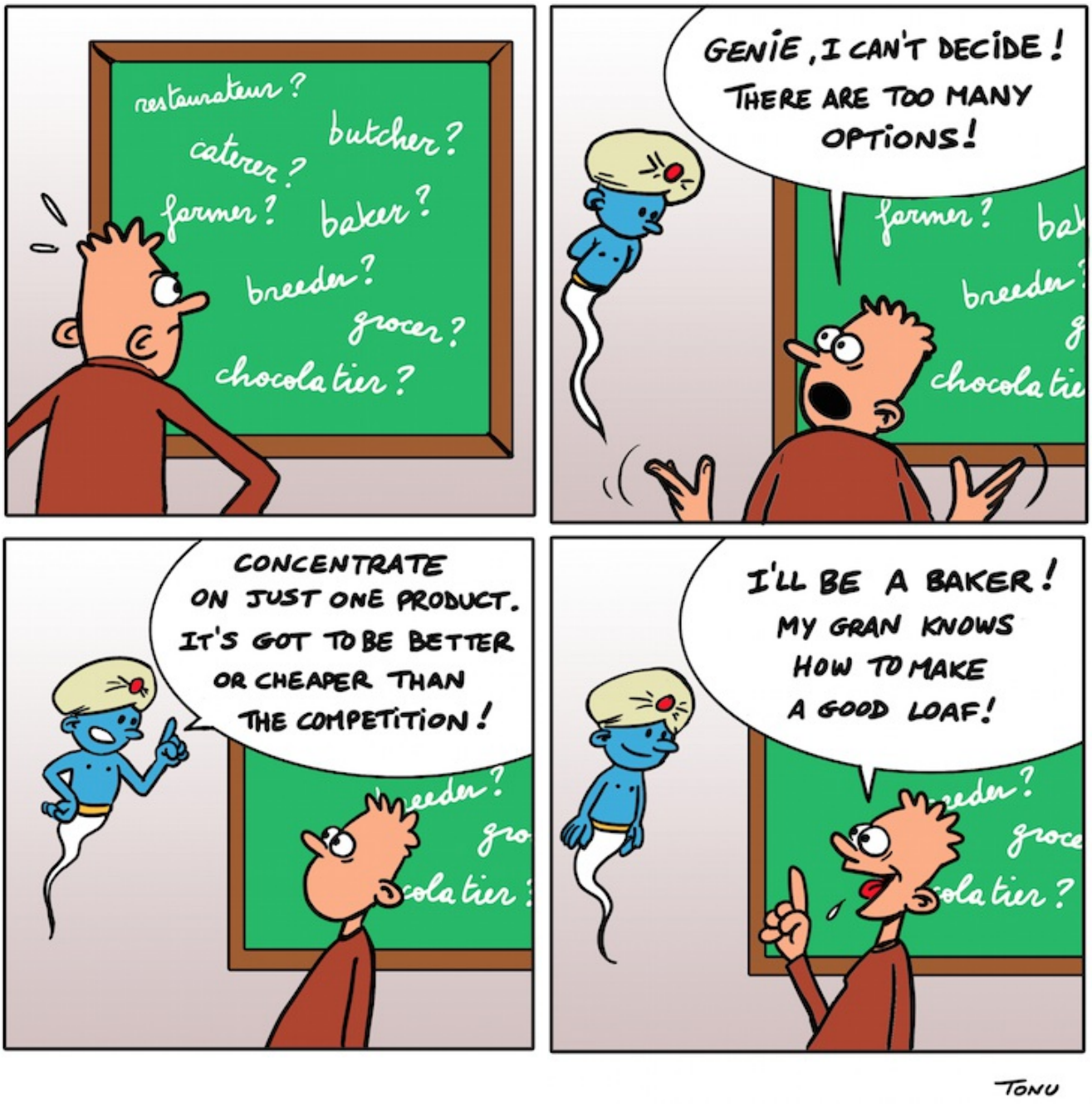
- Decision maker: this is the person we must convince to purchase the Offer
- Subscriber: the one who signs the purchase contract or the order
- Recipient: the person to whom we deliver the Product
- User: the person who derives Value from using the Product
- Beneficiary: the person who benefits from the Service
- Payer: the person who pays the compensation of the Offer

The same natural personal can play different roles: but it is not always the case. In the context of successive partnerships, the same Enterprise can be both customer and suppliers of the Products.

- Example of a fabric company for car seats; some distinguish between:
  - the direct customer (car-seat manufacturer),
  - the end customer (driver),
  - the strategic customer (automobile manufacturer)
- Ditto with the aluminum sheet-metal manufacturer, its customer manufacturing cans, its customer Coca Cola and the end customer who drinks coke.

These examples illustrate the fact that we must define the concept of Customer (or rather the **concepts** of Customer) before we begin Modeling the Enterprise.

# Identify a Product Model




## 1. The Value is brought by the Offer

Once the Value has been defined, we have to design a **Value Proposition**: we will call it the **Offer** for short.
For the same Value, different Products can be used. They can be assembled within the same Offer or compete with each other. As an example, to obtain the Value "be in good health", we can turn to a dietitian, a sports coach or a Tai Chi instructor.

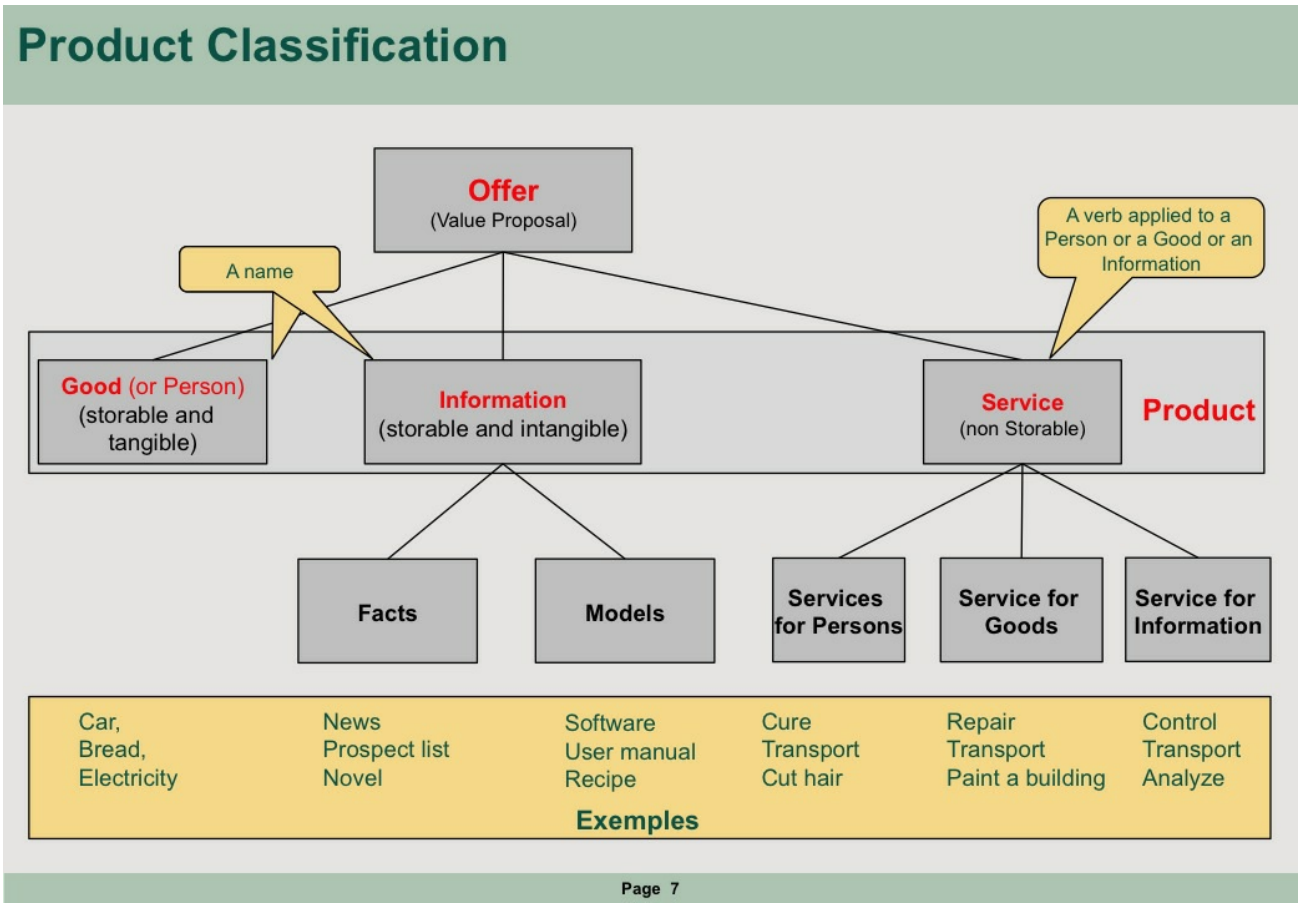
## 2. An Offer is made up of Products

There are three kinds of Products:

- The **Good** which is **storable**, tangible and defined by a **noun** (e.g., a car, electricity)

- **Information** which is **storable**, intangible and defined by a **noun** (e.g., stock market prices, software)
- The **Service** (e.g., cut someone's hair, repair Goods) which is not storable and is defined by a **verb**.

The Information **life cycle** is close to the Goods life cycle: we Produce, Distribute, stock, deliver and use Information as we do with Goods.
On the other hand, we Distribute and Produce a Service, but we cannot stock it, deliver it or use it.




# 3. Two kinds of Information Products: Fact or Model

**Facts** are mainly pieces of Operational Information.
Examples: news, weather information, stock market prices, music, account status...

**Models** formalize the real world, in **document** or **software** form, to simplify it, communicate it and transform it.
Examples: know-how, instruction manual, procedure, music score...

The **knowledge** industry consists of transferring Information: facts and Models.
We can, for example, analyze, in educational programs, the things that concern Facts (date of a battle, quotation) or the things that concern Models (a mathematical equation or spelling rules).

# 4. Models

## 4.1 Product Model for Goods and Information

It formalizes:

- The **structure** of the Product: nomenclature and options
- The **Use Model** (or Usage Mode of the Product): in the form of an instruction manual

or software
- The **Product Value**
  - The **basic Value** attached to the functionalities the Product provides
  - The **Usability Value** of the product (e.g., the success of the iPhone is down to its ease of use)
  - The **image Value**
- The **cost** of producing the Product

## 4.2 Product Model for Services

It formalizes:

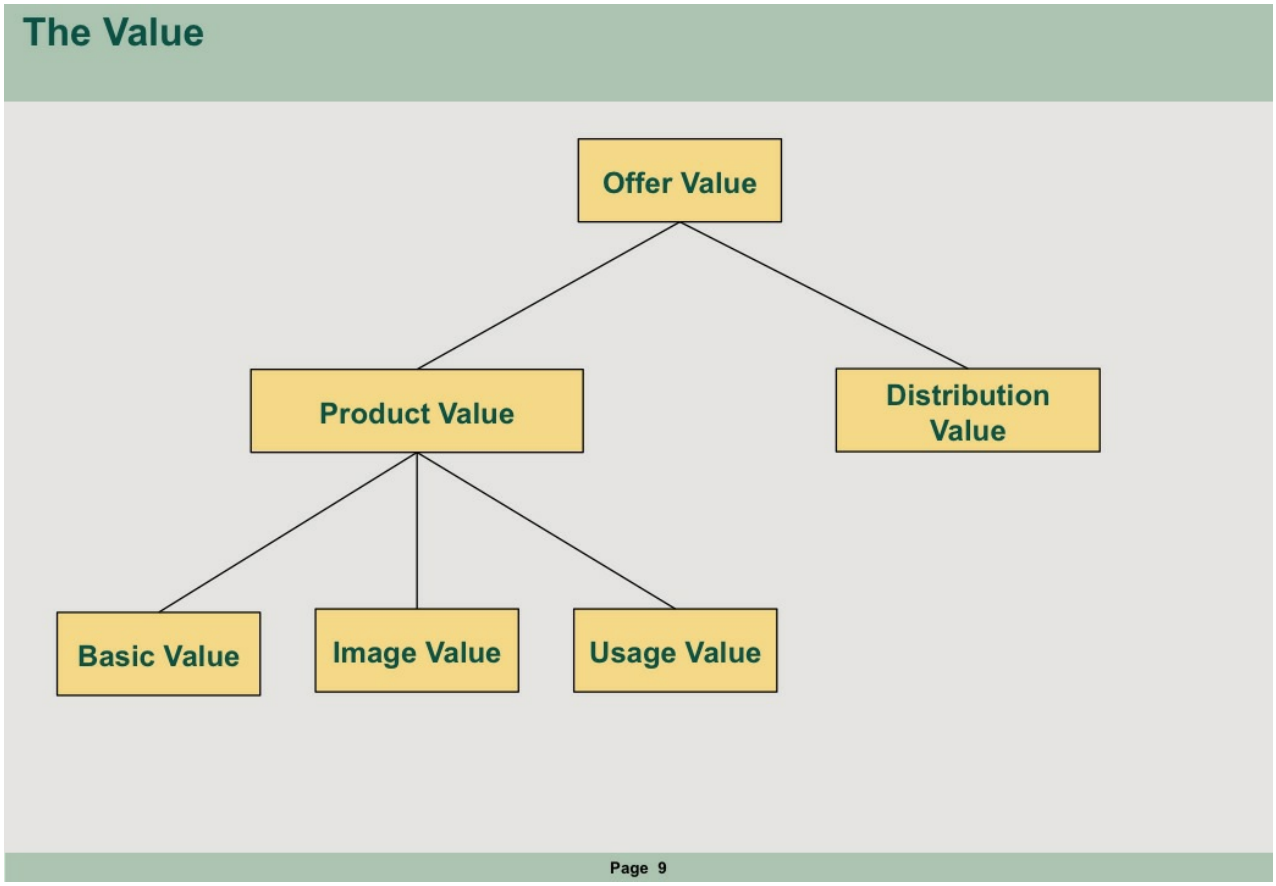
- The Process **Model**, as it is an Action
- The **Service Value**, which is its basic Value
- The **cost** of producing the Service

## 4.3 Offer Model

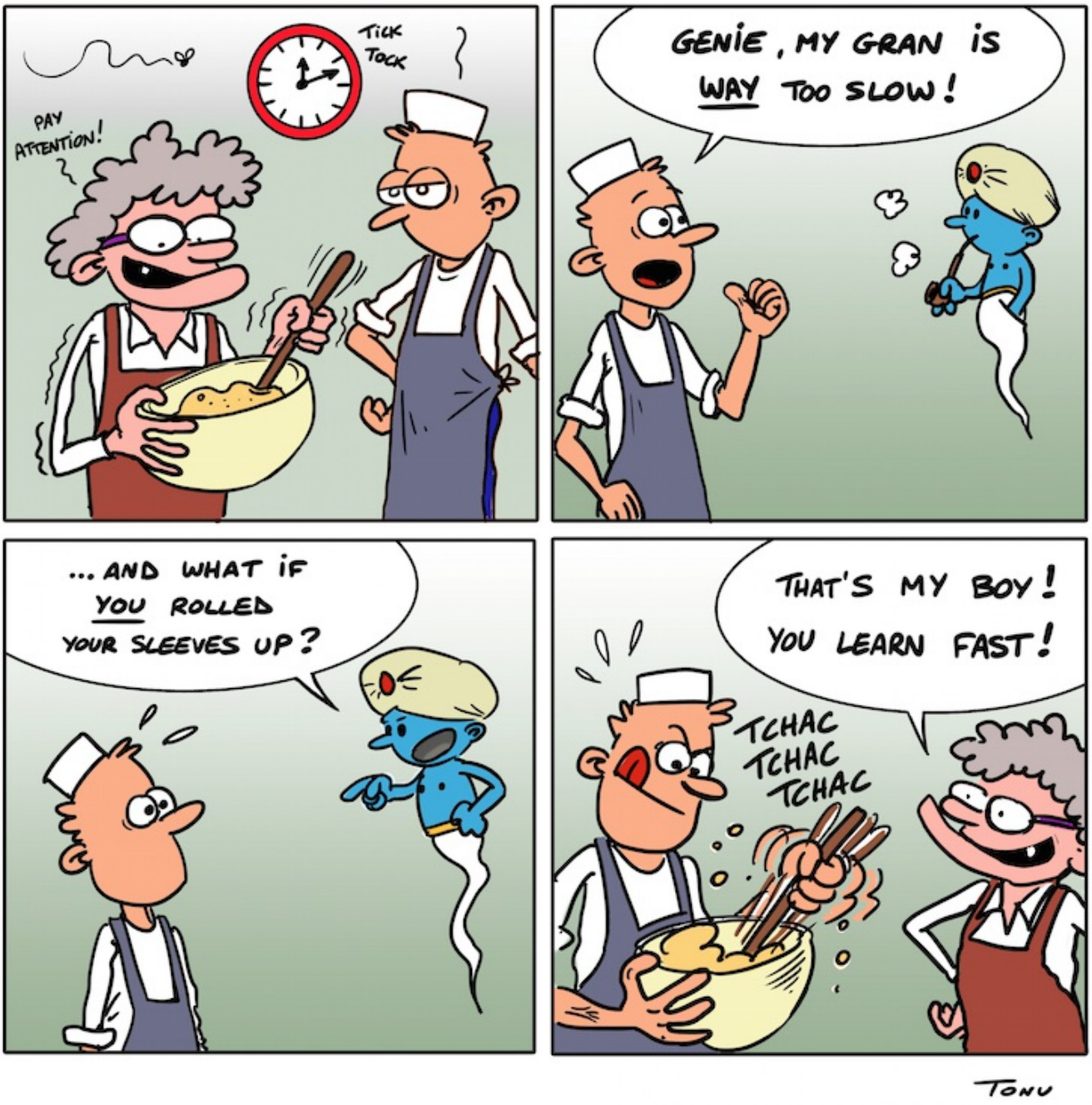
The same Product can be part of several Offers: either because we create different combinations of Products for different Customers, or because we Distribute them through different channels. We therefore need not only to formalize the Product Model, but also the Offer Model (which is proposed by the **Distribution**).

The Offer Model formalizes:

- The assembly of Products (this can be reduced to just one)
- The **Distribution Processes** (eg., direct order through Internet): how we identify the right Offer, how we are welcomed, how we buy, how we deliver and install the products
- The **conditions** of the Offer: price Model, conditions of eligibility
- The **Value of the Offer** which is not only the Value of the Product, but also the Value of the Distribution Processes : we are more inclined to buying if we are welcomed well, in a nice location, if we can buy through Internet, if we receive advice, delivery...


The Value
Offer Value
Product Value
Distribution Value
Basic Value
Image Value
Usage Value
Page 9

# Gain know-how




## 1. Learn before innovating

Starting an enterprise presupposes knowing how to do something better than the others:

- either a better Offer because it brings a **new Value**,
- or an offer than brings the same Value at a **lower price**.

This presupposes that we **start by understanding** the Offers on the Market and/or the Processes which allow them to be Produced and Distributed.

If the transfer of know-how does not exist, each human being would be forced to reinvent everything him- or herself: humanity would not be able to progress, as all knowledge accumulated during a lifetime would not be passed on.

Luckily, from our earliest age, our parents, relations, school, enterprise, media... transfer know-how to us, the basis to which we can add our own contributions, which may perhaps be useful to our successors. This is the principle of research: we base our original result on a list of publications containing the leavens of this result. We learn how to do well by observing those that know how, **before** innovating ourselves.

We must therefore be on our guard against speeches that recommend learning nothing from the others so as not to stifle our imagination. Even geniuses like Mozart or Picasso would not have been able to express themselves if we had not taught them musical notation or drawing techniques.

That is what explains that a majority of entrepreneurs have started by learning a job before thinking about improving it.

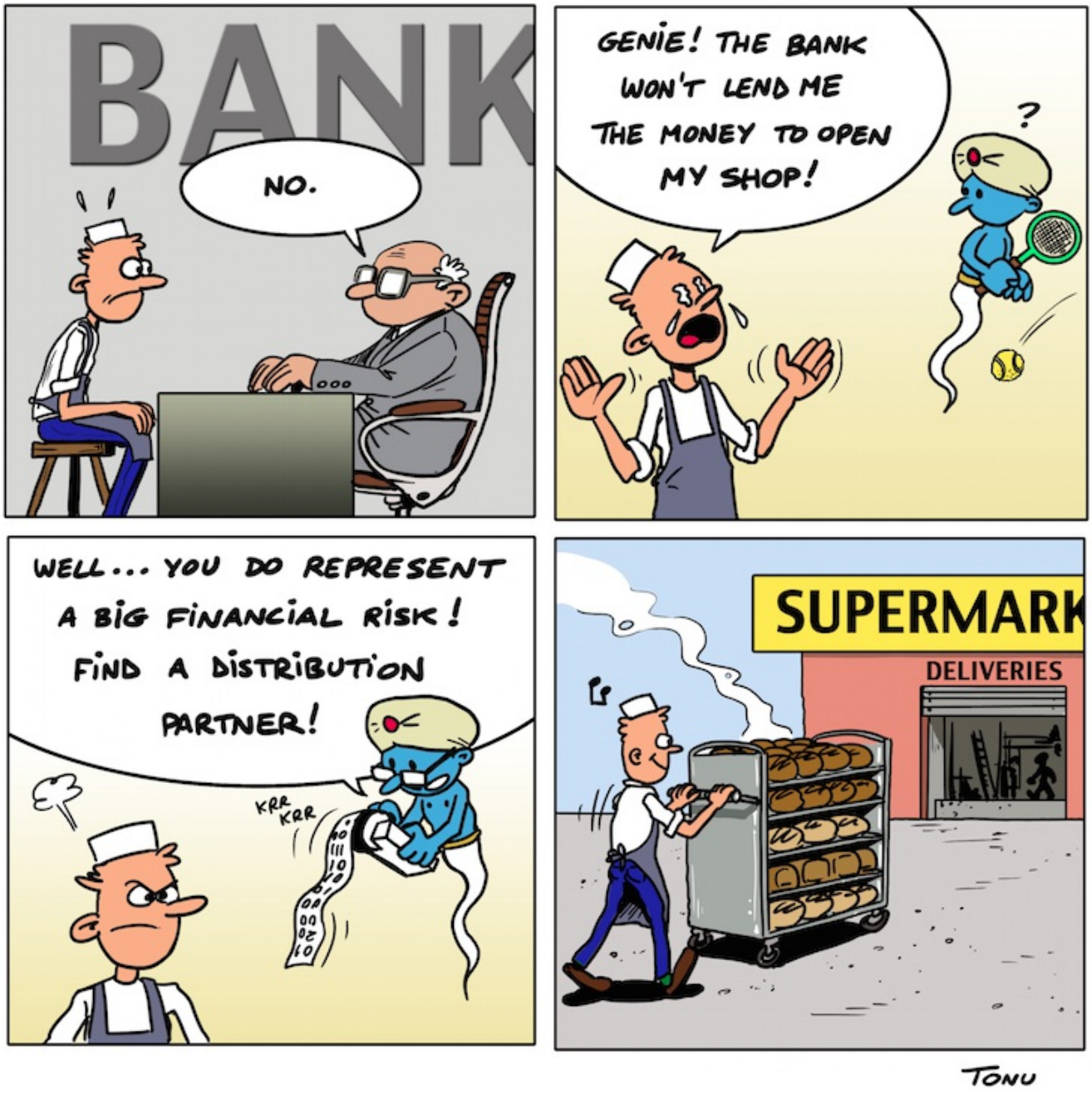
In other words, creation requires 90% work and 10% talent; the work taking place before and after innovation.

## 2. Continuous innovation or disruptive innovation

Improvement can be a succession of minor breakthroughs (see the German automobile industry) or disruptive (Google versus the encyclopedia). Continuous innovation is more the distinctive feature of large Enterprises, whereas disruptive innovation is suited to Enterprise start-ups. If a large Enterprise wants to invest in disruptive innovation, it is recommended that it isolate the team that has to create the new Model in an isolated structure: this is "**intrapreneurship**".

The strategy of some large companies consists of acquiring small innovative companies because they recognize they are not able to innovate: not because they lack smart brains, but because their governance and procedures do not favor innovation.

## Adopt the right attitude: tenacity, inventiveness and economy




# 1. What qualities do entrepreneurs have?

Managers need to have certain qualities that enable them to carry out their role:

- Information level: active observer, circulating information, spokesperson
- Inter-personal contact level: as both leader and liaison officer
- Decision-action level: as an entrepreneur, allocate resources, manage disruptions and negotiate

All these roles have been clearly defined in Henri Mintzberg's work.

That said, there is an important difference between **Operational managers** who apply an existing Model and **Transformation managers** who invent the new Models: an Entrepreneur is,

first and foremost, a Transformation manager.

Entrepreneurship, whether it is within one's own Enterprise or starting up an enterprise requires knowing how to live with **uncertainty**.

Starting an Enterprise is an **obstacle course**: the Entrepreneur will be faced with a host of unexpected situations in which he or she will have to make quick decisions without having all the facts.

## 1.1 Entrepreneurs are creative

The main quality is creativity: entrepreneurs need to be able to find a competitive advantage, that is to define an **original** Offer, or to provide the same Offer as the others do at a **lower cost**, which then requires creating an **original** Operation Model. Not only do they have to be able to, they must also strongly believe they can.

## 1.2 They know how to question their objectives to adapt them to the realities

That said, experience show that the first steps often lead to adapting the Offer in function of the reactions of the first prospects or customers: being convinced does not mean being obtuse. We have to keep the backbone, the principles that we do not want to depart from, and adapt the Offer taking into account the customers' reactions.

## 1.3 They accept having to take risks

Entrepreneurs cannot apply the principle of precaution: it is impossible to guarantee that an enterprise start-up will succeed. We can even say that failures are more common than successes. They must therefore know how to take initiatives, be self-confident, autonomous and not give up at the first hurdle.

We grow with each happy or unhappy experience if we know how to learn lessons from what we have been through. Taking risks is rewarded as, even if we fail, we come out stronger for the next time (discussed in part in The Lean Startup by Eric Ries; it is also the subject of "the learning organization" by Peter Senge).

## 1.4 They have an enormous amount of energy and tenacity

Starting up an Enterprise requires a huge amount of energy in very diverse domains: designing the offer, organizing its distribution, sales activities, management, seeking financial resources, managing staff, legal and accounting aspects...
The challenges:

- Sales: how do we convince reticent prospects to trust a start-up?
- Team: how do we identify talents and get them to accept this adventure when we cannot offer high salaries?
- Management: how do we control spending even when the first successes are there?
- Finance: how do we find sources of funding even though we do not inspire confidence?
- Legal: how do we build contracts that protect a budding Enterprise?

Entrepreneurs have very diverse profiles, but they all have an above-average **energy potential** to progressively solve these different challenges.

## 1.5 They have a permanent will to learn

The worst danger that lies in wait for Entrepreneurs is **self-satisfaction**: many Entrepreneurs let the initial success go to their heads and did not know how to monitor changes in the Market. Entrepreneurs have to be permanently curious: they have to seek to understand what the others are doing well, to use it to their advantage in their own development.

# 2. What attitude does the Transformation team have?

The Transformation team should be in harmony with the attitude of its manager.

- It accepts **uncertainty**
- It knows how to **work hard**, without counting the hours, in crucial periods such as new Product launches.
- It is very **supportive**: it is important to help team members in difficulty.
- It seeks to develop its **Competence** and not to obtain a hierarchical position.
- It **celebrates** every success!

# An Enterprise needs Resources


An Enterprise in action can be summarized as "Resources which execute Models". The essential Resources are:

- Actors
- Information
- Financial resources
- Components
- Premises and equipment

## 1. Actors execute Actions

To Operate, that is to say to Produce, Distribute Goods, Information and Services, the Enterprise uses Resources. The main Resources are the Actors who will Produce, Distribute, manage or

control and, in the first place **Human-Actors** who execute the Operational Processes. They are above all, the **employees** of the Enterprise or temporary workers, consultants who work inside the company.

But, in an ever-increasing way, **Customers** also actively participate in the Distribution Processes by ordering direct certain products, as do **Partners**, who collaborate inside end-to-end Processes provided by the Enterprise.

However, there is a second category of Actors: the **IT-Actors** which know how to act if we program their behavior in advance: server, PC, tablet, smartphones, and everything that is programmable are also Actors because they are capable of executing Actions while following a Model.

## 2. We execute Actions thanks to Information

In order to act, the enterprise bases itself on **Information** that it collects over time **internally** or that it purchases **externally**. The Information concerns:

- Its Market
- Its Products
- Its prospects or Customers
- Its Contracts
- Its Resources
- Its activity
- Its accounts
- ...

These are Facts and not Models: Models, whether they are in software or procedure form, are not part of the Resources.

## 3. Financial Resources

As the Enterprise spends **before** generating revenue, financial Resources are needed to purchase other Resources, train one's Actors, build and deploy the new Model, Produce and Distribute and wait for the Customers to pay.

## 4. Components

To manufacture Goods, we rely on material components, even raw materials, which simplify the Enterprise Production and enable it to focus on what it is best at. These products purchased externally become Enterprise Resources.

To build Models, we also take advantage of intangible components.

## 5. Premises and equipment

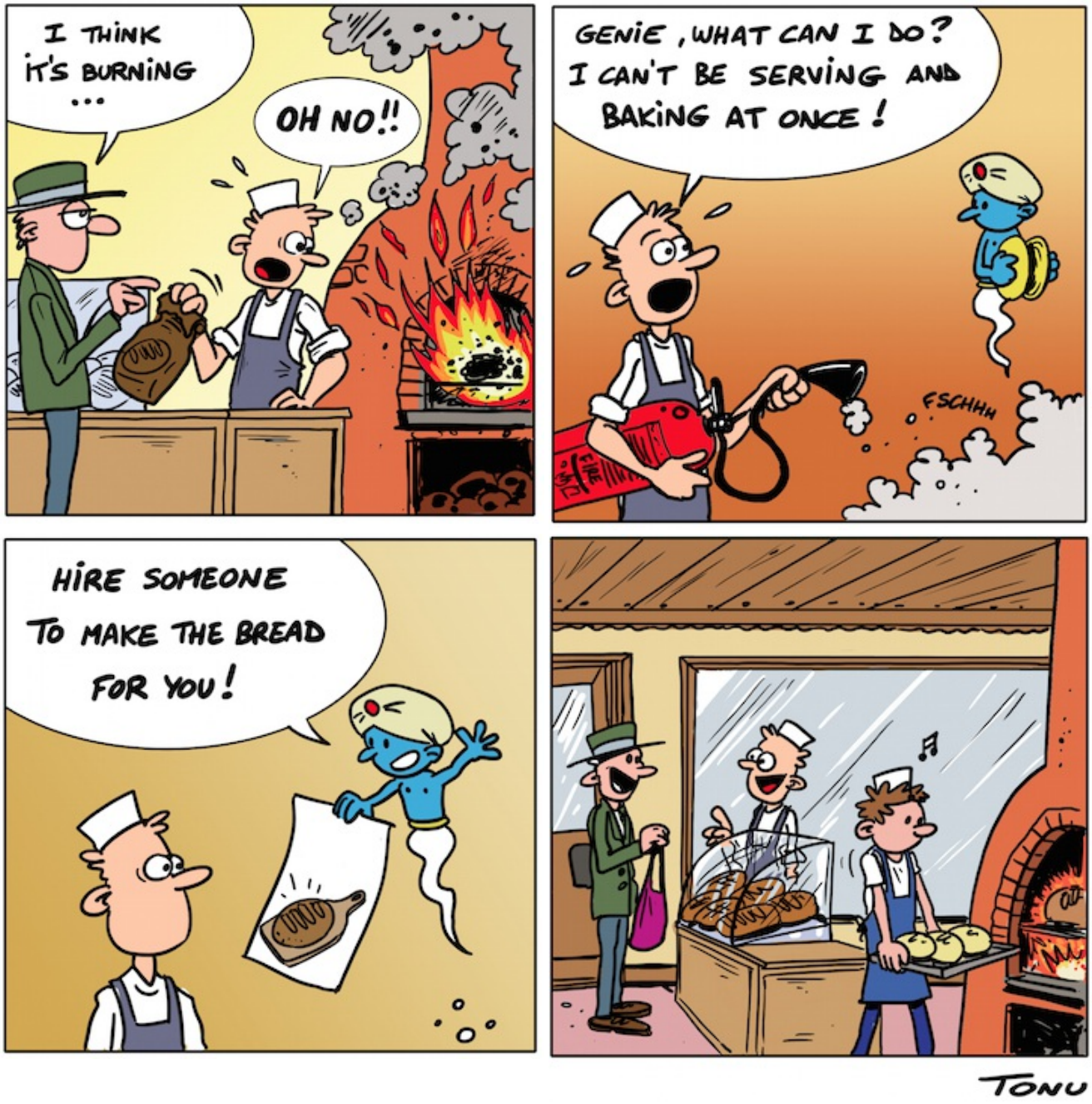
Actors need a place from where to act, whether it be in the headquarters, administrative buildings, design and engineering companies, factories or warehouses.
Mobiles, using file-sharing in the Cloud and collaborative solutions enable us, today, to conduct an increasing part of the activity outside the Enterprise. Nevertheless, Actors need to meet up from time to time to live together and create human relations that are essential for team work. But premises are no longer used in the same way: the gradual disappearance of paper and the digitization of information enables workstations to become commonplace and shared spaces, smaller than in the past, are used dynamically.

## 6. Resources change with the life of the Enterprise

At any given time, Resources are limited by the financial means of the Enterprise. The challenge is in finding the right compromises to optimize the usage of the financial budget:

- Should we increase salaries to motivate employees or recruit more employees?
- Should we increase training efforts or give more free time?
- Should we choose prestigious premises to build the Image of the Enterprise or grow R&D?
- Should we buy market information or progressively collect this information internally?
- ...

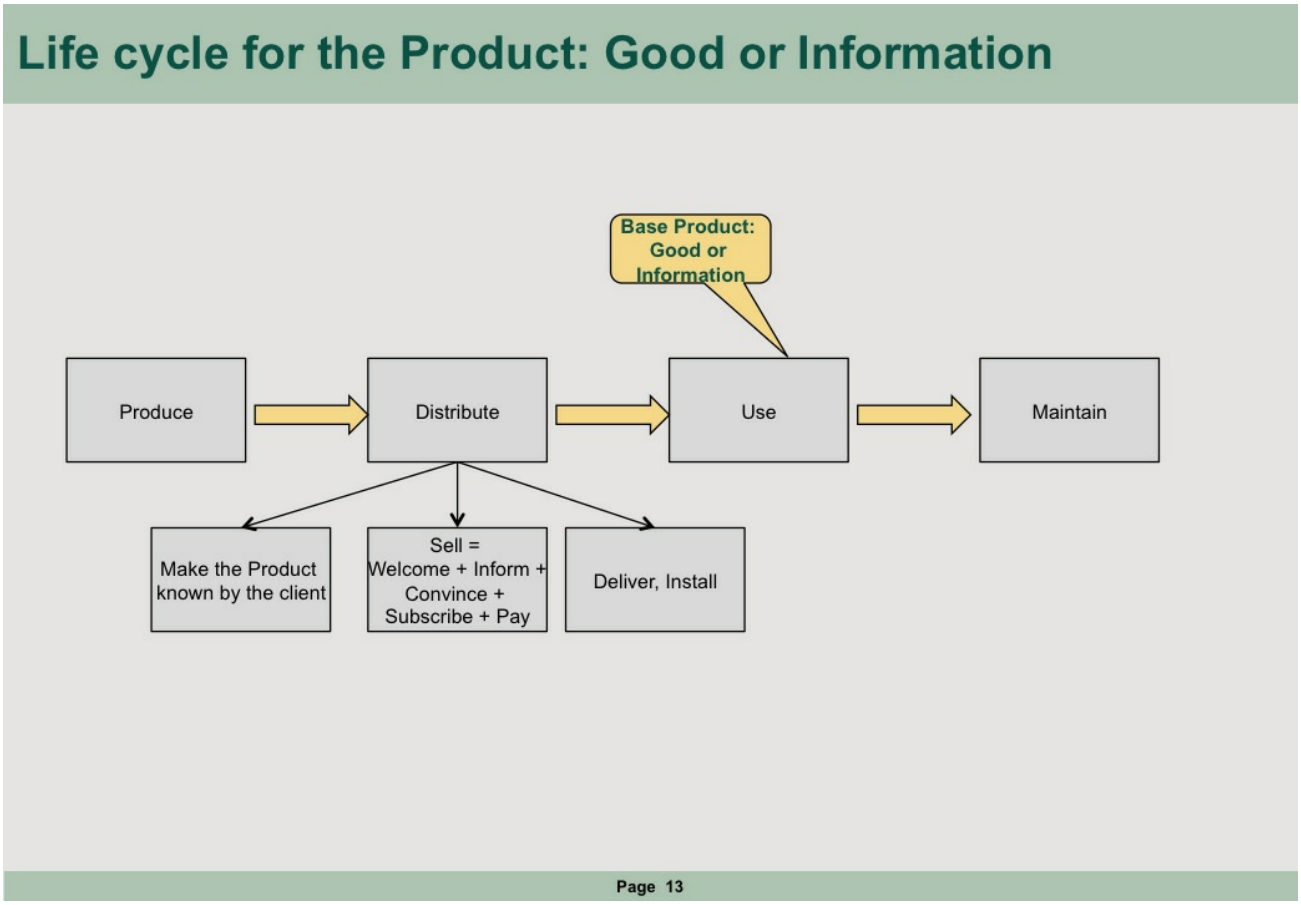
# Organize the Operations




## 1. The life cycle

The Product life cycle is made up of several stages: Design and make changes to the Model, Produce, Distribute, Use and Maintain.
The design and changes to the Model are part of the "Transformation" as we will see later, whereas the other 4 stages are part of the "Operations": we will first concentrate at the Operations.

## 2. The Product life cycle: Goods and Information




## 2.1 Produce

- Acquire production **tools**
- Use **Component-Products** purchased from other Enterprises
- Produce Goods or Information made available for Distribution

## 2.2 Distribute

- Promote the Product and the Distributor
- Sell: welcome, convince, manage contracts, invoice, recover debts
- Make available: deliver, install

Downloading a book without having to go to a shop, whenever you want to, can be a positive Distribution Value for some, whereas others will prefer the Value of being able to flick through the pages in a bookshop and ask for advice.

## 2.3 Use the Product

**Basic Value**
Using the Product enables the Customer to benefit from the basic Value.
As it happens, using a book generates the "pleasure of reading" Value.

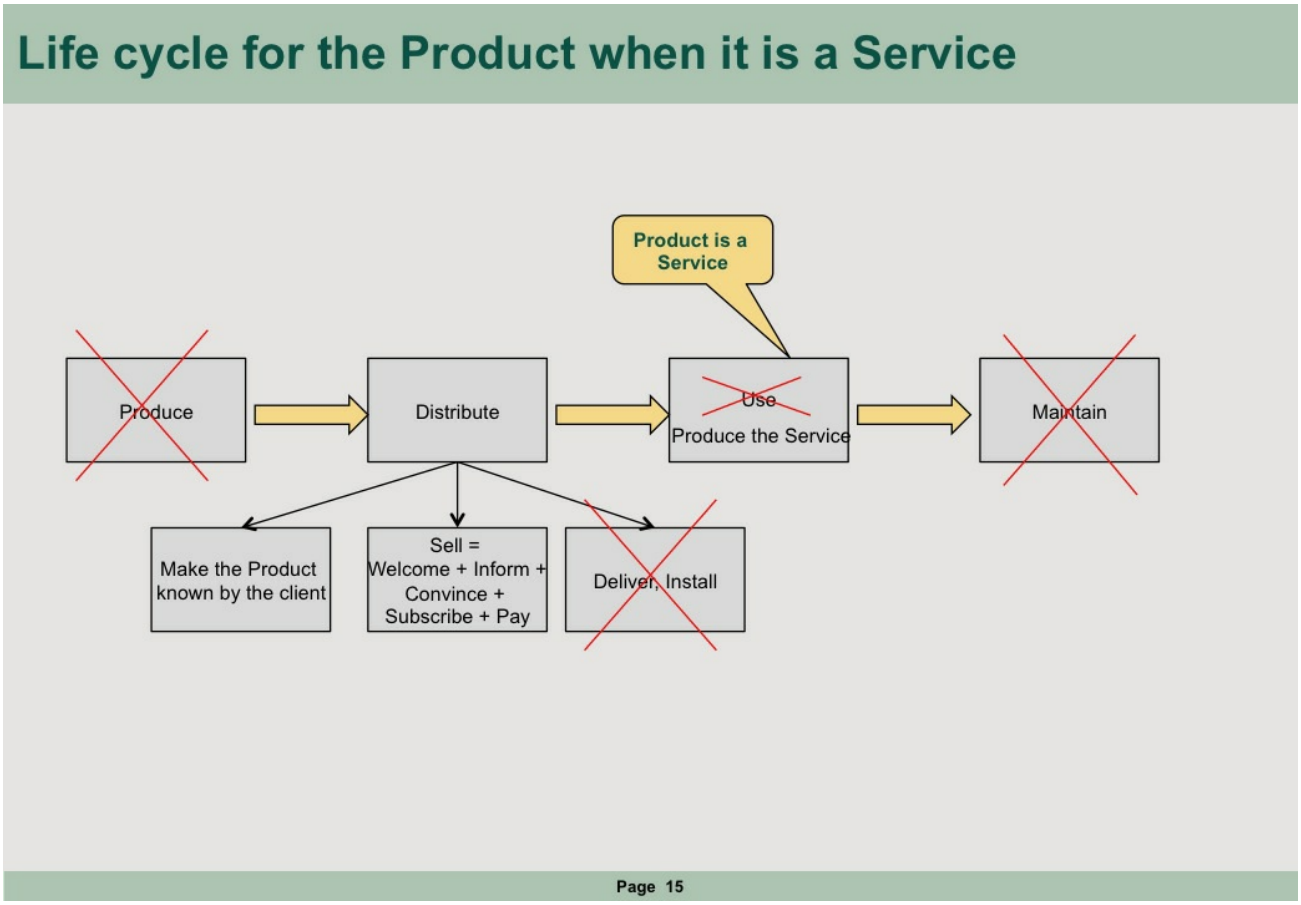
**Usability Value**
What is the difference between buying a paper book or downloading the same book in an electronic format?
The book's content (Information) is the same (same basic Value), but the **usability** is not the same. Certain people prefer a paper format out of habit, or for the smell of the paper, or for the feel of the cover, or to make notes on the pages.
Others will prefer the electronic version to be able to easily carry their library around with them, to pay less, or to increase the font size.

### 2.4 Maintain

Goods may need, at a later date, to be updated, repaired, controlled and even disposed of... that we have grouped together under the term Development.

# 3. Service life cycle

The service life cycle is simpler because it is not storable.




- Distribute: Promote the Service and Sell it
- Execute the Service

# 4. Manage the Resources and management

Production and Distribution Processes are the main Processes of the Enterprise.
But any Enterprise also needs to execute two other families of Processes to manage its Resources and steer the Enterprise.

### 4.1 Manage the resources

Resources are necessary to execute the previous Processes: employees, partners, IT, finance, premises...
Operational Processes are needed to manage these Resources.

- Management of human resources
- Management of partners, such as distributors or suppliers
- Management of IT resources
- Management of financial resources
- Management of premises resources

## 4.2 Management

**Managing** the enterprise includes defining the operational objectives and their result: follow production and distribution levels, development of resources, accounts...
It also includes providing legal and statutory information (such as Financial Accountancy).

## To grow, formalize the know-how in an Operation Model


# 1. Formalize know-how into a Model to grow

When the workings of the Enterprise become complex, **its Operations** must be **Modeled**, without which the Human-Actors will work in a muddle. Those who know how to do something must take the time to properly describe how it is done for the benefit of the others. Another consequence is that Modeling also enables us to improve our way of doing things as it highlights any complexity or inconsistencies.

A **Model** is a simplified representation of a real system to better apprehend it: the "system" can be a Product or the Enterprise Operations. We can thus define a Model for a Good (e.g., car Model), an Information Model, a Service Model, an Offer Model or an Operation Model (e.g., "How to Produce" or "How to Distribute"). The Model enables us to understand, memorize, communicate

and train those who will execute it. To take an example, we can more easily execute a recipe (Action Model) if the quantities of the ingredients (Information Model) and the cooking time are accurate.

Thanks to this Model, we will be able to train and guide the other Actors to increase know-how. One of the difficulties is that those who know how to do something, do not always know how to Model this knowledge properly: we have to help them by making Modeling specialists available.

# 2. Model Actors, Actions and Information

For the Operations, as **Actors** execute **Actions** with **Information**, we therefore have to formalize:

- the **Human-Actor Model** formalizes the Role (Seller, Producer, Administrator): what are their rights and responsibilities?
- the **Action Model**, what we often call
  - **Process** Model (like "Sell", "Produce", "Manage")
  - **Function** Model that makes up the Processes (like "Fix Price", "Print").
- the **Information Model** which formalizes the Objects (Customer, Product, Contract, Account)..., their identification, their relations, their attributes and their types

Thanks to these 3 levels of Modeling, we are able to describe, very precisely, the right way in which the Actors should act.

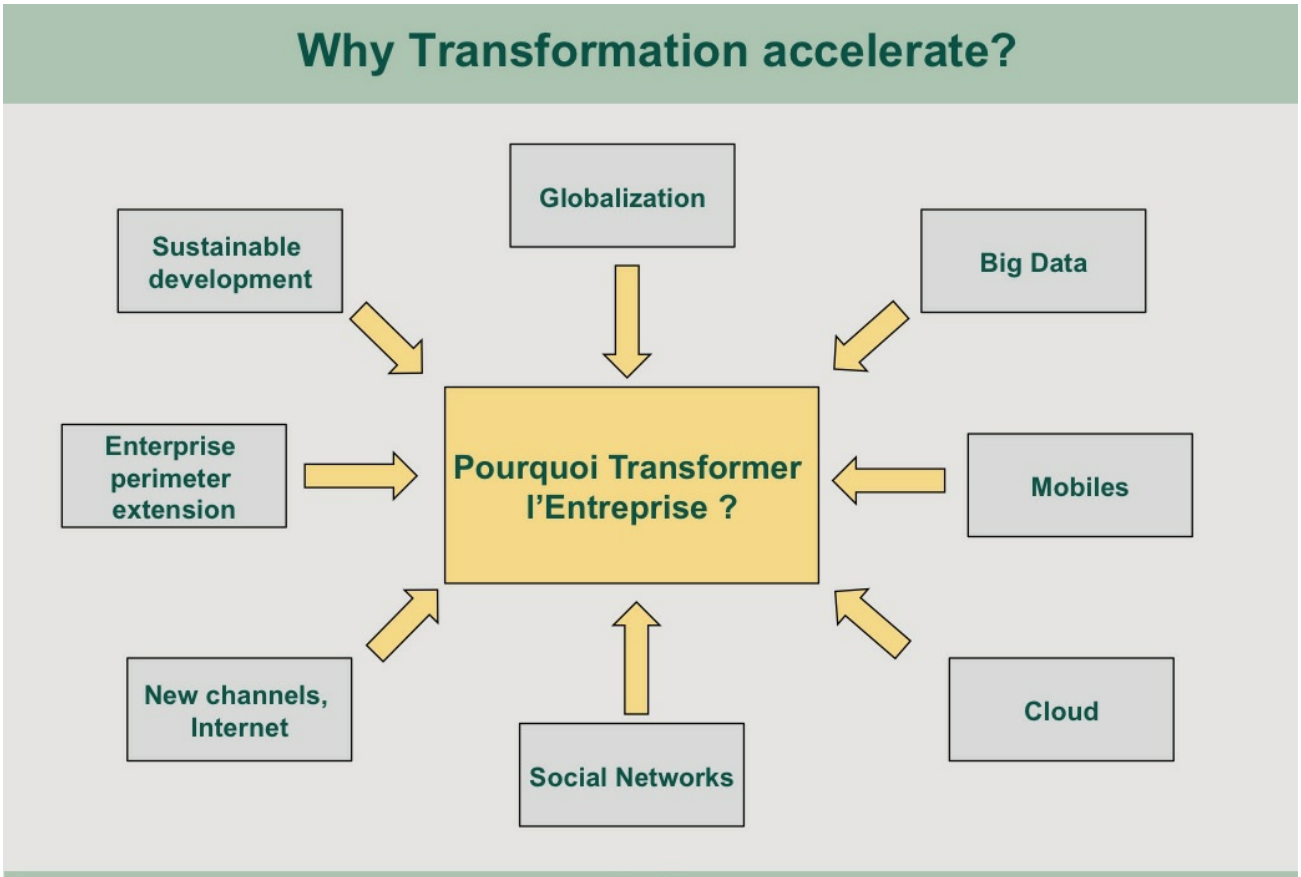
# 3. Global Modeling and detailed Modeling

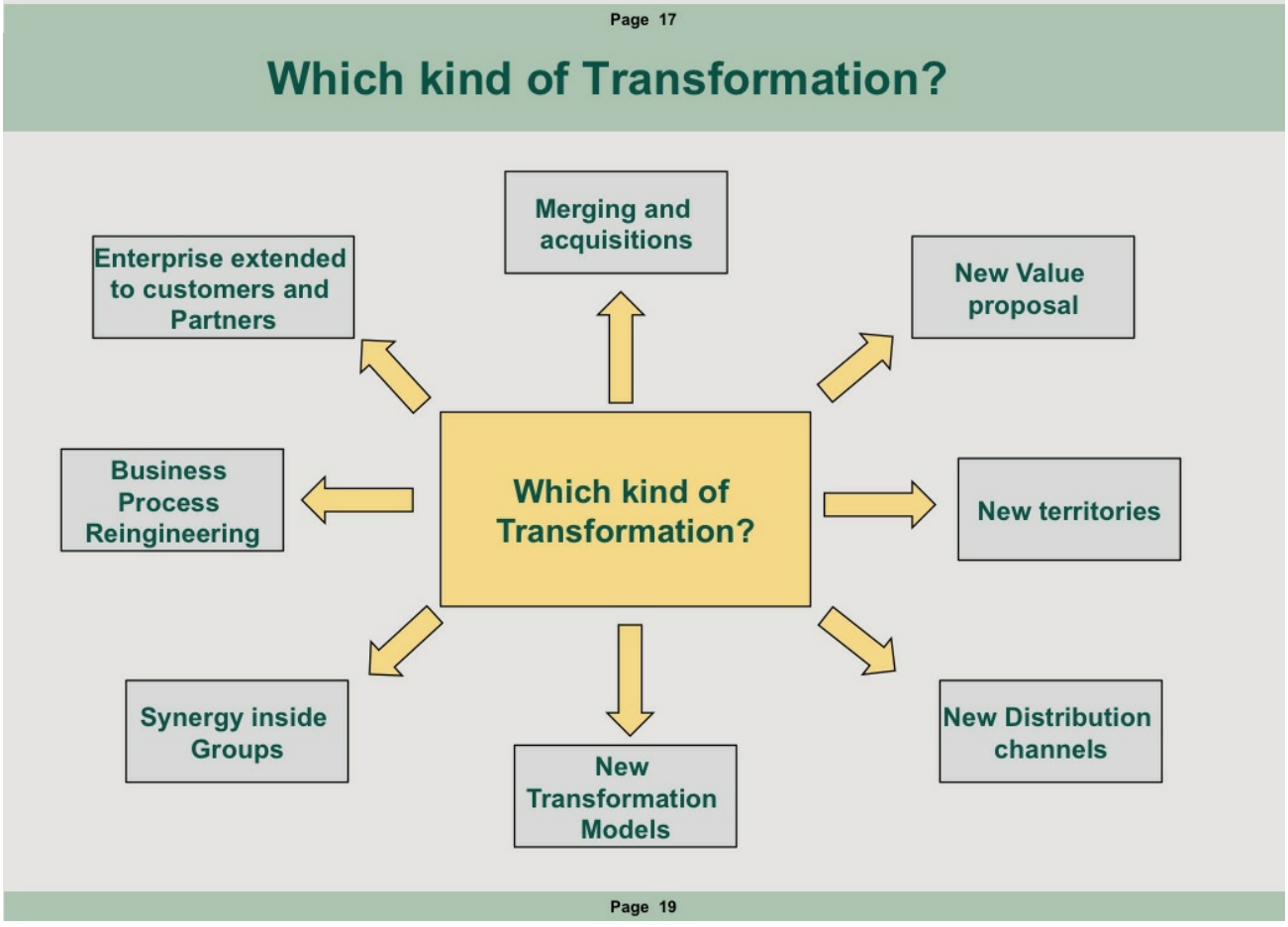
Modeling can be **global** via Maps (business Object maps, Process maps, Function maps,...) or **detailed**.
Global Modeling is essential when the Enterprise has become too complex: We must create a framework in which the various detailed Models will take place if we wish to avoid a patchwork of heterogeneous Solutions, which have difficulties in communicating between each other (see 4.1).

## Alongside the Operations is Transformation


# 1. What does "Transform" mean?

While the Product Models or Operation Models were stable, Transformation was a marginal activity. In a stable mode, we would improve the Product features and internal Processes incrementally.
The main part of the activity was therefore to Operate (Produce and Distribute) according to the established Model. Today, the rhythm of change is such that the Enterprise must permanently adapt its Models and create new ones: we are entering the Transformation domain.




**Transform means that we acquire a new Model and Deploy it.**
"Acquire a new Model" means either acquiring a pre-built Model (for example, buying a software package or a license for a new procedure), building the new Model ourselves, or modifying the

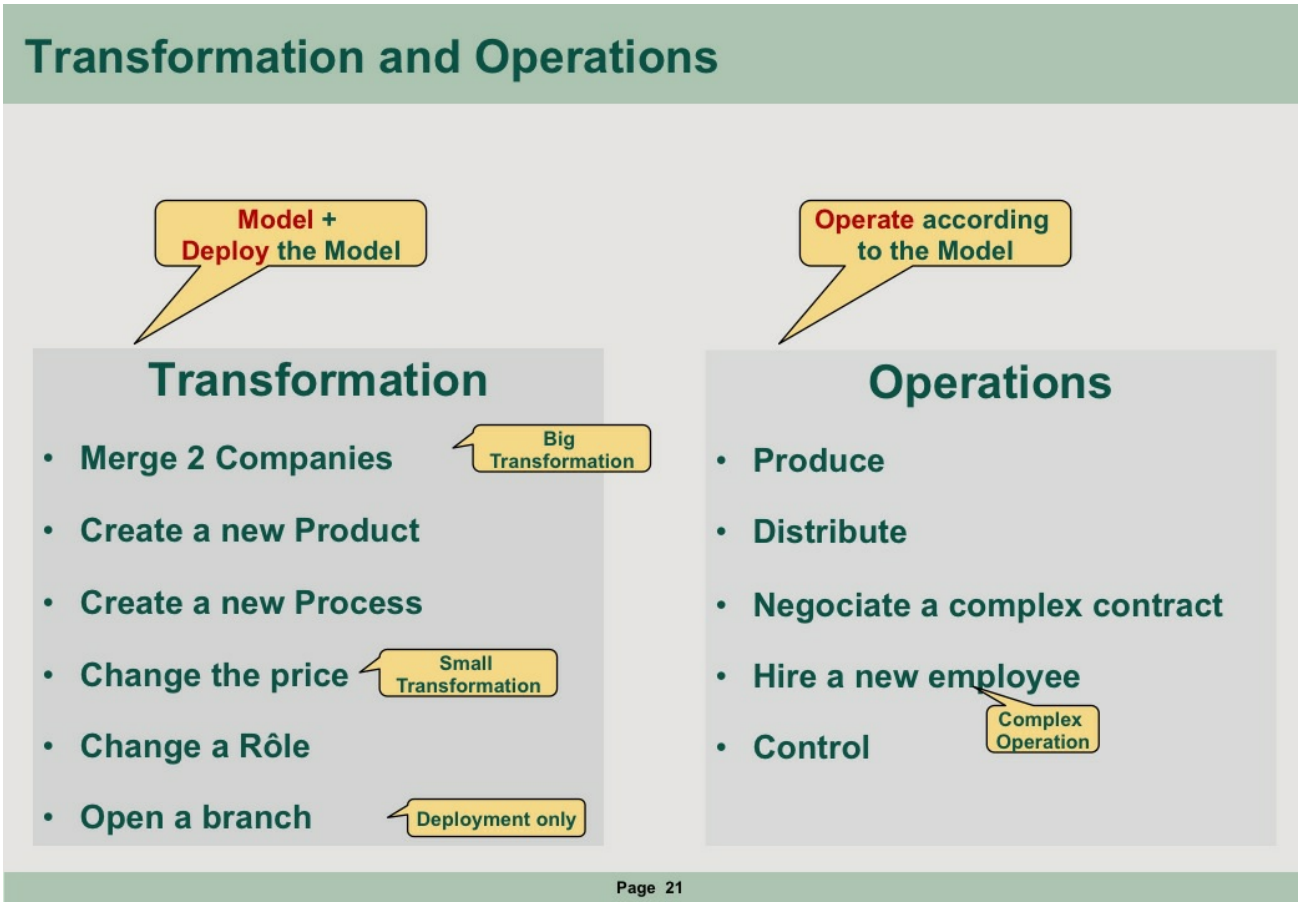
existing Model.

The new Model can be:

- A **Product Model** or an **Offer Model** which combines several Products: this is the innovation of the Offer (more aesthetic, more efficient, less expensive...)
- An **Operation Model**: this is the innovation of the Operations (Process optimization, productivity or quality gains...)

There are therefore only 2 jobs in the Enterprise:

- **Operate** according to the existing Model with the Operational Resources: this is what enables the Enterprise to live in the present
- **Transform**, that is to say, prepare the future by building new Models and by deploying them




## 2. Transformation Processes

Transformation Processes are very different from Operation Processes. It is no longer a question of Distributing or Producing, but of managing Projects, defining a Road map, changing a price, modifying a Process, opening a new branch or a new shop, driving an Architecture project, deploying a new Solution, maintaining Solutions... We often use the term methodology to name the Model of Transformation Processes.

**Deployment** consists in adapting the Operational Resources to a new Model: reorganizing, allocating Human-Actors to new Organizational units, training, adapting premises, installing IT infrastructure, migrating information from one Model to another...

Remarks:

- a **new deployment without change of Model** such as opening a new shop or a new branch is nonetheless a Transformation, limited to the deployment phase. But, it is less risky as the Model has already been tested.

- **Small** transformations exist (changing a price) which often take place through configuration (see the scene "The Model must be Modifiable by the Business") as do **large** transformations (merging 2 enterprises). We will focus mainly on large Transformations which pose the most problems.

# 3. Carry out a Transformation assessment to convince others that we must deal with it

The areas for improvement in the Operations like optimizing the Supply Chain, the Sales or Back-Office Processes, are well identified by Executive Management. Means are budgeted to improving Operational Processes.

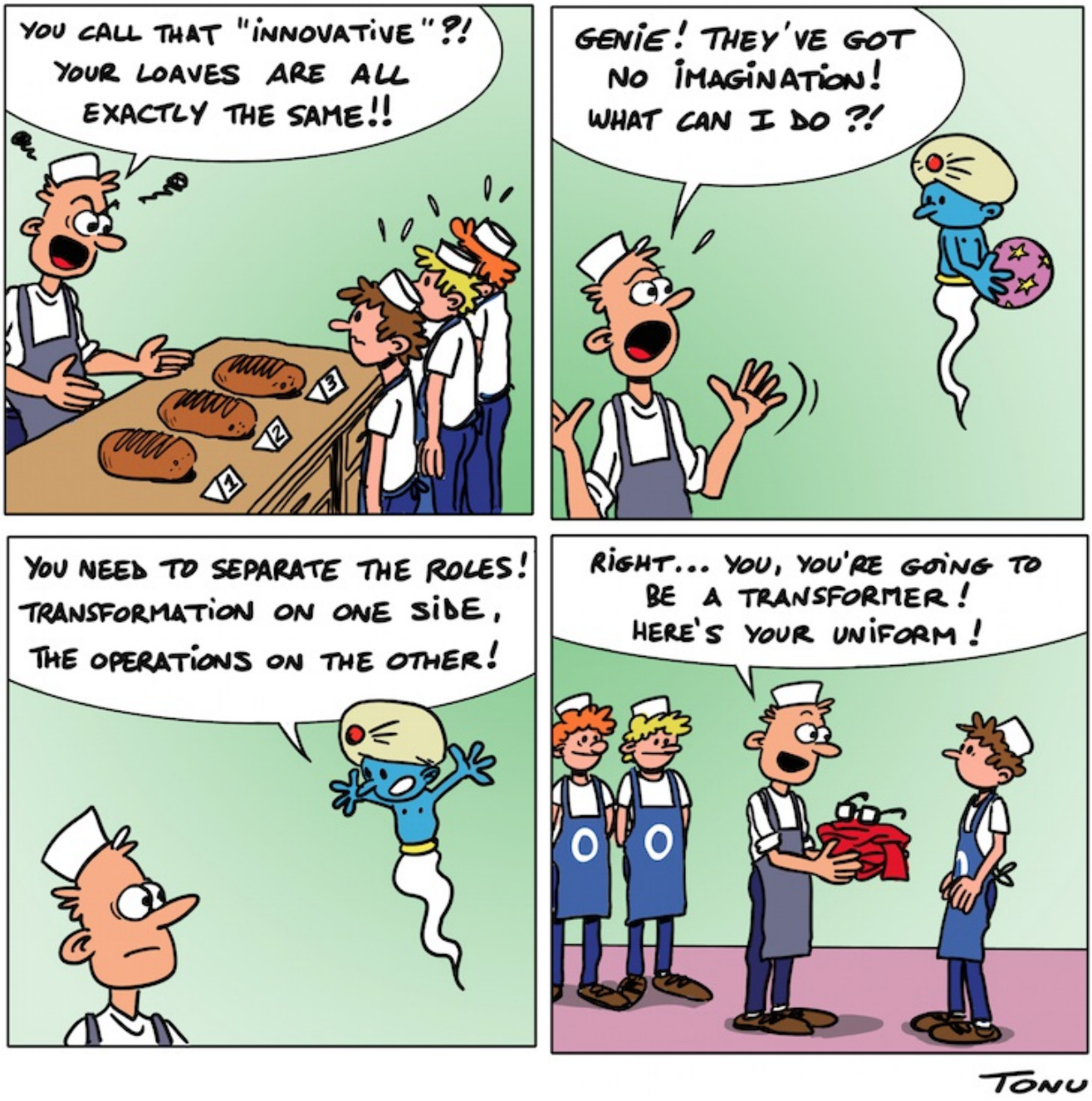
On the contrary, Executive Management only allocates, in general, a few resources to improving Transformation Processes. On the one hand because it is rarely aware of the overall cost of the Transformation; on the other hand because it does not believe that it is possible to progress strongly in this domain: the projects carry risks and there is not a lot we can do about it.

It is therefore necessary to carry out:

- An **assessment** of what the Enterprise Transformation **costs**: not only the IT Development costs, but also the costs for the Business Actors, Transformation Tools, management and related governance, training for operational Actors, dual processing, migration of information...
- A survey on the level of satisfaction regarding the design or Solution modification **timescales**, and on the **quality** of the Solutions.

This survey will automatically highlight that the global Transformation cost and the frustration of the Business regarding the Agility of the Solution is important enough for us to tackle this problem head on.

# Relieve the Transformers of the Operations




Short-term concerns always win over the long term: an operational occurrence in the Operations needs to be dealt with as a **priority** compared with occurrences that may affect the Transformation Projects that are underway. If we want to successfully complete complex Transformations **quickly**, then we must **relieve the Transformers of Operational tasks**. To put it plainly, we must clearly separate the Operations from the Transformation.

IT departments have understood this well as they have traditionally isolated "studies" from the "IT operations".

But this obvious idea is not that easy to implement. Indeed, it is met with reticence by those in the Operations:

- they produce the Enterprise revenues, whereas Transformation is a cost center: they therefore feel justified in managing the actions which use up the profits they have generated

- they know what does not work well and what needs to be improved
- they are close to the field and can come up with realistic Transformations, whereas Transformations hatched in an ivory tower by Transformers cut off from the reality of the Operations can fail
- they feel apprehensive about the level of change acceptable to the Operational Actors

We can leave Operational Actors to lead simple Transformations that only require slight modifications of the Model.
On the other hand, we cannot let them carry out heavy Transformations that require new Models to be created:

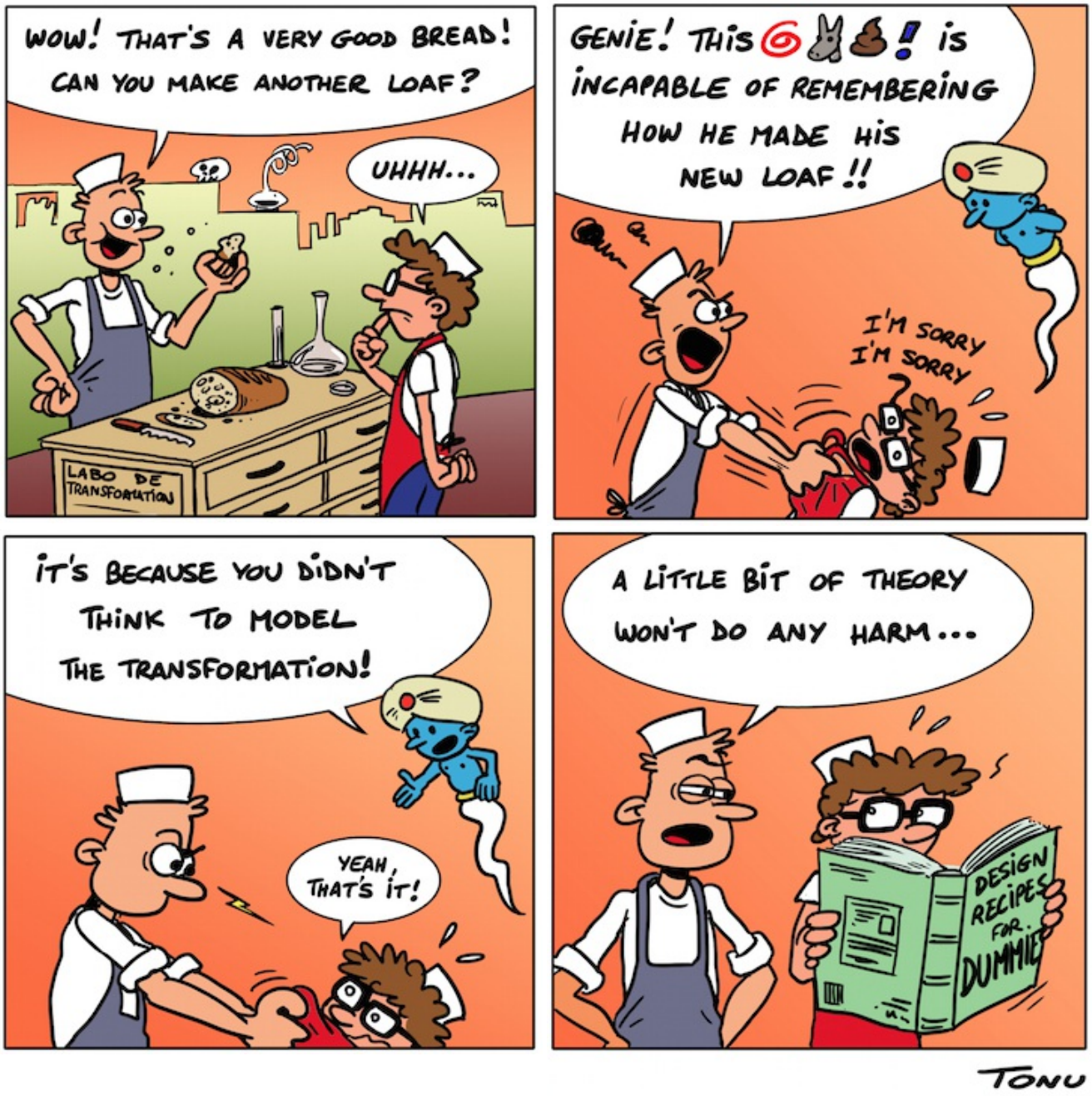
- they will always be absorbed as a priority by the Operations, to the detriment of the Transformation
- they lack, in general, enough time to make an assessment: they know how to optimize their Model, but they rarely know how to question it
- they rarely have the ability to build new Models
- they underestimate their teams' ability to accept change, if it is well accompanied

How do we convince the Operational managers?

- They will not direct the Transformation, but they will participate in it
  - When the Transformation Goal is being defined, their input is invaluable
  - During the formulation of the new Model, their reactions are invaluable
  - When the time comes to Deploy, they are very closely involved
  - And they will provide an assessment of the Transformation when the new Model is fully up and running
- Avoid the tunnel effect and present rapid, concrete results of the Transformation as successive deliverables to increase the credibility of the Transformation
- Do not hesitate to explain that the revenues they generate are only possible because some people have built the Model they use and that others manage their resources.

If one Business Line wishes to favor Agility, it is essential that it **separates** the Transformation responsibilities from the Operational ones.

# Formalize the Transformation Model




## 1. The need for a Transformation Model

Innovation itself certainly calls for imagination and creativity. But, to be effective, it must also base itself on a method, Tools, rigor, a definition of each person's Role, traceability, in short on a "Transformation Model" which describes how to Transform well.

## 2. The Transformation Model is different from the Operation Model

The **Transformation Model** is decomposed, like the Operation Model, into an Actor Model, Action Model and Information Model.

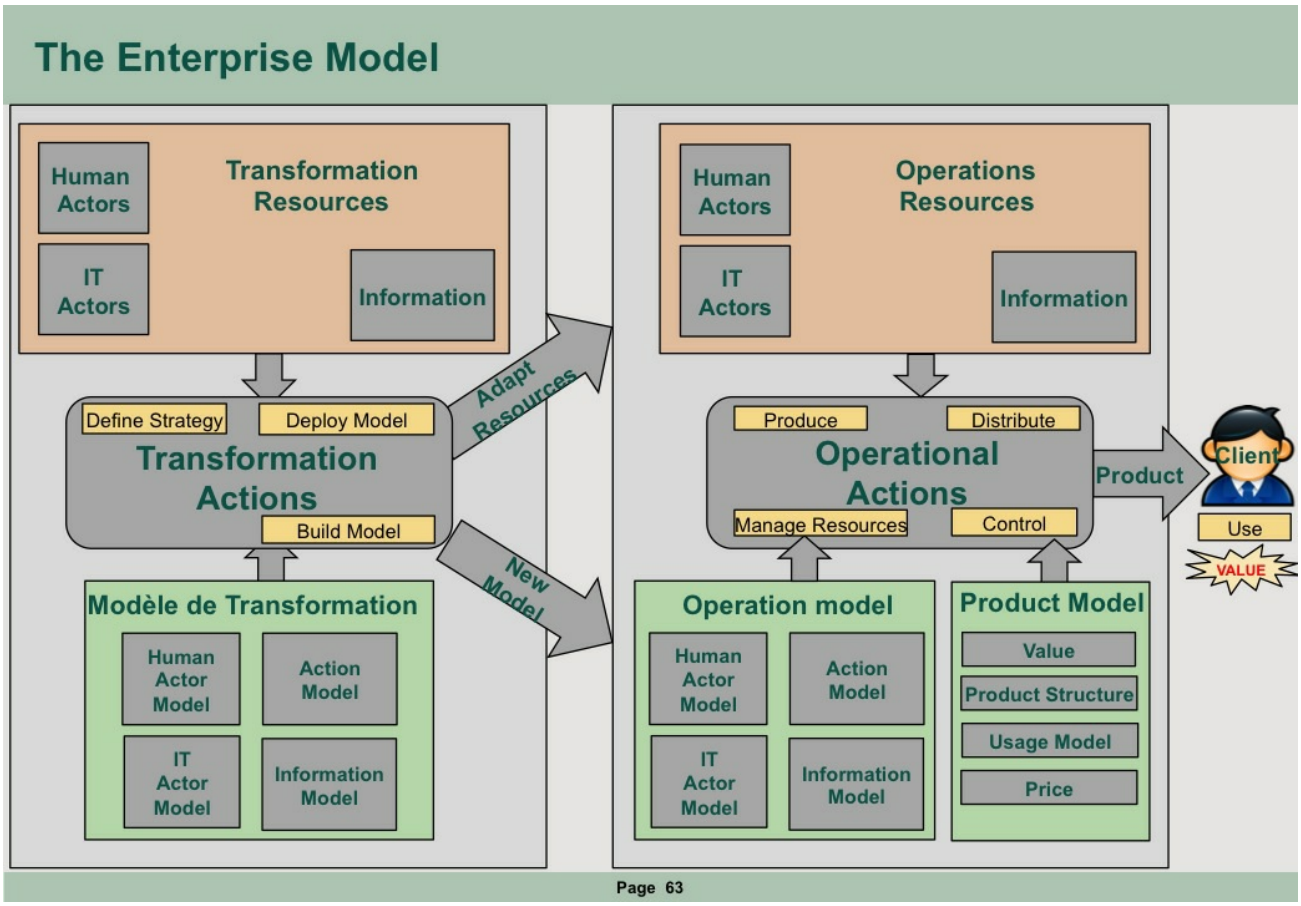
- the **Human-Actor Model** formalizes the Roles of the "Strategist", "Project Manager", "Architect", "Contracting Owner", "trainer",... that is to say all the Human-Actors who contribute to the Transformation.
- the **Action Model** formalizes the **Processes** (e.g., "Build a Solution", "deploy a Solution",

"design a Product"...) or the **Functions** that make up the Processes ("describe a Problem", "Evaluate a project timescale", "Test a Function"). This is what we usually call "methodology" or "Approach".
- the **Information Model** formalizes the Transformation Objects such as Project, Test, Schedule...

But this Model must be adapted to the risky and complex nature of the Transformation.

As much as the Operation Models should be rigorously defined to guarantee efficient and standardized Production and Distribution, the Transformation Models should leave some freedom, the possibility to proceed by trial and error and to iterate, and to reserve a large share linked to testing and training... The difficulty is finding the right balance between Modeling rigor and the freedom to innovate.

# 3. Summary of the Enterprise Model



## Act 2: Grow through geographic expansion

## Geographic growth is the simplest


# 1. Most Enterprises like to grow

We sometimes hear enterprise bosses state that they would like their Enterprise to stay small, to remain a human-scale enterprise: "small is beautiful"!
But this desire to stabilize the enterprise is often an elegant way of admitting one's inability to grow.
For others, it is a desire to not have to manage staff or to reach the social thresholds that impose constraints on the Enterprise. Rare are the Enterprises who actually display a non growth strategy.

Most Enterprises want to grow.

- First, it is recognition of being successful and a source of pride for the employees.
- It is also a guarantee of confidence for the customers.
- Finally, it is a question of critical size.

The Operational Resources are generally proportional to the volume of the Operations, whereas the Transformation Resources are mainly linked to the complexity of the Models. If an Enterprise decides to spend 10% of its Operational revenue on Transformation investments, the strike capacity of the large enterprise will be far more important. A simple example is that of an advertising campaign which aims to build an Image: the impact will be far greater for the large enterprise.

# 2. The simplest growth is one that does not change the Model

The simplest growth is one that does not change the Model and only asks for an increase in Resources. It is therefore not a question of establishing partnerships that will change the Operation Model, or creating a new Product line: we only have to use the existing Model across a larger territory.
Management Processes need to be adapted to territory growth, in particular when they are a substitute for control carried out previously by on-site presence.

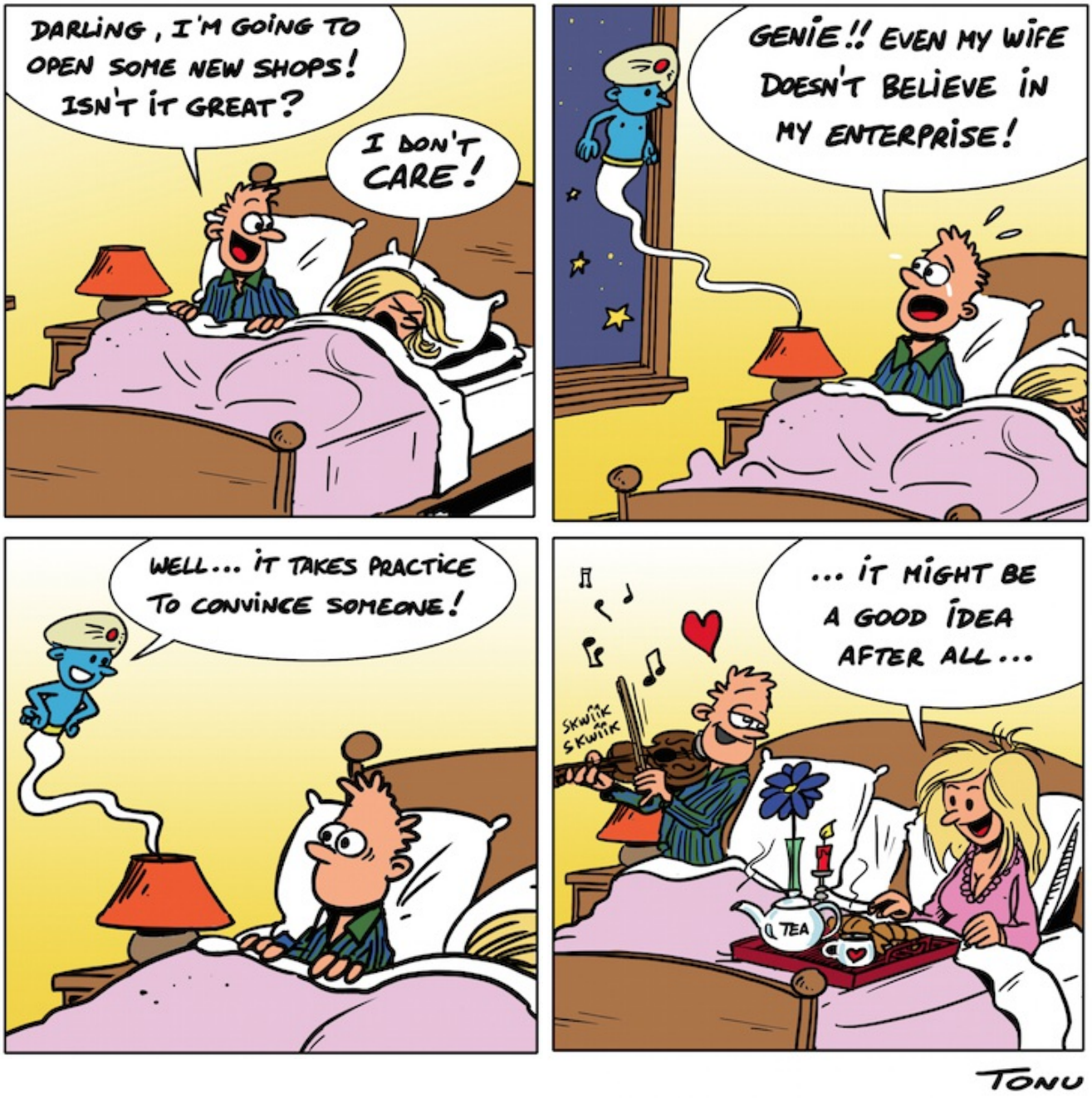
# 3. International Approach

Enterprises nowadays think "Global".
An international approach can be concerned with the Usage, Production or Distribution Processes

- Good ideas circulate quickly and are rapidly copied, transport costs less and less (containers, increasingly efficient engines, the best optimization of transport thanks to information systems): there is a fundamental trend towards the "globalization" of Products. Products are less and less dependent on the locality and make geographic expansion increasingly easy. If the Product can be transported, the Customer can **Use it** in a different country to the one it was bought in: the product just has to be adaptable to local standards (electrical plugs for example).
- We can **Produce** in different countries: if the Product Models are the same, the production Processes are also the same. We can dynamically split the Production between the different Organizational Units in order to optimize the use of our resources.
- We can **Distribute** in different countries: the Product Models sometimes have to be adapted to take regulatory and cultural specificities into account (e.g., building products). The digital revolution and the possibility to sell direct on line, and therefore to free ourselves from physical distribution networks, have drastically changed the way we regard geography and territory. What counts now is the logistics around stocks and flows (e.g., Amazon). For fresh produce, like bread, the context is different.

Remark: in all cases, the local Resources management Processes may need to be adapted, especially Human Resources management.

## Learn to convince




# 1. We need to permanently convince people

The ability to convince is fundamental: it is not enough to be right, we need the support of others. Some people are naturally more gifted than others when it comes to convincing someone. But we can improve through training, something that business schools have been doing for ever.

There are two main approaches:

- **"Manipulative" Negotiation**: we use every trick and manipulation to end up succeeding. For example, we use the "recurrent yes" method (to make you say yes several times on obvious or minor points to then obtain yes on something important).
- **Reasoned Negotiation**: we apply a certain ethic and seek an agreement that optimizes the sum of both parties' values. This approach leads to a more sustainable agreement

because both parties accept the basics of it. It is not an agreement obtained through cunning that one of the parties may later feel frustrated about.

We suggest developing the reasoned negotiation by **Harvard (Fisher and Ury)**.

# 2. How do we convince someone using reasoned negotiation?

The first remark is that we are more likely to convince someone if we believe in it ourselves: if we have **strong beliefs**, they will naturally show through and are the best asset for getting someone to adhere. The Vision does not only call upon reason, it can also arouse passion or emotion.

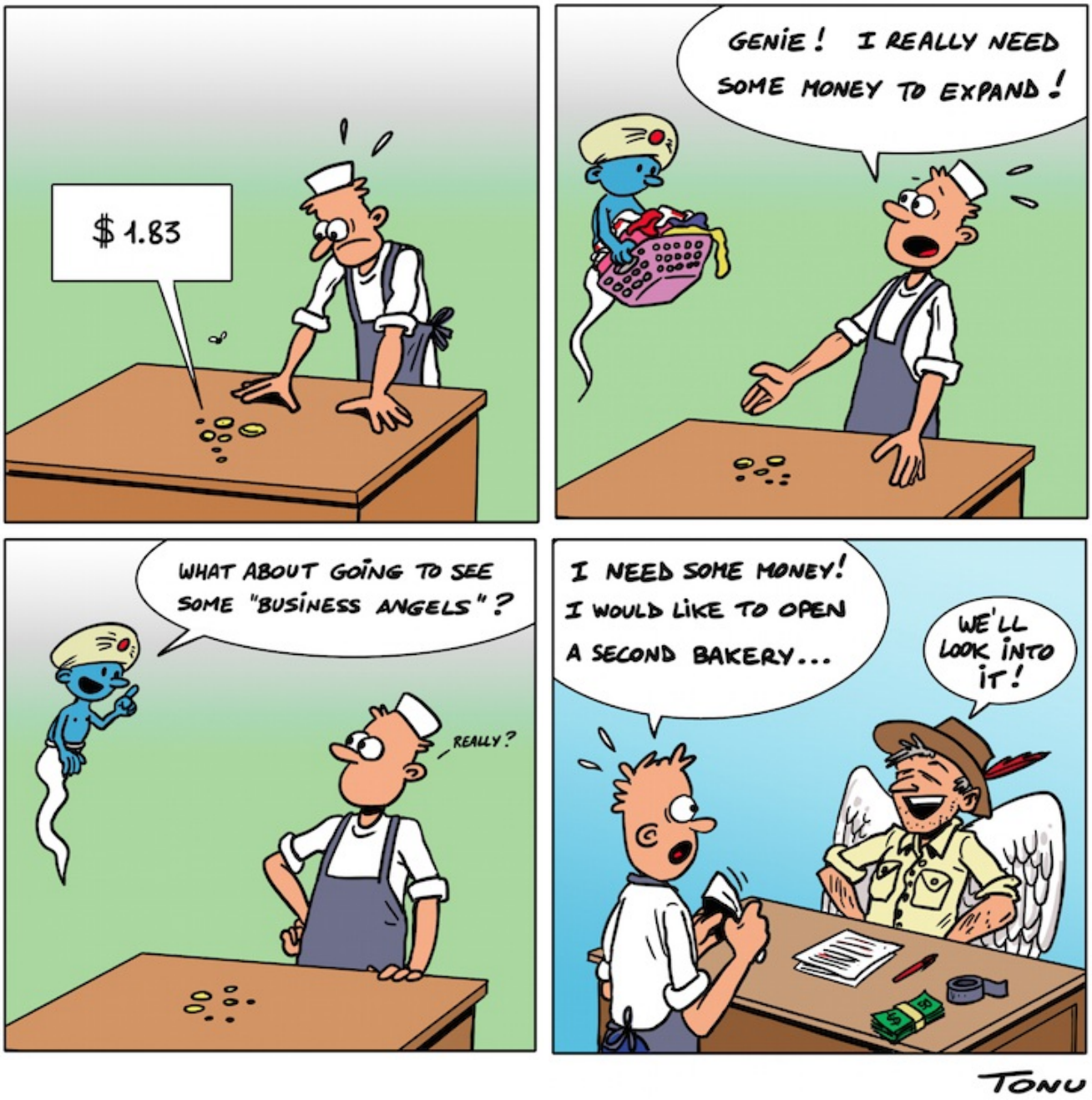
The methods revolve around the same principles:

- Identify **who** needs convincing
- Everything starts from the **motivations** of the person to be convinced: adapt what you say and build your arguments based on these motivations.
- Do not over-argue: look for the main arguments and **forget about the others**
- Keep it **simple**
- A standard presentation plan when looking to convince someone:
-
    - Look for the underlying motivations of each party: what Value are they looking for, how can they be assessed according to importance?
    - Validate the **motivations** of those that we would like to convince
    - Explain the **target** that answers these motivations: show proof
    - Show the **pathway** to get there
    - **Summarize** the advantages
    - Try to get an **immediate agreement** or, at worst, fix a next step
    - Be **gentle with the person, be tough on the ideas**

One rule is surprising; it is **easier** to create a value differential if there are **diverging interests** than if the interests are identical. We therefore build options to satisfy the **main interests**, financing them through **concessions on the less important interests**.

We also need to define the BATNA (Best Alternative To Negotiate Agreement): what should we do if agreement fails?
An agreement is theoretically satisfactory if it passes above each BATNA. It is of course vital to discover the BATNA of the other person because we have to offer them more to get his/her decision. The BATNA must be kept secret and never divulged, as, if it is bad, you will have no more chances to gain satisfaction and if it is very good, it can lead to a break or an official attack to devalue it and cause you to have doubts!

# Finance the start-up through business angels




## 1. The Business Angels are the first investors

**Transformation** requires an investment expense that will only bear fruit several years down the line.
Before benefiting from the results, the Model must be built, tested, prototyped, the Enterprise must have premises or legal advice and the initial Resources must be in place. The Enterprise is looking for **start-up financing**.

Starting an enterprise therefore requires a risky investment. The Shareholder is the one who invests and takes this risk because he/she thinks that the project can succeed. The Shareholder's objective is that the enterprise succeeds in turning a profit over the long term.

We generally start with "**love money**": close family and friends give small sums of money to the Entrepreneur to encourage him/her. But it is generally insufficient: we have to turn to "**Business**

**Angels**". They are individuals, often former entrepreneurs who have sold their enterprise, who know how to take important risks, who are really taken with a project and who finance start-ups, using their personal fortune, at the most uncertain time. It is extremely difficult to write a credible business plan at the start up of an enterprise.

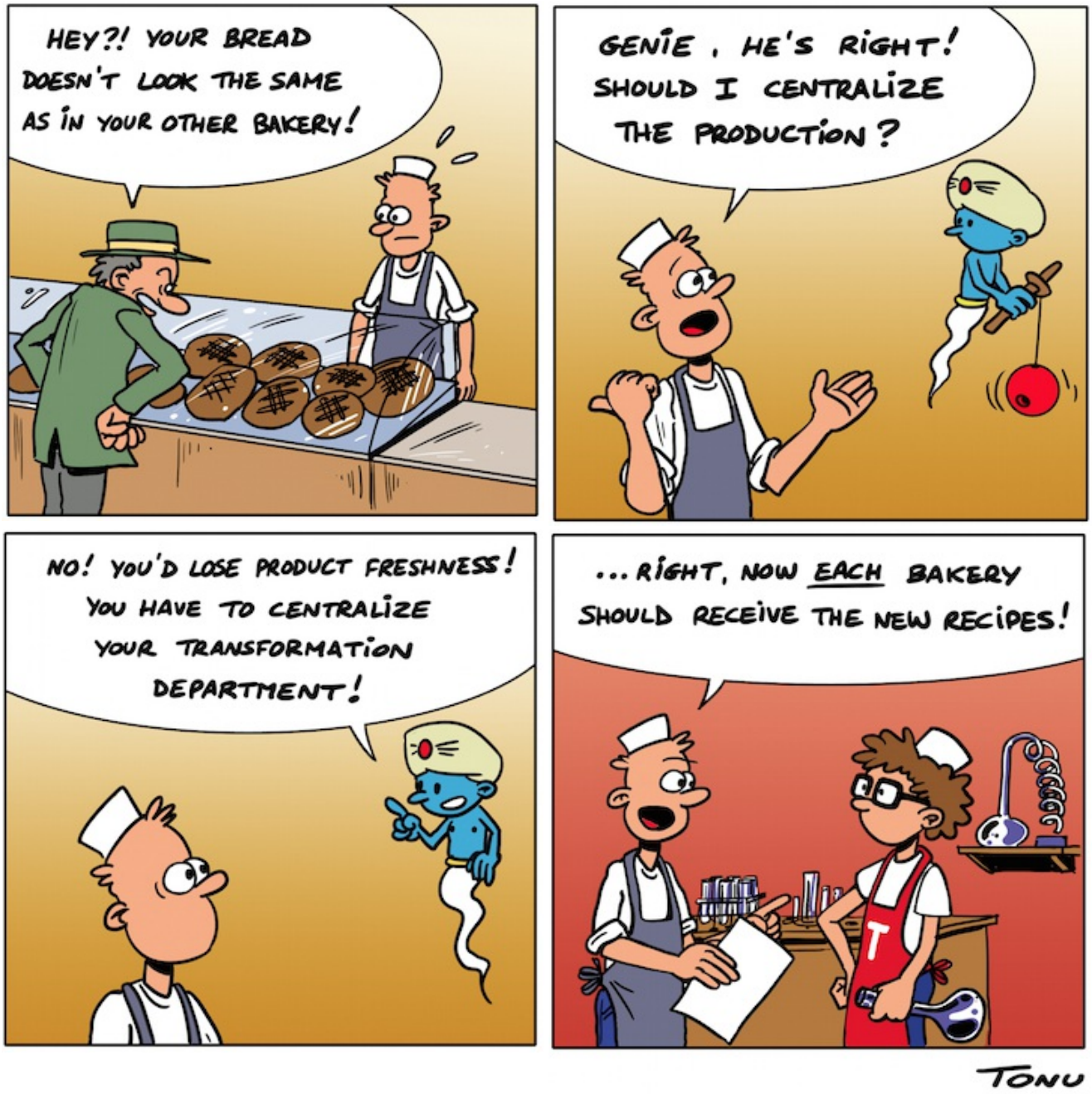
Their criteria are less based on these business plans than on the confidence they place in a team or an idea. They not only bring money, but also operational support and a network of contacts.

## 2. "Crowdfunding" enables individuals to participate in start-ups

"Crowdfunding" is a means of financing young entrepreneurial start-ups through individuals who invest modest funds via participative platforms on the Internet.
It is a fast way of locating financing; it is also a way of testing the market's interest in the product and to increase visibility.

See "les Echos" (5/11/2013): *Les start-ups americaines vont pouvoir se passer des banques pour se financer* (American start-ups will be able to bypass banks for financing)
*With no hope of getting a loan from their banker, American start-ups may soon be able to turn to any American to finance their projects. In the coming weeks, they will be able to sell part of their capital on the participative financial platforms found on the Internet. Start-ups will be able to raise up to 1 million dollars on the Web and approach non-accredited investors. "We want this market to prosper, while protecting the investors", emphasized Mary-Jo White, who presides over the regulatory body of the Stock Exchange. Enterprises can present their projects on the Internet portals and, in this way, convince individuals to enter into their capital. "This should enable the financial participative platforms to enter into a whole new era", commented Slave Rubin, CEO of Indiegogo, a platform created in San Francisco five years ago. The formalities to fulfill will be far lighter than those required to be listed on the Stock Exchange. Only those Enterprises raising more than 500,000 dollars will have to have their financial accounts audited. The others will be able to simply present their financial status once a year. Participative financing comes in several forms: some Internet sites offer money in the form of loans. The Lending Club is the most important one: it has enabled individuals to borrow around 2 billion dollars this year and posted revenues of 100 million. The second player, Prosper, is five times smaller. Other sites enable enterprises to obtain donations and to thank donors by distributing presents (T-shirts, watches, video games, etc.).*
See (in French) the file compiled by Croissance Plus.

## Centralize the Transformation




# 1. Why unify the Model?

If different Organizational units of a same Enterprise propose similar Offers, the Operation Models must be very close.
This leads Enterprises to centralize the Building (or purchase) of Models.
One example is Crédit Agricole: in the past, each of the 90 Regional Branches had its own IT team. But the costs incurred were found to be so high compared to its competitors that the IT Organizational units began to merge. The objective well underway is that there will only be one at the end.
The reason is not only to do with **cost** reductions but the capacity for **synergy** between Organizational units: if they share the same Model, every time a good initiative is identified in one of the subsidiaries, it can be quickly leveraged, with ease, by the other ones. It is far quicker to share **Models** than **ideas** which will need to be transformed into versions of the original Model

existing in each subsidiary. Lastly, it is a means of **harmonizing** Enterprise Information such as customer information or management information.

It is thanks to the uniqueness of the Model that Enterprises like McDonald's have been able to flourish. It is the franchise principle: we propose the same Model to Organizational units that carry out the bulk of the Operations ("the bulk" and not "all" as the parent company may want to central purchasing, for example).

## 2. How do we centralize the Transformation?

Many Enterprises keep the Transformation teams spread out independently on the pretext that local needs are specific. More often than not, it is a local excuse to keep one's local autonomy. It is very difficult to change this position: the opponents are determined and will not give up until their users voice their satisfaction with the new centralized Model which has been proposed to them.

To standardize the Models, the Transformation Organizational unit that creates them just needs to be centralized. The Models will then be the same for everyone.
The steps are as follows:

- **Check** if the **Offer Models** are similar or not. Similar does not mean "identical": there may be differences in language, tax systems, regulations... which justify adapting the single Model to each Organizational unit (see the topic on "configuration").
- **Isolate what is specific to each Organizational unit** and check that it can be respected with the same Model: be careful not to take all of the exotic demands at face value from the "separatist" Organizational units.
- Set up a **centralized Transformation** Organizational unit which builds or purchases the common Model, and also supports it at its customers, the different Business Units.
- If personalization is important, set up a small personalization team per Organizational unit.

On the other had, we do not recommend a federal approach which consists in favoring consensus : we bring together a certain number of representatives from the local Organizational units so that they build the new Model together. This scenario has the advantage of making everyone participate, but it is not very efficient as we only build an efficient new Model if one person heads the Transformation.

We can use this approach to gather enhancement requests, to manage a project portfolio adapting the existing Model, but we cannot use it to build innovative Models.

## 3. Better to have spread-out Transformation teams than a poor centralized Transformation team

If, to define a centralized Model, we decide to do away with the scattered Transformation teams and replace them with a centralized Transformation team, but that the latter is ineffective, it would be better not to change anything. It is certainly stating the obvious, but it is necessary to repeat it to ensure that the centralized Transformation team is managed by high-level, experienced people: see "Managing the Transformation Human Resources".

## 4. Centralizing the Transformation does not mean reducing innovation

The Model has to evolve.
Nothing stops us from setting up autonomous Transformation teams, who are there to build and test new Models. This often takes the form of intrapreneurship. These teams can be based in any Organizational unit.

But, when successful, the centralized Transformation team should take over the innovation and adapt the common Model so that the innovation benefits all: it is not the team who created the

innovation that should deploy it.

# Centralize purchasing


## 1. Centralize or decentralize?

A group made up of different Organizational units can be managed in different ways.
The simplest form of management for executive management is to let each Organizational unit be completely autonomous except for the financial consolidation. It is then just a question of distributing the brownie points and black marks and getting rid of the Organizational units that do not generate any profit.
This purely financial approach has the advantage of:

- Lightening the tasks of executive management: starting up an activity or acquiring an enterprise, then following the dashboards is enough.
- Motivating each Organizational unit, which is free to manage itself as it so wishes, providing that it generates a profit.

Unfortunately this approach has its obvious limits: no synergy in the Group, duplication of

investments, absence of economies of scale...

Conversely, we can set up a very centralized approach: the Models are the same, the Resources are managed centrally, the Customers are shared...
This approach has the opposite advantages to the decentralized approach: we can develop a consistent Image, a uniform Culture, a single Product catalog, global Customer management and global Resources management.
But executive management has its work cut out and the Organizational units feel deprived of all autonomy.

The key question is therefore: what is the right level of centralization taking into account the type of activity of the Enterprise?

## 2. Centralize the management of Resources

An Enterprise is nothing more than an agent which executes an Enterprise **Model** thanks to the Enterprise **Resources**.
We can therefore centralize either the Models or the Resources or both.

Centralizing the Models means:

- imposing the same Operation or Transformation Model on everyone, that is to say changing the work habits of those who Operate or Transform in accordance with an autonomous Model
- imposing the same Offer Model, that is to say removing all particularisms as regards Products.

It is easier to centralize Resource management and it is often the first step of a Group approach.
To take an example, if the Organizational units group their **purchasing** together,

- the Group will achieve economies of scale for everyone,
- it will be able to control more easily as there will not be any collusion with suppliers
- it will also save us looking at as many products to buy as there are Organizational units
- it will be easier, at a later stage, to impose a Product Model if the Components that are used to assemble the Product are the same.

It is the same for the other key Resources:

- centralize human Resources management to ensure that the recruitment, promotion and development criteria are identical
- centralize **financial management** to manage the liquid assets globally, negotiate more favorable credit conditions
- centralize the **management of premises** to share the same space between different Organizational units

# Generalize the principle:
# Centralize the Models and decentralize the Operations


## 1. To centralize or not to centralize...

"Centralize" means

- Homogeneity of both Resources and Models
- Better control over the Operations
- Economies of scale

but

- heavy to manage: economies of scale can be insufficient to compensate for the increased cost of complexity

- demotivating for the decentralized teams

"Decentralize" means

- simplicity of management for the Enterprise boss
- autonomy of the teams

but

- heterogeneity of the Models: staff rooted in autonomous decentralized Organizational units
- multiplication of investments
- difficulties to exchange best practices

Enterprises navigate between both forms of organization, some do not hesitate to alternate between them.
See *Challenges* (09/11/2013): *pourquoi certains souhaitent supprimer les fonctions de support centralisées* (article in French: why some would like to get rid of centralized support functions)

# 2. Centralize the Models and decentralize the Resources

A third form of organization exists which enables us to have our cake and eat it: **centralize the Models** to standardize the Processes, Roles and Information; but, **decentralize the Resources** to leave local teams with some autonomy.
Let's take some examples:

- Production
  - Centralize the Process Models, the way we Produce
  - Decentralize the Production in different locations where each one manages its purchases, its staff, its premises while respecting the Production Model
- Distribution
  - Centralize the Offer Models (the definition of what we sell) and Distribution Models (how we sell)
  - Decentralize the Distribution Resources, as happens with Franchises
- Human Resources Management
  - HR Management centralizes the Models: recruitment Model, training Model, evaluation Model
  - Each Organizational unit applies the Model locally: it manages its own staff while respecting the general rules defined in the Model

# 3. The organizational structures for the Operations and Transformation are not parallel

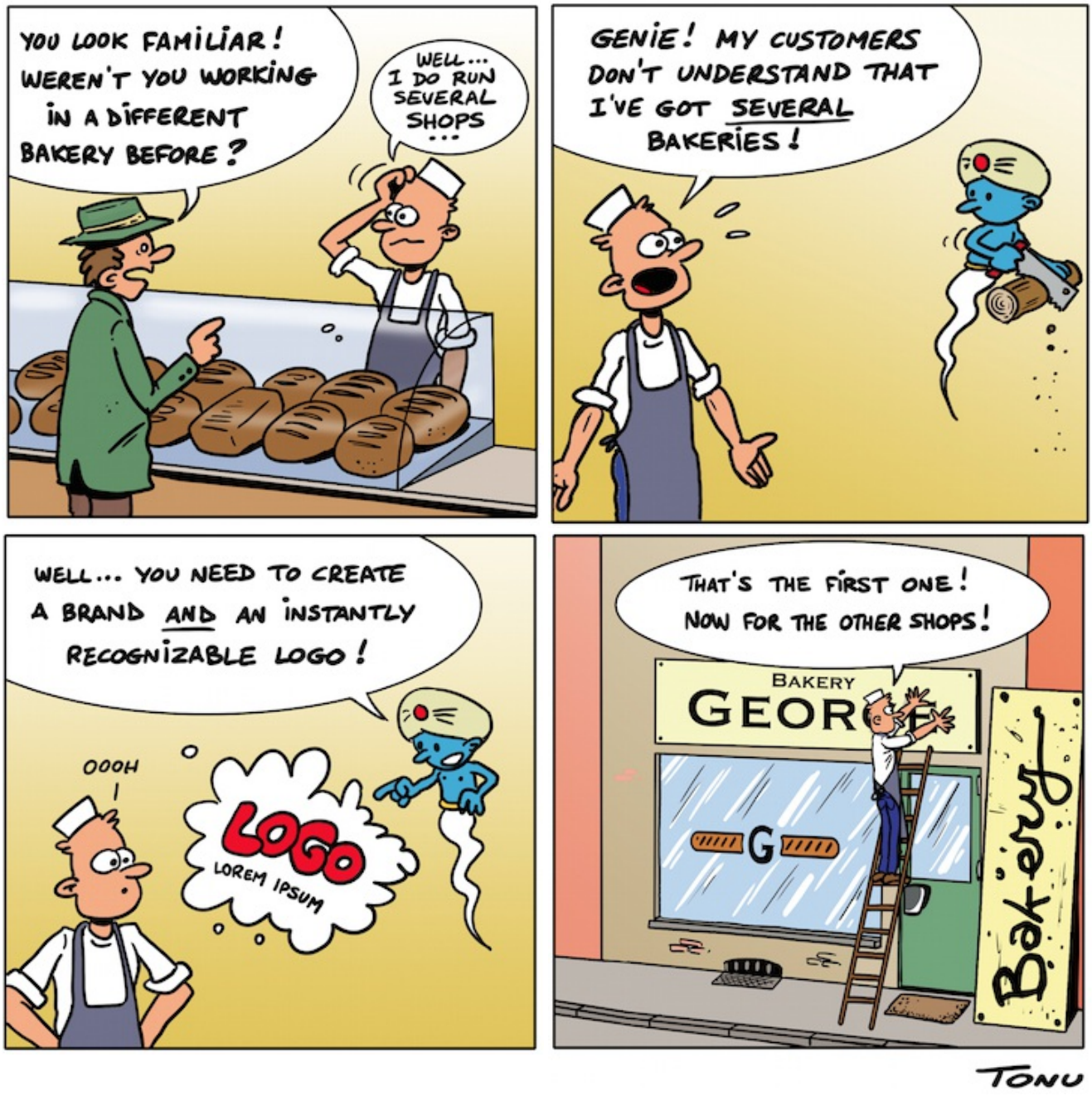
To apply this rule:

- The Transformation teams are organized by Model group.
- The Operational teams are organized by geographic region, or by Product line, or by Process domain.

This means that the organizational structures for Transformation and the Operations are not parallel.

Remark: the geographic division which was predominant in the Operations, in view of the time taken to move things around, is being gradually replaced today by a split by product line (e.g., Products for the enterprise or the individual) or by Process domain (e.g., separating Production from distribution).

# Power of image and brand




## 1. Why develop an image and a brand?

We buy a Product for the Value that it brings:

- Basic Value
- Usability Value
- Distribution Value
- Maintenance Value
- Image Value

A brand is a sign that allows us to distinguish the Products made or provided by an Enterprise from those of other Enterprises.
The brand can bring an image Value that can influence the consumer: it can bring prestige or trust.

It can help with Customer loyalty. It encourages the Enterprise to guarantee the high quality of its new products, so as not to damage the Image and brand.
The brand can enable us to satisfy the top layers of Maslow's hierarchy of needs: social recognition, membership of a group.

In a certain number of areas, Customers sometimes buy a brand more than a Product.

But be careful, image can also become negative and a strong brand recognition may turn against it.

## 2. How do we develop a brand?

Numerous possibilities exist to develop a brand.
There are classic ways like advertising or retail outlets or visual identity or the quality of the service...
But there are also digital ways of developing a brand. Social networks represent an important vector to broadcast a brand.

The size of the Enterprise plays a considerable role on the investments regarding image, especially for "B to C".

## 3. How do we take care of a brand?

Managing one's brand requires dynamically monitoring and managing the social networks, to intervene rapidly in conversations between Internet users in which the brand is being severely criticized. Large enterprises acquire monitoring and intervention tools for social networks (e.g., Radian6), and teams to exploit the data and take part in forums.

## Centralize the customer information


# 1. Why do we need to gather customer Information together?

Thanks to the potential of digital, customer relations are greatly changing: from a classic approach based on segmentation by **customer group** to a **personalized** approach, made possible by new digital means which are:

- connectivity of all sorts of **mobiles** linked to the **Cloud** which manages Information accessible anytime.
- **big data** to understand the customer context and offer the right product at the right time
- **social networks** which offer a lot of Information
- exchange **volumes** and **bandwidth** which are continuously increasing
- **education** and **equipment** of consumers

The amount of Information available on a Customer is increasingly considerably: by intelligently analyzing this Information, we can manage to better understand Customer behavior, reactions, expectations and therefore propose Offers to them that are more likely to satisfy them. The Customer can receive less solicitation messages and more targeted messages.
We go

- From a mass approach to a highly-personalized approach
- From an asynchronous mode (communication to customers happens after segment analysis) to a synchronous mode (we can inform customers via their smartphone, that they may be interested in certain opportunities in a nearby shop or in the shop where they actually are)
- From a mode where the multi-channel existed but in a compartmentalized way, to a cross-channel mode where customers can go dynamically from one channel to another, for the same Distribution Process. For example: signal on the smartphone, indication in the aisle, information picked up at the checkout.
- From a "pull" mode, where the Enterprise communicates to the end customer (or prescriber), particularly using advertising, to **attract him/her towards the product**, to a "push" mode, where the enterprise **pushes the product** towards the consumer.

| | | | | |
|---|---|---|---|---|
| **Classic Approach** | mass communication | asynchronous | Multi-channel, but compartmentalized channels | pull |
| **Digital Approach** | highly personalized | Real time | Cross-channel | push |

Huge use of customer information for purposes other than proposing targeted Offers.
It also enables us to manage

- A **Customer account** which groups together all the financial exchanges between an Enterprise and its Customer, linking credits which may go beyond the financing of a single Offer.
- Customer **risk**: is he or she a good payer or a defaulter?
- **Profitability** per customer: by taking an overall look at the Customer, we can find out his/her overall profitability and, in so doing, are able to reduce the cost on one Offer because we can make up for it on another, we can know whether to offer complementary insurance because he/she is careful...

Customer Information is useful not only in the Operations, as we have just described, but also in the **Transformation**: to rapidly renew products, we have to understand Customer behavior and expectations.

# 2. How do we centralize Customer Information?

## 2.1 Define the customer Information Model

To manage risk, profitability by customer or to know how to propose the right offer, we have to gather together the customer information spread out across the different Organizational units of the Enterprise.

The first action is, as always, to define the Customer Information Model

- Give a **definition** to the concept of Customer
-
    - Dissociate the Decision maker, the Subscriber, the Payer, the Beneficiary, the Product User...
    - Define whether the same person can play several of these roles
    - Define whether the Customer can be a natural person or a legal entity
    - Define whether the Subscriber role can be held be several persons (for example, a

joint bank account)

- Identify different **information groups** concerning the customer: legal information, family information, professional information, all forms of address, heritage, equipped with products, behavior...
- Define the **relations** between these Objects: a customer may have several addresses
- Define the **information** stored in each Object
- Define how to **identify** a customer: user-name format, who manages it, how do you deduplicate?
- Define the **Types** of each piece of information: how should an address, a name, a social and occupational category be represented...?

It is the rigor of this Information Model that will facilitate all the actions described above. We are not always capable of ensuring that all Customer Information respects this Model as the origin of the bits of Information can be extremely diverse.
Fortunately, today's **search engines** are able to work with imperfect and badly formatted data.

## 2.2 Model the Customer management Processes

Once the Information has been Modeled, we can Model the Actions that are based on this data:

- get to know the **opinions** Customers have on the brands, the Image, the Product and Service Offers of the Enterprise or its competitors
- deduce the Customer **expectations** from their behavior
- define the **right Offer** at the **right time** by using the geolocation possibilities
- select customer panels to continuously test our offers: apply "**lean startup**" principles (http://en.wikipedia.org/wiki/Lean_Startup) which recommend quickly testing our new offers by Customer panels to refine the Model before it is finalized.

## 2.3 Protect Customer Information

The large amount of Information available raises the problem of confidentiality and of the protection of private life.
Legislation is more or less restrictive or permissive depending on the country.
We have to respect certain principles:

- The Customer Information used must be accessible by the Customer who may forbid its use.
- Should an incident occur, the Customer and the data protection authorities must be immediately informed.
- We have to clearly define who is authorized to query which elements of customer data
- We also have to define a strategy of updating data: decide who is able to create or modify it. Should we inform a Customer manager for updates to sensitive Information?

## 2.4 Store Customer Information

The trend is to store the Information in the **Cloud** so that it remains accessible to all kinds of terminals, mobile or not.
But differences in legislation mean that we are not indifferent to where the information is stored (see the recent trials and tribulations of the NSA): Europe provides more protection than the USA.
The other trend is to **gather together** Information and not to distribute it. If the database is distributed, what criteria should be used to divide it up: by region? by customer type? Should we plan to have a duplication and replication system, which is relatively complex?

## Expand into international markets


# 1. In the past, we contrasted local activity to activity "abroad"

In the past, Enterprises were mainly national: they produced and distributed in the country or region where they were set up.
For the few international enterprises, we distinguished in their organization, the national department from the "foreign" department. The latter which grouped together the activities from the countries in which we produced and distributed, was more often than not considered a minor activity.

# 2. Today, Enterprises are working for the global market

But, since the last 30 years, the world has changed a lot:

- Enterprises began to **Distribute** in foreign countries, while keeping, in general, a local Production
- Then they decided to **Produce** in countries close to their customers.
- Then they decided to Produce in those countries where Production costs were low.

An increasing number of Products have become international, that is to say distributed in every country. Distinctions are gradually becoming blurred: it is increasingly difficult, when we travel, to bring back an original object.

Why are most of the Products offered in different countries the same?
There are three main reasons:

- People are traveling more and more and **identify the best products**: good products have worldwide success, bad products disappear. Product convergence is underway, especially for storable Products (Goods and Information) but not for Services.
- Costs and timescales for **transporting Information** have been greatly reduced
- Costs for **transporting Goods** have also considerably decreased despite the sudden rise in the cost of fuel, thanks to containers, which are vehicle or cargo-sized and to transport optimization Models.

Convergence is underway.

- It is the case for **Goods**: Ikea, Apple, Mercedes, Nestlé, Sanofi, Michelin... offer almost identical Goods in different countries.
- It is also true, in part, for **Information**
  - **Operational Information** is partially global: the cinema and music are international, news is not due to language and local interest differences
  - **Models** are becoming global: this is significant in software package industry (see the CEISAR white paper)
- **Service** is still partly protected: but we can see the Banks and Insurance companies bringing the Product Models in different countries where they operate closer, even if regulatory differences still require us to personalize the Models.

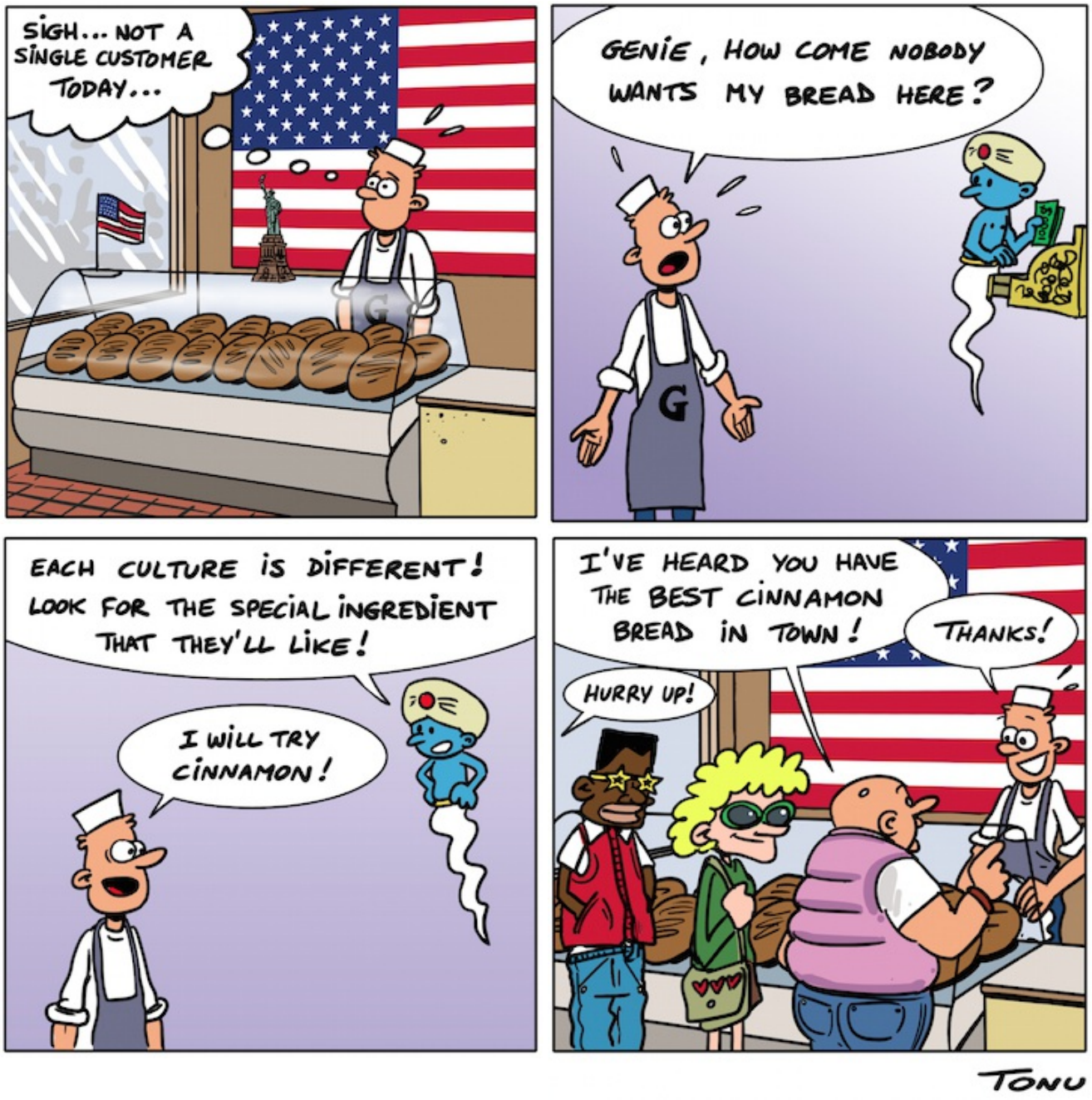
# 3. Going from a local market to a global market changes the rules of the game

The consequences of globalization are clear:

- The size of the market leads to more sizable Enterprises: there is a reduction in the number of Enterprises who are able to remain in mature markets.
- Economies of scale are considerable.
- Enterprise management becomes international and distributed.

Luckily, the "big brother" effect is compensated by the rhythm of Enterprise startups which bring original value propositions.

## Differentiate the products by country




# 1. Some specializations by country remain

The specializations by country that remain are linked to 2 domains:

- Regulations, tax system, local norms which are imposed by the local authorities.
- The fundamentals of Culture: language, dietary habits, personal image (in the USA, you need to have a big car or a big house to be well viewed)

# 2. How do we personalize the Products and Services?

To **personalize Goods**, the most effective technique consists in isolating the parts that can be personalized as Components, so that the architecture of the Goods and most of the pieces are reusable and that assembling the specific parts is easy.

To **personalize Operational Information**,

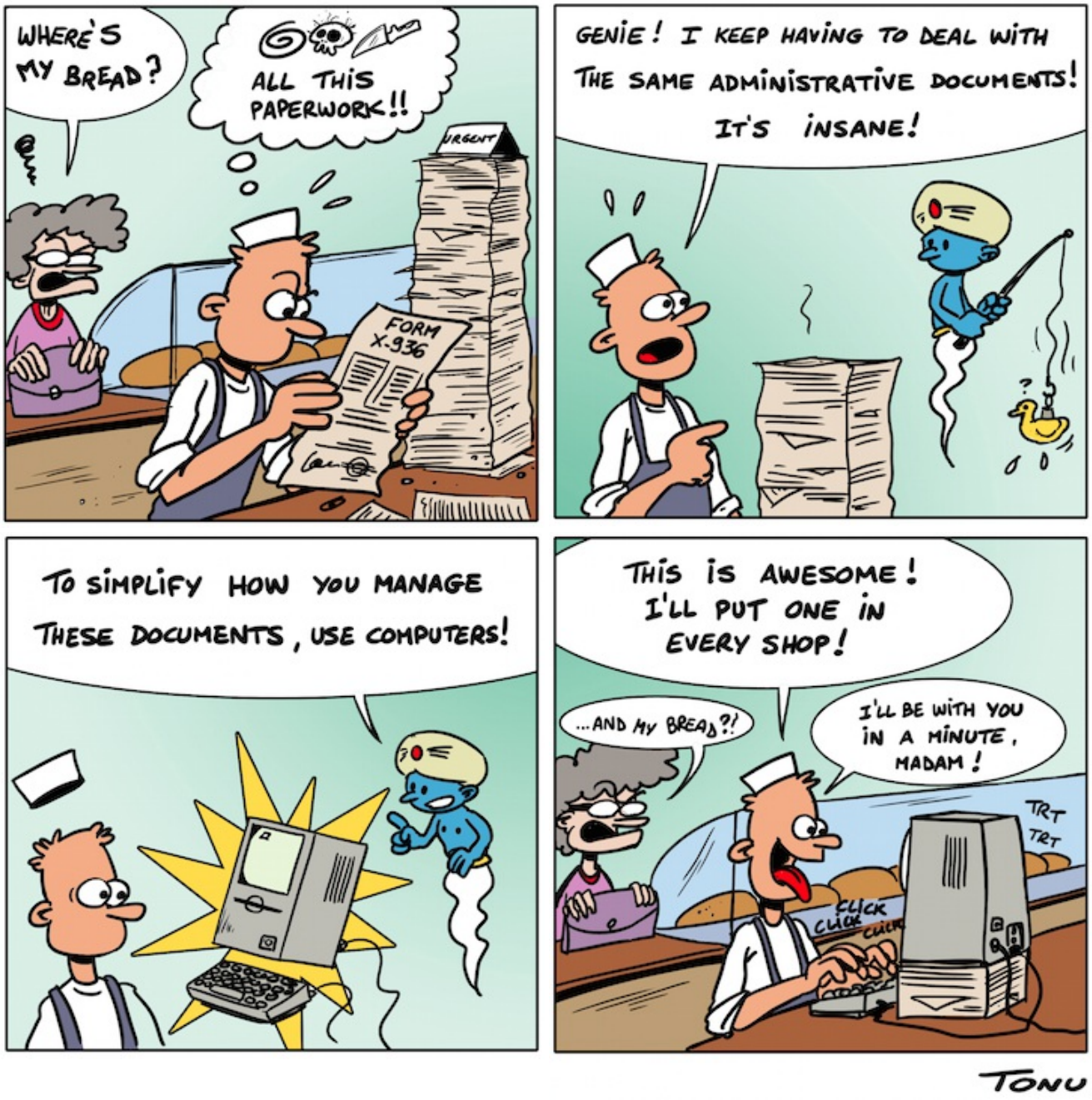
- we have to begin by translating the language. In view of the differences in concept in the different cultures, a literal translation is not always enough: we need to know how to adapt the Information.
- We also have to try to locally produce  the information that is only used locally. It is not always possible: for example, satellites can provide weather information that is only used locally.

To **personalize Information for the Model**, we have to do as for Goods, that is to say build in the form of components to isolate the local part and play with the language.

To **personalize the Services**, there is less pressure as Services are generally executed locally: only the Service Model has to be built in component form, as described above, to adapt language, taxation and regulation factors.

## Act 3: Improve efficiency through digital capabilities

# Digital Actors help Human Actors




## 1. Certain Actions can be executed by Digital Actors

Traditionally, Actions were executed by Human-Actors.
But, since the arrival of programmable machines, a growing part of the Actions is executed by these machines which we refer to as "Digital Actor" or "IT-Actor" for short.
They are workstations, servers, smartphones, tablets or any "programmable" object (or rather one that is "intelligent") that can be incorporated in Goods today.
We often combine the Human-Actor and the IT-Actor to increase the capability of the Human Actor.

Using the "Human-Actor" and "IT-Actor" terms may not please some people who do not want us to be able to associate a human being with a machine. But the use of the word "Actor" to represent

the one who executes the Actions is used in order to show that the Actions of a same business process can be executed by one or the other. That certain Actions are taken into account by IT-Actors relieves the Human-Actors of Activities with little added value, which enables them to focus on more noble tasks.

In any case, an IT-Actor can only execute the Actions for which it has been programmed by the Human-Actor. The IT-Actor cannot decide or invent or innovate, but it can be a good tool to accompany the Human-Actor in his or her tasks.
The Human-Actor is an autonomous social system with his/her own objectives (like an enterprise) whereas the IT-Actors are not autonomous and do not have any objectives. This leads to constraints and different results in the use of these resources: as an example, we have to give meaning to a Transformation so that the Human-Actors will accept it, whereas the IT-Actors do not have to accept it.

# 2. The IT-Actor Model is software

Procedure and software are the Models that the Actor must follow.
A **procedure** is a list of instructions for the Human-Actor to carry out his/her activity properly. For example, a recipe contains the list of ingredients and the list of actions to execute with these ingredients. It is an **Information Model** (the ingredients) and an **Action Model** (the list of Actions).
The list of instructions for the IT-Actor is called **software**: it also includes its Information Model and its Action Model. **Software is nothing more than a Model**.

# Standardize Solutions from the same functional domain


In a decentralized Group, each Organizational unit decides how it wants to Operate.
This means that the Organizational unit chooses its Model and especially the IT Solution that goes with it, whether it be to manage the Distribution, Production, Resources or management. It is **far simpler** for the director of the Group, who does not have to get involved in these choices, and more motivating for each decentralized Organizational unit, which gains in **autonomy**. This is why many Groups have historically preferred to multiply decentralized Solutions.

But this approach has limitations:

- The **Transformation cost** is higher: each Organizational unit carries out similar studies for its choice of Solution, purchases its software packages, links the Solutions to each other, builds training supports, manages its own hotline, carries out upgrades,...

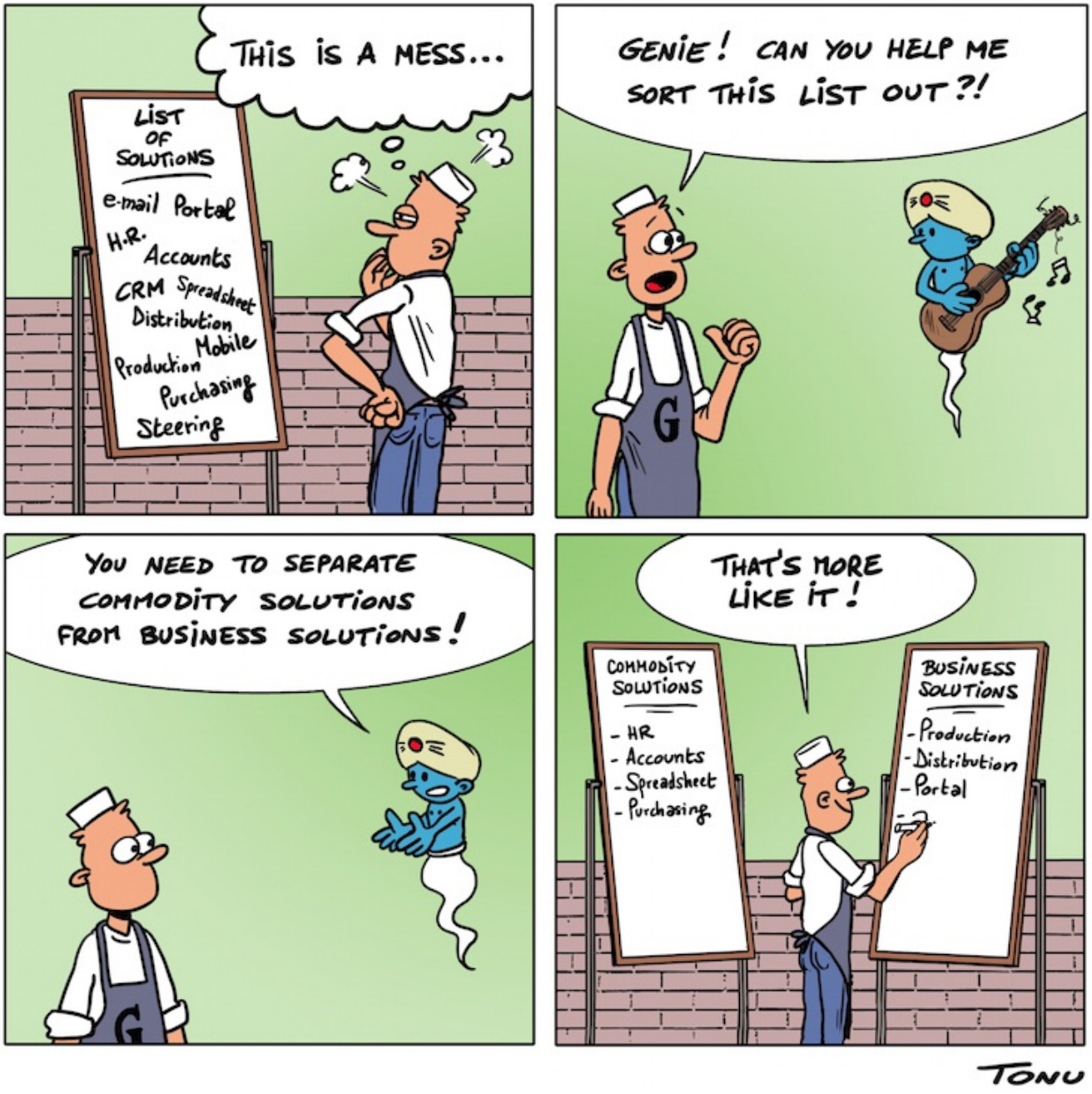
- **Operational synergy** is impossible: how do we build shared service centers if they do not use the same Model? How do we duplicate best practices if the Models are different?
- It is more difficult to **transfer** employees from one Organizational unit to another, as a new Model will have to be used which requires an important effort to made by the employee. The opportunities offered to the Enterprise employees are more limited.
- It is more difficult to obtain good **performance management information**: it is difficult to add apples and oranges.

It is therefore recommended that we choose **Solution Models that are as close as possible** for the different Organizational units. We have to learn how to maximize the common part and minimize the specific part.
We go

- from a time where value was given to subsidiarity, "small is beautiful", decentralization,
- to a time where value is given to synergy, economies of scale, coherence and centralization.

Therefore we can only recommend extreme vigilance on this point: each time an Organizational unit would like to choose a specific Solution, we must ask it to justify how its mission is different to the other Organizational units and in what way the standard Solution will not enable it to carry out its mission.
It is not a simple task as certain Organizational units are likely to highlight their specificities to keep their specific Model and their autonomy.

# Distinguish Commodity Solutions from Business Solutions




## 1. What is a Solution?

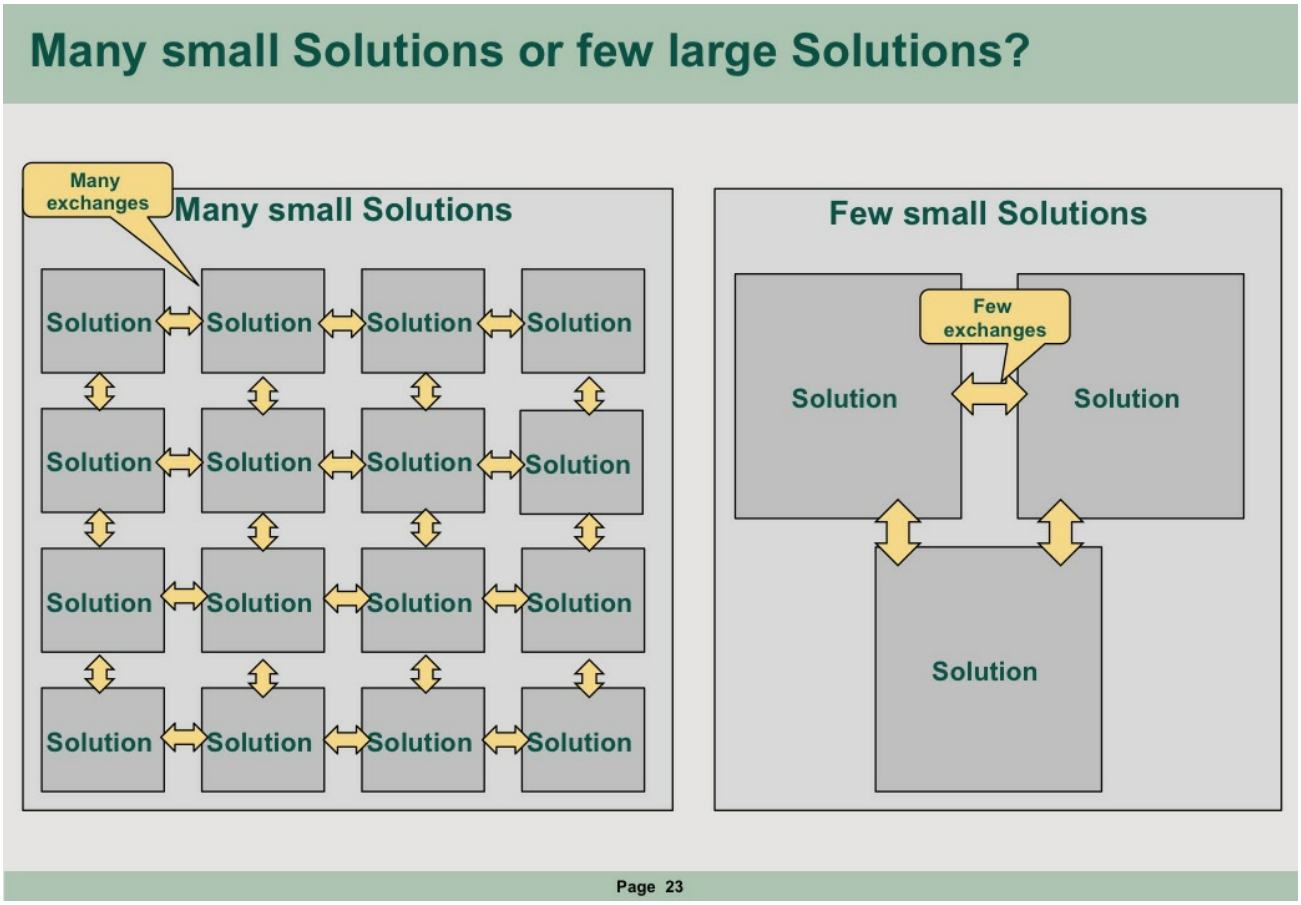
A Solution is one part of the Operation Model modeled by the same Transformation team: it covers one or several **Processes**, sometimes simply **Functions** of a Process (e.g., pricing Solution or security Solution or Word-processing Solution).
A Solution composes both the procedural Model for Human-Actors **and** the software Model for IT-Actors. This is why we prefer the term Solution to the traditional term of Application, which is limited to software and does not include the human procedures.

## 2. Well-interfaced Solutions

The main quality of a Solution is the absence of redundancy.
The functional coverage must be clear: a given Function is only provided by one Solution.
As an example, if a Solution manages Customer Information, all the other Solutions should feed into it or reuse the Functions it provides to update the Customer.
Each Solution therefore provides a list of interfaces, which are reusable by the other Solutions.

# 3. Many simple Solutions or a few large-scale Solutions?

We can use a multitude of small Solutions: each Solution is easy to pick and use, but interfaces are numerous and create a complex system, hard to upgrade.
We cannot recommend enough using a **small number of key Solutions**: each one is more complex because it uses part of the aforementioned interfaces, but the links between Solutions are far simpler. The possibility of integrating a complex system in two stages, first within the Solution and then between Solutions, means we gain in terms of testing, robustness and modularity.




# 4. Functional Domains

We can classify the Operational Solutions in different domains:

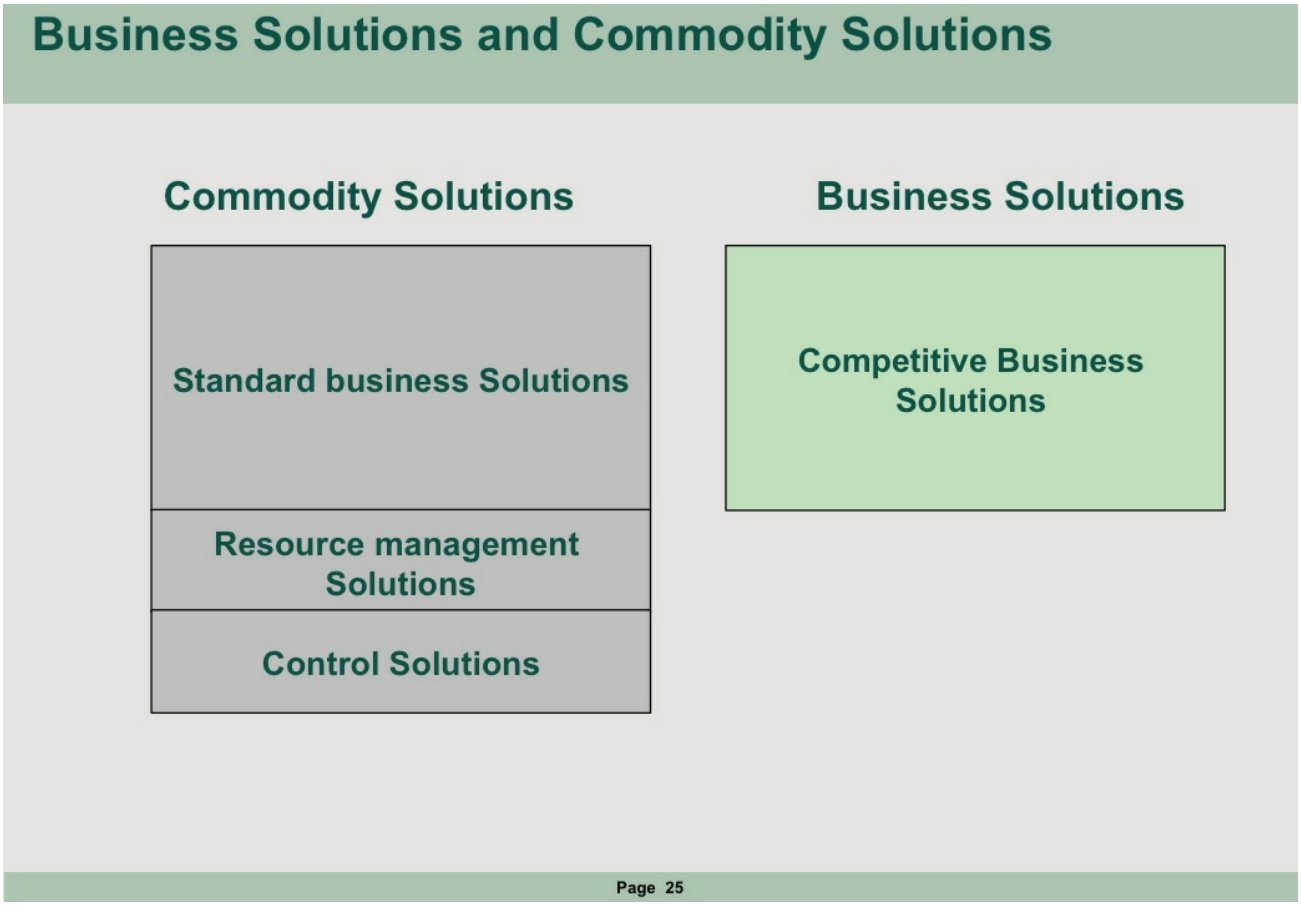
- Production of the Product be it a Good, Information, or a Service
- Distribution of the Product Offer
- Usage of the Product (for Goods and Information)
- Product development and maintenance
- Resource management: to manage Human Resources, IT Resources, Financial Resources, premises...
- Managing the Enterprise

# 5. Commodity Solutions and Business Solutions

**Business Solutions**, sometimes called "Vertical Solutions", are the Distribution and Production

Solutions.
**Commodity Solutions**, sometimes called "Horizontal Solutions" are so called as they are the same, whatever the business: payroll or accounts are Commodity Solutions reused by Enterprises with different Businesses.
At any given time, among the Business Solutions, some are differentiating, others are not. For example, insurance companies consider that managing claims is today commonplace. However, the day when they transform the current reimbursement processes into new Services, these Claims Solutions will become differentiating.

So as not to complicate this presentation, we will keep the term "Business Solutions" for the competitive Business Solutions.



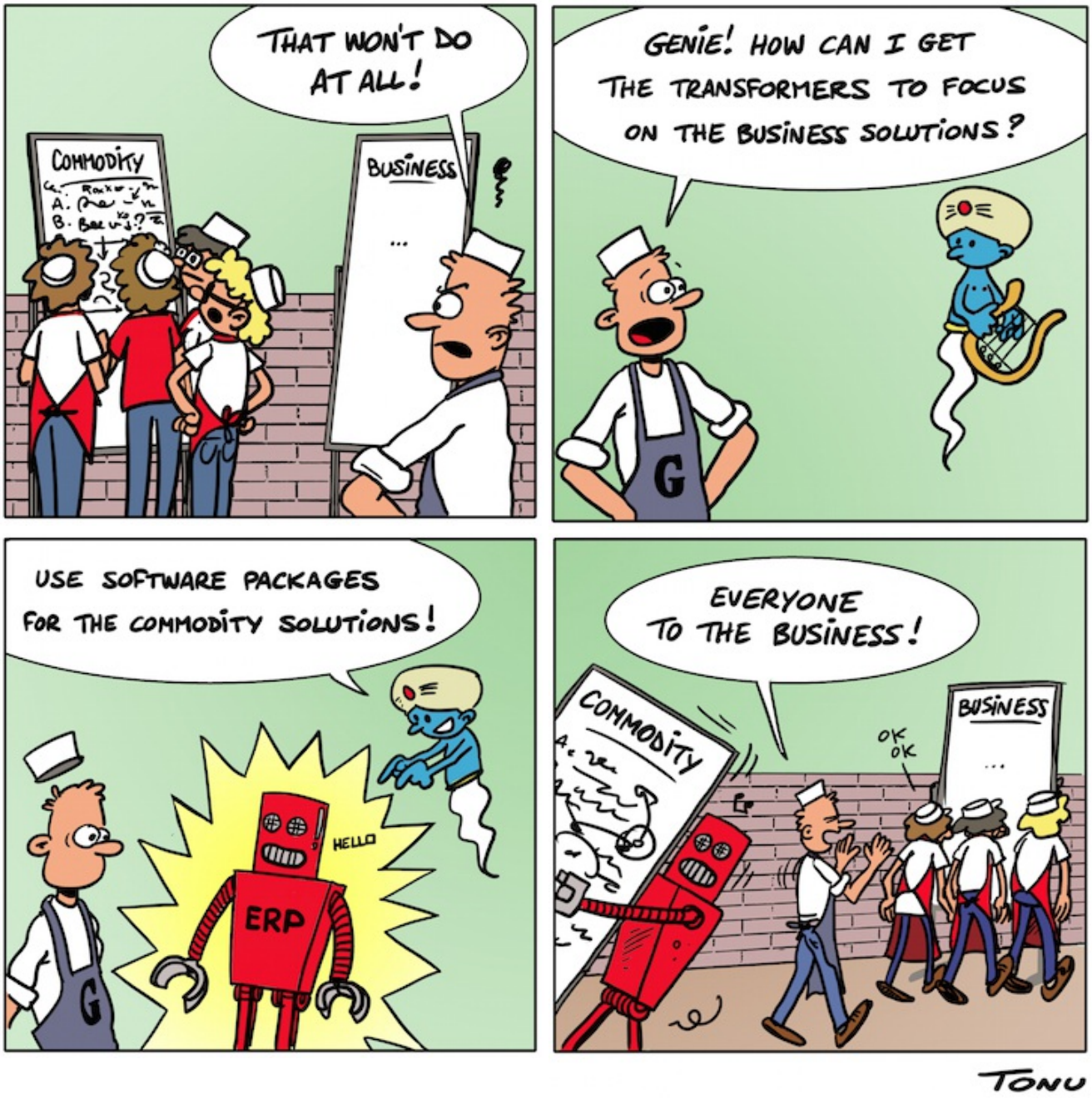

# 6. Keep the responsibility of Business Solutions

The Business Solutions are the ones that enable us to differentiate ourselves. It is therefore important to control the Model and its execution.
Controlling the Model is more important than controlling its execution.
Thus Apple built its Product Model but has these Products Produced by Chinese enterprises for competitiveness reasons in compliance with a production Model, the essential of which has been defined by Apple.

In the same way, Apple does not produce the Weather Information or iTunes music: it contents itself with building a Usability Model with this Information, which is the real competitive advantage.

Regarding Distribution, Apple built a Distribution Model for its Apple stores, but it also leaves its partners the task of Distributing according to their own Model.

# Use Software packages for Commodity Solutions




## 1. A software package industry has grown up around Commodity Solutions

As Commodity Solution needs are similar between Enterprises with different activities, a software package industry has developed around these Solutions, and has done so all the more rapidly due to the huge size of the Market.
Oracle, SAP, Microsoft, Google... have built and Distributed Commodity Solutions used by a growing number of Enterprises.

These Software packages today have 2 forms of distribution: either in **license** form, which enables internal use in the Enterprise, or in **SaaS** or Cloud form (see the CEISAR Cloud white paper).
In both cases, software development is handled by the supplier even if adaptations are often

necessary.
Today, we no longer therefore develop a human resources management Solution internally, we turn to a software package that includes not only the software but also the related human Procedures.
The advantage, for Enterprises, is that

- The Business managers can visualize, when choosing, a **concrete** Solution that is already available and offers part of the desired Processes. If the Business Units have suffered from past failures during a badly managed project, they can feel relieved and **reassured** to rely on a Solution that has already been deployed successfully in other Enterprises.
- The Business managers' **task of defining new Processes** is simplified: they carry out more of a gap analysis, at the time of choosing, to check that no functionalities are missing, which is far less demanding.
- **Costs** and **deadlines** should be reduced compared to an independent Solution as the Software package investment and development are shared between several Enterprises, including developments due to changes in regulations.
- the Software package has been better trialled and tested, has better availability and richer functionalities than an internal Solution.

On the other hand, the Software package can turn out to be heavy for smaller customers if the package is built as superset of needs for everyone and not as a modular set where we only select what is useful.

# 2. The Software package can only be an answer to Business Solutions if it is built from Components.

A Business Software package industry is developing today.
However, it is more difficult: on the one hand, the market is limited to enterprises with the same Business, which might discourage investors. On the the other hand, the will to differentiate is stronger, which requires the software package to have easy modularity, therefore being more difficult to build.

The Software package Solution can only be an answer to Business Solutions if it is built from reusable Components: the modularity must mix the capability to be able to stand out with the robustness of common architecture, but in that case, the level of requirements is higher for the Software vendor.
As Businesses change ever quicker, it is extremely difficult for Business Software package suppliers to satisfy existing customers and, at the same time, go after prospects who are seeking, from the software package, the latest digital functionalities.

# 3. How do we select a Software package?

We have to take into consideration not only the available Functionalities, but also the overall costs and deadlines.

## 3.1 Functionalities and ease of use

An already-available Solution is always more attractive than a Solution that must be Built.
A "Commodity Solution" will, of course, evolve with changes to the organization and regulations, but it will always develop within a known perimeter.

Therefore, the first criteria of choice are naturally the availability of the required **functionalities** for the Enterprise and ease of use, that is to say everything the end user sees.

The other criteria are the installation time, the cost (particularly for future additions), the longevity of the supplier, the ability for it to be integrated into the enterprise Architecture (duplication of data, specific user interface, specialization of the Transformers), the possibility for differentiation, and the speed of development.
It must be the full cost: the purchase of the Software package (license or use rights) is only a

negligible part of the total cost. We have to add:

- Cost of personalizing the software package during its installation and its life in the enterprise
- Cost of interfacing it with other Solutions
- Cost of migrating data to the software package
- Cost of upgrades
- Cost of optimization and tuning
- Cost of Deployment: training, hardware installation
- Cost of use for the end user: the cost is higher if the usability Model of the Software package is specific
- Cost of operating the solution

## 3.2 Do not forget the software package's capability to evolve

But one essential criterion is often neglected at the time of choosing: the **evolution capability** of the Software package.
Yet, Solutions have to evolve:

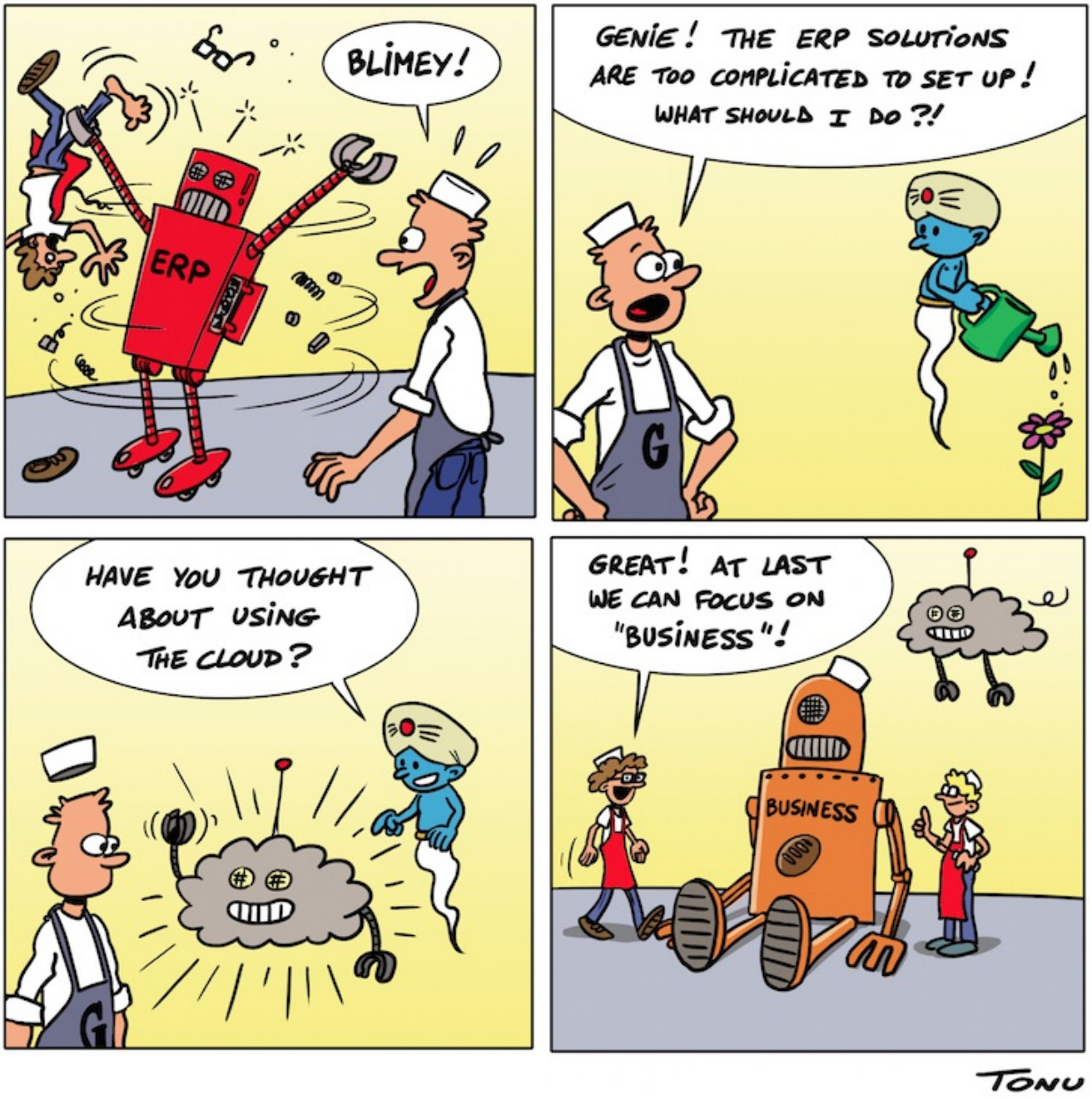
- Regulation changes
- Technology changes
- We are able to automate an increasing amount of functionalities and the Software package is gradually enriched
- We connect an increasing number of Mobiles to the Solution
- We share some of the functionalities with partners and customers
- The Software package has to be interfaced with other Solutions.

If the Software package has a real capability to evolve, it will be easy for the vendor to add potentially missing Functions or to optimize its performances or its reliability, in other words to gradually compensate for any weaknesses in the Software package. Otherwise, the installed software package will quickly age and will have to be replaced rapidly, which is not always really understood by the users.

To judge the capability of the software package to evolve, we must not hesitate to understand its **architecture**, to question existing Customers of the software package on how easy it is to **upgrade** or **personalize** the Software package.

*For more information, see the software packages white paper.*

# Leverage the Cloud




Installing a **software package** is generally more complicated than anticipated.
We have to configure the software, install the IT hardware, test, optimize the performances, train the operating systems team, manage upgrades, adapt the configuration to the workload...

The **Cloud** is one way of simplifying the life of the Enterprise by transferring these concerns to the **supplier**: not only does the software used belong to the external supplier, but operating it is managed externally, which enables its use to be commercialized as on-demand or subscription-based consumption, and not as a license.

This phenomenon will drastically change the relations the Enterprise has with its information technology:

- The requirements of the Cloud will increase the **quality** of the software operated

- Using Cloud Solutions will foster simplicity: the supplier delivers some part of the support, it is its own interest to reduce the number of calls
- Service quality can only grow: contracting a « SLA » (Service Level Agreement) pressures the supplier into assuring a high level of service if he does not want to loose its Customer or impact his reputation
- Enterprises will be a lot more **agile** when it comes to installing new Solutions
- **Small Enterprises** will be able to acquire modern, affordable Solutions
- The Cloud will help the **extended Enterprise** to function by integrating not only its staff but also its partners and customers into the shared Processes
- The Cloud will promote **mobility** and innovative forms of organization
- The Cloud will enable us to **imagine** Solutions that were not possible before.

To obtain this value, the Enterprise has to take up the following challenges:

- How do we ensure the protection of the data that is "elsewhere"?
- How can we accept a standard Solution that we cannot modify, especially when it automates the core of the business?
- How do we integrate "SaaS" Solutions without increasing the complexity of the Enterprise Model?
- How do we adapt our organization and share out the roles between Business and IT?
-
    - How do we strengthen the role of the architects to evaluate the quality and integrate Cloud services into the Enterprise Model?
    - How do we increase the configurators' role at the expense of the developers'?
    - How do we manage the disappearance of the IT operators of the Enterprise?
    - How do we make the role of quality manager for service and security emerge?
- How do we control new contractual practices and correctly plan for the SaaS Solution release?

The migration towards this new form of consuming information technology does not only impact the Customers, it is also going to dramatically change the positions the suppliers have: the existing actors (vendors, builders, integrators, operators, IT services company...) will either have to adapt or change business.

- large traditional Vendors may **disappear** if they do not convert to SaaS
- Software Vendors will have to be much more **rigorous** about their product quality
- the **size** of Business Software Vendors will increase in order to meet more important investments than a software package for large vertical applications, and will coexist with a multitude of tiny vendors for autonomous applications
- **Integrators** will see their market shrink as specific developments will be limited: they will reconvert to configuration and interface building work to make the Cloud Solutions and existing Solutions coexist, as well as Data migration
- **BPO** Suppliers will grow: the economies of scale generated by the SaaS Model will see them become more competitive
- **Infrastructure Suppliers** will become increasingly powerful to facilitate scalability: they will become the main buyers of IT hardware suppliers. They will act as the large supermarkets were able to do with their suppliers. Some (like Google) will even be able to assemble their own servers from components bought directly.
- **Consulting companies** will have to be able to advise their customers so that they benefit from Cloud offers in their business.

The Cloud accompanies the generalization of agile Approaches and the continuous development of highly configurable off-the-shelf solutions, with few specific developments, and more minor but more frequent developments than for historical software packages. Cloud projects bring faster results more cheaply, minimizing the risks of failure (and associated cost) and enabling the gradual building of solutions with an increasingly wide scope.

Among the main features of a good SaaS solution, we have selected:

- A good **configuration** capability: the same Solution is exploited by many customers who

must be able to personalize it to their needs through configuration and not through specific development
- A good **integration** capability (both between Clouds and with the internal IS) thanks to rich and stable APIs
- A good **evolution** capability which requires a sound Architecture base and a process of continuous development

The SaaS Solution life cycle includes the following phases:

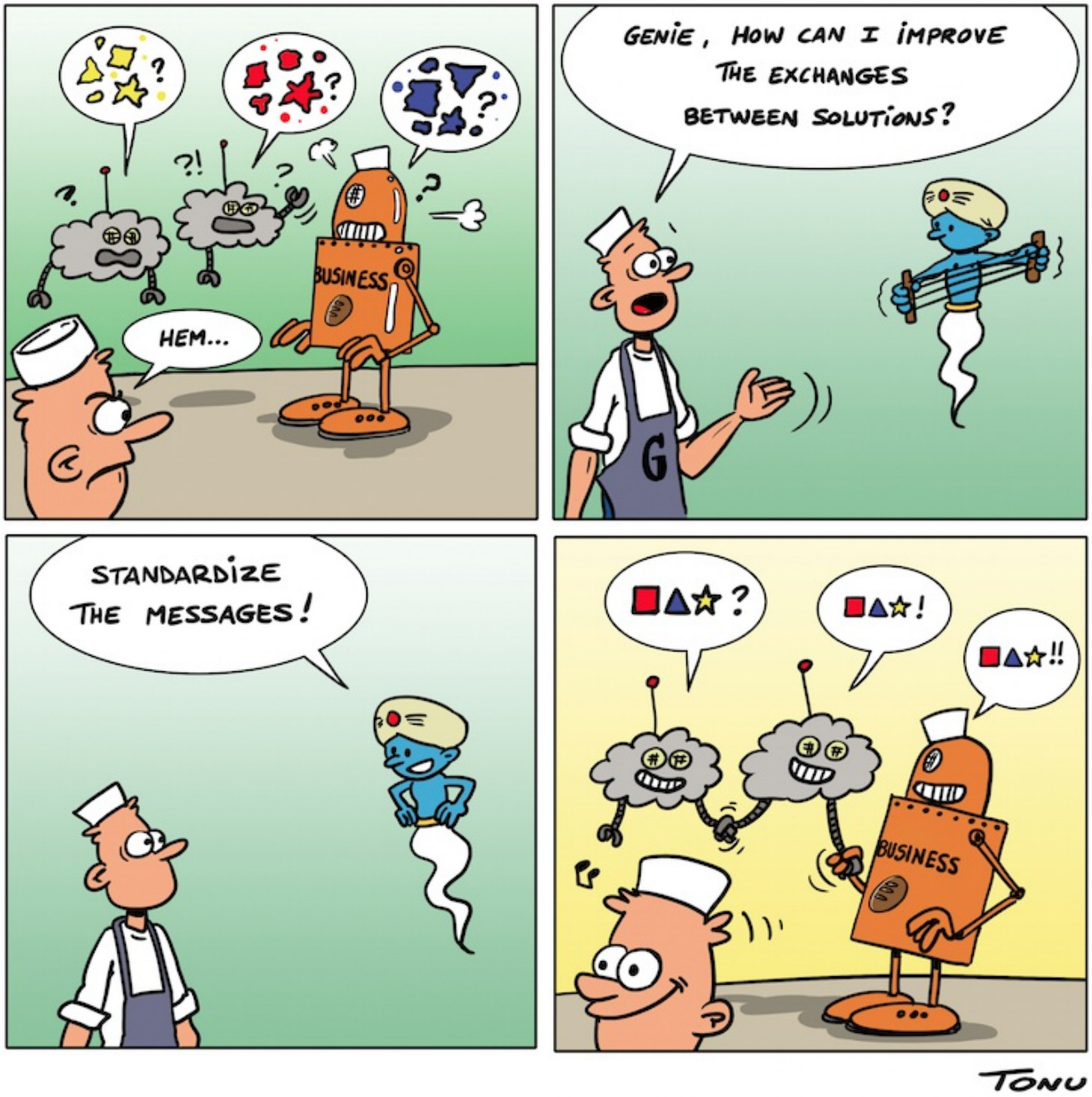
- Evaluate: the enterprise should carry out active monitoring to judge the quality and security of the Cloud solutions. During an implementation project, it must audit the development, evolution and operation capabilities of the Solution by the supplier. It must also identify the type of contract proposed and the level of negotiation possible.
- Decide: the Enterprise must evaluate itself regarding its maturity faced with the Cloud, know the existing setup, the initial costs, know the level of criticality of the solution to be built, build a business case and check the quality of the solution proposed and the reliability of the supplier
- Implement: the Enterprise should be aware that the implementation has to go mainly through configuration and that the requirements must be simple and prioritized. Implementation is generally iterative and goes through configuration rather than specific development. An important part concerns data migration and integration with the existing setup which is only possible if the solution provides good APIs.
- Deploy: Deployment can be made easier thanks to a simple and user-friendly interface and an international access thanks to the Internet. But, the need to give up on certain Enterprise specificities to align oneself with the standard of the SaaS solution may entail a significant change management.
- Operate: The governance of the services in production must be prepared in the implementation project and include precise supervision of the service performance indicators. The vendor exploiting the solution moves closer to their customers and the role of service manager becomes professionalized.
- Terminate: The end of contract should be planned when the the contract is first signed with the conditions of data recovery (format and timescale). This point should also be reviewed each time the service evolves.

The Cloud appears as a heavy trend, the culmination of the gradual evolution of technologies and practices, that will profoundly renew the roles of all the actors of the ecosystem, as much in the enterprises using the Cloud as at the suppliers. In the choice and implementation of solutions, it strengthens the role of the enterprise architect, who is the conductor of the building and evolution of an agile IS, and who knows how to leverage the full potential of this new mode of making IT resources and software available.

*See the white paper from CEISAR*

## Act 4: Equip oneself with a Foundation

## Interconnect the Solutions from different domains




# 1. Dynamism and order

The diversity of the Transformations leads to a growing number of Solutions: we have to not only manage Distribution, Production, Resources and enterprise management Solutions, but also

- Accompany the new Offers made possible by the digital opportunities
- Manage the multi-channel
- Add data analysis Solutions (Big Data)
- Interconnect partner Solutions, which are increasingly integrated with those of the Enterprise
- Allow the different Models to access the different Solutions

Do we need a **framework** to ensure that these multiple Solutions are part of a coherent whole or

should we leave every **freedom** to each Solution to not harm the dynamics?

In actual fact, we do not have a choice; different factors lead us to create an overall framework:

- Group the information that enables us to manage the **Customer** together: his/her behavior, Product equipment, expectations, profitability, risk...
- Agglomerate coherent **management** data
- Control the **end-to-end Processes** whatever the channels used
- Offer a **uniform Usage** to Users so that they are not afraid of crossing from one Solution to another.
- Share **identification** and **security** Functions

The real difficulty is in setting up a framework that not only does not slow down the initiatives, but actually accelerates the Transformation projects.

Different methods contribute to this. We have grouped them together under the term "Foundation":

- **Interconnect** the Solutions from the different domains properly
- Reuse **Components** to build new **Product** Models
- Reuse **Components** to build new **Solution** Models
- Provide consistent user **Usage**
- Harmonize the **Transformation Processes**

In this scene, we cover the first method: Solution exchanges.

# 2. The different Solutions need to exchange with each other

## 2.1 Solutions supply other Solutions

The first Solutions installed in the Enterprise were independent of each other. We had to re-enter the same information in the different Solutions, which represented a heavy workload, data entry errors, update discrepancies, in short a growing inconsistency in the Enterprise information system.
Then, we understood that we had to interconnect the Solutions to avoid these difficulties and let the information from one Solution flow automatically to another; for example:

- Production feeds into Distribution.
- Payroll feeds into accounting.
- All Solutions feed into the management Solution.

## 2.2 Data is shared

The same **Customer Information** is useful in different Solutions: Distribution Solutions of the different Product lines, maintenance Solutions, billing Solutions... We must therefore be able to share the same Customer information between different Solutions.

In the same way, Information which describes the **Enterprise structure** or the user rights and responsibilities are useful in all the Solutions.

## 2.3 End-to-end Processes cross several Solutions

Customer order management can cross different Solutions: quotation Solution, contract management Solution, billing Solution, payment and litigation management Solution, delivery management Solution, sales statistics Solution... Each Solution must be able to feed into the next Solution in the context of an **end-to-end Process**, while safeguarding the context of the

Process.

# 3. Different types of exchanges between Solutions

There are different types of exchanges between Solutions:

- Synchronous or asynchronous
- Query or updates of information
- Triggering IT Services (that are frequently implemented today as a Web Service)
- Data replication
- Flows between Solutions

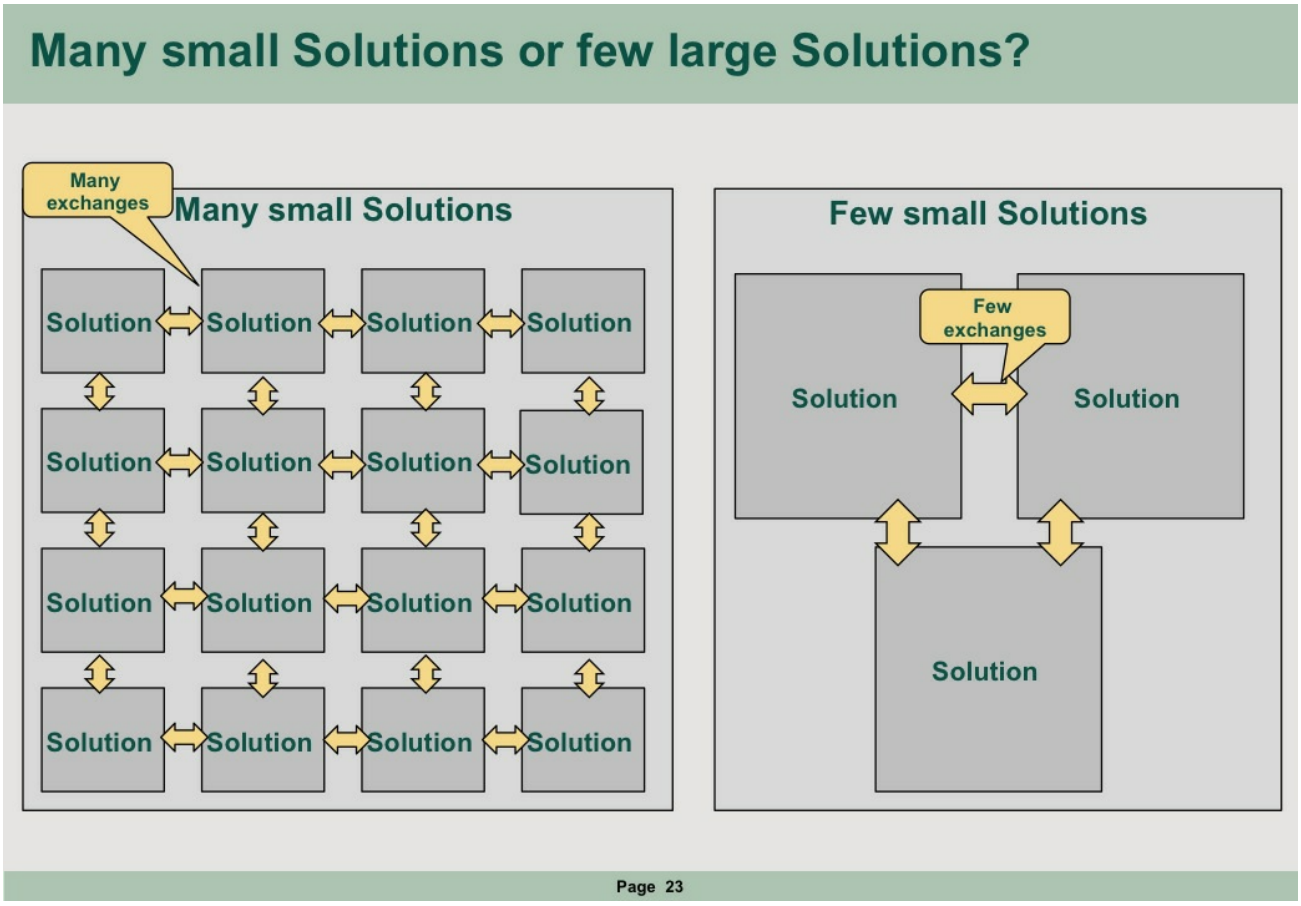
# 4. How do we properly define the exchanges between Solutions?

The multiplication of exchanges generates significant complexity: certain IT directors complain that they have become plumbers, spending their time connecting Solutions.

It is quite common that building these exchanges takes more effort than building the functionalities awaited for by the users of the Solution.

To limit this complexity, there are 3 ways:

1. Limit the number of Solutions: look for Solutions broad in scope so that a certain number of exchanges are taken into account inside each Solution.




2. Group together the different exchanges in wide reaching Services. For example, if we need to

- know the customer's name by his/her login, for a sales Solution
- know the customer's address by his/her login, for a billing Solution

- know the customer's account by his/her login for a customer payment management Solution

we can then set up a single exchange Service that, from the customer login, provides all 3 pieces of information: it is then up to everyone to only use what he/she needs. The challenge is finding the right compromise between reducing the amount of exchange types and the increasing heaviness of each exchange.

3. Tool making exchanges so that each build of an exchange Service is quicker.

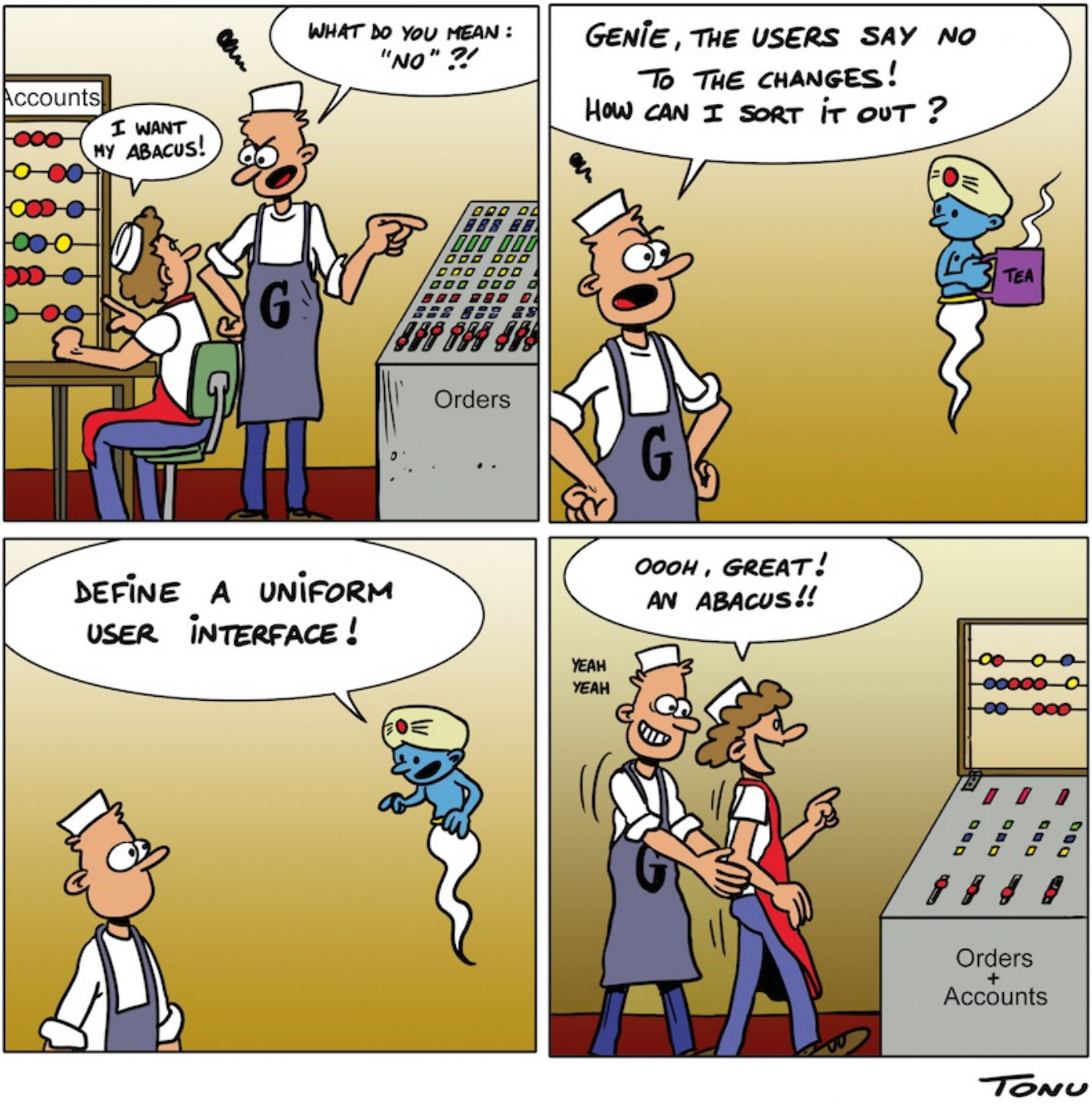
# 5. An overall vision is necessary

The most important thing is to deal with the exchanges as a coherent whole and not as continuous additions of exchange formats as the Solutions are set up.
The right approach consists in:

- Defining the Enterprise data Model
- Defining the Solutions map
- From there, deducing the list of exchange Services.

*See the white paper from CEISAR*

## Define a simple and uniform use




# 1. The Usability Value becomes essential

The user admires what is complex, but likes what he/she understands: it is what guides his/her purchase motivations. Ease of use is one of the keys to success.

We remember the survey that showed that 50% of video-recorder users only knew how to play the cassettes and did not know how to record on them! The many buttons and functions provided by the manufacturers to justify the more expensive prices made no sense.
On the other hand, the design and ease of use of a device like the iPhone was the main reason for its success. This is also an opportunity to show that being rich in functionality does not necessarily mean having a complex user interface.

The automobile manufacturers are thinking about new interfaces that will replace the many

buttons that can be found on the dashboard. Several possibilities are being examined: go down a menu to choose a specific item (radio volume, temperature, indicators...) with a simple hand movement in front of a screen, thanks to a gesture recognition system, or even by using cameras to detect the driver's eye movements. Advantage: being less distracted enables us to concentrate on the driving.

## 2. Uniform usability facilitates acceptance

Uniform Usability has profound consequences on the acceptance of new Solutions by the users. Learning one way of Using something takes time: some say that you have to forget a functionality 7 times in order to assimilate it. If new Solutions are provided that respect the same usability as previous Solutions, then the Actors are on known ground: they will like the new Solution because they know how to discover and navigate their way through it; the difficulty is transformed into a pleasure. The Enterprise gains a more flexible organization as it becomes far simpler to change where we assign the Actors.

## 3. How do we obtain uniform usability?

There are several approaches:

- Use **Solutions** that are **broad in scope**: the same development team, whether it be in-house or external (software package or Cloud) provides different functionalities with the same user interface. It is therefore in our interests to seek Solutions offering the maximum of functionalities. If the development is in-house, we must make sure that the development team has established its standards properly (that can be based on well-established standards) and that they ensure they are respected within the team; the best method is to carry out ergonomic reviews during the development.
- Provide **Components** to Solution builders who generate standardized user interfaces, whether for the "look and feel" or the navigation.
- Choose Solutions where the "user interface" is **isolated** so that it can be personalized.

# Make different Products with the same Components


The automobile industry is a good example of reusing Components to design new car Models (see the example of the Volkswagen platforms).

The gains are considerable:

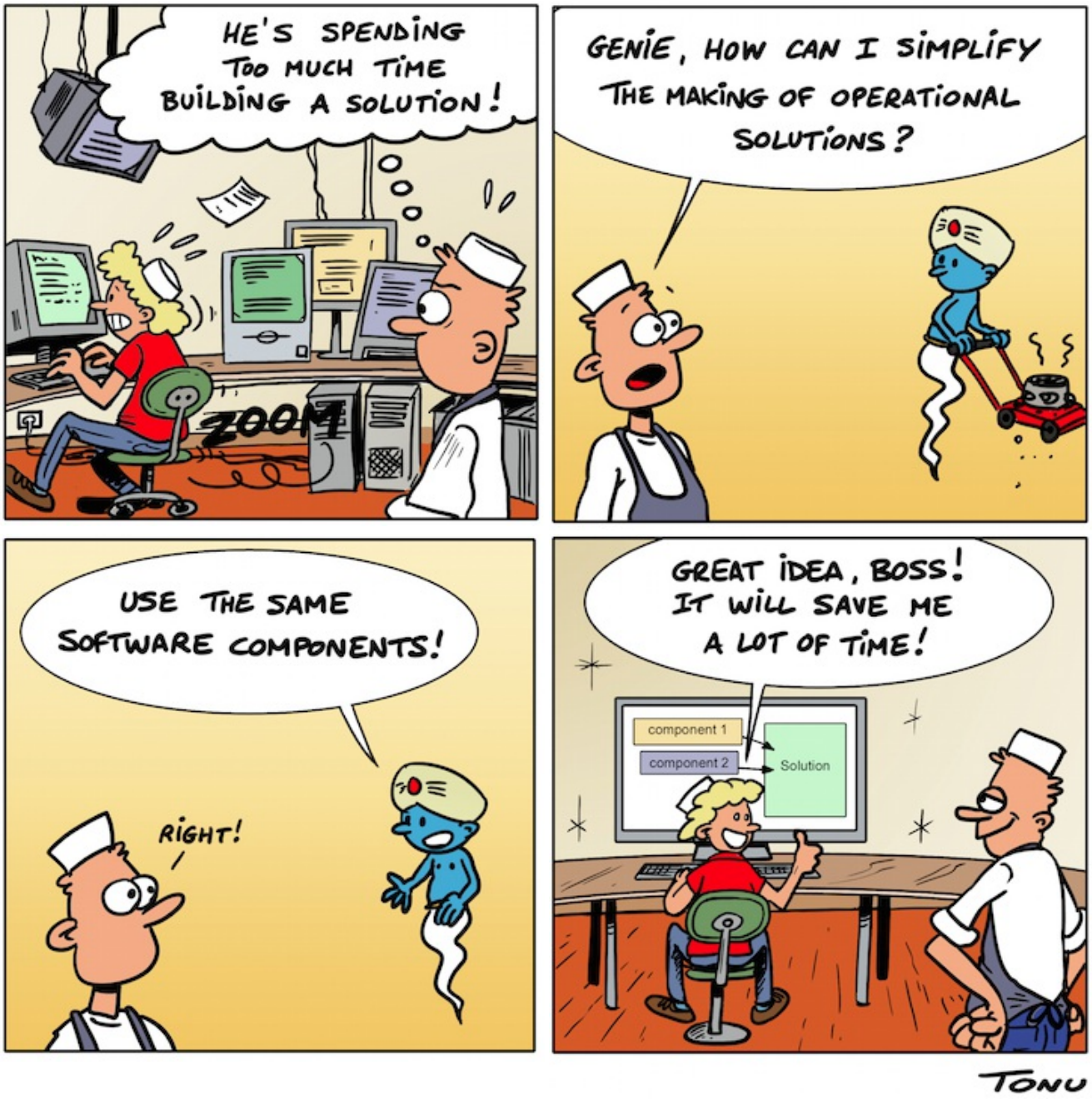
- Time and cost savings on the design of new Models
- Better reliability as the majority of the components of the new vehicles have already been tried and tested
- Economies of scale linked to reuse: the same components are reused on many vehicle models.

But this approach requires strong management. Designers of new Models seek to differentiate themselves and are not naturally led to reuse what already exists for the others.

It is therefore important to provide the means and power to those who build and distribute the components.

We find the same approach in the **Operation Model**.

## Make different Operational Solutions with the same Components




# 1. What advantages are there in reusing components to build a new Solution?

To become more agile in building Solutions, we need to reuse in order to have less to build. There are 2 forms of reuse: reusing Software packages or reusing Components .

Reusing Software packages has had increased success over the last few years for **Commodity Solutions**, for which the needs are similar between Enterprises.

But an economic activity only reaches maturity when it is capable of reusing common components to build **Business Solutions**: it took 200 years for Industry to arrive at its current maturity. The

time taken to build a new automobile Model has been reduced from 5 years to 18 months by reusing Components.

It will take a certain amount of time for the Software industry to do the same. But we can have great hope. Trends like "**SOA**", "reusable Component", "Object approach" are all heading in this way, and the results obtained in a certain number of Solution Models prove that a 70% reuse rate are realistic, that is to say that we **only have 30% of the Model to Build** to satisfy a specific requirement.

Software package vendors are themselves going through this dramatic change : the new Software package offers Built are often Component-based.

As with Product modeling, we find the same advantages:

- Time and cost savings on the design of new Solution Models
- Better reliability as the majority of the components have already been tried and tested
- Uniform usability which makes life easier for the users.

# 2. Do not confuse Architecture and Components

Architecture and Components both contribute to organizing and sorting things out.
But they go about it in two ways:

- Architecture provides an overall vision of the Model that the different Solutions are part of
- Components are reusable Modules that we can assemble to build specific Solutions.

# 3. How to build components

It is more difficult to Build Components than a Solution: Components must reuse Components, they must be versioned , documented and must satisfy extremely diverse needs.

If we do not have any experience, it is important to begin modestly, know from the offset that we will have to iterate and redevelop some components, but do not give up on the idea of a high reuse rate. (*See the CEISAR white paper on the Foundation).*

It is alone not enough to build good components, they must also be easily accessible and comprehensible.

# 4. Acquire components

We can acquire a bank of components or make them ourselves.
If we want to avoid the time needed for maturation, we can purchase a component framework from outside then adapt it to our context.

One of the most efficient scenarios, if we use a Software Package as the central Solution of our Information Systems, is when we purchase components from the vendor: interfacing work will be easier and the use will be standardized. This option is limited to the software package vendor's goodwill as they must accept to supply the components that are used to build their own Solutions.

# 5. Conditions for success

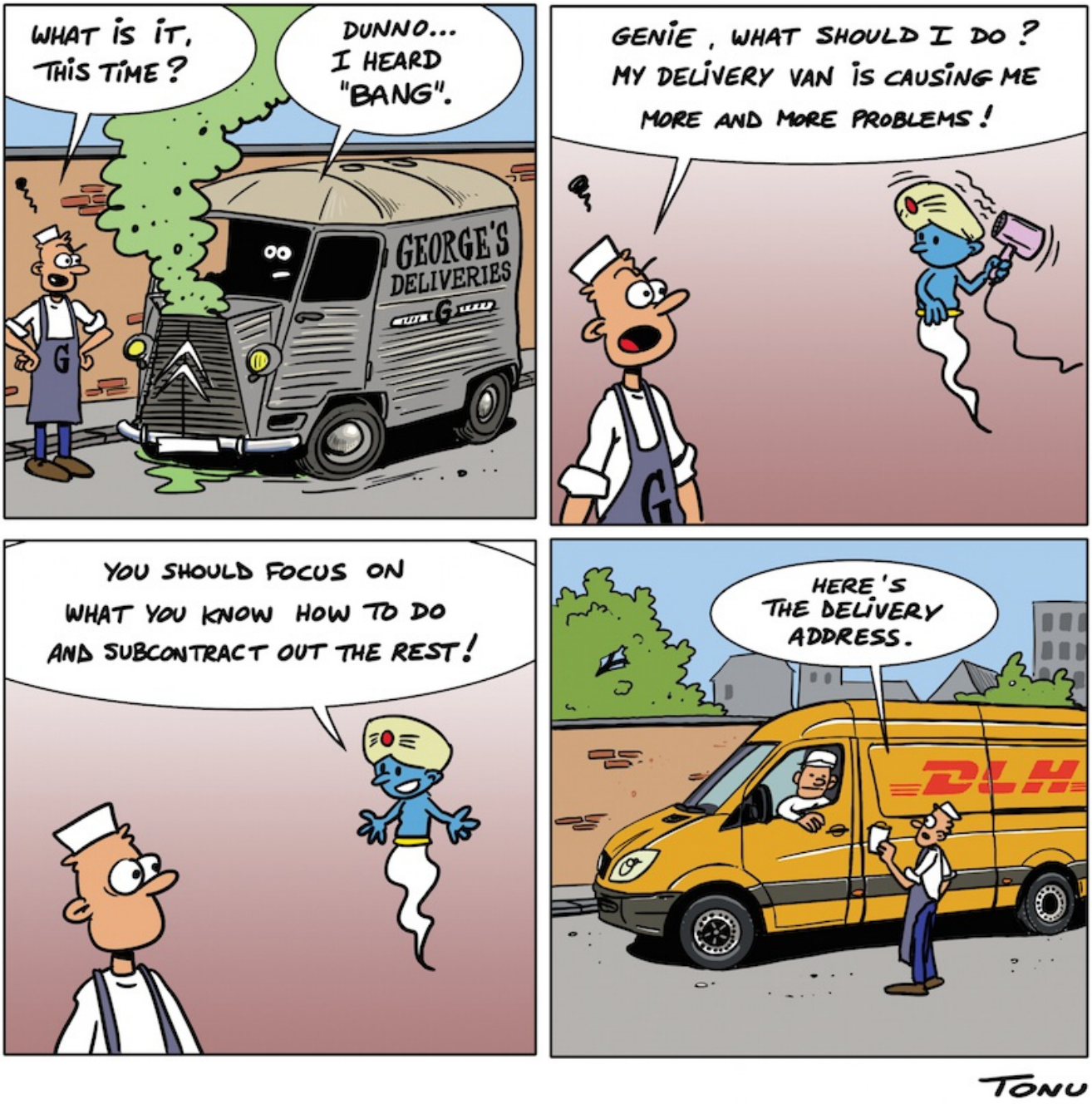
Successfully reusing Components en masse requires the conditions for success:

- **Isolate** in a "Foundation" team those that build, support and recover the Components.
- Ensure that they have gained the **know-how** to build components: interface quality, structure of small components that are reused by each other and not a flat list of big components, versioning, in-depth use of Patterns, forward compatibility, suitable configuration management...
- Encourage and control **reuse** in the Solution teams.

*See the CEISAR white paper on the Foundation*

## Act 5: Extend one's enterprise to partners

## Concentrate on one's strong points




An Enterprise does not do everything alone. It is part of a Value chain. It purchases Goods from other Enterprises to use them in the Production.
It also acquires Information.
Finally, it purchases Services which can replace the Processes that it would have executed itself.
In all cases, we can use the terms supplier, partner or outsourcing.
The real challenge is knowing how to focus on one's strong points, even if we think we know how to carry out the other activities.

# 1. Outsourcing Processes?

To manage to Offer a Product to one's Customers, a cascade of Processes is necessary.

An Enterprise can choose to carry out all these Processes itself; this is an integrated Enterprise.

It can also choose to concentrate on what it knows how to do best and **outsource some Processes** to partners: consulting assignment, cleaning buildings, transporting Goods...

The reason behind outsourcing is either a **better value** (e.g., turning to a consulting company) or a **lower price** (e.g., turning to a cleaning company).

In today's economic world, an Enterprise is never fully integrated: the Baker does not farm the fields to grow the wheat to get flour himself, he does not build the tractors needed for the fields and he does not produce the diesel needed to make the tractor run.

This means that the Enterprise has to purchase Products or Services from other Enterprises:

- Intermediary **Goods** which will serve to assemble the Goods Offered by the Enterprise: spare parts, tools, means of transport, supplies, premises...
- **Information** needed to carry out the Processes that it executes itself: Models (such as Software) or Facts (prospect files, market data)
- **Services** when the expected result is a change in the state of the Goods (cleaning premises, machine maintenance) or People from the Enterprise (training, coaching, consulting). In this case, the Enterprise no longer wants to control the Processes, it is happy to benefit from their results.

We can outsource **Operational Processes**.

- Outsourcing Production.
  - e.g., outsourcing production to offshore regions (factory for Goods or BPO for Service)
- Outsourcing Distribution
  - e.g., partnership with Distribution network
- Outsourcing Resources management:
  - e.g., cleaning premises
  - e.g., operating IT (Cloud or other)

On the other hand, we do not outsource Human resources management or the enterprise management Processes.

We can outsource the **Transformation Processes**.

- Outsourcing Product Modeling
  - e.g., we use Product Models that are already available, if our business is Distribution
- Outsourcing Operation Modeling
  - e.g., we subcontract Distribution and Production to only keep the Product design, in this case we sell Product licenses
  - e.g., we purchase an IT Operation Model (Software package)

# 2. Outsourcing Resources

The Enterprise can retain responsibility for executing the Process but involve **external Resources** that do not belong to it: renting premises, using temporary workers, get Components, purchases IT resources on the Cloud,...

As regards the Processes that the Enterprise would like to control, it needs Human-Actors. These Actors can be Enterprise employees or external Actors: but the responsibility falls to the Enterprise. It keeps the responsibility for its Process even if it outsources the Resource.

As an example, external Actors can be **customers** who carry out part of the Processes helped by an IT Solution (Operation Model) supplied by the Enterprise. In this case, the Enterprise does retain control of its Process, even if it outsources the Resources.

# 3. What advice?

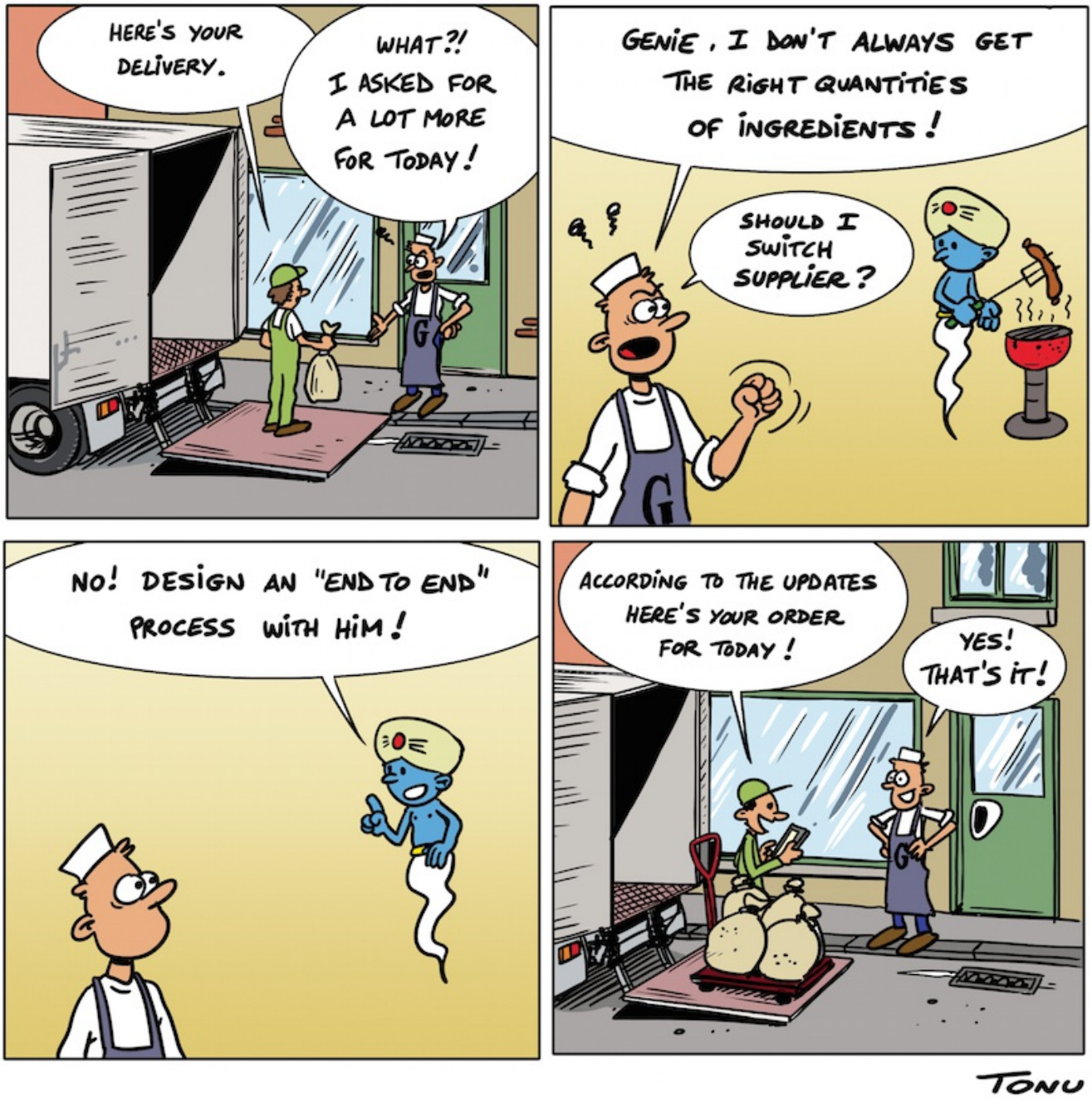
An enterprise cannot be the best in every domain.

Choices must be made.
The criteria are the following:

- Keep the Process that we know how to execute better than the other Enterprises
  - Either because we have **more productive Resources:** labor costs, motivation, know-how, work time
  - Or because we have **better Models.** We can compensate for lower Resource productivity by a more efficient Model. This is one incentive to control Product and Operation Modeling internally and know how to protect them.
- Ensure that we can adapt the Process Resources kept in case of a turnaround in the market. It is obvious that involving external Actors (temporary workers, customers, partners...) provides more flexibility to Enterprises in countries where employment law is too restrictive.

In all cases, the enterprise must keep full RESPONSIBILITY for its Products and Services even when it subcontracts them. It has to be answerable for the quality and security of its Products and Services, even if all or part of them are subcontracted.
In the same way, the Enterprise is responsible for protecting its customer data, even if it is in third-party systems.

# Model end-to-end Processes




## 1. Modeling Process aimed at external Actors

Enterprises began by Modeling internal Processes executed by the own employees. As these processes have today spread to the outside world, they also have to model Processes aimed at suppliers, partners or customers.

One of the difficulties is providing a uniform user interface by user type. To encourage the Customer to use the distribution Solutions (e.g., order) and maintenance Solution (e.g., incident management) that the Enterprise provides, the use of both Solutions must be the same. Yet the extensions of the internal Solutions towards the Customers often generate as many user interfaces as original Solutions.

If we do not want to totally overhaul the internal Solutions to adapt them to the Customers, we

have to have allowed for the internal Solutions Architecture to clearly **isolate the user interface**.
The quality level must also be excellent.

- If the internal Solutions are of an average quality, the employee will be irritated but will not resign.
- If the external Solutions are of an average quality, the consequences can be far more serious: we can lose Customers or Partners quickly.
- Moreover, the increase in the number of users will only make managing the hotline and customer service more costly.

## 2. Interface the external Processes with the internal Processes

Partners use their own Solutions that we have to coexist with.
For example, an insurance broker sells the contract but must then transfer the information to the company that takes on the risk. If he/she carries out part of the claims management, the same constraint exists.
To help these Solutions to communicate, we need inter-professional exchange **standards** that guide each intervening party.
We also need everyone to provide an **IT Services** exchange **catalog** which enables partners to connect to each other.

# Connect mobiles


## 1. The spread of Mobiles

The spread of Mobiles is exponential today. It enables their user to access a set of available functions anytime, anywhere that were previously only available from a landline.

The rhythm of use is growing more quickly than the number of connected users. To give a concrete example, BNP noticed that, over a 30-month period, the number of connected customers had risen by 15% but that the number of connections had grown by 60%.

The variety of Mobiles has led Enterprises to understand the use habits.
For example, "Voyages SNCF" noticed that:

- Smartphones are used by young people, particularly in the morning
- Tablets are used by an urban and educated population, mostly in the evening

- PCs or laptops are used during office hours, most probably at the workplace.

## 2. The bring-your-own-device (BYOD) phenomenon

Some employees are victims of the "Sunday-evening, Monday-morning" syndrome: Sunday evening, they use their modern mobile that they are used to. Monday morning, they find themselves in front of their old PC at work and struggle to accept it, hence the desire to be able to use their mobile within the Enterprise. Integrating these mobiles into the Enterprise information system then poses problems linked to security, compatibility with existing applications, ergonomic standards...

To control this integration, some enterprises have preferred to give their employees new mobiles that they can also use for personal needs: it is then easier for the Enterprise to manage the security or updates of applications.

## 3. The "no contact required"

The "no contact required mobiles", which respect the Near Field Communication (NFC) standard, enable us today to pay for our purchases, pay for the car park, buy and validate transport tickets, use loyalty points or even read tags to get practical information. You just have to choose an NFC-compatible phone: estimated to represent 50% of the market at the end of 2013.

## 4. Connected Objects

Equipped with a chip or a sensor, all the objects around us are empowered to be connected and produce data.

These are not only Mobiles (Smartphones and Tablets), but also:

- Interactive kiosks made available by enterprises
- Digital stickers
- Various sensors
- Connected Vehicles
- Digital glasses, watches...

Les Echos - L'internet des objets (article in French: "The Internet of things")

*In 2020, the number of connected objects could reach 80 billion worldwide, according to Idate, a think tank specialized in the digital economy...*

## 5. Not only increased comfort but also a change of Model

The new connected objects often lead to a profound Transformation of the Enterprise Model. It is not just a question of computerizing existing processes (e.g., direct sales on a smartphone versus sales in a physical network). Whole sections of the enterprise are brought into question by these new technologies. As an example, remote expertise in the insurance industry deeply modifies the expert's role.

## Act 6: New Value Proposition: Distributing sandwiches via the Internet

## Imagine a new Value proposition


# 1. Usage Value and Usability Value

To avoid any misinterpretations, we propose distinguishing Usage Value from Usability Value:

- The **Usage Value** is the value obtained by basic Functions of a Product: access emails via one's telephone is a new Usage Value
- The **Usability Value** is attached to how easy it is to Use the Product: if the user email interface is complex, the Usage Value is positive, but its Usability Value is weak.

# 2. Why look for a new Value proposition?

Digital deeply transforms the Offers proposed to the Customer. The Values that the Customers expect, whether they are Individuals or Enterprises, are stable; but the way of satisfying them is changing fast. It is no longer enough to copy and improve, but to innovate deeply by proposing

disruptive Products/Services. "It is not by improving the candle that we created the electric light bulb".
We will give a few short examples. We can each add to this list according to our own observations.

## 2.1 Information Products are in danger

All enterprise models based on the Information Product are in danger today: new models are appearing.

The music sector has been dramatically changed by iTunes and its successors. The global turnover for music went from 27 billion dollars to 22 billion between 2000 and 2012. Spotify uses a subscription-based model, but is testing a new model based on free music and advertising.

The information sector is currently in crisis: press, television are confronted by the fact that young people do not consume information in the same way.

The publishing sector is also in crisis: more than half of the books in the USA are today read on tablets; bookshop chains (Borders, Barnes and Nobles) are disappearing.

Google uses a camera with a one gigapixel resolution to digitalize works of art and make them accessible to everyone with a degree of precision unavailable in the museums: what consequences will that have in the museums?

Two sectors have not yet been impacted as greatly as they have more complex Models: education and health. But we can expect dramatic changes.

## 2.2 Digital will dramatically change education

- The extremely fast dissemination of **Moocs** ("Massive Open Online Courses") started in the United States in 2011 when Stanford university opened its first online lessons on artificial intelligence. The lessons are distributed free of charge on the Internet. Students can follow the lessons of the best teachers and obtain a certificate (payment required) at the end of the course. The learning is free, but the diploma has to be paid for. The consequence for the universities is a new revenue source in certification and an additional means of selection to identify the best students who they can then invite to their prestigious university. Lectures will gradually be done away with: why go to a lecture hall when you can stay at home and access, free of charge, when you want, the lessons from the best teachers? Teachers move towards question-and-answer sessions, practical work, case studies, in short, periods when the student becomes active. What remains is to find the economic Model: we will most likely have to go through the three unavoidable steps of Digital innovation. First, build the Offer and check its relevance, then generate volume and finally look for profitability, by making the students pay for the certificate at the end of the course, or through advertising, or by selling the lessons to other teaching organizations seeking to widen their offer, or developing continuing professional development courses with enterprises, or creating partnerships with education publishers... (see " Le Nouvel Economiste : La déferlante des MOOCs" Article in French: "The rush of MOOCs").
- Providing each student with a computer enables him/her to access lessons (many in math, rarer in French or history) and exercises, to access knowledge via the Internet, to work in a group... Students' digital equipment: by equipping their pupils and teachers with computers, the Quebec district of Eastern Townships halved the number of pupils who left school with no qualifications: 22% of school children left school with no qualifications compared with 42% in 2002.

## 2.3 Digital will dramatically change health

An increasing number of **remote diagnoses** can be made today: the equipment is available, efficient and connectable.

The length of hospital stays can be reduced in many cases without risk, provided that the home-care Service Offer is developed and that remote monitoring systems are properly operational. But to overhaul the Health Model, we need to make the many intervening parties agree, especially within the Civil Service who are not used to rapid Transformations. In particular, the question "who pays whom" is essential in achieving the economic balance of the system.
This evolution will take us from curative medicine to preventive medicine, and this for the benefit of all.

### 2.4 Even Goods Industries will be impacted

The new tractors will be guided by satellite: a GPS, with a precision of 2cm and not 10 meters like those in cars, enables farmers to avoid going over the same ground twice when they are working: a saving of 10% to 20% in fuel and products. Combine harvesters adapt their own speed according to the crop they are harvesting, which enables a 20% improvement in hourly output.

The "Google car" tested in California is an autonomous car, with a driver.
Who will win in this competition?

- The established manufacturers because they know how to build chassis, gearboxes, engines; because they have dealership networks and a loyal customer base?
- Or newcomers like Google, because they master information systems or location data?

Whoever wins, how will the insurance companies analyze the responsibilities?

### 2.5 Collaborative sites propose new forms of usage

Collaborative sites propose new forms of usage of which we will give 2 examples:

- Example of a collaborative site proposing rooms: **airbnb** (Nouvelobs)
  On this website, individuals offer rooms to tourists. Value for the renter: not only the room will be less expensive than a hotel, but the customer will benefit from advice given by the homeowners on restaurants or shops in the neighborhood. We are welcomed as a person and not a customer.
- Example of "**covoiturage.fr**", the market leader in Europe of carpooling. The site connects drivers and potential passengers who make the same journey in order to share costs.

Competition is becoming fierce not only for traditional actors, but also for the State, whose revenue service is impacted.

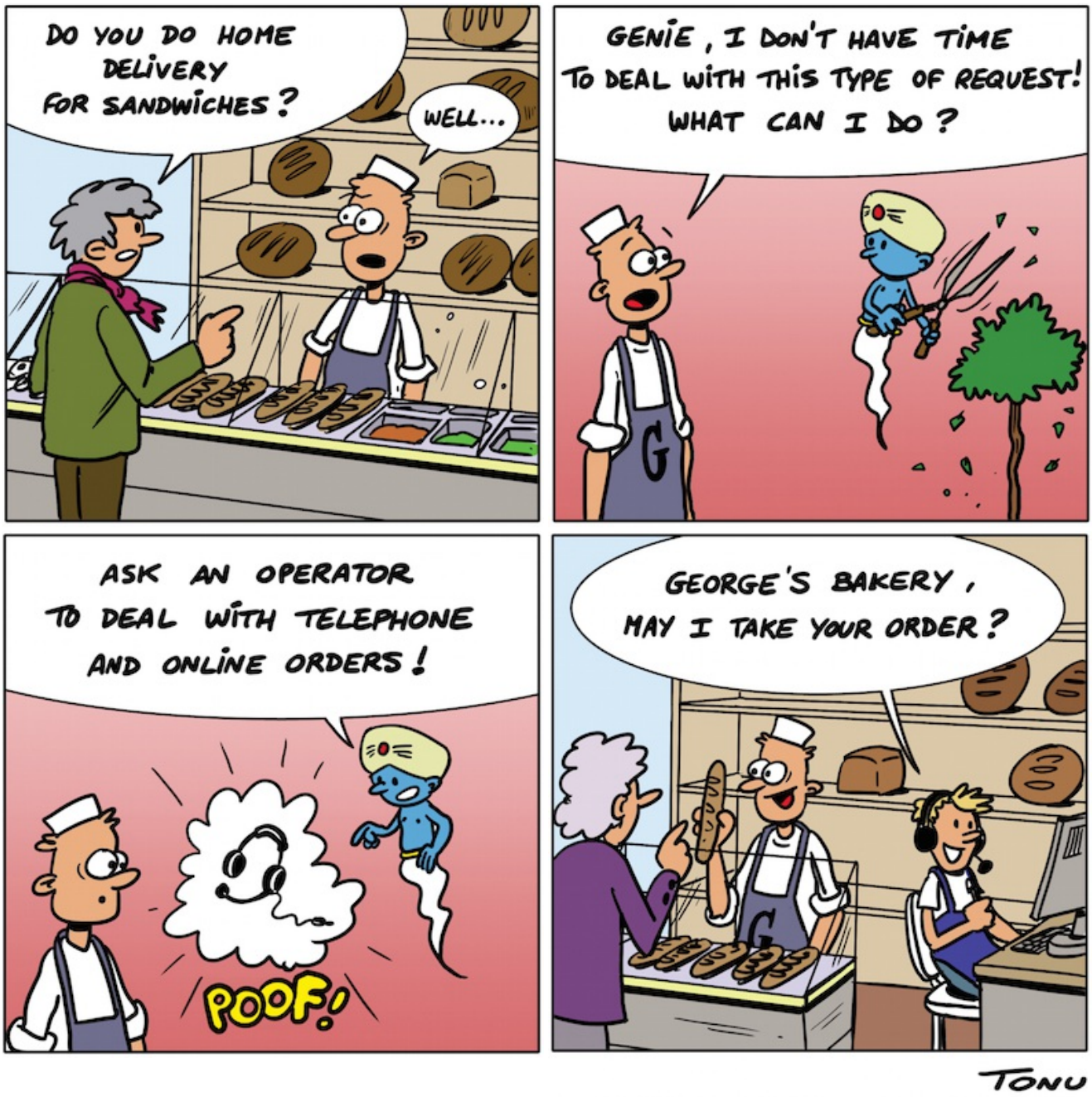
## 3. Start with the Value rather than improving a Product

As we described above, the first task is deciding what Value we want to satisfy.

In a world where **progressive** improvement is favored, the right design process is copying the best thing available, then seeking to improve it. Business monitoring is vital to identify competitors' good ideas. We begin by leveraging what has already been invented by the others, then we improve it by adding our personal touch. If we want to go through the whole learning process by ourselves, we will not be very efficient.

In a world where **disruptive** Products dominate, this approach is less efficient: it is better to start with the Value the Customer is waiting for to imagine Products that do not yet exist today. It is one way of being the first on a new Market, which represents a considerable advantage, but is not enough, as the difficulties of Yahoo on search engines or Blackberry on smartphones have shown.

# Offer a new distribution Value




## 1. The Offer Value is not the Product or Service Value: the Customer experience

Many Enterprises do not Produce: they are content to Distribute. They have understood that what is important is not controlling the Product, but **controlling the Customer**: we look to be present upstream in the Value chain, as close to the customer as possible.

Remember that the Offer Value is not only found in the Product Value, but also in:

- The **Distribution Value**
  - Good Operational marketing, whether it is advertising, a mailing campaign, or a personalized offer, is an asset to successful selling
  - The quality of the welcome, whether it is in a shop, on the phone or on the Internet is also an asset

        - Product availability
        - Transaction speed
        - Delivery and installation efficiency
    - And, of course, the **cost** and financing facilities

The "Customer Experience" refers to the set of emotions and feelings felt by a customer before, during and after buying a Product (see the Definition of the Customer experience).
This one term groups together the different themes that enable us to attract and retain the Customer because we consider today that these themes should answer a concerted strategy to:

- Identify the Customer whatever the contact point
- Understand the Customer expectations
- Understand how they view the Brand and the Products of an Enterprise
- Analyze which contact points are favored by which Customers: which channels, which means of access
- Imagine how to keep the Clientele
- Evaluate the cost of acquiring a Customer
- Be informed of each Customer's consumption to know how to react

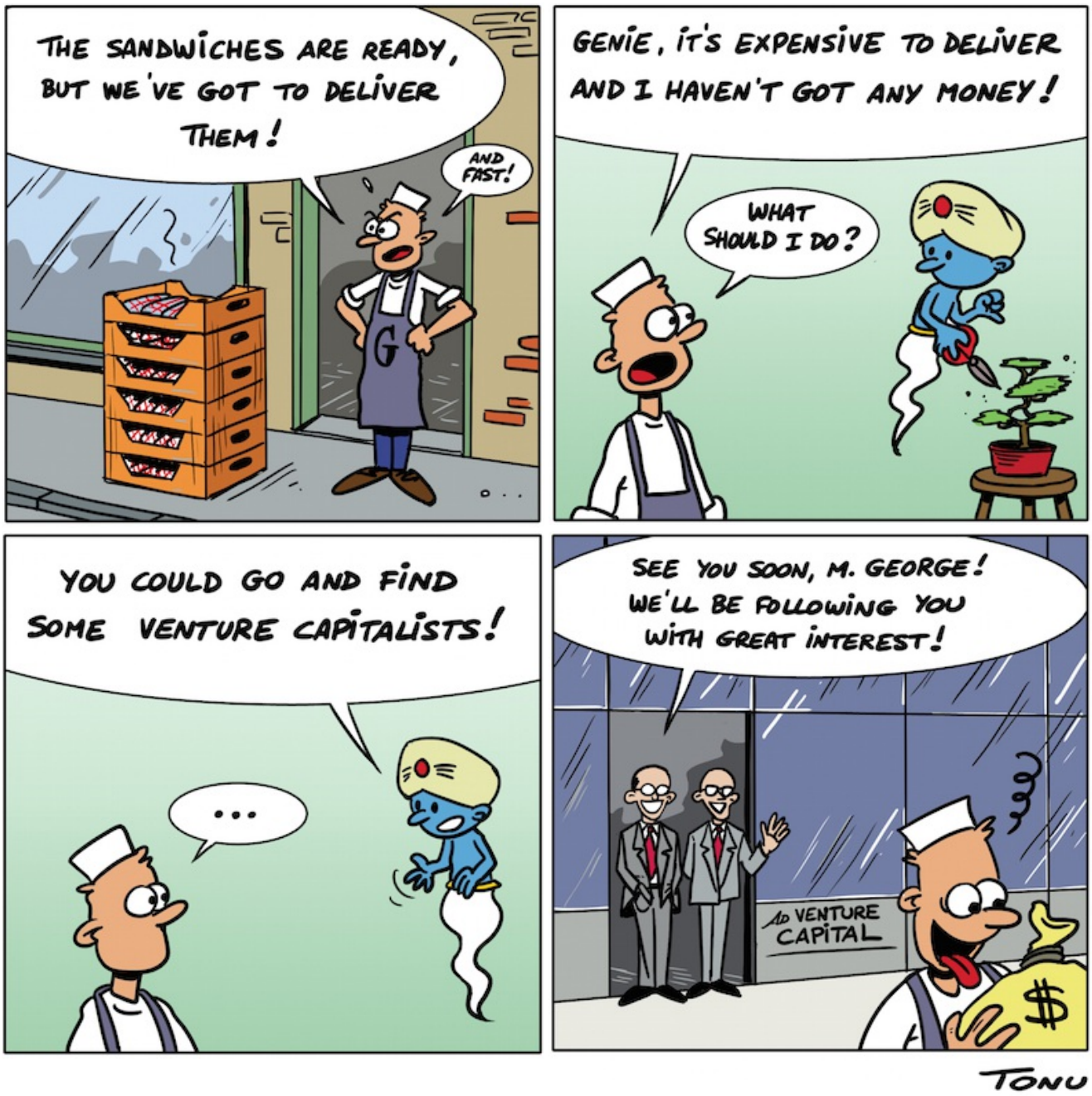
# 2. Some examples

- One interesting example is that of **Shazam** who created a Solution on mobiles to identify a piece of music from listening to it. When the tune has been identified, the user can buy it by simply clicking on a button. Today, 12% of music sales is through Shazam!
- **Booking.com** today has almost a 40% share of the hotel market and recovers an average margin of 20% to 30% on the sales made. It is a real threat to the profitability of the hotel industry which ensures the Production.
- **Supermarkets** have not innovated in Products, but in a new Distribution Model, which have allowed them to reach a size and negotiation power so as to reduce their Product purchase costs. Today, they evolve by proposing own-label Products.
- **Buying Products on the Internet** is a new form of distribution that reduces the classic forms of distribution by the same amount
    - Goods: practically all goods can be bought over the Internet
    - Information: music, cinema, news...
    - Services: carpooling,...

# 3. The "SOLOMO"

"Social Local Mobile" is a new form of distribution that targets Mobile users using social networks and geolocation.
The idea is to enter into a personalized relationship with customers to give them targeted information when they pass close by a sales outlet: it is a very relevant geolocated Offer.
Les Echos: Solomo (article in French)

# Venture Capital financing




## 1. After the Business Angels, the Enterprise needs Venture capitalists

The Business Angel can help at the start, but does not have sufficient means to support phase 2, that of growth or a costly research and development phase. To give a rough estimate, financing needs are then above one million euros.
This is where the **Venture Capitalists** come in, who have more means at their disposal than the Business Angels. They invest amounts ranging from 1 to 10 million euros over 3 to 5 years to finance **high-growth**, innovative Enterprises. They are no longer individuals, like the Business Angels, but specialist management companies who use investment fund resources dedicated to this category of Enterprise.

The **Corporate Venture** enables large groups to invest as minority shareholders in small

companies that are growing. It is one way for these groups to carry out strategic monitoring, to become aware of innovative procedures and to integrate a new entrepreneurial dimension into the Enterprise Culture.

## 2. Short- or long-term Approach?

In the digital world, we look to create volume before making the Model profitable: a short-term profitability approach does not make sense. **Operational** Actions such as "control Product quality" or "control customer satisfaction" have a price. Getting rid of these expenses increases margins in the short term, but compromises the future.

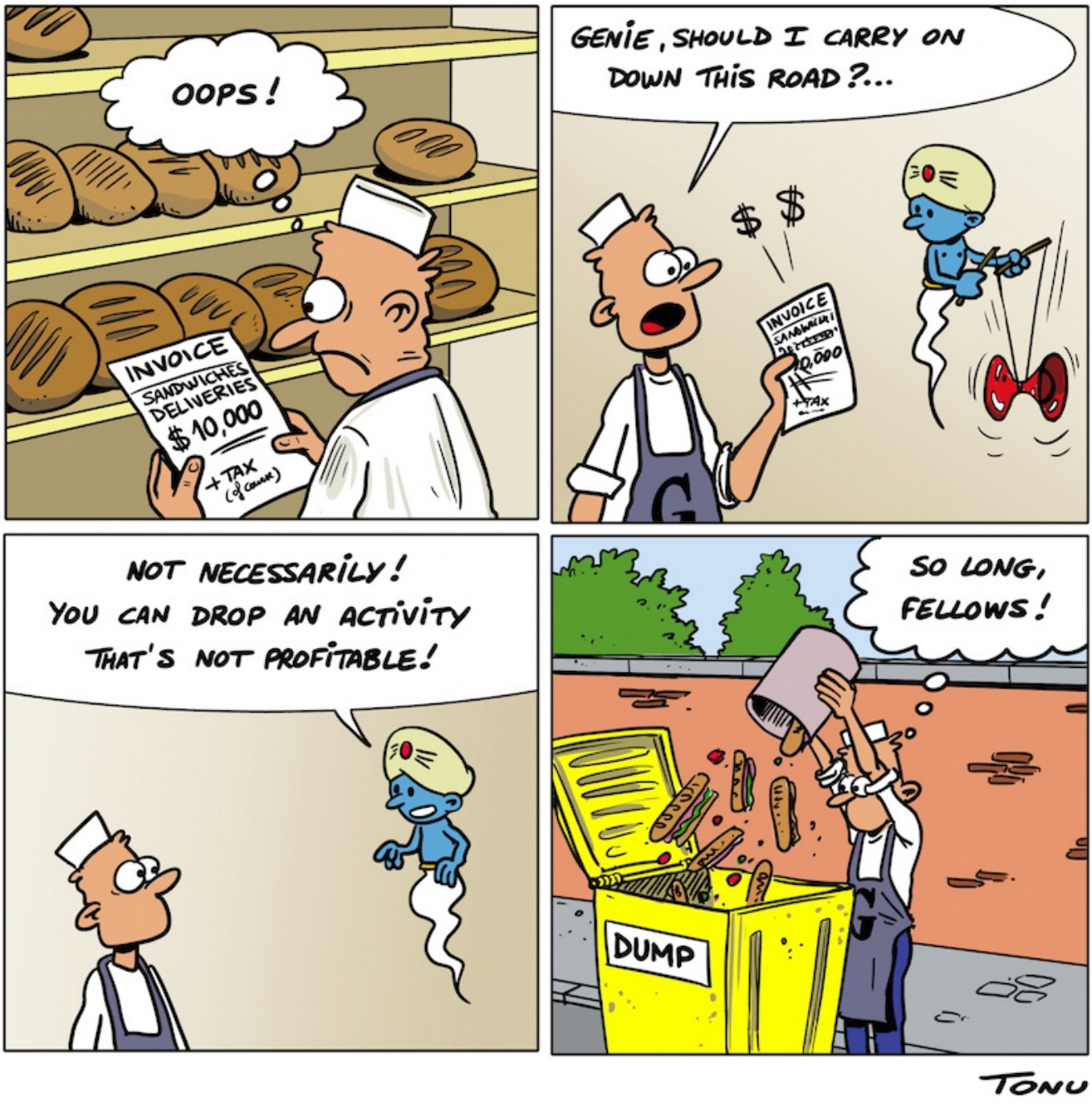
However, some complain that enterprises are often only managed according to quarterly results. Quarterly results do make sense if they are part of a long-term budget framework which includes these investment expenses.

We therefore have to choose Venture Capitalists who have a **long-term vision** and accept to adapt the strategy as the enterprise progresses in its Market: it is important that they are not just financiers, but that they have a vision of what the Enterprise start-up can achieve in the long term.

## 3. How to present your Enterprise to attract investors?

- Make it **simple** : they can only like what they understand.
- Show how your Product gives your Enterprise a **competitive advantage**: which differentiators?
- Highlight the **quality of the management team** : this is the most important criteria for investors.
- Develop a financial plan: it is rarely realized as planned later on, but it brings a Model to build upon for the future
- Avoid The Top Ten Lies of Entrepreneurs by Guy Kawasaki
    - "Our projections are conservative."
    - "(Big name research firm) says our market will be $50 billion in 4 years."
    - "(Big name company) is going to sign our purchase order next week."
    - "Key employees are set to join us as soon as we get funded."
    - "No one is doing what we're doing."
    - "No one can do what we're doing."
    - "Hurry because several other venture capital firms are interested."
    - "Oracle is too big/dumb/slow to be a threat."
    - "We have a proven management team."
    - "Patents make our product defensible."
    - "All we have to do is get 1% of the market."

# Abandon what doesn't work




If an Enterprise is in deficit but there is hope that it can turn things around, we can ask the financial sector to support the Enterprise during its convalescence phase.

But should we save enterprises who cannot become profitable again to save jobs?
But should a Group keep some loss-making activities because it is making profits elsewhere?

These are topical questions that give rise to controversy.

Yet the answer is obvious: financing loss-making activities reduces by as much the ability to invest in activities of the future and jeopardizes the enterprise.

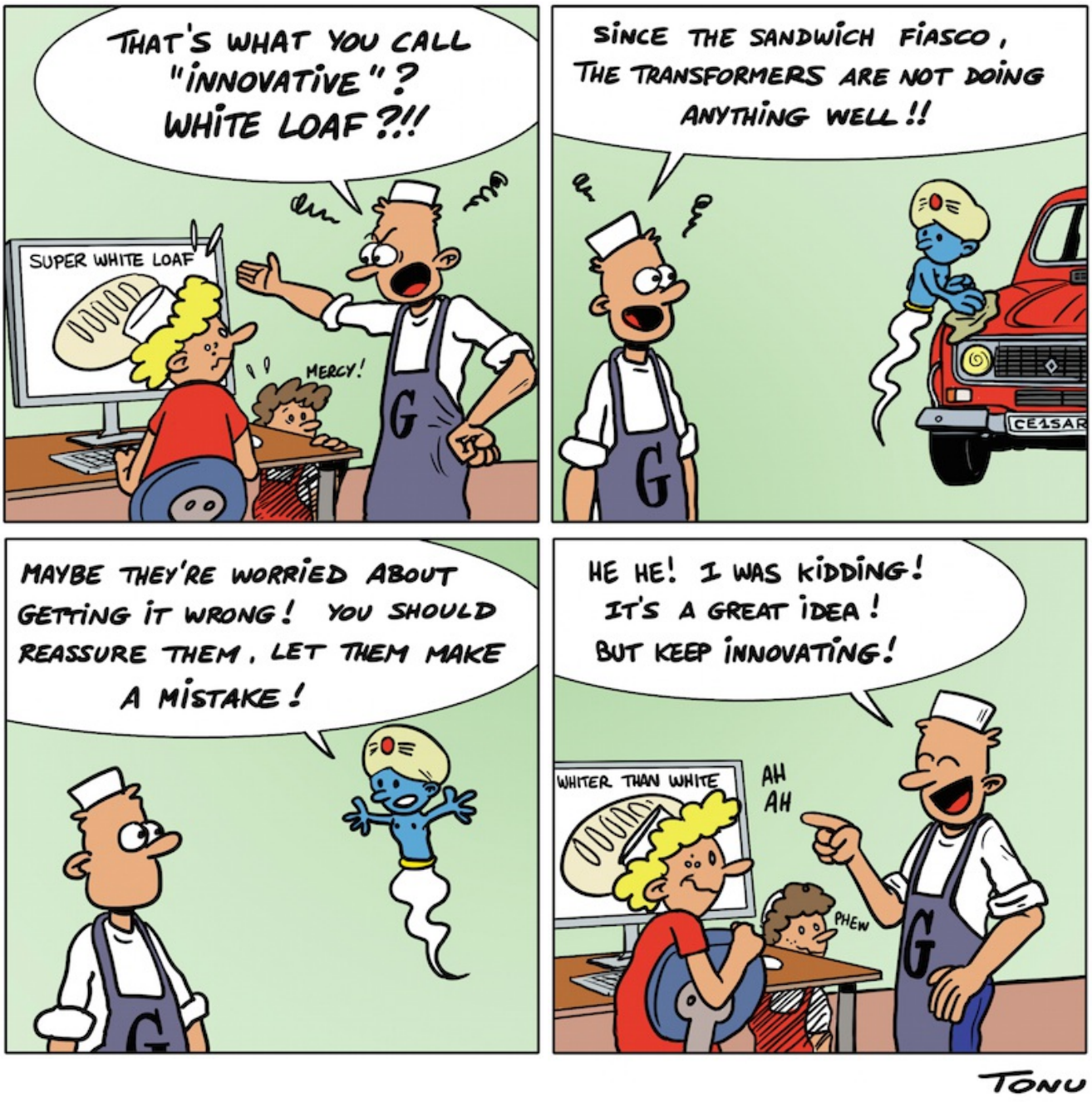
Rather than financing what does not work, it would be better to dedicate the same resource to creating future activities, training the employees concerned in order to direct them towards new jobs.

We also have to prepare people to change activities several times in their professional life: if the lifespan of an enterprise has gone from 35 years to 10 years (see the forecast from Standard and Poor's), there will be few employees who will spend their working life in the same enterprise. Job security is not linked to **belonging to a system** whose lifespan is increasingly short, it comes with the required **Competence** to perform new jobs. We have to help those who do not have this Competence to gain it. Continuing professional development is one of the keys to success in facing these changes.

In summary, we have to prepare Actors to accept the idea

- that the Enterprise has a limited lifespan,
- that they will have to change jobs several times in their working life,
- that continuing training is vital and
- that Competence is their best asset.

# Transformers are allowed to make mistakes




Americans view enterprise start-up attempts positively, even when they are not successful. The French view them negatively.

**Operational** activities are predictable: when the Model is sound, successfully Operating is within the reach of everyone who correctly applies the Model.

On the other hand, uncertainty is quite normal in the **Transformation**. It is impossible to always be right: even Steve Jobs met with failure with Lisa or with Next.

Wanting to apply the principle of precaution, seeking to implement heavy methodologies to avoid mistakes, setting up bureaucratic principles of governance,... can only stifle creativity. In practice, the better the quality of teams, the less procedures they need.

On the other hand, it is important to look at the reality in front of us and reject the "it will work

soon" syndrome when it become repetitive: we have to have the courage to give up on an initiative even if it represents a lot of investment. Nothing is worse than hearing: "as we have already spent a lot of money, we can't give up"; what is lost, is lost!

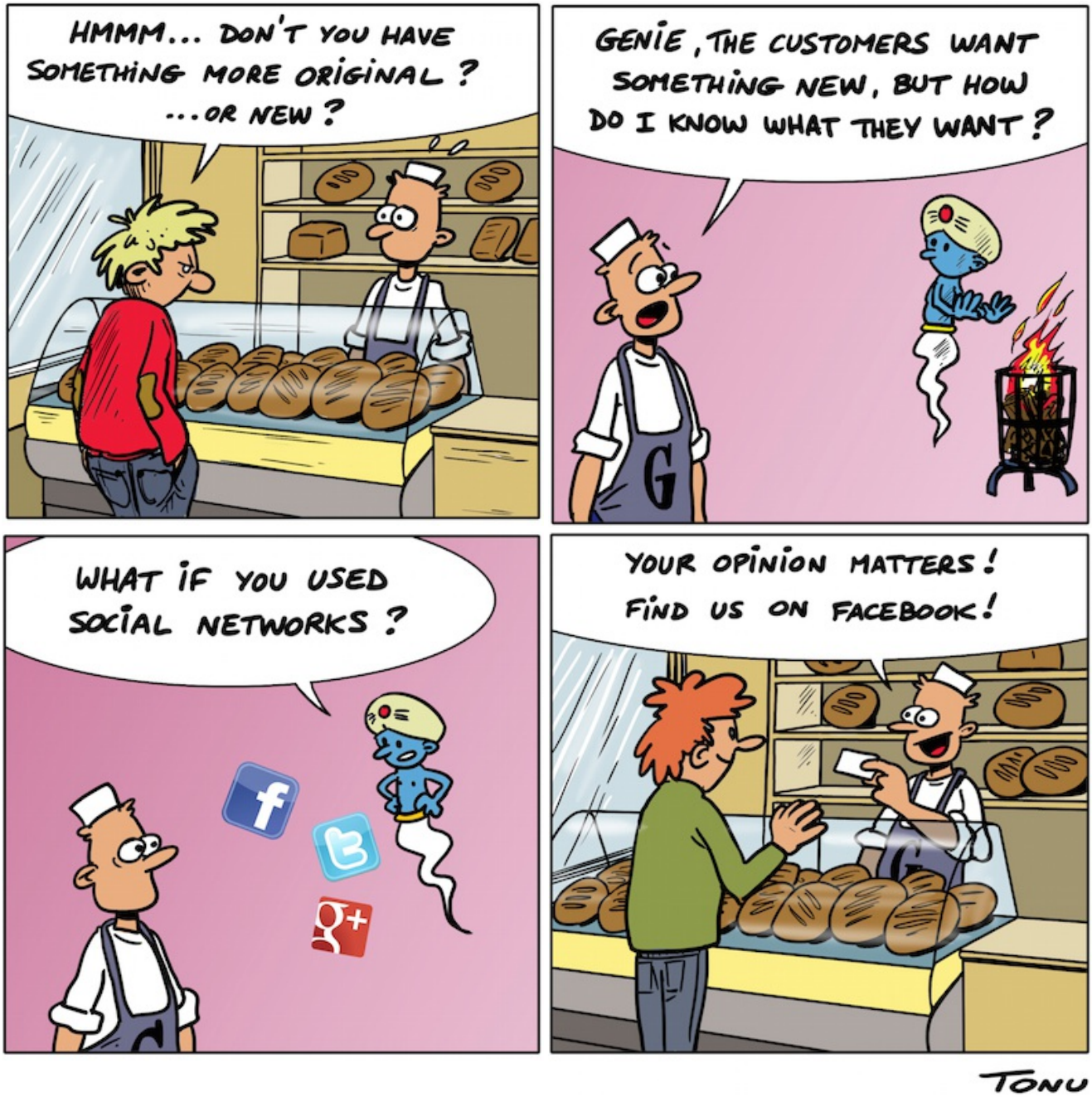
Do not discourage Transformers who have suffered failure. A good method is to ask them to analyze the reasons for this failure, then identify the measures needed to avoid making the same mistakes twice. This analysis generally gives them the courage to restart a difficult project.

To limit the consequences of possible failures, we can use approaches that enable us to test out ideas at less cost:

- Test and learn
- Proof of concept
- Lean Startup
- Sandpit

## Act 7: New Value Proposition: Selling Models

## Use social networks to know Customer expectations




# 1. Use social networks to know customer expectations

The mass of information scattered across social networks or in search engine queries can help Enterprises to better understand their customers' behavior or expectations: they use "big data" techniques which enable them to analyze this vast quantity of information.

"Big data" does not only cover the notion of huge volumes of data. It is defined by Volume, Variety and Speed.

- Variety: the data can be structured in databases, as we have always done, but it can also be unordered such as the data we collect from social networks,
- Speed: "flash" technology lets us process immense volumes of data in random access

memory (RAM) that were previously accessible only on the disk.

# 2. The other functions of social networks

Social networks are a two-way source of exchange. They are not just content with "branding". In 2013, 38% of enterprises already use them to answer a request for information or help from the Customer, or to deal with a complaint.

Some banks use data collected on social networks to analyze the credit risk. As an example, they can use rules as:

- The more friends we have, the lower the risk
- If we have a friend who is a bad debtor, we risk not being eligible for credit

Some recruitment firms also use social networks to get a better feel for the traits of the candidates.

# 3. Social networks to free up traditional email

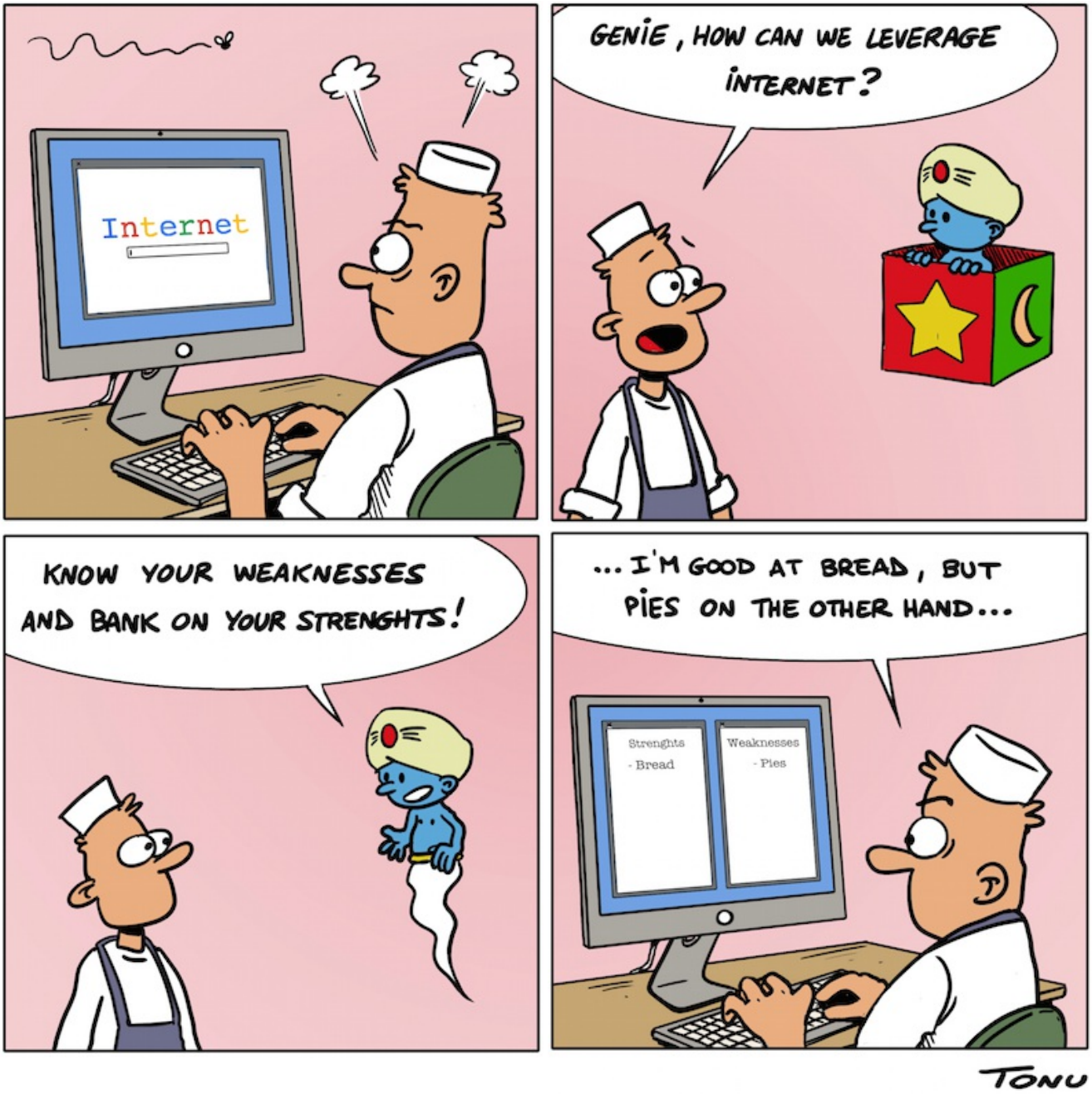
An increasing number of Enterprises have made internal social networks available to their employees or close partners. They are often used to replace email which has become a communication carryall. As we do with trash, we need to sort out and segment the usages:

- Instant messages are more the domain of Chats if we are at work or SMS text messages if we are on our mobile
- Sharing files within the same team uses more systems such as Dropbox or Skydrive
- Emails by group goes through social networks
- Personal conversations are often done over the phone (Skype or others)
- Of course, we also need to use email but it should be limited to what the other modes do not take into account.

Email is still dominant, but exchanges via social networks are growing rapidly.

Some Enterprises are even considering getting rid of email and replacing it by internal social networks.

## Know one's strengths and weaknesses




Some enterprises are pulled by the Customer, others by the Product (technology enterprises). The former first wonder about customer needs, the latter focus first on the Product and then look for a Customer... For them, know-how is key.

We cannot be the best at everything. Common sense has to lead the enterprise to position itself on markets where its qualities give it a competitive advantage and where its weaknesses will not hamper it too much. But that presupposes that we can clearly see our strengths and weaknesses.

As an Enterprise is only a Model executed by Resources, we have to analyze the whole Enterprise Model and its Resources, as they have been described earlier, to identify our strong and weak points.

To give some examples of questions:

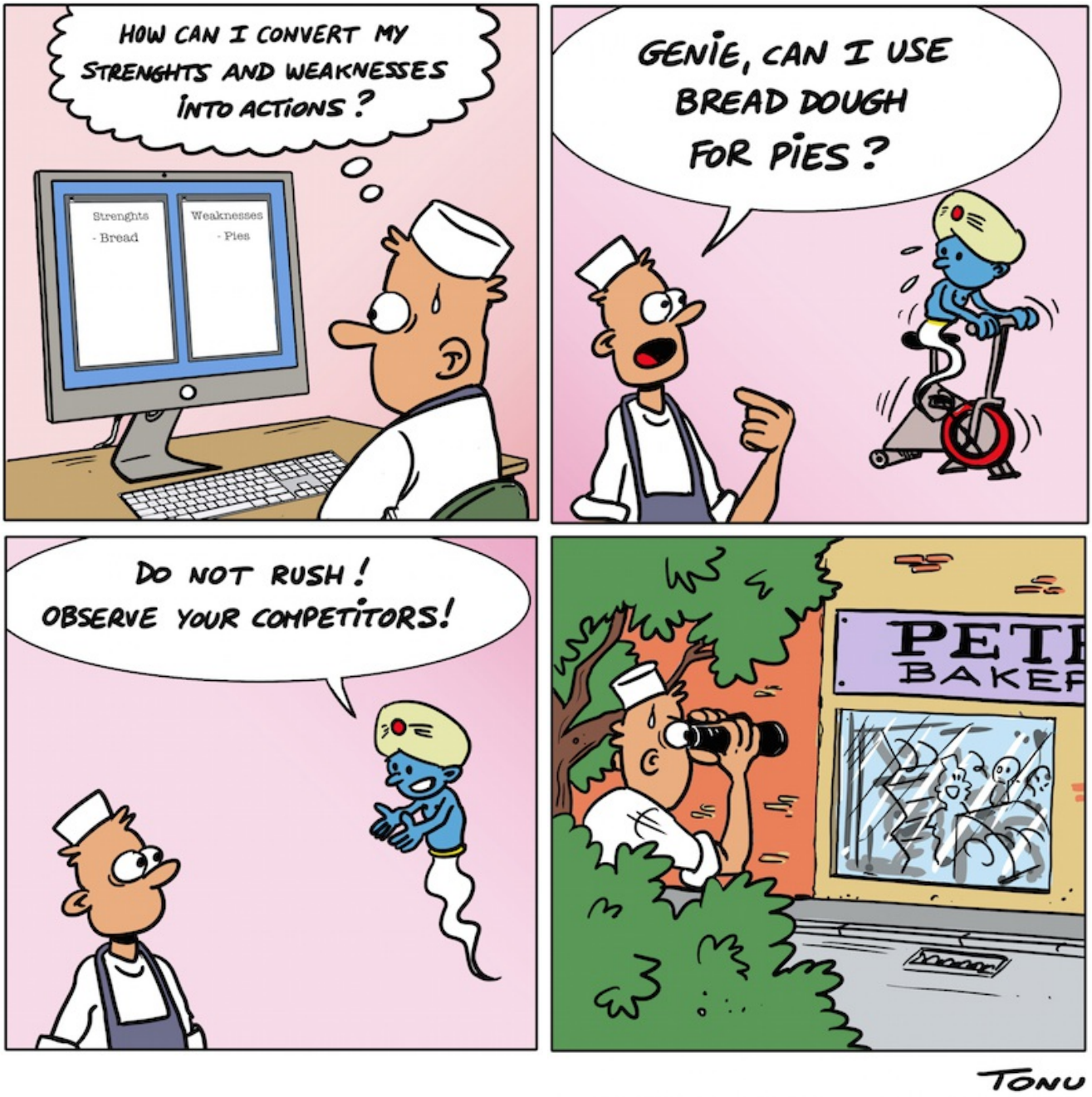
- Offer Model

        - Is the Offer adapted to new Markets, for which Values?
        - Is the Product Offer overabundant?
        - Is the Product quality good enough?
        - Is the Product Usage easy and consistent?
    - Operation Model
        - Is the Distribution Model innovative?
        - What is the Production cost compared to the competitors'?
        - What weight do the procedures have in the decisions?
    - Transformation Model
        - Is the speed of the Transformation quick enough?
        - Do we have project managers capable of managing all the dimensions of a Transformation and do we know how to retain them?
        - Is a powerful Foundation team in charge of the overall coherence?
    - Image
        - traditional or innovative?
        - Product quality or competitive prices?
    - Culture
        - Proud of our model or fear of the competition?
        - Have the staff understood that there needs to be a profound change?
        - Capable or not of taking risks?
        - Antagonisms or positive collaboration between Business and IT?
    - Human Resources
        - Is change accepted by the Operational staff?
        - Organization by project or by competence?
        - Do we have enough quality Transformers? Do we know how to retain them?
    - Information Resources
        - Do we have enough Customer information?
        - Do we have enough feedback on the level of satisfaction of Products and Services?
        - Is the management information pertinent?
    - Financial Resources
        - Can we give ourselves the means to Transform deeply?

External opinion is as important as internal opinion: customers and partners are the best source of information.

Once again, we feel that the biggest strength is the ability to reinvent ourselves quickly: what counts above all else is Enterprise Agility. On the opposite, whatever its current competitive advantage, the greatest weakness is not knowing how to change in time. Who would have believed, several years ago, that the likes of Kodak, Blackberry, Nokia, Sony, Peugeot... would have difficulties.

# Business monitoring




## 1. Know how to doubt

Self-satisfaction is the biggest danger to Enterprises and particularly to Large Enterprises.
Large groups can disappear quickly today: the Image may be excellent, the finances flourishing, the staff proud of their Enterprise, and then all these assets can crumble in several years: we all remember the examples of Kodak, Nokia, Peugeot, Surcouf,...

## 2. Carry out business monitoring

One way of remaining vigilant is to carry out business monitoring.
Start from the Value to enlarge the number of competitors: we need to do more than watch what the traditional competitors are doing, we need to also understand what the newcomers offer, who seek to bring the same Value with different Products. The competitors of a TV channel are not only

the other TV channels, but especially the usage of Internet which is leading young people to gradually turn away from classic television.

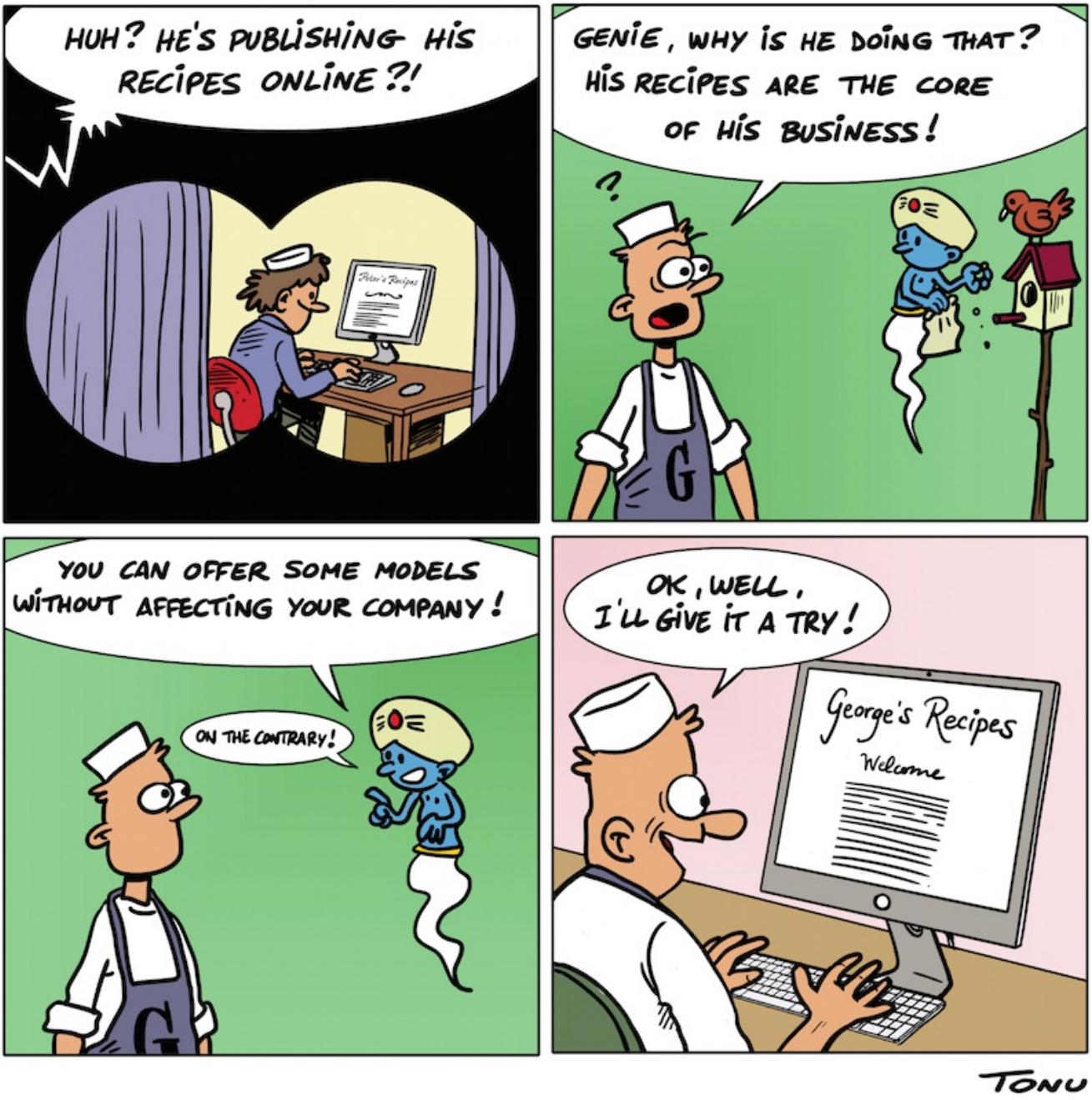
# 3. But do not hesitate to stand out from the crowd

That said, the quicker the products evolve, the more difficult it is to rely on competitive analysis to understand the Market evolution.
In markets where Product renewal is quick, we have to focus more on the **Customer** than on the **competitors** to imagine new products:

- the rhythm of change is too great for us to wait for the competitors to have developed new Offers to make decisions
- the big success stories (Google, Amazon, Apple...) were guided, first and foremost, by observing what the customer expected

One of the difficulties is to remain objective in this analysis.
To give an example, **online banking** has met with considerable success in England but not in France: is it due to a cultural difference between consumer behavior or to differences in the Offer? As the cultures are similar, we should be wary of those who blame customer behavior while their Offer is perhaps inadequate.

# Offer Models




## 1. What Enterprises distribute Models?

Products are Goods or Information or Services.
Information-Products can be of two types:

- **Facts**, mainly **Operational Information** on the Customers, contracts, accounts, Market... in short, any information that is used to Operate in the Enterprise.
- **Models**, whether they are for Human-Actors in document form, or IT-Actors in software form.

To give some examples of Model-Products:

- Roland Moreno invented the smart card in 1974: for 24 years he benefited from the royalties of his patents that are nothing more than Models

- Software vendors (Microsoft, Oracle, Sales Forces, SAP) build and distribute software that are only Models
- A franchiser (like Afflelou or McDonald's) distributes a Model to the franchisees: a brand, an Operating Model, Product and Service Models
- A pharmaceutical laboratory can Distribute the license to one of the molecules it has discovered to drug manufacturers.
- A composer can receive royalties on what he/she has created, which is nothing more than a Model "Operated" by musicians or singers.

# 2. Why buy a Model from the outside?

3 reasons have led to the development of the industry of Models:

- The **speed** of Transformations leads Enterprises to look for Models outside because they do not have time to build them.
- The **complexity** of new Models, which assemble sophisticated Product Offers and which involve external Actors (Customers and partners), make the task of internal Transformers increasingly arduous. The growing share of IT-Actors (Mobiles, PC, smart tools...) requires more and more software: good software is extremely difficult to build.
- The **breakup of activities** between different partners promotes the emergence of Enterprises specialized in Models that enable them to cooperate.

# 3. What advantages are there to Distributing Models?

### The Operations are simple

One of the characteristics of Enterprises that build and Distribute Models is that the Transformation activities take up a bigger part than the Operations.
The Model industry has the advantage that, after investment for Building the Model, the efforts to Produce it are limited to the duplication of the Models, which is only the duplication of Information. It is not by chance that Microsoft, Google, Amazon, SAP, Oracle, SalesForce.com... are so prosperous.
Imagine an innovative Model, test it out to check its validity, then Distribute this Model: your Customers will manage the Operations, you will receive royalties, or rights of use or licenses without having to do any more than maintain the software...

### Software can be modified

To benefit from new Goods (e.g., a new car which corresponds to a new Model), we need to change Goods (in this case, the car). On the other hand, to benefit from a new Model, we can content ourselves to replace the old Model with the new one: it is a new version of the same Model. We can perpetuate the customer relationship; customers will prefer to move to the next version of the installed Model rather than replacing the existing Model, so long as its supplier is able to evolve the Model so that it remains competitive.

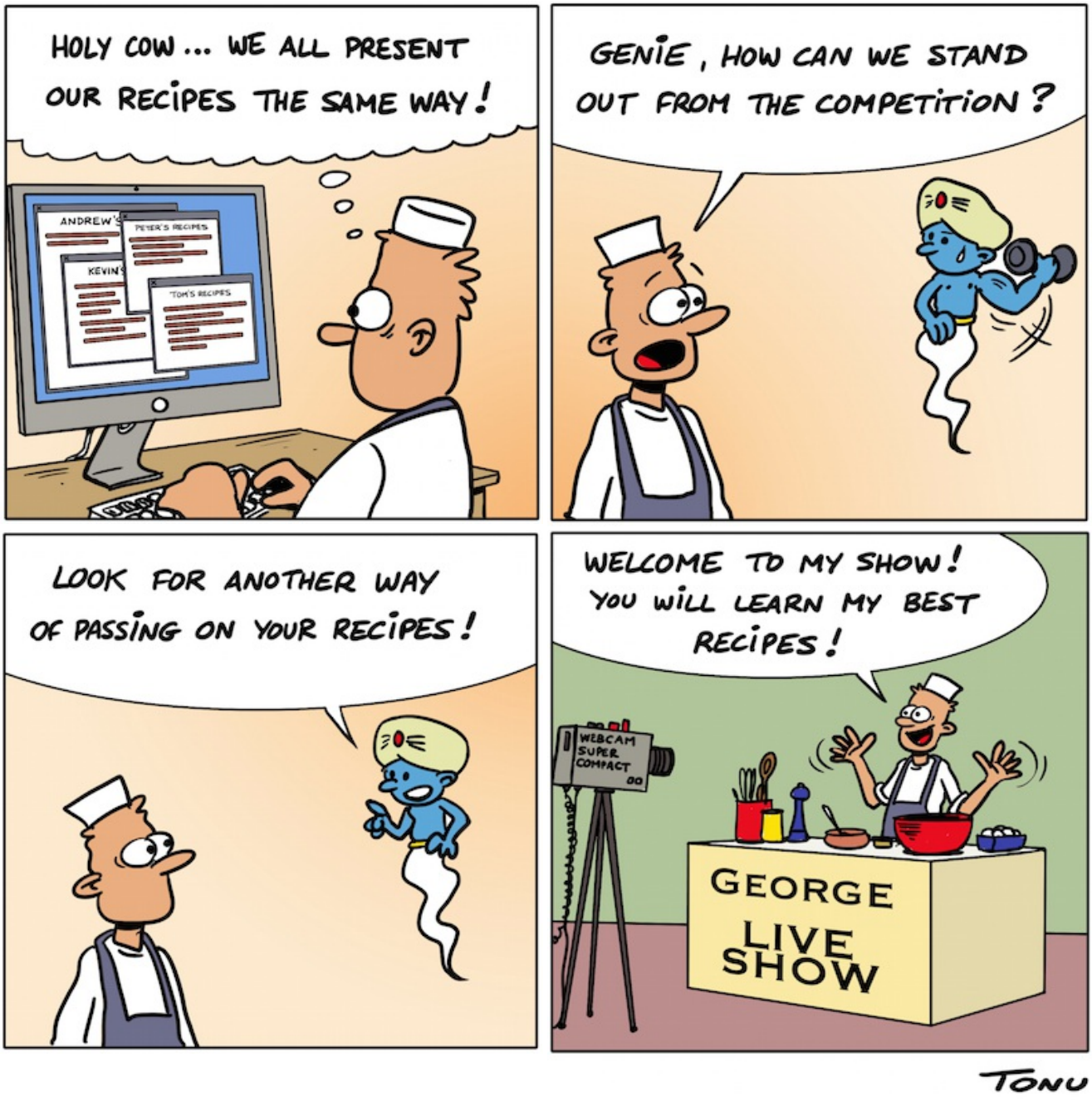
In actual fact, upgrades are not always so simple in Software Models: we may have to convert data to adapt it to a new Model, sometimes it is necessary to adapt the interfaces to other established Solutions, we have to train users to the new functionalities...

# 4. Is the Model modifiable by the Customer or not?

So that the Customer who buys a Model can benefit from future versions, he/she must not modify the Model. This is why software vendors do not deliver their Product sources, the other reason being to keep their know-how.

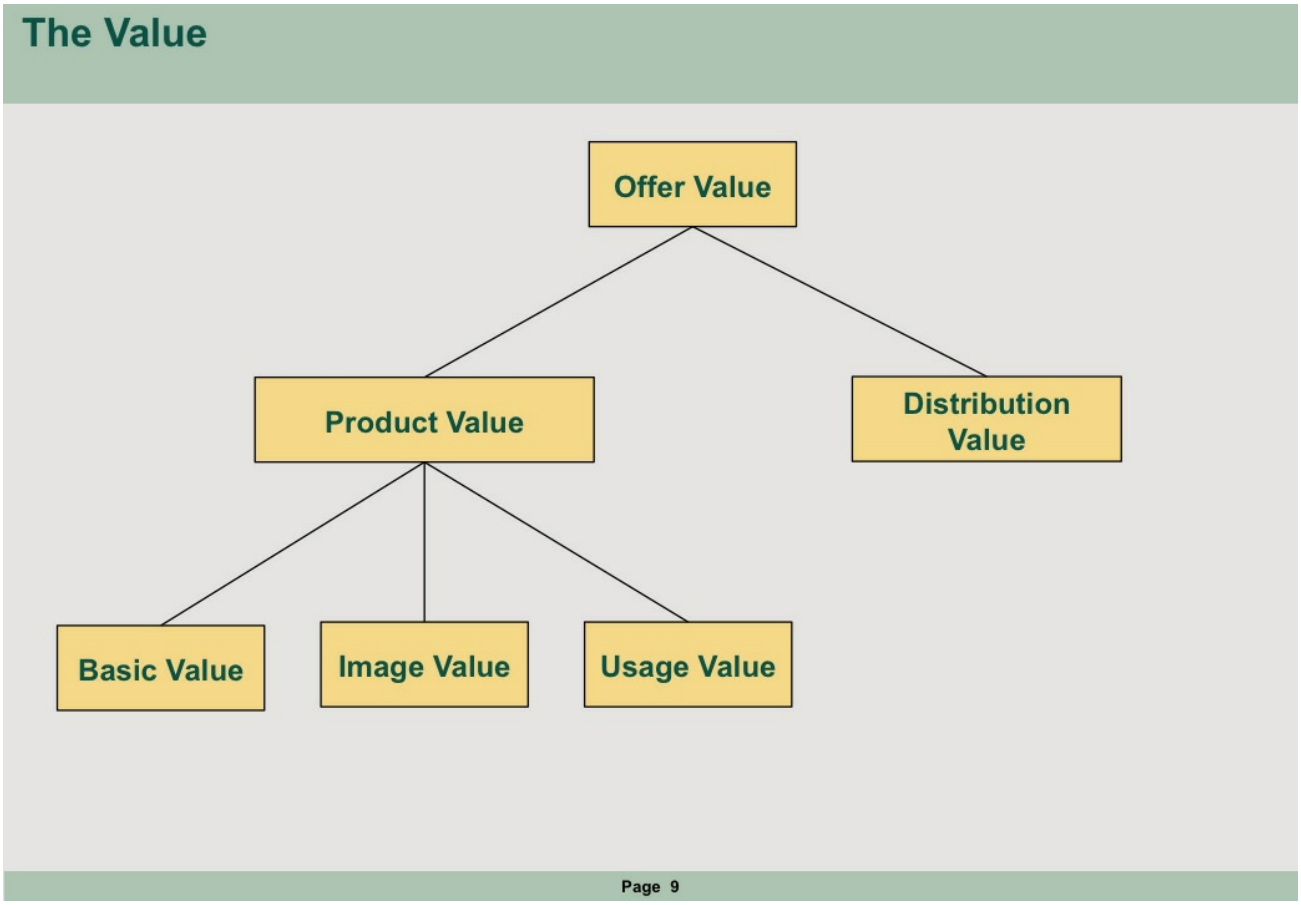
But there is also another trend developing: **freeware**, which is not only free of charge, but also provides access to the internal code. We can therefore modify it, assemble it, distribute increments in the same form, in the context of an open community.

# Offer a new usage Value




## 1. What is a usage Value?

The Value of an offer has been described in the "Product Model".




Digital can impact Offers on different levels:

- New, more comfortable usability: we do not change the basic Value, but we make **using** it easier
  - e.g., music is easily accessible from one's smartphone
  - e.g., Wikipedia versus a paper Encyclopedia
  - e.g., video conference versus a in-person meeting, without have to go anywhere
- But also a new **Distribution** Model: we enable an easier mode of distribution for the Customer
  - e.g., download a book onto his/her tablet
  - e.g., order on the Internet
- And lastly, new **basic Values**
  - e.g., find the best route thanks to GPS navigation
  - e.g., smart shoes that inform the wearer of the efforts made
  - e.g., Bitcoin as a new currency

These examples show that all sectors of activity are impacted: whether it be Goods, Information or Services. Not only are the established positions called into question but major change is extremely quick.

## 2. How do we imagine new usages?

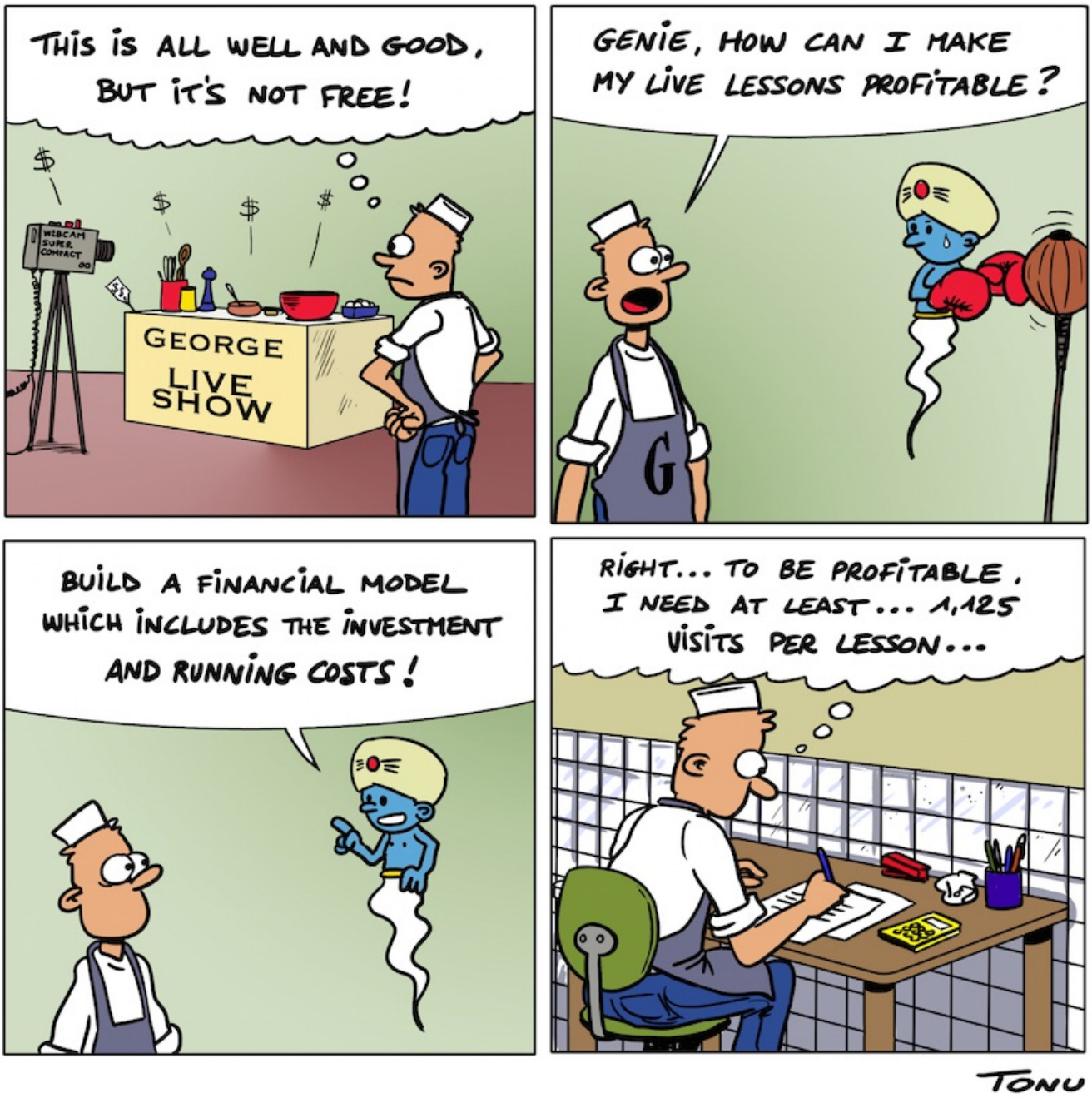
This is obviously the question that we would like to know how to answer: we could introduce innovative Offers on the market before anyone else and quickly build a new Google, Facebook, SalesForce or Amazon... Unfortunately, we have not found a miracle recipe. As much as we can rationalize optimizing the Operational Processes, it is difficult to image these new usages.

At the most, we can apply certain principles such as:

- Ensure that the innovation provides a real **Value**
- Think "**Global**": it is a global Market

- Start by **prototyping** the new idea on a sample to verify its success
- As always, most innovations will will not be very convincing: **do not give up** at the first sign of failure
- Try to **protect the innovation** by patents
- As soon as an idea has been validated, **quickly ramp up the volume** to take market share
- Only seek **profitability** when the volume is there
- Always keep **control of the Model**, which will not stop you from subcontracting all or part of the Operations
- **Listen to the competition** who will try to copy and perfect the innovation

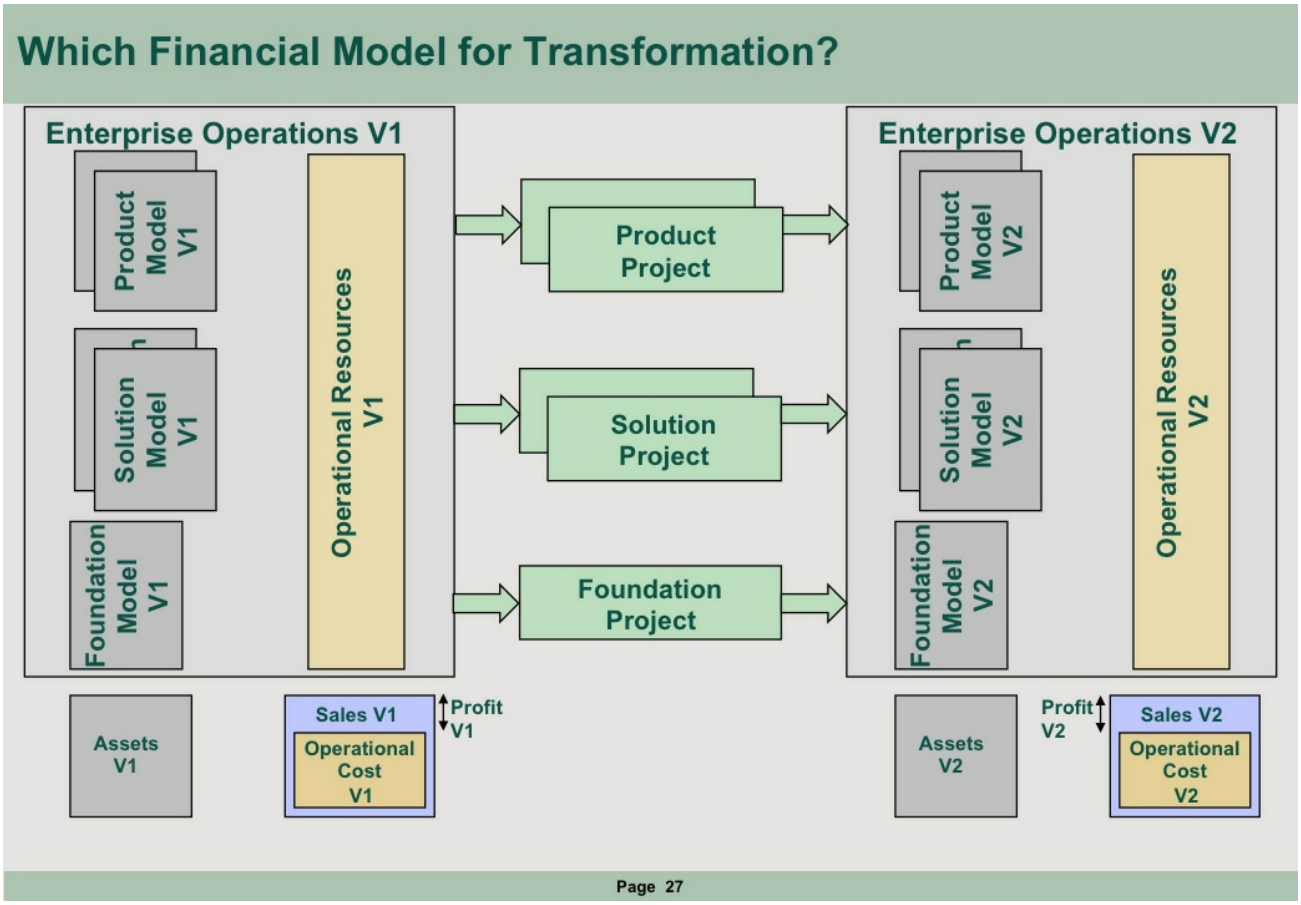
# Build a financial Model




## 1. What is a financial Model?

A financial Model lets us simulate the financial situation of the Enterprise according to investment, operating and growth assumptions.

As an example, a financial Model to decide on a new Product or Solution of Foundation project is similar to the one shown in the diagram below.




At any given time, the Enterprise Operates with a Model V1: it respects its Product Model; it Operates by respecting the Operation Model, that is to say the Solution Models which are themselves built by respecting the Foundation Model.
Its Operational costs have to be deducted from its Operational revenues to deduce its Profit.

But, if the Enterprise is considering Operating with a Model V2, for example to increase its productivity (new Solution) or to launch a new Product range, we will have to Build then Deploy this Model, which requires a Project whose investment cost has to be compensated for by the increase in future Operational profits.
The number of years during which the new Model will be active depends on the time period between both versions of the Model.

To reach a good level of precision, we have to know how to evaluate the time required, and therefore the cost, of a Transformation, which is the first challenge as we systematically underestimate the time needed to Transform. We also have to be able to evaluate the differences in cost and Operating revenues between both versions, which is the second challenge. Finally, we have to determine the lifespan of the new Model to know over how many years it will be amortized.

## 2. How do we evaluate the intangible Capital?

But the result of a Transformation Project also changes the Capital of the Enterprise. When we think of "Capital", we tend to think of "Goods": premises, equipment and stocks of intermediary Goods are part of that.

But the Enterprise also possesses an **intangible Capital** that can evolve with the help of a Project:

- The Transformation Model and its consequence, Agility, can also evolve
- The Foundation is enriched: it participates in the overall consistency of the Enterprise.
- The Image of the Enterprise can be changed
- The Enterprise Culture may also be transformed

We manage today to value the image through the brand: evaluations and rankings are made every

year to associate an amount to a brand.

It is extremely difficult to value the Enterprise Culture and its Transformation Model. Yet, everyone understands that the Value of an Enterprise is also linked to its capability of evolving quickly: but how do we quantify it? Financiers include this evaluation in what they call "goodwill", but the evaluation methods are still really empirical.

### 2.1 Do not forget the Foundation

As we mentioned earlier, a Foundation represents everything that can be reused to contribute to the **common good** in the Enterprise, especially the Enterprise Architecture and all the reusable Model Components that can be used to assemble new Solution Models or new Product Models.
When we invest in a Foundation, we increase costs without any immediate results. Profits will come when Building new Models that use the Foundation.
The profitability of a Foundation can only be reached at the second level: not only must the Foundation be well built, but it must also be used by the new Solution Models or by the new Product Models. It is therefore a very difficult investment to bring about; and yet, those who have managed have profited from a sustainable competitive advantage.

We can mention the examples of German automobile manufacturers who have a high rate of reuse of their components to assemble their new Models.
We can also cite technology companies like Amazon, Apple or SalesForce who reuse, to a large extent, the software components to build new Solutions for themselves, and who have now made these components available in the form of a new Offer for other software vendors.

The economic Model must therefore include the intangible capital that the Foundation represents. Building new Product Models or new Solution Models more quickly has a Value: **agility** is part of the capital of an Enterprise.
But how do we evaluate the Agility Value? How do we prove that the investment is worth the candle? Once again, we come up against the limits of a financial approach.
The same reasoning can be held for the other intangible elements, Culture or Image, which require investments whose profitability is hard to prove.

One method consists in following a **checklist of criteria** to identify where we bring Value:

- Criteria linked to the Transformation
  - More Agility?
  - Better "time to market"?
  - More robust Solutions?
- Criteria linked to the Operations
  - Enable more rapid Growth?
  - Improve the customer Relationship?
  - Generate international Products?
  - Help Synergy between different entities of a group?
  - Better control of risks?
  - Provides better management information?
  - Improves quality of service?
- Criteria on the Resources
  - Facilitate human resources management?
  - Reduce Operational costs?
  - Reduce Transformation costs?
  - Facilitate Mergers?
  - Enable better control of the IT Operations?

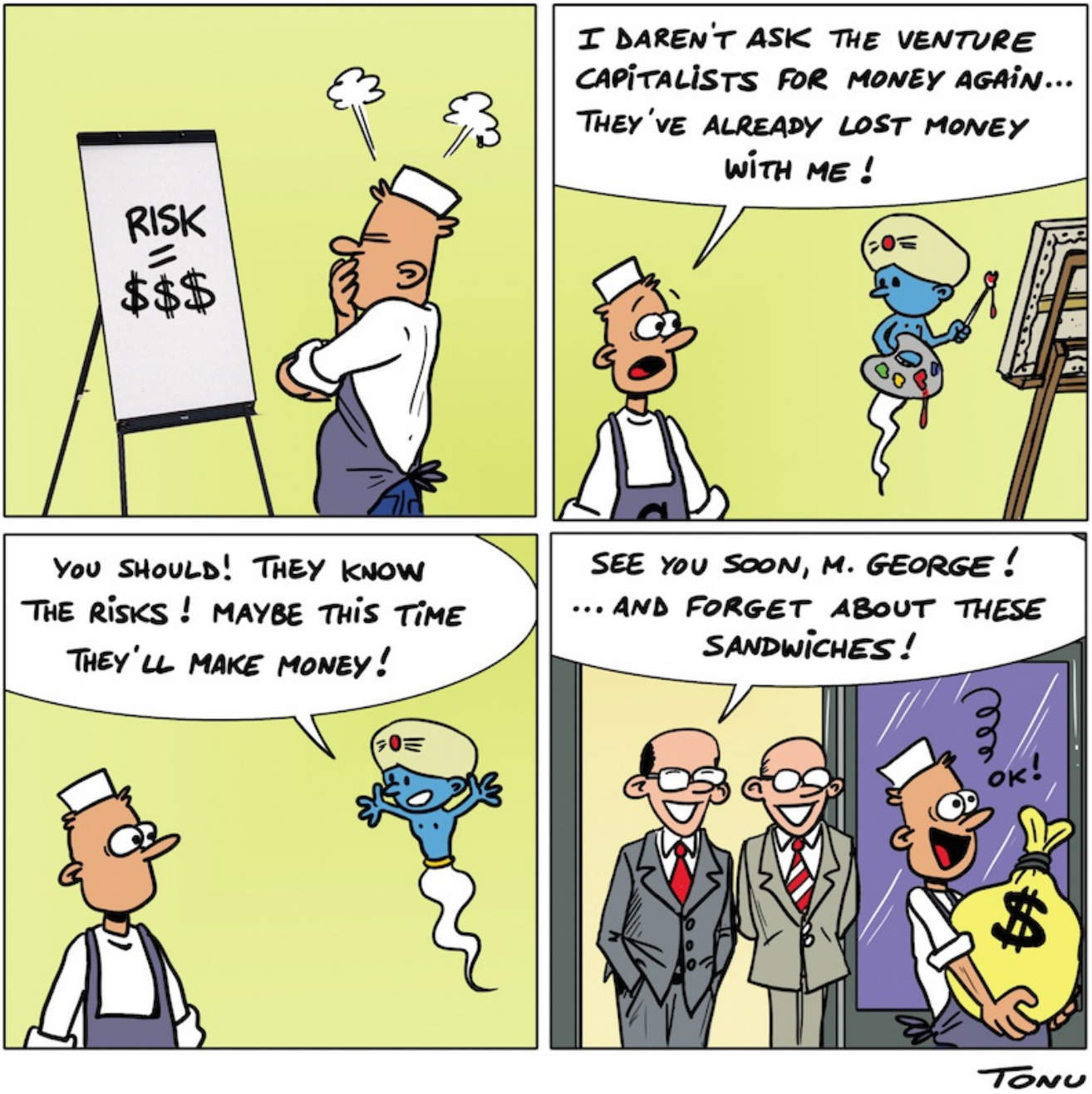
# 3. Financing by advertising

In 2013, YouTube captures 20.5% of the video advertising market on the Internet, as do Google or Facebook (see the emarketer study in French).

An increasing share of advertising budgets are now given over to finance Internet-accessible

Solutions. The revenue no longer comes from Distributing the Offer (that we have to pay for), but from the online advertising that must be taken into account in the financial Model.

## The financial risk is high for investors




# 1. Difficult to draw up a balance sheet for disruptive Transformations

Whether it is an Enterprise start-up or disruptive Transformation in an existing Enterprise, the risk is considerable.

In truth, the deeper the Transformation, the more difficult it is to predict the financial consequences: there is always one part risk in any Transformation, which explains why financial talents alone are not sufficient to make a decision; we need to have deep fundamental beliefs about the viability of a Transformation.
Few innovative Enterprises were started by financiers: risk taking requires a Competence and a Business vision. A Transformation often succeeds because the initiator has an innovative vision of his/her Market and does not follow in the footsteps of his/her colleagues.

Startup Business Plans are never respected: investors know this well and give priority to the trust placed in the management team over unpredictable financial forecasts. They nevertheless want a Business Plan that acts as a reference point and is updated progressively to reflect events.

## 2. The right usage before profitability

Aware of the difficulty of establishing a realistic balance sheet in disruptive Transformations, investors have modified their approach in 3 stages:

- Stage 1: prove that the Product or Solution works. With a new product, for example, we have to find the **right usage**
- Stage 2: create volume and get a large **share of the market**
- Stage 3: look for **profitability**

Recent experiences with new technologies have shown that seeking profitability from the outset generally leads to failure.

## 3. Do not be embarrassed to fail

We cannot alway win: it is by investing in several startups that investors spread their risks.
The Entrepreneur or project manager in the Enterprise should not be embarrassed about failing: the experience gained increases the likelihood of success of future Transformations.
And, once again, do not hesitate to give up on a Transformation if it proves to be inefficient.

## Act 8: Merge with a cake-shop network

# Mergers are one way of growing quickly


There are only 2 ways of succeeding: either do **better** than the competitors (a better Offer or a better organization), or propose the same Offer at a **better price**.
In the second case, size enables us to achieve economies of scale at the same time as it allows for a better visibility, and inspires more confidence.

Growth can be organic: it is the success of the Enterprise on its Market that makes it grow gradually.

But some directors want to go more quickly: the solution then consists in partnering with colleagues or acquiring them, then merging the Enterprises to more quickly reach the critical size sought.

It is also one way of acquiring complementary Models:

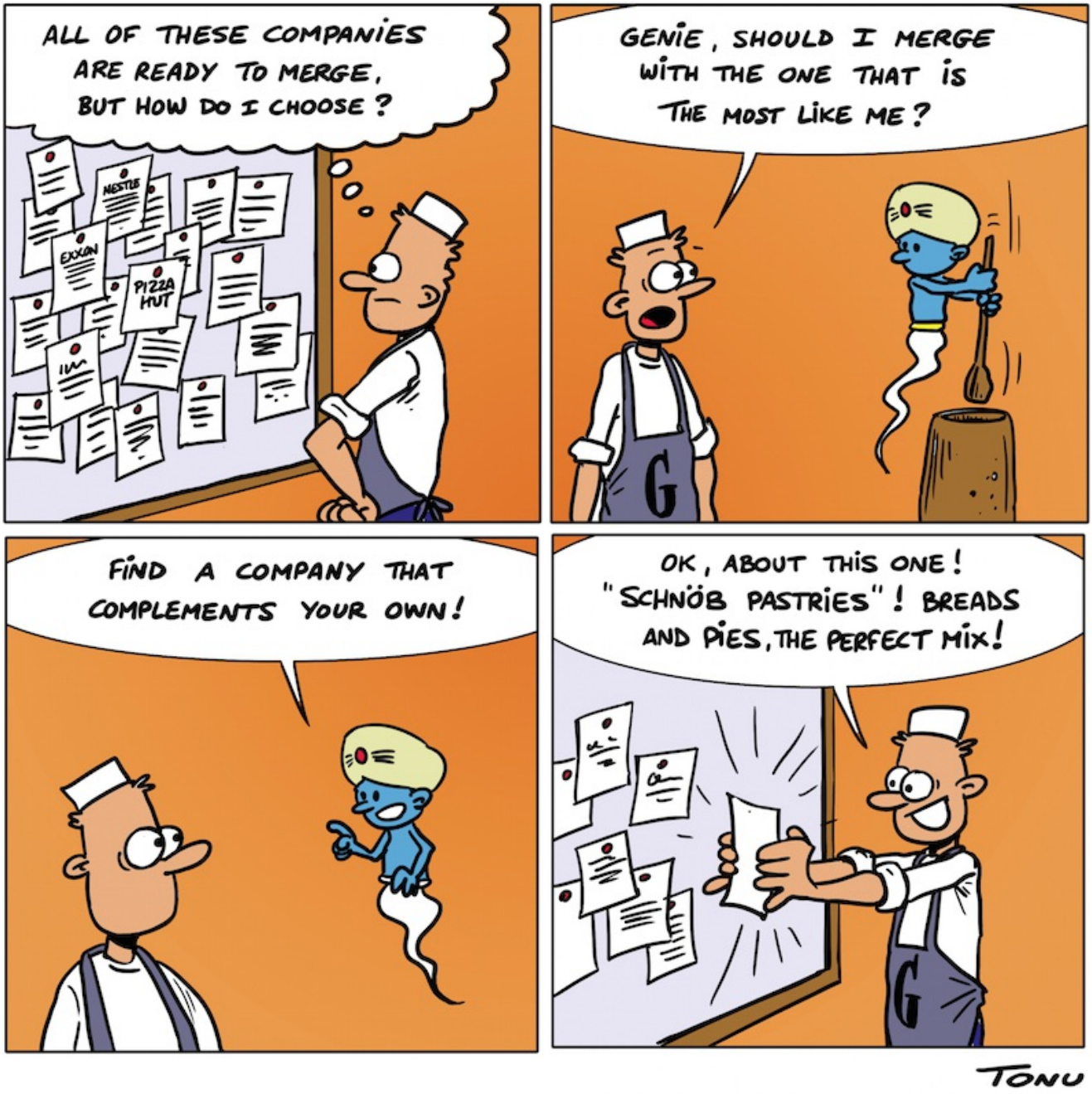
- new Product Models,
- Operational Models that can be more efficient in certain domains,

- more agile Transformation Models,
- a positive Image

An Enterprise merger considerably accelerates growth. But it is riskier than organic growth because it confronts two different Enterprise Models that will have to be brought closer together for us to truly benefit from the merger and not just be content with the addition of 2 operating accounts:

- bring the Offers closer
- bring the Operation Models closer
- bring the Transformation Models closer
- bring the Cultures closer
- bring the Images closer

# The merger must have a Business interest




Merging two identical enterprises is always is Culture shock: we know our customers, our products, our hierarchy, our tools, our premises and we struggle to become aware of another world. The merger will be difficult anyway.

But the game is worth the candle if a complementarity exists that strengthens both of the original entities. If the Offers are different, we can think about having each entity sell both ranges in its distribution network. We double the market potential.
If the **territories** are complementary, we can do the same thing.
If we choose to select and then generalize the best **Operational Processes** from each one, we can improve our overall productivity.
If we bring together the best Transformation practices to build a new Approach, we can gain in Agility.

# Finance through banks


## 1. The Banks come into play

Risk analysis in banks forces them to only finance organizations for which they have financial visibility.
They cannot therefore be the first port of call for the first, very risky, steps of starting up the Enterprise. It is impossible for the Enterprise, in its early days, to provide sufficient guarantees to the banks: whatever the initial successes may be, we do not have enough background history to prove the repayment capability.
Bankers become involved when the new Model has proved itself and when we need to finance the growth.
The method goes more through a financial analysis than in the analysis of the validity of the project or the quality of the teams that were the key criteria in the first stages.

## 2. Listing on the Stock Exchange is also an option

From at least 10 million euros, being listed on the stock exchange can be a source of financing.
The procedure is quite complex, coming with certain legal and regulatory constraints.
It is one way of diversifying the sources of financing, of increasing the renown of the Enterprise and offering an exit route for historical investors (Business Angels or Venture Capitalists). It can also have an important impact on employee motivation.

# Merging cultures is difficult


Enterprises that operate on the same Market may have very different Cultures:

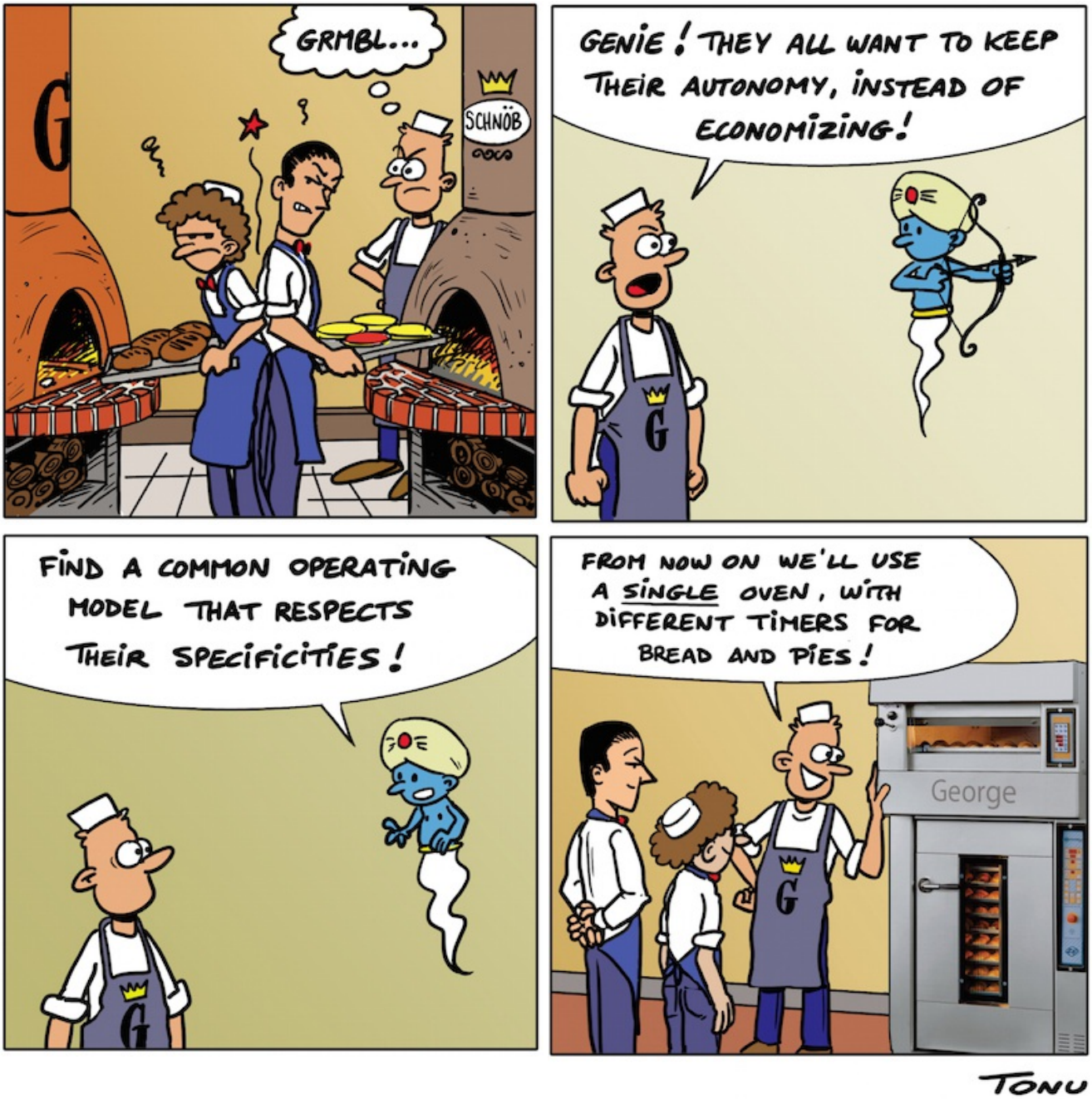
- The circulation of internal information is limited or, quite the reverse, is generalized
- The hierarchy is distant from the field or it is close to its teams
- The executive management is interested in Transformation or in the Operations
- Digital is considered to be a secondary concern, or a vital domain
- We accept or not to take risks
- We look for egalitarianism or to reward the best
- We favor the Customer or the staff
- We spend a lot of time and effort on reporting or we content ourselves with a few information summaries
- ...

Even if the Products and the Processes are close, the merger can come up against different Cultures

and lead to a rejection from one party or the other. It is not enough to harmonize the Models, we also have to adapt the Cultures.

In actual fact, the most efficient way of merging rapidly consists in defining the organization structure and appointing, **from the outset**, the new managers in order to act quickly.
It is the best way of setting the teams to work and avoiding that each person goes over, in his/her own corner, the chances that each manager has of being chosen to fill the positions of responsibility.

## Economies of scale presuppose merging the Operational Models




# 1. Converge towards a same Operational Model

If each Enterprise continues to apply its own Models, the merger does not really happen: it is simply a financial operation that consists in adding up 2 distinct operating accounts.

To achieve a successful merger, not only do the Product Models have to converge, but the Operation Models (Production, Distribution, human Resources management or enterprise management) have to be identical too.

We cannot merge 2 branches or 2 back offices, if each one works with its own methods and its own IT applications.
We cannot merge the Transformation teams if each one is working on its own Architecture.

This merger of Processes is difficult because it requires many people to change their work methods: not only employees but also partners and customers.

But the most difficult and costly is to merge the information systems: many mergers stumble over the difficulty of integrating one enterprise's data into the Model of the other, of introducing the original functionalities of the other enterprise into the Model of the first one.

# 2. What target Model do we choose?

If we decide to only keep one Operation Model for both Enterprises, the difficult question is to choose the target Model.
There are three scenarios:

- We blend the Solutions coming from each Enterprise to obtain a mixed Model
  - It is a way of not upsetting anyone and balancing the efforts
  - But the result is, in general, a patchwork of unrelated Solutions that will be difficult to evolve
- We favor the Model of one of the 2 enterprises which becomes the common Model: the other Enterprise has to migrate to this new model
  - It is the quickest way to succeed with the merger
  - But the Enterprise that has to migrate may feel penalized
- We build a new Model and we wait for it to be available before gradually migrating to it
  - We build something new, when the merger takes place, and no one is favored
  - But we have to wait for the new Model to be ready

Our recommendation is:

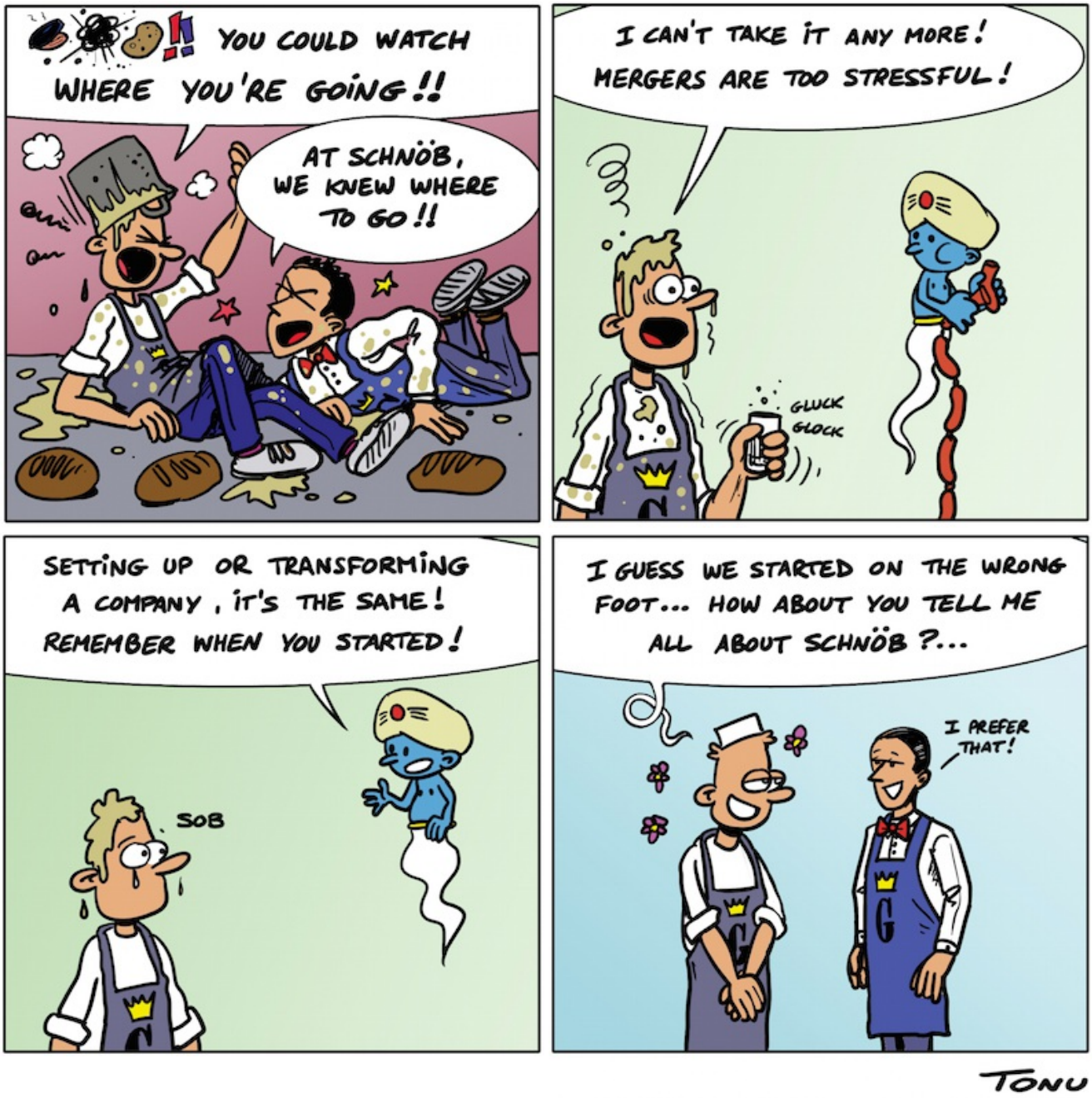
- Never blend: the result is too complex; if we choose to do it nonetheless for political reasons, select the best Solutions and do not seek to respect a balance in order not to vex anyone.
- Prefer the second scenario to go quickly: select one of the 2 information systems so that we do not have to interface the Solutions coming from 2 different worlds. Again, the criterion of choice here is to choose the **best system** and not **the one from the biggest enterprise**.
- As a second step, rebuild a modern Model for all concerned. But experience has shown that, after the efforts of merging, it is very difficult to mobilize the troops for another deep change.

Among the main criteria for choosing a Solution:

- Is there an Architecture or a Foundation around which we can make the Model grow?
- Are the Processes efficient?
- What is the quality of the customer information?
- Is the technology obsolete?
- Are the performance and quality of service of a good quality?
- Is the Solution able to evolve quickly?

## Act 9: Execute the Transformation well

**Transforming an Enterprise uses the same practices as starting up an Enterprise**




# 1. Disruptive Transformation and Enterprise start-ups

## 1.1 Innovation for all Products

Innovation can concern the elements that make up the Offer (Goods, Information, Services) or the Operation Model.

- To innovate in Goods, we have to use sciences that affect Goods: physics, electronics, chemistry, energy...
- To innovate in Information, we have to use IT
- To innovate in Services, we have to use Process modeling and IT

- To innovate in the Operation Model, we also have to use Process modeling and IT

If we want to make enterprises such as Google, Amazon, SalesForce or SAP emerge, we must know how to invest not only in classic engineering sciences, but also in **software** or **Process modeling**.

### 1.2 Innovation in the Large Enterprise or in the start-up

We often compare the large enterprise, which benefits from strong notoriety, a loyal customer base, considerable economies of scale, an international Market and reliable products, to the small, innovative enterprise, which demonstrates its originality but does not have the means to impose itself on the international markets.
And yet, carrying out a disruptive Transformation in a large enterprise can be just as difficult, even more difficult than starting up an Enterprise.

## 2. The same practices are used in the Large Enterprise and in the Start-up

Whether we carry out a disruptive Transformation within a Large Enterprise or whether we start up a new, innovative Enterprise, we use the same basic practices:

- Model new Products and Services
- Choose a Transformation Model
- Equip ourselves with a Foundation
- Set up Operation and Transformation Resources
- Know how to build a financial Model

### 2.1 Know how to create new Product or Service Models

Whether the innovation is a Product innovation or an Operation Model innovation, we have to know how to build new Models, in particular the associated software.

The large, innovative enterprises have kept control of the building of their business software: Google, Microsoft, Facebook, Amazon, SalesForce... develop and maintain their own business software, even if they often use commodity software packages to manage their Resources.

The best software developers are some of the most well-paid employees.

Small teams made up of excellent experts in Modeling are one of the keys to success.

### 2.2 Choose a Transformation Model

Before we throw ourselves into the Transformation Project or Enterprise start-up, we have to decide with which Approach and with which Tools we would like to build the Operation Model or Offer Model: we have to choose the Transformation Model.

### 2.3 Equip ourselves with a Foundation

As explained previously, if we want to have consistency and fluidity between the different Enterprise activities, we need to define an Enterprise Architecture, which is the framework that the different Solutions will fit into. Good Interfaces enable the Solutions to exchange without us having to duplicate the information or re-enter it.
If we want to build an agile Enterprise, we have to equip ourselves with a Component bank which speeds up the process of Building new Models.

Enterprise Architecture and Components are part of the Foundation.

### 2.4 Set up Operation and Transformation Resources

Separate Operations and Transformation: it is a necessary condition for Agility.

### 2.5 Know how to build a financial Model

see the corresponding scene

## 3. What is more difficult in the Enterprise start-up

### 3.1 Find financing

Unlike the large enterprise, the start-up struggles to find financing because it represents a considerable risk.

### 3.2 Find and convince customers

The Image and notoriety of the Enterprise start-up is often a handicap in promoting new offers: how can we make people know that we provide an innovative Offer?

### 3.3 Gather all competences together

Launching a new Offer requires us not only to call on engineering and marketing competences, but also sales, legal, financial, accounting competences... which are available in the large enterprise but are often lacking in the small one.

## 4. What is more difficult in the Large Enterprise

### 4.1 Difficulty to innovate

The governance of Large Enterprises favors continuity, security, the principle of precaution... but stifles innovation: it is difficult to transgress the rules to imagine new value propositions. Even when innovation is present, the road is difficult: the Kodak laboratories had invented digital photography, but the Enterprise floundered because it did not know how to call its traditional Model into question. Innovation is necessary but it is not enough: the Enterprise, and especially its top management, has to take hold of and carry the Transformation to its end.

### 4.2 Change management

The social body of the Enterprise does not like change, in particular when the Enterprise is prosperous, because change of any kind requires an effort, a calling into question of the hierarchies in place: how do we explain that we have to change Model when the current Model works?
Even if the Enterprise employees are convinced, we also have to convince the customers and partners who like to keep their habits: see the difficulty that Microsoft has to impose Windows 8.

### 4.3 Migration from the old Model

Migrating Enterprise information from an old Model to a new one can prove to be extremely complex.

## 4.4 Internal procedures

The unwieldiness of a large enterprise can be incompatible with fast decision-making.

## 4.5 Self-satisfaction

The Employees should be proud of their Enterprise. But this pride occasionally borders on self-satisfaction: the fact of belonging to a large and powerful enterprise leads some to ignore the initiatives of others, especially if they are a lot smaller.

## 4.6 Negative notoriety

Notoriety can be positive or negative; in the second case, it is difficult to get back on one's feet and it is often better to start again from scratch by setting up subsidiaries that have an unsullied image.
Some Enterprises, like Adidas or Monoprix, have succeeded in transforming and modernizing their image.

**The Transformation team should be multidisciplinary and have a leader**


## 1. Any innovation is an iterative compromise between business hopes and technical possibilities

In a slowly-changing world, we can gradually improve the existing Products and Processes by successive touches. As we remain in a known environment and as the changes are gradual, we can proceed using a classic Approach, of a sequential kind (or "waterfall").

- The Strategists define the Goal
- Marketing defines the new Product
- The Project owners translate the new Product or Process definition into specifications
- The organization defines the roles and allocates the tasks

- The IT staff build or adapt the software according to the specifications
- The testers validate the Solution
- The HR department trains the staff

In a rapidly-changing world, we cannot use the same method: task serialization is a handicap to speed and the objectives may be called into question depending on the unpredictable success of the new Model.

It therefore becomes more efficient to group together, in a single team, all the intervening parties of the Transformation: Strategists, Marketing, Organization, IT, change management, HR...

The dynamic compromise between business hopes and technical possibilities emerges through successive iterations, within the same team.

## 2. Can we function with distributed teams?

A Project team always works best when its members are located in the same building. But it is not always possible.
We can, nonetheless, apply an agile approach between distributed teams if they share efficient communication tools, not only between individuals but also between Models.
Ideally, we should provide each local team with a local repository and aggregate the contributions from each local team at a Central level, in a global repository, which manages versioning, consistency and provides the Views that each person wants.

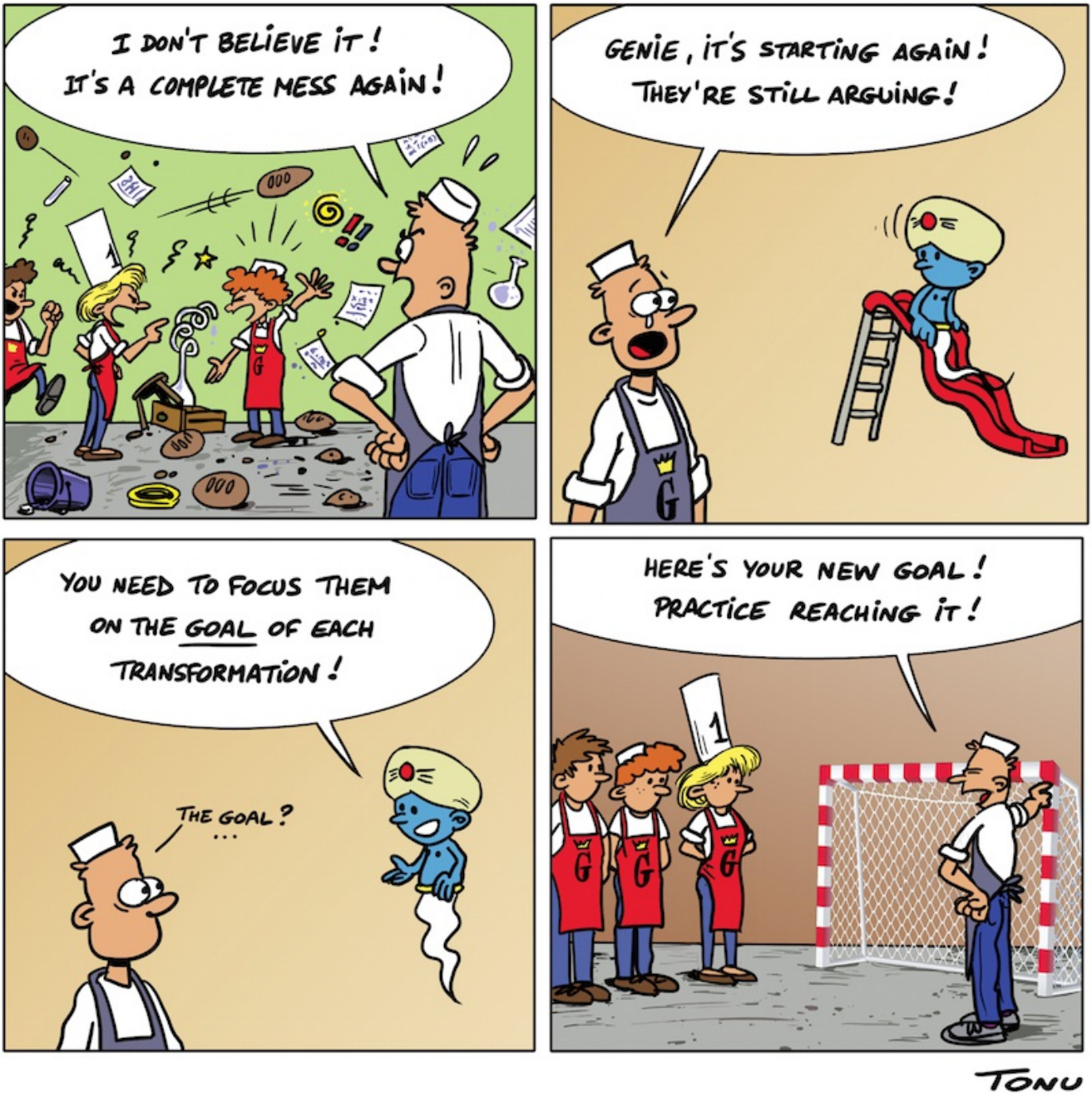
## 3. The versatile team must have one, and only one, leader

We cannot carry out a Transformation quickly if the decisions are not fast. A leader is necessary. But what profile should we give preference to for this leader?

It is difficult to find the rare bird who is able to understand all of the disciplines. And yet it is the lot of the boss of the enterprise start-up to associate these talents, while at the same time gaining respect through his/her Competence, ability to make clear decisions in an uncertain environment and natural leadership.

The simple qualities of **administrator** are not enough, we also have to **understand the new Enterprise Models** in order to know how to criticize or perfect them.

# Formalize and share the Goal




## 1. The Goal of the Transformation must be formalized

It is surprising that the Goal of Transformations is rarely formalized: all Transformation projects generate a plethora of documents on the scheduling, budgets, teams, governance, specifications, test results..., but it is rare to find a short and precise document that defines the Goal to be reached.
In most cases, it is because this goal is implicit and considered by everyone to be obvious. But the absence of this document generates a waste of energy: some will follow different and pointless pathways, simply because they have not understood the Goal.
To make good decisions, the Actors should always refer back to the Goal that they are seeing to reach: in what way do the actions proposed help to reach the Goal?

## 2. How to describe a Goal

A Goal is described in 3 parts:

1. The **scope**: which combines several dimensions, such as

- geographic,
- product line,
- process domain

2. The **objectives** and related **indicators**: the objectives can be gains in productivity, improvements in the quality of service, a new geographic territory, new Product launches, a change of Image... The indicators are generally quantifiable elements that will enable us to verify that the objectives have been reached.

3. The **constraints** that we set to the project: timescale, budget, involvement of which teams...

# 3. The Goal should be widely communicated

So that all Actors in a Project make the right daily decisions, it is important to systematically refer to the Goal, which presupposes that each person has understood and remembered the Goal in order to properly **align** the Solution with this Goal. We therefore have to give a **simple explanation** (**one** page) and communicate it to everyone, for example by displaying it in all the Transformation teams' offices.
Circulating the Goal is useful for the Transformation Actors, but it is also useful for the Operational Actors: the fact that everyone understands the objective that the Enterprise wants to reach helps the acceptance of the Transformation.

## All Transformation Actors must share the same Transformation language


# 1. A multidisciplinary transformation team must share the same Transformation language

Each profession uses its own jargon: it is difficult for the other disciplines to really understand the strategists, marketing, legal experts, IT staff...
As we want them to work together, they have to share the same language.

# 2. A Glossary is proposed

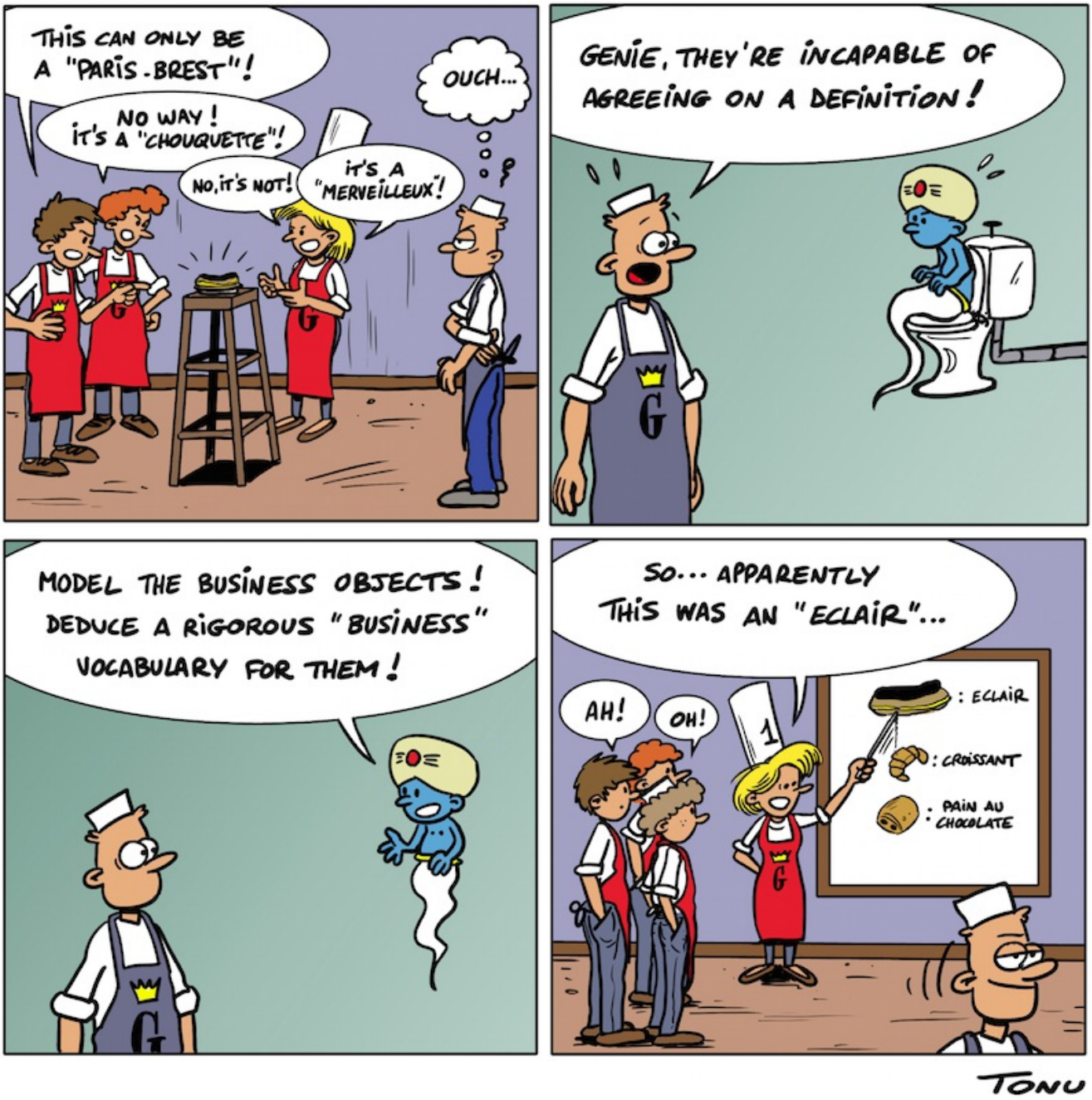
A glossary is proposed at the end of this work which can act as the basis for establishing a

common language in each enterprise.

**Principles**

- The terms selected must be comprehensible to all the Actors
- The definitions should be **short** and extendable by Role
- The terms in the glossary begin by a **capital letter**
- No Service homonyms
  - Business-Service, IT-Service, Software-Service, by default mean Business-Service
  - Architecture: Architecture Model or Architecture Discipline, by default mean "Architecture Model"

**All the Transformation Actors must share the same Business language**




## 1. The Transformation language is not enough: we also have to define a Business language

Rigor is also necessary when using Business terms: so that the experts understand each other, they have to share clear concepts.

When the Business defines its Goal and its Solution Model, it must use specific Business terms that will help it clarify its Model and **communicate more easily** with IT. It is also a means of **properly structuring the** Solution **Model** which relies, above all, on the Information Model and helps make the **Reusable Components** emerge.

Terms that the Enterprise uses every day, such as Customer, Product, Contract, Service, Partner... are rarely defined with precision: but, why waste time defining Terms that everybody knows?
Simply because in practice, more often than not, each term groups several Entities together.
For example, "Customer" means:

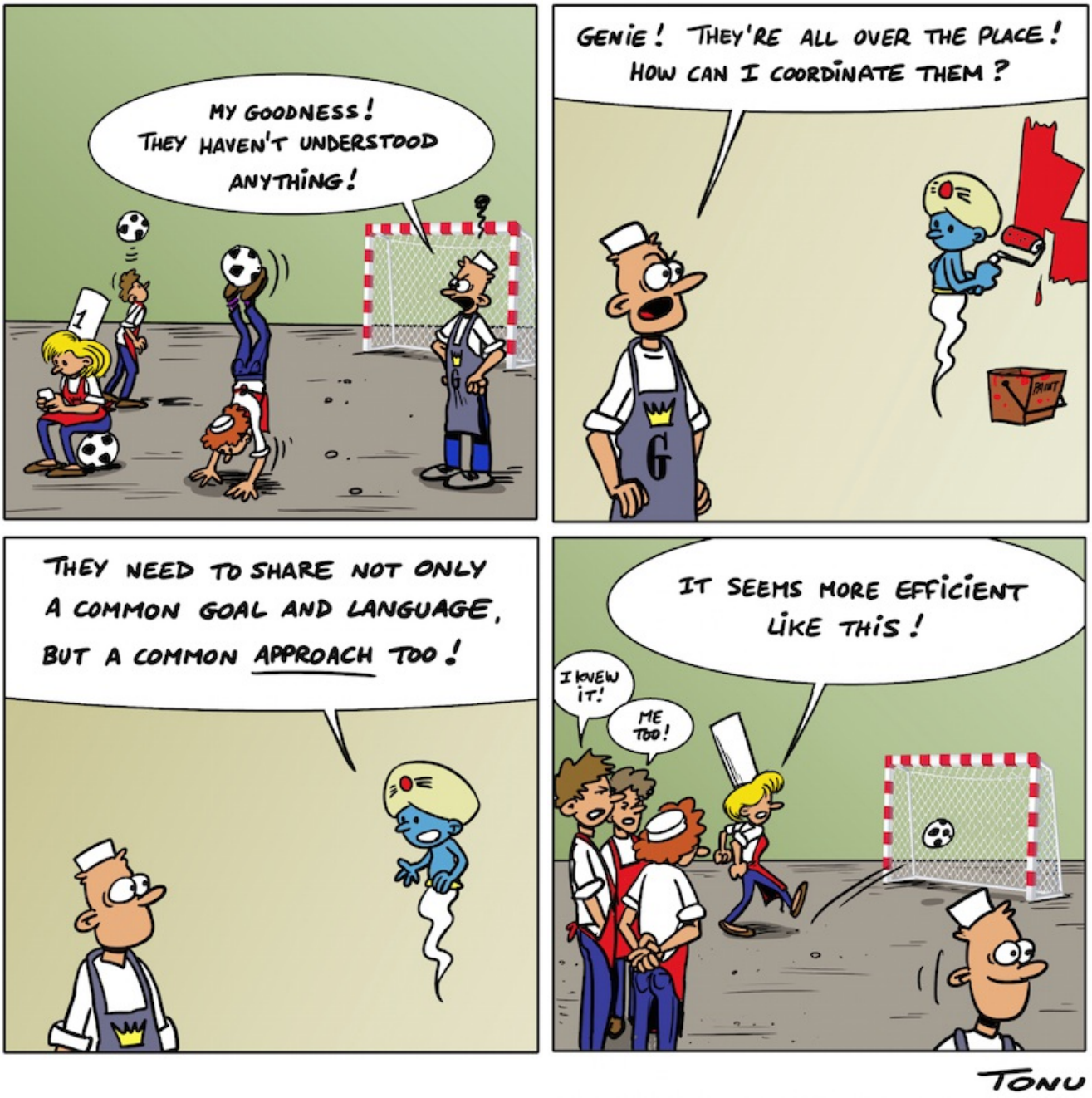
- the one who we sell to (for the sales rep),
- the one who subscribes (for the company lawyer),
- the one who pays (for the accountant),
- the one who we deliver the Product to (for the beneficiary),
- the one who benefits from the Product Value (for the user).

The same person can act several of these roles.
However, asking to track the increase in the number of customers is only useful if we clarify what we mean by Customer.
We can carry out the same exercise with "Offer", "Product", "Service", "Contract", "Resource", "Flight" at Air France or "Train" at the SNCF...

## 2. How do we go about it?

- Appoint a manager of the Business glossary (it is not a full-time job).
- Use the dictionaries from inter-professional organizations.
- Try to only have **a few definitions**: look for the "GCD" (the greatest common divisor: the subset common to the different Enterprise activities) and not the "LCD" (the lowest common denominator: the superset of everyone's needs). The Large enterprises who embarked on this approach only managed to impose a common language when they reduced their glossary to the bare minimum. In practice, we only need to rigorously define 50 to 100 Business terms to considerably improve the dialog.
- Refine the glossary through successive **versions**: ask all contributors to use it and gradually refine it.
- This Glossary is **extended by the** Business Entities Model (relations, inheritance, login names, life cycle of each Entity) which represents the Solutions Architecture. This definition uses the verbs "Be" for the inheritance and "Have" for the relations.

# The Transformation Actors share Approach and Tools




## 1. Decomposition of the Transformation approach

### 1.1 The Transformation Processes

There are **different Transformation Processes**: Design a new Product, Build a Solution, Build an innovative Solution, but also Modify a Solution (evolution or correcting "bugs"), Build Foundations, Define a Road Map,...
This document develops, in particular, the Building Process or Modification of Solutions.

### 1.2 A Transformation Process is broken down into Phases

We know how to formalize the **Operational** Processes well, like the order Process, and we progressively automate them. It is much more difficult for the **Transformation** Processes like the Solution Building Process, in view of the uncertainty linked to any project.
Much progress has nevertheless been accomplished: each Enterprise has defined its own Transformation Process by breaking it down into progressive **Phases**. Each Phase has an objective and a deliverable. An end-of-Phase Milestone enables us to take stock of the situation, validate the Deliverable and move on to the next phase.

### 1.3 Each Phase calls for Functions

The **Processes are different** from one enterprise to another, but they all reuse **identical Transformation Functions**. We distinguish two types of Functions:

- the **Engineering Functions** (like "Model Processes") in order to **Build** the Solution
- the **Management Functions** (like "Plan") to **Manage** the Project well.

The Engineering Functions are the **Business Functions** of the Transformation: they have to be accomplished whatever the established Organization. The Management Functions are the **Organizational Functions** of the Transformation: they depend closely on the Organization and the Approach. They are therefore **specific** to each Enterprise, whereas the Engineering Functions are **universal**.

### 1.4 Progress in Management but inadequacies in Engineering

Progress has especially been around **Management**.
We have to continue down this path of improving Management, but not go overboard: too many Management tasks prevent the project manager from focusing on the Engineering Functions. We can **Manage very well** the project of a **badly built** Solution.

## 2. How do we Transform the Transformation Model?

### 2.1 A multidisciplinary approach

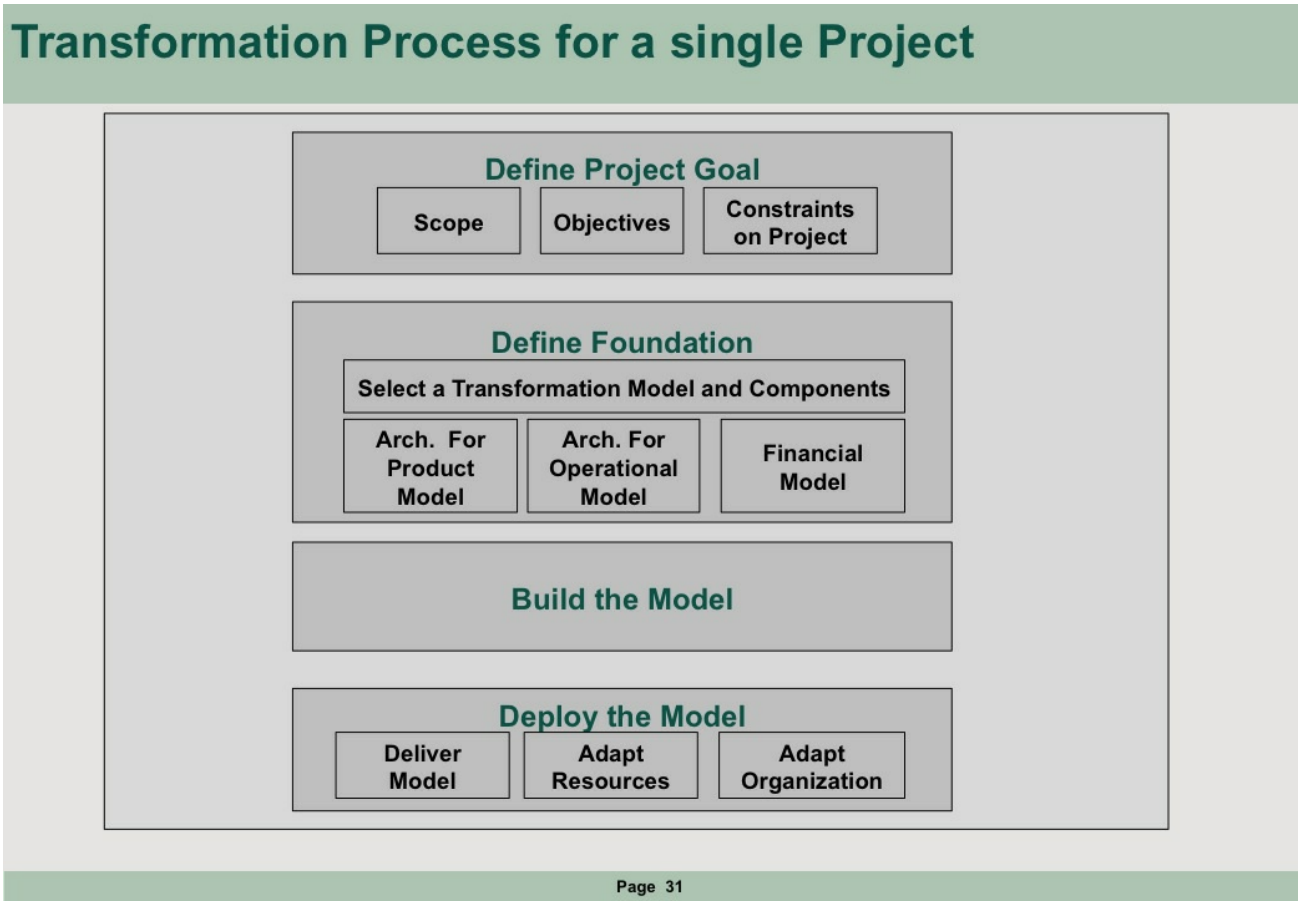
Define a **single** Approach, shared by all the participants in the Transformation, especially Business and IT, which is not yet the case in all Enterprises.
This Approach will be accepted better if we have defined the Transformation language and Business language beforehand.

### 2.2 Shared Modeling tools

The Transformation must be tooled like the Operations: both to manage and to build.

- Tools to manage: manage the scheduling, resources, incidents, budget, communication...
- Engineering tools: design and 3D simulation tools, mapping tools, tools for modeling products, requirements, processes, software, user interfaces, tools to control the quality of the Model, test, document, analyze the performances, collaborative tools, configuration management tools...

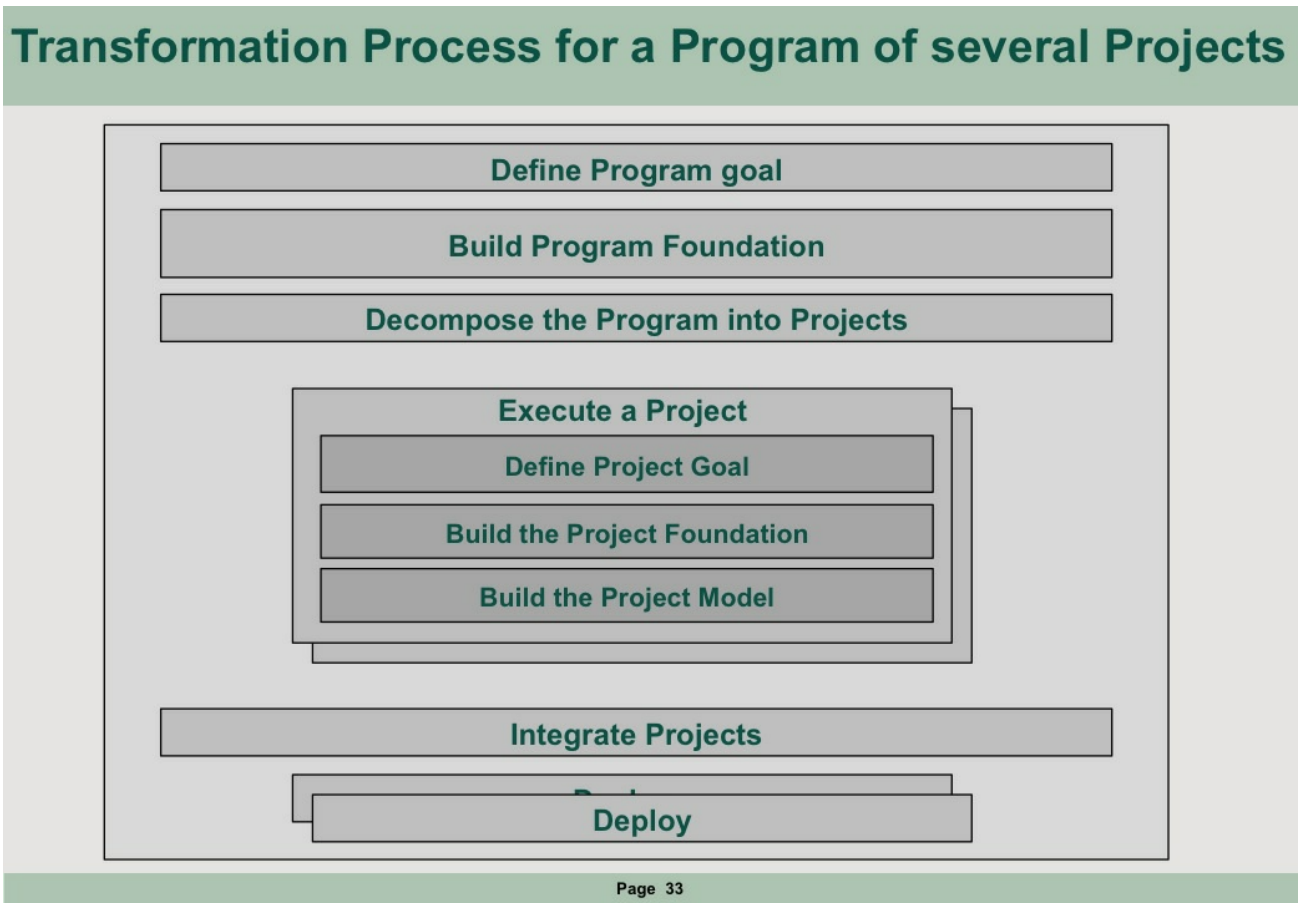
## 3. Transformation Model for a Project




This schema briefly describes the Transformation Process for a simple Project.

- Define the **Goal** of the Transformation Project (one or two pages maximum)
-
    - Define the **scope** and the associated volumetrics: geographic, Product Line, Process Domain, customer segment...
    - Define the **objectives** and the associated indicators to check the result obtained. Define the **Project constraints**, in particular regarding the budgets and timescales.
- Define the Foundation
- **Build** (or Modify) the Solution **Model** or the Offer Model
- **Deploy** the Solution or the Product Model
-
    - Migrate the information
    - Reorganize Organizational units and premises
    - Allocate and train the Operational Actors
    - Allocate, install, configure the IT-Actors
    - Prepare the hotline

# 4. Transformation Model for a Program comprising several Projects




When the Transformation ambition is considerable, we have to divide the Program up into Projects to avoid the tunnel effect and isolate the short-lived Projects.
We then distinguish

- the overall **Goal** of the Program and the individual Goal of each Project, contributing to the overall Goal
- the **Foundation** of the Program which will provide the overall coherence.

Moreover, we have to **decompose** the Program into Projects, which enables us to provide an initial estimation of cost and timescale for the whole Program. These costs and timescales are then specified Project by Project.

As the Projects are carried out by distributed teams, it is necessary to **integrate** the results of each team to ensure that the teams overall are working well while respecting the reliability and performance conditions.

Finally, we have to **Deploy**: either Project by Project, or by grouping the results of several Projects together.

Remark: for the more complex Programs, we can consider having **more than 2 levels** (Program, Project), but the principle is the same: to obtain a division in controllable Transformation units, **we need a Foundation**: Transformation Model, components and Architecture.

We recommend carrying out an **assessment** of the Programs and Projects:

- Has the **Goal** actually been reached? check with the indicators
- What lessons can be learned?
-
    - Can the Transformation approach be improved?
    - Assessment of the architecture
    - What new Components are reusable for the Foundation?

**All the Transformation Actors must share the same team spirit**


# 1. A Transformation program is a series of problems

Unlike the Operations, Transformation follows a broken line, each angle representing a new, unexpected problem that has to be overcome: we find out that the specifications are not complete, that a Transformer has been taken ill, that the last update of software tools did not run properly, that one team is behind compared to the others, that the performances obtained are disastrous, that the users do not like the usage of the new Solution...

At each hurdle, we have to find a remedy. The leader's talent is not enough to solve everything; it is essential that the whole team sticks together and supports the program's success and is involved in seeking the appropriate solution.

## 2. How to create solidarity in a Transformation team

- A leader respected by his/her team
- A shared Goal
- A realistic schedule: there is nothing worse than setting the team up to fail from the outset of the Program
- Celebrate each intermediary success without waiting for the end of the program
- Congratulate in public, reproach in private
- Live in the same premises and have lunch together: working from home avoids business trips, but harms the team spirit

## 3. The boss' Role is vital

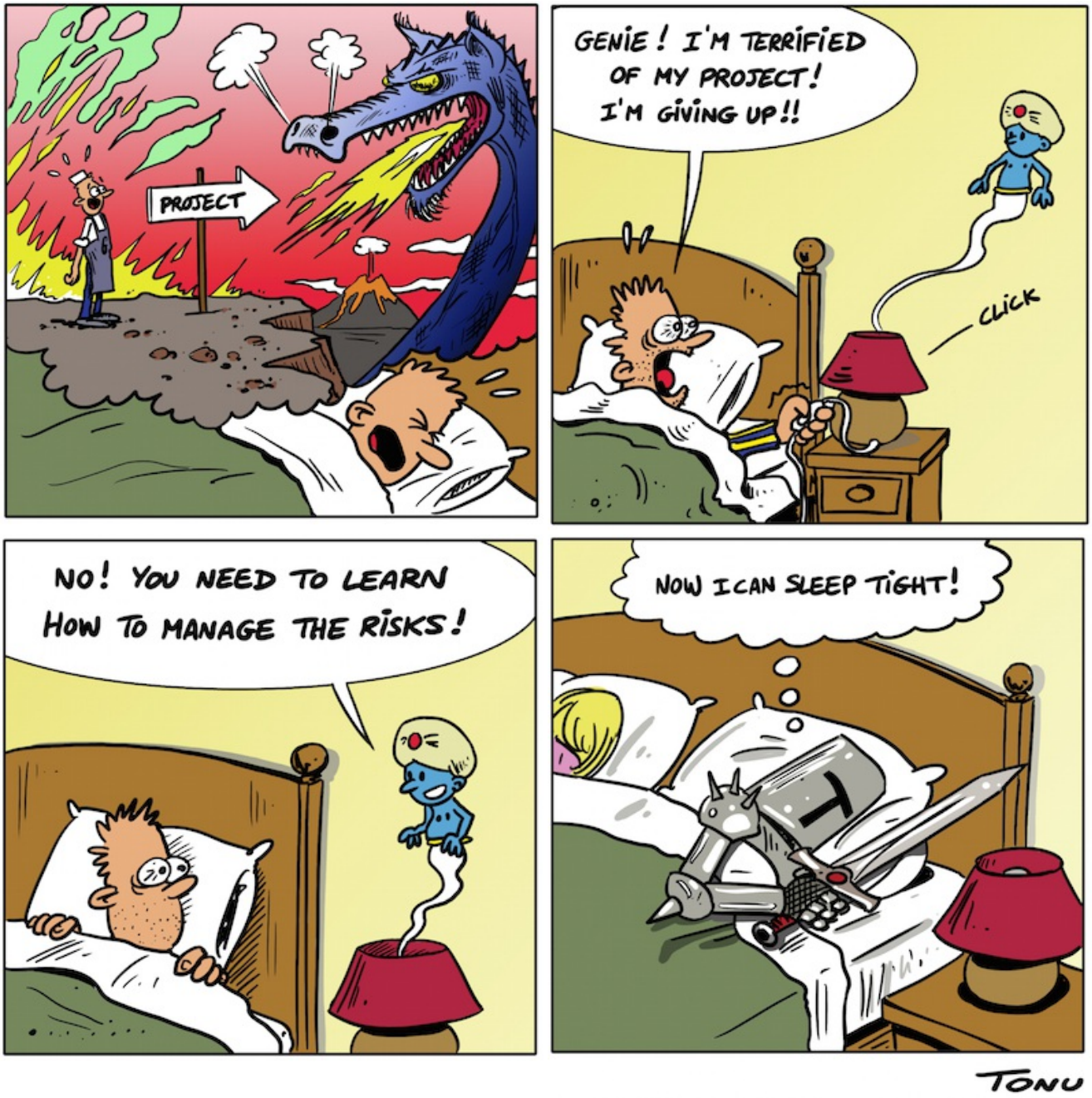
In reality, there are two "bosses":

- The Transformation **sponsor**: he/she defines the Goal, decides to launch the Transformation and allocates the budget, and appoints the Transformation project manager
- The Transformation **Project manager**: he/she builds the target Model that answers the Goal and deploys it.

The sponsor has a considerable role to play, not only in the decision-making, but also in the support that he/she gives to the Project over its duration, and in particular during the difficult phases. The sponsor is the one who will communicate to the Enterprise the why of the Transformation and who will encourage the project in the difficult phases.

The project manager will create the culture within the team: if he/she works hard, is transparent, competent, enthusiastic and knows how to take care of his/her troops, then the team will follow him/her.

# Risk management




## 1. Transformation is more complex than the Operations

We know how to Model the "Order" Operational Process, but we do not know how to Model the "Building a Solution" Transformation Process very well. There are many uncertainties:

- Uncertainties about the deliverable
- Uncertainties about the Solution Architecture
- Uncertainties about the course of the Project
- Uncertainties about the level of acceptance by the Operational Actors.

They require the Project manager to have the necessary talents to make decisions in an uncertain environment.
We have gradually managed to Model the **Transformation Management** Functions:

governance, scheduling, budget, resources, communication...
We still have not managed to properly Model the **Transformation Engineering** Functions: how to define the Goal, how to Architect a Solution, how to reuse Components, how to Build a Solution, how to integrate different Models, how to give flexibility to the organization...

### 1.1 Different Transformation Processes

There are **different Transformation Processes**: Build a Product Model or a new Solution, Build Foundations, Define a Road Map,...
Furthermore, alongside the Foundation Projects and Solution Projects, some Enterprises have created the category "Innovative Project". These Projects entail a greater part of **risk**. They are often eliminated by the classic Governance Processes which do not encourage risk-taking.
This type of innovative project can represent more than 10% of Transformation Projects. They undertake a different governance and apply a more flexible Transformation Process.

### 1.2 What are the risks?

Actually, the main risk is that of uncertainty: we are not sure that a Transformation project will succeed.

#### 1.2.1 The main risks

- Unrealistic or unstable goal
- Lack of support from the Sponsor in difficult moments of the Project
- Dispersion and average quality of the Transformation team
- Reproduction of what already exists
- Tests are carried out at the end, rather than throughout, the project
- Generalization with tests in a pilot site
- Ignoring the Deployment and acceptance by Operational Actors
- Bureaucracy and slow decision making

#### 1.2.2 The difficulties of a Transformation program may see investors lose confidence

A business model's vulnerability comes from how quickly information travels. For a large Enterprise, one piece of bad news can result in several billion dollars being wiped off its market capitalization: see the example of Tesla (in French).

## 2. How can we protect ourselves from the risks of Transformation?

We cannot guarantee the success of a Transformation Program.
We can only increase the likelihood of success and identify the risks at the source.

### 2.1 We can increase the likelihood of success

By actions such as:

- An **experienced program management team**: if a team has previously succeeded in a Transformation Program, then it has a strong likelihood of succeeding again. We have to know how to retain these talents in the Enterprise.
- A smaller design team, but a high quality one
- Total transparency: do not hide problems
- A fast decision-making process
- Splitting the Program up and having Projects manageable by small teams

- The ability to dynamically arbitrate between the needs and the technical possibilities: we should not brace ourselves against pre-agreed specifications if we discover less expensive alternatives to solve the same problem
- A continuous desire to not reproduce what already exists.

## 2.2 Identify risks at the source

To identify the risks as early as possible, we use different techniques:

- A precise (but modifiable) schedule that enables us to see any delays, often synonymous with problems.
- A project-tracking system, which highlights the exceptions and not the things that are going according to plan.

Remarks: project methods like PMI or CMMI detail, over many chapters, aspects of risk management: identification of risks, defining coverage actions, risk acceptance, etc.

## Divide a program up into projects


# 1. The difficulties of an overly-complex Program

A complex system is difficult to build in view of the number of interactions between its elements. We often talk about a "can of worms" to refer to complex IT Solutions. Each time we change one part of the Solution, we destabilize the overall solution: the "non-regression" tests, which are designed to check that the Solution remains robust despite the most recent changes, sometimes turn out to be far more expensive than the modification itself: we can carry out a modification in a week and then execute non-regression tests during 2 months. When a problem appears, correcting it takes even longer as we have to search for the cause in too big a whole.
The only method consists in architecturing the Solution in Modules so that the number of interactions is more limited.
This strategy does not just have an advantage of overall simplification. It also enables us to deliver the initial concrete results quickly to the Operational users, it avoids the tunnel effect and **gives**

**credibility to the Program** because the initial Projects deliver Value fast through the first Modules. It is also a way of **gradually testing** the viability of the Solution without waiting for an overall delivery at the end of the Program.
But how do we divide up a Program into Projects?

# 2. How to divide up a Program into Projects

## 2.1 Build the framework before the Solution

Each Project of the Program will be easier if it is part of a well-structured framework. The clarity of scope, the preciseness of the interfaces with the other Solutions, the reuse of Information access Functions are all assets in focusing the energy of the Project manager on his/her Model and not on the environment.
Before beginning to build the initial solution, we have to have a framework at our disposal into which the solution will fit. This framework has two dimensions:

- the **Enterprise Architecture** which defines the Solutions map and the relations between the Solutions
- the **Reusable Components** to build each Solution

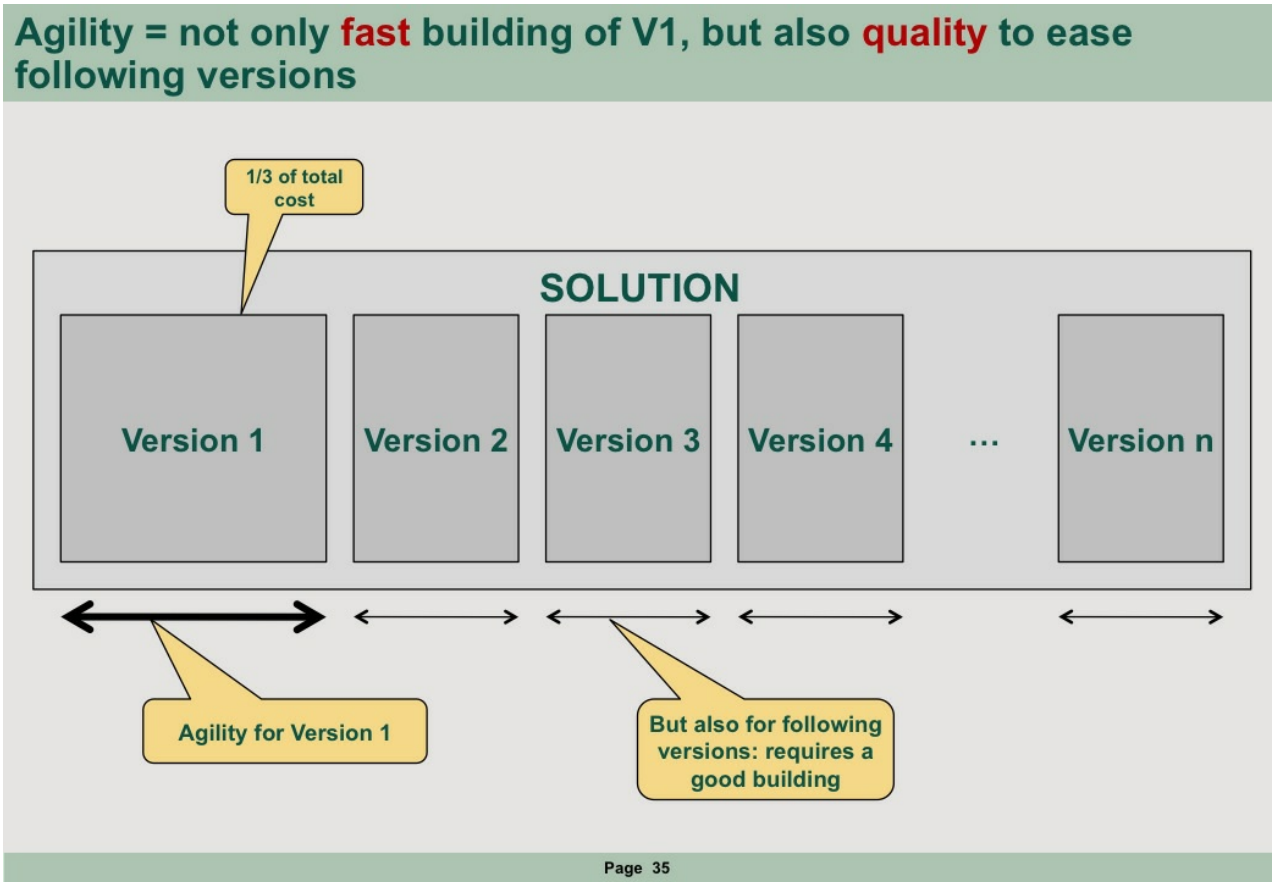
## 2.2 Simplify the existing system little by little

When the complex Solution is already in position and we want to simplify it, another possible strategy is to gradually simplify the Solution as the Projects advance.

To define the **road map** of gradual simplification, we have to isolate little by little the information access components to make data evolution independent of software evolution. Then we gradually isolate the reusable Business Functions. From there, deduce standardized interfaces between Solutions.
This strategy has been defined in the CEISAR white paper entitled "**Simplify Legacy Systems**": we recommend that interested readers download this white paper from www.ceisar.org.

## 2.3 The first Version of a Solution is the most difficult

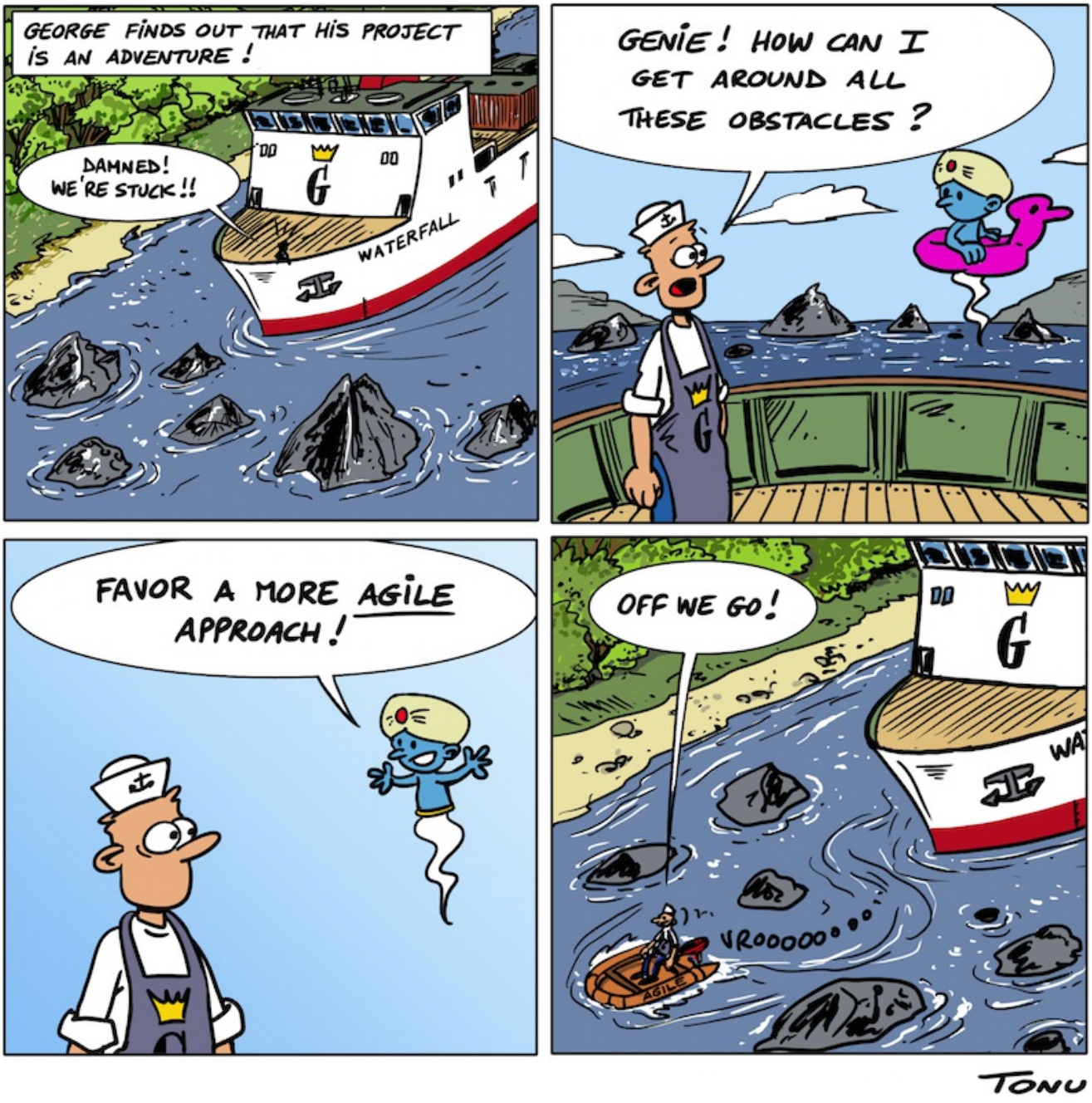
Creating Version 1 of the Solution is more expensive than each of the following Versions as it includes Building the Solution and the Foundation Architecture, which will be kept by each of the following Versions.
But, in view of the life span of the Solutions, the total costs of Versions 2 to n represent 2 to 3 times the original cost of Version 1. Agility is therefore not only the art of Building the first version of the Solution **fast**, it is also the art of Building, right from this first Version, an **efficient** Solution Architecture enabling the following Versions to benefit. Agility at a given time (T) must not mortgage agility at T+1: this is the definition of sustainable development applied to Information Systems.




## 2.4 Give a "Business" meaning to the division

It is also important that the project milestones correspond to "business stages", that is to say stages that make sense to the business and not just to IT. Otherwise, it will be very difficult to get the buy-in of business partners to the project. We must therefore be more than capable of "telling a story" around the project, and talking about the business value brought at each step.

# Favor an agile Approach




## 1. Agility: a major competitive advantage

### 1.1 Reactivity and Agility

**Reactivity** is the art of Operating fast and well, whereas **Agility** is the Art of Transforming fast and well.

### 1.2 Being able to "Move Fast" is more important than "Seeing ahead"

As it is impossible to guarantee that the innovations initiated in the Enterprise will be better than those of the competition, it seems that the safest strategy to increase efficiency is not only to

innovate ourselves, but also to perfect more **agile** Transformation Processes than our competitors, to be able to rapidly implement innovations coming from elsewhere and to correct our weaknesses. In this sense, Agility becomes the **main quality of an Enterprise**: identify our weaknesses and benefit from our Agility to adjust things.

To be agile, the Large Enterprise has to acquire start-ups, copy successful innovations from certain competitors, and obtain a real competitive advantage, **not because it "sees ahead", but because it "moves fast"**. Strategy is no longer based on increasingly unpredictable forecasts of the future, but on seizing opportunities that the Enterprise knows how to leverage more quickly than its competitors.

# 2. How can we be more agile?

## 2.1 Agility: a question rarely tackled

The Operational Actors identify domains where they are looking for more reactivity and better quality of service. We allocate budgets to these projects which aim to optimize the Operational Model. On the other hand, we struggle to identify that we can also improve the Transformation Processes to give us more agility and more quality. The main reason is that we do not think that we can Transform more rapidly and, anyway, the budgetary processes have no room for this type of investment.

## 2.2 Adopt an agile Approach

Do not force the Business to specify everything, proceed by successive versions, put the emphasis on the quality of the Building of the Solution so that modifications in future Versions can be accepted fast.

## 2.3 Accelerate the building of Models

### 2.3.1 A global vision of the architecture

Rather than the **exhaustive** inventorying of needs, favor Building Solution Architecture that is able to integrate successive increments as, and when, the needs mature.

### 2.3.2 Component-based Building

A **new form of reuse**: no longer only reusing Software package Solutions that we can adapt a little, but also reusing Components that we can assemble to build innovative and differentiating Solutions, especially for the Business.

### 2.3.3 Suitable tools

The Agile approach is more efficient if it relies on a **single Model** of the Solution: any modification, be it from a Business or an IT change, updates the single Model and instantly modifies the views offered to each person. This is what we refer to as "Round Trip" tools.

### 2.3.4 The power of Configuration

Isolate the parts of the Solution that change frequently, to enable them to be modified as a Configuration: it should be possible to change a price, an eligibility rule, a commission... without having the skills of an IT developer.

### 2.3.5 Care taken with ease of usage

As users look for a uniform usage, the best way of respecting this homogeneity is by making standardized **usage components** available, which, when reused by Solution builders, will guarantee the standardization of usage much better than making **documentation standards** available.

## 2.4 Attitude

As also mentioned, other factors affect agility: multidisciplinary team and not separate teams, knowing how to take risks, suitable governance, not a lot of bureaucracy...

# 3. From a linear approach to an agile approach

## 3.1 A linear approach for Commodity Solutions

To acquire Commodity Solutions, Enterprises have developed a Linear Approach which favors security or reliability over agility.
This slowness of the internal Transformation, which results in an increase in the costs linked to it, is characterized by the drawing up of a Contract which contains the Business Model to be translated into an IT Model, the separation of Business and IT teams, and a multiplicity of Roles with their relations to be managed. This heaviness enabled a **Software package** industry to rapidly develop, positioning itself as an alternative to custom Solutions, which are too long to implement.

## 3.2 An Agile approach for Business Solutions

In the **Agile** Approach, we do not detail the whole Business Vision of the Model before beginning the IT Modeling. The definition of the Functionalities is progressively refined **iteration** after iteration. This approach is preferable when the specifications are uncertain and we want to build Evolutive Solutions rather than final Solutions, which applies more to Business Solutions than to Commodity Solutions. This approach is quicker, it removes the tunnel effect, it does not force the Business to define its full Model before handing over to IT, it makes for a smoother Business-IT relationship, it allows for gradual verification, it lets Business and IT come together in one mixed team, responsible for the Solution, but it can only be chosen **if the global Solution Model supports later additions.**

### 3.2.1 How do we succeed?

To succeed, we have to follow the following rules:

- In the analysis of a Business Process, separate the Core Business and the **Organization**: we define the "what" before we define the "who"
- follow the **sequence** "Objects, Functions, Processes": to analyze the Process « the salesperson sells a Product » we first have to analyze what is a « Product », then the Action « sell », then define what is a « salesperson »
- reuse a maximum number of **Components**
- plan for all the elements of the Model to evolve (all Model elements must be integrated into **Version** management)
- ask the **best** "Modelers" to focus on Building Solution or Foundation Models; do not overload them with pure management tasks, they are a rare resource
- have the quality of the Global Model **certified** by experts
- limit functional specifications by **date** and not by functional scope.

### 3.2.2 Know how to stop a version

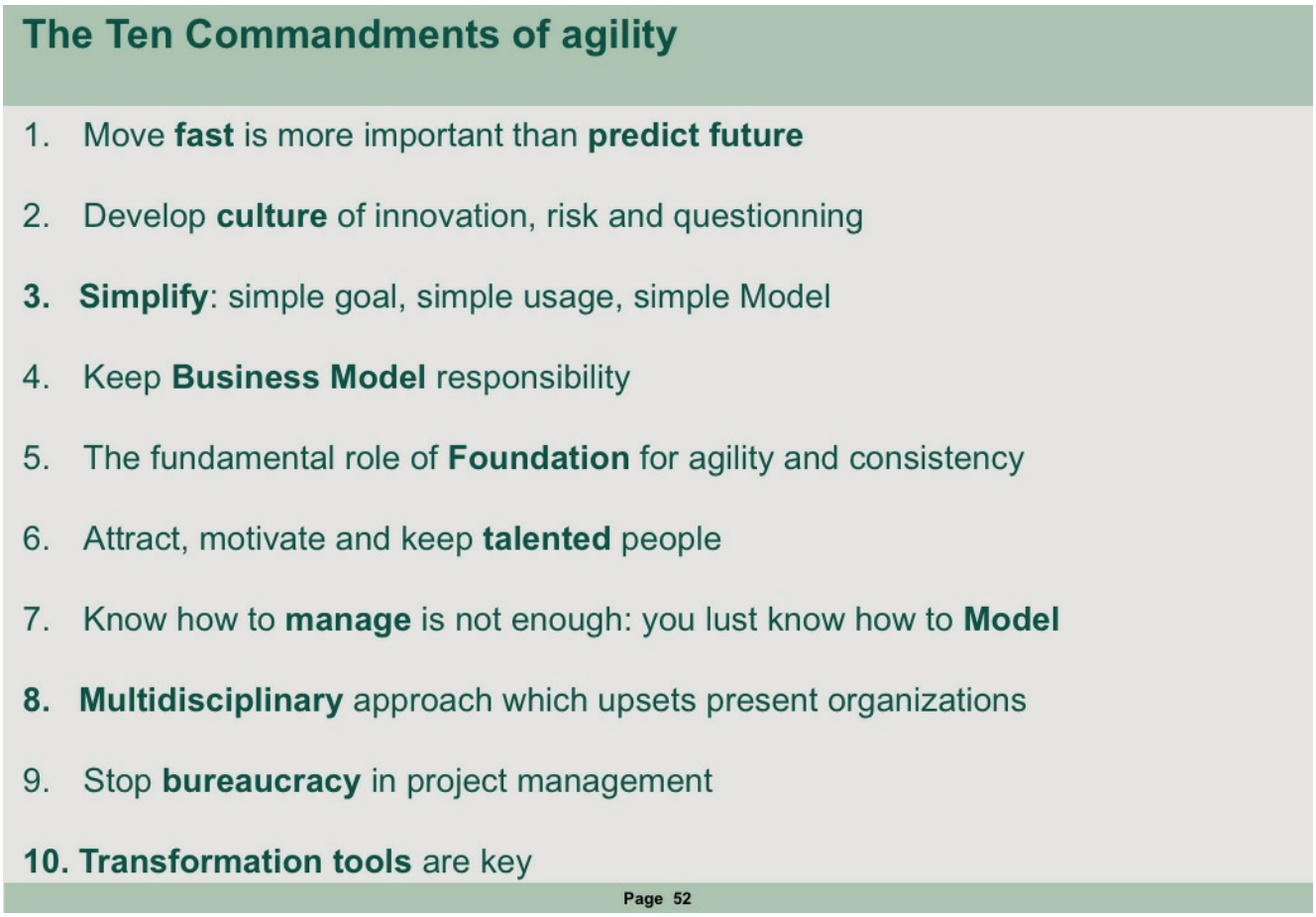
As we do not define everything that has to be done before starting, it is important to define a rule for stopping a Version.
We have to apply the following rule: "as it is easy to go to a new incremental version, deliver the first version quickly so as to offer a first level of service to the Actors".
Naturally, if it is to replace an old Solution, we must at least keep the Functions offered by the old Solution.
The right rule is to set a realistic **timescale** for the first version, which means that the project Owners have to appoint a leader who is able to **select** the progressive requirements to meet the deadline. This principle of "sobriety" is the key to a successful, agile approach.

# 4. The agile method to design Goods

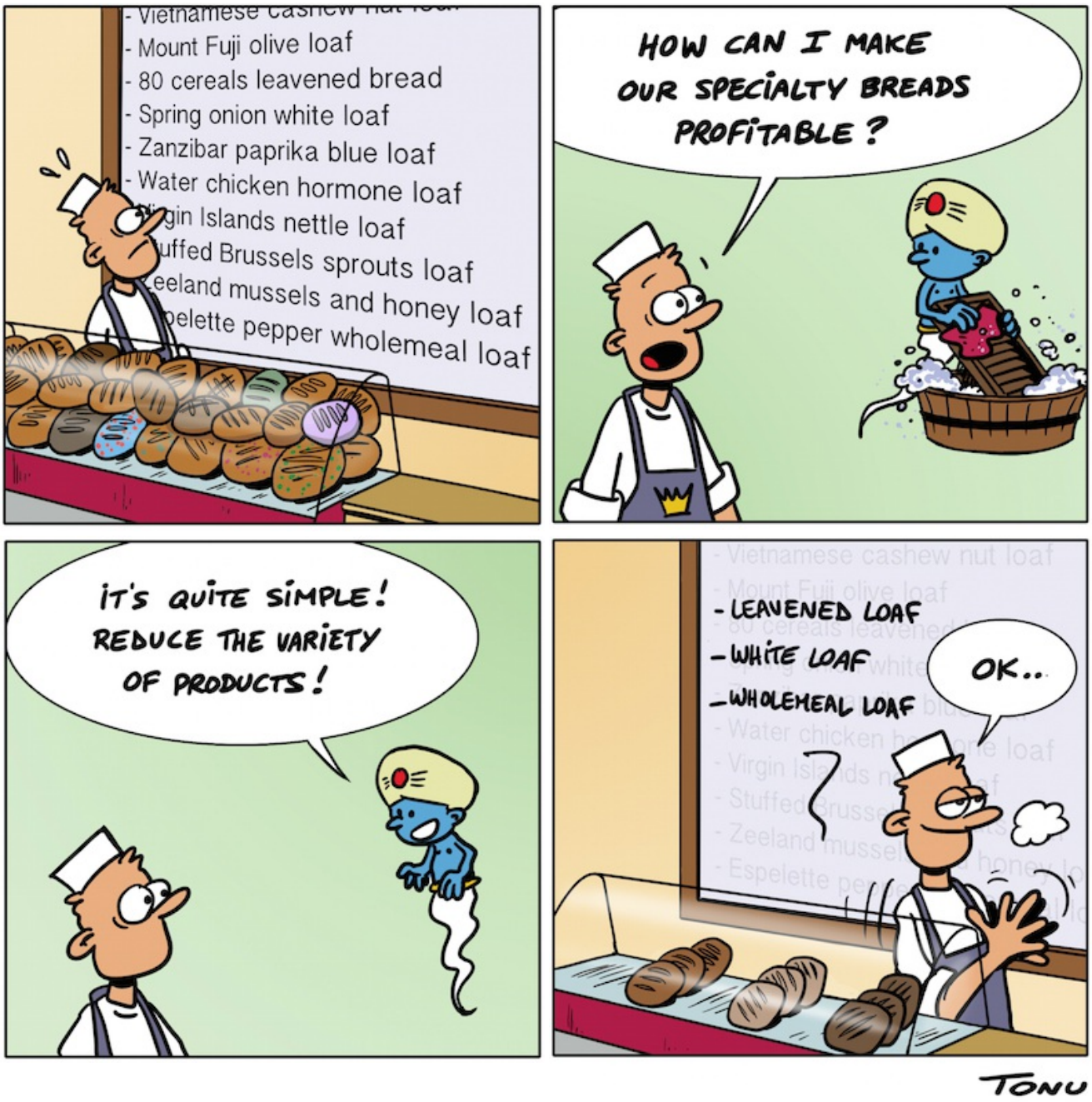
The agile method is massively used today to Build Solutions or Services.
Some have also decided to Build Goods Models, taking inspiration from agile methods.

See this article.

In a nutshell:



# Reduce the variety of Products




## 1. The dilemma of multiple Offers

How do we resolve the dilemma between "offering a large variety of Products in order to be sure of seducing all customers" and "reducing the number of Offers to simplify the running of our Enterprise"?

It is easier to multiply Offers than to reduce them: for marketing, adding an Offer means a new customer target or the modification of an existing Offer and thus an increase in revenue.

But this multiplication of Offers generates complexity:

- Complexity of Producing many Offers
- Distribution complexity: are the sales reps able to integrate all the Offers? How do they present them all?

- Complexity of the After-sales service
- Complexity of the software which supports everything
- Stock management complexity

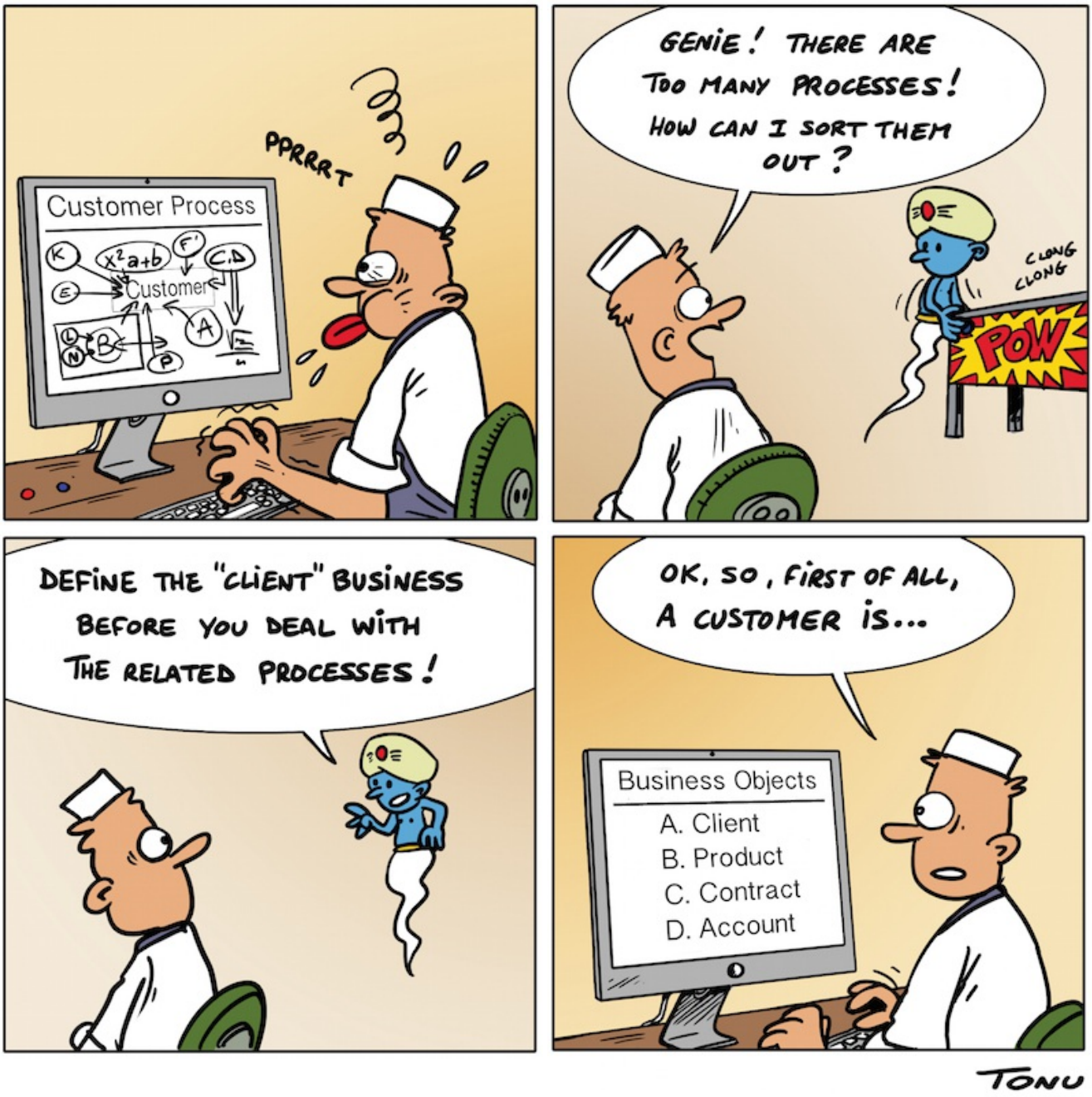
In short, the accumulation of Offers generates complexity, which has an Operational cost and harms Agility; it also breaks up the full resources of the Transformation, which is a rare resource.

This is why some enterprises make a point of honor of reducing their Offers. Apple, the company with the largest stock market capitalization, is a good example having reduced its offer to a smartphone, a tablet and a computer. Variants of these products exist, but they are very limited in number. The contrast with Sony is considerable; they propose computers, smartphones and tablets, but also portable CD players, radios, cameras, televisions, voice recorders, video cameras, medical material, videoconferencing systems,... and furthermore, they produce content.

## 2. Focus on what we know how to do best

We cannot be the best in all products.
It is not because we are able to Produce and Distribute a Product that we should: we have to ensure that we will be capable of doing it better than the competition over the long-term and that we focus our energy on the essentials.
One of the roles of the CEO of the Enterprise is knowing how to **say "no"** to the very many enticing initiatives put forward by the different organizational units of the Enterprise: this was Steve Job's strength.

## The Business Objects are defined before the Processes




# 1. Current approaches emphasize Processes and not Objects

We automate Actions: if we want to computerize all or part of the way the Enterprise works, we look at how it Operates today:

- How the worker Produces a Product
- How the seller distributes the Offer
- How the HR director recruits the employees

The description of each of these Operations breaks down into 3 parts: Actor, Action and Object (we

were taught that a sentence was made up of a "Subject", a "Verb" and a "Complement").

| Actor | Action | Object |
|---|---|---|
| Worker | Produces | Product |
| Seller | Distributes | Offer |
| HR Director | Recruits | Employee |

The observation focuses first on what we see: the Actors.
Then on the fact that these Actors get active: the Actions.
The Object, which is the focus of the action, only appears at the end.

This is why it appeared quite natural to analyze the Actions before the Objects. Most current Approaches are based on analyzing Processes.

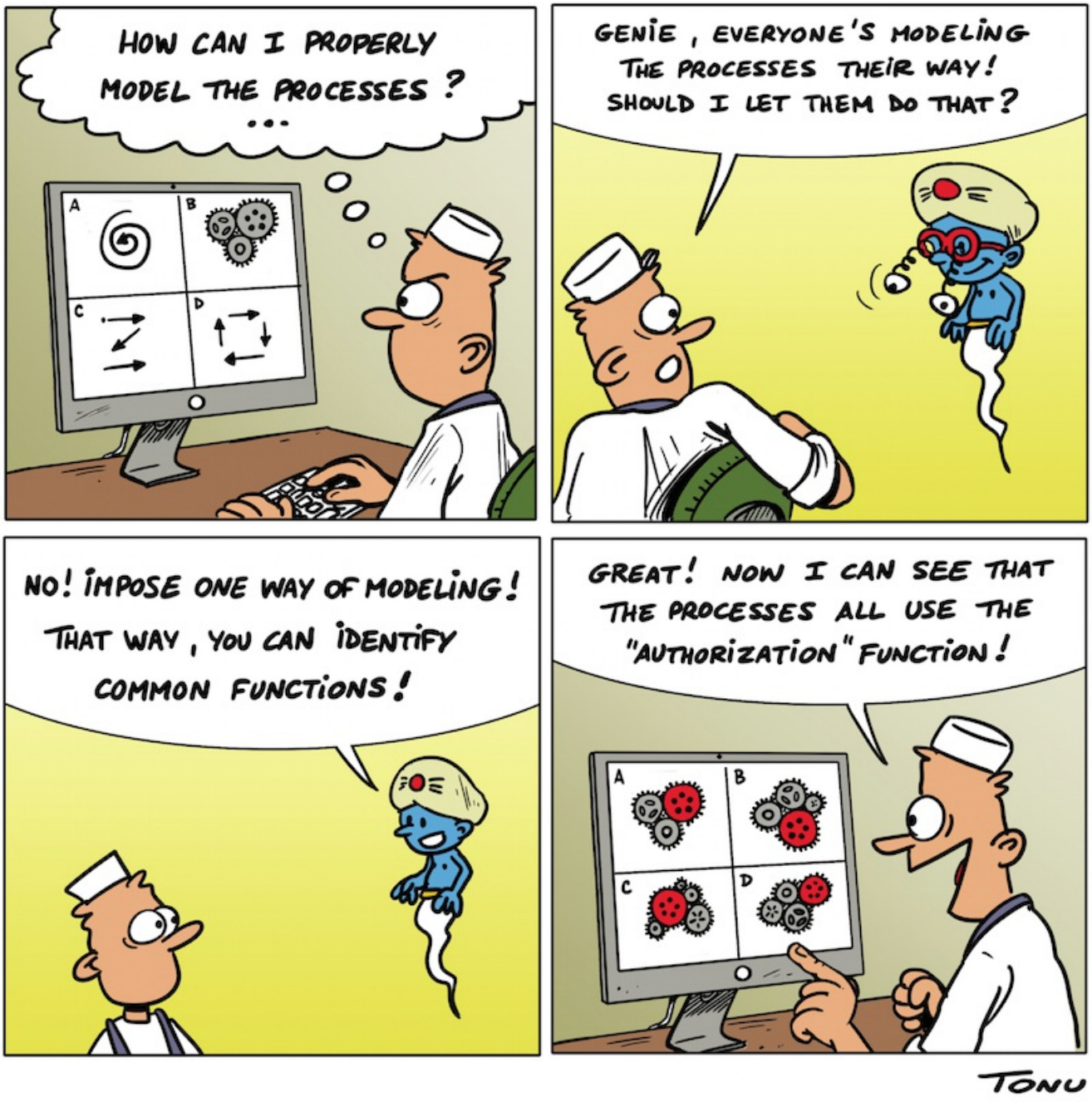
## 2. Begin by analyzing the objects, then the Actions and finally the Actors

Our recommendation is to do the exact opposite: begin by analyzing the Objects before the Actions.
First, because there are far fewer Objects than possible Actions on these Objects: it is therefore one way of **sorting out the Actions**.
Then, because it is impossible to precisely define "create Customer" if we have not defined "Customer" beforehand. Processes are only **accurate** if they rely on well-defined Objects.

Finally, we recommend dealing with the Actors only after the Actions have been dealt with: the fact that some Processes previously carried out by Enterprise staff are today executed by Customers or partners, requires flexibility in terms of Actors: the business Process is the same but the organization and the role of each person may evolve.

This recommendation seems simplistic, but it has profound consequences on the robustness of the Enterprise Model: we can only recommend to first rely on a **Business Glossary** (see previous scenes).

# Model and optimize a Process




## 1. What is a Process?

An Enterprise must act to bring Value to its Customer: it has to Produce Products, distribute Offers, manage its Resources and manage how it operates.

**But how do we distinguish between a Process and a Function within a Process?**
A Process is a sequence of Functions that are executed from an independent event: a customer request, or a partner or internal request, or a date...
Recruit an employee, sell an Offer, and buy Components are Processes.
On the other hand, "calculate a price" is not a Process, but a Function inserted into a Process: it is used in Processes like "Take out a Contract", "Make a quotation" or "Bill monthly".

**But how do we distinguish between a Process and a Process domain?**

The "Human Resources management Processes" Domain is made up of different Processes, such as:

- Process "Recruit an employee"
- Process "Evaluate an employee"
- Process "Train an employee"
- Process "Transfer an employee"

Each of these Processes is triggered by an independent event.

This is what enable us to distinguish between Process and Process Domain.
As an example, "Manage staff" is not a Process as there is no unambiguous independent event, it is a Process domain.

# 2. How do we model a Process?

A Process is a suite of Functions executed by one or several Actors helped by Information. We therefore have to Model the Functions, Actors and Information.

But beforehand, we have to define the start and end of the Process:

- Start: what is the **trigger event**, what are the **input elements**?
- End: what are the output elements and **who** are they for?

The Product of the Process is a Good or Information if we consider Production Processes. But if it is about Producing a Service, the Product of the Process is a change of State on the Object that the Service concerns: premises cleaned or new haircut or Goods transported to another place. The beneficiary of the Process can be internal or external: a Customer or a Partner.

**The sequence of Functions** is not necessarily sequential. For example, "take out a contract" may require an expert's involvement if the contract is for a large amount. Moreover, a Function can link other Functions. For example, the Function "check customer eligibility" calls on the Functions "get customer age" and "check that the customer is not already known as a bad debtor". We must therefore represent the Function chain algorithm.

Functions rely on **Information**: Product, Customer, Contract, Account... information. We therefore have to represent this Information.
Finally, these Functions will be executed by **Actors**: Human-Actors (employees, partners or customers) or IT-Actors.

Different **forms of** Process **Modeling** exist that respect these principles: it is recommended that we **choose one** and only one within an Enterprise to facilitate the dialog between the different Transformation Actors.

# 3. A single Operational Model with different views

When we have to transform a Process Model into software, there are 2 methods available:

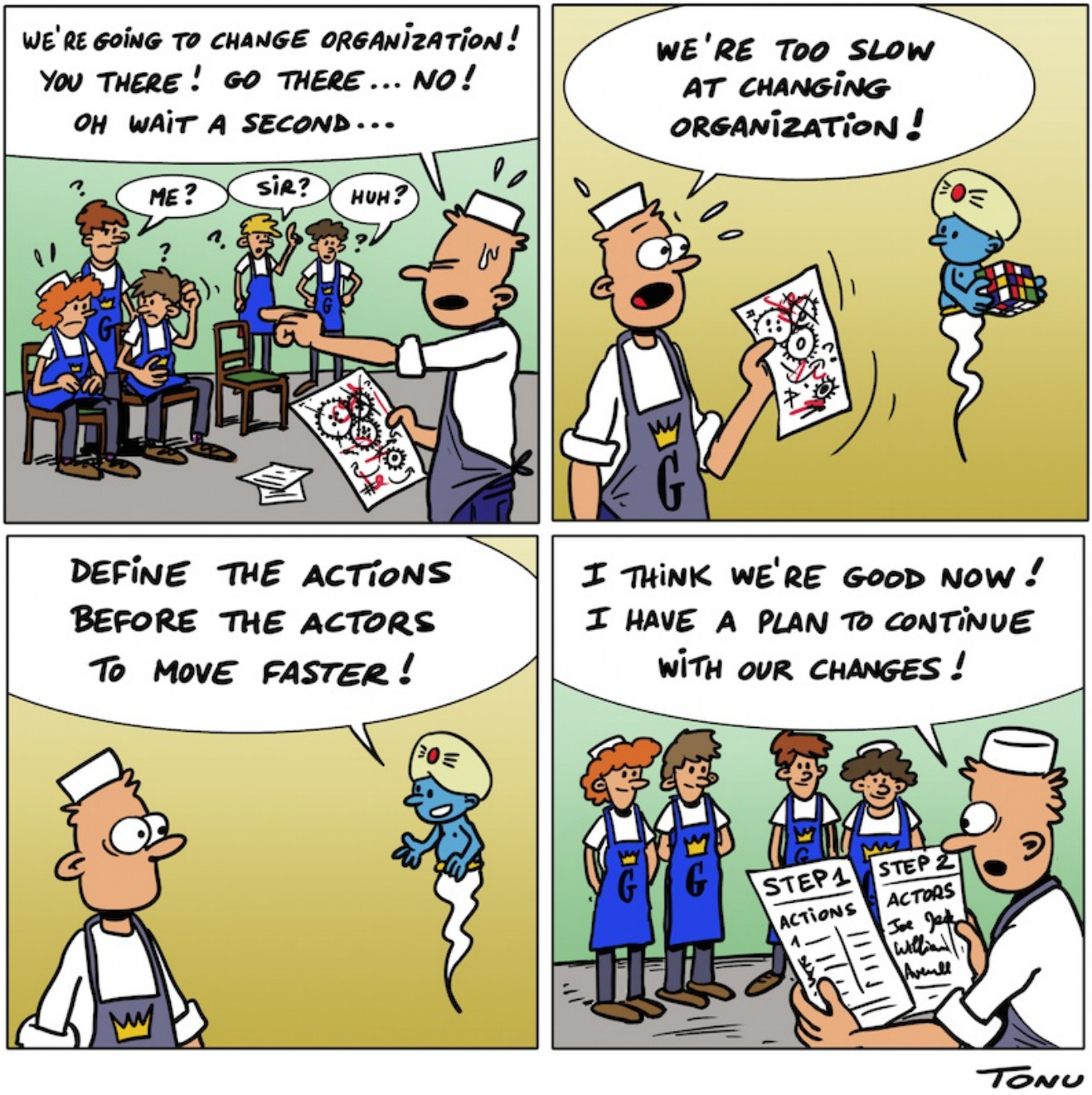
- either we use a Modeling tool specifically for Processes, and then the IT developers have to translate the Process Model into software
- or we use a single tool: we Model the Processes in the tool and the software is automatically generated

It is clear that the second type of tool is more efficient as it removes human intervention (the translation of the Process Model into software): we gain in time, money, reliability and Agility. In particular, we no longer need to check, through testing, that both Models are coherent.
We prefer using a **single Model** rather than 2 separate Models for the Business and IT parts, and we provide **Views** of this Model adapted to each Actor: it is a valuable tool for a multidisciplinary approach.

There are not yet any universal Tools that transform the whole of a Business Model into an IT Model, but progress has been made on some parts of the Model:

- Model Information and automatically generate the IT Data Model
- Model the business Rules directly in a rules engine
- Model a Process and automatically generate the navigation and allocation
- Model the presentation and automatically generate the GUI Model
- Model the Business Intelligence deliverables and automatically generate Information production

**The same Process Model must support different forms of organization**




# 1. Business Process and Organized Process

A Process for taking out a contract has to link Functions such as:

- Get customer information (already stored or new)
- Identify the Product purchased and its options
- Check that the Customer is eligible
- Calculate the price
- Print the contract
- Sign the contract
- Bill
- Pay

- Deliver

This series of **Actions** relies on Product, Customer, Contract and Account **Information**.
In this list, we do not have to determine which **Actor** will carry out the different Functions.
This is the **Business Process**.
Different forms of Organization (and therefore different "Organized Processes") can be defined to execute this Business Process:

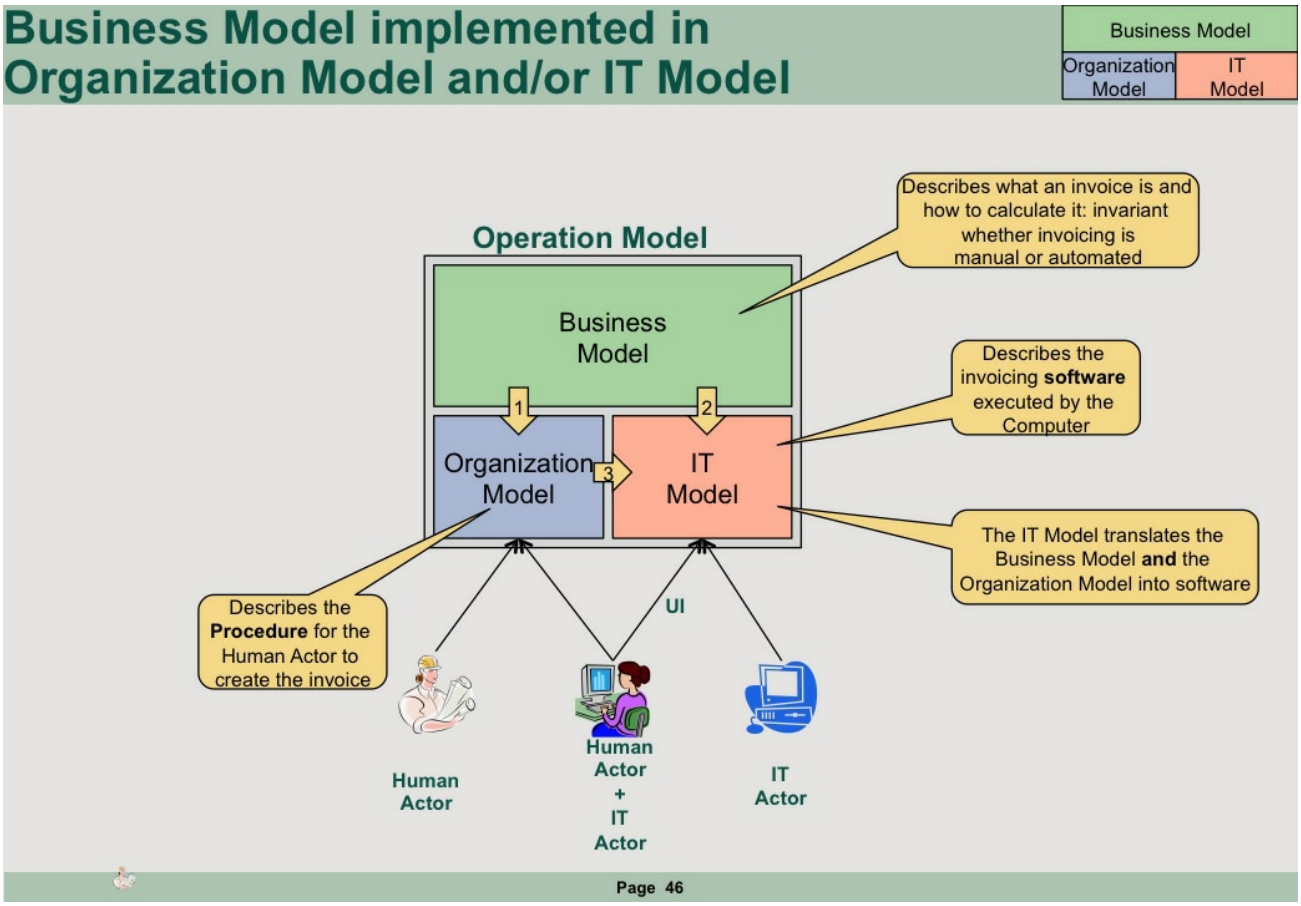
- For example, the order can be entered in the enterprise branch by the employee, or by a partner who distributes the product or by the customer on the Internet
- For example, the delivery can be carried out by the enterprise's delivery department, or a delivery partner or the Post Office.

To summarize, "**Business**" is anything that defines the Business independent of the Organization chosen by the Enterprise: definition of the Products, Information on the Customers, Partner choices, pricing rules...
"**Organization**" refers to the organization chart, the Role of the Actors, the authorizations, the responsibilities, the allocation of Activities...
This fundamental difference should help us to build a Solution that leans on the Business and supports the different Organizations (successive or parallel).

# 2. The Approach

The Approach is simple: we must not rush into defining who does what, but focus on the Actions and Information of the Business Process.
Then, as a second step, choose one or several forms of organization.
Finally, computerize the Business Process and isolate the Organizational part so that we can rapidly change the organization without losing the investment already made.

The following example shows how to analyze billing:



A Software Solution lasts for 20 years, Organizations are in perpetual movement: how do we analyze the Processes so that the Software deduced from them easily supports the different successive or simultaneous Organizations?

If you ask Operational Actors to describe the way they work or would like to work, they will naturally describe what they experience on a daily basis: **Actors** execute **Activities**. They will rarely talk to you about Processes or Information.
From what the operational Actor says, the approach of the Solution Builder should be the following one:

- We define the **Business Model**: Information, Business Functions and their sequence

    1. Define **Business Objects** such as "Customer", "Product", "Order", "Delivery": these are the most stable elements of the Enterprise Architecture.
    2. Define the life cycle of each of these Objects to show the **Core Business Functions**: "create", "control", "modify", "query", "set price", "debit"...
    3. Define the **Business Processes** and decompose them by reusing the Business Functions defined in step 2 and by adding new ones that we can only find by analyzing the Process requirements.

- We define the different forms of **organization**: who does what (Human-Actors and IT-Actors)

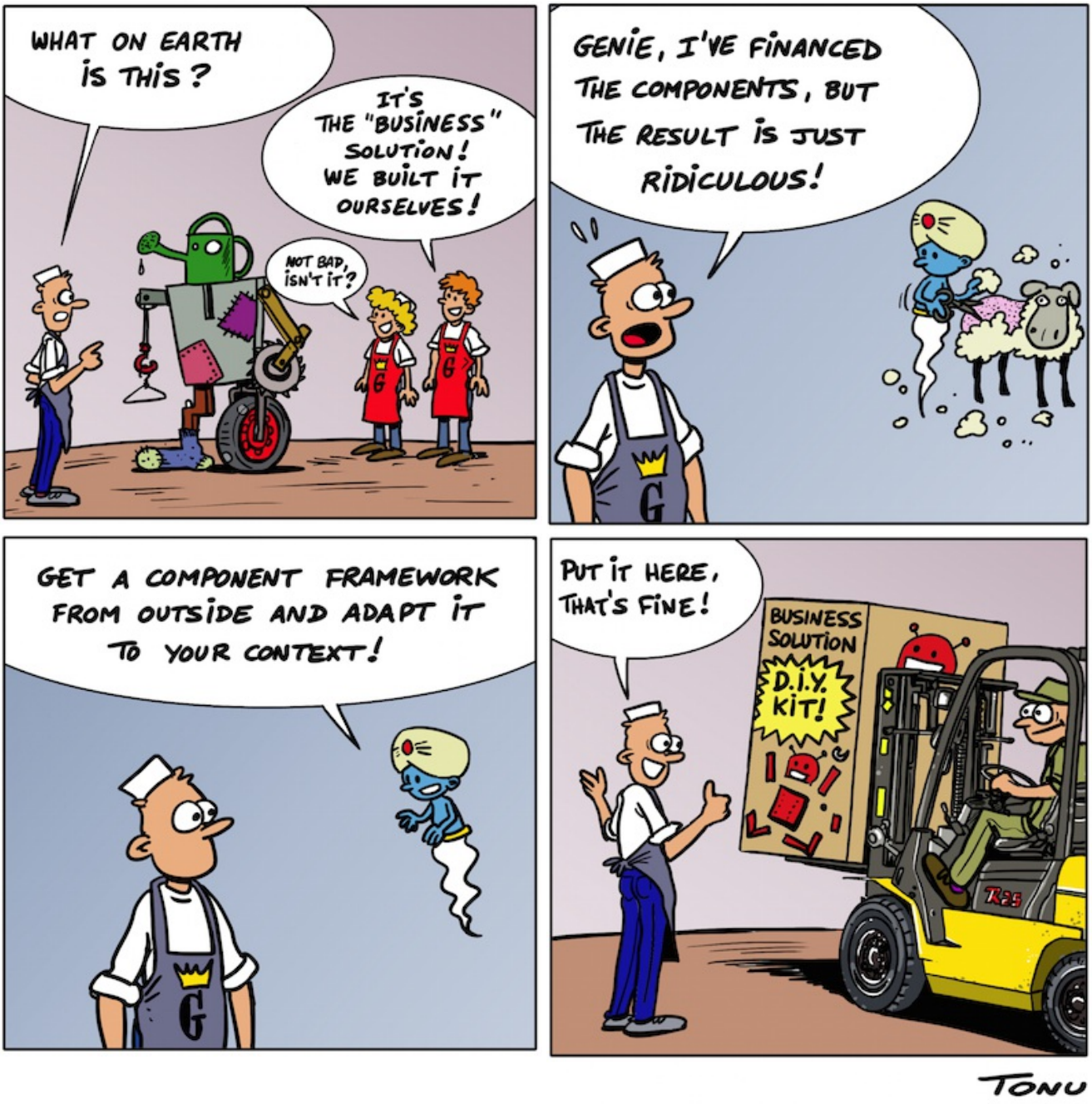
    4. Define what **Roles** are attributed to the Actors or Organizational units that group them together.
    5. Define the **Organized Processes** divided up into **Activities** (an Activity is executed by only one Actor) to allocate them to the Roles defined in 4. Decompose each Organized Process into Business Functions and Organization Functions (the latter are part of the Foundations)

- We build the **software** by separating the business part from the organization part.

The interest of this 5-step Building approach, separating Business and Organization, is multiple:

- The Solutions, based on the Business, are perennial and can support **changes in organization**
- We make the **reusable** Objects and Functions appear because we analyze the Business Objects before the Processes.
- We can begin Building the Solution **without waiting** for the whole Organization to be defined.
- The Solution can easily adapt itself to external partners.

## Build and support Components well




# 1. It is difficult to build Components

How do we take the needs of all the potential customers of the Components into account when we do not know them all?
Experience plays a considerable role: those who have already built components know the requirements better. This is the reason why it is recommended to check on the market if an efficient component Framework exists before launching into building your own one.
As the components are used by everyone, the consequences of a component evolving are more difficult to predict. How do we ensure that the existing Solutions are not disrupted by new versions of the components? Should we give priority to:

1. **forward compatibility**: we ensure that the new versions of the components will not disrupt the Solutions that already use them, but we limit the evolutions of the components.

2. **growing efficacy**: we do not hesitate to overhaul a component because we have developed a new design, but forward compatibility is not guaranteed.
The Components enable us to modularize Solution Building. But decomposing into modules means many successive calls that may impact on the **performances**.

Seeking the **causes of an incident** are more complex as they can arise from the Components, which call each other.

# 2. It is difficult to use Components

Even if the Components are well built, it is not enough: the Solution builders must use them.
If the refusal is "political" (we want to keep our independence), we have to introduce the right governance (see the chapter on Transformation governance).
But, the refusal may be technical:

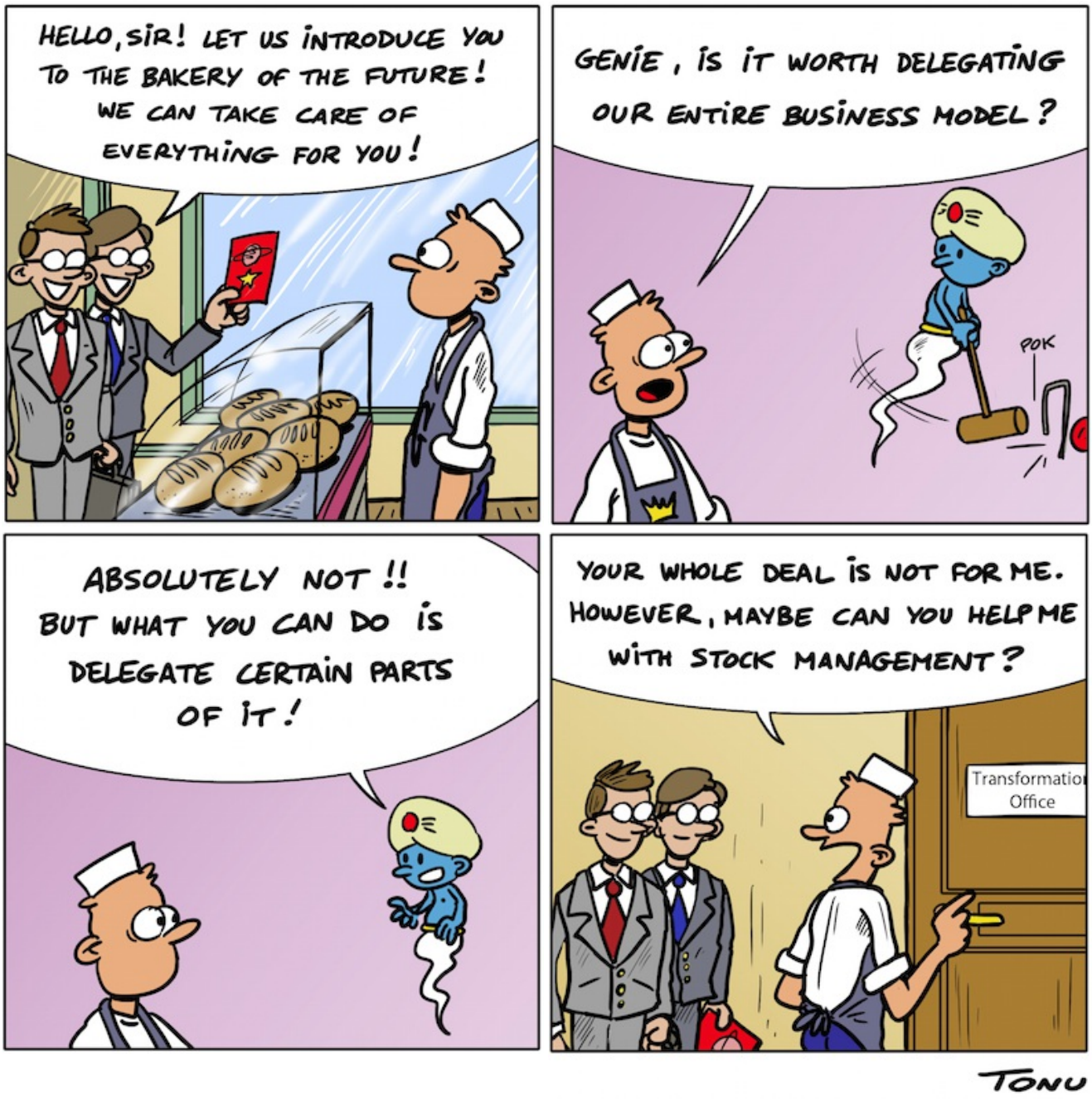
- Too many, we do not know how to find them again
- We do not understand what they do
- We do not know how to use them
- We do not know if we should install the new version

We must therefore introduce rules for building Components.

# 3. Some rules for building components

- Components are small: 1 or 2 pages of code. Otherwise, we have to decompose them into several components.
- Components should reuse components.
- Components are versioned: not only in the documentation but also when calling the Component.
- Think of the customer of the component when we name it.
- Organize a Component catalog.
- Give examples of use.

# Keep responsibility for the Business Solutions




## 1. Why keep responsibility for the Business Solutions?

One trend has dominated these last years: IT developments must be outsourced.
The main reason is the difficulty of building software in-house: too late, too expensive, too incomprehensible!
The reason given is more often than not "my job is automobiles, not IT" or "my job is insurance, not IT".
We could say "my job is automobiles, not human resources management" as well; so let's outsource human resources management...
Rather than incisive sentences, we need to reflect. There are 2 parts to IT activities:

- The Transformation: build the Model for the IT-Actors (what we call software)
- The Operations: operate this software on the IT-Actor platforms.

In view of the increasing role of IT-Actors, a growing proportion of the Enterprise Model finds itself in software. Should the Enterprise keep control of this software, or is it in its interest to outsource it?

The Enterprise uses Commodity Solutions, similar to those used by its competitors, and Business Solutions, which can give it a competitive advantage.

Commodity Solutions are computerized in the form of Software packages or standardized Cloud Service. We have given up the desire to control them: the consequences will not be too serious as these Solutions do not provide a competitive advantage.
On the other hand, the question arises for the Business Solutions.

One approach consists in defining the Business Model in-house, then outsourcing its translation into software externally. This translation is increasingly considered as a task that can be outsourced without any real value add: **IT Transformers** are often **not highly regarded** in the Enterprise.
There are 2 dangers with this approach:

- On one hand, if the Business Model is well built at the start, it is **rarely updated** later on, whereas the software undergoes successive modifications. Finally, only the IT Model represents the up-to-date reality of the Enterprise Model: when it outsources its IT Model, it becomes more difficult for the Enterprise to know precisely how it works.
- On the other hand, outsourcing will act as a break on the implementation of an Agile Approach in Enterprises: it is difficult to play the compromise game between functional Wishes and IT Possibilities if we outsource and fix the relations with IT by contract.

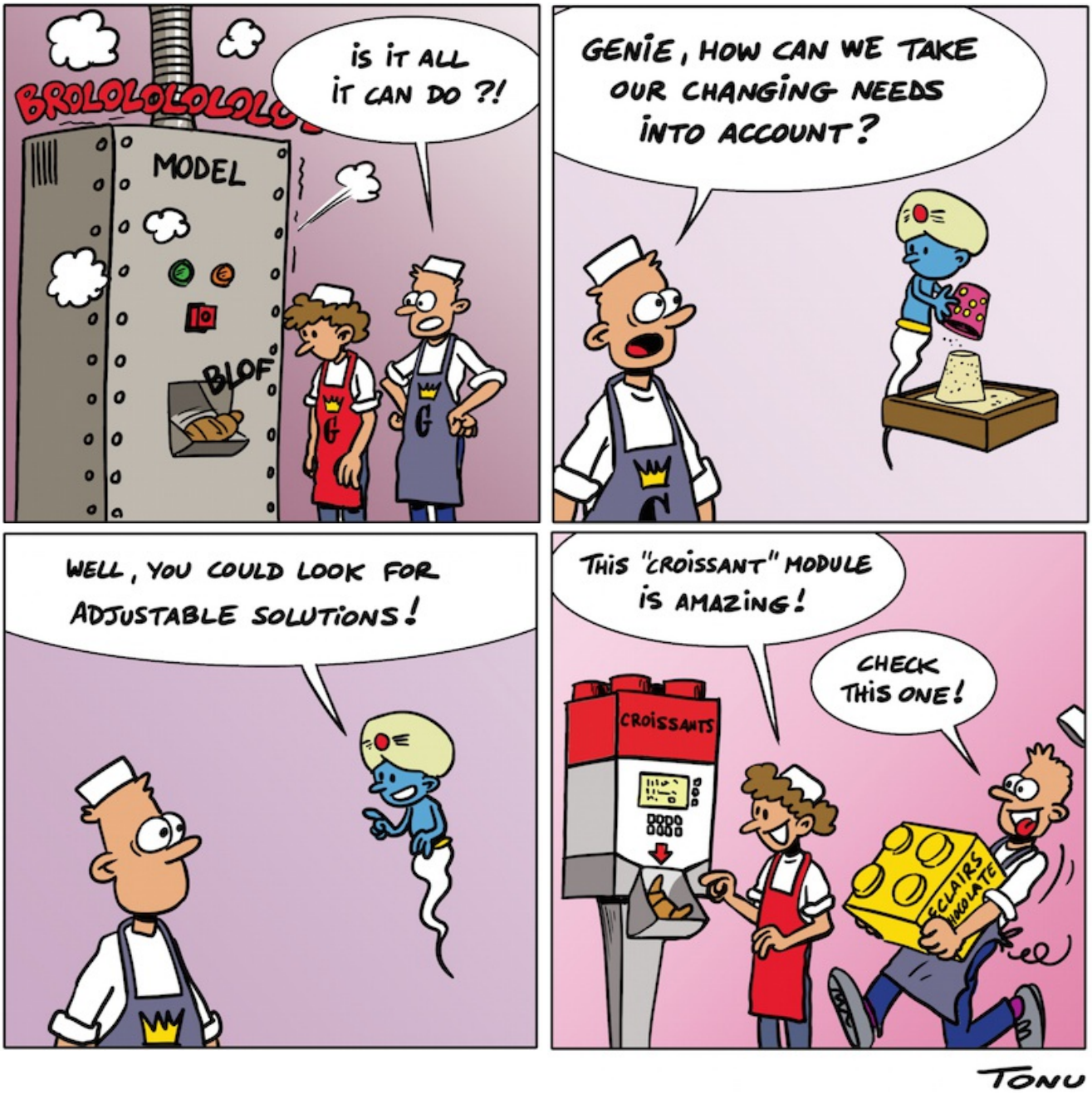
This is why a growing number of Enterprises today are looking to keep control of their Business Solutions. They feel that using a Software package for a Business Solution stops them from differentiating themselves from their competitors. They want to keep responsibility for their Business Model and, in particular, the associated software, to remain original.

To give an example, "Voyages SNCF" finally recently decided to do its developments in-house: impossible to coordinate the graphic artists, marketing, customer service and IT without having them work as a team inside the enterprise. Likewise, Axa-France decided to rebuild its teams of developers for the same reasons.

## 2. Can we subcontract IT developments?

Subcontracting IT developments is possible. But it is preferable that it proceeds by small, successive contracts, rather than a global contract that is supposed to contain everything: a subcontract Contract can be drawn up for each Version, after a sufficient number of iterations have been carried out, to select the requirements of the current Version.
It is desirable that each Enterprise keeps a core of recognized IT competences, controlling the overall design of the IT Model, otherwise, one day, it will find itself dependent on suppliers, who will have evolved the Solutions without the Enterprise keeping the knowledge of its Business Model.

## The Business must be able to modify the Model




# 1. Identify what often changes in the Model

A Model changes over time.
Certain elements of the Model are rarely modified. For example, the definition of the Business Objects and their relations: if we defined that a Contract is only linked to one Customer, it is unlikely that it will change.
But other elements change more frequently: pricing, conditions of eligibility, allocation of tasks between Actors, adding data to the Objects... that is to say the rules (or Functions), the sequencing of Processes and Information.
Identifying what is often modified helps to select a Solution that can easily support these modifications.

# 2. Acquire a configurable Solution

One of the main qualities of a Solution is to let us modify the elements that change often by "Configuration".
The idea is a simple one: we isolate the elements that change frequently and provide tools that can be used by non-IT workers to modify these elements. It is a modification of the Model, but it is perfectly limited and does not require complex Functions of programming, integration, non-regression testing...
Among the techniques used: parameterization, rules engine, workflow engine, dynamic data (see the white paper on Software packages).

Not only is it a technique that enables the Enterprise to evolve rapidly, but it is also a means of differentiating oneself from the competition.

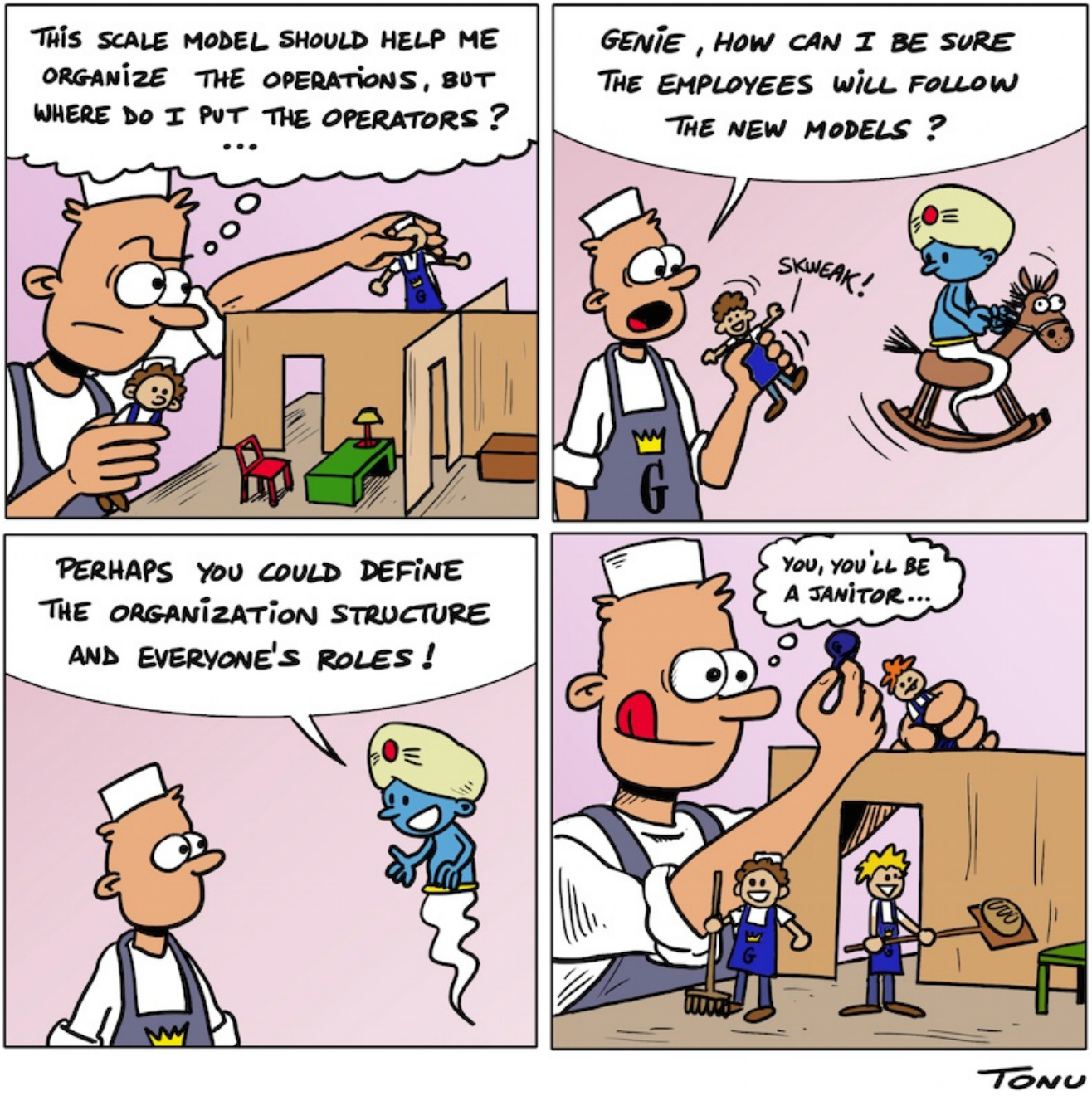
# 3. Personalize the single Model to adapt it to different Organizational units

As we have just seen, configuration techniques allow us to modify the elements that often change quickly.
There is also another use for configuration: **personalize** the single Model for **different Organizational units**.
When a Group is looking to rationalize its Solutions, it still has to respect the **specificities** of each of its Organizational units (or subsidiaries): language, currency, regulation, tax system, commissioning,...
Configuration techniques allow us to maintain different implementations of the same Model: the common Model can evolve through successive versions because the personalization of each Organizational unit has been isolated.

# 4. And tomorrow, Customers will configure their Products

Customers like to buy Products that are specific to them. Some enterprises have devised Offers that can be personalized.
As an example, the shoe "One Many" from Newfeel, distributed by Decathlon, gives you an infinite number of combinations of patterns, material and colors: a Customer creates a single Model him- or herself!
This is also a case of configuration: the Product architecture is the same, but what "often changes" in customer tastes has been isolated so that the Product Model can be configured.
The icing on the cake: the Model configured by a Customer can be proposed to other customers: one way of staying close to the market, as it is the Customers themselves who guide the Enterprise in Modeling the Products that are aimed at them.

# Define the organization of the Operations and the roles




## 1. The classic Organizations

Every enterprise is structured according to a certain number of dimensions: in most cases, we can find the following dimensions:

- Product line: for example, cars are separated from heavy goods vehicles
- Process domain: for example, Production and Distribution are separated
- Territory: for example, one Organizational Unit per region
- Individual Customers and Enterprise Customers
- Form of Distribution: online Banking is separated from retail Banking

The first 3 are the most used.

The most important dimension was the Territory; today, it is the Product line quite simply because transport speed, communication facilities and making Products commonplace have erased the differences between countries.

An Enterprise can of course combine these dimensions: for example, it creates a division per Product line and each of these divisions is decomposed into Production Department and Distribution Department.

# 2. The acceleration of Transformation has seen another dimension appear

If the Enterprise wants its Transformations to be quicker than those of its competitors, it asks the Transformers to focus 200% of their energy on making the Transformation succeed, which requires that their time is not taken up with operational tasks.
A new demarcation line has appeared **between Operations and Transformation**.
And within the Transformation, the separation between Solutions and Foundation (remember that the Foundation contains the Models common to the different Solutions) can be seen.

To take an example, a **Group** is decomposed into several **Companies**, each one responsible for a Product line.

## 2.1 Structure of a Company

Each Company is structured in 2 Organizational Units: Transformation and Operations.
The **Operations** manage the present and generate the Company revenues:

- Production
- Distribution
- Resources; human resources, financial, premises, information...
- Managing the Company

The **Transformation** prepares the future:

- Definition of the strategy
- Building new Models to Offer Products and Service, Operation Model, Image
- Deployment of the new Models: training Actors, installing hardware, information migration...

## 2.2 Structure of the Group

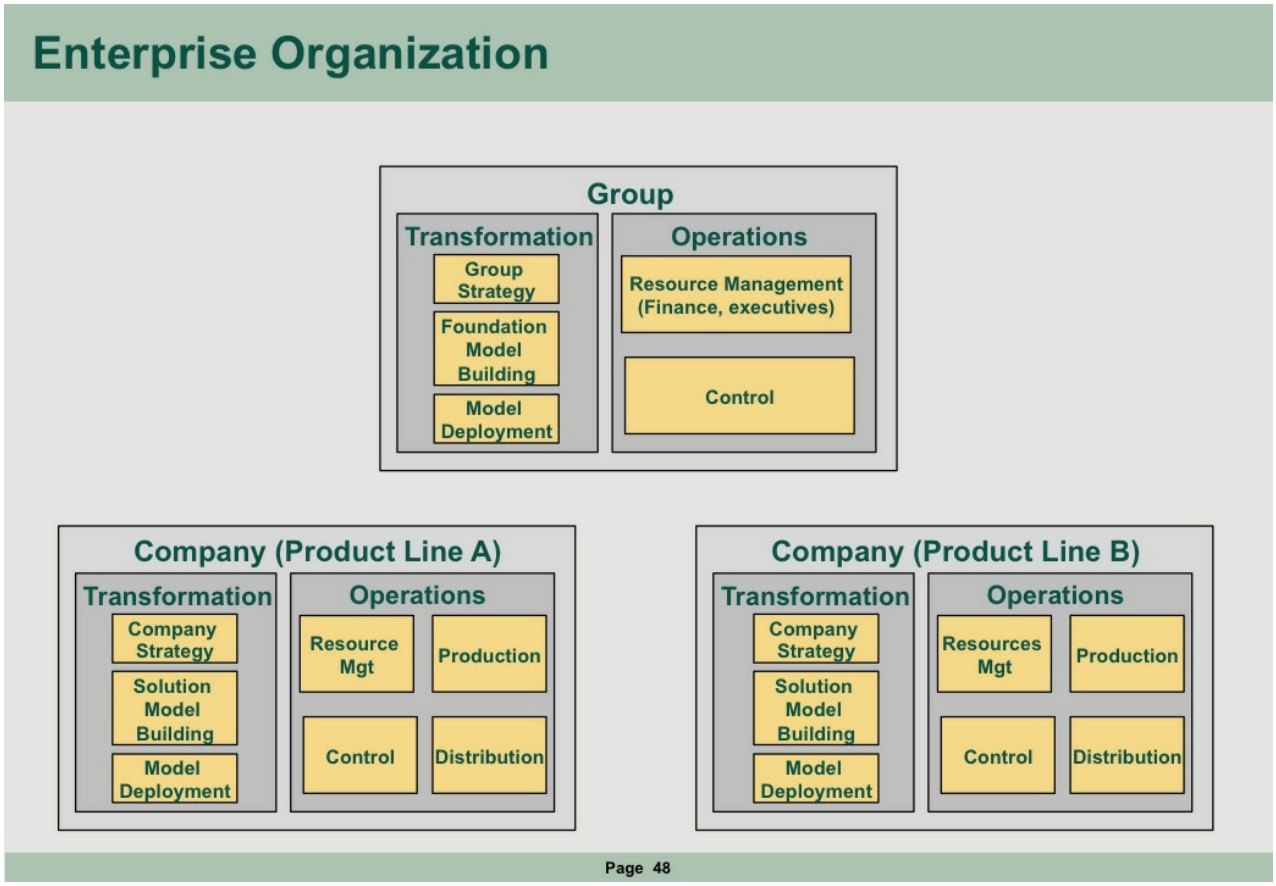
The Group is structured in the same way: Operations and Transformation.

The **Operations** do not concern the Distribution or the Production that are managed by the Companies, but

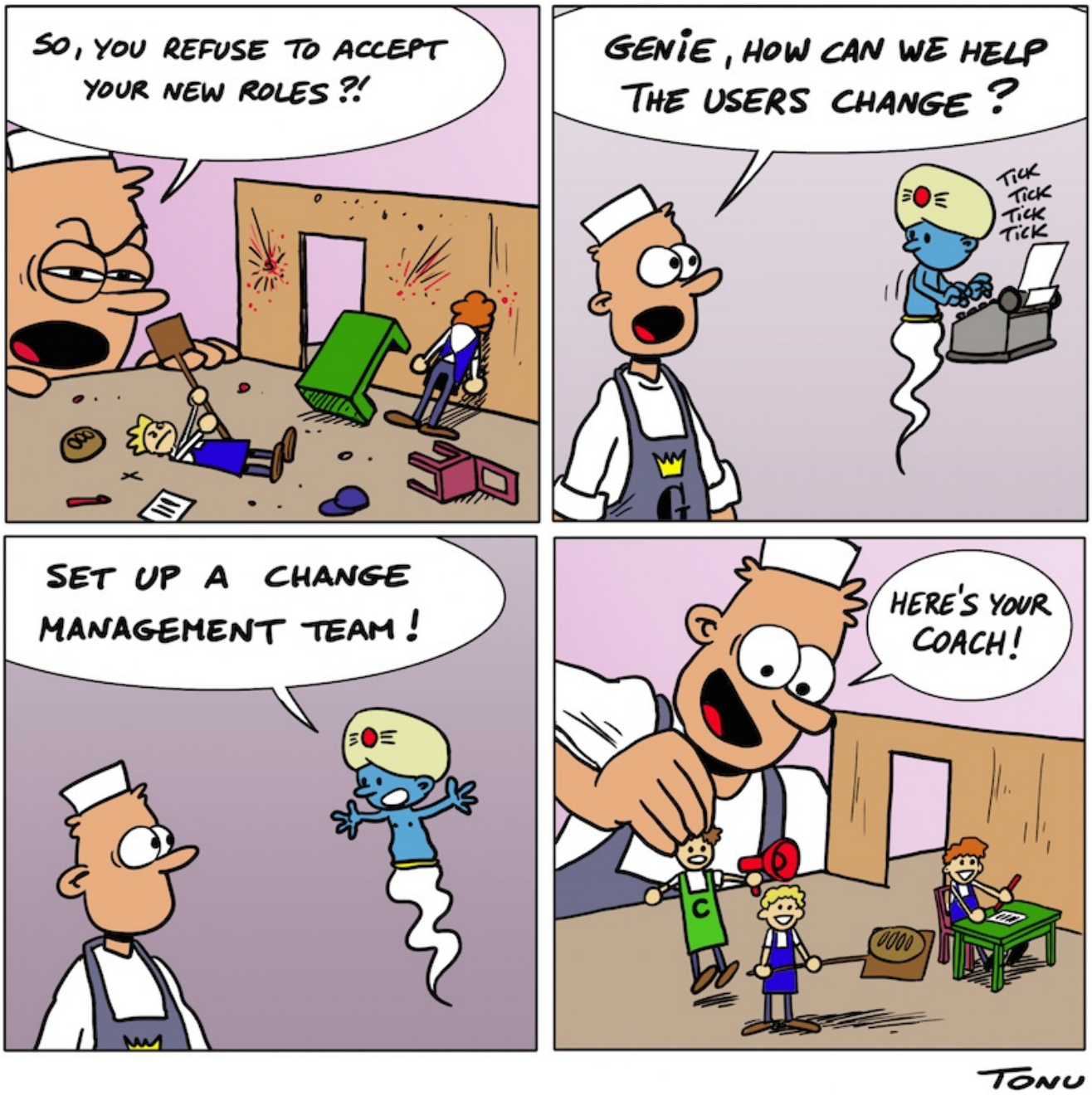
- the management of **Resources** specific to the Group, management of senior managers, financial management
- **managing** the company as a whole

The **Transformation** prepares the future:

- definition of the Group **strategy**
- building the Group **Foundation** which represents all the Models that can be reused by the Companies: human resources Model, financial management Model, shareable Solutions, common Components, common Transformation Approach...
- deployment of the common Models to the Company Transformation teams


Enterprise Organization
Group
Transformation
Group Strategy
Foundation Model Building
Model Deployment
Operations
Resource Management (Finance, executives)
Control
Company (Product Line A)
Transformation
Company Strategy
Solution Model Building
Model Deployment
Operations
Resource Mgt
Production
Control
Distribution
Company (Product Line B)
Transformation
Company Strategy
Solution Model Building
Model Deployment
Operations
Resources Mgt
Production
Control
Distribution
Page 48

**It isn't enough to acquire a good Model,
we also have to manage change for the users**




# 1. Deploying a new Model causes disruption

Actors like what they know, especially their own ways of working.
A new Operation Model represents an effort to adapt:

- Modification of the organizational structures and associated premises
- Change of everyone's role
- new IT application
- Information migration from one Solution to another
- Installation of new hardware and software

# 2. Looking for formulas to simplify Change

Aware that the speed of the Transformations may cause Actors, whether they are internal or external, to reject them, Enterprises lean on different principles:

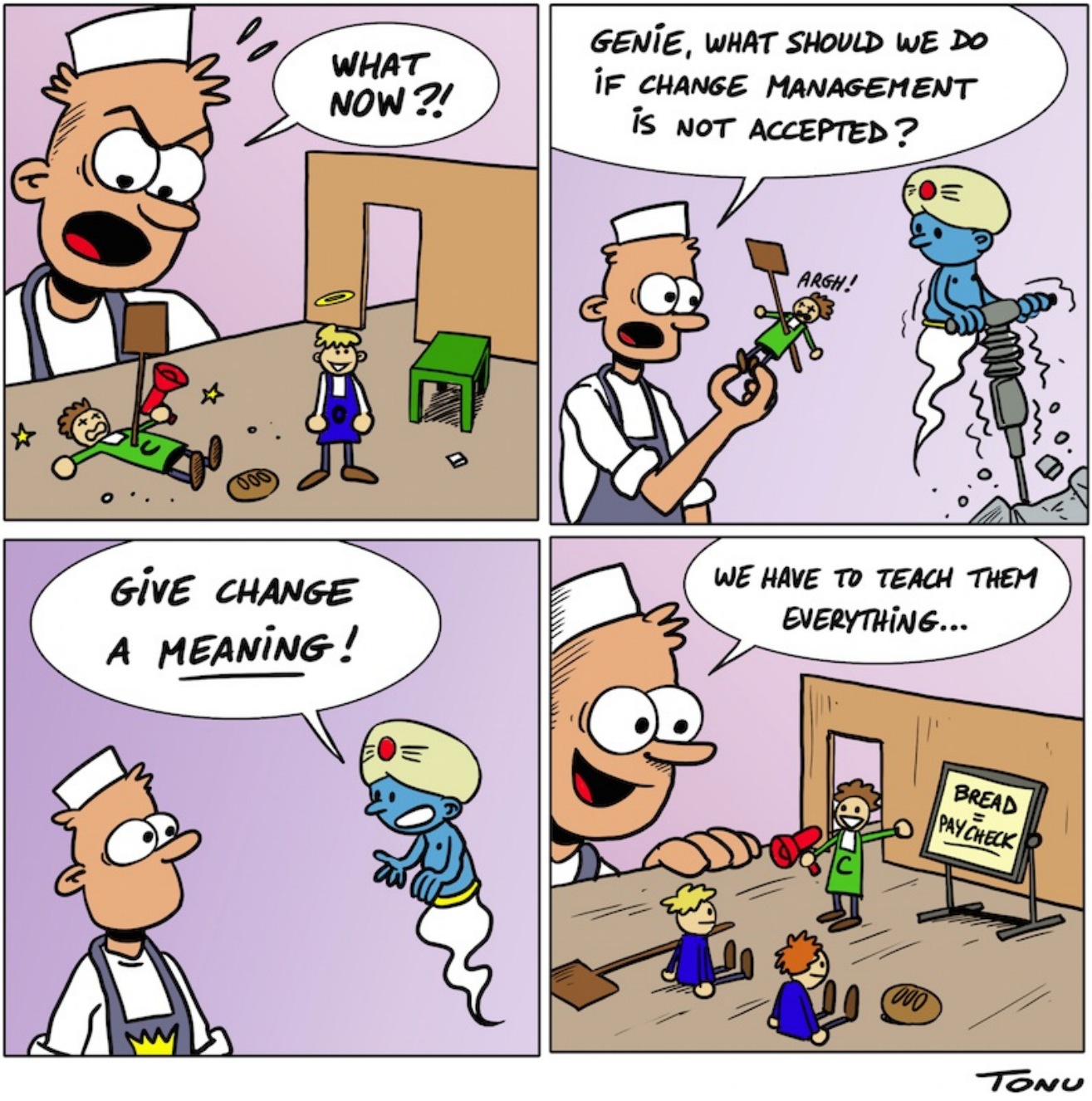
- Significant efforts are made in **training** to explain how the new Solutions work.
- The availability of an easily-accessible **hotline** can help them at any moment.
- The massive and standardized use of the **Internet** no longer requires us to change workstation and adapt the telecoms network to each new Solution: we simply need to have a workstation available with an Internet browser to access the new Solutions; this simplifies things greatly compared with the past. It means that managing premises becomes simpler: many Enterprises today have "open spaces" where employees can easily set themselves up, so long as they have an easy access to the Internet and therefore, their information.
- Using **Mobile Solutions**, accessible from a smartphone or a tablet, means that we can work remotely, which can be particularly comfortable for the Enterprise employee.
- Searching for **usage standards**, whether at the initiative of the Enterprise or its suppliers, simplifies the learning phase of each new Solution.
- Old data stored in old Solutions has to migrate to the new Solutions. The rigorous control of this Information is often greater in the new Solutions, which reject part of the old Information. Concepts evolve and it is difficult to have consistency in the information in both Solutions. It is often the most difficult thing with a Deployment: many deployments are slowed down by the difficulty of migrating information. **Automating** information **migration** is a success factor, but the tools are still inadequate today.

# 3. Adapting oneself to the new enterprise culture

This progress helps with the deployment of new Solutions.
But there is one domain where progress is not yet enough: the new Enterprise Culture.
The new forms of organization should take this cultural change into account: present generations do not behave like the previous ones. This generation is characterized by simple principles:

- **Information is free**: our power no longer comes from holding information, but from our ability to share it.
- Neither does our power come from our hierarchical level, but from our **Competence**.
- The requirement for **Actor confidence**; Actors need autonomy.
- Horizontal exchanges and **collective intelligence**.

## Give meaning




# 1. Can we refuse the Transformation?

We often hear: the Transformation should only take place if the Actors accept it.
In other words, if the Actors do not want it, we should not look to Transform.
It is clear that it is a lot sounder to carry out a Transformation in a favorable context than in a hostile context.
But the question is not "can we refuse the Transformation if the social consequences are too difficult" but rather "as we have to Transform the Enterprise to continue to exist, how can we accompany the change to make it easier?".

# 2. How can we make the Actors accept a Transformation?

As mentioned earlier, we can lower stress levels by standardizing the usage, providing more

mobility, facilitating Information migrations... In short, by contributing to provide **fluidity**.

But it is not enough: we also have to give **meaning** to the change so that everyone understands that it is a thought-out initiative which should provide a competitive advantage to the Enterprise and that they will contribute to it.
The **Vision** must be communicated to everyone, by clearly separating the "why" we are Transforming (what we call the Goal) from the description of the new Enterprise Model, which will enable us to satisfy this Goal.
When the Actors understand the Vision, they **accept more easily the consequences** of it, even if they seem negative to them.

According to John Kotter, a pertinent vision should be all these things at once:

- **Imaginable**, that is to say that it must convey a clear image of the future;
- **Desirable** in such a way that each stakeholder in the change gets something out of it;
- **Realistic** to be able to be broken down into reachable objectives;
- **Precise** to be able to truly guide the action and decision-making;
- **Flexible** in such a way as to allow involvement and appropriation through taking initiatives;
- **Communicable**, that is to say easily transmittable and explainable. Anyone should be able to understand the vision in under five minutes.

Once the Vision has been defined, the Transformation has to be described over time: at what rate will the change reach the organization and when will the Actors be impacted? It is advisable to avoid the tunnel effect and to move forwards by gradual increments (see the agile Approach) to give credibility to the approach.
If the different Actors share a common vision of the future, it will be far easier to have them **work together** to build it.

If their job is impacted, we have to offer them future prospects: here again, training plays an important role.

# 3. Complete consensus is illusory

The ideal situation would be that we manage in this way to convince everyone before starting the Transformation. But be careful of other-worldliness!
No matter what efforts are made, we will always find three types of population in the Enterprise:

1. The **positives** state that they have been waiting for this change forever, they do not understand why it was not done sooner.

2. The **inconstants** do not have a firm belief and waver from one side to another according to the successes or failures of the Transformation.

3. The **negatives** refuse the change and efforts made, declaring that it will never work, that things are complicated enough today as it is, that there is already enough to do without spending energy changing the Model...

The proportions of the 3 groups depend

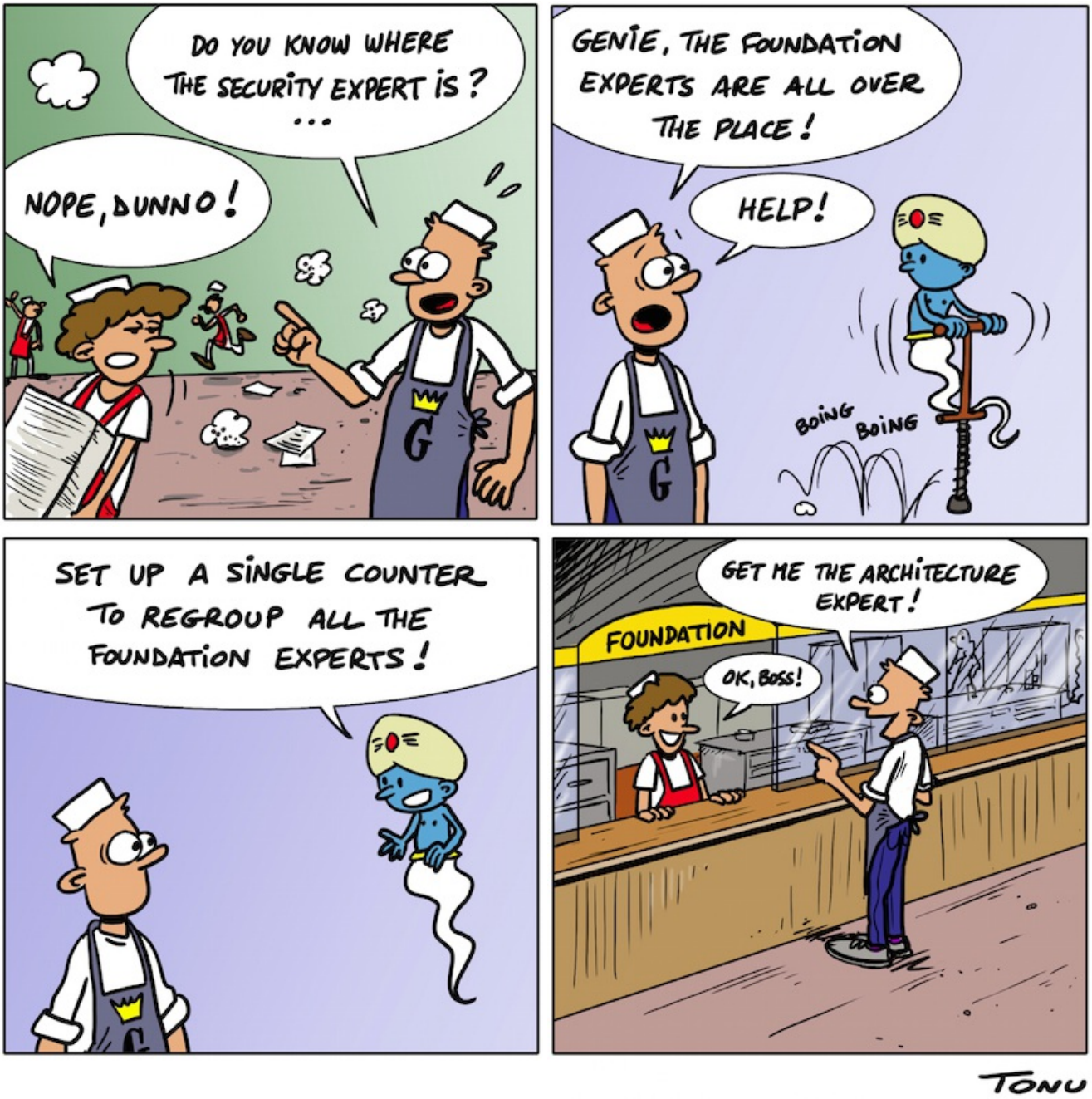
- on the Enterprise **Culture**: digital Enterprises that continuously evolve their Offer accept change more than traditional industries
- on the **status** of the employees: if job security is guaranteed, the 3rd group is powerful; if, conversely, everyone's job depends on the success of the Enterprise, the first group is important
- on its **history**: in particular, previous failures can increase the 3rd population.

Experience has shown that we cannot convince the 3rd population in the preparatory phases of a Transformation, which must not stop us from informing them like everyone else.
If we look for a complete consensus before beginning a Transformation, we risk never starting it.

It is recommended to rely on the first group, for the initial phases which are generally the most difficult: we accumulate Building the Enterprise Architecture, the Components, we teach the new Transformation team to work together and we try out the first Models.
The "positives" will have to put up with the initial problems.
If successful, facts are stubborn and the inconstants will rejoin the first group.
If success is confirmed, we will no longer hear from the diehards. Some of them will even recognize that the Transformation has been a success.

## Act 10: Organize the Transformation

# Define the organization and roles of the Transformation




## 1. Ask for a Transformation assessment

We have to carry out an assessment of what the Transformation **costs** the Enterprise: not only the IT Development costs, but also the cost for the Business Actors, Transformation Tools, related management and governance costs, training for the operational Actors, dual processing, information migration,... An assessment of the level of satisfaction regarding the Solution design or modification **timescales**, and on the **quality** of the Solutions.

This assessment will inevitably show that the overall Transformation cost and the Business frustrations regarding the Agility of the Solutions deserve that we tackle this problem head-on.

# 2. How do we sort the Transformation roles?

There are many Roles in the Transformation and each organization has defined its own roles. We advise sorting the roles (in red below) by main function. As an example:

- Strategy: Strategist, Sponsor
- Product and Service Offer
-
    - Marketing: Marketer
    - Component Foundation for building the Product and Service Models: Product/Service Architect (e.g., component manager in the automobile industry)
    - Product/Service Models: product manager/Service, Product/service designer
- Operation Model
    - Enterprise Architecture: Business Architect, urbanist, data administrator
    - Solution Models: project manager, analyst, IT Solution designer, Solution configurator
    - Component Foundation for building Solution Models: Business Architect and Technical Architect, Security Architect, ergonomist, foundation support for Solution builders
    - Solution Deployment: Trainer, change manager, organizer, Solution evaluator
- Transformation Models: methodologist, quality manager
- Solution Support for users: hotline

There can be different **hierarchical** levels in each of the categories, according to the size of the team.

Furthermore, new roles appear. For example, "Big Data" requires roles such as "Chief data officer", "Director of the customer experience", "Chief Customer Officer", "Data scientist" or "Data Officer".

The only recommendation that we can make is not to multiply the Actor Roles: if we have a choice, it is better to have **few Actors of a high level** and thus versatile, rather than **many specialist Actors**. It is not because "quality", "security", "ergonomics", "urbanism", "organization", "method", "integration", "performance", "tests"... are important that we have to have as many different Roles. The **multiplication of Roles** heightens problems of monitoring, coordination and integration and makes the teams **less responsible**. Once again, look rather for quality contributors **able to assume several roles**.

For each role, we then have to detail the mission. For example, the "sponsor" defines the Goal, approves the new Enterprise Model that respects this Goal, agrees to the budget, follows the progress of the Transformation and supports the Transformation team in the challenges it faces.

# 3. Focus on the Roles concerning the "Common Good"

The roles of Solution or Product/Service Builders exist in Enterprises today, under different names. On the other hand, transversal Roles which create synergy, coherence and economies of scale within the Enterprise are not always present. These are roles which are in charge of the **common Good** and come under the "Foundation" team:

- **Component** Supplier for building **Product Models**
- **Component** Supplier for building **Solution Models**
- Enterprise Architects who define the overall Solutions plan
- Methodologists who propose the Transformation Model reused by different teams
- Builders of Solutions that can be reused by a Group's different Organizational units
- Builders of Product Models that can be reused by a Group's different Organizational units

They represent everything that contributes to organizing, simplifying, bringing coherence and facilitating synergy: the **Common Good** is managed by the **Foundation** team.
The total or partial absence of these Roles is due to the fact that there is no transversal Business

Organizational unit. It is of paramount importance to explain to Executive Management why these concerns are important, so that quality-level Business resources can take charge of them within the IT Department or elsewhere.

# 4. Organizational Principles of the Transformation

Once the Roles have been defined, we can choose an Organization that is adapted to Agility and coherence. The organizational principles are simple:

- **Separate Operations and Transformation** because present concerns always take priority over future concerns.
- **Isolate the Common Good**: bring together multidisciplinary competences within the same team
- Organize by deliverable (multidisciplinary) and not by competence
- The Enterprise must **control its Global Model** and only subcontract out to partners the detail Models: it is the key to its ability to evolve.

# 5. Divide into teams of 7 people

The consensus of opinion is to have a **number of 5 to 12** Transformers by team, **ideally 7**.

- Below 5 and the team becomes vulnerable to external events and a lack of creativity.
- Above 12 and productivity and cohesion decrease, power struggles develop.

# 6. The qualities of a Transformation Project Manger: Modeler and Manager

- Project managers who are only **Managers** without have **Modeling** skills will not have the commonsense reflexes of someone who knows what is realistic, what can be implemented easily and what will profoundly question the Model. To compensate for this lack of judgment or experience, they are likely to protect themselves by weighing down the control and reporting procedures, or holding meetings to obtain a consensus that protects them.
- Project managers who are only **Modelers** without having **Management** skills will be faced with other, well-identified problems: among which, they will not know how to stop the influx of requirements and will neglect reporting or documentation aspects.
- If they have Business and IT competences, then they are the perfect profile, but it is rare.
- It they do not have both, then they must surround themselves with experts who can bring this competence.

# 7. Changes in the role of the IT Department

This movement challenges the current role of the IT Department:

- On the one hand, separating Operations and Transformation results in the separation of IT production from development
- On the other hand, the multidisciplinary nature of the projects which leads to profoundly transforming the business model and not just computerizing existing processes: the IT Department is no longer the main contributor to the projects.
- "BYOD" ("Bring Your Own Device") Actors want to use their own Mobiles at work
- 30% to 40% of IT spending takes place in the Business.
- Digital leads to profoundly transforming the business model and not just computerizing existing processes. Enterprises today are ready to rethink their Organization:
    - Some enterprises, like Voyage SNCF, have removed altogether the IT Department. They prefer to form work groups with a mix of IT workers, marketing and customer management.
    - Others, like Air France, have set up digital committees where the IT management plays an important role.
    - Others, like Pernod-Ricard, have set up a "Digital Accelerator Team" which groups together 80 experts worldwide to list the initiatives in progress and initiate Digital Transformation Projects.
    - BNPParibas has isolated its online banking activity which acts as a prototype for

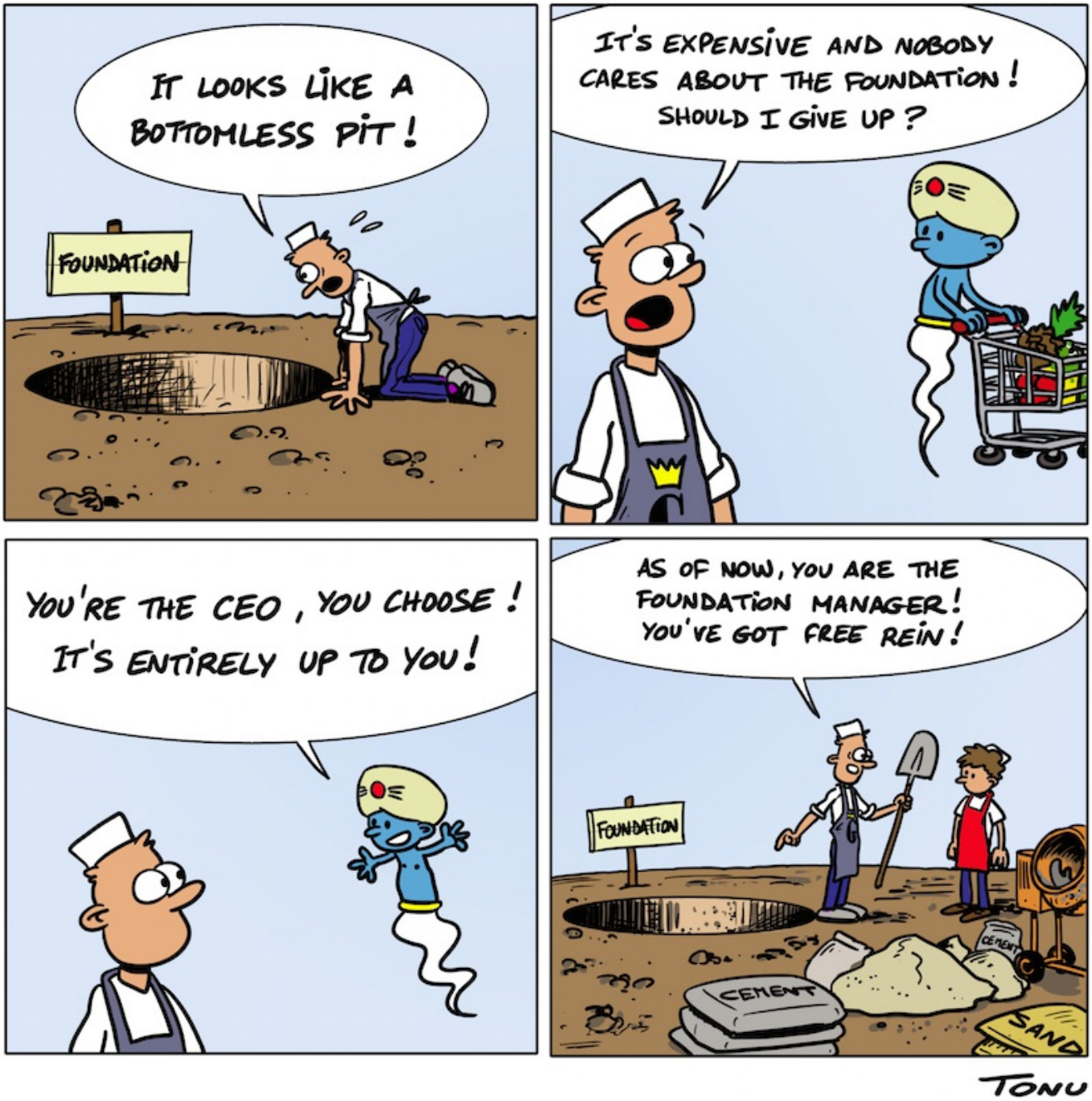
> the rest of the bank. It enables it to throw off the unwieldiness of the established organization. It only represents 4% of the activity today, but is growing by 15% to 20% per year.

The IT Department retains two assets for playing a fundamental role in these new Transformations:

- it is the only department of the Enterprise with experience in promoting the common good, it understands best what a Foundation is, which brings coherence to the Enterprise
- it knows about software complexity and can bring a dose of realism in the decision phases

Its current representatives should play a key role in the new Transformation teams.

**Isolate the Foundation team and allocate a budget to it**




## 1. The common Good is taken into account by a Foundation team

The Transformation aims to build Product Offers and Solution Models. So that these Offers and Solutions are built in a coherent framework, we also have to take the "**Common Good**" into account.

This is why, alongside the teams in charge of these projects (new Products and new Solutions), there are other small and often distributed teams, such as "Methodology", "Urbanism or Enterprise Architecture", "mapping", "Enterprise Glossary", "Security", "usage norms", "Architecture", "Development Tools", "technical Infrastructure", "Interfaces", "Components",...

Our suggestion is to create **one** Organizational unit, that we can call "Foundation" (or "Common Good"), responsible for everything that is shareable between the Business Lines, so that the Solution Project Manager only has to deal with **one** internal supplier. In what can be reused, we find:

- The **Transformation Model**: how we Transform (Approach and Tools)
- The **Components** to build **Products and Services**
- The **components** to build **Solutions**
- The **shareable Solutions** which can be common to several Organizational Units
- The **Enterprise Architecture** which describes the global structure of the Solutions and their exchanges.

## 2. The "Foundation" team only exists through the determination of Executive Management

Each Business Unit is judged on its result. It is therefore illusory to ask it to work for the "common good" of the Enterprise: it naturally has a selfish behavior.

It is up to Executive Management to take responsibility for this common good. To do so, it needs to set up a **Structure** which works for the good of the Business Units and brings together excellent competences in both Business and IT.

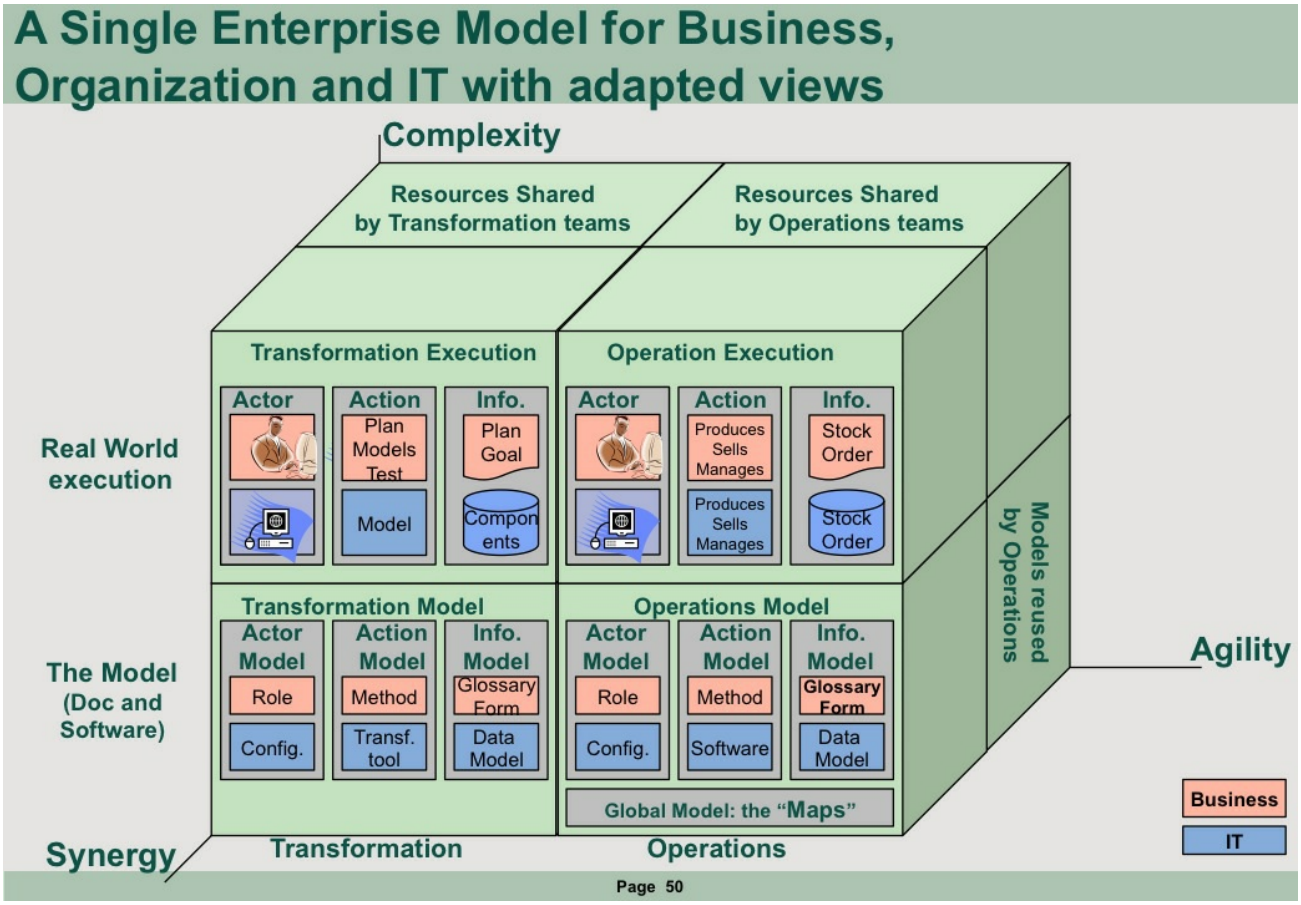
One of the major difficulties is that we struggle to justify the economic profitability of the Foundation. We know that we need coherence to simplify the overall system but it is difficult to monetize what is structural. We do not justify in economic terms the need for electricity or communication means in the Enterprise, we know we do; it is the same for the Foundation.

## 3. Objectives given to the Foundation: Agility, Synergy, Simplification

The Role of the "Foundations" Organizational unit is to increase Agility in both its characteristics "fast" and "well"; to identify the right level of **Synergy** between Business Lines or Subsidiaries and to provide ideas to **simplify** the Operation Model.

Define the **road map** to **simplify the Enterprise Architecture**. A project will be easier if it fits into a well-structured Enterprise Architecture. The clarity of scope, the precision of the interfaces with other Solutions, the reuse of Information access Functions are some of the assets that help focus the Project Manager's energy on the Solution Model and not on his/her environment. The real difficulty is in establishing a strategy of progressive simplification so that each Project contributes to this overall simplification. This first track has been defined in the white paper from CEISAR "**Simplify Legacy Systems**": we recommend that the interested reader downloads this white paper from www.ceisar.org

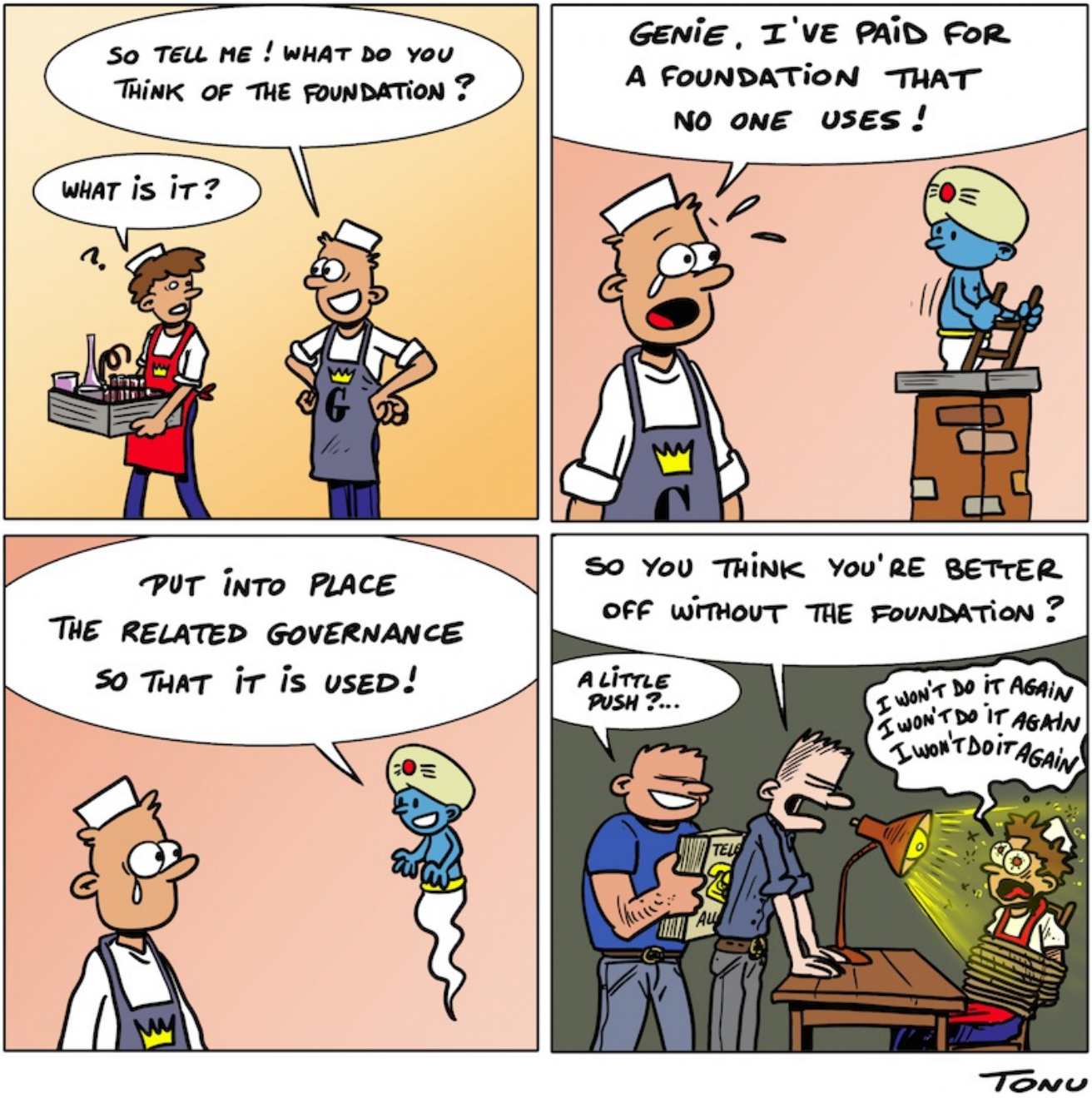
To check that its Goal has been properly reached, the Foundation team has to acquire measuring tools to follow the changes in the **Agility** (speed of change and quality of Solutions) and **Synergy indicators** (the sharing of repositories, the reuse of Solutions or Components).




The CEISAR Cube is a classification framework whose 3 dimensions correspond to the 3 biggest challenges facing Enterprises today: **Complexity, Agility** and **Synergy**.

- managing the growing Complexity requires us to separate the real world from its **Modeling**
- managing Agility requires us to separate Operations and **Transformation**
- managing Synergy requires us to **Share** Resources (human resources, IT or information repository resources) and to **Reuse** Models (Solution or Component or Approach Models)

# Define the right Transformation governance




## 1. What is governance?

Governance is the art of making important decisions and following their application.
As the most important decisions are the changes to the Enterprise Models, which have to do with Transformation, Governance therefore impacts far more on Transformation than the Operations.

## 2. Everyone is president

The characteristic of the digital culture is that information now circulates freely.
Everyone in the enterprise now has the means to formulate a judgment on the diagnosis and what needs to be done to move forward: everyone becomes "President".
The time spent convincing all these "presidents" can delay the decisions.
Moreover, the possibility of direct exchanges favors the setting up of pressure groups outside of

existing bodies: see the "pigeon" movement in France that squeezed trade unions.
Enterprise governance is becoming more delicate. And yet, we have to decide and choose a way.
Dividing governance up into 5 steps is the same, but the content of these steps has evolved:

1. **listen**: **social networks**, whether they are internal or external, are valuable for understanding the reactions and suggestions of customers, partners and employees. One group, Pernod Ricard, implemented an internal social network that is already used by half of its staff one year down the line, which breaks down the barriers between management and its base and has enabled them to collect some remarkable suggestions on how to improve the group.

2. **build scenarios**: we have to go faster than in the past and ensure that the **Foundation** is respected if we want to avoid a pile-up of disparate projects

3. **decide**: the decision must be accompanied by success **indicators**; we have to be able to objectively justify the degree of success of the initiative at its end to better communicate it

4. **communicate** the decision: everyone must understand the **meaning** of the decision; do not hesitate to use the internal social networks

5. **support** its decisions: transformation projects are difficult, especially in their initial stages; decision-makers must not only decide and provide the budget, they have to ensure that the project is successful and **support** the Transformation teams.

# 3. Foundation Governance

The biggest difficulty is being able to launch multiple Projects while respecting a certain coherence: the Foundation team has to manage the Common Good. But the Solution Project managers can view the Foundation team as a constraint that does not leave them the independence they need to successfully see their Project through. What should we do?

## 3.1 A competent and recognized Foundation director

The director of the Foundation Organizational unit must be credible and respected by the Project managers if we want his/her recommendations to be heeded.

## 3.2 The scope of the Foundation is in line with the strategy

We do not have the same needs for Synergy in Industry or in Services. The need for synergy is less strong at Total than at BNPParibas.
In the oil industry, jobs like exploration, refining, distribution and chemistry are extremely different and hardly justify having a common bank of Components. We can, on the other hand, pool commodity Solutions to manage human resources, finance or team collaboration.
In the bank, we only deal with Services and Information: the Product/Service Models are based on IT and Process Modeling. Expenditure on IT easily represents 10% of the turnover. The potential synergy is considerable between the Product lines, the set-ups in the different countries, and the front- and back-offices. The Foundation will play a crucial role here.
The scope and the means of the Foundation Organizational unit must be adapted to the ambition of the Group.
We need an important Foundation Organizational unit if the Group applies the "Centralize the Models and decentralize the Resources" principle.

## 3.3 Communicate about the Goal of the Foundation

We have to remind people about the Goal of the Foundation:

- Agility: because we reuse common Solutions or components

- Quality of the Solutions: because everything that is reused has already been tested
- Ability to exchange not only good ideas, but also good Solutions
- Coherence of management tools
- Uniqueness of information: we only enter it once, we are able to present all the information about the Customer
- A real ease to transfer employees as they find a uniform usage.

### 3.4 Project managers must respect the Foundation or prove that it is inadequate

We have to find **Governance** rules that lead the Business Lines to reuse what the Foundation Team can offer.
Every Project must respect the Foundation or demonstrate that it is not possible, but the project team **cannot ignore** the investment made by the group.
The control of the Project's conformity with the Foundation must happen **before** the decision-making body meets to validate the project and budget; it is the only way of avoiding comments from the Project manager like "to stay within the budget and timescale that you have given me, I cannot use the Foundation this time".

### 3.5 Do not penalize those who play the Foundation game by internal cross-charges

There is a big temptation to bill the Foundation to its Customers (the Business Solution builders) to turn it into a profit-making center and not a difficult-to-justify investment.
But, if we want to incite people to use it, we should do the opposite: the fist Customers of the Foundation, who will come up against the initial problems, should be paid to thank them for their solidarity.
It is only when the Foundation use has become generalized that we can consider billing for it.

### 3.6 The Foundation Organizational unit should behave like a supplier

The Foundation Organizational unit behaves like a Supplier to the Business Lines and not as a hierarchically superior structure. The Foundation Organizational unit is generally split in 2: those that build the Foundation and those that support it.

### 3.7 The Foundation Organizational unit improves its offer thanks to Business initiatives

Incite the Foundation Organizational unit to **recover** and package Components coming from the Business Lines.
The Business Units must behave as a source of **proposals** vis-a-vis the Foundation Team.

### 3.8 Follow indicators to gage the effectiveness of the Foundation

Follow the **progression** of indicators like agility, quality... anything that may reassure people about the concrete use of the Foundation.

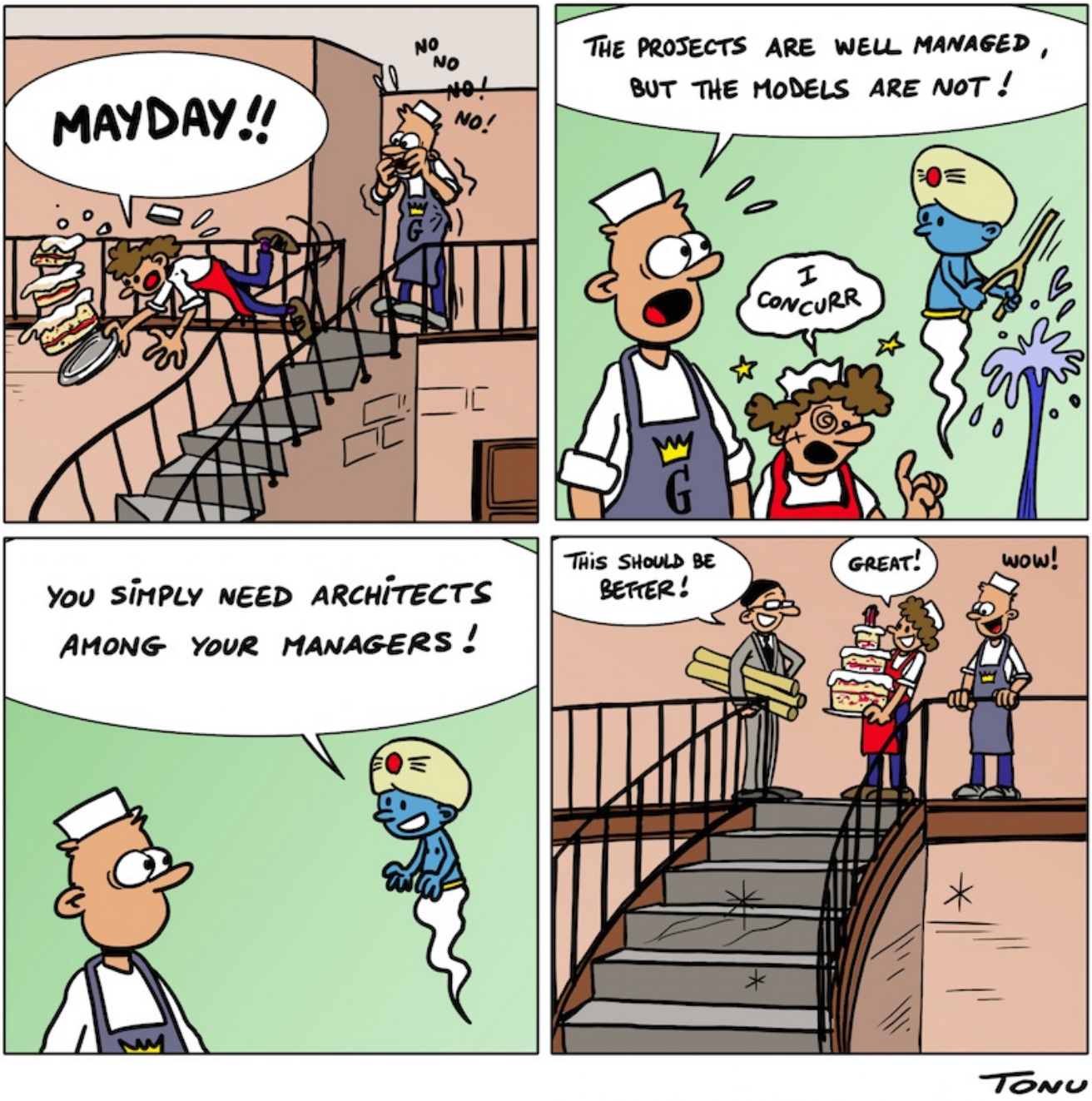
## 4. Governance of purchasing

The Purchasing function is meaningful when the Product to buy has been clearly identified.
But, when it comes to judging the quality of an expert of a Model, merely discussing the price or purchase conditions is not enough. The Purchasing function consequently has perverse effects: to economize, we may go for a mediocre quality.

To avoid this trap, here are some suggestions:

- Emphasize the **Quality**, and not just the price
- Give preference to partnerships with small **innovative** structures
- Find new forms of partnership between hourly wage and fixed bid contracts
- Favor Cloud solutions (see the related theme)

**Good project management is not enough to produce a good Model**




# 1. We have made good progress in Project Management

Changing Operational Processes via a Transformation Process is a difficult task.

Changing the Transformation Processes themselves, with the aim of making them more efficient is even more difficult because they are **more complex**.
A lot of projects suffer from bad project management: Phases are not formalized, deliverables are not provided, the schedule is incomplete, the assignment of actors is not anticipated, reporting is forgotten about, decision are not formalized, incident monitoring is random...
To manage this complexity, we have formalized project management: governance, schedule, budget, resources, communication...
Organizations which define methodologies, like CMMI or Open Group (TOGAF®), have gone very

thoroughly into the Transformation Management Functions. Enterprises have noticeably improved by leveraging this trend and by controlling Project Management ever better. We have gone from a time of improvisation to a time of continuous improvement, which is clearly demonstrated by the levels of maturity of the CMMI, which defines how an Enterprise progresses in its project management.

## 2. But we still do not control Transformations well: why?

Despite all these efforts, the success rate of Transformation Projects is still poor (see the results published by the Standish Group).

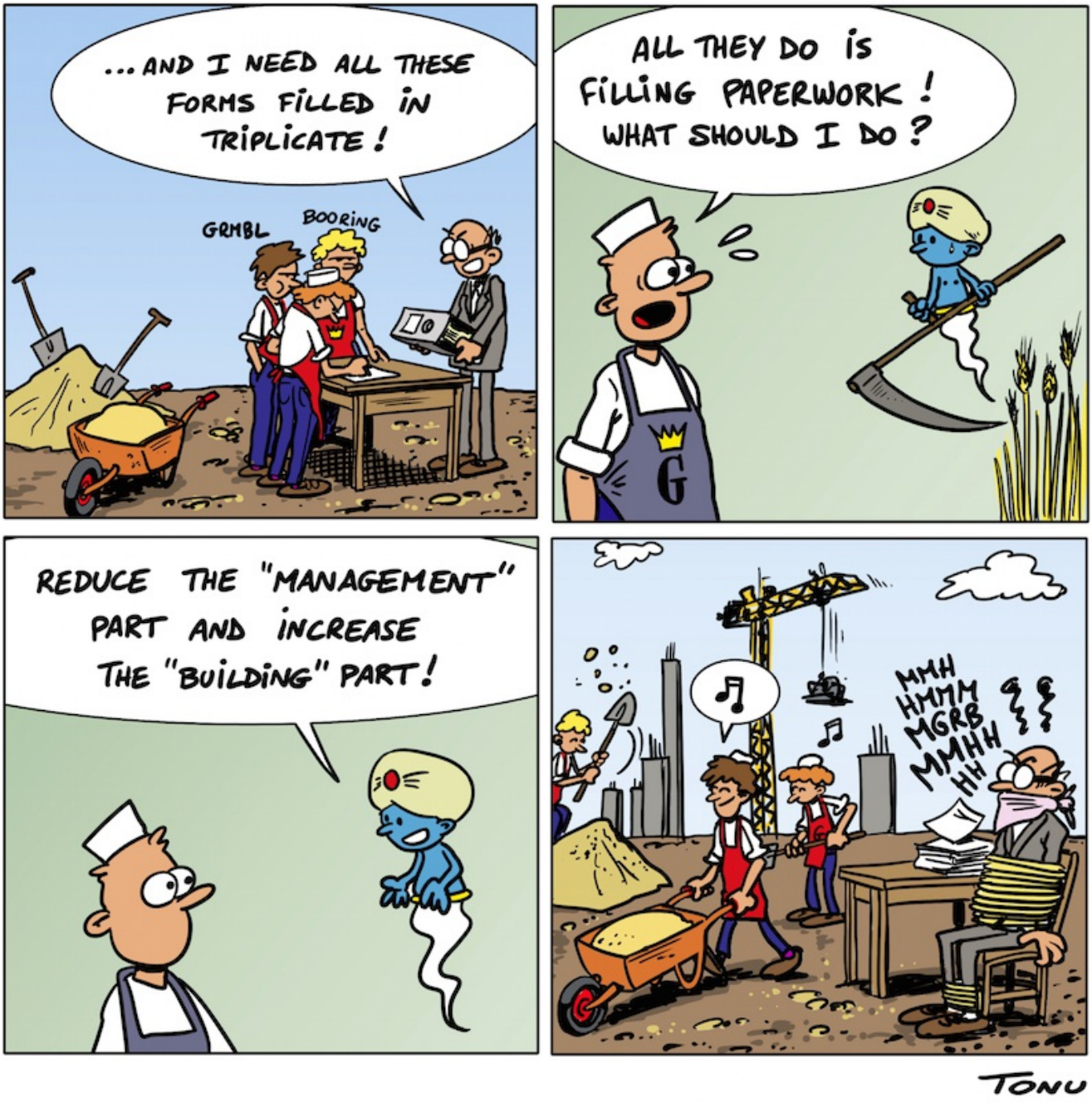
And the recommendations from the Standish Group are mainly recommendations of good Management:

- Commitment from management
- Involvement of users
- Experience of project managers
- Formulation of business objectives
- Scope restricted to the main requirements
- Normalized technological infrastructure
- Precise and stable specifications
- Formal methodologies that are used
- Reliable and rigorous estimations
- Others: dividing up deliverables, Competence of staff, etc.

We have to continue down this path of improving Management, but not to excess: too many Management tasks prevent the project manager from spending his/her time on Engineering Functions.
Our analysis is that we can **Manage well** the project of a **badly Built** Solution.
In other words, it is not enough to manage the project and its resources well, we also have to ensure that the new Model is a success. **Engineering** is as important as **management**.

But we have not managed, as yet, to successfully Model Engineering Functions: how to define the Goal, how to design a new Offer, how to Architect a Solution, how to reuse Components, how to Build a Solution that supports different Organizations. There is still a lot of progress to make in this domain.

## Bureaucracy can make a Transformation fail




# 1. How do Transformers spend their time?

Excessive meetings and reporting kill the Transformation.
Some project managers start the week with a full diary of meetings:

- Meeting with his/her team to provide a status on the progress of the project
- Meeting with his/her boss
- Meetings with his/her internal customers
- Meetings with his/her partners
- Meetings with future users of the Model
- Meetings with suppliers

They no longer have any time left to build the Model or to check that it is well built.

Some spend a lot of time reporting on what they have done or will do rather than actually doing.

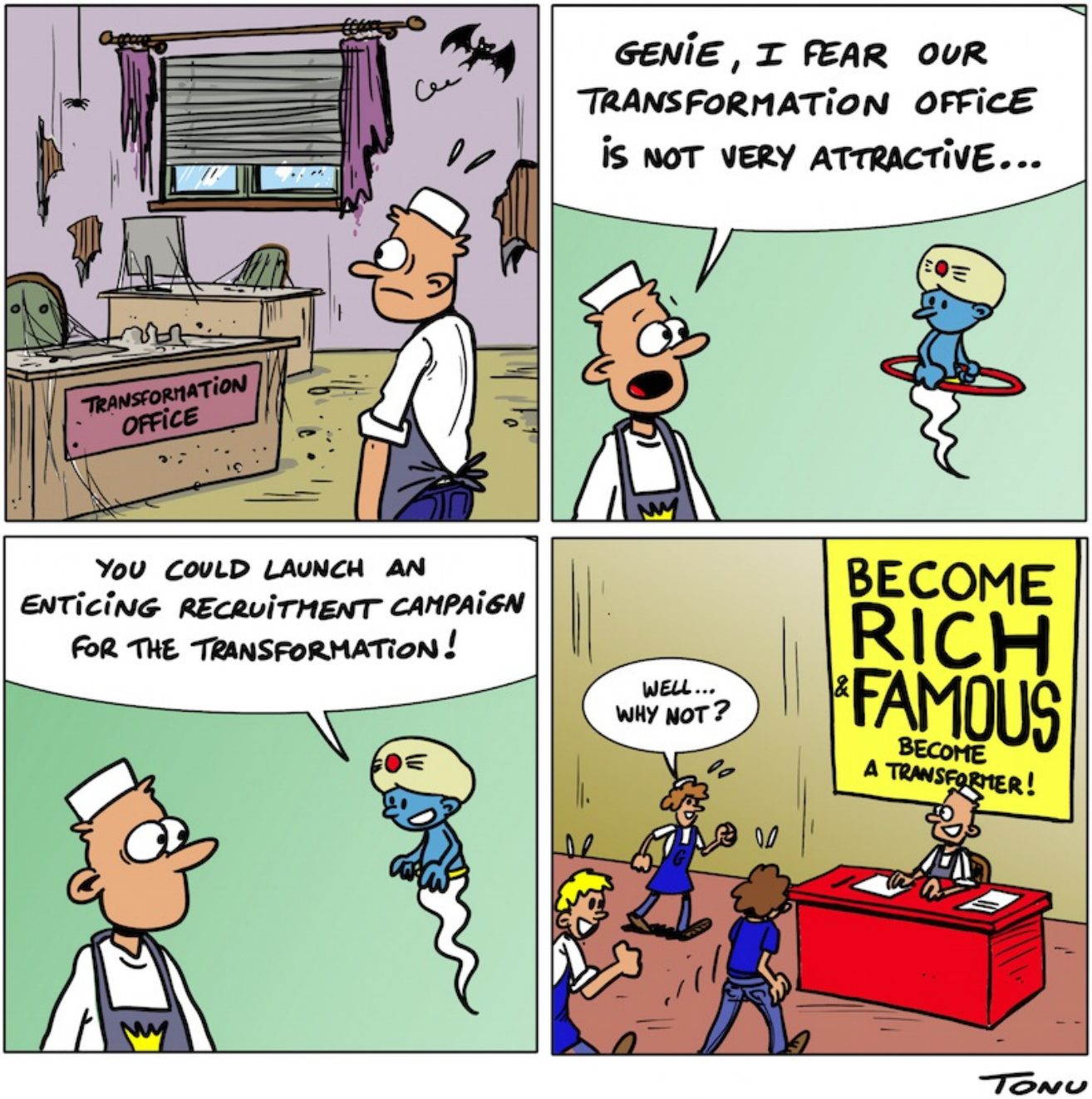
These constraints start out from a good intention: if we are properly informed, we can react in time. But this lack of confidence is often linked to past failures: we wanted to learn a lesson from an unsuccessful project, by adding layers of control.

In reality, if the Model is well built, the project is generally a success: managers should spend more time on their new Model than on managing resources and the schedule, which presupposes that we show good pedagogic judgment to make the Model accessible, through prototypes, clear explanations, examples...

# 2. Recommendations to limit bureaucracy

- Meetings
  - There are information meetings and work meetings where we deal with problems
    - Limit the number of information meetings: use communication channels (Intranet, for example)
    - Ban work meetings of more than 5 people (a real-life example, extremely effective)
  - A meeting should not be used to describe what is working normally (reporting is available for those that are interested) but to manage the exceptions.
  - Imperative: set a fixed time to end the meeting.
  - All topics are prepared.
  - The minutes are concise: not the reporting of exchanges, but actions: **who** is doing **what** for **when**.
- Reporting
  - Automate the production of a simple reporting: think carefully about a small number of key indicators
  - Automate the consulting of the reporting
- Quality of the transformers: if the Transformers are of a high quality, there are not so many of them and they have less need to communicate. Bureaucracy particularly concerns organizations where the transformers are of an average level.

# Managing Transformation human resources and the Transformation culture




## 1. The Transformation is more difficult than the Operations

We know how to Model the "Order" Operational Process, but we do not really know how to properly Model the "Build a Solution" Transformation Process.
There are many uncertainties:

- Uncertainties about the deliverable
- Uncertainties about the Solution Architecture
- Uncertainties about how the Project will go
- Uncertainties about the level of acceptance from the Operational Actors.

They require the Project manager to have the necessary talents to make decisions in an uncertain environment.

## 2. How do we choose Transformers?

- **Select** Actors with the function "Business Solution Builder", of **high quality**, able to carry out as many Transformation tasks as possible, to avoid the multiplication of Roles (see above).
- Choose and Train Project managers that are not only **Managers**, but also **Builders**. We have often promoted the former because they were enough in a linear Approach adapted to **Commodity Solutions**. They are not enough today to apply an agile Approach adapted to **Competitive Solutions**: the quality of the Solution Architecture is key. As good Builders are rarer than good Managers, we recommend choosing not the best managers, but the **best Builders**, for the Competitive Solutions "Project Manager", even if we have to support these Builders with people in charge of the Project administration. This is an important cultural change in the Organizations, which have rather pushed the Roles of Managers to the front.
- It seems that the main characteristics of those who succeed are
  - **Empathy**: we have to know how to make ourselves liked by the people we work with; the authoritarian manager is no longer acceptable.
  - Accepting **permanent change**.

## 3. How do we increase the standing of Transformers?

How do we keep the entrepreneurial spirit going in our teams?
If we want to appoint talented people to the **Business** Transformation, we have to **add prestige** (salary, training, recognition) **to the Transformation function** and not define the essential criterion for recognition as the size of the teams (Operational) that we manage.

## 4. How do we manage the Transformers?

Managing HR in the Transformation needs a pertinent Model: selection of the best, career opportunities, full-time job, right to fail...
High salaries or outward signs of recognition, like the size of the office or parking space, must not only be reserved for those who manage a lot of people: we must also include the good Transformers.
Protect Project managers from an excess of Management tasks (see "Bureaucracy can make a Transformation fail").

## 5. How do we train Transformers?

If we want the Business Builders to make the Transformation their own and work efficiently with IT, then we have to train them in Transformation. They must assimilate the **Approach**, the **Transformation Tools**, the **Foundation** and the Architecture of the **existing Model** that they need to Transform.
The prerequisite for any training is therefore to define the Approach, the Tools and the Foundations.
If it is a question of the **existing** Approach, Tools and Foundations, this training can be delivered by the internal teams who are already using them.
If the Enterprise wants to **improve** its Approach, its Tools and its Foundations, then it has to call in external experts who can bring new elements to the existing teams: concrete examples of Enterprise Projects that have succeeded using different Approaches, Tools and Foundations, and the conditions of success.
Provide suitable **Training** to these Builders to help them mature more quickly:

- Understand the **Digital opportunities** (Mobiles, Social Networks, Big Data, Cloud, Internet...) to benefit from them
- Know how to analyze new **Offers** made up of Goods, Information and Services which leverage Digital

- Know how to imagine an **extended enterprise** which involves partners and Customers in Processes which go way beyond the limits of the enterprise
- Understand what an **Enterprise Model** is: Product/Service Model, Operation Model, Transformation Model, Financial Model, Image, Culture
- Understand what **Enterprise Architecture** is: alignment with the Strategy, decomposition between Operations and Transformation, between the real World and the Model, seeking Synergy by Reusing Models and Sharing Resources
- Know how to execute a Project: **decomposition into Engineering Functions** (Understand the Context, Define the Goal, Build the Solution, Verify the Solution, Adapt the Solution, Configure the Solution, Deploy the Solution).
- Understand **linear Approach** and **Agile Approach**
- **Soft skills**: prepare the Business Transformers to separate the essentials, the structure from the detail, and not to only count on the "school-like" respect of the Transformation Project for their project to succeed.
- **Knowledge of the Foundations**: we first have to explain that is is possible to Build Product Offers or Solutions for very **specific** needs through **common** Foundations. Once this principle is accepted, we have to explain the variety of Solutions and Components that we can reuse.
- Detail each **Engineering Function** in its Business and technical component, especially
  - how to define the **Goal**
  - how to build the Business Entities and the **Business language**
  - define the **usage** standards
  - how to leverage the **social networks**
  - solve **security** problems

# 6. Should we encourage or retain colleagues tempted to launch their own activity?

Talents capable of leading a Transformation Program are rare. We have to retain them, but not by offering them an Operational management responsibility.

To retain these talents, we have to set up an "**intrapreneurial**" function: give an isolated team within the Enterprise the possibility to innovate without being a prisoner of the administration of the large enterprise.
We can also set up subsidiaries that are managed, during the growth phase, by those who have succeeded in difficult Transformation projects.

# 7. Transformation culture is specific to each Enterprise

Enterprises have extremely different cultures, which favor Transformation or not. As an example:

- the initiative can come from the top or the bottom
- the Enterprise likes or does not like Transformation
- the Transformation is separate, or not, from the Operations
- the Transformation teams are managed like the others or have a separate status
- the Transformation teams are often renewed or not
- the Approaches are mainly linear or Agile
- the Enterprise accepts risk or not
- the Operational actors accept or refuse the Transformations
- the internal population is young or old
- the internal population has job security or knows that their jobs depend on the health of the Enterprise
- ...

It is extremely difficult to change the Transformation culture.
It is advisable to really **understand the Enterprise Culture** before beginning any Transformations to understand the degree of difficulty that we will face.

# Epilogue

# Selling one's enterprise


## 1. Why sell one's enterprise?

Entrepreneurs like challenge, innovation and risk.
They may want to sell their Enterprise when the Enterprise is going well: its Model is pertinent, growth is there, the results too.
It is often the time when they want another adventure.

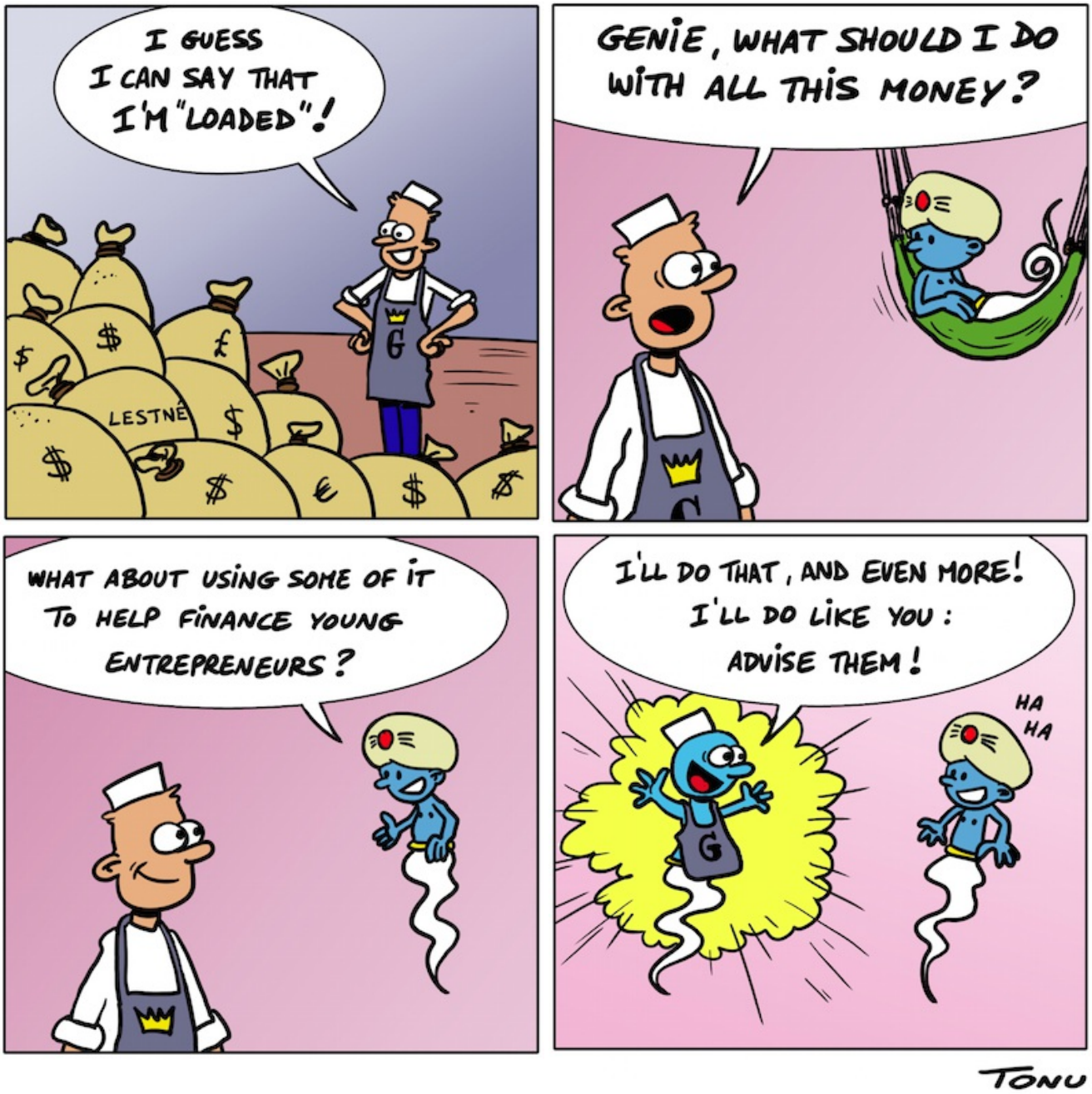
## 2. How do we go about finding a buyer?

This is one experience the entrepreneur does not have: it is better to get help from specialists who understand the Enterprise Model, know how to present it, know who the potentials buyers are and how to lead this type of negotiation.

# 3. What do we do after selling?

Carry on, but it is often for a limited time: it is difficult to stay in a larger structure with its own standards, its own Culture, when we were responsible for our enterprise.
If the acquisition results in a merger with another, already existing entity at the buyer's, the challenge may interest the entrepreneur.
They can also restart an activity or act as a business angel to help other entrepreneurs.

# Help young entrepreneurs




We can help Entrepreneurs financially, but we can also provide them with useful advice.
To reduce the failure rate in Enterprise start-ups, we can help the entrepreneur starting out.
The best coach is the one who has already successful started up an Enterprise, who has already gone through the challenges and who can, with his/her advice, save the novice entrepreneur time.
The same approach could be applied to large Enterprise Transformations: we should not hesitate to coach the project manager, so long as this coaching is carried out, again, by someone who has successfully managed large projects.

# CEISAR Glossary

*"Naming things badly adds to the misfortunes of the world"* (Albert Camus 1944).

**Principles**

- The selected terms must be understandable for both **Business** and IT Actors
- The definitions should be **short** and extendable by Role
- The glossary concepts start with a **capital**
- No homonyms
  - Service: Business-Service, IT-Service, Software-Service, by default means "Business-Service"
  - Architecture: Architecture Model or Architecture Discipline, by default mean "Architecture Model"

**Action**
Task executed by an Actor. Can be a Process or a Function or an Activity. Recursive: an Action is decomposed into Actions. Always named by a verb.

**Activity**
Group of Functions of an Organized Process executed by the same Actor at the same time.

Example: taking an order and the delivery are both Activities from the same Order Process.

**Actor**
One who executes an Action. A difference is made between a **Human-Actor**, an **IT-Actor** (programmable machine or "digital object") or an **Assisted-Actor** (when a Human-Actor and an IT-Actor are combined).

**Agility and Reactivity**
Agility is the ability to Transform fast and well. It enables us to reduce the time between the arrival of a new idea and its availability in the Enterprise Operations. **Reactivity** is the ability to Operate fast and well.

**Approach**
Transformation process. We generally distinguish:
- **Linear Approach**: each stage must be finished before starting on the next one (e.g., define all requirements before starting to build the Model)
- **Agile Approach**: we proceed by iteration

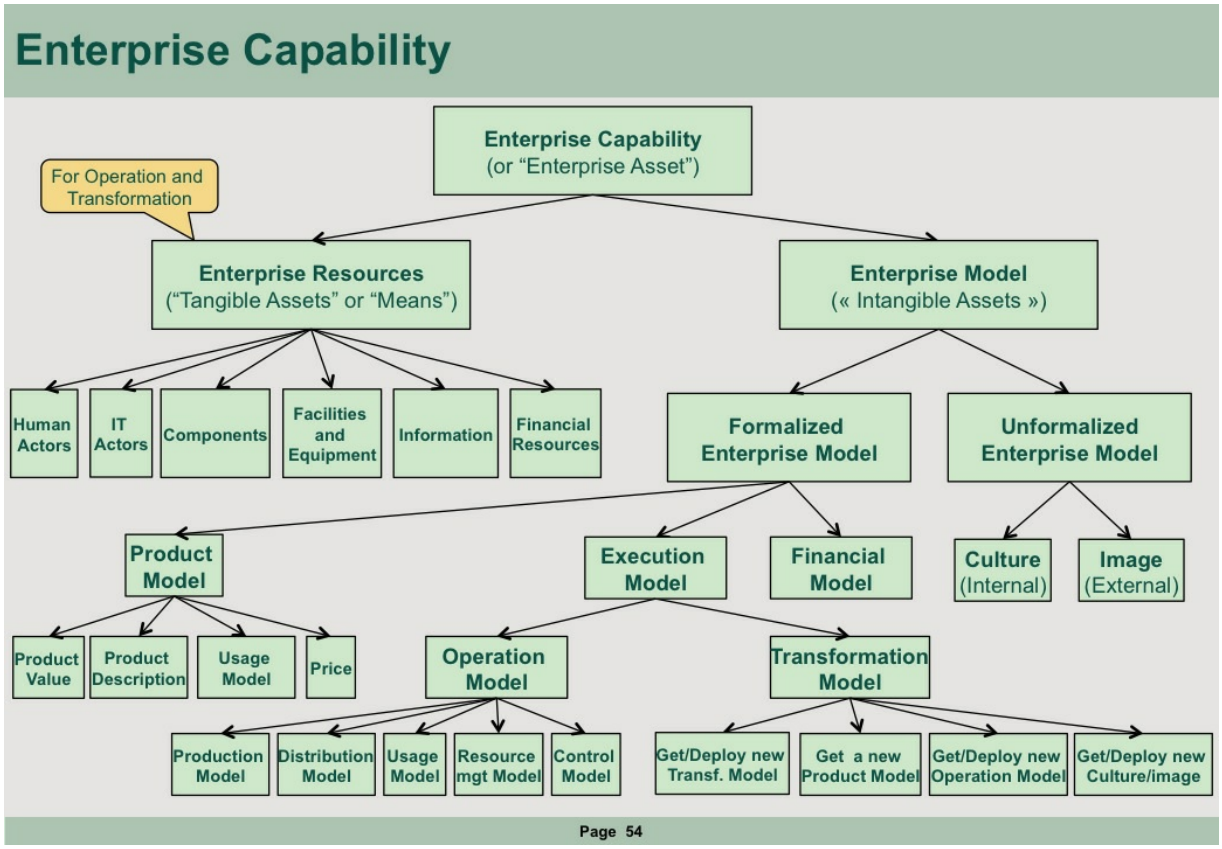
**Architecture**
2 meanings:
- Deliverable: **Architecture Description** represents the structure (in the sense of structuring elements) of a Model (cf. IEEE 1471 standard).
- Practical: **Architecture Discipline** represents the Transformation Processes to build the Architecture Description.

**Building a Model**
Action of creating a new Model or of modifying an existing Model. After being Built, the Model must be Deployed.

**Business Process**
Business Function chain (excluding Organization Functions) triggered by an independent Business Event and executed to deliver Value to an Actor who is the "customer" of the Process.
Example: Hire a new employee, Handle an order, Sell a Product...

**Business Solution**
Solution whose needs are specific to a Business, such as "Production" Solutions. It is often a Solution that enables us to differentiate ourselves from the competition.

**Capability**
Enterprise Capability is what the Enterprise is capable of doing with its Resources and Models.




**Commodity Solution**
Solution whose needs are the same in different Businesses.
E.g., Solutions for accounting, payroll...
They are relatively stable and are often implemented as Software Packages or Cloud services.

**Competence**
Defines what the Actor knows how to do (and not "should do"). For IT-Actors, the competence is called "Configuration".

**Culture**
Enterprise employee behavior Model.

**Customer**
One for whom the Product is intended. A Customer regroups different roles which may or may not be carried out by the same individual or legal entity:

- Beneficiary of the Product Value
- Product User
- Decision Maker
- Payer
- Recipient of the information

**Deployment**
Part of the Transformation Process that aims to adapt the Operational Resources to the new Model that has been Built: reorganization, training, installing IT hardware, loading software, data migration, adapting the premises,...

**Enterprise**

An Enterprise is an agent that brings Value to its Customers through a Product. It covers not only capitalistic Enterprises, but also public institutions, universities, research centers, associations... An Enterprise may be a legal entity, a subpart of a legal entity or a network of legal entities.

**Enterprise Architecture**

**The Enterprise Architecture Description** represents the structure of the Enterprise Operation Model and Transformation Model.

It generally takes the form of maps, which provide a global view, in order to better understand a complex Model: Process Map, Entity Map, Function map and Solutions map are the most frequently used.

**The Enterprise Architecture Discipline** represents all the Transformation Processes and principles necessary for building the Enterprise Architecture.

**Fact**

A Fact is a piece of Information describing the reality either in the Operations (e.g., data on a Customer) or the Transformation (e.g., project schedule).

**Function (or Rule)**

Action within a Process. A Function can call other Functions. The same Function can be reused in different Processes. A **Business Function** is independent of the Organization chosen by the Enterprise. Example: "verify a piece of information", "calculate a price". An **Organization Function** is added to implement the Organization. Example: "verify authorization".

**Foundation**

Groups together everything that can be reused for the **common good** in the Enterprise:

- The Transformation Model
- The Enterprise Architecture
- The Solution Components and Product Components
- The Products reusable in the different Organizational Units of a group
- The Solutions reusable in the different Organizational Units of a group

Reusing Models is a way of creating synergy and harmonizing the work methods in the different Organization Units.

**Goal**

What the Enterprise would like to reach at the end of a Transformation. Not to be confused with Enterprise Model: the Goal describes "why" we Transform, whereas the new Enterprise Model contains the "how".

A Goal includes:

- The scope: geographic, Product line, Process domain
- The objectives (productivity, time to market, new product, new Market, new partners,...) and the related indicators
- The constraints on the Transformation Program: budget, deadlines, extent of involvement of internal teams, Approach,...

**Image**

Model of the way the outside world perceives the Enterprise (customers and prospects, partners, competitors and the authorities).

**Information**

That which enables the brain to communicate, both in input and output.

A piece of information is either a Fact or a Model.

**Market**

Real or virtual space where Products are exchanged. A space is defined by all or part of the following dimensions:

- the Values sought by the customers
- the Product lines
- the customer segments
- the geographic territory

E.g., vehicle market for seniors in Asia

**Model**

Representation that simplifies the real world to better apprehend, memorize, communicate and modify it.
The Model can be **global** via Maps (business Entity maps, Process maps, Function maps, Block maps, Service maps...) or **detailed**.

**Actor Model**

Formalizes the Role of the Actors.

**Action Model**

Describes the instructions given to an Actor to ensure the proper execution of the Actions.

- For Human-Actors, the instructions are **documentation** (procedures, user guide, instruction manual, recipe,...)
- For IT-Actors, the instructions are **software**
- For Assisted-Actors, the instructions are documentation + software

We distinguish

- the **Process** Model ("Sell", "Produce", "Manage")
- the Model of the **Functions** which make up the Processes ("Fixing the price", "Print").

**Enterprise Model**

Formalizes the running of the Enterprise. It includes the Models that can be formalized:

- The Product Model
- The Execution Model
- 
  - The Operation Model
  - The Transformation Model
- The Financial Model

and the Models that cannot:

- The Image (for outside the Enterprise)
- The Culture (for inside the Enterprise)

**Execution Model**

The Execution Model has 3 parts:

- The **Actor Model**: the Human-Actor Model is called "**Roles**" (Seller, Producer, Administrator), the IT-Actor Model is called "**IT-Configurations**" (Hardware, Software, network).
- The **Action Model**
- The **Information Model** of the Customer, Product, Contract, Account...

**Information Model (or Data Model)**

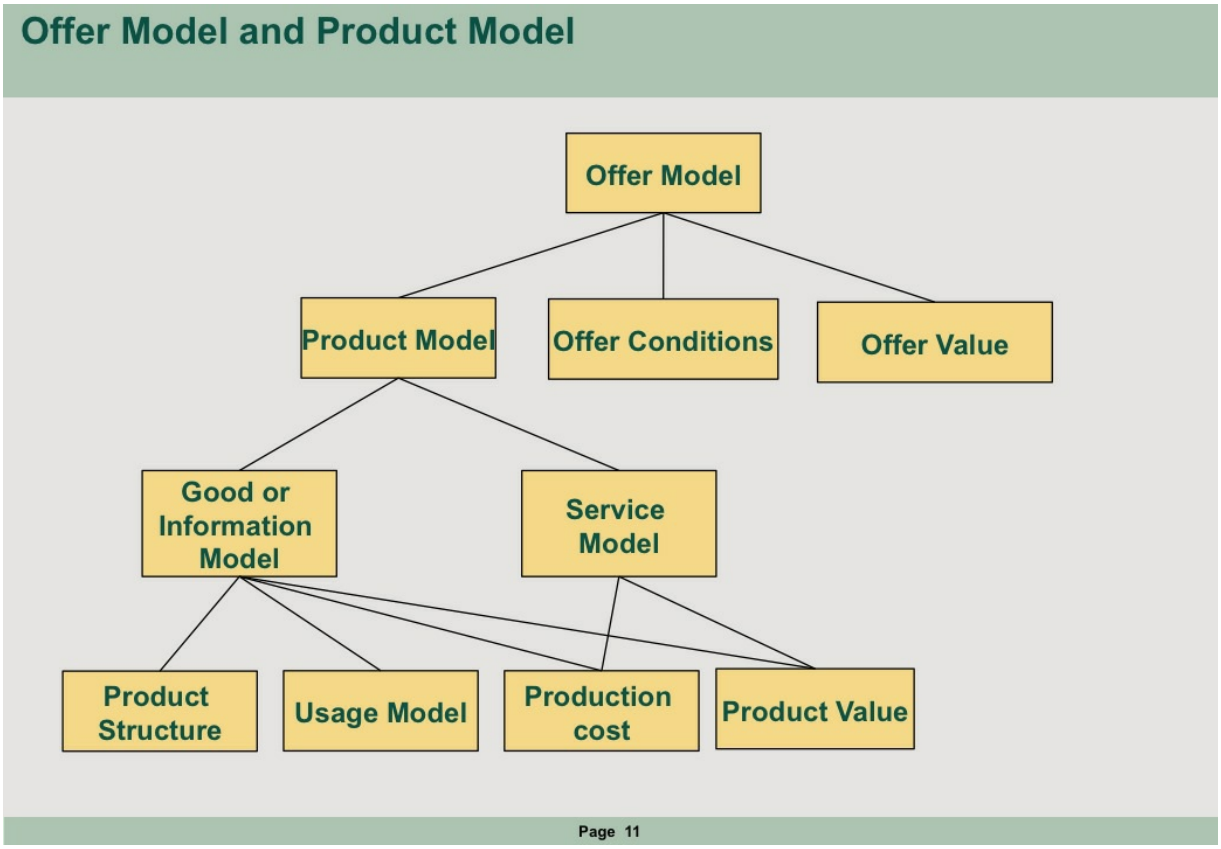
Defines the common Business language, the Object Models, their relations and inheritance. Describes how the Objects are identified, versioned, linked together and detailed with Attributes and Types.

**Operation Model**

An Execution Model which describes the Operations: Production Model, Distribution Model, Resource management Model, Management Model.

**Product Model**

Formalizes the Product decomposition, its Utilization Model, its Production cost and the Value it brings.




**Service Model**
Formalizes the Production cost of the Service and the Value it brings.

**Transformation Model**
An Execution Model which describes the Transformation: to Define the Goal, Define the Architecture and Foundation, Build the Model and Deploy the Model.

**Object**
Identifiable element from the real world. For example: Mr. Smith, Mr. Smith's Contract, Mr. Smith's Account...
A **Business Object** is required for the Business, independently of the Enterprise Organization. Example: Product, Customer, Contract or Account.
An **Organization Object** is required for the Organization of the work. Example: Organizational Unit, Position, Rights, Duty, Role.

**Offer**
What the Enterprise Distributes. The **Offer Model** is made up of:
- the Product(s) or Service(s) that constitute(s) the Offer
- the conditions of the Offer (cost, eligibility)
- the Value the Offer brings

**Operations**
All the Processes and Resources that contribute to delivering the Product to the Customer: essentially, Produce, Distribute, manage the Resources and Drive the Enterprise.

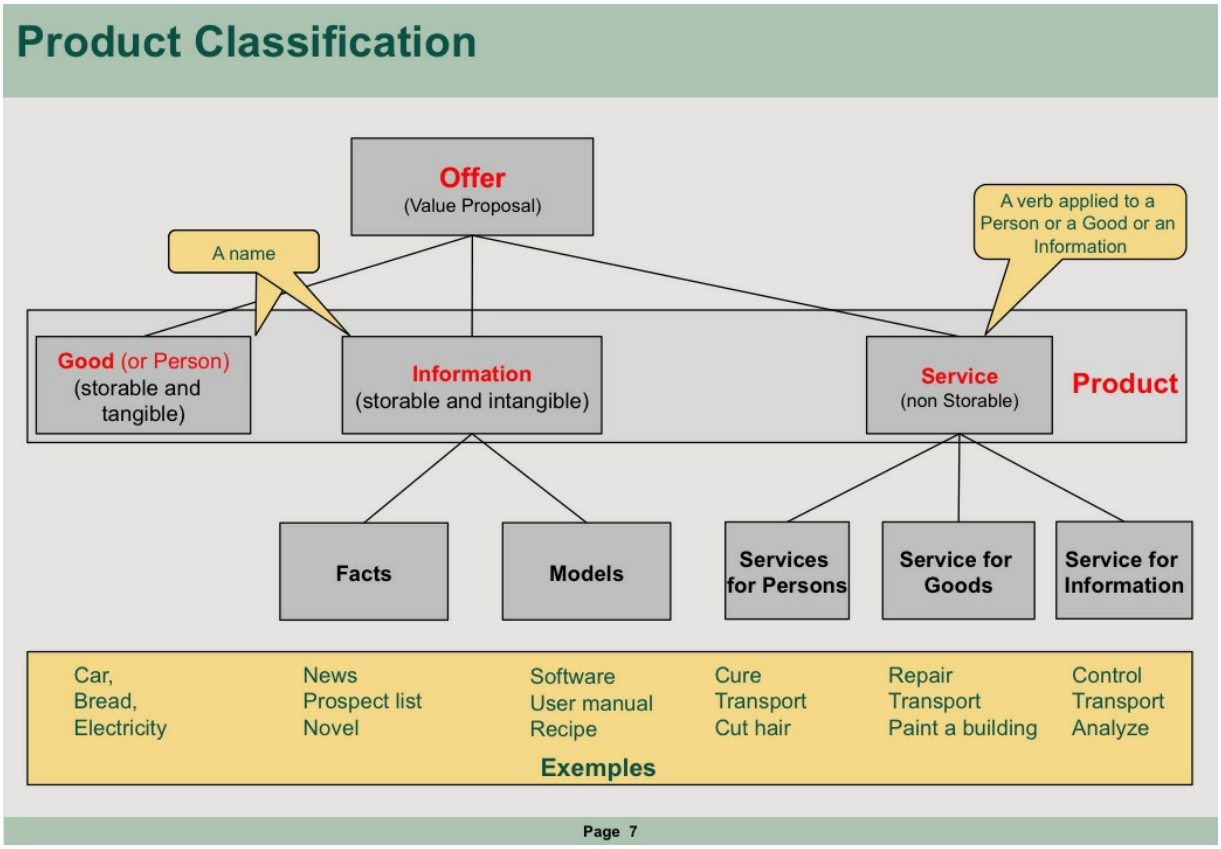
**Organizational Unit (or Unit)**
Node of the hierarchical structure of an Enterprise like Management, Department, Branch. The smallest Organizational Unit is that of Position, to which we can only allocate one Actor. Example: "Sales rep N°2 in branch X", "Assistant to the CEO".
The Human-Actors and IT-Actors are assigned to Organizational Units.

**Organized Process (or Process)**
Set of Business Process Functions triggered by an Organization Event.
Example: the Business Process "Manage a Customer Order" can be expressed in two

Organized Processes "Get the Customer Order" triggered by the Event "Customer inquiry" and " Deliver the Products" triggered by the Event "The truck is full".

**Product**

Object which brings Value to the Customer.
A Product can be **Goods** (like a car or a sandwich or electricity), or **Information** (like News, customer data or a Model), or a Service.
Goods and Information are stockable, a Service is not.
A Product line is a set of similar Product Models.




**Resources**

Means that are required to execute Operation and Transformation Models.
They are first and foremost the Actors: Human-Actors and IT-Actors, but also the Information, financial means, premises, Components, supplies and equipment.

**Reusable Component**

Model elements that can be assembled to build more important Models. There are 2 types of Components:

- Product Components which are part of the Product Model
- Solution Components which are part of the Operation Model

**Reuse and Sharing**

Definition of the Shared Resources or Reusable Models grouped together in the "Foundation".

**Role**

Rights and duties of an Actor. Not to be confused with an Actor's Competence.

**Solution**

Coherent grouping together of Action Models and Information Models.
A Solution is both

- software for the IT-Actors, called "Application"
- documentation for the Human-Actors describing Processes and Functions

Solutions have different levels of granularity: for example, a CRM Solution gathers together Processes, whereas a Pricing Solution gathers together Functions.

**Solution Component**

Model element which is reusable by different Solutions: Class, Function, information model, type, pattern...
Distinguish the Black Component (public interface, hidden implementation like a "black box") from the White Component (inheritance, types, patterns).

**Transformation**

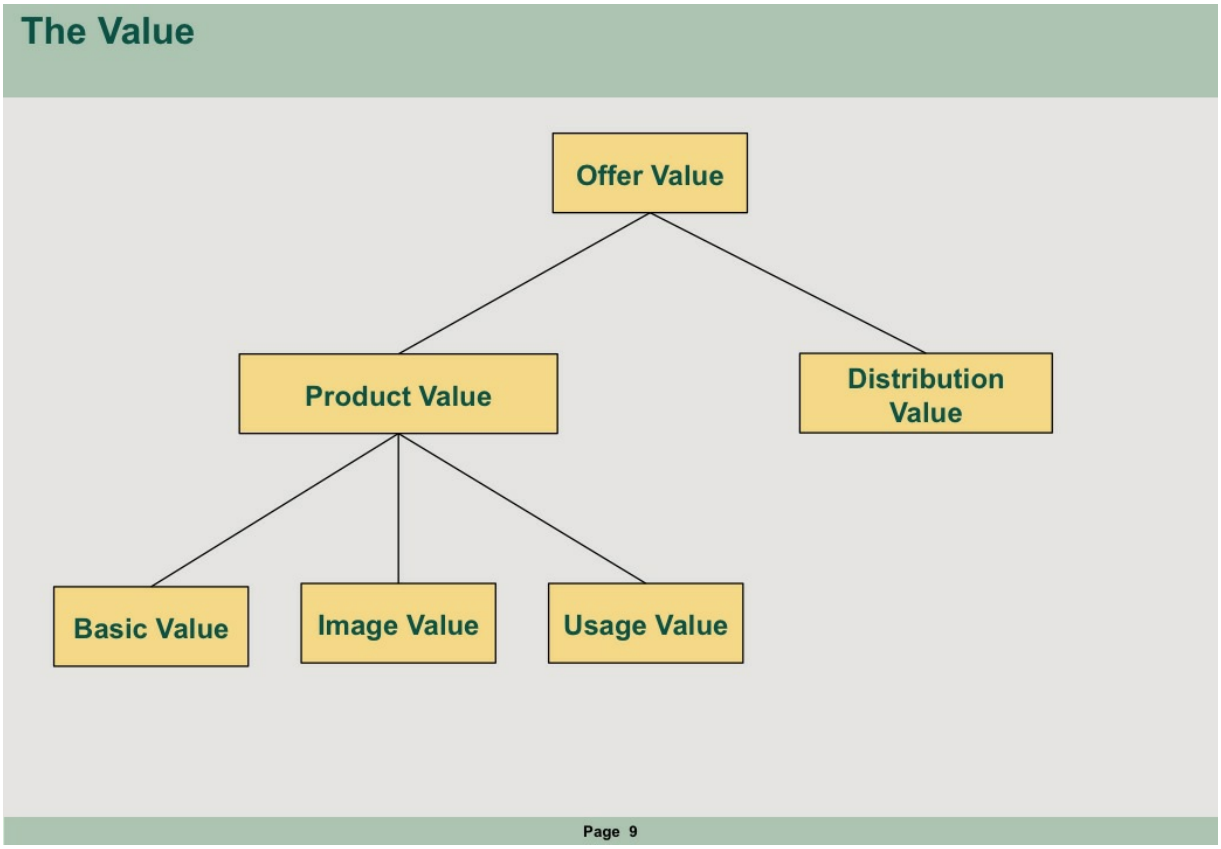
Creation/Modification of an Enterprise Model and adaption of the Operational Resources to this Model.

**Transformation Tools**

Tools to support the execution of the Transformation Functions.
E.g., Tools for simulation, mapping, requirements management, Process modeling, analysis/design, Development, programming, quality control, collaborative work, tests, configuration management, documentation, integration...

**Value**

What the Customer seeks: satisfy the basic needs, security, knowledge, image, simplicity, comfort, power, pleasure,...
The basic Value includes the essential functionalities of a product or Service: a vehicle enables us to get about with a certain degree of comfort, safety and performance.




**View**

Presentation of a part of a Model adapted to an Actor. The same Model has to offer different views: one for the Business expert, one for the IT developer, one of the operational actor, one for the architect...

**Vision**

The Vision describes the Transformation Goal (why are we Transforming?) and the new Enterprise Model, which enables this Goal to be satisfied (the new Offer Model, Operation

Model, the new Image or the new Culture).

# Contributors

The story of the Baker was written by the co-founders of CEISAR and Tonu, illustrator at Cartoonbase:

- Jean-René Lyon
- Pierre-Frédéric Rouberties
- Tonu

This publication has been enhanced through the participation of an open community of experienced professionals in the Transformation of complex organizations and enterprise start-ups. Anybody willing to formalize his/her experience in this area can contribute. To date, the contributors are (in alphabetical order):

- Marc Blangy
- Jérome Brun
- Laurent Mondemé
- Gérard Thomas

## Jean-René Lyon

Ingénieur Ecole Centrale Paris 1970
MS Computer Science Stanford University 1972

A unique professional activity for 40 years: **how to build well-structured information systems adapted to the enterprise strategy**, with 4 roles:

- As a **Consultant**, (1972-1978).
- As **Information Systems Director** in large financial institutions:
  - IS Architecture director - BNP (1979-1982)
  - CIO at Crédit du Nord: the design, development and implementation of a full banking system on an innovative architecture, which enabled important gains in productivity and reduced 'time to market'.
  - CIO at AXA Group: definition and implementation of the Group IS policy (1988 - 1992)
- As an **Entrepreneur**
  - Lyon-Consultants (1992-2000): consulting company helping large enterprises simplify their information system. Awarded first prize by Syntec for French IT start-ups in 1997. Reached 200 consultants in 6 years and was then acquired by CGI to become CGI-France.
  - Wyde (2001-2013): software vendor providing development tools and an insurance software package (http://www.mphasis.wyde.com/) built from the most modern methods and tools. 200 consultants. 50% of activity in North America. Acquired by HP-MPhasis in August 2011.
- As a **Teacher**:
  - founded CEISAR (in 2007), Center of Excellence for Enterprise Architecture at l'Ecole Centrale Paris (www.ceisar.org) to formalize the best practices in Enterprise Architecture and then use them to train both students and enterprises.
  - initiated the setting up of the Institut de la Transformation Numérique des Entreprises (ITNE: Enterprise Digital Transformation Institute) under the direction of the CIGREF and the CNAM.

## Pierre-Frédéric Rouberties

Ecole Centrale Paris 1992
Executive MBA: ESSEC & Mannheim Business School 2008

Fifteen years of experience in the field implementing and managing IS for large French and international groups:

- Information Systems Director at AstraZeneca France (2000-2007)
- He learnt how to manage the complexity of the enterprise and its information systems as well as how to:
  - Align the enterprise IS/IT with the business
  - Deliver a return on investment on the IS/IT assets
  - Define a target enterprise architecture
  - Manage major transformation projects (ERP, Sales Force Automation & CRM, Business Intelligence)
  - Optimize the day-to-day IT operations (application management, outsourcing infrastructure management) using best practices like COBIT & ITIL
  - Manage and develop a team

Since 2007:

- Deputy Director of CEISAR, chair of Enterprise Architecture at Ecole Centrale Paris, contributed to the development of case studies in large enterprises (Air France, AXA, BNPParibas, Michelin, TOTAL...) and training courses on these themes
- developed IS training programs for degree courses and executive education at Centrale Paris (Master of IS Architecture, Executive Certificate "Architecture and Cloud Computing", Business & Enterprise Architecture Program with EuroCIO), teaches enterprise architecture, the design and management of enterprise IS at Centrale Paris
- Freelance consultant in strategy and information systems

---

## Tonu

Graduated in history of art and archeology from Louvain Catholic University in 2008
Cartoonist with the agency CartoonBase since 2008.

A job focused around custom-made cartoons, mainly in the enterprise world, with a single goal: to communicate ideas. Increasingly used, communication through drawing can prove formidably effective. Based in Brussels, Tonu already has many customers from large French and international groups (BNP-Paribas, GDF-Suez, Generali, Crédit Agricole, Air France-KLM, Club Med, UNIFE, Johnson & Johnson, Linklaters, Baker & McKenzie, ...). He works on both commissions and producing live cartoons during different events (seminars, conferences, debates...). Some examples of his work produced live (USI 2012). You can follow him on his Facebook page.

---

## Marc Blangy

Graduate from ESCP and doctorate in Organization Sociology from Science Po. Paris

Marc Blangy joined the AXA Group in 1998 where he has managed many IT and Business Transformation projects in the various fields of the insurance business. For the last three years, he has been more specifically involved in the Digital Transformation of the AXA France IT Organization. Marc Blangy is now in charge of the IT division

for AXA GIE. He has followed on the CEISAR work since 2008 and has been directly involved in several white papers as a contributor or proof-reader.

## Jérome Brun

Ecole Normale Supérieure, 1969-1973
Agrégé in Mathematics, and PhD in Science

- Teacher and researcher at Université de Nice, 1973-1985
- Since 1985, career in IT Services in two groups (which have both been directed by former Finance ministers, a pure coincidence) :
  - GSI Group, 1985-1997, manager in various subsidiaries including Tecsi, a hightech gem
  - Atos Group, 1998-2013, manager in various subsidiaries. Latest job : Senior VP in charge of global Cloud Services
- Since october 2013, founder and CEO of BASEP Consulting, a consultancy dedicated to the Digital Transformation of businesses.

## Gérard Thomas

Supélec Ingénieur ESE (Electrotechnics, Automation) 1969
IAE DESS in Enterprise Management (Finance Specialization)

- HEC, Centrale, Corps d'état des Mines, Les Mines de Paris, Corps d'état des Ponts and Supélec:
  Teacher of degree courses and executive education.
- Harvard Negotiation: Correspondent under contract.
- APM: Expert for negotiation.
- Gérard Thomas Conseil: Managing director, advisor, trainer in Quality, Negotiation and Management.

Previously:

- 1988 to 1992: Furuno France - Marine Electronics (Japan), Training and Sales Director (turnover 100M French Francs).
- 1985 to 1988: French company PHILIPS - Head of Management Training (5,000 managers).
- 1976 to 1984: PHILIPS T.V. Video and Computer Science - Sales representative then Sales Director for the Paris region and national Sales Director.
- 1973 to 1975: Equipements et Technique - technical sales representative.
- 1970 to 1972: ENGINS MATRA - Engineer, satellite group project.

# Table of Contents