

# CEISAR Executive Summary on Foundation

This document summarizes the White Paper produced by the CEISAR on Foundation.

We will describe:

- Foundation is a powerful way of achieving Enterprise Goals: why?
- The benefits of Exchange Foundation and Building Foundation?
- What works today?
- How to decide Foundation?
- What is a good Foundation?
- How to get good Foundation?
- How can Solution teams Reuse Foundation?
- How can Package and Foundation coexist?

**A recap of CEISAR Terminology used by CEISAR:** “Operations”, “Transformation”, “Solutions”, “Models”

For each Business Domain (CRM, Accounting, Product design, Sales, Delivery, Business Intelligence) Enterprise **Operates** thanks to **Solution Models**: Actor Roles, Process Models, Information models, Function Models, together describe how Human Actors and Computer Actors Operate efficiently. It is the role of **Operations** teams to produce, sell and administrate the Enterprise according to the Operation Model: the majority of employees belong to Operations.

It is the role of **Transformation** teams to Build or Modify Models: they work on a Project basis rather than a continuous activity basis.

---

# 1 Foundation to achieve Enterprise Goals

## 1.1 From Goals to Challenges and Foundation

Enterprise Top Management defines **goals**: to increase Productivity, to merge companies, to improve Service to customer, to facilitate decision making, to be more agile and deploy best products and processes in the different Business Units.

Achieving these goals represents major **challenges** for the Enterprise, such as:

- **reducing operations complexity**: too many Products, Processes, Countries, distribution channels
- **developing Business Synergy**: how to centralize part of Model to help the different Business Units which redo the same work, while respecting specificities of each Business Line or Country
- **sharing and aggregating Information**: when each Business Unit owns its Information Model, it is very difficult to aggregate, compare and share information on Customer or Products
- **simplifying Enterprise Transformation**: projects to launch a new Product, transform a Process, or merge companies are too cumbersome and costly
- **computerizing Evolving Solutions** and not only Commodity Solutions: requires a different approach which progressively delivers successive versions of Solutions

This white paper explains that one of the most efficient way of solving these Challenges is to identify, isolate and carefully manage what can be Reused among the different Solution Models: this is called **Foundation**.

## 1.2 Operation Foundation and Transformation Foundation

As Enterprise activity is divided into Operations (manage the present) and Transformation (prepare the future), we need an Operation Foundation and a Transformation Foundation.

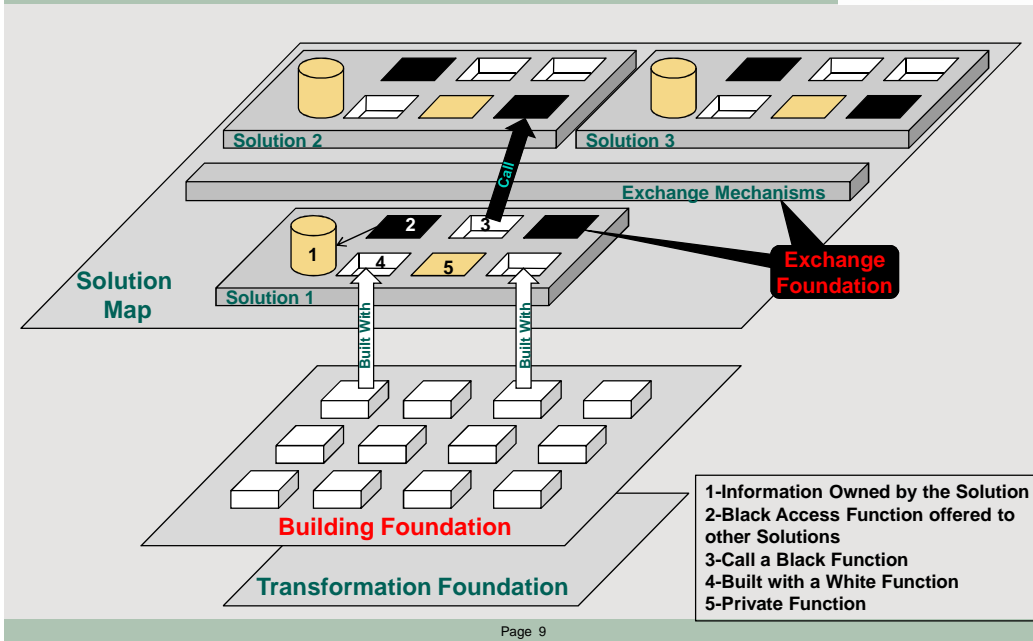
**Operation Foundation** groups all Models Reused in Operations: **Solution Models Reused** by all Business Units, **Reusable Components** to Build Solutions, **Information Models** including a Business glossary...

**Transformation Foundation** groups all Models Reused in Transformation, mainly Approach and Tools Reused to Transform the Enterprise: how to model new Processes and new Products.

Operation Foundation can be achieved in 2 steps:

First, Build and Deploy an **Exchange Foundation** (or SOA Foundation): if Components allow Solutions to **inter-Operate**, we avoid Information duplication, enhancing readability.

Then offer a **Building Foundation**: if 80% of a new Solution Model is made up of pre-Built Components, cost and time to Build it can be drastically reduced; "Reuse by Components" becomes an alternative to "Reuse by Package". Building Foundation only works if **same Transformation Tools** are Reused by Solution and Foundation teams.

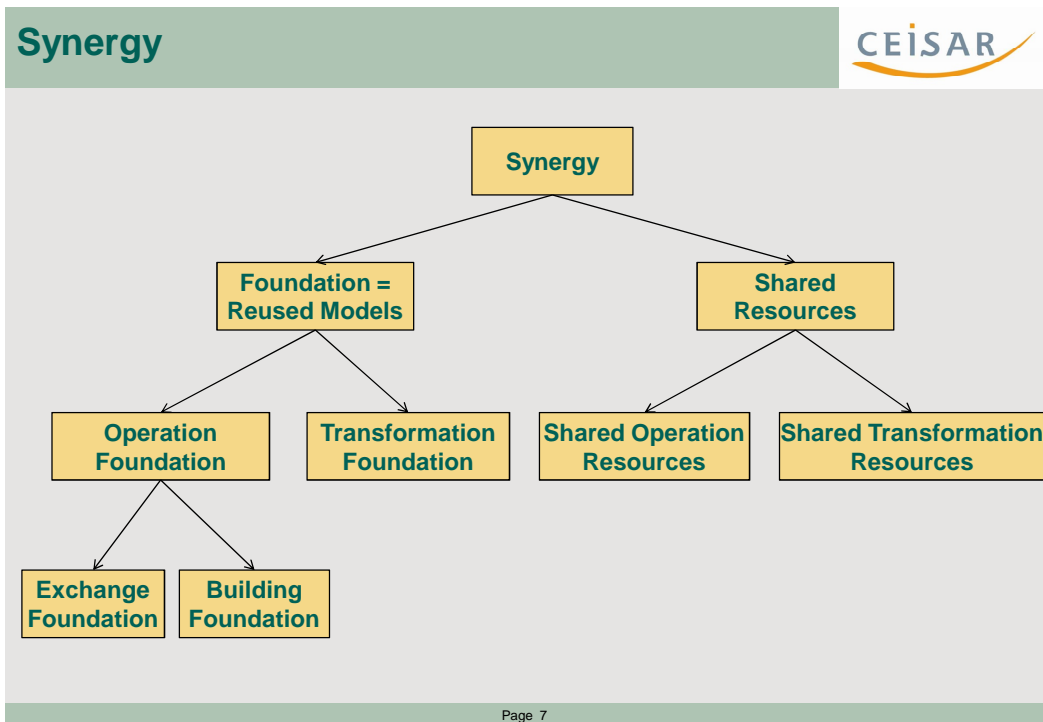


A Solution Model is reduced because it Reuses:

- **Black Functions** offered by other Solutions to interoperate. Ex:
  - the CRM Solution Model offers the Black Functions to access Customer Information owned by the Customer Solution
  - the Accounting Solution offers the Black Function to generate Accounting Entry
  - the Security Solution offers the Black Function to execute an authorization check
- **White Functions** offered by the Building Foundation. Ex:
  - the Process “Subscribe a Life Insurance Subscription” is a Specialization of “Subscribe an Insurance Subscription”
  - the “Subscription Windows” is Built with pre-Built UI components

Most Black Functions are offered by the Exchange Foundation, and White Functions are offered by the Building Foundation.

## 1.3 Synergy is not only “Reuse Models” but also “Share Resources”



Reuse Model is not the only way to develop synergy:

- If the Enterprise decides to centralize Human Resources Management into a single team which works for all Business Units: this is **sharing of Human resources**. *Note that they will work efficiently if they apply the same HR Model. It is also possible to ask each Business Unit to apply a unique Reusable HR Model, but keep HR units decentralized at Business Unit level.*
- If the Enterprise decides to centralize IT Operation centers, they **Share Computer Actors**
- If the Enterprise decides to Centralize all Customer Information in a unique Referential, they **Share Information**. *They could also Reuse the Customer Information Model without Centralizing Information.*

In this white paper, we focus on Reusing and not on Sharing.

Apart from CEISAR, many other organizations have developed recommendations on Foundation: they call it “Building Blocks”, “Reusable Assets”, “Enterprise Continuum”... We are all pushing the same synergy idea.

The white paper gives references to TOGAF recommendations: they develop more the management aspects (how to decide and organize), while CEISAR develops more the Engineering side (how to build). We also give references to IT Provider documentation (coming from Google, IBM, Microsoft, Oracle, Sales Force, SAP).

---

## 2 The Benefits of Exchange and Building Foundations?

**Exchange Foundation** allows Solutions to Interoperate; the benefits are important:

- **Visibility** of Enterprise Architecture: current and future Solution Maps help to understand the Enterprise Architecture, to align Solutions to strategy, to manage a project portfolio
- **Single Information:** exchanges between Solutions allow us to define what information is owned by which Solution. It is then possible through Exchanges, to query or to update Information owned by another Solution. Information is consistent and duplicate data entry is avoided.
- **Modularity:** it is possible to specialize Solutions by Functional domains: Accounting Solution, Business Intelligence Solution, Call Center Solution, and to concentrate specialized knowledge by Solution.

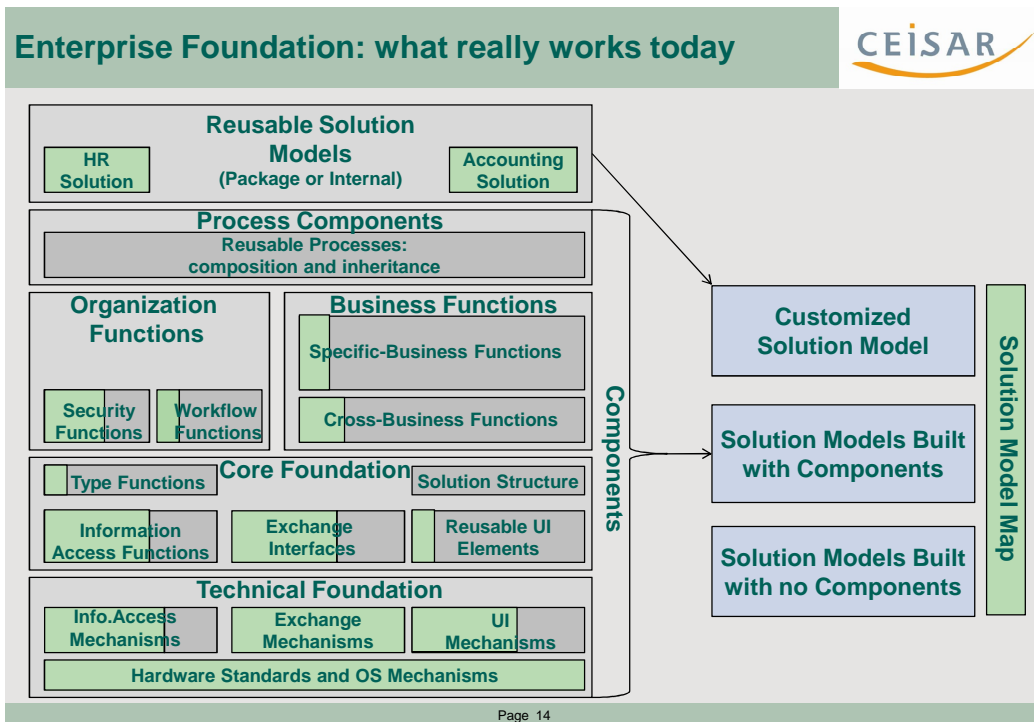
Service Oriented Architecture (**SOA**) is a well known example of Exchange Foundation

**Building Foundation** allows us to Build or Modify Solution Models to take advantage of pre-Built components. Obviously, Solution Models must be built with the **same Transformation tools** as the Building Foundation. This brings complementary benefits:

- **Agility** and time to market: gains can be much higher than with just Exchange Foundation. You can divide the time and money for new Solutions by three if Foundation is powerful (see after)
- **User Interface Consistency:** all Solutions may be Operated with the same User Interface, which reduces deployment efforts, increases productivity and facilitates staff mobility in the Organization
- **Global simplification:** the size of the Enterprise Model (easy to measure with Software) is dramatically reduced when the Reuse rate is high.
- **Lower risk** : the more pre-tested software you use, the less risk you take of generating defects in your Solution (it is also an advantage of Exchange Foundation, but at a lower level)

## 3 What works today?

### 3.1 What works in large Enterprises



This diagram summarizes what is in effect today in most Enterprises (in green).

Most Enterprises Reuse Solution Packages for Commodity Solutions and are developing Exchange Foundation. Very few have an Enterprise Building Foundation.

What works:

- **Hardware and OS standards** have been well defined by most Enterprises
- **Information Access Mechanisms:** DBMS standard usage but no Business Object access Functions, no Business Transaction Mechanisms, versioning is not always managed
- **Information Model:** some Repositories are Shared like Customer File, Organization File. Very few have defined a Business glossary. Some have defined an Entity-Relation Model for main Business Entities.
- **Exchange Mechanisms:** most Enterprises have defined a standard middleware and they widely Reuse it to allow Black Function calls between Solutions
- **Exchange Interfaces:** list of Interfaces between Solutions is generally documented, yet they are not always up to date if tools are heterogeneous.
- **UI mechanisms:** mechanisms exist to build all kinds of user interfaces; but there are very few mechanisms which enable management of Reusable UI elements like Type presentations, inheritance of Windows, composition of UI elements
- **UI elements Models:** few Enterprises Reuse UI elements
- **Type Functions** do not really exist
- **Organization Functions** exist for Single Sign-on; Security Functions exist, but there is not a single Reusable Security Function; Workflow and To-Do Lists are not generalized
- **Reusable Business Functions** are not very numerous: difficulty in specifying them, difficulty in splitting what is stable from what often changes; some Enterprises have started building SOA Business Components.
- Processes are not Built with **Process elements** by composition or inheritance (process Patterns)
- Enterprises have Built or Bought **Solution Models** Reused by the different Companies of the Group. This is mainly done for **Commodity Solutions** (no real competitive advantage, close requirements between different Enterprises), particularly true in the industrial domain.

- Enterprise Architects have developed **Enterprise Maps** to offer a global view of Entities, Processes and Solutions. It helps to understand the Enterprise Architecture, and align future Maps with enterprise Strategy. But:
  - these Maps are generally designed with **tools independent from tools to Build Solutions**: these Maps and the real world are not always synchronized
  - **Entity Map** is almost never available
  - **Process Maps** exist for Organized Processes, but do not exist for **Business Processes**: these Maps are attached to organization and must be modified when Organization changes.

## 3.2 What Providers are offering or preparing

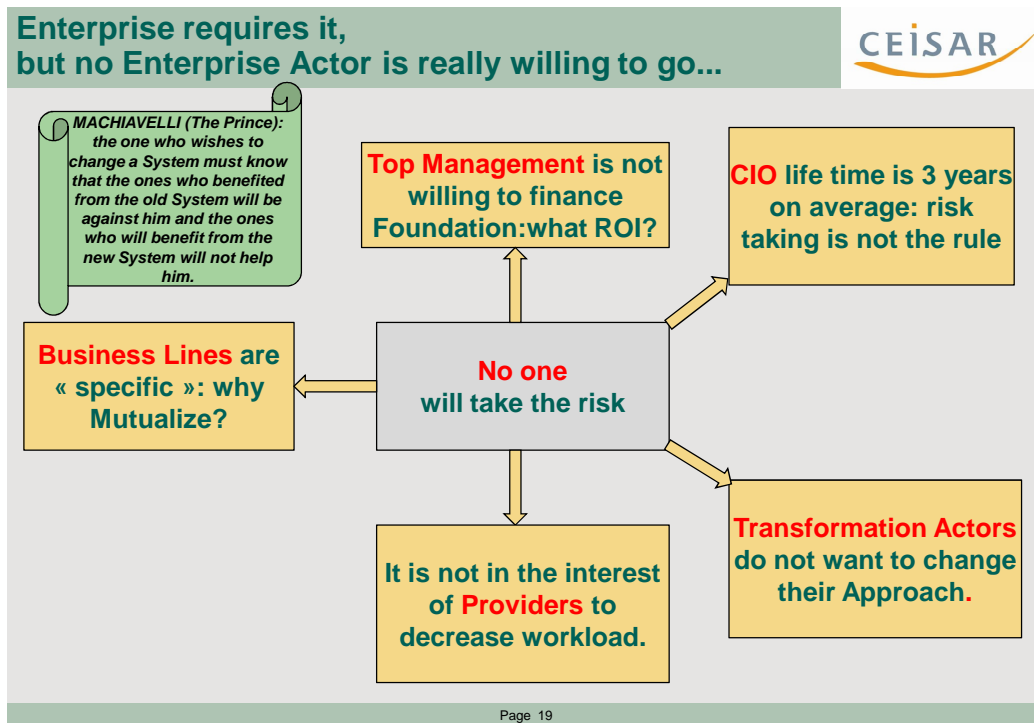
The trend is to offer Building Foundations.

Sales Force already does it for simple Solutions, but Google, IBM, Microsoft, Oracle, SAP and many smaller players are preparing offers to help Enterprises get good Building Foundations which will be based on:

- a **unique Meta Model** representing all the concepts required for Transformation: Business Entity, specialization, Relation, Process, Function, Attribute, Type ...
- a set of **consistent Transformation tools** which all respect the unique Meta Model: each modification executed by one the tools modifies the Meta Model and, so, modifies the view offered by the other tools. *For example if the Developer adds an Attribute to a Business Entity, it modifies the Business Entity Map provided to Business Analysts.*
- a set of **Basic Components**: Reusable Security Function , Reusable Workflow Functions, Reusable UI elements, Reusable Types
- a set of **Cross-Business Components** Reusable by any Enterprise: Actors, Roles, Organization Structure, Account, Contact...
- a set of **Specific Business Components**, specialized by Business Domain: they give priority to Bank, Insurance, Telco, Utility, Distribution and Industry.
- a set of **Maps by industry**: Process Map, Entity Map, Solution Map to help identify Black Functions and insert new Solutions.

## 4 How to decide a Foundation?

Creating a Foundation, using it, adapting the Approach, represents a risk which no one wants to take, as summarized in the following slide.



To set this change in motion, Top Management must be both convinced and involved. Because Foundation is "Common Good" it comes under the responsibility of Top Management and not Business Lines. If the Top management is not involved, no one will take the initiative.

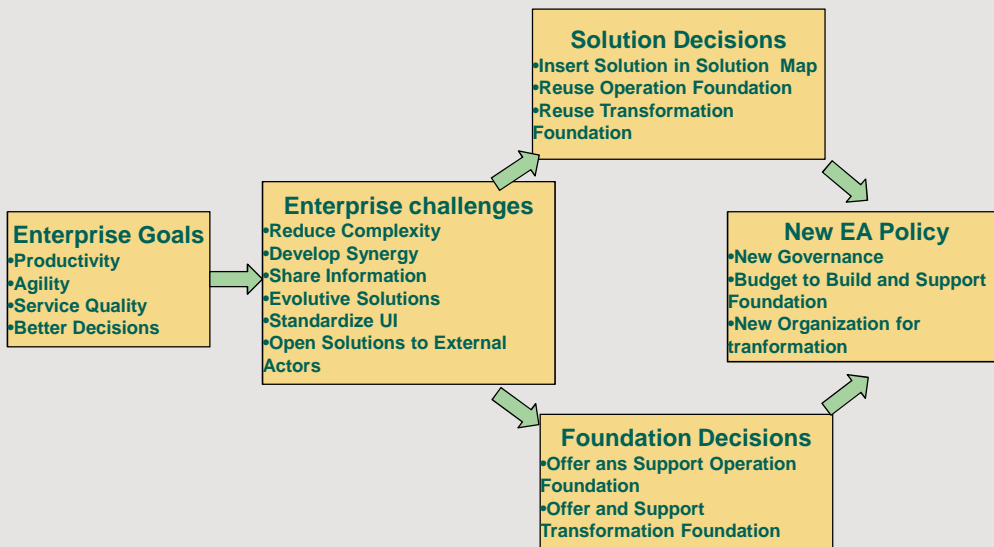
Yet if Top Management is not convinced, do not drop Mutualization, try to do it at **Business Line level**, and if Business Line Managers are not convinced, then apply it inside a **large project**: it will be less efficient but local success will allow a return later to upper levels.

Foundation is an **abstract** domain for Top Management, far from concrete domains such as finance, marketing, sales, HR... To convince top management, **align Foundation with Enterprise Goals**.

- Define **what Foundation is** in Business language
- Start from **Enterprise Goals**: Productivity, Agility, Quality, Cross-Selling, Mergers and acquisitions
- Explain how difficult it is to achieve these Goals: identify the **Challenges** such as "reduce complexity", "...
- Explain how **Solutions reusing Foundation** solve these challenges: deduct a realistic **action plan** and related Budget for **Foundation**
- Propose **decisions** to be taken by top management to obtain such Solutions and Foundation:
  - **decide New Governance**: ensure that Solutions Reuse Foundation, define **indicators** to check Foundation efficiency (reuse rate, flexibility, modularity...)
  - **decide Budget** to build/buy and support Foundation
  - **decide new Organization** for Transformation



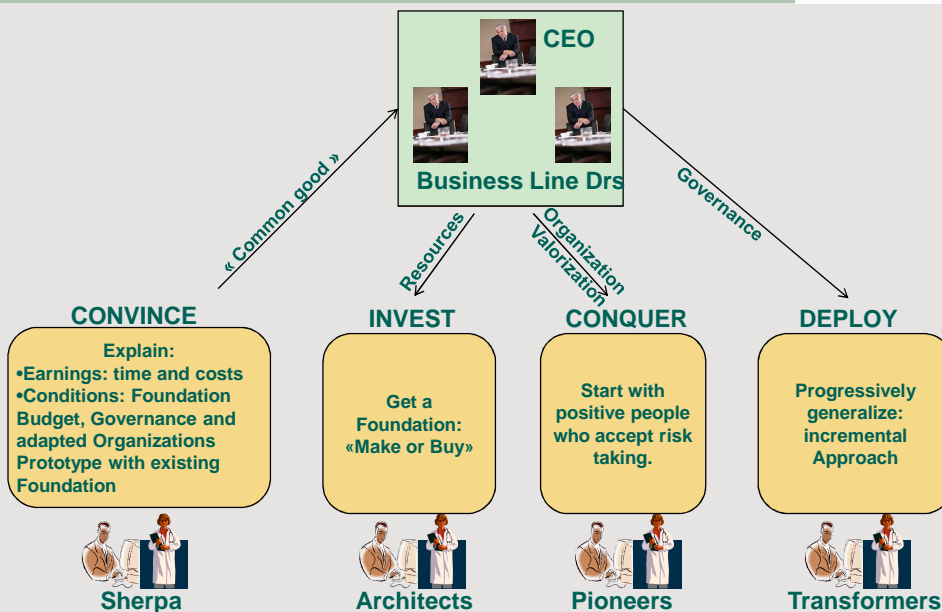
## From Enterprise Goals to Enterprise Architecture decisions



Developing a Foundation is an iterative task. The various elements of the Transformation and the Operation Foundations have to be gradually defined along several steps which can be related to the TOGAF Architecture Development Method phases. The ADM can be useful to define your global implementation approach for Enterprise Architecture, and especially to develop your Foundations

All Enterprises must have an **Exchange Foundation** to allow interoperability. Enterprises for which **Agility** is strategic should also have a **Building Foundation**. The choice is not between Exchange or Building Foundation, it is “Exchange Foundation only” or “Exchange Foundation and Building Foundation”.

## Scenario: Go

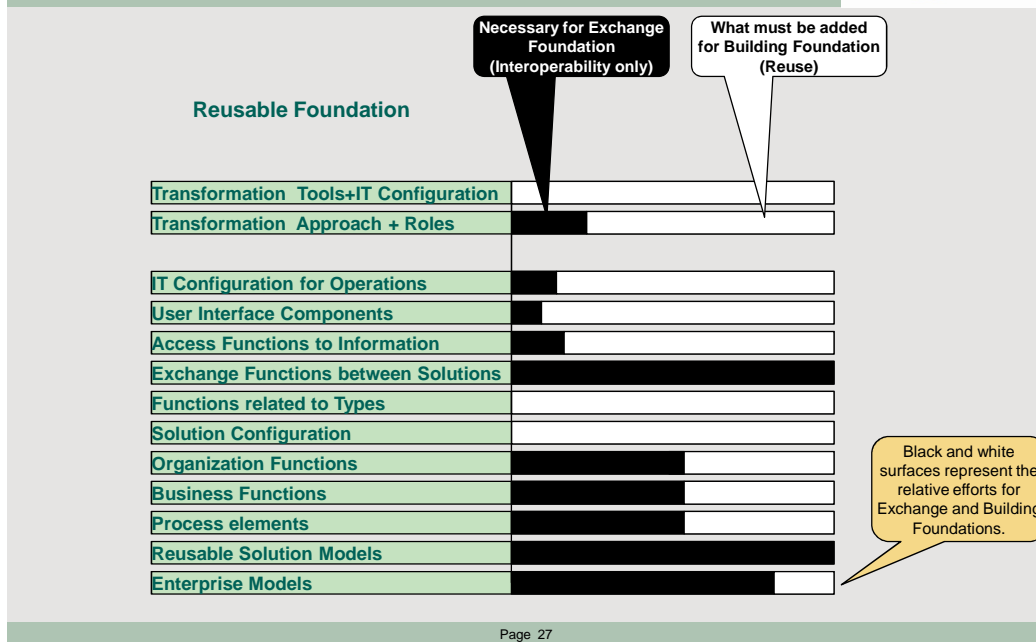


**Exchange Foundation** can be used **progressively** by existing Solutions, while **Building Foundation** can only be Reused when a Solution Model has to be created or deeply renewed. Building Foundation is more intrusive as you impose Transformation Tools.

A good strategy is to **start with Exchange Foundation**, because everything that is done for Exchange Foundation will be useful for Building Foundation: creation of the Foundation team, definition of a Reusable information Model, first level of Governance to respect Foundation... are all first steps for future Building Foundation.

# 5 What is a good Foundation?

## Exchange and Building Foundations



Page 27

### 5.1 What is a good Exchange Foundation?

A good Exchange Foundation includes:

- **map of Solutions:** who owns Information and Functions
- **Information Model** used to define exchange formats between Solutions: Business glossary, Entity-Relation Model, Attributes and Types
- **Exchange mechanisms** (Middleware, conversion and addressing Information)
- **Exchange Interfaces** (or “Adapters”) to call the Black Functions which are referenced inside a repository

### 5.2 What is a good Building Foundation?

Solutions are Built with the **same Transformation Tools**.

Components include: Reusable Types, inheritance of pre-built Classes, Reusable UI Components, workflow mechanisms replicated in different Solutions for task assignment, Process patterns... (see details in White Paper).

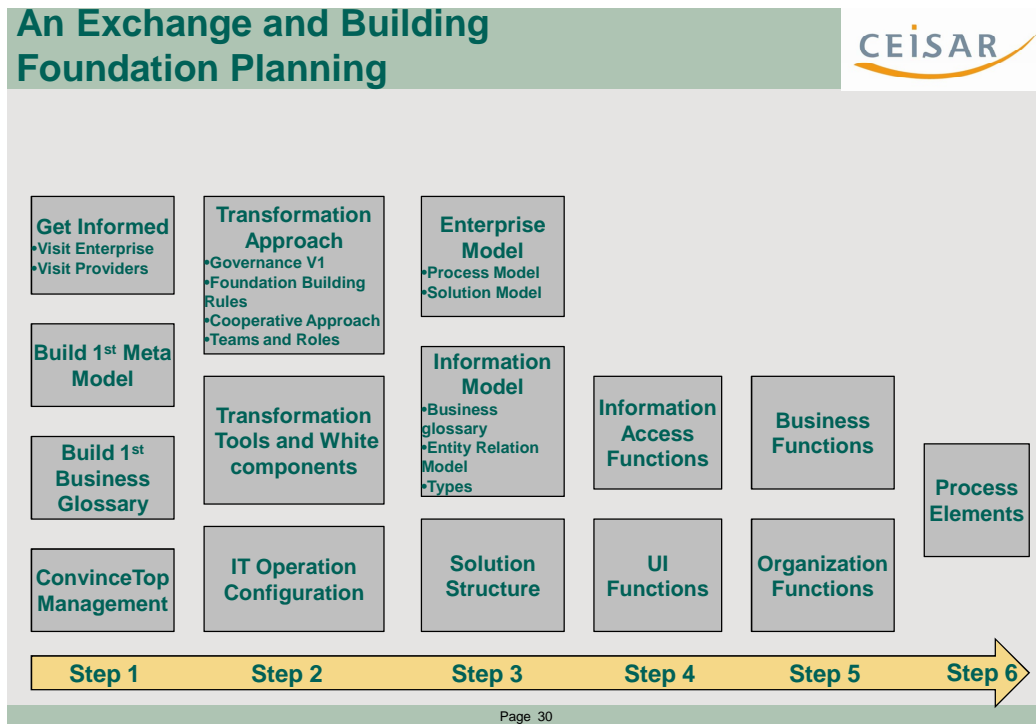
Efficiency of the Building Foundation relies on **quality of Tools**: prefer an integrated set of Tools rather than best of breed tools; look for powerful features such as versioning, Object oriented mechanisms, Business transaction... (see more detailed list in white paper).

In the white paper is given an example of an Insurance Package for multiple Business Lines: it shows that

total investment for Building Foundation is 40,000 Functions. This major investment allows Building of a new Business Line in about 2,000 Functions: each **Solution Model represents just 5% of the Foundation investment**.

## 6 How to get a good Foundation

The slide presents main steps when Building a Foundation. They are more detailed in the white paper.



The key difficulty is **abstraction**: how to discover what is common among Business Models which look so different?

When Business Operation Actors present their activities, they always focus on what is specific to their Business Domain, they never present what is common to other Business Domains. The task of the Transformation Actors is to progressively discover what can be Mutualized.

Between a full Foundation now and no Foundation at all, there may exist different levels. It is possible to start Building Foundation in **small steps**:

- group all teams working for common good into one Foundation team
- define a Business Glossary, and a Business Entity Model
- define a single Middleware
- implement the Cooperative Approach on some evolving project (see Cooperative Approach definition below)
- try out some modern Transformation Environments
- ...

You can also **buy a Foundation**. Yet a bad Foundation is worse than no Foundation at all. Be careful to select a Building Foundation offer:

- which works efficiently: best way is to Build a real Solution **Prototype**
- which is **customizable**
- which is **easy to use** for Solution Builders
- which manages **ascending compatibility**: Solution Models should not be modified when a new version of a Building Foundation is available

---

## 7 How Solution teams Reuse Foundation

### 7.1 Ease Cooperative Approach with Building Foundation

The **Contractual Approach** is the main approach used today in Enterprises. It is based on a sequential process (the “waterfall project”) whose main item is the “Contract” which defines all Requirements. It is adapted to **Commodity Solutions** for which all requirements can be defined (see White Paper on Agility).

Enterprises now require a **Cooperative Approach** (or **Agile Approach**) for **Evolving Solutions**. The objective is not to complete the Solution in the first version, but get it to a point where a set of capabilities can be tested and delivered to first users, and then to deliver other Versions at short intervals allowing to progressively discover needs and solve the compromise between what is desirable and what is possible: this Approach works better if a **Building Foundation** is available to guarantee Solution Flexibility and extension.

Reusing the same strong Building Foundation enables the breaking down of a large project into **smaller projects**: consistency is achieved by using the same Foundation and not by full detailed design of the large project before building it.

### 7.2 Governance

Governance must check that Solution Builders do Reuse Foundation.

The conformity check must be done **before** and not **after** Solution Project approval, so that the decision committee can check that Foundation is applied.

If Foundation is not suitable for a given Solution, the Project leader must **bring the proof**: he cannot ignore the Foundation which represents an investment made by the Enterprise.

### 7.3 Organization

If a Group is broken down into Business Units, an efficient Organization target is to apply these Rules:

**Rule 1: each Business Unit splits Operations and Transformation**

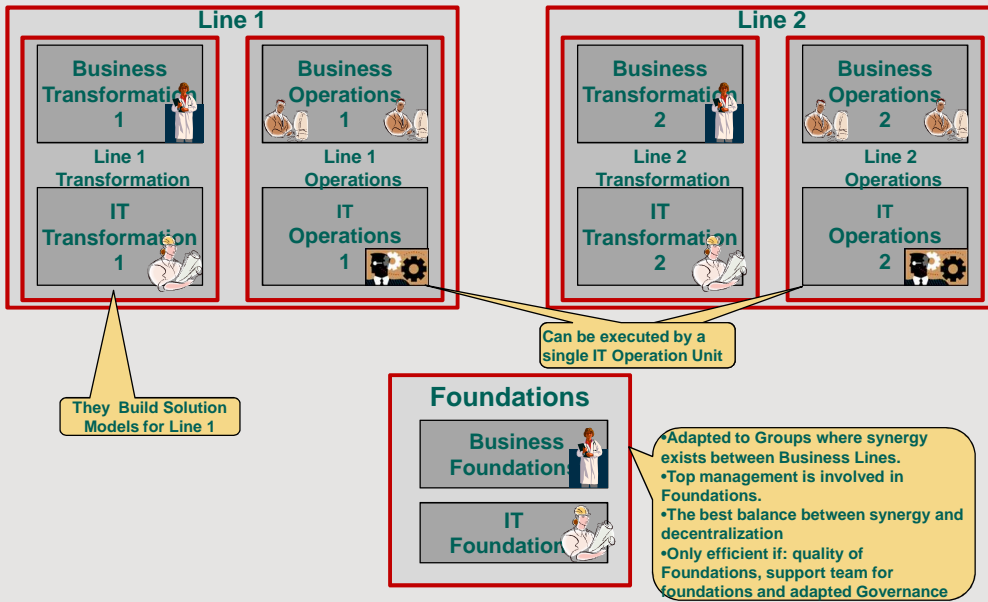
**Rule 2: Each Transformation team includes Business and IT people**

**Rule 3: a Foundation team is created at Group level which works for all Business Units**

**Rule 4: the Foundation team includes a “Support Team”**

With this scenario, the traditional IS department is shared in 3 ways:

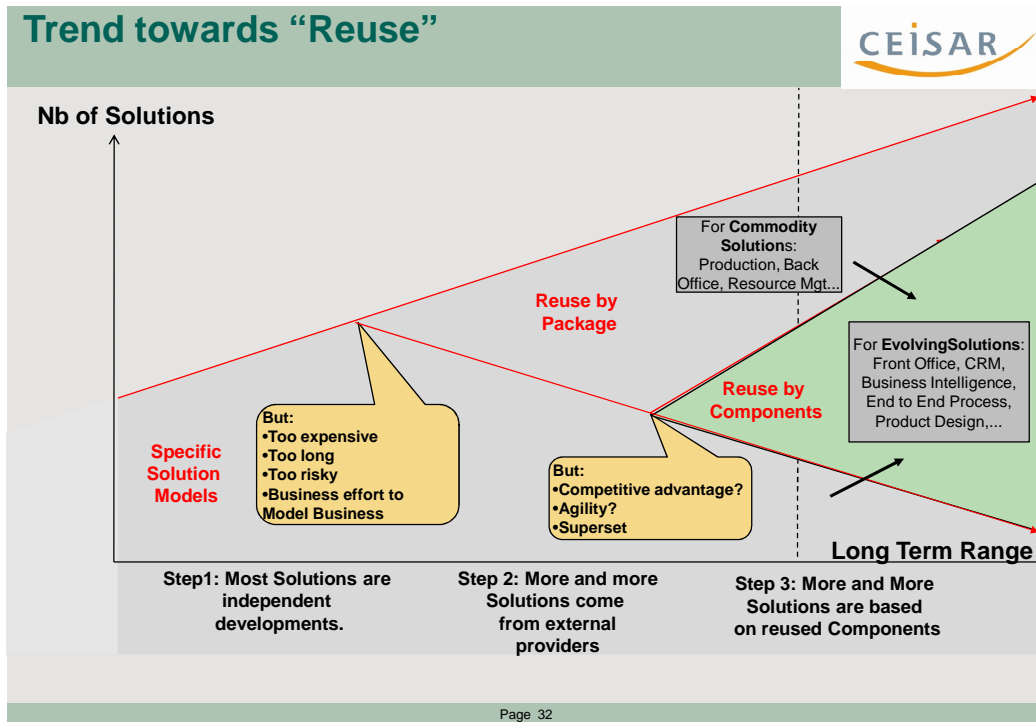
- IT Operations joins Operations
- Foundation joins the Group level
- Each Business Unit owns its Transformation team



**Exchange Foundation:** if an Enterprise only Reuses Exchange Foundation, the Foundation team is **light** (manages the Maps, defines the Reusable Information Model, defines exchange Mechanisms, manages Black Function repository for all).

**Building Foundation:** if an Enterprise also Reuse Building Foundations, the role and the size of the Foundation team are much more important, which is largely compensated by a decrease in size of Solution teams

## 8 Package and Foundation



**Step1:** Formerly, Enterprises used to Build **Specific Solution Models** by themselves. But the number of Solutions increased and the process became too expensive; it was too slow, too risky and constrained Business Analysts to describe all requirements.

**Step2:** So, industry of **Package Solutions** emerged: they offered good Solutions when requirements were close from one Enterprise to the other: Sap, PeopleSoft merged as large Enterprises providing Packaged Solutions. But these Solutions were adapted to Commodity Solutions and were not really adapted to Evolving Solutions characterized by changing requirements and competitive advantage.

**Step3:** "**Reuse by Components**" is now emerging as a complement to "Reuse by Packages". These Components are grouped inside the Building Foundation.

When Enterprises decide to use a Building Foundation approach:

- they use this Building Foundation for new **Evolving Solutions**
- they buy Solution Packages for **Commodity Solutions**

All Solutions (Evolving and Commodity) interoperate through Exchange Foundation.

### 8.1 Exchange Foundation

Generally, the key criteria in selecting a Package are:

- delivered functionalities
- cost for license, customization and deployment

Add another criteria: **Exchange Foundation compatibility**, at least for:

- **Entity Model:** business concepts, identifiers, relations
- **exchanges** between Package and Specific Solutions built with Enterprise Foundation
  - Information access from Package to Foundation Repositories
  - Information access from Foundation to Package Repositories
- **IT Infrastructure:** must at least interconnect with same middleware.

## 8.2 Building Foundation

Package Providers have Built their offer on a Building Foundation which is generally not delivered to Enterprise by package providers for different reasons:

- Package provider wants to **protect** its software
- Foundation is **not packaged** as a Software product unlike the Solutions
- **Revenue model** is **not clear**: customers are ready to pay for Solutions, they are not yet ready to pay for Foundation

But this may change in the future. Package Providers could propose not only Package Solutions, but also their own Foundation to customers because:

- Enterprises realize that Building Foundation is tough work: it costs more than they thought and takes a lot of time before reaching high Reuse rate; why not concentrate on specific Business Solutions **rather than Foundation? Revenue Model** should be better for Foundation Supplier.
- Helping its Customer to Build its own specific Solutions improves the **image** of Package Solution providers who are often accused of locking Functionalities inside a closed Solution
- **Coexistence** of Specific Solutions and Package Solutions is easier to achieve when they are based on same Foundation.

First target should be Enterprises which use many Package Solutions from the same supplier; they realize that the number of Evolving Solutions is increasing and do not want to invest in Foundation. They should naturally ask Package provider to also become Foundation provider.

The market is not mature today, but it certainly holds promise for Package Providers and Enterprises.

### Conclusion

Foundation is not a side topic.

Good Foundation can reduce efforts to Build and Deploy new Solution Models by three. Foundation can represent a huge competitive advantage for an Enterprise in terms of time to market and cost.

It can also be an efficient way of creating synergy in large groups.

But it is difficult to achieve. It requires:

- long term strategy
- top management involvement
- new organization and new governance for Transformation teams
- selecting competent Business Actors and highly skilled IT Architects for Foundation teams
- experience in Foundation Architecture
- budgets to Build or Buy/Customize Foundation
- new Transformation Approaches

You can proceed gradually or rapidly according to your strategy and your budget.

You can use Exchange Foundation only to begin with.

You can deploy it on a limited number of Solutions.

But do not ignore Foundation.

*Remark: for readers who want to know more, they can consult a far more detailed white paper on the subject.*