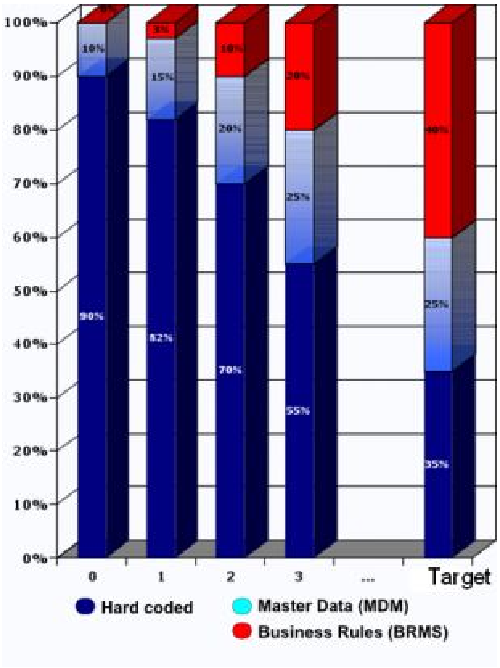
Agility Chain Management System (ACMS)

Companies should evaluate their IS assets through their stocks of Master Data (MDM), Business Rules (BRMS) and Processes (BPM). The right connection between MDM, BRMS and BPM forms the concept of Agility Chain Management System (ACMS).

The ACMS foundation brings more IS agility and IS traceability. Reference and Master Data, Business Rules and Processes become understandable and manageable by Business Users and all Stakeholders, not by IT specialists only which is the case when these IS Assets are hard-coded within opaque software or handle with tools having a low-level of IS governance such as spreadsheets: lack of version management, right and security management, traceability and auditability, etc.

As you can see with this figure (source: SMABTP insurance company in the field of Construction, based in France, Paris) the percentage level of hard-coded implementation of Master Data and Business Rules decreases for the benefit of business repositories brought by MDM and BRMS.

Thanks to these repositories, Business Users hold a direct access to Reference/Master Data and Business Rules to govern them in an efficient way: version management of Data and Rules, right management, complete auditability and traceability, etc



IT viewpoint about IS Agility characteristics

A commonly recurring issue in information technology is the engineering of agile systems, which can adapt fast enough to rapidly changing needs. In the last 20 years many initiatives focussing on special aspects of an information system have attempted to increase agility. In particular, we have seen the object-oriented approach for reusability of information processings and workflow engines for the flexibility at (business) process-level. There has also been considerable gain in experience in the use of rule engines, with the most common appliance areas being pricing and scoring clients. Companies have begun to realize the value added of these rule systems in terms of flexibility and easily maintainable extensibility, although fields of use stay still somewhat limited. We should also mention specific developments in the area of parameter management, especially within companies needing to customize their applications with respect to different contexts (language, legislation, distribution channels, etc.). Despite these approaches, one can observe very limited agility in even the most advanced existing systems.

First, companies have been working on certain aspects of agility of their systems, but never on their totality. For example, they have been concentrating on workflow, as well as the object-oriented approach, but without integrating rule engines. Such a partial effort may not suffice because agility is a chain, and it is only as strong as its weakest link. If a properly configured process in the workflow is flexible enough but the rules for its progression across different stages are not, the system will not be agile. A modification with respect to rules will implicate considerable costs with the usual claims on development time as well as the required skills for the efforts undertaken.

Second, the technologies of the time were less mature than they are now. For example, the management of parametrization did not yet have a quality market of software vendors and adequate products at its disposal. Companies had to build their own systems for parameter management with the usual constraints on maintainability and functionality. Moreover, most of these efforts within an enterprise got insulated or isolated in uncoordinated efforts spread across different projects. Nowadays, we have commercial solutions for master data management (MDM) specialized in the management of referential data and parametrization. A company now can easily equip itself with a unified solution for the management of Master Data and parametrization. and this is a considerable advance.

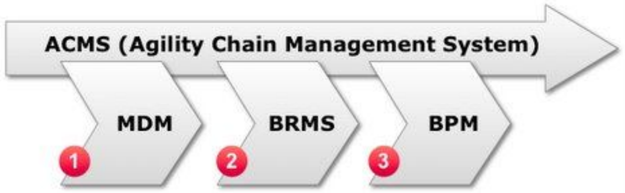
"Agility is a chain, and it is only as strong as its weakest link. If a properly configured process in the workflow is flexible enough but the rules for its progression across different stages are not, the system will not be agile"

"A company now can easily equip itself with a unified solution for the management of Master Data and parametrization, and this is a considerable advance."

Agility Chain Management System (ACMS)

Agility is within grasp

IS agility is within grasp if we address all aspects of an information system at the same time: Reference and Master Data and parameters (MDM), Business Rules (BRMS) and Processes (BPM), while making use of the appropriate methodology of design. If only one of these aspects is neglected, none of the efforts can yield any result. This is the principle of the "agility chain", and its determinant, the ACMS (Agility Chain Management System).



The figure on the right illustrates the order of steps in the implementation of the technical agility enablers.

Step 1: it corresponds to the gathering of Referential, Master and Meta-Data as well as parameters into a MDM (Master Data Management). This will increase their quality as well as centralize their management and retrieval.

Step 2: the BRMS (Business Rules Management System) will build up on the MDM - a rule may thus adapt to executional contexts. For example, a rule may test the total of a customer's purchase against a value stored in a table. This value may be a threshold, and may depend on executional contexts such as: organization, type of purchase, country, version, etc.

Step 3: finally, the agility chain is completed by the business process management, which can harness the benefits of the rule engine and parameter management. The rule engine increases the flexibility of the process across its different stages. Parameter management enables the configuration of the process' service calls, depending on executional contexts.

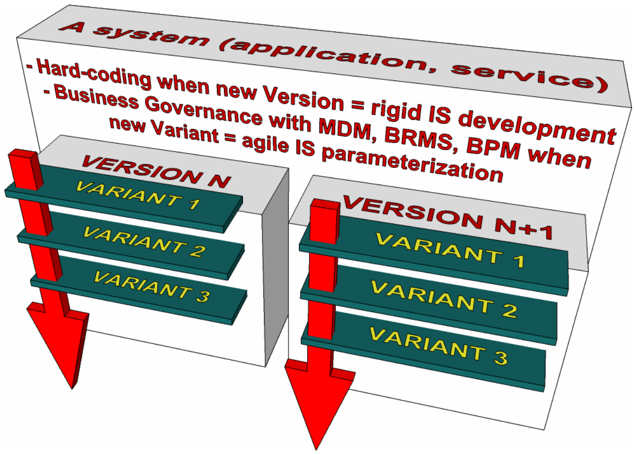
Obviously these three steps can be conducted with an iterative life-cycle project. However, the first step should be an Enterprise Architecture plan at the level of your data. For more information see the MDM modeling procedures deliver by the MDM Alliance Group (www.mdmalliancegroup.com) a sister community of Sustainable IT Architecture.

Version versus Variant

The distinction between version and variant of a system (application, service) is a prerequisite for the understanding of the path to agility-enabling. When a system is to be updated, there are 2 options to handle this modification:

1- If intervention on the underlying data is necessary, or altering its code, one creates a new version of the service and software development is inevitable. Since the piece of software has been modified, one has to plan a whole software production process which can reveal itself to be heavy.

2- If it is possible to intervene at the level of simple parametrization of data, of rules or chaining of services, the tools of ACMS - namely MDM, BRMS and BPM - are used without intervention in the code. This configuration points to a certain version of the software and leads to the creation of a variant (figure on the right). In this case, production is considerably accelerated.


A system (application, service)
- Hard-coding when new Version = rigid IS development
- Business Governance with MDM, BRMS, BPM when new Variant = agile IS parameterization
VERSION N
VARIANT 1
VARIANT 2
VARIANT 3
VERSION N+1
VARIANT 1
VARIANT 2
VARIANT 3

It is thus necessary to manage multiple variants per version of every system. If the tools of the ACMS are sufficiently user-friendly and secure, skilled users may create their own variants without any necessity for software engineering. The software itself has not been modified, it is thus not necessary to follow a whole production cycle. Merely data, parameters, rules or processes have changed, and thus the delivery time of the modifications is reduced. However, this flexibility doesn't eliminate the process of testing and in particular regression testing.