Agile Enterprise Architecture

PUMA

Architecture of a generation
of high-performance enterprises

Jean-Pierre Vickoff
Teamlog
Agile Enterprise Architecture
Table of Contents

Vectors of Enterprise Dynamics ................................................................. 3
"Service" Orientation and "Business" Process .............................................. 3
Action Space and Temporal Aspect ......................................................... 5
Incrementally Iterative Solutions Approach .............................................. 5
Multi-Concern Global Vision ....................................................................... 6
The Governance Aspect ............................................................................. 7
The Theory of Agile Quality ......................................................................... 7
The Agile dimensions of change ............................................................... 7
Survey of Compositions of Agile Solutions ................................................. 7
Services: Identification and Granularity ..................................................... 9
Documentation And Theories Of "Spaces" ................................................ 10
« Turn-Key » Components ......................................................................... 10
Extreme Agility In Development Quality .................................................. 11
The Agile Solution Space ........................................................................... 11
Foundations of Agile Development ......................................................... 11
The PUMA Project Engine ....................................................................... 13
A Globalist Paradigm .................................................................................. 15
Agile Enterprise Manifesto ......................................................................... 15
The PUMA Agile Enterprise Framework ................................................... 15
For A Free Method ..................................................................................... 15
Agile Enterprise Architecture

An organization is functionally Agile when its operational components (human resources, operational processes, information and technological systems) collaborate in synergy (formalized and instrumented) anticipating or collecting change with the purpose of dynamically compensating for it, then integrating it. The Agile organization is thus a community which continuously regulates its processes. In this context, Agility means efficiency for the immediate future on the basis of action led pragmatism pushed to the extreme. According to the Gartner Group, the Agile company must be “Real-time, service-oriented and event-controlled”.

Vectors of Enterprise Dynamics

"Service" Orientation and "Business" Process

A well performing Company is "service oriented". The source of its actions is in alignment with the requirements of its customers. Its means are the performance and the quality of its processes. As for the concept of agility, it sticks to a set of values optimizing the implementation of the components of this goal.

In order to be Agile, the company must continuously control the dynamics of the evolution of its business process, human resources and information system. At this level, agility requires a projection into the future which must be instrumented via formal techniques, such as rational anticipation. This practice makes it possible to anticipate the dimension of emerging technological or functional developments and their foreseeable impacts on the components of the organization.
To understand the architecture of the company on the basis of formal modeling is an essential precondition to any form of action in the process field. On the other hand, it should be obvious that it is useless to seek the key of evolution in an approach centered on the structure of the IS. The answer is not in the structure but in the **dynamics of the process**.

More precisely in a double dynamics:

- the first, in the present, is an immediate reaction of operational adaptation of the process;
- the second, in the immediate future, is a rational anticipation of technological and functional developments.

The first concern of the Agile enterprise is to strategically model customer requirements. This formalization includes the state, at present and in the immediate future, of the technological solutions likely to trigger an operational response. The second necessary tool is a modeling of the business process. It makes it possible for the organization to formalize the processes having to support its missions. It is only then that the concept of technical architecture appears which is applied to an information-processing system or to an industrial production system. On this last point, regarding information systems, the two currently dominant technological orientations are: processes instrumentation (BPM) by means of an independent functionalities orchestration engine, as well as the architecture of design and implementation of these functionalities (SOA).

*In the world of architectural framework buzzwords reigns the greatest inaccuracy. The absence of global vision in the case of the data processing specialists is certainly the cause. Their proposals are currently limited to technical or applied aspects of an information system. This vision of urbanization that they describe as “Company” disregards the modeling of competencies, motivations and types of collaboration authorized with human resources and neither does it imply an anticipatory model of the environment in evolution.*
Action Space and Temporal Aspect

Moreover, even at the level of the IS, the approaches are dramatically static and mono temporal. At same time, for the Company to be Agile, it must operationally combine the tensions created by the differences between a present of constraints, a past of structures and a future of emergences. In a simple and exhaustive answer, the Process for the Urbanization of Agile Methods (PUMA) materializes the action space where these multiple tensions are being expressed, managed and anticipated.

In this approach, the dynamics of company evolution are structured in 6 models (figure 1) that improve upon the "Gartner Architecture for Real Time Reactive Exchange" reference structure. A more complete version is published in "Systèmes d’information et processus Agile (Hermès, 2003)".

Incrementally Iterative Solutions Approach

This Enterprise Architecture offers at last a formalized and justified answer to the question “According to what and how must we change the process, the IS as well as competences of our human resources”. The dynamics of these principles is detailed on Entreprise-Agile.com; On its own, the "Agile Solution" (Business Projection) model structure is given by 4 classes of concerns, characteristic of the new Expression of Requirements. Indeed, the stratification of the traditional approaches was not only completely upset in its levels of abstraction, it conceded its place to a multidimensional holistic approach perfected on multiple levels during an incrementally iterative refinement.

Nevertheless, these aspects are explored in a fundamental order. On the other hand - all the relative complexity of this operation and its relevance reside in this principle - they must be globally anticipate in an iterative way, in order to account for the globality of the interrelationships and the induced dependencies.

The Requirements, are initially regarded as "Visions", becoming "Problem Frames" by further refinement, then "Specifications" and finally "Services". This last aspect is a central point and it is detailed further in terms of identification and granularity.

![Figure 2. Agile Solution Model (Business Projection)](image)

The solution model is unique in its structure and approach on formalization. On the other hand, according to the type of the project, the emphasis will be put more precisely on specific aspects.
Multi-Concern Global Vision

For a rational mind, to level off “action” models, "structure" views, solution approaches and methods of project managing, represents a "heretic abstraction". However, it is precisely this reality that the organizer and the data processing specialist are confronted with. Fortunately, the philosophy of Agility - pragmatic empiricism - offers the capacity to allow for abstract bridges between representatives of different levels, without the need to justify them other than by their utility.

The complexity of the current solutions is such that the project leader is confronted with the necessity of concretely having a modelled vision which at the same time is global and multi-concerned and where enterprise dynamics, process control, solution approach and implementation views materialize on the same level while being located in different logical layers. In this diagram (page 1), the principles of Agility relating to urbanization or to a process are treated within the dynamic model, while the concept of “view” is only a static representation. In addition, the solution model includes a project engine comprising proposals for Agile architecture (SOA) and a piloting method that varies from Agile to extremely Agile.

It is neither by error, nor by chance that this synthesis does not appear at the beginning of the presentation whereas it seems it could frame the other models. An Agile explanation would be possible through the use of a metaphor (XP collaborative practice n° 2), such as "vehicle movement".
To highlight the gap that concretely separates pragmatic empiricism from Cartesian rationalism, I present you with a simple anecdote related to the definition of Agility itself. The requesting person expected the elaboration of an elementary decomposition similar to the sentence introducing this communication. My answer was "Agility is Google". Indeed, it is enough to observe this elegant and powerful business model to understand the concept of Agility in its whole sense (commercial, financial, technological, temporal and even trans-organisational).

The Governance Aspect

The addition of the functionalities covered by the 6 modules and the 9 interactions that compose the Agile Enterprise architecture, naturally covers in a relatively similar but reduced way the principles of Governance covered by ITIL and Cobit. The standard responsibility actions (to plan, to implement, to run and to supervise) are approached pragmatically. Moreover, Puma overflowing the IS framework to include all "Enterprise" issues but, on the other hand, it does not directly cover financial aspects and security. From a different angle, the two standards specify that which the company needs, but not how to obtain it, whereas the PUMA action system determines the two aspects.

Thus, PUMA would be a form of Ultra Light Governance, whose simplified vision will be appropriate for organizations that could not, or would not, like to initiate complex and heavy controls while aspiring to reasonable control of their processes.

The Theory of Agile Quality

Agility is also often confronted with the concept of standards of quality. The comparative definition suggested by PUMA is as follows:

- Traditional quality imposes an obligation and a formalization of means but does not specify for which result.
- Agile quality proposes an obligation for the result but requires autonomy for the means.

The Agile dimensions of change

Survey of Compositions of Agile Solutions

Currently, with regard to business vision and the applicative architecture required to support it, two prevalent approaches collaborate complementarily in the search for the optimal solution. On one hand, the Business Process Management (BPM) is related to the control of the operational process. On the other, the Services Oriented Architecture (SOA) brings an elegant and urbanized solution to the development of information system components:

- BPM is based on business modeling in order to optimize and adapt the whole set of activities. By remodeling the organization around its core processes, BPM stands out as the main drive for operational performance.
- SOA, through its approach of designing and implementing services in the image of independent applicative "bricks", facilitates process instrumentation. SOA has the advantage of producing simple, modular, loosely coupled components, therefore allowing for quick recomposition of the applicative frame of functionalities that they provide.
BPM projects are an option of PUMA controlled projects (figure 5).

The SOA technical architecture consists in breaking up functionalities such as they are perceived by the user, into a set of basic elements named "services". The following stage of the implementation consists in accurately describing their interaction diagram within the framework of a business process. The services will then be developed and deployed in the form of independent software components. The objective of this type of architecture is to suppress the production of bloated applications that are unable to support an evolving business logic. Ideally, the encapsulation of each service must guarantee its reusability and its interoperability. Finally, the most valuable advantage of the "service" is the ease of evolution of the framework into which it fits. Currently there are no official specifications of an SOA architecture, only a set of governing concepts and standards - but the subject is too technical to be approached here.

Both SOA and BPM collaborate in the same strategy: to ensure that Enterprise objectives are attained within a context of quick evolution and often related to an obligation of concurrent differentiation.
Services: Identification and Granularity

Identifying “services” and their granularity represents the essential concern when designing SOA. Since, it should be pointed out, "Nothing should be taken for granted. Nothing is given. Everything is built."

PUMA’s answer comprises two derivations emerging from two self-validating UML models and a classification of service components taking into account their stability:

- **Functional-organisational derivation** (*presentation layer*). It is instrumented from **use cases**. The ideal granularity of the "useful service" corresponds to the smallest decomposition of a case, for which the "service" request makes it possible to obtain in return functionally usable information, i.e. a complete « customer useful service » type.

- **Business derivation** (semantic / stability) (*data model layer*). It is instrumented from **classes** (or **Entity-Relation**). The granularity of the "useful service" is given by the possibility of composition between classes whose semantic range had authorized a comparison (generally a **business object** regrouping functionally indivisible classes).

Object Stability

To approach this concept, it is necessary to understand the modeling axes. The object oriented approach imposed the prevalence of the modeling axes in the design of an information system. Component re-use, redeployment and refactoring have become the main objective, therefore to ensure the perenniality of a system it is necessary to implement "design for modification" techniques. Thus, an effective design is based on the complementarity of three distinct axes of modeling:

- the static axis fixes the data structure,
- the dynamic axis defines the process,
- the functional axis details the processing.

The concept of adaptation is the target aimed at by the control of these axes. An organization incapable of quickly mutating if its information system imposes heavy refactorings at each step of its forced walk towards productivity. The role of these axes is to limit the range of modifications during development and during the whole applicative life cycle.

The static axis is the most stable. By definition, to reach it at its base, a major change in the company’s business is required. The dynamic axis sticks to the organisational aspect. Only a reorganization of actors and processes may have an impact on its definition. The other surface changes, the manner in which the products or offered services are being treated, are limited to a redefinition of the processes through the functional axis.

The composition of business objects assembled in order to respond to a request for a "service", will seek to limit associations of classes belonging to different layers of stability. In a multi-layered architecture, the other rules are possibly used as constraints or assertions facilitating the validation of the solution.

*Note: This approach was published on several occasions and for the first time in the report entitled “Re-engineering of Application Development …” (Gartner Group, 1999) as well as in “Piloter les projets informatique …” (Editions d’Organization, 2000) under the title of "Concept de stabilité des objets ".*
Documentation And Theorie Of "Spaces"

Regarding information systems, even the most sophisticated situations lead to binary execution. On the other hand, to paraphrase one of the geniuses of relative complexity: "Things must be done as simply as possible, but not in a simplistic way."

During the resolution of situations, even complex ones, that the organizations must manage, a structurally simplified approach can solve many of the difficulties of formalization. It is thus on the basis of a dissociation of responsibilities between the "Requirements Space" and the "Solution Space" that a PUMA project is built. This dissociation of formalization spaces does not involve a separation of the implied resources, as the contrary, it involves Agility in terms of developing under the relatively continuous presence of the "customer".

For PUMA, at the time of a project, the two work-spaces are as simply structured as a form with questions and answers. They correspond on the "customer" side to the expression of requirements and on the "technician" side to the expression of a solution (see table below)

<table>
<thead>
<tr>
<th>Requirements Space</th>
<th>Solutions Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional questions</td>
<td>Technical answers</td>
</tr>
<tr>
<td>Requirements</td>
<td>Project framing</td>
</tr>
<tr>
<td>Expressed constraints</td>
<td>Accepted constraints</td>
</tr>
<tr>
<td>The quality of the application validation</td>
<td>Project performance (monitoring)</td>
</tr>
</tbody>
</table>

Thus, a PUMA project comprises (at its root) only two documents with identical structure that follow the new requirements expression strategy ordered by the Solution Model.

The requirements are prioritized with respect to the applicative dynamics imposed by the "Customer", but they integrate, during the collaboration with the technicians, the non-functional constraints. One last point to note about Agile: the non-redundancy of information in the documentation within a single space and between different spaces. Within the framework of consequent projects, in order not to complicate them unreasonably, these documents can refer to specialized appendices.

Thus, structure simplification and purity make it possible for the actors to position themselves in order to understand their roles and their responsibilities in the performance of a project and their implication in the quality of an application.

« Turn-Key » Components

More generally speaking, obtaining an Agile application is based on a research model for "turn-key" components (model in 7 layers corresponding to layers in the production cycle. Described in the book "Système d’Information et Processus Agile"). At times, generic solutions are non-existent (new requirements, new technologies) or unsuited to the context (competing differentiation). In these cases, a specific development is justified, and four aspects will contribute to the performance of the project and the efficiency of the emerging solution:

- Flexibility of application architecture, and particularly ease of redeployment which results in privileging service oriented techniques.
- The relevance, the quality and the reliability of the solution, which generally result from industrialized Agile development practices (Frameworks, Design patterns).
- Control of the deadlines, costs and other constraints of projects to which the PUMA piloting techniques offer elegant and tested solutions.
- Agility and integration of tools and humans in a concept of "application factory".
The Agile company requires Agile architecture, Agile techniques, Agile methods and Agile tools.

Extreme Agility In Development Quality

In the implementation of the PUMA framework, flexibility must be regarded as a faculty to adapt to evolution and rigour must be regarded as the capacity to respect an orchestration of formally defined good practices.

Without flexibility, rigour leads to bureaucracy and, in a rapidly evolving world, expensive and slow practices lead to failure. In the same way, flexibility without rigour leads to chaos. The implementation of the PUMA framework corresponds to a research of balance between rigour and flexibility, which leaves place for: creativity, adaptation to unforeseen elements and to continuous improvement; all this while equipping the project with a set of clearly defined guidelines that drive the engagement of human resources and support excellence of realization. It is according to this philosophy that it is necessary to understand, interpret and use the PUMA framework.

With regard to more specialized practices of Agility, related to the piloting of a project, to communication or development, PUMA is based on the standards of the Agile Alliance (Process, Modeling, Programming, etc) and operationally federates the whole set of these concepts.

The agile approaches that inspired PUMA (particularly eXtreme Programming), were the subject of many communications over the Internet so their specificities will not be detailed here. On the other hand, a recall of the main principles could favorably set into context the PUMA project control aspect.

In Agile mode, a development hoping to associate performance and quality, induces a simple phasing associated with iterative project leadership. The need for a strong coupling between a form of modeling such as UML and a cycle of incremental development is also highlighted. Communications and work methods are collaborative and highly facilitate the rational re-evaluation of objectives and priorities. Team work is naturally reactive to change, economical of means and in continuous search for effectiveness. During the Construction of the solution, systematic and "permanent validation" practices naturally avoid the materialization of the traditional "project risks".

In short, the processes of communication, decision and production are in perfect synergy and at the optimal level of granularity.

The Agile Solution Space

Foundations of Agile Development

The study of the main Agile methods shows them to be similar in their foundations:

- Respect of urbanization (the positioning of the project within the information system).
- Piloting (resource management, planning, follow-up, quality, reporting, visibility).
- Application engineering (requirements management, design and development, validation of deliverable items).
- Change conduct (organisational impacts and deployment).
Figure 7. The PUMA iterative development model

Agile methods are equipped with a set of common practices. Only techniques that are complementary to one another or better adapted to specific project typologies and sizes differentiate them. PUMA indexes the common and differentiating practices and proposes the optimal method according to a type of project.

PUMA project leading uses either integrally or selectively the set of common practices for which it is advisable to add other context specific judicious practices. All these aspects must fit within a variable level of methodological service.

Some practices are related to collaboration modes:
- end-user participation in workshops,
- workshops with decision-making authority,
- team autonomy and centralized organization (motivation),
- requirements specification and permanent validation.

Other practices are specific to **project piloting**:
- methodological level varying according to project stakes,
- piloting according to stakes and risks,
- global strategic planning based on rapid iterations,
- development milestones based on iterative and incremental active prototyping,
- continuous search for improving the practices.

Lastly, specific techniques governing **code quality**:
- search for technical design and excellence design,
- graphic view of a necessary and sufficient modeling,
- view of necessary and sufficient documentation,
- standards and reasonable code quality techniques (metrics),
- component based architecture,
- automated change management.
Note: For those who would not want to practice Agile techniques: an Agile model is not a representation of reality as coherent or relevant as to all intents. An Agile model IS reality under development. As an example, for eXtreme programming, the model is the code. For Agile Modeling, the model of classes is already the database (within one execution of a generation script). AND, generally speaking, the processing model is a prototype validated functionally by frequent "releases" and technically by a daily "build" (zero-day milestone).

The eXtreme Programming, well rounded method for small projects, the “flashier” approach to Agile development, corresponds overall to the PUMA Construction phase. Its originality lays in the fact that it pushes to the extreme all software engineering production techniques, which complicates its adoption by developers and its generalization over all project types.

The XP planning approach materializes in the form of a planning game which simultaneously involves users and developers. XP recommends also more particular techniques such as pair programming, collective ownership, refactoring and continuous integration. According to project context, PUMA uses the whole or part of these techniques but generally restricts pair programming to the strategic or more complex areas of the application.

The PUMA Project Engine

The PUMA Project Engine includes: an initiation module + a basic platform + 2 extension levels according to the importance of the projects.

![Diagram of Puma-project](image)

**Figure 8. Basic version of Puma-project**

At the first level of control (figure 8) which is essentially "design and development"-oriented, the basic version of the Puma-Project corresponds to a base set of XP (eXtreme Programming) best practices, but includes an improved and "solution"-oriented implementation (rather than pure isolated techniques).

At the second level of control (figure 9), the basic version is enriched by wider project piloting techniques. They are essentially Agile derived traditional practices.
At the third level of control (figure 10), PUMA proposes solutions specific to recurring organizational problems encountered in large projects. Thus, in certain meetings, it will be suggested that actors may specialize their "roles": executive sponsors, ambassadors, visionary users, advising users, without leaving out the organizer-facilitator and the reporters.

PUMA also recommends that the dimension and maturity of the working groups vary according to the phases. They will operate in a work and communications workspace featuring adequate materials. Systematic practices of short daily meetings will aim to improve the motivation of the participants, to synchronize the tasks, to resolve difficult situations and to increase knowledge sharing. Certain controls will impose on the project a strong orientation towards a measurable and immediate goal. In fact, it is the ambition of an iteration which will be thus reinforced. This last point will be resolved in the form of a "Focus" meeting centered on progress visibility and global quality control of the application.

Other practices give priority to functionalities that carry an added value and allow the management of delivery dependencies per reduced functionalities or per topics requiring specialized planning modes and development techniques.
A Globalist Paradigm

Agility, paradigm of a new vision of the organization, imposes itself as a tool for alignment and coherence between the internal forces and the external challenges which constitute the dynamics of the company. To conclude, we present what could be an "Agile Enterprise Manifesto" and a theory of Agile Quality.

Agile Enterprise Manifesto

Organisational agility rises from a shared vision of the mission to accomplish, resting on strong values that are in coherence with real practices:

- With regard to process, the philosophy an Agile company is "The one best way": processes are continuously modelled, simplified, reconfigured.
- With regard to human resources, the principle of the Agile company is "Empowerment": the autonomy of working groups ensures the global regulation of micro-changes.
- With regard to automation, the philosophy of the Agile company is "High-Tech, High-Touch": the rational use of emergent technologies is systematically anticipated.

The PUMA Agile Enterprise Framework

The pragmatic North American ideal describes as "the one best way" the optimal solution applied to the resolution of a problem. Generally this way proves to be as powerful as it is elegant and represents at the same time a model of thought and action. This philosophy of pragmatism has guided the work that initiated PUMA.

PUMA is a dynamic framework of evolution of business processes, information systems and collaborative modes. Based on the Agile movement and the technical standards that it integrates and federates, PUMA represents the first Agile formalization of a global Enterprise model, coupled with a project engine via a solutions model adapted to the new order of current requirements classes.

By modeling the generic high-level structures applicable to all organizations, the ambition of PUMA is to reduce the complexity of their management. If a professional who faces the PUMA solution thinks “that’s immediately obvious. I had it in mind already but I never had the time to formalize it,” then the challenge of Agility will be taken.

For A Free Method

On the matter of method usage, the author Jean-Pierre Vickoff as well as TEAMLOG, the company he collaborates with for the usage of PUMA, wishes to distribute the approach to all professionals through periodic publications about its structure and practices, via the main professional organizations, the media and on the Web: www.Entreprise-Agile.com
Synthesis of the 6 Action Models

**MA-1 - Business Projection, Resources and Supporting Technologies Model**
This model has the role of building the future system of operation. It integrates trend information and the divergences arising from the "Technical and Functional Anticipation" and the "Process Monitoring and Continuous Optimization" models, in order to structure the future evolution of the company in terms of **convergence trajectories** applying to:
- operational processes,
- human resources competencies
- technological architecture and information system architecture.

*It is within this domain that company projects are situated.*

**MA-2 - Technical and Functional Anticipation Model**
The main roles of this domain are:
- to anticipate which emergent customer requirements or technologies are likely, in an immediate future, to have an impact on the company;
- to feed this information to the "Business Projection, Resources and Supporting Technologies Model" domain.

**MA-3 - Process Monitoring and Continuous Optimization Model**
This domain is composed of the organizational structures and the means allowing it:
- to measure performance components,
- to detect possible divergences between processes and the reality of operations.

These organizational structures have two principal missions: to improve or adjust the processes and to feed this information to the "Business Projection, Resources and Supporting Technologies Model" domain.

**MA-4 - Adaptation of Competencies and Collaboration Types**
This domain covers the implementation of an agility framework in terms of training and communication, offering the following to Human Resources:
- improvement of their competencies,
- the conditions for their motivation,
- the appropriation of a collective intelligence.

This framework leads to the conditions required the "Process Monitoring and Continuous Optimization Model" domain objective.

**MA-5 - Information Systems and Technological Systems Model**
This domain covers the operation and control (standardized ITIL, Cobit, and other models) of information and technological systems (including, as usual, new technologies for information processing and communication (NTIC), industrial data processing and the use of other forms of process automation).

**MA-6 - Processes Operation Model**
This domain covers primarily the rules applicable to operations composing current processes and to the structure responsible for their execution. **These processes can be supported by automated systems ("Information Systems and Technological Systems Model" domain) or they can be manually operated.**
### Synthesis of the 6 Interaction Models

**MI-1** - Feedback on all of the functional or technical incidents noted at the time of process operation. This flow is directed towards the resources responsible for the corrective maintenance of data processing and information systems for immediate attention. The instrumentation is done through a light "anomaly correction" type of work-flow.

**MI-2** – Feedback on functional divergences emerging between envisaged conditions and the real operating conditions of the process, including volumetric evolution. This flow is directed towards the resources responsible for anticipating the evolution of the data processing and information systems. Instrumentation recommended in the collaborative space dedicated to rational Anticipation piloting.

**MI-3** – Transfer via the rational Anticipation module of the information that is validated as representing a technical and functional tendency to be taken into account during the development of future ISs. Support for this flow can be implemented as a simple email warning referring to information made available on the collaborative space dedicated to rational Anticipation piloting.

**MI-4** – Change conduct and transfer upon materialization of the information concerning operational competencies related to the installation of the new production system. This flow is materialized as « job descriptions ».

**MI-5** – Upon Framing the architecture and operational constraints, transfer of this information in order to initiate the information processing systems update. Transferring of the applicative solution elements for integration tests at each release (Focus). This flow is materialized by the technical aspect in the Framing stage.

**MI-6** – Technical transition and installation of the new process version.

**MI-7** – Transitioning of organizational directives related to the new process operation. This flow, emerging from a training program, is materialized by instruction change.

**MI-8** – Real time feedback of incidents and operational divergences towards the continuous process optimization module in order to formalize an immediate corrective measure, then to initialize a procedure of corrective or evolutionary maintenance. This flow is materialized by a documentary "Test" work-flow and/or a system of automated monitoring.

**MI-9** – Directives for the immediate correction of the manual part of the process if it is the case, or for temporary alternatives around the automated process. This flow is materialized by a corrective measure on operation instructions and possibly leads to the allocation of new resources.
Table des illustrations

Figure 1. Agile Enterprise Model (PUMA Architecture) ........................................................................... 4
Figure 2. Agile Solution Model (Business Projection) ............................................................................... 5
Figure 3. Specialization of the Solution Model ........................................................................................ 6
Figure 4. Global vision of Agile concerns ................................................................................................ 6
Figure 5. PUMA-BPM ............................................................................................................................. 8
Figure 6. Identification and Granularity of SOA Services .......................................................................... 8
Figure 7. The PUMA iterative development model .................................................................................... 12
Figure 8. Basic version of Puma-project .................................................................................................... 13
Figure 9. PUMA project piloting extensions ............................................................................................. 14
Figure 10. PUMA specialized extensions for consequent projects ......................................................... 14

Concerning the use of the PUMA method, its author Jean-Pierre Vickoff as well as TEAMLOG, the company with
which he collaborates in creating TMF (Teamlog Methodology Framework), wish to render the theoretical
principles accessible to the entirety of the profession by regularly publishing, via the media and the Web, on its
structure and its practices by the intermediary of professional organizational principles www.Entreprise-
Agile.com

© Jean-Pierre Vickoff. The text and graphics in this document have been registered and copyrighted
by the author Jean-Pierre Vickoff, as well as by the medias who initially distributed them. You may
copy and distribute, without alterations, all or part of this document in any format, commercially or non-
commercially, on the condition that this licence and copyright notice are reproduced on each copy, and
that you do not add any additional restrictions. You may also, while respecting the author's ownership,
quote brief passages.