INTRODUCTION

Operating in highly turbulent environments, organizations today are faced with the need to continually adjust their infrastructure and strategies in order to remain competitive. Globalization and continual technological evolution are the main drivers of this turbulence (Dove, 1999b). To adapt at the same pace as their changing environment, organizations have to be agile. Loosely defined, an agile enterprise is one that is characterized by change proficiency. Change proficiency is the defining characteristic of agility and denotes the competency in which an adaptive transformation occurs (Dove, Benson, & Hartman, 1996). In a more detailed definition, an agile enterprise is one that is characterized as a fast moving, adaptable, and robust business, which is capable of rapid adaptation in response to unexpected and unpredicted changes and events, market opportunities, and customer requirements (Henbury, 1996).

According to Dove (1999b), agility is very much related to the ability to manage and apply knowledge effectively. Dove (1999b) felicitously associates agility with cats. A cat is both physically adept at movement and also mentally adept at choosing useful movement appropriate for the situation. If a cat has merely the ability to move quickly but moves inappropriately and to no gain (e.g., a cat on a hot tin roof), it might be called spastic or confused but never agile. On the other hand, a cat that knows what should be done but finds itself unable to move (e.g., a cat that’s got itself up a tree), might be called catatonic, confused, or paralyzed but never agile.

This example implies that agility cannot be easily attained. It requires knowledge, experience, and skill. Enterprise agility depends on many factors such as personnel capabilities, information technology (IT) infrastructure, business strategy, and so forth. When an enterprise is agile, all its constituents are agile and vice versa. This article focuses particularly on IT infrastructure. It defines agility in IT infrastructure and explains how it contributes to enterprise sensing and response agility. Sensing agility is defined as a firm’s ability to rapidly discover and interpret the market opportunities through its information systems, and it concerns not only an ability to distinguish information from noise quickly, but also to transform apparent noise into meaning faster (Haeckel, 1999). Response agility relates to the organizational capability to quickly transform knowledge into action in response to the environmental signals (Haeckel, 1999).

BACKGROUND

The term agility has over a decade of use in manufacturing practices, where it has been defined as a principle competitive issue (Kidd, 1994; Dove, 1994a; Goldman, Roger, & Kenneth, 1995). Dove (1999a, 2005) has introduced the principles for agile systems at an abstract level so that they can be interpreted either from a business or a technical perspective. The term system is used to characterize a group of interacting modules sharing a common framework and serving a common purpose. At the business level, modules represent groups of people while at the technical level correspond to software components or machines. These principles are summarized in Table 1.

Dove (1995) has also defined four agility metrics, namely time, cost, robustness, and scope. The first concerns the time required to complete a transformation. The second defines the cost regarding the transformation implementation. Robustness measures the strength and quality of the change process. Scope indicates how much latitude for change can be accommodated. Kidd (1994) has additionally defined a fifth agility metric which is the frequency of change.

The concept of agility has also been employed in the research area of information systems (IS) development where the term is much more recent (Aydin, Harmsen, Slooten, & Stegwee, 2004, Levine, 2005). Agile IS development concerns a new methodology paradigm proposed as an alternative to traditional disciplined methodologies.
for software development because these methodologies are no longer successful for rapidly changing environments due to their bureaucratic nature (Conboy & Fitzgerald, 2004; Nerur, Mahapatra, & Mangalaraj, 2005). Agile development methodologies (Abrahamsson, Salo, Ronkainen, & Warsta, 2002), such as Extreme Programming (Beck, 1999) and SCRUM (Schwaber & Beedle, 2002), promise faster development times and higher customer satisfaction. Extreme Programming and SCRUM constitute instantiations of the Agile Manifesto (Fowler & Highsmith, 2001), which was published by the Agile Alliance in 2001 (www.agilealliance.com). The basic principles of the Agile Manifesto are: first, individuals and interactions over processes and tools; second, working software over comprehensive documentation; third, customer collaboration over contract negotiation; and fourth,

<table>
<thead>
<tr>
<th>Table 1. Agile design principles (Source: Dove, 1999a)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Encapsulated Unit Modularity</strong> System of interacting unit not intimately integrated. Internal workings unknown externally.</td>
</tr>
<tr>
<td><strong>Plug Compatibility</strong> System units share common interaction and interface standards, and are easily inserted or removed.</td>
</tr>
<tr>
<td><strong>Facilitated Unit Reusability</strong> Standardized unit replication information, unit modification tools, unit capability catalogs.</td>
</tr>
<tr>
<td><strong>Non-Hierarchical Interaction</strong> Empowered self-directed units that communicate negotiate and interact directly among themselves.</td>
</tr>
<tr>
<td><strong>Dynamic Late Binding Relationships</strong> Relationships are transient when possible; fixed binding is postponed until immediately necessary.</td>
</tr>
<tr>
<td><strong>Distributed Control &amp; Information</strong> Units respond to objectives; decisions made at point of knowledge; data retained locally but accessible globally.</td>
</tr>
<tr>
<td><strong>Self-Organizing Relationships</strong> Dynamic unit alliances and scheduling; open bidding; and other self-adapting behaviors.</td>
</tr>
<tr>
<td><strong>Scalable Size</strong> Unrestricted unit populations that permit large increases and decreases in total unit population.</td>
</tr>
<tr>
<td><strong>Unit Redundancy</strong> Duplicate unit types or capabilities to provide capacity fluctuation options and fault tolerance.</td>
</tr>
</tbody>
</table>
AGILE IT INFRASTRUCTURES

IT infrastructure plays a critical role in an organization’s competitive advantage as it is the enabling foundation of shared information technology capabilities upon which business depends (McKay & Brockway, 1989). IT infrastructure includes the hardware, operating systems, software applications, data, and the underlying network required to support business operations, as well as the enterprise personnel that interacts with it. Consequently, it comprises two interrelated but distinct components: (a) technical IT infrastructure and (b) human IT infrastructure (Broadbent & Weill, 1997). Enterprise agility requires both components to be agile. The following paragraphs explain the meaning of agility in human and technical IT infrastructure.

Agility in Human IT Infrastructure

Agility in human IT infrastructure implies that the relevant employees have adopted the perspective that they work in an unstable environment where any change may occur at anytime and that they have the skills, knowledge, and expertise to cope with change efficiently. To be specific, managers should be able to make the right decisions at the right time, while other employees, such as software engineers and business process owners, should have the knowledge to immediately configure or implement a solution. According to Lee, Trauth and Farwell (1995), human IT infrastructure needs four types of knowledge and skills: technology management, business functional, interpersonal, and technical. Technology management knowledge and skills concern an understanding of where and how to deploy IT effectively and profitably by achieving the strategic goals of an enterprise. Business functional knowledge and skills refer to the ability to understand business problems and develop the required technical solutions. Interpersonal and management knowledge and skills involve abilities such as planning, organizing, teaching, and leading. Lastly, technical knowledge and skills include abilities in technical areas, such as computer operating systems, telecommunications, application specific software, and so forth.

Agility in Technical IT Infrastructure

Agile IT infrastructure is one that has been developed according to an agile IT architecture. IT architecture is a framework which forms a guide for an IT infrastructure implementation by indicating technology components in an integrated view that shows how they collaborate to deliver business or technical services. Earl (1989) suggests a typical IT architecture has blueprints for the computing, data, communications, and the application systems of the organization. When an IT architecture is not developed down to a detailed technical level and forms a more abstract view instead, it is called conceptual architecture.

In order to be deemed agile, a technical IT architecture should offer a high degree of automation, integration, and flexibility (Alexopoulou, Kanellis, & Martakos, 2004). A conceptual agile IT architecture is illustrated in Figure 1. This architecture ensures high automation because it is based on executable business processes. An executable business process is a kind of enterprise process, whose life cycle is controlled by a business process engine (Nickull et al., 2001). Executable business processes are described in XML-based business process languages such as Business Process Modeling Language (BPMI, 2001), XLANG of Microsoft (Thatte, 2001) and Business Process Execution Language (Thatte et al., 2003), which are machine-readable.

As shown in Figure 1, the Business Process Engine (BPE) constitutes the heart of the architecture since it interacts through the exchange of messages with: (a) users via a document-based Worklist Browser, (b) customers via a Web Browser, (c) trading partners via the B2B engine, and (d) applications and components via the component management service (CMS). The BPE reads and executes business logic defined in process definition documents and acts as a coordinator of activities spanning across the enterprise entities, invoking for each activity the entity that is responsible for performing it. Whenever messages sent by the BPE need to be
transformed into another format, a transformation mechanism is used. For example, if a message is to be directed to a worklist browser, it must be first transformed into HTML. Likewise, at the application component level, if for example CORBA (Common Object Request Broker Architecture) is used, then the messages sent by the BPE will have to be transformed into CORBA IDL messages. Overall, the B2B engine will have to transform them onto the format required by the protocol used in the specific business collaboration, since the B2B engine is able to support various B2B protocols (Dabous, Rhabi, Ray, & Benatallah, 2003).

The CMS finds and invokes the appropriate application components that deliver the requested business service. These components are called business aware components, while the components implementing the fundamental infrastructure services are called framework service components (Raymer, Afrin, & Trivedi, 2001). The components can intercommunicate over a common communication infrastructure. Legacy applications can be connected to the communication infrastructure via adapters. The CMS together with this infrastructure constitute an enterprise application integration (EAI) (Puschmann & Alt, 2004) which follows some of the principles of the NGOSS (New Generation Operations System and Software) framework (Raymer, Afrin, & Trivedi, 2001). NGOSS is an initiative of the TeleManagement Forum set to develop a framework for rapid and flexible integration of operations and business support systems in telecommunications, but it can be equally applied to
Agile Information Technology Infrastructures

Table 2. Integration and flexibility capabilities of agile IT infrastructures (Source: Alexopoulou et al., 2004)

<table>
<thead>
<tr>
<th>Integration</th>
<th>Flexibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Processes</td>
<td>Support for business processes that span multiple applications regardless of whether these applications belong to a single or to different companies.</td>
</tr>
<tr>
<td></td>
<td>A business process definition can be altered without requiring modification of the application components.</td>
</tr>
<tr>
<td>Data</td>
<td>Data reside in any data source anywhere and can be used by any application or system anywhere.</td>
</tr>
<tr>
<td></td>
<td>Data can be easily transformed from one format to another at run time.</td>
</tr>
<tr>
<td>Application Components</td>
<td>Components can communicate efficiently with each other as well as with legacy applications.</td>
</tr>
<tr>
<td></td>
<td>New components can be easily embodied into the existing architecture and also components can be re-used across multiple business scenarios.</td>
</tr>
</tbody>
</table>

other business areas as well. NGOSS defines a service-oriented modular system framework, which is based on a collection of loosely coupled, re-usable components that perform business services. Two very good candidates for the implementation of NGOSS are Web Services (Fremantle, Weerawarana, & Khalaf, 2002) and the Java 2 Platform Enterprise Edition (J2EE) architecture (Sun Microsystems, 2003). Web services are loosely-coupled, re-usable software components that semantically encapsulate discrete functionality and are distributed and semantically accessible over standard Internet protocols (Stencil Group, 2001). J2EE is a set of specifications for developing multi-tier enterprise applications in Java.

The integration and flexibility capabilities of this architecture are described at three different levels, namely application components, data, and business process level, in Table 2.

**HOW AGILE IT INFRASTRUCTURES CONTRIBUTE TO ENTERPRISE AGILITY**

The sensing agility of an enterprise is facilitated by agility in human IT infrastructure, since timely discerning an imminent need for change depends on the knowledge, experience, and skills like perspicacity of the relative enterprise personnel. However, the acquirement of the necessary knowledge is significantly facilitated by an agile technical IT infrastructure as the latter ensures the availability of the right information at the right time by enabling a seamless flow of information.

As far as response enterprise agility is concerned, it is equally facilitated by agility in both human and technical IT infrastructure. Human agility implies that the right decisions are timely made in response to a change. On the other hand, an agile technical IT infrastructure provides for a rapid and cost-effective implementation of the decided-upon solutions. Following, we will further elaborate on the way the agile IT architecture described earlier accommodates change.

According to Dove (1994b), an agile enterprise can employ business process reengineering as a core competency when a transformation need arises. In the described architecture, abstraction of the business process flow into an entity (BPE) separate from the application components themselves allows an easier and more flexible way to alter the business process logic whenever new circumstances arise or a modification is needed. The only action required in such a case is an update in the business process definition that is executed by the BPE, while no modification is needed at the application component level. Separating process control removes the need for the
individual components to have knowledge of the business logic associated with process operation. When invoked by process control, a component simply performs the service offered through its interface. Also, components can be re-used across multiple business scenarios.

To achieve even greater flexibility at runtime, business process definition could follow the methodology described by ShuiGuang, Zhen, ZhaoHui and LiCan (2004). According to this methodology, a business process is composed of general activities, which are predefined in detail at design time, and flexible activities, which are like a “black box”, representing an undetermined sub-process without detailed specification at build-time. In other words, flexible activities encapsulate the uncertain sub-process at run time. At run time, depending on current circumstances, a flexible activity can be replaced by a concrete sub-process composed of selected activities from existing or newly-added activities (constituting a pool of activities) based on selection and composition constraints.

However, this method implies that activities included in the aforementioned pool correspond to predictable situations, but what about the case of a situation that has not been predicted? Agility, as we mentioned earlier concerns also efficient response to unexpected change. In case of an unexpected change, the required services to accommodate it may not be offered by the existing IT infrastructure simply because the relevant application components may not exist. In such a case, the most important role is played by the human IT infrastructure because it has to be decided whether the required services will be outsourced or developed in-house or acquired through commercial, off-the-shelf products. As agility is a function of both cost and time (Dove, 1996 REFERENCES), it cannot be claimed that a specific solution will be a panacea to all situations.

From the perspective of an agile technical IT infrastructure, the only contribution that can be offered is the facilitation of an easy and rapid incorporation of the new application into the existing infrastructure; either this application has been developed in-house, purchased, or outsourced. The aforementioned IT architecture supports plug and play components, which means that new components can be easily embodied into the infrastructure and communicate with the already existing applications via the common communication vehicle.

FUTURE TRENDS

As stated earlier, software implementation may be inevitable in case new services are required upon an environmental change. However, as programming is a difficult and time-consuming task, developing models instead of code would significantly augment enterprise agility. Toward this direction, several initiatives have already emerged. Borland is developing a software product called Themis, which will have a module that will turn models automatically into programming code (The Economist, 2004). OMG has proposed the Model Driven Architecture (MDA) (www.omg.org/mda) initiative that addresses the problem of integration and interoperability by making UML models and modeling artifacts more executable. In other words, MDA is about using modeling languages as programming languages rather than merely as design languages. Likewise, Web Modeling Language (WebML) (www.webml.org), which is a notation for specifying complex Web sites at a conceptual level, enables a model-driven approach to Web site development. It is expected that this shift of emphasis on models and their automatic translation to source and object code underlines the trends for both research and practice in this area.

CONCLUSION

The pace of business change drives organizations to respond as quickly as possible. In order to cope with continually changing environments they have to be agile. IT infrastructure facilitates and enables agility within an organization. From a technical perspective, an agile IT infrastructure requires a high degree of automation, integration and flexibility. The fact that this paper focuses mainly on technical IT infrastructure does not imply that the human part has less impact on enterprise agility. In fact, human IT infrastructure also plays a critical role in an organization’s ability to sense environmental change and respond efficiently and effectively to that change. Even if a “super agile” technical IT infrastructure has been developed, it will be of no value at all if the relative personnel are unable to understand and adjust to new circumstances. Humans are key players; they discern environmental changes and make the appropriate decisions to cope with these changes. Of course, they must always be supported by an agile technical IT infrastructure. Otherwise stagnation will follow with the enterprise being unable to implement its strategies and hence ensure its continuous profitability and survival.

REFERENCES


---

**KEY TERMS**

**Agile Enterprise:** A fast moving, adaptable and robust business, which is capable of rapid adaptation in response to unexpected and unpredicted changes and events, market opportunities and customer requirements.

**Agile IT Infrastructure:** A highly automated, integrated and flexible IT infrastructure which enables efficient and effective response to planned, as well as unanticipated change.

**Agility:** Efficient and effective response to planned as well as unanticipated change.

**Executable Business Process:** A kind of enterprise business process, whose life cycle is controlled by a Business Process Engine.

**IT Architecture:** A framework which forms a guide for an IT infrastructure implementation by indicating technology components in an integrated view that shows how they collaborate to deliver business or technical services.

**IT Infrastructure:** The enabling foundation of shared information technology capabilities upon which business depends.

**Response Agility:** The organizational capability to quickly transform knowledge into action in response to the environmental signals.

**Sensing Agility:** A firm’s ability to rapidly discover and interpret the market opportunities through its information systems.

**System:** A group of interacting modules sharing a common framework and serving a common purpose.