

The International JOURNAL

of

KNOWLEDGE, CULTURE
& CHANGE MANAGEMENT

Volume 9, Number 5

Designing for Agility as an Organizational Capability: Learning from a Software Development Firm

James Sena, Jean-Francois Coget and A.B. (Rami) Shani

THE INTERNATIONAL JOURNAL OF KNOWLEDGE, CULTURE AND CHANGE
MANAGEMENT

<http://www.Management-Journal.com>

First published in 2009 in Melbourne, Australia by Common Ground Publishing Pty Ltd
www.CommonGroundPublishing.com.

© 2009 (individual papers), the author(s)

© 2009 (selection and editorial matter) Common Ground

Authors are responsible for the accuracy of citations, quotations, diagrams, tables and maps.

All rights reserved. Apart from fair use for the purposes of study, research, criticism or review as permitted under the Copyright Act (Australia), no part of this work may be reproduced without written permission from the publisher. For permissions and other inquiries, please contact

[<cg-support@commongroundpublishing.com>](mailto:cg-support@commongroundpublishing.com).

ISSN: 1447-9524

Publisher Site: <http://www.Management-Journal.com>

THE INTERNATIONAL JOURNAL OF KNOWLEDGE, CULTURE AND CHANGE MANAGEMENT is peer-reviewed, supported by rigorous processes of criterion-referenced article ranking and qualitative commentary, ensuring that only intellectual work of the greatest substance and highest significance is published.

Typeset in Common Ground Markup Language using CGCreator multichannel typesetting system

<http://www.commongroundpublishing.com/software/>

Designing for Agility as an Organizational Capability: Learning from a Software Development Firm

James Sena, California Polytechnic State University, California, USA

Jean-Francois Coget, California Polytechnic State University,
California, USA

A.B. (Rami) Shani, California Polytechnic State University, California,
USA

Abstract: The purpose of this paper is to investigate the nature of agility in an organizational setting -- how a software development firm (SDF) developed, maintained and enhanced agility as it changed from a developer of experimental prototypes to a product-based provider. Qualitative approach based on extensive interviews and on-site observations at two points in time separated by a five-year interval. Agility mechanisms tend to be dynamic and evolve over time. At time 1, SDF achieved agility by adopting a formal platform-based product design and an informal, organic organizational structure. By time 2, SDF had adopted a modular-based product design and a more formal structure. Implications include: (1) interdisciplinary-based framework to understand agility in the workplace; (2) multiple forms of agility and the dynamics among them; (3) re-conceptualization of agility as a new organizational capability; and, (4) causal relationship between agility and other organizational learning mechanisms. Limitations include: (1) the tentative theory building (as opposed to theory testing) qualitative approach; and, (2) single case study within a specific industry. (Practical Implications: 1) By adopting agility mechanism software development firms may overcome strategic challenges in the software industry: extensive reworks, death marches, and client support services; (2) over time managers should explore alternative mechanisms to sustain agility; and (3) agility-by-design is likely to facilitate firm success and growth.

Keywords: Agility, Software Development Firm, Learning Mechanisms, Organizational Capability

A GILITY HAS BEEN a part of the management literature for some time. An *agile mind* is defined by Merriam-Webster online (2005) as having a quick, resourceful and adaptive character. So we can infer that *agile organizations* respond quickly; are resourceful; and are able to adapt to their environment. Table 1 summarizes several definitions given in the management literature. All the definitions provided in Table 1 point to a common thread: agility represents an organization's capability to cope with external and internal changes that are unpredictable and uncertain (van Oosterhout, et al, 2006). Unpredictability represents the difficulty to predict whether or when a given event will happen. Uncertainty represents the difficulty to predict what the effects will be if a given event happens and what the organization's response will be.

Table 1: Agility Frames of Reference

Frame	Definition	Reference
Agility	The ability to thrive in a competitive environment of continuous and unanticipated change and to respond quickly to rapidly changing, fragmenting global markets that are served by networked competitors with routine access to a worldwide production system and are driven by demand for high-quality, high-performance, low-cost, customer-configured products and service.	(Goldman et al., 1995)
	is primarily concerned with the ability of enterprises to cope with unexpected changes, to survive unprecedented threats from the business environment, and to take advantage of changes as opportunities.	Sharifi & Zhang, 2000
	The ability of an organization to thrive in a continuously changing, unpredictable business environment.	Dove,2001.
	depends on the ability of an organization to develop and exploit its inter and intra organizational capabilities to successfully compete in an uncertain and unpredictable business environment.	Hooper et al.,2001.
	Successful exploration of competitive bases (speed, flexibility, innovation pro-activity, quality, and profitability) through the integration of reconfigurable resources, and best practices in a knowledge-rich environment to provide customer-driven products and services in a fast-changing market environment.	Ramasesh et al., 2001
	The continual readiness of an entity to rapidly or inherently, proactively or reactively, embrace change, through high quality, simplistic, economical components and relationships with its environment.	Conboy & Fitzgerald, 2004, p. 37

Agile Organizations	respond quickly, are resourceful and able to adapt to their environment: external market stimuli – opportunities presented by customers; threats posed by competitors; and, change in demand .Minimize costs and time-scales of any change in terms of the initial outlay and subsequent operations.	Sharifi & Zang, 2001 Boden, 2004
Business Agility	Ability to swiftly and easily change businesses and business processes beyond the normal level of flexibility to effectively manage unpredictable external and internal changes	Van Oosterhout et al, 2006
Information Technology Agility	IT agility means continuous close coordination between business and IT people to respond effectively to constantly changing situations. Agility is needed if customers value a product or service primarily because it quickly responds to their evolving needs	Hugos, 2007
Strategic agility	Relates to the ability to respond efficiently and effectively in emerging market opportunities by taking advantage of existing software development capabilities. The efficiency of response is primarily defined in terms of time. Its effectiveness can be defined in terms of alignment with organizational goals and competitiveness enhancements.	Fink, 2007

Agility was initially viewed by organizations as similar to flexibility: just another property that could assist change and adaptation. Later agility was expanded to manufacturing as an alternative to traditional operations and production management costing systems or mass/lean production (Sharifi & Zang, 2001). Other studies focusing on organization capability, organizational learning, organizational learning mechanisms, and knowledge management further broadened agility into an organizing and design paradigm (Bonabeau et al, 2008; Dove, 1999; Ebrahimpur, 2001; Gartner 2004; Lyytinen & Rose, 2006; Shani & Docherty, 2003).

Dealing with change has always been an important issue in organizations. In areas where change is predictable and the needed response is largely predetermined, organizations need to be flexible. Volberda & Rutges (1999, p. 101) define flexibility as “the degree to which an organization has a variety of managerial capabilities, and the speed at which they can be activated, to raise the control capacity of management and improve the controllability of the organization”. Volberda (1997) distinguishes three types of flexibility: operational flexibility (referring to reactive routines to familiar changes based on structures or goals of the organization); structural flexibility (referring to the capacity of the management to adapt its decision and communication processes within a given structure by which this can happen; and, strategic flexibility (referring to capacity of the management to react in unstructured non-routine unfamiliar changes that have far-reaching consequences and needs quick response). Operational

and structural flexibility can be programmed into an organization's processes to envelop and extend strategic flexibility to a certain extent (Overby et al., 2005). But when an organization needs to adapt to changes that were not considered when organizational processes and systems were established agility is needed. In such situations, organizational responses must be more radical and innovative than in situations where simple flexibility would suffice.

An agile firm designs its organization, structures, processes and products to respond to changes dictated by the business environment. The speed with which the organization can respond to customer requests, market dynamics and emerging technology options is critical to its agility. This includes the time to "sense" relevant events; to interpret what is happening and to assess the consequences for the organization; to explore options and to decide which actions and responses to take (Mathiassen and Pries-Heje, 2006).

Just as there are many types of organizational designs, there are various ways to design and manage agility (Teece et al., 1997). Firms must continuously adapt their capabilities in order to maintain competitiveness (Overby et al., 2006). Agility can be viewed as a general organizing paradigm rather than just a way to organize manufacturing (Ebrahimpur, 2001). Agile organizations possess the ability to manage and apply knowledge existing within and outside the firm boundaries to respond to expected and unexpected changes to exploit new business opportunities.

Although agility has been a part of the management literature for some time the notion of agility applied to software development teams is fairly new. Agility from a software perspective arose from the literature on flexible and lean manufacturing (Borjesson et al, 2006; Dove, 2001; Kidd 1995) and has been adopted by organizations producing software using *agile programming methodologies* (Aoyama, 1998; Cockburn, 2001). Agile programming sub divides an application development project into small modularized pieces. Each piece, addressed one at a time in short time frames, adds to the application and represents a part of the functionality. The partial application is deployed with the expectation that the user can complete some portion of work with it, even if the application does not do everything defined in the system specifications.

In this paper we look at how a software development firm (SDF) catering to military clients developed and maintained its agility as it grew over a five-year interval. At time 1, the firm developed mechanisms to be agile while competing as an experimental software research firm for its military clients. SDF developed methodologies and techniques to protect and safeguard their development environment from encroachment by large research firms. At time 2, agility took on a different meaning as the firm moved into a production type operation. The firm needed to find ways to not only develop new software but also to support applications for its clients. We describe our methodology and then identify key mechanisms through which agility is accomplished in the software development industry, and the benefits that are associated with them.

Methodology

This paper reports a longitudinal field study that looks at the dynamics of a growing software development firm [SDF] at two points in time separated by a five-year interval. In earlier studies, we looked at how SDF's operations and structure aided new product development and knowledge management. In this study we use those findings (expressed as time 1: then) and revisit SDF to assess the firm's agility (expressed as time 2 □. The firm's environment,

its growth, and changing demographics, among other factors, set the stage for our examining agility. We looked at the product development process, firm's organizational structure, and organization of the software development teams as the main mechanisms through which agility is realized. An overview of the software development firm is presented and product development steps are explained. Product development steps include initiation, design, construction, and delivery – followed by reflections on development, and agility. The field research consisted of collection of archival material, frequent observation of the work environment, and in-depth interviews with senior management, product team leaders and developers over a three month period on a weekly basis – usually six hours a week.

The research methodology includes four phases at time 1 and time 2:

Phase 1: Exploratory semi-structured interviews with senior management, observation of senior management strategic meetings, and collection of archival data (Organizational Charts, brochures, product prospectuses).

Phase 2: In-depth structured interviews with a cross-segment of employees in product development and observation of team meetings, problem solving sessions, and technical discussions in situ.

Phase 3: In-depth semi-structured interviews with the technical and product managers of a representative product in development, and collection of archival data (Product specifications, schedules, work allocation charts, etc...).

Phase 4: In-depth structured team interviews with the team leaders of the product development.

During phases 1 and 2, the interviews had a twofold purpose: (1) to develop a basic understanding about the changing work and products—work processes, management practices, the interface among different product development projects, support groups and customers; and (2) to identify the work practices and design factors that enabled agility. During phases 3 and 4, the team interviews with the team leaders had a threefold purpose: (1) to learn about the sub-teams and their intra-dynamics; (2) to learn how sub-teams interact with each other; and (3) to learn how they collaborate and share information. The representative product development was tracked from negotiation to proposal acceptance, product development, and delivery.

An Overview of the SDF Company

SDF is in the business of building, implementing, and supporting agent-based “Cooperative Decision Making” tools for distributed problem solving. Application areas include: facilities management, transportation planning, military logistics and control, and engineering design. The firm started as a university-based research group in 1989 and became an independent firm in 1994. SDF's main competitive advantage was and remains the agent-based methodology expertise dealing with spatial problems for organizing engineering design considering space management, space constraints, and storage priorities from an architectural perspective. A series of software agents are used to abet human decision making. Collaborative agents are self-contained, intelligent, adaptive software capsules used as building blocks to construct complex software products. Using collaborating agents to develop software products provides the flexibility and range needed for continuous improvement of product design.

SDF experienced significant growth in volume of sales and size. Sales have more than tripled over the five year period. SDF has no debt, leases its facilities – keeping its asset holdings to a minimum. The growth in organization size from owner-manager-entrepreneur to professional management is a result of the need to run the business. The two owners are still involved with strategic management, but have delegated much of the fiscal, marketing and operational aspects to professional staff. Part of SDF's growth was generated by the emerging need to support operational products at military sites. SDF personnel have become responsible for directly operating its software and training military personnel to operate it throughout the world. A growing demand in specialized products has led SDF to change the way it develops new software, dedicating software development teams to specific projects. SDF has grown from a staff of about 45 full-time and 50 part-time employees to over 150 full-time and about 20 to 25 part-time employees.

The Organization of SDF

SDF is organized around the product team. Most departmental units operate with relative autonomy. Support for product work is provided by cooperative groups that are separate from the department structure. At time 1, as portrayed in figure 1, leadership of the product team was divided between a product leader and a technical leader, who shared decision-making for all operations. This dual leadership for new product development was intended to address problems associated with external and internal direction, such as miscommunication among different work groups or conflict management, and as check-and-balance control mechanisms. There was no clear separation of responsibility between the two leaders, who cooperated in a spirit of camaraderie. The product team was divided into software development specialists and information technology specialists, who also cooperated in a spirit of camaraderie.

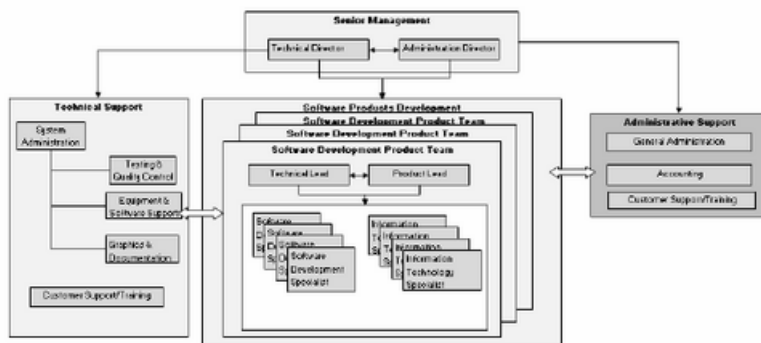


Figure 1: SDF Organization-Time 1

This changed at time 2, as shown in Figure 2. The technical leader solely manages the project development team, and coordinates with the product leader, who interacts with the client. This choice shows the technical complexity of the products under development. As the software that SDF produces became more complex, SDF decided to adopt agile programming techniques. This change called for the establishment of a strong technical leadership and support and the administrative leadership faded as a support function. The team members

are all considered as developers – each being capable of performing all software development product tasks. They are cross-trained in the specialties needed to create the software system – the Graphical User Interface [GUI]; data base modelling; and, the selection/designation of specific agents from the software toolkit. The following service groups support software development teams: system testing, customer support, and network administration. During the bid proposal phase, it is common for the development team targeted to be involved in proposal preparation. After the contract is awarded the technical and product leader dialogue with the development team to re-define the parameters of the deliverable product. Once all team members and the leaders have a common understanding of the product goals and customer deliverables the team begins the development process phase. The tasks are set to span up to a month, which constitutes a period of work. Together the technical and product leaders provide feedback and contribute to task designation.

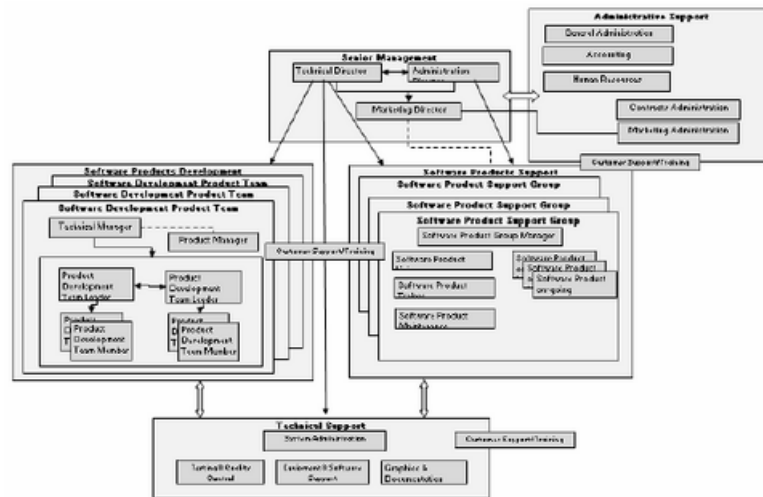


Figure 2: SDF Organization –Time 2

At time 1 although each team was devoted to a specific product development project, it was also usual for some of the team members to work on several product development projects. Task definitions needed to take into account what percentage of time the various members could allocate to the product development in the period of consideration. The team members provided the hours for which they were available. The available hours were matched to the task requirements under the direction of the two leaders. Priorities were set for the tasks. During each of these periods the specifications and tasks were reviewed with the customer. Given the customer feedback the tasks could change and the product content was further clarified. There was no grand design for the product. The team employed an *incremental approach* in which it maintained an extensive ongoing dialog with the client to anticipate problems before they became acute.

Disputes or differences had to be resolved through discussion or were brought to senior management for resolution. This was not a significant problem because the work content and work constituency were homogeneous, the size of the organization was small and the atmosphere was informal. New products involved technology transfer from existing products and adherence to grounded technologies using the spatial agent approach. Specifically, SDF

used artificial intelligence techniques that enabled it to acquire data and construct rules resembling Lego building. The creation various tool kits provided SDF with a dynamic way to grow and incorporate new knowledge and software capabilities. This was one of the key elements in SDF's agility at time 1.

At time 2 development teams and individual team members have become part of a cohesive team under the direction of the Technical Leader. Each team is now dedicated to the one product development and does not get involved with other products. Agile programming techniques in short time frames have been introduced. A portion of the developed module is deployed with the expectation that the user can work with it, even if the module application does not do everything defined in the product specification. Each piece can be developed as a mini-project in its own right that lasts from one to four weeks. As a result, developers can figure out quickly which piece of an application proves troublesome. Also, issues can be worked through as they occur rather than after the entire system is built. The Agile project manager thus oversees the planning, requirements, design, coding, testing and documentation stages on a feature by feature basis.

The key element in SDF's agile programming methodology is face-to-face communication, supported with written documents used as discussion points. In other words, rather than have each team member work on their own on various project pieces, everyone works on them as a team. Unlike other programming methods, agile programming relies on teams composed of highly differentiated members. A team includes project managers, designers, developers, testers, customers, and writers. Because a project piece is small enough for everyone to understand and for the stakeholders of the piece to work together, it is usually possible to complete it in with little or no rework.

The most important ingredient in agile development is that the development process includes the stakeholders. The customer/user is involved with the project from the outset, which means that the development team makes fewer wrong assumptions about how the user will interact with the application and about the needed steps to perform a task. This process differs markedly from the "write a spec, throw it over the wall, and then ignore it approach" that is common in many software development efforts. Instead, agility flows from all the stakeholders, especially the developers are aware of the big picture. Business stakeholders are co-located with small, autonomous development teams; the teams rely less on up-front requirements and documentation than on face-to-face conversations; those conversations provide a continuous dialogue for software design, testing and refocusing. The constant refocusing leads to more timely and useful business tools

The technical leader oversees the development process; reviewing task assignment and work flow, and overseeing the scheduling of testing and quality control to evaluate the finished product. The product leader manages the client. He interfaces with the technical leader but is not involved in the software development. The hardware, services and supplies, management and customer liaison, training and documentation are the responsibility of the entire development team in coordination with the service groups. The product leader serves as the coordinator, liaison and integrator with/for these groups. Everyone contributes to the documentation and training material.

The Product Development Process

At time 1 responsibility for specific product development elements were divided among the product leader, the technical leader, and various support groups. In figure 3 we show this division by grouping the various tasks on panels contained on the platform. The product leaders are shown in parallel panels. The support groups are depicted as blocks that overlap the product team responsibility areas/panels. The design and development phase depict the system requirement activities include the presentation and design activities, which are the responsibility of the technical leader. The technical leader has to draw from the programming team, which requires that he deal with a programmer pool that consists of part-time workers and a small cadre of experienced programmers. The estimating and scheduling activity is also jointly shared.

The choice to have dual product leaders – the product leader and the technical leader shows a “stocks of knowledge” philosophy. The technical leader provided coordination mechanisms to support software version control, libraries of shared and re-usable code, and the application of agent-based technologies. We could think of these as storage banks (sources of knowledge) on the platform. The technical leader oversaw the software team; made assignments and reviewed the work of the software developers; and, coordinated and scheduled testing and quality control. The product manager interfaced with the technical leader but was not involved in the software development. Instead the product manager handled the external interfaces with the customer and management. Knowledge about customer needs and expectations were tempered and translated to agent technologies using a standardized framework for work definition.

The hardware, services and supplies, management and customer liaison, training and documentation were the responsibility of the product manager in coordination with the service groups. When the product work was finished the product was delivered to the customer by the product leader and the technical leader. At time 1, SDF’s clients were military, with modular contracts spread over a series of years or periods. A pre-specified product was not always delivered. Instead products changed as features and capabilities were added. SDF marketed a core product that consisted of agents. This typical product was tailored to fit the needs of each customer by adding other agents. Most agents had been previously developed and were modified to meet special customer needs, or new agents were developed to meet unique needs of the customer. This agent-based technology was one of the main mechanisms of SDF’s agility, accounting for its success and viability.

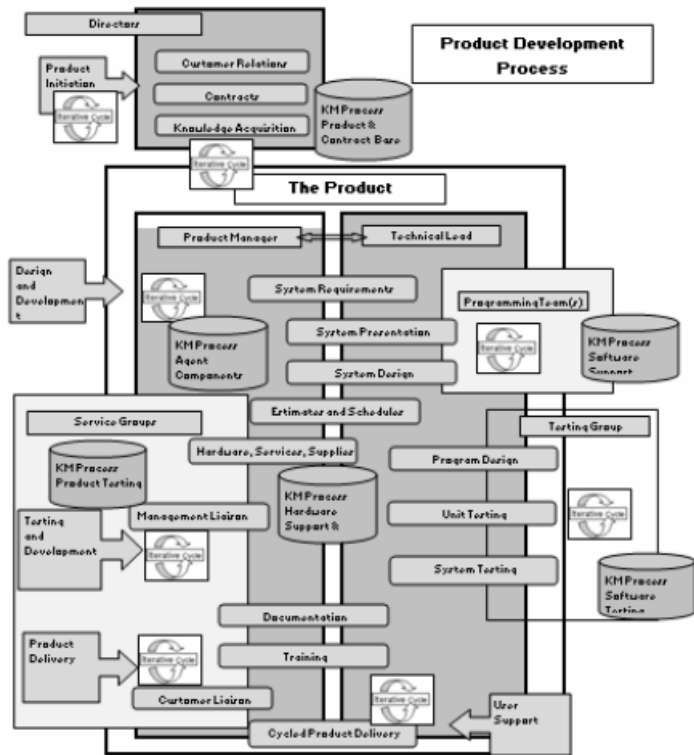


Figure 3: Product Development Time 1

At time 2, SDF has adopted a “modular platform-based architecture” philosophy for product development (Figure 4). Before the modular architecture process is deployed, was the product development phase is begun with the Request for Proposal (RFP) – this involves customer liaison and knowledge acquisition. As in most organizations of this nature the goal of this interactive process is the awarding of a contract. The technical leader is responsible for specific product development elements in concert with his development team. In figure 4, the support groups are depicted as blocks placed on the platform. The development team is the central part of the product development process. On the right side of the platform are the technical elements (modules) – including the product requirements defined in the product proposal; the system requirements; system design; and system presentation. On the left side are the external interface modules – including the estimates and schedule; the documentation and training; and the customer liaison. The design and development phase depict the system requirement activities as being jointly shared by the development team and the technical leader. The technical leader manages the members of the development team, which are mostly full-time experienced programmers, a significant change from time 1.

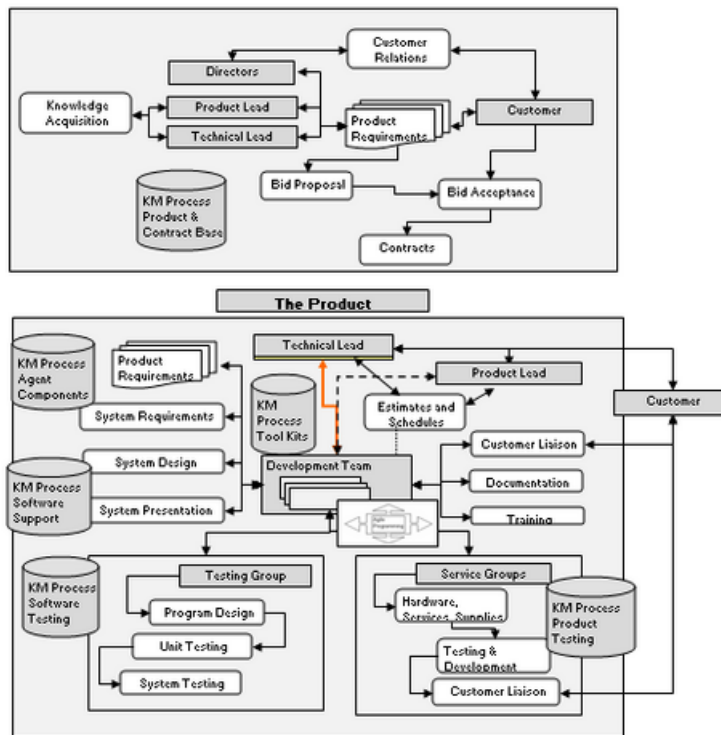


Figure 4: Process Development Time 2

Between time 1 and 2 SDF has worked mostly with military clients, with contracts spread over a series of years. SDF has frequently been involved in demonstration and experimental products. The military was interested in experimental work and viewed the SDF's products as prototypes. Often there was a fine line between prototype and functional systems. But there was a real business value in quickly getting a system into production that worked but covered only 80% of what was needed (Hugos, 2006a). Gradually, by adding features and capabilities to its prototypes, SDF created full-fledged, industry-standard products that allowed it to capture long-term contracts. To diversify and limit their dependence on the government sector SDF made some ventures into the commercial market with some success.

Product Initiation

Product initiation has not changed between times 1 and 2. The directors of the firm make primary contacts with military and private firms. Once an opportunity is identified dialogue is started with the agency or the firm. As the work scope takes form, the product and technical leaders are introduced. In some cases area experts or consultants help in the knowledge collection from the customer. This is a key aspect of SDF's modular approach. It allows a more stable product goal and scope definition.

Before a contract is negotiated and formalized time is set aside to develop a shared understanding of the product's purpose and scope. The customer needs to have a clear picture of

the product scope to ensure that there are no “surprises” as the product is being created. Every effort is made to involve the customer throughout the development and delivery processes. Constraints and deliverables are clearly specified. This is an important aspect of agility – developing a moving target can be counter productive (Simon, 2007). In many software product development efforts, especially government and military work, considerable time can be spent on re-work and re-negotiation to accommodate changes in the product scope over the product life cycle. This is not the case for SDF – leading to a most-favoured vendor designation by several military commands.

Product Design

The product design consists of identifying system requirements, deciding on a format for system presentation, and specifying the system design. The spatial agent technology employed relies on data acquisition by artificial intelligence techniques data definition and rule specification. Throughout the product development there needs to be an internal customer champion who maintains consistent, on-going contact with the product developers and the customer. At time 2, these roles are assumed by the product and technical leaders. SDF has created a mapping tool (IMT) to assist in the inclusion of heterogeneous data elements from a variety of sources and storage sources into a homogeneous database. The IMT instrument provides a methodology for clear, concise and consistent data definition for all products and a basis for common understanding by the development team and its customers. SDF’s data repositories of objects and data elements allows to take advantage of the similarities that exist across products in the military environment and to “do it right” the first time.

As a starting point the customer is shown and approves the front-end (Graphical User Interface) without unfolding the to-be-completed functionality. This functionality is supported by reusable software and hardware formulations to construct a collaborative data management product, called the ICDM Tool Kit. This tool kit is one of SDF’s primary competitive advantages because it insulates, insures and protects their intellectual property. Source code is not shared or provided to the customer. Products developed in most government and military work are usually available to any company for use. The wording in all SDF contracts explicitly *protects* the source code contained in their Tool Kits.

The product and the technical leaders compile the estimates and schedules for the product construction, with inputs from the development team. Arrangements are made for testing with the testing groups and documentation and training with the service groups. Another key element for SDF’s agility is the deployment of support teams as an overlay on the modular product structure – another reusable feature. Since SDF has major ongoing products with the military which have are completely functional systems these business units also provide a supporting function. This experience aids in the new product developments.

Between time 1 and 2, product design and implementation have changed. At time 2, product work is now divided into modules. Each module is developed for a period lasting from 2 weeks to a month. Modules involve the coding or some form of design using the tool kits and agents. The modules are first defined by the technical leader and refined by the development team members working on the various modules. Concrete deliverables are expected at the end of the period.

SDF’s agile development philosophy is intended to reduce or eliminate the frequent *death marches* experienced in the computer industry. Agile development relies on team empower-

ment and self direction, which in turns depends on the selection of team members who are willing to assume responsibility and to self-monitor their work. “Extreme programming”, employed at time 1, focused on short iterative development cycles with overreaching, long-term project goals. The agile approach adopted at time 2 takes the project a step at a time. The task elements are contained in their Task Tool Kit, an elaborate scheduling and task management template that constitute a “roadmap” for the development team to define their deliverables, distribute work, and the associate time frames for the forthcoming period. The kit features a variety of diagnostic controls – plans, budgets, goals, objectives and performance indicators – that support the task processes. The kit allows the leaders to manage by empowering their teams (Simons, 2007).

Product Construction and Delivery

At time 1, the product construction consisted of parallel operations. The technical leader and the product manager worked independently from the development team. The computer programs were written by the development team. *Reusable modules* formed the basis and foundation by allowing the team to take advantage of technology and expertise already tried and tested. This was another agility ingredient – they did not have to create a new product from scratch. The product definition and design meant that most development team members were interchangeable.

Each module went through a variety of tests – unit testing of each module and system testing of the modules together. Before a product was presented to the customer it was tested for internal reliability and adherence to specifications. While testing and programming was underway documentation was updated. Another support group produced professional quality manuals, online and web-based materials. When the software product reached a stage where the user needed to be involved (and since the documentation was to be presented to the customer) the documentation staff accompanied the product and technical leaders and key members of the development team to the customer site. While the programs were being developed, tested and documented the service group arranged for the purchase and configuration of the network and workstations or for the installation of the system on the customer’s hardware and network. After the equipment became operational the new software was installed by the technical leader and area specialists with support from the service groups. Here the on-site user testing and development was supported by the service group and the technical leader. Training was conducted simultaneously and enhanced documentation needs are noted.

At time 1, the software development team included developers, who defined the work scope, and programmers, who coded. At time 2, the distinction between developer and programmer disappeared. At time 2, a typical team is constituted by a team leader with twelve years of experience with the firm; two senior leaders with eight years of experience; two staff developers under each senior leader and two to three students. Nonetheless, the team functions as a flat hierarchy. Every team member is aware of the product’s development progress and all team members work for and with everyone else. The work flow is constant, with a steady pressure throughout a monthly cycle. The team defines the work for the period. The team is aware of the deadline for the project deliverables, and there are no “big surprises”. A team member can feel free to go to either leader at any time. Support from the leaders is given when solicited – the work is then reviewed and accurate feedback is given. Every week progress is a checked, work reallocated, and task assignments adjusted if needed.

The software tasks are divided into sub-tasks – every task is associated with a concrete deliverable – usually consisting of a combination of document, report and code. Most team members have one or more specialties, as mentioned earlier, such as database hanged, Graphic User Interface [GUI] developer, Artificial Intelligence [AI] Agent developer, or Test Data creator. Usually the team leaders possess domain knowledge (e.g. an understanding of the customer product, such as Docking System for the Trucking Industry.). Design comes from the bottom up – team leaders do not do the “grunt” work. The team operates in an egalitarian manner – no one owns the code. Two repository levels – one for Source Code (ICDM Toolkit) and the other for the Components – are used to support programming, and shared by all projects and teams. SDF does not believe in documenting the internal code thoroughly and rigidly. Instead access to the developer who built the code (resident experts) is considered more important. Formal documentation is mostly elaborated for the client.

SDF is customer and product centric. SDF’s agility consists in approaching situations as they arise, a philosophy captured by the phrase “think big, start small, and deliver quickly.” In the literature, agile programming has been characterized by four guiding principles: (1) build good systems quickly ; (2) automate only well-defined routine activities; (3) empower people, not computers, to handle exceptions; and, (4) continually build systems based on experience (Hugos, 2006b). These principles are similar to SDF’s principles for agile development: (1) people over processes; (2) customer collaboration over contract negotiation, (3) flexibility over following the plan, and (4) focus on products rather than design.

Requirements change quickly in SDF’s industry. At time 1, case-based software development tools were used to deal with these, but the SDF teams felt these were too often set in iron. They did not allow programmers to dynamically interact with the customer. As development complexity raises the need for personnel continuity emerged – “it is all about the people”. At time 2, the product leaders, acting as the customer, use their domain expertise to role-play with the programmers to help them understand and anticipate clients’ needs.

Product Development and Agility

Table 2 summarizes the mechanisms and requisite capabilities for agility in the progression from time 1 to time 2. As its products hanged from prototype-proof of concept to fully-workable industry-proof systems, SDF’s has had to start providing software support to its clients and sometimes locating support staff at the client’ site.

Table 2: Mechanisms and Capabilities Related to Agility

	Time 1	Time 2
Design Strategy	Focused – Prototype, demonstration systems for Military and Government agencies	Differentiation – Fully operational, rich featured systems for Military and Government agencies and private, commercial clients
	Internal Development – build staff expertise and core software pieces	Diversification – Some inroads in commercial market for agent-based systems

Design Platform	Platform Architecture – Dual Leadership	Modular-based Architecture – Technical Leadership
Knowledge Database Management	Database specialists used to create Database structures -- requirement to incorporate into adaptive knowledge database structure	Created and deployed Integrated Mapping Tool [IMT] to create homogeneous data structures from heterogeneous data sources
Core Competency	Agent-based spatial Management orientation	Agent-based spatial Management orientation

SDF avoids the project “ramp up -- team assembly” that often occurs in software development, where team members are assembled and scheduled much like contractors and transition in and out of the development based on their skill-set. By contrast, at SDF, teams and team clusters (e.g. development team, testing, and service) provide identification, stability, and long term balance. Table 3 depicts team and software development at SDF.

Table 3: Team Capability Features

	Time 1	Time 2
Team Orientation	Team-based organization with Technical and Product Leads jointly directing Team work (extensive hands-on involvement)	Self-directed Teams under hands-off direction of Technical Lead. Product Lead serves as liaison and facilitator. Both leads support, mentor and coach development team
	Promotes teamwork -- all team members actively participate and contribute regardless of experience level	Team members are cross-trained, software specialists -- they focus on software issues rather than domain specifics (Complimentary skill sets)
	Team comprised of mix of part-time and full-time developers with wide variation in skill set . The team was divided into developers, coders and testers – each having a specific skill set	Team composed of experienced cadre of senior developers with complementary skill sets and minimal part-time developers
Social Aspects	Emphasis on customer involvement at beginning and end of product development	Regard customer to be partner in product specification and development -- continuous involvement through & after product completion
Software Development Approach	Dual product team leadership -- responsibilities shared between Product and Technical Leads--	Technical Lead is sole team leader and assumes responsibility as Domain Specialist Product lead handles liaison with Customer and Support groups

In the software groups, we observed that information sharing and decision making happened as much through face-to-face dialogs and mediated help sessions as through formal help-desks and online retrieval from technical libraries, although this has evolved between times 1 and 2, where at time 1 team members spent considerable time in face-to-face meetings, and at time 2, online instant messaging has replaced much of the face-to-face communication. Through ongoing communication the project scope is clarified and refined (Geoff and Jones, 2007). Focusing on small tasks, as opposed to placing the work to be done in a long term perspective, makes the work more manageable. The shared belief that "...it is better to develop the product correctly the first time rather than having to do expensive re-makes" is a driving force for ongoing deliberations. We observed that there is little re-do in SDF's product development efforts. In the product construction phase, use of the task tool kit to aid the product development team is critical. Software coding and testing is confined to specific tasks and incorporated as needed. The product is reviewed by the customer at each stage of development. Equipment, network infrastructure, training and documentation are integrated and phased to interface with the software product as it reaches completion.

Discussion and Conclusions

This section discusses lessons derived from this study. Developing the agility to address the challenges that confront many organizations today is a growing concern. This challenge has been addressed in the scholarly literature from a variety of theoretical and disciplinary perspectives. In this manuscript, based on the literature on new product development, organizational theory and design, organizational capability, organizational change and development and, information systems, we argue that agility needs an interdisciplinary framework. In this section of the manuscript we chose to identify and explore three of the many areas that emerged from the study as needing future research and dialogue.

The Multiple Forms of Agility

Agility took on different forms in the two times depicted in this paper. During time 1 SDF tested ideas for the military. It was organized to respond in a fluid fashion. Its platform architecture enabled it to insert and remove components depending on the scope of work. Its agent technology and tool boxes allowed software to be reused, expanded and adapted while insulating the core to protect SDF's intellectual capital. SDF's organizational structure was informal and flexible, providing matrix-style team and team-member participation in several projects. Management at the senior and product development team level was divided between technical and product/administrative management. This reflected a participative style which was supportive of the experimental work. Most of SDF's work force was young and inexperienced. Developers and programmers thus had to be nurtured and supported by more experienced product managers.

At time 2, SDF still served as an experimental extension to test ideas for the military. Many of these experiments led to fully-functional production systems. SDF has grown to support and maintain these operating products. SDF is still organized to flexibly respond on the experimental side but has also developed mechanisms to ensure training, technical support, and product enhancement. The organizational structure has changed to, focus on product development, and product support. The architecture has changed from a platform to a mod-

ular design to accommodate the complexity of work. The structure has also become more complex and formal, including administrative, sales, and contract support. At time 2, SDF o introduced a data management tool box to support its changing product base.

Management at the senior and product development team level is no longer divided between technical and product/administrative management. This recognizes the technical nature of the product work. The work force has matured and become professionally experienced – turnover is minimal – the roles of the technical and product managers is still to nurture and support the product team’s development but at a distance – teams are given extensive autonomy. This study provides support to the argument that agility can be designed and managed, design being viewed as choices among alternatives. We further argue that these alternatives can be described as dimensions fitting neatly into the platform (matrix) and modular (semi-organic) architecture schema, each fulfilling requirements for agility. Theoretical and empirical research is needed to capture the dynamic interface between alternative design choices or forms and agility.

Agility as an Organization Capability

A common theme in the literature is that organizational capabilities are composite bundles of coherent competences, skills and technologies, rather than single discrete skills (Mohrman et al., 2006). Developing new capabilities entails the ability to “integrate, build, and reconfigure internal and external competencies to address rapidly changing environments” (Teece et al., 1997: 516). Much of the literature on organizational capabilities rests on the assumption that knowledge is the strategically most important resource of the firm. Knowledge grows through two generic processes (Ghoshal & Moran, 1996), the combination of previously unconnected knowledge that leads to novel solutions and the exchange of knowledge between actors. Both of these are social processes, and occur within a network of connections that characterizes an activity system.

At time 2, SDF reflects the capability to support organizational agility through the development techniques that have been introduced. The ability to develop and incorporate the Task Tool Box provides the support for agile programming work. The Task Tool Box aided the software development teams’ interface with clients. The interface has become a direct interface without the need to rely on liaisons (the technical and product leader) to insure that the customer is aware of the content and progress of product development work. Since most of the literature does not focus on the interface between organizational capability and agility, additional theoretical and empirical research work is needed.

Learning Mechanisms and Agility

The accelerated pace of change needs agility. The development of organizational capabilities for agility requires the need to create the space and time for social processes to evolve. Learning mechanisms are viewed as the formal and informal configurations – structures, processes, procedures, rules, tools, methods, and physical configurations – created in the social system of the firm to develop and enhance performance and learning while developing human capital (Friedman et al, 2001; Lipshitz et al, 2006; Shani & Docherty, 2003). Just like the many types of organization designs, there are also many ways to design and manage organization learning mechanisms. The design of a specific learning configuration is viewed

as a rational choice between alternatives based on needs and desired outcomes. Exploring alternatives and making informed choices serves as a potential new organizational capability.

The establishment of mechanisms for information-sharing and deliberations at SDF has fostered an ongoing opportunity to improve business results and learning at all levels and across the firm. The modular architecture and clear chain of command as evidenced by the technical director and technical leaders provide a context to support a work environment that enhances the organization's ability to develop. At SDF, a wide tapestry of learning mechanisms has been created. The structural, procedural, and cognitive mechanisms have been combined to facilitate the learning processes in different units, in work groups, and between individuals or groups from different units and levels in an organization. For example, each one of the structural learning mechanism that have been created includes a variety of different competence, experience, values, professional or organizational identity, goals and priorities. It is the bundling of the competences, skills, technology and other resources that have created new knowledge and new organizational capabilities. Yet, we need additional theoretical and empirical research work that looks at the role and design of the different mechanisms and the combination of learning mechanisms on organization agility development.

More practically, agility has enabled SDF to overcome three strategic challenges typical of the software industry: extensive reworks, death marches, and client support services. Our study of the company at two points in time separated by a five years interval has shown how agility – a firm ability to continuously transform – contributes to sustained growth, effectiveness, and client satisfaction.

References

- Ansoff, H.I. (1980). Strategic issues management. *Strategic Management Journal*, 2 (1), 132-148.
- Aoyama, M. (1998). Web-based agile software development, *IEEE Software*, 15(6), 55-65
- Boden, T. (2004). The grid enterprise – structuring the agile business of the future. *BT Technology Journal*, 22 (1), 107-117
- Bonabeau, E., Nodick, N., Armstrong, R. (2008). A more rational approach to new product development. *Harvard Business Review*, 86 (3), 96-102.
- Borjesson, A., Martinsson, F., Timmeras, M. (2006). Agile improvement practices in software organizations. *European Journal of Information Systems*, 15, 169-182.
- Cockburn, A. (2001). *Agile software development*. Reading: Addison- Wesley.
- Conboy, K.B., and Fitzgerald, B., (2004). Towards a conceptual framework of agile methods: A study of agility in different disciplines. *WISER '04*, ACM, New York.
- Dove, R. (1999). The meaning of life and the meaning of agile. *Production*, 106, November, 14-15.
- Dove, R. (2001). *Response ability: The language, structure and culture of the agile enterprise*. New York: John Wiley & Sons.
- Ebrahimpur, G., and Jacob, M. (2001). Restructuring for agility at Volvo car technical services (VCTS). *European Journal of Innovation Management*, 4 (2), 64-72.
- Fink, L. (2007). Gaining agility through it personnel capabilities: The mediating role of IT infrastructure capabilities. *Journal of the Association for Information Systems*, 8 (9), 440-419.
- Friedman, V. J., Lipshitz, R., & Overmeer, W. (2001). Creating conditions for organizational learning. In A. B. Antal, M. Dierkes, J. Child & I. Nonaka (Eds.), *Handbook of organizational learning and knowledge*, pp. 757-774. New York: Oxford University Press.
- Gartner Research (2004). *Designing the agile organization: Design principles and practices*. January 6, 2004. Gartner Inc. Stamford, CT.
- Geoff, R. and Jones, G. (2007). Leading clever people. *Harvard Business Review*, March, 72-79.

- Ghoshal, S., & Moran, P. (1996). Bad for practice: A critique of the transaction cost theory. *Academy of Management Review*, 21, 13–47.
- Goldman, S., Nagel, R and Preiss K. (1995). *Agile competitors and virtual organizations*. New York :Van Nostrand Reinhold.
- Hooper, M.J., Steeple, D., and Winters, C.N. (2001). Costing customer value: An approach for the agile enterprise. *International Journal of Operations and Production Management*, 21, 56, 630-644.
- Hugos, M.H. (2006a). The value of IT agility. *Computerworld*. Jul 10, 2006. 40 (28), 22-24
- Hugos, M.H. (2006b). Agility lessons learned. *Computerworld*. Aug 7, 2006. 40 (32), 20-22.
- Hugos, M.H. (2007). The agility factor. *Computerworld*. Aug 13, 2007. 41 (33), 26-28.
- Kidd, P.T. (1995). *Agile manufacturing: Forging new frontiers*. London: Addison-Wesley.
- Lipshitz, R., Friedman, V., & Popper, M. (2006). *The demystification of organizational learning*. Los Angeles, CA: Sage.
- Mathiassen, L. and Pries-Heje J. (2006). Business agility and diffusion of information technology. *European Journal of Information Systems*. 15 (2), 116- 122.
- Merriam-Webster online (2005). <http://www.m-w.com/dictionary/agile>. Found 11 November 2005.
- Mohrman, S., Docherty, P., Shani, A.B. (Rami), Teigland, R., Schenkel, A.J. (2006). *The development of new organizational capabilities: A design-based model for managerial action*. Paper presented at the Academy Of Management Annual Conference, Atlanta, August.
- Overby, E. Bharadawaj, A, and Sambamurthy. (2006). Enterprise agility and the enabling role of Information Technology. *European Journal of Information Systems*, 15 (2), 120 -132.
- Ramasesh R, Kulkarni S and Jayakumar M., (2001). Agility in manufacturing systems: An exploratory modelling framework and simulation. *Integrated Manufacturing Systems* 12(6/7), 534–548.
- Shani, A. B. (Rami), Sena, J., Olin, T., (2003). Knowledge management and new product development: A study of two companies. *European Journal of Innovation Management*, 6 (3), 137-149.
- Sena, J., and Shani, A.B. (Rami), (2003), Knowledge Management Processes and Learning. In Shani, A.B. (Rami) and Docherty, P. (Eds), *Learning By Design*. London: Blackwell Publishing, 127-144.
- Shani, A.B (Rami) & Docherty, P., (2003). *Learning by design: Building sustainable organizations*. London: Blackwell Publishing.
- Sharifi,H., and Zang, Z. (2001). Agile manufacturing in practice: Application of a methodology. *International Journal of Operations and Production Management*, 21 (56), 772-794.
- Simon, R. (2007). Control in an age of empowerment. *Harvard Business Review On Point: Executive Edition – The Creating Company*, Spring, pp. 106-114.
- Teece D., Pisano G., and Shuen A. (1997). Dynamic capabilities and strategic management. *Strategic Management Journal*. 7 (18), 509-533.
- Van Oosterhout, M., Waarts, E., and van Hilleegersberg, J. (2006). Change factors requiring agility and implications for IT. *European Journal of Information Systems*, 15, 132–145.
- Volberda, H.W. and Rutges, A. (1999). FARSYS: A knowledge-based system for managing strategic change. *Decision Support Systems*, 26, 99–123.
- Volberda, H.W. (1997). Building flexible organizations for fast-moving markets. *Long Range Planning*, 30(2), 169–183.
- Van Oosterhout, M, Waarts, E., and Van Hilleegersberg, J. (2006). Change factors requiring agility and implications for IT. *European Journal of Information Systems*, 15 (2), 132-145.
- Zahra, S.A., and George, G. (2003). Absorptive capacity: A review, reconceptualization and extension. *Academy of Management Review*, 2 (27), 185-203.

About the Authors

Dr. James Sena

Professor of Management & Information Systems Education Ph.D. Management Information Systems, University of Kentucky M.S. Information Science, University of Dayton, MBA Operations Research, Xavier University of Ohio, Graduate work, Actuarial Science, University of Iowa, B.S. Mathematics, Xavier University of Ohio, Teaching Specializations Organization Systems and Technology, Project Management, Computer Security. Network Systems, Java, Data Structures, Management Information Systems, Decision Support and Expert Systems, Operations research, Data Warehouse & Data Mining Additional College/University Experience Visiting Research Scholar, Norwegian Institute of Technology, Fulbright Award, Trondheim, Norway; Assistant Professor, Texas A and M University, College Station, Texas; Assistant Professor, University of Houston, Clear Lake City, Texas; Assistant Professor, University of Louisville. Research Interests and Experience Knowledge Management, CISCO Network Management; Agent-based components; Platform Architectures; Sustainable Work Systems; Process Analysis and Reengineering, Research and Development of group decision making and organizational analysis computer software. Use of Windows software development tools to create management support systems - Visual Studio, Java, MS Excel & Access, and impact of computer-based information technologies. Areas Published Computer Security, Learning Mechanisms, Sustainable Work Systems, knowledge Management, Intellectual Capital, Accounting Information Systems, Deployment of Software Tools in Organizations, Impact and Assessment of Local Area Networks in Organizations, Expert Systems, Decision Support Systems, Impact and Assessment of Information Technologies on Organizations.

Dr. Jean-Francois Coget

Jean-François Coget is an Assistant Professor of Management at the Orfalea College of Business at CalPoly, San Luis Obispo, where he teaches Organizational Behavior and Business Consulting classes. He earned his Ph.D. from the Anderson School at UCLA, and was previously a faculty member at HEC Paris. His research interests include emotions in organizations, intuitive decision-making, and charismatic leadership. His overarching professional goal is to contribute to the creation and the dissemination of actionable knowledge and practices that can help managers-citizens to organize a productive, creative, ethical, and sustainable cooperation among people.

A.B. (Rami) Shani

California Polytechnic State University, USA

EDITORS

Mary Kalantzis, University of Illinois, Urbana-Champaign, USA.

Bill Cope, University of Illinois, Urbana-Champaign, USA.

EDITORIAL ADVISORY BOARD

Verna Allee, Verna Allee Associates, Martinez, USA.

Zainal Ariffin, Universiti Sains Malaysia, Penang, Malaysia.

Robert Brooks, Monash University, Melbourne, Australia.

Bruce Cronin, University of Greenwich, London, UK.

Rod Dinnutt, William Bethway and Associates, Melbourne, Australia.

Judith Ellis, Enterprise Knowledge, Melbourne, Australia.

Andrea Fried, Technology University Chemnitz, Chemnitz, Germany.

David Gurteen, Gurteen Knowledge, Fleet, UK.

David Hakken, University of Indiana, Bloomington, USA.

Sabine Hoffmann, American University of the Middle East, Kuwait.

Stavros Ioannides, Pantion University, Athens, Athens, Greece.

Margaret Jackson, RMIT University, Melbourne, Australia.

Paul James, RMIT University, Melbourne, Australia.

Leslie Johnson, University of Greenwich, London, UK.

Eleni Karantzola, University of the Aegean, Rhodes, Greece.

Gerasimos Kouzelis, University of Athens, Athens, Greece.

Krishan Kumar, University of Virginia, Charlottesville, USA.

Martyn Laycock, University of Greenwich; managingtransitions.net, London, UK.

David Lyon, Queens University, Ontario, Canada.

Bill Martin, RMIT University, Melbourne, Australia.

Pumela Msweli-Mbanga, University of Kwazulu-Natal, South Africa.

Alexandra Roth, Northeastern University, Boston, USA.

Claudia Schmitz, Cenandu Learning Agency, Cologne, Germany.

Kirpal Singh, Singapore Management University, Singapore.

Dave Snowden, Cynefin Centre for Organisational Complexity, UK.

Voicu Sucal, Technical University of Cluj-Napoca, Napoca, Romania.

Chryssi Vitsilakis-Soroniatis, University of the Aegean, Rhodes, Greece.

JoAnn Wiggins, Walla Walla University, Walla Walla, USA.

Alan Zaremba, Northeastern University, Boston, USA.

THE UNIVERSITY PRESS JOURNALS



Creates a space for dialogue on innovative theories and practices in the arts, and their inter-relationships with society.

ISSN: 1833-1866

<http://www.Arts-Journal.com>



Examines the meaning and purpose of 'design' while also speaking in grounded ways about the task of design and the use of designed artefacts and processes.

ISSN: 1833-1874

<http://www.Design-Journal.com>



Maps and interprets new trends and patterns in globalisation.

ISSN 1835-4432

<http://www.GlobalStudiesJournal.com>



Sets out to foster inquiry, invite dialogue and build a body of knowledge on the nature and future of learning.

ISSN: 1447-9540

<http://www.Learning-Journal.com>



Addresses the key question: How can the institution of the museum become more inclusive?

ISSN 1835-2014

<http://www.Museum-Journal.com>



Draws from the various fields and perspectives through which we can address fundamental questions of sustainability.

ISSN: 1832-2077

<http://www.Sustainability-Journal.com>



Investigates the affordances for learning in the digital media, in school and throughout everyday life.

ISSN 1835-2030

<http://www.ULJournal.com>



Explores the past, present and future of books, publishing, libraries, information, literacy and learning in the information society.

ISSN: 1447-9567

<http://www.Book-Journal.com>



Provides a forum for discussion and builds a body of knowledge on the forms and dynamics of difference and diversity.

ISSN: 1447-9583

<http://www.Diversity-Journal.com>



Discusses the role of the humanities in contemplating the future and the human, in an era otherwise dominated by scientific, technical and economic rationalisms.

ISSN: 1447-9559

<http://www.Humanities-Journal.com>



Creates a space for discussion of the nature and future of organisations, in all their forms and manifestations.

ISSN: 1447-9575

<http://www.Management-Journal.com>



Discusses disciplinary and interdisciplinary approaches to knowledge creation within and across the various social sciences and between the social, natural and applied sciences.

ISSN: 1833-1882

<http://www.Socialsciences-Journal.com>



Focuses on a range of critically important themes in the various fields that address the complex and subtle relationships between technology, knowledge and society.

ISSN: 1832-3669

<http://www.Technology-Journal.com>



Explores the meaning and purpose of the academy in times of striking social transformation.

ISSN 1835-2030

<http://www.Universities-Journal.com>

FOR SUBSCRIPTION INFORMATION, PLEASE CONTACT

subscriptions@commonground.com.au