

Introducing Principles for Agile Systems

Rick Dove, Sr. Fellow, Agility Forum, dove@well.com, Paradigm Shift International, 505-586-1536

Being Agile means being a master of change, and allows one to seize opportunity as well as initiate innovations. How Agile your company or any of its constituent elements is, is a function of both opportunity management and innovation management - one brings robust viability and the other brings preemptive leadership. Having one without the other is not sufficient in these times of quickening unpredictable change. Having neither is a time bomb with a short fuse today.

How much of each is needed at any time is a relative question - relative to the dynamics of the competitive operating environment. Though it is only necessary to be as Agile as the competition, it can be extremely advantageous to be more Agile.

All of this talk about "how Agile" and "more Agile" implies we can quantify the concept, and compare similar elements for their degrees of Agility. However, as the associated figure shows, there is some question about value tradeoffs between an increment of

"Getting to your chosen spot and staying there is a job for masters at business engineering, not business administration."

leadership and an increment of viability.

Leadership wins if the leader always chooses the most optimal path to advance - but one false step allows a competitor to seize the advantage; putting the previous leader in reaction mode. A competitor with excellent viability can track the leader, waiting for that sure-to-come mistake. Poor viability may then keep the fallen-from-grace ex-leader

spending scarce resources on catch-up thereafter.

Choosing a desired spot in the Agile quadrant is one of the important ways to strategically differentiate yourself from your competitors. Getting to your chosen spot is another issue entirely -

and a job for masters at business engineering, not business administration.

How innovative/opportunistic are you - relative to your competitive needs and environmental situation? How fast are the rules changing in your market? Are you able to respond fast enough, can you introduce a few changes of your own? Importantly - what allows you to do that? We will look shortly at some promising design principles to answer this last question.

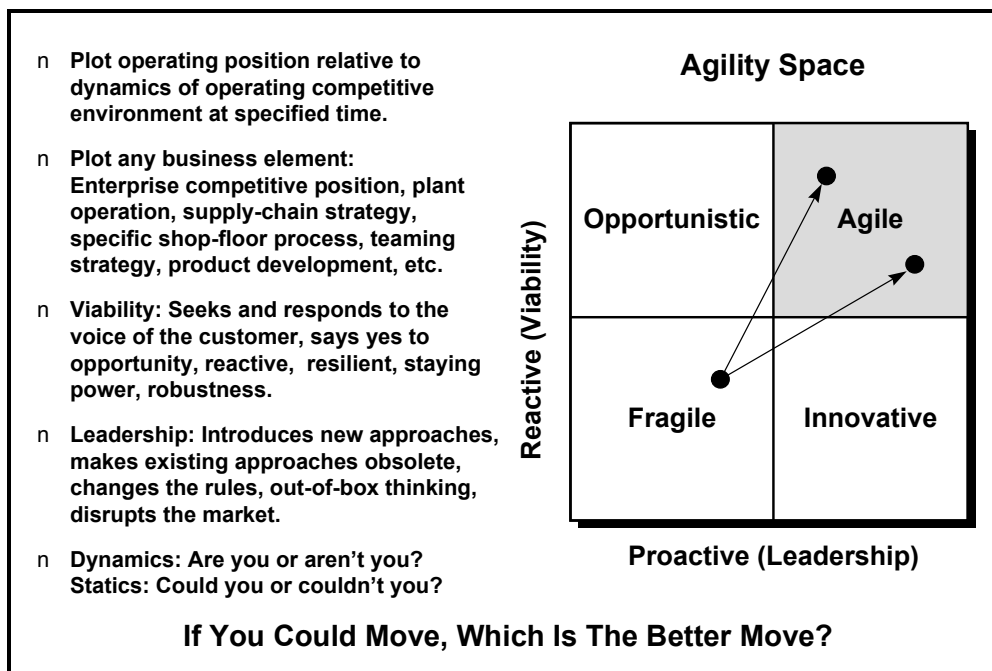
The search for metrics and analytical techniques that can pinpoint an enterprise in the Agility space is receiving a lot of attention today. Self analysis tests that ask lists of questions are one form, house-of-quality QFD-like templates are another. These have a certain appeal in that they deal with familiar concepts that enjoy intuitive association with Agility: teaming, empowerment, partnering, short-cycles, integrated process and product development, and so forth.

But experience shows us that simply saying yes to these questions will not tell us anything useful - too many people, for instance, will say yes to having empowered teams when the yes-ness has nothing to do with the quality of the implementation, or if the implementation promotes Agility.

Better to ask how well we respond to critical types of unexpected situations, how often we lead with a meaningful innovation, how proficient we are at a variety of identified change we feel to be strategically important. For sure, empowered teams

embody an organizational structure and business practice that can help us be more Agile if they are designed and supported with that end in mind.

Firstly, we must locate ourselves (qualitatively) in the Agile space relative to our environmental and competitive realities - a subject of some depth that will be covered at another time. Then, where we find ourselves wanting, we must select and design appropriate strategies to move us to where we want to be. The selection of appropriate strategies will change with the times, and differ from market to



market; and today might include concepts like mass customization, virtual enterprise, empowerment, commonization, listening to your customer, and other such.

Strategic concepts by themselves are open to a wide range of interpretation, and are often interpreted incorrectly. Commonization in shop-floor controls, for instance, doesn't pay Agility dividends if it is interpreted as buying controls from one vendor; empowerment doesn't pay without an information and support infrastructure; and customer listening doesn't pay when competitors change the rules.

Regardless of the strategies chosen, effective implementation will employ a common set of fundamental design principles that promote proficiency at change.

Designing Agile systems, whether they be entire enterprises or any of their critical elements like business practices, operating procedures, supply-chain strategies, and production processes, means designing a sustainable proficiency at change into the very nature of the system. With the business engineer's eye we will be interested in both the statics and the dynamics of these systems - where the static part is the fundamental system architecture and the dynamic part is the day-to-day reengineering that reconfigures these systems as needed.

Seeking and sustaining a desired opportunistic/innovative profile will rely upon the Agility of these systems, which in turn will be impeded or enabled by their underlying architectures. Earlier essays have discussed Reusable/Reconfigurable/Scalable (RRS) system strategies that employ a Framework/Module approach. The accompanying table completes the engineering strategy by advancing a set of design principles for these RRS systems. These principles have emerged from observations of both natural and man-made systems that exhibit RRS characteristics, with contributions from the Forum's Agile Practice Reference Base, Kevin Kelly's thought-

provoking book (reference below), and the sizable body of knowledge growing out of object oriented systems design.

We will explore the application of these early stage principles in the next issue, tying them into various business strategies critical for the Agile enterprise.

Business strategists recognize the imperative of the Agile enterprise, with virtually all popular business writers today extolling the need for change proficiency of one kind or another: Peter Drucker, Tom Peters, Michael Hammer, Peter Senge, and all the rest are talking about today's dominant effects of a faster paced business environment. The newly released Agile Competitors and Virtual Corporations by Goldman, Nagel, and Preiss provides an encompassing view for most of these concepts with many examples. Richard D'Aveni's excellent Hypercompetition focuses on wielding change proficiency as a preemptive business strategy. And Kevin Kelly's Out of Control provides fundamental examples for the business engineer who would design and build Agile enterprises and systems.

Agile System Principles

Any organization of interacting units is a "system": an enterprise of business resources, a team of people, a cell of workstations, a contract of clauses, or a network of suppliers.

Self Contained Units

System composed of distinct, separable, self-sufficient units not intimately integrated.

Plug Compatibility

System units share common interaction and interface standards, and are easily inserted or removed.

Facilitated Re-Use

Unit inventory management, modification tools, and designated maintenance responsibilities.

Non-Hierarchical Interaction

Non-hierarchical direct negotiation, communication, and interaction among system units.

Deferred Commitment

Relationships are transient when possible; fixed binding is postponed until immediately necessary.

Distributed Control & Information

Units respond to objectives; decisions made at point of knowledge; data retained locally but accessible globally.

Self Organizing Relationships

Dynamic unit alliances and scheduling; open bidding; and other self adapting behaviors.

Flexible Capacity

Unrestricted unit populations that permit large increases and decreases in total unit population.

Unit Redundancy

Duplicate unit types or capabilities to provide capacity fluctuation options and fault tolerance.

Evolving Standards

Evolving open system framework capable of accommodating legacy, common, or completely new units.