AN AGILE SYSTEMS FRAMEWORK: A FOUNDATION TOOL

Steve Benson, Rick Dove, Joe Kann

Introduction

Few would argue that information and automation systems are critical enablers for modern production. So its seems very likely that they will also play a key role in the agile production environment of the future. But what will an agile information system and automation system look like? Will they require advanced computing and communications technologies that are beyond our current capabilities? Will they be so sophisticated that they eliminate the need for people in production?

Answering these and other related questions is one of the key missions of the Agile Production Focus Group. This is not a straight-forward task. First, there is the challenge always associated with visualizing the future. Good crystal balls are pretty hard to come by. Second, is the added element of complexity of scope. While it's possible to describe future requirements for a single application in a single industry, it was not nearly as simple to extend those thoughts over a broader scope - e.g., small vs. large users, process vs. discrete industries, engineering information systems vs. shop floor management systems vs. real-time automation systems, one technology vs. another, etc.

It was decided that a detailed description of THE agile information system was not practical or even possible. Instead, the Group developed a general framework of system values, principles, and attributes which would specifically support agility. The framework is intended to guide researchers, vendors, implementors and users as agile systems evolve. Readers should look at it as providing some initial sign posts on the journey to agility. Mapping of the framework onto a particular implementation is left to the reader as an exercise of discovery.

The framework presented here is only a start. It too must evolve as the broader concepts of agility come more clearly into focus. Feedback to the AMEF and the Agile Production Focus Group is encouraged.

Approach

A three-tiered description of an agile system is presented below: values are first established upon which we build key principles, which then illuminate obvious attributes. At the highest level are the values upon which agility is based. Values are defined in the dictionary as: standards, or qualities considered inherently worthwhile or desirable. Next come the 1st Principles: basic truths, laws or assumptions required to meet a value. Finally come the attributes: the characteristics of systems resulting from the adherence to the 1st principles.

At the end of this report is a description of how these same attributes can, and should, be extended across all aspects of agile production, i.e. the organizational structures, the machines and processes , the designs of products, and the structure of supply chains.

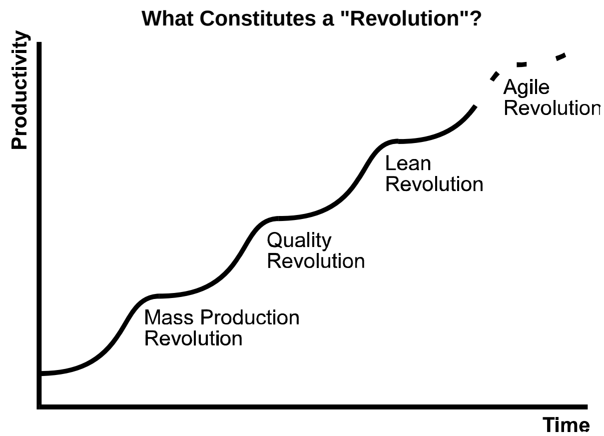
Values: So You Say You Want a Revolution

Much has now been said and written about the upcoming agile "revolution". But what are the broad values that will drive the revolution? To answer this question it is useful to consider the values that have driven previous revolutions in industrial production.

From the early days of industrialization, progress has been characterized by a slow but steady pace of productivity improvement, interrupted by an occasional, significant production revolution. This is depicted in the figure below. Each step built on the lessons of the past. Each resulted in productivity gains far in excess of what was previously possible. And most significantly, each resulted from a fundamental rethinking of the basic underlying values of production.

This of course raises the question: if there is going to be an agile revolution, what fundamental value of production needs to come into focus? Consider the definition of agility: the ability to thrive on, and respond effectively to, unforeseen change. It follows that the agile revolution will usher in a time when the total costs and benefits associated with **change** are finally focused upon.

In this preliminary document we are focused on the information systems area. Not because this area of technology is the only impediment or enabler of an agile operation, but because it is one of the major impediments and enablers, and needs to be understood along with the impacts of organizational structures, human factors, production equipment architecture, and others.



o	Mass Production Revolution:	Value of Scale
o	Quality Revolution:	Value of Defects
o	Lean Revolution:	Value of Time
o	Agile Revolution:	Value of ?

Thus the fundamental premise upon which agile information and automation systems will be built is a recognition of the **value of change**. In the production area, as in the rest of the enterprise, agile systems must vastly improve the ability to enact continuous daily improvements as well as major changes in system functionality, capacity and configuration.

<u>1st Principles of Agile Systems Architecture</u>

So where will this capability come from? Will there be a magical new computing technology that will arrive on the scene and suddenly make a system agile? We do not believe so. What we do believe is that, just like in previous production revolutions, the true breakthroughs will come not from the parts which make up the system, but from the structural relationship <u>between</u> the parts of the system. Implicit in this statement is the need for an architecture that enables agility.

Three differentiations from traditional architectures can be summed up in the following principles: communications transparency, user apparentness, and human/organizational enablement.

Communications Transparency:

A key requirement for agile systems is the transparent distribution of data anywhere it is needed, in a timely fashion. By transparent distribution we mean delivery of information without regard for the physical location of the sender or receiver, or the physical communications media connecting them. A system that thrives on change must be immune to changes in the physical communicationsinfrastructure as well as the change in location of senders and receivers.

The scope of this transparency extends from the highest levels of the enterprise information system down to the "smart" sensors and actuators that populate the factory automation system. Transparency must also extend beyond traditional bounds of the system to include all "agents" of the system, whether they be machine, human or hybrids. In addition, transparency must extend beyond the bounds of a single enterprise to encompass the 3rd party suppliers, channel partners, and even customers at the point of sale. The Agile Production Focus Group fully concurs that the vision of Factory America Net is critical for agile production.

User Apparentness:

The ability to deal with unexpected change efficiently is for the most part largely lacking in today's systems. The brittle nature of these structures is due in part to the lack of communications transparency. But it is also a result of the lack of visibility that the user has to the system. You can't change what you don't understand.

This is a much deeper issue than simply a nice user interface on a terminal. It requires a system structure that can be explicitly visualized at whatever level is appropriate for the user to speedily enact the change required. Clear and concise architectural structures must be apparent to the users.

Architecture compliance is required in order to achieve agile systems. This means that the architecture, and specifically the interactions between subsystems, must not be vague and interpretive. User apparency of the architecture will go farther to insure architecture compliance than any set of standards and procedure manuals can ever hope to achieve.

Human/Organizational Enablement:

Humans are recognized as the most vital part of the agile production environment. It is our natural ability to deal with uncertainty and ambiguity that makes our role so critical to achieving agility (remember, agility involves responding to **unexpected** change). Therefore agile systems must above all be human enabling. They must be viewed and embraced by workers as tools for production, not hindrances.

There is a technical and business architecture that guides the enterprise. The technical architecture must be able to map to the business architecture and not constrain changes in the business. For agility, the systems must act as both enablers of and catalysts for change.

Agile Systems Architecture Attributes

How can an honest architecture for agile systems not also, itself, be agile? The architecture must be viewed as revisable and living. This implies that there will be current best practices in the architecture. As we learn what works and what doesn't, the architecture must undergo version changes in order to reflect the newest understandings. The architecture itself must exemplify agility.

As a starting point, the following six attributes are offered as potentially fundamental to an agile systems architecture. There are surely others, and those that are offered will surely be better articulated as experience is gathered.

- Peer Communications.
- Encapsulation / Modularity.
- Loose Coupling.
- Distributed Control / Functionality.
- Redundancy.
- Scalability.

Although we have decomposed the architectural issues into separate areas, as you read through the following pages it will be apparent that they are not separate and distinct, but rather connected and somewhat overlapping. The attempt here is to postulate first level attributes for the architecture of agile systems, saving rigor for another time.

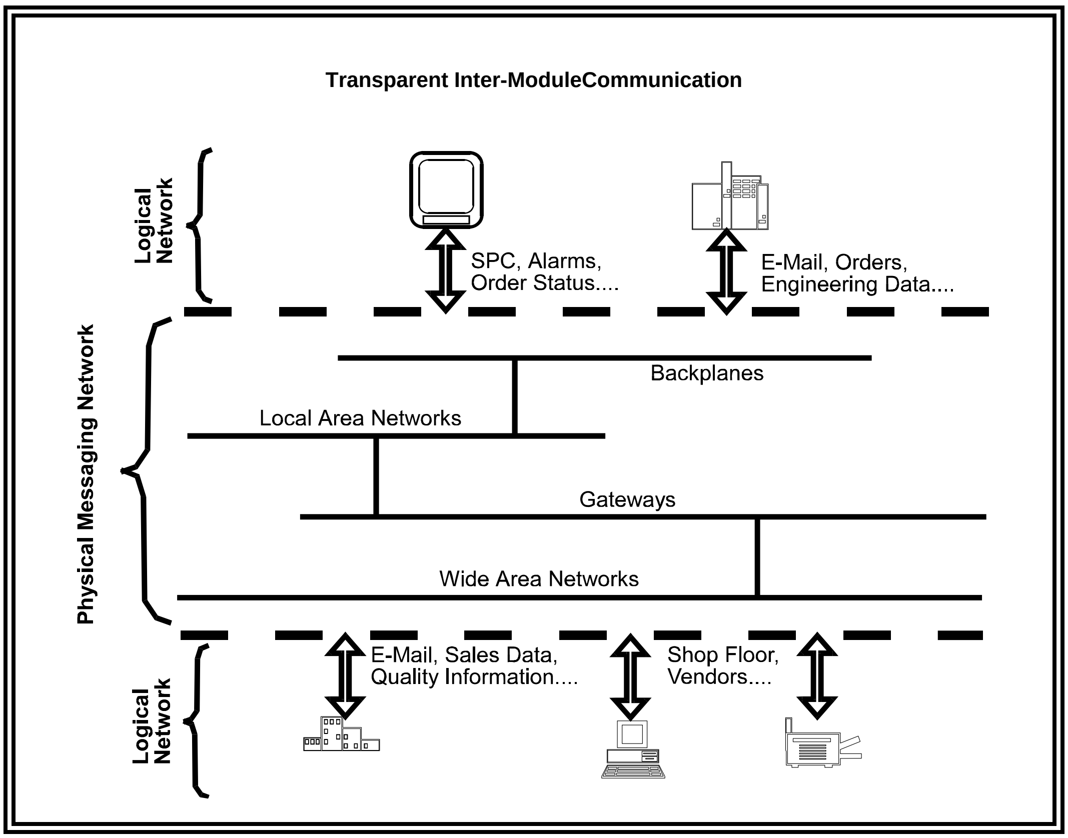
Peer Communications:

- The base architecture for Agile Systems is based on a computing environment that is networked and distributed.

- Communications between application modules in partnership will be transparent to the partners. One will not have to know where the other physically resides, be it another local task, across a backplane, on a cluster, down the hall, halfway across the country, on the other side of the world, or up in earth orbit. The figure below illustrates this idea of a logical network separate from the message switching network, hiding the complexity of the underlying physical communications structure.

- Communication between modules on the network will be intrinsically peer to peer, each physically capable of connecting with another. Logically, artificial hierarchies may be constructed.

- Interaction with the user will be intrinsically easy, with local "preferences" facilitated by the system.



Encapsulation / Modularity:

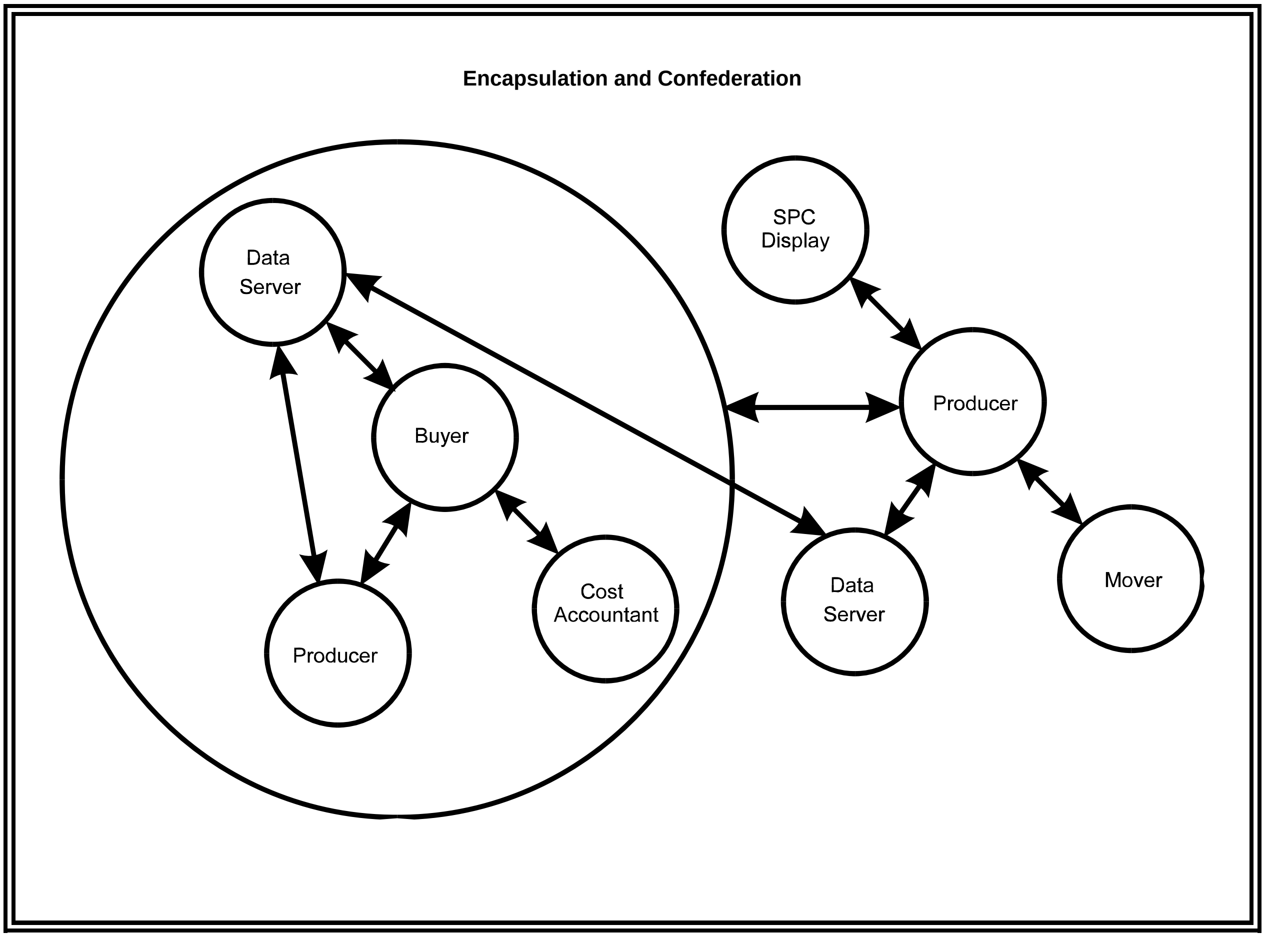
- o By defining modules in the form of encapsulated objects on the network, the internals of the modules are isolated from (opaque to) each other. This enables transparent migration of the physical, algorithmic, and data content of an object while its functional perimeter remains unchanged. This provides a migration strategy for both older legacy systems and newer system modules that will become obsolete in the course of continuous change.

- o Data management functions will be encapsulated just like other modules, and service all appropriate requests within security boundaries.

- o From the point of view of the agile system, everything is a resource: machines, computers, people, materials, soft tools, etc. This implies that groups of resources can band together in order to perform a task, decouple and reform in another group to perform other tasks. As resources group, they in turn can be viewed as a resource. thus providing a recursiveness to the execution environment.

Loose Coupling:

- o Modules will interact with each other through mechanisms that do not pierce, or require knowledge beyond, the "opaque" perimeter of another module. This encapsulation serves to



localize and reduce side effects in a system that undergoes constant change.

- The architecture will support large (unlimited) numbers of modules of arbitrarily small size in order to encourage the decomposition of systems into (relatively) fine granularity. This fine granularity facilitates continuous change by reducing the potential for inhibiting deleterious side effects.

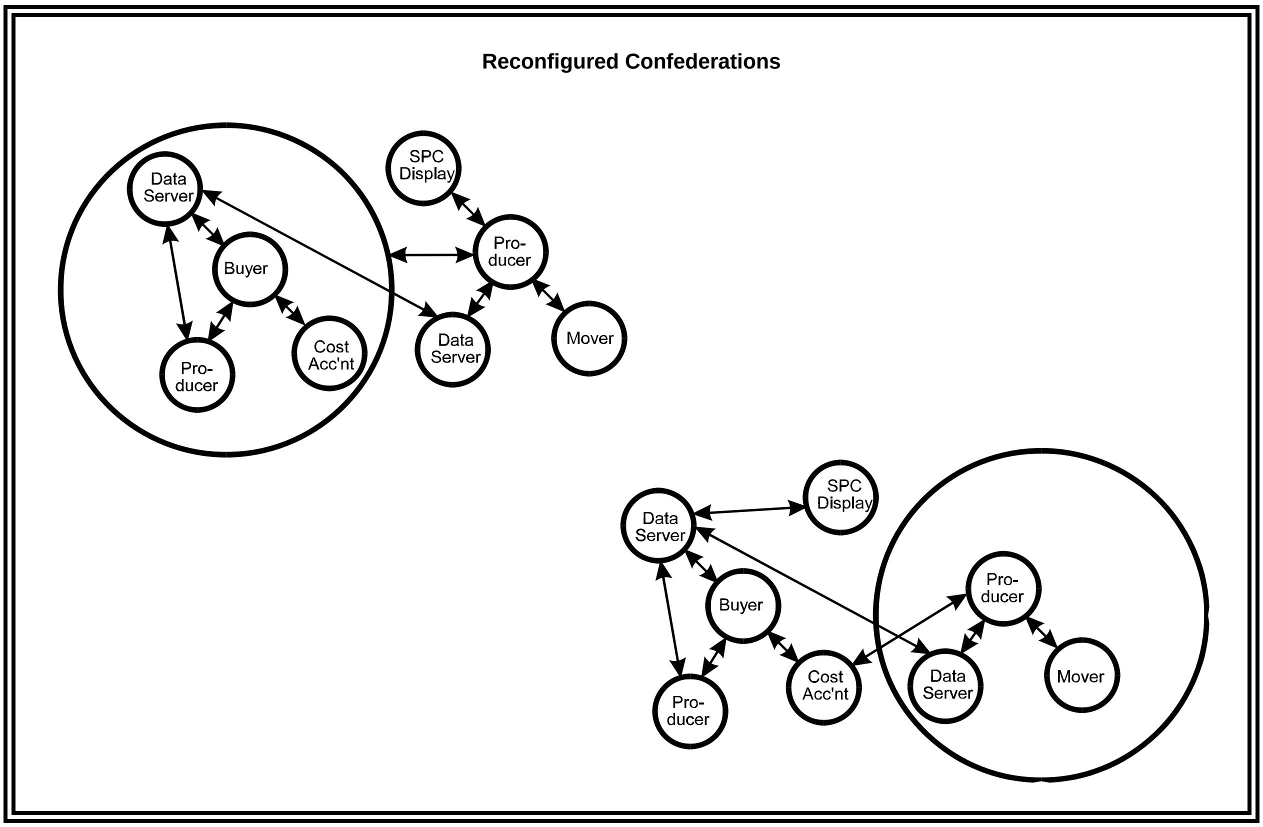
- The architecture will not require that modules know where they exist physically, in order to minimize the potential of side effects from relocating them. Furthermore, the architecture will not require confederations of modules that form in pursuit of an objective to have fixed physical locations during the life of the confederation.

- All relationships between modules are considered temporary, although some may last for an indefinite period of time. Dynamic "confederations" will form to accomplish objectives and disband to reform into different confederations as needed.

- Relationships among modules are formed in real-time. Such "late binding" facilitates the replacement with improved, and insertion of additional, capabilities at any time.

Distributed Control / Functionality:

- A basic tenet of agility is that individuals be empowered in order to make decisions quickly and at the point of maximum knowledge. This same principle applies to the information systems architecture as well. For instance, functionality is distributed to the point where it is most needed



when intelligent sensors perform first level data reduction; making some closed loop decisions immediately and greatly reducing communications data traffic.

- o The architecture will enable (but not require) decentralized planning and scheduling, such as autonomous module bidding schemes, so that some modules might continue to function even when portions of the total system fails to perform as expected.

Redundancy:

- o The architecture will support virtually unlimited module redundancy to enable, for instance, operational robustness, side-by-side testing of new capabilities, and increases in system capacity.
- o Nothing in the architecture will inhibit the free insertion and deletion of redundant modules while a system is operational.

Scalability:

- o To meet changing performance requirements the architecture must not inhibit the free relocation of software modules from one platform to another. For example, when faster cycle times, more memory, or different underlying operating systems become important.
- o The architecture must support the addition, subtraction, and replacement of modules during system operation, allowing module improvement and system capacity change to occur on the fly.

Extending the Agile Systems Framework

Is an agile automation / information system enough for a business to achieve agile production? If history is any indication, the answer is No. In previous production revolutions, change was not limited to just a single aspect of the enterprise. While a single technology or technique (e.g., JIT in the Lean Revolution) may gain wide visibility, the changes ultimately impact all aspects of the business. This includes the information and automation systems, machines and processes, organizations and workers roles, and product designs and supplier relations. Examples from each of the prior revolutions are as follows:

Mass Production Revolution (value of scale):

- o Scientific management.
- o Interchangeable parts.
- o High volume / repetitive automation.
- o Little variety (any color as long as it's black).
- o Vertical integration.
- o People viewed as labor / narrow job description.

Quality Revolution (value of quality):

- o Continuous process improvement.
- o Quality circles/ worker participation.
- o In-process inspection / SPC.
- o Machine - process qualification / Cpk.
- o Taguchi / design of experiments / GD&T.
- o Supplier certification.

Lean Revolution (value of time):

- o Time-based competitiveness.
- o Process synchronization / inventory minimization / pull systems / JIT.
- o Rapid change-over / flexible automation.
- o DFM / DFA / Group Technology.
- o Concurrent engineering.
- o EDI.
- o Self-directed teams / empowered workers.

In each of these cases a single change in the value structure was translated into a general set of competitiveness concepts (e.g., scientific management) which drove all aspects of the business. For agility to be a successful revolution, we believe a similar value-driven philosophy must take hold. The resulting broad-reaching changes would not only complement the changes described above for the agile information systems architecture, but are in fact necessary if the benefits of such an architecture are to be realized.

We will refer to this fundamental concept as **architecture-based competitiveness** because we believe that all aspects of production must be "architected" for true agility to be achieved. If this is true, the attributes used to describe the agile systems architecture would be analogous to the attributes needed for other aspects of production.

(The rest of this page is intentionally blank)

The table below is a partial working profile of agile system architecture attributes for information systems with examples of analogues in other "systems" in the production environment. Readers are encouraged to place their own production models into this framework as an illuminating exercise. Though more tuning on this initial tool concept is required and encouraged, the tool has already been useful in exploring, explaining, and analyzing agility in production environments.

Partial Working Profile of Agile Production "System" Attributes Examples

	Information Automation	Organization	Machine/ Process	Product Design	Supply Chain
Peer Communication	Message-based, distributed, local preferences, location independent addressing.	Flat organizations, bossless teams.	Flexible transport, minimal impact of transport.	"Flat" product designs (non-hierarchical structures).	Transparent internal/external supply sources, virtual corps, EDI, Factory America Net
Encapsulation/ Modularity	Systems as interacting collections of "opaque" encapsulated modules.	User skill profiles, business process models.	Modular mechanisms & tooling, fully modeled machine capabilities & interfaces.	Modular product architecture, fully modeled component functionality & interfaces, no single point of failure.	Fully modeled supplier capability profiles.
Loose Coupling	Real-time, late-binding dynamic confederations, messaging between modules.	Shared specialists, process oriented, not functionally oriented.	Asynchronous flow, virtual cells, batch chemical processing.	Generic kernel products & platforms, late/local value add.	Many preferred suppliers (bid as you go).
Distributed Control Function	Functional decisions made at point of knowledge, distributed planning and scheduling.	Empowered workers, with accountable performance measures.	In-process gauging, adaptable processes at point of value add.	Intelligence throughout product, escort data through production.	Integrated free market model, contract/bidding/ teaming mechanisms.
Redundancy	No confusion among identical modules.	Cross functional workers	Multi-function/ multi-spindle machines	Multi-purpose components.	Second source suppliers.
Scalability	Fractal recursion, dynamic module population, transparent adds & cuts.	Continuous learning, rapid response teams.	Machines & systems with surge capacity, plug upgradable & reconfigurable subsystems.	Field upgradable designs, reusable product platforms.	Fine-grain supplier community.

Acknowledgements

Authors:

Steve Benson FACT, Inc., 508-470-0852
Rick Dove Paradigm Shift International, 510-652-7528
Joe Kann Allen-Bradley, 414-382-2252

Principal Collaborators:

Curt Holmes Martin Marietta Energy Systems, 615-574-3247
Vernon Lott Texas Instruments, 214-956-6843
John Oleson Dow Corning Corporation, 517-496-4516
Sal Scaringella United Technologies, Norden Systems, 203-852-5179
Gilbert Staffend Allied Signal Automotive, 313-827-5012
Keith Jessen Rockwell International, Collins, 319-395-5544
Raja Seshadri Honeywell, 612-785-4050
Mike Wald Alcatel Network Systems, 214-996-5293

Reviewers, Other Contributors, and Focus Group Members:

Clint Ackerman Digital Equipment Corp, 508-635-7758
Len Allgaier General Motors, 313-947-0686
Karl Burr Electronic Data Systems, 313-524-8371
Mike D'Orlando United Technologies, Norden Systems, 203-852-7871
Patrick Eicker Sandia Laboratories, 505-844-5827
Mike Griesmeyer Sandia Laboratories, 505-844-0282
Robin Hale Rockwell International, Rocketdyne, 818-710-4345
Al Hrncir Texas Instruments, 214-956-6469
John Kohls Institute of Advanced Manufacturing Sciences, 513-948-2111
Richard Neal Martin Marietta Energy Systems, 615-574-1862
Ray Patterson Department of Defense, Army, 803-760-4344
Frank Plonka Chrysler, 313-577-484
Deborah Reich Department of Labor, 202-219-5769
Frank Reynolds Eastman Kodak, 716-477-6187
Dave Standen Chrysler, 313-956-7478
Stephanie Swirsky Department of Labor, 202-219-5769
Orapong Thien-Ngern Lehigh University - Iacocca Institute, 215-758-5517
Jerry Tuschner Martin Marietta Missile Systems, 407-356-3973
Gary Vasilash Gardner Publications, Production Magazine, 313-473-0800
Bill Wadsworth Wadsworth Consulting, 215-356-3361
Curt Ward Raychem, 510-795-8997
Robert Woelfling Hershey Chocolate, 717-534-6400