This material is based upon work supported by the National Science Foundation under Grant No. DDM-9320951. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

Agility Engineering: Lego Lessons

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

Remember the child's round-peg, square-hole hammer toy? It had a framework with 6 or 8 uniquely shaped holes and a set of individually shaped wooden pegs. The trick was to integrate each of the pegs into a completed system by finding its uniquely compatible hole. These toys taught us valuable lessons about compatibility.

A more valuable lesson might have been about incompatibility, however. The framework had a fixed number of holes that demanded filling. A missing peg rendered the system incomplete. Spare pegs could not be bought separately, and trying to replace a lost peg with one from a friend's set generally found a different peg geometry.

Contrast that system with the Legos that younger generations are growing up with. The framework has a simple repetition of identical sockets on a standard grid pattern, and can be extended indefinitely by simply attaching additional framework sheets together. The modules come in various simple forms, all with an identical socket structure. Macro-modules can be assembled from basic pieces and replicated as often as needed to build or expand complex systems quickly. Losing a few pieces is hardly noticeable. The framework is so simple that compatible modules from competitors are readily available with special characteristics and pricing advantages. And the observed useful lifetime of the reconfigurable Lego set far exceeds the peg-pounder.

"We find less use for the MBA on board and more need for a new MBE - the Master in Business Engineering."

Legos even eclipsed Erector Sets. Though you could build almost anything with an Erector Set, without a framework every project was a custom construction effort that consumed too much time in the piece-interfacing activity.

A design strategy of reusable modules, reconfigurable within a scalable framework (RRS - introduced in a prior essay) can engineer Agility into a wide variety of Agile systems. Framework and module are the two distinct key objects here, and reconfigurableness is the key desirable characteristic. Though this may be oversimplified for some, and too mechanistic for others, it is a useful way to begin examining our business Agility-engineering task.

The accompanying table shows a cross section of enterprise elements, and suggests how specific systems within these elements might be viewed as modules and frameworks. Simply looking at these systems as modules and frameworks does not make them Agile of course. From this viewpoint, however, we can engineer the nature of the modules and frameworks to give us the desired reconfigurableness.

Certain corporations in Japan have put these concepts to fruitful technical use. The "software factory" is a Japanese approach to creating very large software systems in very short times, at very low costs, and with a remarkable lack of "bugs" (see Japan's Software Factories: A Challenge to US Management, Michael A. Cusomano, Oxford Press, 1991). They do this by constructing new systems from previously used and proven modules that are readily drawn from a well-maintained pool. They have established a standard framework so that all modules are plug compatible. Though they have not employed object-oriented programming environments, they build their modules as stand-alone encapsulated units to minimize side-effects when modules are combined in new systems. Though American programmers scorn the approach as non-creative and lacking in elegance, the fact is that these "engineered" systems get up and running bug free to satisfy a need in a fraction of the cost and time of traditional hand crafted systems.

Closer to home, Sun Microsystems describes their

SAMPLE SYSTEMS VIEWED AS MODULES AND FRAMEWORKS

Enterprise Element	Specific System	Case	Modules	Framework
Organization	Cross Functional Teams	Xerox*	Multi-Skilled Workers	Labor Contract
People	Learning	J. Doe	Skills	Prior Knowledge/Experience
Procedure	Dual-Use Accounting	Defense Contractor	Data Capture Software	MIS Legacy Wrapper
Information System	Order Entry	Rover*	Application Programs	Inter-Module Message Protocol
Control System	Adaptive Feedback	Saginaw Machine*	Sensors & Machines	Generic Machine Model
Plant Facility	Reconfigurable Factory	Texas Instruments*	Workstations	Physical Facility
Material Handling	Global JIT Production	Global Transpark	Planes/Trains/Ships/Trucks	Operating Contracts
Production Process	Flexible Machining	Saturn Corp*	LeBlond Makino Equipment	LeBlond Makino Standards
Production Equipment	Modular Fixtures	Watervleit*	Clamps/Components	Fixture Base Plate
Change-Over Process	Electronic Assembly	Solectron*	Assemblers/Machines	Culture/Information-System
Supply-Chain	Pre-Qualified Pool	Collins Avionics*	Component Suppliers	4.5 Sigma Qualification Program
Distribution-Chain	Brand Market	General Motors	Dealers/Customers	Product Positioning

*For case details see "1994 Best Agile Practice Reference Base", Agility Forum, 1995.

This material is based upon work supported by the National Science Foundation under Grant No. DDM-9320951. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

corporate "strategy for change" based on five S's: Small, Simple, Separable, Scalable, Stateless. Production Magazine's January 1995 issue reports this on page 61 as: "When you do a project, do one that's small and simple, one that has elements that can be removed and reused (i.e., separable) elsewhere (i.e., stateless) and expanded if necessary (i.e., scalable)". Sun is described as a $5-billion workstation manufacturer with 95% of its revenues from products less than 18 months old, with a 30% change in production techniques annually.

It should come as no surprise that Agile operating modes are beginning to emerge informally from many companies in the high-flux industries of electronics and software. As we try to formalize these understandings we look across all industries for emerging patterns like the RRS strategy, and we analyze these patterns for their contributions to change proficiency.

We recognize a need for change proficiently in a variety of ways -- all of which we loosely call reconfigurableness: increase in capacity (e.g. add more modules), increase in capability (e.g. add different modules), continuous improvement (e.g. modify a module), migration to a different operational foundation (e.g. modify the framework), creation of new capability (e.g. develop a new module type), reconfiguration of relationships (e.g. change the groupings of modules that interact directly with each other), recovery from a module dysfunction (e.g. reconfigure module relationships during operational performance), and respond to a performance-time surprise (e.g. change operating priorities).

The framework in our RRS strategy is basically a set of standards that defines a plug compatible environment. The concept of standards in the world of Agility is a two edged sword, and a central engineering issue. Standards are necessary to eliminate module interfacing as a problem; yet they will also determine and restrict the range of employable modules. So the framework must facilitate its own migration with time, and will probably benefit from simplicity as well.

If we provide embedded utility services within the framework we will benefit if they are implemented as modules - constructed as plug compatible self-contained modular units that provide fundamental common services to other modules. In the production environment MRP is a good example of something that should be implemented as a module rather than as an integral inseparable part of the framework. In the organizational environment teaming is a good example - especially as we learn about the pitfalls of early teaming approaches and recognize the need to evolve these systems over time.

There is much more to be said about design principles for RRS systems that will have to wait for another time. But we have said a little about frameworks, so we will also say a little about modules.

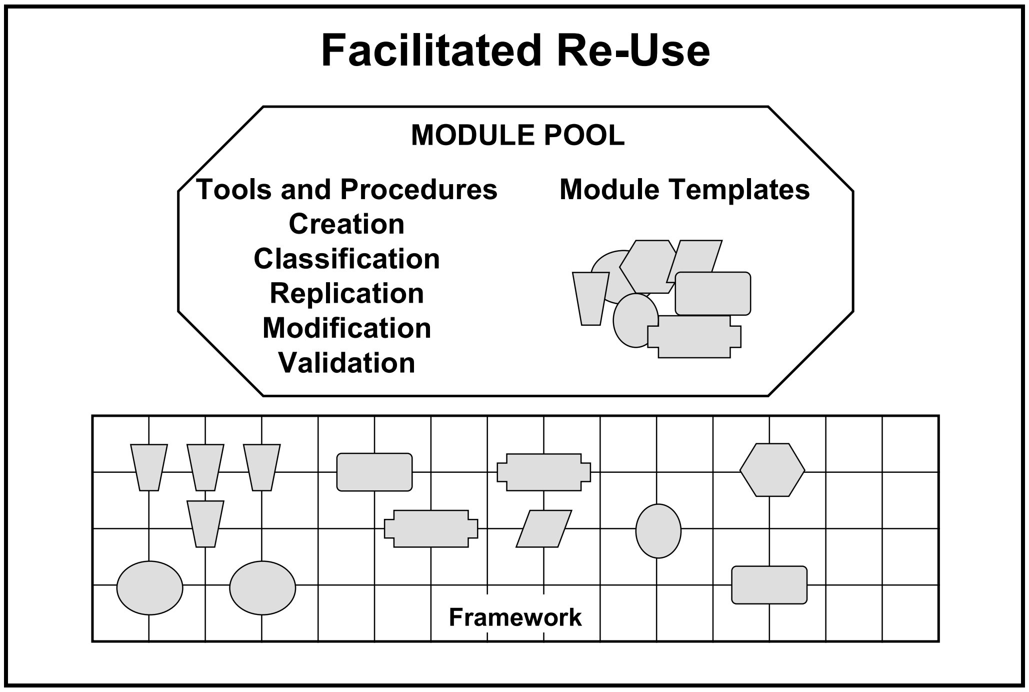
A key issue with modules is "facilitated" reusability. Modularity by itself does not provide the degree of reusability we look for. A support capability is needed that facilitates the creation, replication, and modification of modules. This will include some sort of repository for unused modules or templates for reproducing them; procedures for cataloging and finding modules; and tools for replicating, modifying, creating, and testing modules. This support capability is the engine of Agility, and the domain of the business engineer. And it becomes the seat of organizational learning and knowledge

Westinghouse Electric and AT&T have both created a corporate consulting pool containing a wealth of proven skilled personnel. These pools were formed initially when downsizing in various business units liberated valued competencies and skills that the corporation was afraid to lose. The concept in both companies has since evolved into a valued resource that can provide a "plug compatible module" with immediate effectiveness to any hot spot in the organization. The framework for these modules is best described as the corporate culture.

With the increasing complexity of global economic systems and business environments, the metaphor of management at the helm is passé. Piloting a ship through troubled waters is the wrong picture when even the waters are changing; demanding that the ship change accordingly as it goes. We find less use for the MBA on board and more need for a new MBE - the Master in Business Engineering.

We have covered a lot of ground in a small space here and will plumb the depths with a more narrow focus in later segments. In the meantime, the Agility Forum (610-758-5510) has published an initial "Best Agile Practice Reference Base" that can shed more light on this area with industrial examples.

Resource management is the enablement foundation for Agile enterprise - it is what change proficiency is all about - and it is what we have been focusing on in this series. By itself it does not guarantee an Agile enterprise, which also requires Agile opportunity management and Agile innovation management - concepts we will put in perspective at another time.


Facilitated Re-Use
MODULE POOL
Tools and Procedures
Creation
Classification
Replication
Modification
Validation
Module Templates
Framework