Principles of Evolutionary Architecture & Design for Agile Project Management

by Dr. David F. Rico, PMP, ACP, CSM

Large scale change of enterprise level architecture and infrastructure presents a challenge, especially in today's networked world. The challenges of today’s enterprise information systems are somewhat unique, compared to when the electronic computer emerged nearly 70 years ago. An early challenge was to stabilize computer technology in the first decade or so. Once it was stabilized, the next challenge was to build useful, large-scale automated systems, such as weapons, corporate networks, and business automation functions. Standardizing programming languages was a major concern during this era.

Commercialization of information systems such as mainframe computers, software applications, and even networks began to emerge in the 1960s. Several challenges faced these early adopters of information system technology. Skilled professionals and especially managers were in short supply. The management and technical disciplines were just emerging (i.e., project management, systems development, software development, etc.). Early projects were complex requiring years, thousands of people, and billions of dollars. The terms management crisis and software crisis emerged with respect to information systems.

The first 25 years spawned at least two major phenomena with respect to information systems. The first was the waterfall life cycle. That is, the notion that systems development is a linear, sequential process consisting of planning, analysis, design, development, and testing. This notion or theory is still pervasive today, even in the early 21st century. The second is that information system and network architectures of large enterprises and organizations became amalgamations of incremental changes over long periods of time. They are very sensitive, fragile, and subject to breakage with the slightest perturbation.

Numerous firms specialize in helping monolithic enterprises design large, scalable, available, and reliable networks. The problem is that most enterprises are burdened by their legacy infrastructures. Furthermore, they are unable or simply unwilling to change. In other words, enterprises don't mind if you diagnose and optimize the performance of their existing information systems and networks. However, they don't have the resources or patience to design a new infrastructure from scratch. Very little power, space, and cooling is available and many enterprises simply do not allow any new data centers.

What we're talking about is the difference between radical vs. incremental change. Radical change is expensive, long, difficult, and counterproductive. Michael Hammer urged us to "Obliterate it, not automate it!" in the 1990s. However, today, organizational psychologists realize that radical change is counterproductive. It is traumatic and results in good old resistance to change. Therefore, modern change experts advise us that iterative and incremental change is better and more successful. That is, evolutionary change is easier, cheaper, less risky, faster, easier to validate, and involves less resistance-to-change.

Unfortunately, information system and network designers have believed in designing the system right the first time to last for 100 years (at any cost). Then, of course, traditionalists believe the only remaining challenge is to resist any changes to the basic architecture, design, and implementation. Traditional project management and systems development paradigms have evolved to support this low-cost-of-change paradigm. That is, lock down the scope, WBS, schedule, cost, etc., up front and then prevent changes to bring the project to completion in within a 5% to 10% margin of error.

There are too many things wrong with traditional project management and systems development. It's impossible to know 100% of the scope, what you do document is wrong, it results in too much waste and security vulnerabilities, and nothing is prioritized so no business value is obtained. Worse yet, when something new of genuine value is identified, it is rejected, because it is out-of-scope. Therefore, the traditional paradigm is double trouble. No initial value and no late value are ever allowed. Traditional project management and systems development bred a culture of naysayers rather than innovators.

Technology and innovation researchers realize that true product and service requirements exist as hidden, inexpressible needs. That is, they exist as tacit, unspoken, or unwritten requirements. Furthermore, they cannot be elicited, coerced, or teased out of customers by traditional means no matter how hard one tries, how much time is spent, or how many resources are applied. This is a problem for traditional project management and systems development paradigms, whose basic premise is that 100% of the scope can be captured in the form of a WBS and meticulously tracked using earned value management.

Less than 30% of a customer's initial requirements can be captured in the form of written documentation. Historically, this dilemma has created several problems. First, project analysts believe 100% of the customer's requirements are knowable. Second, the analysts will document more requirements than are possibly known. Remember Parkinson's Law, "Work expands so as to fill the time available for its completion." In traditional paradigms, the number of requirements artificially expands to fill the amount of space and documentation allocated and allowed for their capture, storage, analysis, and management.

However, creating artificial requirements in traditional paradigms causes a host of related issues. The artificial requirements are generally wrong, do not reflect the customer's needs, and cost more than what is necessary to produce. This usually leads to cost and schedule overruns, program escalation, and eventually cancellation. From a mathematical standpoint, this causes delays, extended wait times, and unmanageably long queues. Furthermore, this also leads designers to create larger, wasteful, and defect-ridden architectures than what are necessary, and results in larger attack surfaces vulnerable to security incidents.

Enter agile project management. In it, there is a concept called "evolutionary systems/software development," "evolutionary architecture and design," or simply "refactoring." It is really not new; it comes from the old plan-do-check-act (PDCA)-based

TQM paradigms. Both lean and agile paradigms contain a major pillar or value called "continuous improvement" Some even say this is the most important aspect of any value, principle, technical practice, or tool. It is more of an organizational vs. technical or individual mindset. It simply involves creating innovatively new solutions for each problem encountered.

Apart from W. Edwards Deming and Joseph M. Juran, it can even be traced to historical Japanese culture, Kaizen, and the Toyota Production System, where perfection is obtained from a never-ending process of perpetual refinement. In this world view, continuous improvement represents the never-ending pursuit of perfection and process improvement is regarded or even revered as a journey rather than a destination. One simply cannot put a journey without beginning or end on a Gantt chart replete with milestones, deliverables, and constraints, and track its completion using earned value management.

Good Japanese manufacturers have known this for years. Process and product architectures are constantly reengineered and simplified with every product revision. Fewer moving parts in both the manufacturing process and product architectures do wonders for business success. It reduces cost, waste, and defects, and increases quality, reliability, speed, and customer satisfaction. We say this because not all Japanese firms are created equal. That is, there are some individuals, firms, and institutions that embrace these principles rather well (i.e., Toyota, NEC, and a few other well-regarded firms).

NEC's plasma displays are an excellent example. In the 1980s, the processes were so complex that the manufacturing lines had low yields, high defects, and the costs of individual plasma displays were very high. Now that manufacturing process and product architectures were slowly reengineered with each successive generation, costs are low, yields are high, and reliability is off the scales. In spite of Japan's and NEC's prowess with regards to continuous improvement as a means of developing near-perfect products and services, many of the finest consumer electronics are now outsourced to Korea and China.

On a side note, cancer survival rates are higher in cities where doctors are more willing to try experimental treatments than in cities locked into 50 year old cancer treatment regimens (i.e., the cure for cancer is not a singular miracle drug, but the willingness to change, try new things, and attempt a never ending variety of solutions). Recent studies have shown that patients in the Pacific Northwest have some of the highest cancer survival rates in the world. Many attribute this to their culture of risk-taking, when it comes to the application of experimental treatments, drugs, and other techniques.

Notice the term "experimental." This is no accident as agile project management is often compared to a mindset, culture, and process of experimentation. That is, the willingness to take risks, attempt multiple solutions using rapid experimentation and iteration, and even rewarding people for failure. Traditional paradigms reward project managers for completing projects on time, budget, and scope, while punishing those who cannot. Agile project management rewards people for taking risks, trying new things, and even failing more often than they succeed. "No risk, no reward" is often the mantra of agile methods.

One technique within agile methods for continuous improvement is called "refactoring." Others include iterations, continuous integration, standups, retrospectives, or other TQM, Six Sigma, and Lean tools. One must constantly reengineer, change, and simplify process and product architectures among successive iterations. Nothing stays the same, is sacred, or survives from iteration to iteration. If it can be reworked, simplified, and made better, then it can, will, and must be changed. This is true at the enterprise level and contrary to the traditional principle of "design-it-once-and-reuse-it-forever" to achieve system quality.

Empirical data shows that big up front design is impossible, expensive, and leads to failure. It also shows that little or no change also suffers from the same fate (i.e., adapt or die). More importantly, the cost of change is lower with "refactoring" and agile methods reduce overall costs, increase quality, and increase satisfaction with successive iterations. This is contrary to the belief that agile methods are a deliver-it-now-fix-it-later paradigm. This is somewhat of a misnomer, since traditional paradigms are a deliver-it-later-and-fix-it-if-you-still-have-the-time-money-and-patience paradigm.

The mantra of the 21st century should be "adapt, change, overcome, and iterate or die!" This is in contrast to the "prevent change or die" attitude that traditional paradigms have carved into our psyches over the last 50 years. Yes, continuous improvement and change is psychologically hard, organization change is very difficult, and the human race is not very good at change (to-date). However, we must break out of the traditional beliefs that change is bad and we must become masters of change as individuals, teams, groups, business units, enterprises, industry sectors, nations, and as a world community.

Our knowledge as a discipline with respect to the principles of "evolutionary project management, systems and software development, architecture, design, development, test, and evaluation" is meager. The principle of big up front architecture and design is still considered a best practice. It is the result of traditional project management and systems development methods based on the notion that 100% of the scope is predictable at the project's start. However, we're slowly coming to the realization that "successful architecture and design is an amalgamation of renovations over long periods of time!"

- Cathedrals: Many historical cathedrals are excellent examples of evolutionary design principles. They were envisioned by their creators to be functional places of worship as well as architectural wonders. They were often meant to be symbols of political, economic, and religious power by newly created nation-states. This presented many opportunities as well as problems. They were free from political bureaucracy hindering their design and construction. However, newly forming nation states had other problems. They were often at war with indigenous populations as well as global powers for regional control. Financial resources were in short supply, trade routes were hazardous at best, and the industrial revolution hadn't quite taken root yet, so much needed resources weren't available on short demand. Oftentimes, their creators only had enough resources to retain the services of a renowned architect and establish the foundation and walls. Other major features were added in the ensuing decades and even centuries. These included roofs, windows, furnishings, decorations, adornments, artwork, musical accompaniments, and bells. Time and politics are cruel and take a toll on fragile items such as wood, glass, furnishings, and other perishable items. However, they weren't built as a part of a one-time, top-down, big up front blue print or design. Part of the problem is the concept of linear, waterfall-based project management and systems development. The original plans are so grandiose that they require far more resources to implement than are available at the time. However, it's not just the immense cost of implementing the original requirements, but also the immense cost of requirements expansion during the original building phase. In operations research theory, too much detail causes elongated delays, wait times, and queues, more of which leads to high initial project failure. In other words, the immense structures cannot possibly be completed on the original schedule. Large structural projects often go unfinished for decades or even centuries. This opens the door to evolutionary design principles. That is, large structures evolve over time until they can become operational, functional, or even useful to sets of stakeholders, customers, and market participants. Prime examples of this phenomenon include skyscrapers, museums, school buildings, university campuses, sprawling government agency campuses, military bases, airports, state and federal capital buildings, hotels, bridges, roads, highways, waterways, canals, cities, etc. Tourists will often flock to see these modern day wonders of the world, once these historical structures reach maturity over many decades and even centuries. Therefore, large-scale cathedrals become amalgamations of renovations over long periods of time. Today, North American cathedrals are some of the most beautiful structures in existence.

- Military Aircraft. Most if not all military fighter jets, bombers, transports, and helicopters are also excellent examples of evolutionary design principles. In actuality, the number of new military aircraft is very low, because of their immense cost and risk of development. The length of time between completely new airframes is often measured in decades. This creates gaps in mission capabilities over large spanses of time. Therefore, new airframes are proposed to fill these mission gaps and significant goals, objectives, and requirements called key performance parameters are established for future aircraft. These often come in the form of fly higher, fly faster, have longer ranges, carry heavier payloads, or have lower radar cross sections. However, new aircraft acquisitions are often planned as traditional, linear top-down systems development projects. Mission planning takes years, requirements development takes decades, and prohibitively expensive one-off prototypes are created which bankrupt the creators of competing designs. A single firm is usually chosen to undertake a decade-long detailed systems development and design phase. Finally, manufacturing begins two or three decades after inception, resulting in what is known as "reduced delivery order quantities" (i.e., fewer aircraft units are created, because of the immense cost and schedule overruns experienced often due to over-engineering). However, that's when the principles of evolutionary design are just getting started. Oftentimes, the initial operating capabilities of the aircraft's subsystems are far below customer expectations and the delivered units fail to satisfy even basic expectations. Worse yet, half of the initial units are often cannibalized to keep the other half in the air. Therefore, the customers begin a series of evolutionary increments called "block upgrades." Groups of subsystems are enhanced or replaced in a series of 20 to 30 block upgrades over a period of 10 to 20 years. Entire airframes, engines, mission computers, cockpits and heads up displays, radars, communications, munitions and weapons, etc., are refined, repaired, reengineered, and sometimes completely replaced. Magic happens after about 20 block upgrades or so. To the layperson, the external appearance and architecture looks identical to the initial rendering of the 40 year old artist's conception. However, the internal capabilities are completely different and do not even resemble the detailed designs created during the so-called "systems engineering" phase. Better yet, military aircraft that have undergone 20 or more block upgrades finally end up exceeding the original customer expectations. Therefore, military aircraft have become amalgamations of renovations over long periods of time. Today, 20, 30, 40, and even 50 year old military aircraft are some of the most feared weapon systems on the planet.

- Internet Technologies. Internet technologies are some of the best examples of evolutionary design principles in the early 21st century. Electronic computers emerged in the 1940s and practical computer applications emerged in the 1950s. However, the 1960s marked the creation of the first commercialized mass market mainframe computers. Mainframe operating systems required thousands of people, decades, and billions of dollars to develop. Software applications sprang up overnight for early mainframes and the U.S. Justice department created the legal foundation for the commercial shrink-wrapped software industry. The 1970s were marked by increasing commoditization of computers, operating systems, and their applications. This included miniaturization, increases in processing speed, and radical declines in costs. Although the microcomputer was born in the mid 1970s, it really didn't take shape as a major platform until the 1980s. Unlike hardware, software didn't require large capital investments in factories, manufacturing lines, and raw materials. Large computer programs consisting of millions and even billions of interacting algorithms now fit on the head of a pin. However, early computer programs were shipped on hard disks, reeling tape, removable media such as floppy disks, CDs, and DVDs. Oftentimes, these media were produced and sold by the millions and the pressures to get the software right the first time were immense. Furthermore, large organizations installed these software images on vast quantities of networks, computer servers, and desktops throughout their enterprises. So the pressure to get software right the first time was twofold (i.e., reduce distribution costs and minimize enterprise management costs). However, the popularization of the Internet in the form of the World Wide Web in the 1990s changed all of this. Business and personal computers were now connected on a single world-wide network in a seemingly open fashion. Software could now be distributed and installed electronically in seconds and minutes, instead of burning millions of CDs or deploying a small army of system operators to update enterprise systems. Software is evolved one bit at a time, every few seconds, 24 hours a day, and distributed to the more than 3 billion Internet users rather than use linear, waterfall systems development lifecycles. There are about five Internet enabled devices for each user, so trillions of software updates take place every few seconds. Software is even evolved and zapped into deep space probes, satellites, space stations, and other aircraft and weapon systems. Therefore, Internet technologies are becoming amalgamations of renovations over long periods of time. Today, Internet firms are the most successful businesses in the history of the world with double digit annual growth and trillion dollar market capitalization.

The principles of evolutionary architecture and design involve at least three major challenges. The first is to create methods to help project managers and engineers understand viable alternatives to 19th and 20th century manufacturing paradigms. The second is to promote the use of thin-client business models and technologies and continue moving away from brick-and-mortar thick-client business models. The third is to change behavior and encourage people to fail fast, change rapidly, and perfect solutions by soliciting market feedback from a rapid succession of evolutionary designs over long periods of time.

- Methodological: The creation of evolutionary paradigms that acknowledge linear, waterfall-based paradigms do not work, big up front design was, is, and always will be a convenient adolescent myth, and that complex systems of all sizes must evolve over long periods of time in order to succeed (even if it is an inconvenient truth for many to accept).
- Technological: The exploitation of evolutionary technologies that acknowledge big up front “do it right the first time” paradigms are a myth, and the most cost-effective way to rapidly evolve superior technological solutions that satisfy customers and lead to optimal market capitalization is with thin client business models, technologies, and platforms.
- Psychological: The acknowledgement that traditional paradigms are intuitive, instinctual, and easy to grasp models carved into our psyches for 50 years, which must be patiently overcome by conditioning people to think out-of-the-box and understand how evolutionary principles mimic nature, history, and success with respect to architecture and design.

Our focus is on the methodological aspects of evolutionary architecture and design. Therefore, let us be bold and begin a dialogue and future conversation concerning its tenets. That is, let’s begin to identify a short list of evolutionary design principles. An evolutionary design should have unique missions, goals, and objectives; timeless architectural vision; broad architectural framework; strong and enduring foundation; modular capabilities and functions; basic initial operating capabilities, and multiple design and capability upgrades. (An unspoken principle is to dispense with big up front design.)

- Unique Missions, Goals, & Objectives: Establish highly-unique missions, goals, and objectives that are lofty, ambitious, and provide an order-of-magnitude improvement over prior products, services, and technologies.
- Timeless Architectural Vision: Develop a timeless architectural vision in which the idea or concept will stand the test of time (i.e., fast lightweight aerodynamic airframe, iconic structure symbolizing political and religious power, etc.).
- Broad Architectural Framework: Design a broad architectural framework with large margins that allow individual capabilities to evolve sometimes radically (i.e., a flexible container that can be repurposed as the mission changes).
- Strong and Enduring Foundation: Build a strong and enduring foundation, which although incomplete, establishes an imperishable, indestructible, and long-lasting structure that can survive neglect, change, calamity, and even attack.
- Modular Capabilities and Functions: Allow modular capabilities and functions to be added, completed, or evolved over long periods of time (or to be repaired, replaced, and enhanced in response to the ravages of time, wear, and tear).
- Basic Initial Operating Capabilities: Implement basic initial operating capabilities to demonstrate the functions of the design (while allowing an optimal whole to emerge as system performance is enhanced over long periods of time).
- Multiple Design and Capability Upgrades: Plan for multiple design and capability upgrades in an incremental, iterative, and evolutionary fashion (to place dependable operating capabilities in customer hands as early as possible).

A key element of the paradigm of evolutionary architecture and design is the concept of emergence, emergent design, and emergent properties. Its principles also contain elements of top-down, blueprint architecture and design practices. However, the key difference is to strike a balance between rapidly and cost-effectively producing an enduring architectural vision, while allowing individual capabilities to emerge over long time periods in an almost unplanned fashion. A key idea is that architectures and designs need to evolve in order to respond to changing mission needs and rapid technological evolution.

Proponents of traditional methods often point to historical structures as proof of linear, waterfall big up front design thinking. They point to Egyptian pyramids, the Great Wall of China, Roman aqueducts and roads, temples and palaces, coliseums and other public structures, towers and skyscrapers, bridges and dams, railroads and highways, etc. However, they are missing the simple point that these structures took decades and centuries to evolve into the enduring structures, icons, and symbols they are today. These structures do epitomize architecture, but they represent examples of evolutionary vs. big up front design.

We must design our enterprises to accept change, refinement, evolution, adaptation, etc., as a way of life, and realize change is both psycho-sociological as well as technological phenomenon. That is, we need agile enterprise cultures as well as agile information systems, infrastructures, and technologies. We're getting there a little bit at a time. However, we must be patient, and it will come. On a minor note, traditional linear project management and systems development paradigms, standards, values, principles, practices, and tools are making a comeback in the early 21st century. It's always darkest before the dawn.

Dr. Rico has been a leader in support of major U.S. gov’t agencies for 25 years. He’s led many Cloud, Lean, Agile, SOA, Web Services, Six Sigma, FOSS, ISO 9001, CMMI, and SW-CMM projects. He specializes in IT investment analysis, portfolio valuation, and organizational change. He’s been an international keynote speaker, presented at leading conferences, written six textbooks, and published numerous articles. He’s also a frequent PMI, APLN, INCOSE, SPIN, and conference speaker (http://davidfrico.com).