The History, Evolution and Emergence of Agile Project Management Frameworks

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

What is agile project management and what's its origin? Isn't agile project management an oxymoron? Aren't agile methods the antithesis of traditionalism? Didn't agile methods do away with project management, project managers, and traditional methods? Isn't project management a linear, sequential, top-down, rigid, autocratic command and control paradigm? Isn't project management rooted in Taylorism, Scientific Management, Fordism, mass production, manufacturing, rationalism, and reductionism? Isn't project management better suited for highly-repetitive service operations and construction projects?

Don't agile methods address the challenges of 21st century systems (i.e., high-risk, time-sensitive, R&D-oriented, new product and service development projects)? Aren't today's projects characterized by demanding customers, fast-changing market conditions, and development of highly-exploratory, technology-intensive complex adaptive systems? Aren't agile methods designed for information age knowledge workers empowered to form self organizing teams and adapt unique work systems for each problem they encounter? If all of this is true, where does the notion of agile project management fit in?

Let's start with the definition of traditional projects and project management. A project is defined as "A temporary endeavor undertaken to create a unique product, service, or result." Project management is defined as, "The application of knowledge, skills, tools, and techniques to project activities to meet the project requirements." Today, it is characterized by nine major knowledge areas (i.e., project integration, scope, time, cost, quality, human resource, communications, risk, and procurement management). Its principles and practices are exemplified by de facto standards such as the PMBoK, CMMI, and SEBoK.

Project management has its roots in the origin of civilization itself. One can certainly see how it was applied to the Tower of Babel, Pyramids, Great Wall of China, and Roman Coliseum. Other projects include Railroads, Panama Canal, Hoover Dam, Eiffel Tower, and Empire State Building. Some attribute it to Barcharts (1765), Harmonograms (1896), Gantt Charts (1916), Flowlines (1930), and Line of Balance (1941). Others attribute it to Critical Path (1957), Program Evaluation Review Technique (1958), Work Breakdown Structure (1962), Cost/Schedule Control System (1967), and Iron Triangle (1969).

Agile methods were direct spinoffs of software methods from the 1980s, namely Joint Application Design (1986), Rapid Systems Development (1987), and Rapid Application Development (1991). However, they were rooted in earlier paradigms, such as Total Quality Management (1984), New Product Development Game (1986), Agile Leadership (1989), Agile Manufacturing (1994), and Agile Organizations (1996). The euphoria surrounding businesses as complex adaptive systems was also a major influence, namely Ecosystems (1995), Adaptive Learning (1996), Structured Chaos (1997), etc.

Agile methods formally began in the 1990s with Crystal (1991), Scrum (1993), Dynamic Systems Development (1994), Synch-n-Stabilize (1995), Feature Driven Development (1996), Judo Strategy (1997), and Internet Time (1998). Other agile methods included New Development Rhythm (1989), Adaptive Software Development (1999), Open Source Software Development (1999), Lean Development (2003), and Agile Unified Process (2005). However, the popularity of Extreme Programming (1999) was the singular event leading to the unprecedented success of agile methods by the early 2000s.

The Oil Shock of 1973 undermined the field of strategic planning. "Necessity is the mother of invention" and new theories, paradigms, and approaches arose. These included adaptive, incremental, contingent, and participative forecasting, budgeting, and planning. Henry Mintzberg's 1994 book, "Rise and Fall of Strategic Planning" heralded the demise of top-down blueprint and comprehensive rationalistic planning. The confluence of project management, strategic planning, manufacturing, TQM, software, and complex adaptive system theories established the basis for agile project management as we know it today.

So, what is the definition of agile project management (or how is it defined)? There is no standard definition of agile project management. Traditional methods presume that 100% of a project scope is predictable and can be successfully delivered with rigid processes and documentation given enough time, patience, and resources. Agile methods realize 70% to 80% customer needs are hidden, tacit, and inexpressible. Therefore, user needs must be teased out a little bit at a time with human-intensive interactions, lightweight flexible process and product technologies, and near-term, time-based sense-and-response probes.

Year	Source	Agile Project Management Definition
2004	Alex Chin	Practical and repeatable method for building sound, yet flexible project processes in environments exhibiting high internal and external uncertainty, need for unique expertise, and levels of urgency.
2004	Doug DeCarlo	Art and science of facilitating and managing flow of thoughts, emotions, and interactions to produce value outcomes under turbulent and complex conditions (i.e., high speed, change, uncertainty, and stress).
2005	Sanjiv Augustine	Work of energizing, empowering, and enabling project teams to rapidly and reliably deliver business value by engaging customers and continuously learning and adapting to changing needs and environment.
2010	Robert Wysocki	Industrial model designed for adaptive projects in order to deliver maximum business value to clients from every iteration cycle within limits of client imposed time and cost constraints.
2010	Jim Highsmith	Transformational set of principles, practices, and performance measures enabling project managers to catch up with realities of modern product development by being fast, flexible, and customer-responsive.
2012	Mark Layton	Style focusing on early delivery of business value, continuous improvement of project processes and products, scope flexibility, team input, and delivering well-tested products reflecting customer needs.

Agile methods appeared vastly different to the casual observer. However, they had more in-common than anyone realized. The creators of agile methods gathered together in 2001 to explore these commonalities. They erected a website containing the Agile Manifesto, which still stands today. It contains four major values representing the theoretical tenets of all agile methods (i.e., collaboration, teamwork, iteration, and adaptability). The notion of Agile Project Management emerged circa 2003, the first books appeared in 2004, and the Declaration of Interdependence was formed in 2005 representing its values.

Value	Agile Project Management Values – Declaration of Interdependence
ROI	We increase return on investment by making continuous flow of value our focus.
Collaboration	We deliver reliable results by engaging customers in frequent interactions and shared ownership.
Iteration	We expect uncertainty and manage for it through iterations, anticipation, and adaptation.
Environment	We unleash creativity and innovation by recognizing that individuals are the ultimate source of value, and creating an environment where they can make a difference.
Teamwork	We boost performance through group accountability for results and shared responsibility for team effectiveness.
Adaptability	We improve effectiveness and reliability through situationally specific strategies, processes and practices.

Early agile methods had their own built-in agile or adaptive project management frameworks, namely Scrum and Extreme Programming (XP). There is some debate about which came first, "the chicken or the egg" (with respect to Scrum and XP)? Scrum appeared earlier, and XP borrowed some concepts from it as well as most of its practices from the 1980s (i.e., use of CRC cards, which we now know as user stories, an instrumental tool in most agile project management models). However, XP was responsible for laying down the most comprehensive agile or adaptive project framework in 1998 and 1999.

XP's agile project management model was called "Release Planning." Its rudimentary principles were published in 1999 and its principles and practices were singled out for formalization in yet another textbook in 2001. That is, XP's creators did the impossible. They separated the inseparable (i.e., excised agile project management from technical agile practices). This was a feat not as clearly, concisely, or elegantly done in early versions of Scrum and other agile methods. It spawned a two-pronged, raging debate from the project management (PM), information technology (IT), and agile communities at-large.

The PM and IT communities heralded it as a major breakthrough in the formation and formalization of agile and adaptive project management models. Much earlier methods attempted to establish agile and adaptive frameworks, namely Iterative (1975), Evolutionary (1976), Incremental (1978), Stage Gate (1983), Spiral (1986), etc. However, Release Planning was the most credible agile and adaptive project management model to appear in 25 years. Agile purists resisted the formation of agile project management for fear of regressing into traditional methods by disempowering self-organizing teams.

Release Planning was the emancipation proclamation for PMs who wanted freedom from traditional methods. It kicked off a frenzy of methods, models, frameworks, and textbooks. These included Radical Project Management (2002), Extreme Project Management (2004), Managing Agile Projects (2004), Agile Project Management (2004), Agile Estimating and Planning (2006), Adaptive Project Framework (2010), Agile Software Requirements (2011), and Agile Project Management for Dummies (2012), which is quite good. Today, there are dozens of books on this topic (with more on the way every day).

We don't have the time or space to summarize every agile project management framework. So, let's examine some of the more well-formed ideas. We'll describe Scrum (Schwaber) used by over 70% of projects. We'll also look at XP (Beck), since it spawned many of today's ideas. Then, we'll examine Flexible Project Management (DeCarlo), Agile Leadership Model (Augustine), Agile Project Management (Highsmith), Adaptive Project Framework (Wysocki), and Scalable Delivery Model (Leffingwell). Note that Scrum was retrofitted with Release Planning in the 2000s (but is considered out-of-scope today).

Year	Model	Source	Major Agile Project Management Methods, Models, and Frameworks
2001	XP	Beck	Release planning (Stories, Scope, Velocity), Iteration Planning (Tasks, Schedule, Dev.).
2004	Scrum	Schwaber	Sprint Planning, Sprint (Dev.), Daily Scrum, Sprint Review, Sprint Retrospective.
2004	Flexible	DeCarlo	Visionate (Vision), Speculate (Plan), Innovate (Dev.), Reevaluate (Rev.), Disseminate (Dep.).
2005	Leadership	Augustine	Alignment (Teams, Vision), Emergence (Rules, Collaborate, Coach), Learning (Adaptation).
2010	Agile	Highsmith	Envision (Vision), Speculate (Release), Explore (Iterate), Launch (Deploy), Close (Doc.).
2010	Adaptive	Wysocki	Version Scope, Cycle Plan, Client Checkpoint, Post-Version Review.
2011	Scalable	Leffingwell	Portfolio (Vision, Architecture), Program (Product, Release), Team (Spikes, Iterations).

XP has user stories, release plans, technical tasks, and iterations. Scrum has product and sprint backlogs, daily standups, customer reviews, and retrospectives. Flexible has project visions, project plans, time-boxing, business justification, and deployment. Leadership has culture, visioning, self organization, coaching, and adaptation. Agile has visioning, release planning, iterative development, formalization, and closeout (with Scrum instruments). Adaptive is similar to Agile with lightweight traditional instruments. Scalable has visions, architectures, release plans, and iterations (heavyweight Scrum).

Agile and adaptive project management is clearly in its golden age. It is important to note that there is still a significantly large population of middle managers at the center of the bell curve. They stubbornly believe agile methods are an ill-conceived paradigm that fall short of "more disciplined" traditional paradigms. Their complaints are all too common and read like a extended criminal indictment (i.e., only for small software projects, undisciplined, no requirements, no plans and processes, no documentation, spaghetti code, low quality, not maintainable, not scalable, poor performance and security, etc.)

Conversely, agile methods are highly-disciplined, yet flexible for the ambiguity and uncertainty of today's projects. They include well-defined requirements (user stories); plans (release and iteration plans); just-in-time architectures, designs, and documentation; product assurance (workflow, configuration management, quality assurance, testing, certification, and security automation); etc. Agile methods result in 10 times greater quality at a fraction of the cost of traditional ones (and are adaptable like well-tailored suits, doing away with 80% of the risk, waste, and defects of unpredictable traditional methods).

Case Study – A Traditional vs. Agile Project

Traditional Project*. The goal was to develop a global enterprise repository. Traditional plans, requirements, architectures, designs, testing, and documentation were created using a waterfall model. A relational database management system (RDBMS) was selected along with a new computer system. It seemed like a relatively straightforward project. What could go wrong? The system was fielded when things went awry. The amount of global enterprise data exceeded the RDBMS capacity by 10 times. The project was cancelled after seven years, the firm was fired, and the technology abandoned.*

Agile Project*. The goal was to complete the unfinished global enterprise repository. This time, agile plans, requirements, architectures, designs, testing, and documentation were created using a flexible model. Exploratory iterations were used to investigate leading hardware and software technologies. Open source big data software was selected, along with a powerful, but inexpensive high-performance data center architecture. The hardware was procured in multiple iterations, software was loaded, data was collected, and the system was eventually scaled to store all of the global enterprise data.*

What's the difference*? The traditional project assumed to know 100% of the scope. Trust was placed in the process vs. humans. A big upfront design was created and expensive, incorrect technology was purchased early. Last minute, big bang integration triggered a 10 times cost and schedule overrun. The agile project didn't assume to know all of the scope. A disciplined process was used. Inexpensive technology was evaluated one iteration at a time until a solution was discovered. It had 10 times better cost efficiency and quality, because of right-sized processes, better decision making, and less waste.*

Today, agile methods are used by 80% to 90% of worldwide projects and have clearly "crossed the chasm." They are commonly used by big data firms (Google, Facebook, Yahoo, Amazon, etc.) and major global telecom giants (Nokia, Ericsson, British Tel., Siemens, etc.). They are also used by 60% to 70% of U.S. DoD projects (F−35, F−22, F-18, etc.), highly-regulated industries (DOE, FAA, VA, etc.), and medical device manufacturers (FDA Class III certified products). Agile methods have even penetrated traditional CMMI strongholds (India, China, Latin America, Africa, Middle East, etc.).

How and why did agile methods spread so quickly and completely in just a few years without public sector support (taxpayer funding)? Traditional paradigms from the 1950s (i.e., project management, systems engineering, etc.) took decades, billions of dollars, and many standards to achieve similar gains. On one hand, traditional methods formed a cultural foundation upon which agile methods rest, as well as the impetus for change themselves (i.e., they were too cumbersome and ineffective). The primary drivers of agile success are low barrier to entry (inexpensive) and utter simplicity (small conceptual footprint).

Let's examine this from an organizational change perspective. In Diffusion Theory (Rogers), all technologies go through five stages (i.e., Innovators, Early Adopters, Early Majority, Late Majority, and Laggards). Rogers recognized technologies take a long time to spread, just like traditional methods. In Chasm Theory (Moore), there is a gap between Early Adopters and Early Majority. Moore suggested that euphoric Early Adopters had to "hard-sell" the more skeptical Early Majority, to speed up the adoption process. Hard selling costs and benefits rarely succeed, which explains why traditional methods took so long.

The Satir Model (1991) also consists of five stages (i.e., Status Quo, Novelty, Chaos, Recuperation, and Revival). Satir realized complex Novelties create extended Chaos before Recuperation and Revival. This is why traditional methods took decades and billions of dollars to spread. Conversely, agile methods took about 2 to 3 years to spread globally. (Note. CMMI has 200 practices and 800 outputs, SEBoK has 100 activities and 200 outputs, and Scrum has 5 processes and 5 artifacts.) Leading change models, Drive (2009) and Switch (2010) corroborate the old adage, "the bigger you are, the harder you fall."

Pundits take various positions on traditional vs. agile methods. Some posit it as either-or (i.e., traditional or agile). Some pose it as disciplined (traditional) vs. undisciplined (agile). Agilists pose it as iterative (agile) vs. waterfall (traditional). Some have even posed it as scalable mission and safety critical systems (traditional) vs. small software prototypes (agile). These methods are rapidly converging. CMMI contains agile practices with more to come. PMI offers an agile certification. The systems engineering handbook accepts agile methods as viable alternatives and sugar-coats traditional practices with "agile frosting."

In 2002, Rob Thomsett wrote, "Traditional project management doesn't work anymore: it's inward-looking, static, and can't respond to rapid, constant change." In 2010, Bob Wysocki wrote, "The world of traditional project management belongs to yesterday, don't waste your time using traditional project management on 21st century projects." They were pointing to agile project management, which is a contemporary approach for managing the design, development, and delivery of ambiguously-defined high-technology products and services in a timely fashion that maximizes business value for its customers today.

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).