

32 PRINCIPLES AND PRACTICES FOR A HIGHLY-SUCCESSFUL AGILE CONTRACT STATEMENT-OF-WORK (SOW)

State-ment • of • Work (*stāt'mənt • ŭv • wûrk*). A statement of work (SOW) is a document written by an acquirer, typically a public sector or government organization, outlining all of the needs, requirements, or performance objectives for the design, development, evaluation, implementation, maintenance, operation, or retirement of a new product or service. SOWs typically accompany request for proposals (RFPs), which are short letters asking one or more suppliers to prepare an offer or proposal with respect to the requirements or needs of its associated SOW. RFPs and SOWs, especially in the public sector, are used for solicitation purposes (i.e., seeking innovative bids or proposals from suppliers to fulfill the needs of the acquirer's SOW). Typically, a period of time elapses to collect one or more supplier proposals in response to an RFP and SOW, an independent source selection committee (panel) is established to evaluate supplier proposals, and the source selection committee chooses one. Acquirers often look for best value, least cost, experience in their domain, or in some cases compatibility of the supplier's proposal to the acquirer's SOW. In the end, acquirers establish a contract with one or more suppliers to fulfill the needs of the SOW as it pertains to the supplier's proposal. That is, the proposal is what the acquirer contracts for, not the SOW, which is an acquisition instrument synthesized into a proposal which the acquirer accepts. For instance, an acquirer may ask for 100 computer servers for \$1 million, but a supplier may propose 125 computer servers for \$750,000, so the contract is for the latter vs. the former. SOWs have become too product or process specific, especially in the case of developing new products and services, suggest hundreds of dysfunctional contradictory traditional frameworks and product requirements, and directly lead to acquisition failure.

Ag-ile • State-ment • of • Work (*ăj'əl • stāt'mənt • ŭv • wûrk*). An agile statement of work (SOW) is an innovation, especially in the public sector. Agile SOWs which likewise accompany RFPs, are also process oriented, and suggest one or more agile methods for helping acquirers develop innovative new products and services. Agile SOWs come in a range of specificity, i.e., traditional SOW with some agile terminology, traditional SOW suggesting hybrid traditional and agile approach, lightweight agile SOWs suggesting many lean and agile methods with a smattering of traditional practices like integrated master schedules (IMSs), and purer agile SOWs suggesting a specific lean-agile methodology like Scrum, SAFe, or DevSecOps. The challenge with agile SOWs is that they are very few and far between (i.e., the majority of public sector SOWs are traditional in nature and may only have one or two agile terms among thousands of traditional ones). The dearth of agile SOWs signals to the supplier base that agile methods are simply unimportant, unnecessary, and not very useful. Still other SOWs have more agile terminology signaling a larger commitment to agile delivery methods but are heavier on traditional terms forcing suppliers to propose dysfunctional hybrid approaches. Even when SOWs are clearly agile in nature and it is clear the acquirer wishes the supplier to apply agile methods, they often add an appendix with two or three hundred traditional standards to which the supplier must be ready to apply, either signaling ignorance or lack of trust in agile delivery methods (i.e., break glass in case of emergency). While it is impressive that more and more public sector agencies are directly applying agile delivery methods, public sector RFPs and SOWs are in a sorry state of affairs, not only sending the wrong message, but, in the worst case, nonexistent, causing acquisition failure.

Ag-ile • Frame-work • State-ment • of • Work (*ăj'əl • frām'wûrk' • stāt'mənt • ŭv • wûrk*). An agile framework SOW is an innovation among public sector agencies recognizing the importance of coherent, proven agile frameworks so that larger teams of teams can help develop and deliver larger ecosystems of innovatively new products and services. Furthermore, agile framework SOWs tend to be purer agile SOWs asking suppliers to propose selecting and applying only one agile framework instead of a smorgasbord of hybrid traditional and agile methods. Likewise, they are often devoid of appendices with one or two hundred traditional standards to which a supplier must comply. Once again, this is a leap forward in public sector agile acquisitions, but also come with a few pitfalls. First of all, they are few and far between and are only seen in a small handful of the nearly 200 public sector agencies. Second, they sometimes misconstrue agile frameworks as top-down push driven approaches where the acquirer can merely jam a large business requirements specification into the agile framework's intake system, subsequently choking it to death. Third, and similarly to number two, acquirers believe they are the source of detailed requirements while agile teams exist merely to implement them. Fourth, agile framework SOWs are generally devoid of any lean-agile thinking principles, collaboration imperatives, or business experimentation approaches to allow new products and services to emerge and evolve naturally. Fifth, agile framework SOWs, like traditional ones, are too heavy on the product requirements, rather than serving as collaborative contracts where acquirers and suppliers can co-evolve innovatively new products and services together in unison. However, there seems to be a general disenchantment with agile frameworks just when they're more mature and useful than ever.

These definitions have pretty much laid out the challenges associated with public sector acquisitions and the roles of the SOW in helping shape acquisition success and failure. First of all, ditch the traditional SOW, especially if the goal of the acquirer is to develop innovatively new products and services. Second of all, ditch the hybrid traditional and agile SOWs, because the acquirer wants to dip their toes in the water with a smattering of agile methods. Third of all, ditch the almost-agile SOWs with long lists traditional standards which sends mixed messages to the supplier community. Fourth of all, ditch the agile SOWs with a clear high power-distance hierarchy between acquirers and suppliers, where acquirers will simply toss detailed integrated master schedules (IMSs), business requirements, and enterprise architectures over the wall and freeze agile intake systems in 10 seconds (forever). Fifth of all, begin assembling clear, concise, and yet comprehensive agile framework SOWs that ask suppliers to propose approaches for building innovatively new products and services using proven teams of teams reference models. Sixth of all, acquirers need to start sending clear messages to the supplier community that agile methods are okay, acceptable, safe, reliable, and tenable approaches to building innovatively new products and services in the public sector. While some acquirers (public sector agencies) have taken notice of agile frameworks and have written agile framework SOWs to designate very specific ones, few acquirers have generalized their SOWs to open the door for more than one type of agile framework. Furthermore, most SOWs are still scope or product-requirement heavy, instead of process-oriented and trusting in the agile framework as a system of collaboration to be used by acquirers and suppliers to gradually evolve ecosystems of innovatively new products and services. Let's examine some agile SOW principles and practices that can help acquirers and suppliers accomplish this latter goal.

VISION FOR AN AGILE CONTRACT STATEMENT OF WORK (SOW)

- ✓ ***Provide industry thought leadership.***
- ✓ ***Promote collaborative behaviors.***
- ✓ ***Focus on emergence and experimentation.***
- ✓ ***Encourage design, lean, and agile thinking.***
- ✓ ***Specify lean-agile behaviors and processes.***
- ✓ ***Recommend proven lean-agile frameworks.***
- ✓ ***Rest on openness, transparency, and trust.***
- ✓ ***Avoid traditional frameworks and standards.***
- ✓ ***Steer away from traditional-agile hybrids.***
- ✓ ***Favor high-performance feature teams.***
- ✓ ***Be open to multiple cohesive agile methods.***
- ✓ ***Institute knowledge worker empowerment.***
- ✓ ***Establish a “one-team” acquisition culture.***
- ✓ ***Get fast market, customer, and user feedback.***
- ✓ ***Drive innovative industry behaviors from start.***

A QUICK RETROSPECTIVE OF GOOD, BAD, AND BETTER AGILE STATEMENTS OF WORK (SOWs)

GOOD

- Appear in increasing frequency.
- Attempt to add clarity to customer needs.
- Fairly concise statements of objectives.
- Favor process vs. product requirements.
- Beginning to use lean-agile terminology.
- Occasionally suggest agile methodologies.
- Occasionally suggest agile roles.
- Sometimes require agile methodologies.
- May specify detailed agile practices.
- Occasionally suggest agile frameworks.
- May demand specific agile frameworks.
- Short term in nature to preserve options.
- Occasionally apply incremental funding.
- May indicate agile metrics and dashboards.
- May suggest application lifecycle tools.
- May suggest modern agile practices.
- May suggest agile roles & responsibilities.
- May demand agile certifications.
- May suggest design thinking practices.
- May demand agile security practices.
- May suggest a cloud operating platform.
- Suggests continuous integration & delivery.
- May suggest use of DevOps practices.
- May suggest agile training & onboarding.

BAD

- Written in completely arbitrary language.
- Too comprehensive for fear of ambiguity.
- Contain heavyweight product requirements.
- List 100s of outdated traditional standards.
- Have detailed requirements attached.
- Require an outdated waterfall lifecycle.
- Require integrated master schedules.
- Require earned value management.
- Contain ambiguous open-ended language.
- Send mixed messages to suppliers.
- Broadcast indifference to agile methods.
- Specify strict power distance hierarchy.
- Relegate suppliers to programmers.
- Do not allow suppliers to self-organize.
- Broadcast adherence to traditional thinking.
- Promulgate industry-wide traditionalism.
- Unaware traditionalism leads to failure.
- Unaware agile methods lead to success.
- Specify hybrid traditional-agile approaches.
- Relegate agile to schedule work packages.
- May be too specific about agile methods.
- Cause suppliers to hybridize agile methods.
- Promote lack of trust in agile methods.
- Illustrate inexperience with agile methods.

BETTER

- Illustrate industry thought leadership.
- Display trust in lean and agile methods.
- Be less specific on product requirements.
- Be open to multiple lean and agile methods.
- Require an industry proven agile method.
- Require industry proven agile frameworks.
- Really emphasize lean and agile thinking.
- Require acquirer-supplier collaboration.
- Focus on emergence of product scope.
- Focus on a salient few agile practices.
- Demand adherence to an agile method.
- Demand adherence to an agile framework.
- Expect consistency across agile teams.
- Demand continuous improvement.
- Demand frequent agile assessments.
- Demand collaboration over automation.
- Insist upon low-cost cloud infrastructure.
- Demand easy-to-use tools & technologies.
- Demand frequent agile & tool training.
- Demand value stream mapping.
- Demand agile business model canvases.
- Demand frequent deliveries above else.
- Demand frequent deployments as well.
- Demand DevOps ecosystems & pipelines.

Agile SOW Principles & Practices

1. **AG-ILE • BUSI-NESS • MOD-EL** (*ăj'əl • bīz'nīs • mōd'l*) Copy, image, replica, pattern, facsimile; [To form a broad skeletal framework of the vision, scope, purpose, and constraints](#)

- ✓ **An overall operating model.**
- ✓ **A near-term lean canvas.**
- ✓ **Scope and boundaries.**

It's important to establish the overall operating agile business model of the acquirer so the supplier may propose an adequate lean-agile framework to be successful within the context of that model. This sets a proper context, boundaries, and constraints; identifies the strategic and tactical operating goals and objectives; and describes the overall as-was, as-is, and to-be operating state of the acquirer's enterprise, business unit, division, portfolio, program, project, or even a team (in the case of very small acquisitions for level of effort services). The agile business model should identify the key points of contact, stakeholders, administrative and technical personnel and directors, and even interfacing organizations and stakeholders (as well as other transient external stakeholder groups such as regulators, adjacent markets, special interest groups, compliance and oversight authorities, watchdogs, and even academic or research parties). Of course, the agile business model should identify the key assets, products, services, and service products to be created, modified, or maintained by the supplier within the direct scope, purview, and boundaries of the intended contract. It's important not only to identify the directly affected assets the supplier must create, modify, maintain, or decommission, but legacy assets as well, so the supplier understands which assets to include within the scope of their proposal, enhance, interface, or simply leave alone (not touch). Again, transparency is the key to establishing a proper agile business model and there's no sense in hiding an 800-pound gorilla behind the curtains that may swallow the new products or services the supplier is to create, modify, or maintain within minutes. The agile business model must include other salient factors like key performance parameters or indicators the acquirer uses to evaluate the performance of the operating business unit of interest. These may include revenues, profits, customers, customer satisfaction, employee satisfaction, uptime, continuity-of-operations, mean-time-to-recovery, deployment speed in the case of DevOps, user experience, and other leading and lagging indicators. Other salient factors that should be clearly and concisely outlined within the agile business model include the operating budget, revenues, and profits of the acquirer's direct business unit so that the supplier stays within these boundaries with its proposal; the organizational structure of the acquirer's direct business unit, along with the team directly affected by the supplier's proposal (departments, teams, and groups to which the supplier must interface or operate within); and any other special constraints like the acquirer's operating policies, guidelines, standards, and procedures (as-was, as-is, and to-be). More importantly, the agile business model must focus on describing the solution or solutions that the supplier must create, modify, maintain as clearly as possible without being too specific, which is far easier said than done. The agile business model must focus on the "what" and not the "how," design, or implementation details; must be as goal oriented as possible while leaving ample room for emergence, creativity, and innovation; and resist the temptation to be overly specific, constraining, or detailed. For instance, the acquire should not suggest a detailed integrated master schedule (IMS), overly specific enterprise or system architecture, reams of 15-year-old business requirements, nor try to define the specific solution itself. The goal of an agile business model is to be open and transparent, identify some specificities within reason, and ultimately create a "process" environment and operating contract for collaboration, communication, emergence, creativity, and innovation (i.e., let solutions emerge as a result of the process). In essence, this section is a business model canvas, lean canvas, portfolio canvas, solution canvas, or project canvas.

2. **Ag-ile • Cus-tom-ers** (*ăj'əl • kūs'tə-mərz*) Buyer, client, patron, end-user, purchaser; [To identify the direct primary and secondary customers, market, or end-users of the innovatively new products, services, or service products](#)

- ✓ **Broad mass market or market segment.**
- ✓ **Particular business class or demographic.**
- ✓ **Business-to-business or consumer groups.**

The business model should have a clear description of the market, customers, or end-users of the intended products, services, or service products, whether they are internal, business-to-business, or external recipients. Customer identification could, should, or would come from executive-level enterprise, market segment, division, business, or portfolio planners. That is, there should be a crystal-clear understanding of the internal or external market segment, group, organization, class, or individual consumer of the innovatively new products and services. More importantly, there needs to be a solid understanding of who the competitors are, what the market gaps are, what the pain points in the existing markets are, who the served and underserved consumers are, and the range of possible innovations these customers need. Oftentimes, identifying customers falls upon marketing executives, departments, managers or staff, but now falls upon product management. The responsibility of identifying markets, market segments, customers, or end users may even fall upon lean user experience or business experimentation personnel, which often go hand in hand. In traditional organizations, executives, managers, and professional business, marketing, and product management personnel felt they could personally identify a mass market, market segment, or even small group of consumers willing to purchase a new line of products and services. Lean-agile product managers, user experience, design thinking, or business experimentation personnel make no such assumptions. They may employ a variety of closely related techniques to tease out potential markets, market segments, and consumer classes. They do this by constructing empathy, journey, or experience maps to identify gaps in products, services, or service products; possible new innovations; and areas for further exploration. Once these gaps are identified or hypothesized, lean-agile product managers or user experience personnel will conduct a series of low-cost business experiments using lean startup cycles or startup way techniques to validate these gaps and hypotheses. That is, business experiments lead to tangible quantitative and qualitative data that a real market gap or opportunity exists. Oftentimes, it takes several licks to get to the center of a tootsie roll tootsie

pop to clearly identify these gaps or possible new customers. There are a variety of closely related techniques that can be used for this like 5x5 business experiments (i.e., a series of five, \$5,000 experiments), design sprints, lean startup or startup way techniques, models and simulations, etc. Design sprints are unique in that executives and innovators form a hypothesis, propose and mockup a new solution, and gather real customer feedback in only one week. It may take several weeks to refine a hypothesis. Once a market, customer, customer segment, or consumer group can be identified, then real development and implementation can ensue to fill that market gap or need with an innovation. The point is that new product and development should not occur, ensue, or begin until there is a clear tangible group of consumers or market demand for a significant new investment in an ecosystem of new products and services. This should be part of the business case or return on investment analysis and sometimes involves Monte Carlo simulations based on the output of business experiments that reveal tangible evidence of market demand. Personas are used to characterize the market, market segment, or group of consumers who will consume the product and service innovations (i.e., “business executives who will pay a premium price for an airline ticket with at-will boarding, large front-row seats, dedicated flight attendants, luxury meals and drinks, and quick offboarding at the destination”). These are often called gold or platinum customers but are increasingly more esoteric underserved customers.

3. **Ag·ile · Val·ue · Stream** (ǎj'əl • vǎl'yoo • strēm) Flow, current, course, chain, sequence; [To identify a broad sequence of beginning to end, cradle to grave, or concept to cash product lifecycle stages](#)

- ✓ **Enterprise supply chain.**
- ✓ **Product or service lifecycle.**
- ✓ **Portfolio or business unit lifecycles.**

Part of the acquirer's agile business model are the agile value streams that pertain to the acquirer's immediate business context. A value stream is an overall end-to-end supply chain that depicts a series of broad value-adding enrichment steps an enterprise uses to depict the overall lifecycle of a major product, service, or service product. For instance, an automobile value chain may include research and development stage to conceptualize new automobile designs, an extractive stage to acquire the raw materials necessary for mass production, a transport stage to move the raw materials to the manufacturing site, a construction phase to build new manufacturing facilities for a new line of automobiles, a personnel acquisition phase to acquire and train large numbers of low cost laborers, an engineering phase to develop the detailed designs of the new automobile line, a manufacturing phase to mass produce the detailed engineering designs, a shipment phase to move the new automobiles to the target market, a distribution phase to move the automobiles from the ports of entry to the sales points or showrooms, a marketing phase to advertise the new automobile line to the target market, a sales phase to train the show room personnel how to accentuate the features of the new automobiles, a contracts and financial phase to sell automobiles to individuals or groups, a components manufacturing phase to manufacture replacement parts for its fleet of vehicles, a maintenance phase to service a fleet of vehicles sold to customers, a software maintenance phase to update the embedded software and wirelessly distribute software patches and upgrades, and a decommissioning phase to dispose of vehicles when they reach their end of life. In traditional statements of work, the details of the value stream were not divulged to the supplier and suppliers were left to their own devices to master the details of the acquirer's value stream, however, in the context of an agile statement of work, its important for the acquirer to divulge its end-to-end value stream to the supplier in the spirit of openness, transparency, communication, and collaboration. The enterprises of large acquirers have many value streams and it's important for acquirers to divulge the complete operating model or portfolio of value streams that pertain to the supplier's proposal for innovatively new products, services, or service products. For instance, our automobile manufacturer may produce formula one race cars, industrial vehicles and equipment, military or government vehicles, mass transportation vehicles, luxury vehicles, mid-range sedans, subcompacts and economy vehicles, co-produced vehicles with industry partners or alliances, vehicles for special international markets, or even provide specialized products and services like legal, investment, banking, finance, or healthcare services for large numbers of employees (that is, they may have subsumed, acquired, or intermediated a bank or medical insurance provider to control costs of operation more precisely). Again, in the spirit of transparency, the acquirer should divulge the total scope of its value streams to the intended supplier, not just in the spirit of openness or transparency, but in a direct management or technical line of reasoning. Perhaps, the supplier must create a new product or service that crosses individual value streams, rely on policies or regulations from an adjacent value stream, share components with other value streams, or be subject to bottlenecks, constraints, or impediments emerging from other value streams. In the case of our automobile manufacturer, the supplier may be dependent upon the legal value stream to establish a contract to enable the creation of new products and services, the engineering value stream for new infrastructure designs, or the sales, marketing, and distribution value streams for customer feedback on the supplier's new products and services.

4. **Ag·ile · Vi·sion** (ǎj'əl • vĭzh'ən) Idea, image, dream, ideal, concept; [To identify a broad overall concept, artist's rendering, or as-is vs. to-be state for innovatively new products and services often in the form of an elevator speech](#)

- ✓ **Future state of value streams.**
- ✓ **As-is vs. to-be products and services.**
- ✓ **Compelling vision, mission, and purpose.**

Another major component of the acquirer's agile business model is the agile vision. A vision is a concise, but powerful statement of the acquirer's enterprise, market segment, business unit, operating unit, or direct division or department. More concisely, a vision is a clearly vivid mental visualization or image of the to-be state the acquirer desires to be in or achieve at a not-so distant point in the future based on somewhat aspirational goals and stretch objectives. A common method or technique for capturing visions is often an artist's rendering of the innovatively new product or service. This may be a hand drawn or computer rendering of the new structure such as a bridge, building, office park, or sleek new aircraft, automobile, transportation system, or office space. In the case of information systems, this may be a set of story boards, user experience

wireframes, or website or mobile application renderings. Humans are naturally very visual right-brained creatures, a picture is worth a thousand words, and the best way to communicate the final or future state of a new product or service is through pictures, artists renderings, graphics, or even short videos. In the case of information technology systems, vision statements often come in the form of a short, structured elevator speech (e.g., For <whom>, Who <wants what>, The <new solution>, Is a <unique service>, That <does something special>, Unlike <current legacy or competing solution>, Our Solution <provides valuable outputs, quickly, inexpensively, reliably>). While taking the time to conceptualize, capture, and render a vision, be it graphical, pictorial, in a video, or a highly structured elevator speech (or even a short sentence), writing crystal clear visions not only provides acquirers with a clear focus, but suppliers as well. It's a bit of a chicken-and-egg thing with vision statements (i.e., what comes first the vision or the business model). A business model canvas, lean canvas, or agile canvas is a highly structured exercise used by enterprise, business, marketing, or product managers to concisely scope or bound the constraints of a solution, suite, ecosystem, or portfolio of new solutions. That is, an extensive market or business analysis is performed to identify gaps in the existing portfolio of enterprise products and services, possible new solutions, and the impact, return on investment, or priorities of creating one or more solutions to fill those gaps. Oftentimes, in the case of large enterprises, multiple projects, programs, or portfolios may be funded and kicked off all at once to close those gaps with new solutions. Value stream mapping is a key tool for end-to-end enterprise analysis, identifying the critical value adding supply chain enrichment stages, identifying bottlenecks and delays to quickly delivery valuable products and services, and proposing solutions to fix, enhance, or streamline them. Each solution to value stream holes may become a vision for a new contract to be fulfilled by a supplier. Perhaps, in the case of our automobile manufacturer, commercial bank loans have high interest rates and have long lead times, so the manufacturer may acquire a bank, lower the interest rates, and speed up the loan for new vehicles to 20 minutes so they can sell innovative automobiles to eager consumers as quickly as possible. Or, perhaps, it may simply be an alliance or partnership with a handful of commercial banks for business-to-business agreements, systems, and policies to render new automobile loans quickly. Therefore, the vision statement may be as simple as "one click mobile app bank loans in five minutes for any consumer with an acceptable credit score" (where consumers get an electronic voucher for a new automobile). This type of a simple statement provides the acquirer with a concise scope for a statement of work, a supplier with a concise scope for a proposal, and the developers with a concise scope for mobile app development.

5. **Ag-ile • So-lu-tion** (ǎj'əl • sə-loo'shən) Answer, scope, possibility, explanation, opportunity; [To identify the broad scope, boundaries, parameters, and constraints of innovatively new products, services, and service products](#)

- ✓ **Broad specification of problem space.**
- ✓ **Broad narrative, description, or summary.**
- ✓ **Broad specification of 'what' is needed vs. 'how'.**

The acquirer's agile business model should also contain a broad description of the new solution that the supplier must provide, whole or in-part. That is, lean product management, user experience design, design thinking, and even value stream mapping is used to triangulate on potential types, kinds, and classes of innovatively new products, services, or service products. These activities often result in a vision, artists rendering, or vision statement for new classes of products and services. Acquirers must then continue putting technical boundaries around the expected new products and services in terms of costs, revenues, profits number of users, capacity, volume, performance, functionality, features, characteristics, and even non-functional requirements such as ease of use, quality, maintainability, reliability, safety, etc. Acquirers should resist the traditional temptation to form detailed product-oriented statements of work, business requirements specifications, enterprise or system architecture specifications, blueprints, detailed designs, or even implementation-specific specifications. Lean-agile specifications often come in the form concise one-or-two-page epic-MVP statements, capability or feature business-hypothesis statements, or even a small group of very broad user stories in the form of a story map. The epic-MVP should be broad enough to encompass an entire range of possibilities without too much specificity. A few examples may be, a consumer grade electric vehicle with a range of 400 miles with a price tag of under \$30,000, a self-service mobile app filing system for individual taxpayers with one-click instant refunds in only five minutes, one click airline reservations to any city for under \$200 within 24 hours' notice, self-service mobile app for purchasing the favorite color of your automobile without dealer markup delivered to your door within 24 hours, etc. These statements are not much different than vision statements or highly structured elevator speeches, but epic MVP statements may contain a few more details like a limited (lean) set of features, performance and quality specifications, and of course cost and budgetary constraints as well (along with legal or regulatory specifications). Acquirers should not consider epic-MVP statements, capability or feature business-hypothesis statements, or small groups of user story maps as non-negotiable, ironclad, or irrefutable contractual or legal specifications. They are merely broad specifications that are subject to further validation, scoping, clarification, emergence, innovation, adaptation, addition, and even substitution. Upon further analysis, perhaps one of these epic-MVP statements may need to be split, refined, adjusted, or deleted. Maybe, it's a bit difficult to buy a car on one click in under 24 hours. Perhaps, the individual capabilities or features may need to be split into two epic-MVP statements. Maybe consumers need a mobile app for one-click bank loans and a separate mobile app for one-click 24-hour automobile purchasing. Maybe the one-click 24-hour \$200 airline reservation needs a one-click discount hotel reservation mobile app in the event that a connection flight leaves the traveler stranded, or a one-click automobile reservation mobile app to go along with the one-click airline reservation. There's nothing worse than arriving at your target destination at midnight and all of the car rental agencies are closed or you're standing in the middle of a parking lot in 25-degree weather while 15 people are in front of you haggling for a car they failed to reserve in advance (and you get the leftovers after everyone else got what they didn't deserve). It takes a little practice and skill to be as specific as possible without handing over 15-year-old business requirements specifications with untestable statements. Furthermore, don't nest complex epic-MVPs (i.e., I want a new car, but I'll need a garage, a college degree, job, personality, and income)!

EXAMPLE OF A NEAR-TERM LEAN-AGILE PRODUCT ROADMAP VS. DETAILED INTEGRATED MASTER SCHEDULE (IMS)

Who	Timeframe	1st Half	2nd Half	Next Year	Future Years
Engineering	Theme	Indestructible Hose	Flower Management	Green Evenness	Extensibility
	Features	<ul style="list-style-type: none"> • 20-40' Lengths • No-lead connections • No-kink armor 	<ul style="list-style-type: none"> • Super flexibility • EZPlace stakes • Low-pressure mode 	<ul style="list-style-type: none"> • Microfine sprinkler • Dispersion mgt • Rain sensor 	<ul style="list-style-type: none"> • Industrial/Farming components
	Business Objectives	<ul style="list-style-type: none"> • Increase unit sales • Decrease returns • Decrease defects 	<ul style="list-style-type: none"> • Double selling price 	<ul style="list-style-type: none"> • Increase brand value 	<ul style="list-style-type: none"> • Expand marketshare
	Development Stage	<ul style="list-style-type: none"> • Preproduction 	<ul style="list-style-type: none"> • Prototype 	<ul style="list-style-type: none"> • Discovery 	<ul style="list-style-type: none"> • Concept
	Infrastructure	<ul style="list-style-type: none"> • Santa Fe Plant 	<ul style="list-style-type: none"> • New Mesa Plant 	<ul style="list-style-type: none"> • Cincinnati Plant 	<ul style="list-style-type: none"> • Future Plant
	Dependencies/Risks	<ul style="list-style-type: none"> • Key personnel 	<ul style="list-style-type: none"> • Untested material 	<ul style="list-style-type: none"> • 2nd plant required 	<ul style="list-style-type: none"> • Global competition
Marketing	Product Volume	• 100K Units	• 1M Units	• 4M Units	• 10M Units
	Markets	• Santa Fe & Phoenix	• Southwest & NE	• U.S. & Canada	• Pro Market
	Sales Events	• Partner Showcases	• Lawn & Garden Show	• Hardware Show	• Manufacturing Event
Executives	Confidence Levels	• 90%	• 75%	• 50%	• 25%
	Market Sizes	• \$200 million	• \$2 billion	• \$4 billion	• \$7 billion
	Revenues/Profits	• \$5 million/\$0.7 mil.	• \$50 million/\$2 mil.	• \$200 million/\$15 mil.	• \$500 million/\$50 mil.

6. **Ag-ile • Road-map** (*ăj'əl • rōd'măp*) Chart, guide, outline, depiction, direction; [To identify a broad sequence of new product and service hypotheses, business experiments, and anticipated outcomes](#)

- ✓ Plan with multiple time horizons.
- ✓ Near-term quarterly business plan.
- ✓ Short, medium, and long-term goals.

The acquirer's agile business model should also contain a broad agile roadmap. An agile roadmap is a broadly scoped near-term rolling wave timeline, sequence, or chronology of events stated in the form of quarterly themes, goals, objectives, features, development stages, locations or platforms, dependencies, and risks. It may contain market data such as sales or end user volume, market segments or tiers, and sales, distribution, or customer outlets or access points. Agile roadmaps may also contain confidence levels, market sizes, and revenues per quarter. Agile roadmaps are broad timelines that appear in the form of calendar quarters; in-progress, soon, future, and completed Kanbans; semi-annual timelines; goal-oriented, time-based, and objective-driven roadmaps; and even objectives and key results (OKR) roadmaps. Agile roadmaps are NOT detailed integrated master schedules (IMSs); iron-clad (unbreakable) chains of complex epic-MVPs, capabilities, features, or user story groupings; nor task oriented Gantt charts (with IMSs beneath them). The first quarter goal may be a prototype electric car with a 100-mile range on a test track in 90 days, an electric test vehicle with a 200-mile range on a cross country road test in 180 days, and a base model consumer electric car with a 300-mile range for under \$30,000 within 180 days. That's it, not business requirements, stacks of engineering blueprints, miles of IMSs, and enough detail to measure earned value management (EVM) down to the minute. Each quarter should represent a single modest epic-MVP and each subsequent quarter should build upon the other with a little more detail, fidelity, and functionality. Only the first quarter should have moderately high confidence, the second quarter should have reduced confidence, and the third quarter and beyond should have little to no confidence. Again, quarters or semi-annual lanes should have a theme (prototype car), features (100-mile range), objectives (viable lithium battery), development stage (discovery), infrastructure (R&D plant), dependencies (priority one R&D team), risks (battery safety), product volume (3 units), end-user (test drivers), distribution (presales events), confidence (70%), costs (\$10 million), etc. The second quarter or semi-annual lane may have a bit more detail followed by the third lane with the lowest statistical probability or confidence of the three lanes. The acquirer may forecast more than three lanes in the roadmap for posterity and give the supplier an idea of the types of costs, skills, risks, technologies, and processes that may be needed in the long run, but everything beyond the first quarter is completely negotiable and is subject to discovery, emergence, innovation, and unforeseen market constraints. Any forecast beyond the first three quarters is only for information purposes. Again, the acquirer should resist contracting for too many detailed epics-MVPs, capabilities, features, or user stories for 1, 5, 10, or 15 years; creating ironclad IMSs; detailed roadmaps or Gantt charts; budgeting for scope vs. time; or expecting to EVM on the agile roadmap. Once again, the goal is to set a proper expectation of the acquirer, prepare the supplier to provide an appropriate proposal, and resist any attempt to overload a roadmap with an IMS-like timeline replete with epic-MVPs, capabilities, features, and user stories to load everyone's time down to the minute for the next three to five years. The acquire should not sneak in hidden epic-MVPs (i.e., revamp my data center, information technology infrastructure, and build a state-of-the-art one-click check out mobile app—That's THREE epic-MVPs and is a called a double or triple barreled requirement—Pretty sneaky, huh?—Three for the price of one)! And, of course, the acquirer should not expect the supplier to deliver too many simultaneous high-risk epic-MVPs and violate basic WIP limits as found in traditional contracts.

7. **AG-ILE • FRAME-WORKS** (*ăj'əl • frām'wûrkz'*) Support structure, skeletal enclosure, scaffolding platform, broad architecture; [To identify a small set of proven highly-cohesive lean-agile scaling frameworks with known properties that may be used](#)

- ✓ Lean-agile reference model.
- ✓ Lean-agile organizational guidelines.
- ✓ Proven commercial lean-agile scaling model.

The next major section of the agile statement of work is the “acceptable” agile frameworks section. The first major section was the agile business model section which provided an overall scope of the acquirer's organization, problem space, and acceptable range of product solutions and timelines. This is a short section that specifies an acceptable range of industry proven lean-agile frameworks that the supplier may select from to meet the broad constraints of the acquirer's agile business model. That is, oftentimes, the acquirer is hesitant to identify a specific lean-agile framework and allow the supplier to propose one for use. However, this section should specify the types and kinds of lean-agile frameworks the acquirer believes are sufficient to deliver on the scope of its agile business model. For larger enterprise, portfolio, or systems of systems acquisitions, the acceptable agile frameworks may include Scaled Agile Framework (SAFe), Disciplined Agile Delivery (DaD), Scrum at Scale (S@S), Enterprise Scrum, Large Scale Scrum (LeSS), Recipes for Agile Governance (RAGe), Solutions for Agile Governance in the Enterprise (SAGE), Spotify, Lean Startup, Startup Way, or a similar proven framework. Acceptable agile frameworks for smaller scale project, systems, or point-solutions may include Scrum, Kanban, Extreme Programming (XP), Feature Driven Development (FDD), Crystal Methods, Dynamic Systems Development Method (DSDM), DevOps, DevSecOps, etc. The overall goal is for the acquirer to successfully achieve the scope and expectations of its aforementioned agile business model and for the supplier to select a proven agile framework with known performance properties. This prevents the acquirer from stumbling by failing to proactively specify a range of proven lean-agile frameworks or prevents the supplier from also stumbling and making a poor selection itself. The more common faux pas is for the acquirer to list 100 possible traditional standards from which to choose and then sugar coating this section or putting lipstick on a pig by mentioning a few agile frameworks like Scrum (i.e., use 99 traditional standards AND Scrum too). Likewise, by omitting the 99 traditional standards, this prevents the supplier from choosing an incorrect delivery framework like CMMI, ISO 9001, or some other traditional linear waterfall method like MIL-STD-1521b, DoD-STD-2167a, MIL-STD-498, J-STD-016, ISO/IEC 15288, ISO/IEC 12207, or some insane combination of these. In most cases, the majority of acquirers simply don't care what

framework the suppliers propose to use, the suppliers pick their favorite traditional framework from 1975, everyone botches the proposed delivery framework with a smattering of 100 standards, or the acquirer or supplier simply picks the flavor of the day (i.e., design thinking, DevSecOps, model-based systems engineering, etc.). Another danger is that the acquirer simply shoe horns or force fits traditional thinking into a lean-agile framework (i.e., the acquirer will design detailed requirements and integrated master schedules and the supplier will jam as many user stories into a day as possible and build our over scoped pipe dream in a 10-year 260 sprint IMS replete with earned value management—Oh yeah, and don't forget MIL-STD-1521b, DoD-STD-2167a, MIL-STD-498, J-STD-016, ISO/IEC 15288, ISO/IEC 12207, design thinking, DevSecOps, model based systems engineering, and Scrum too). So, identifying a SMALL list of LEAN-AGILE frameworks WITHOUT 100 traditional standards dating back to 1975 goes a long way to helping both the acquirer and supplier collaboratively fulfill the expectations of the acquirer's business model with some degree of success. And, again, acquirers should not try to force fit traditional linear IMS-driven waterfalls into Scrum for posterity. Suppliers are too gullible to this kind of chaos and think chaos is okay.

8. **AG-ILE • PLAY-BOOK** (*ǎj'əl • plā'book'*) Guidelines, policies, procedures, manual, instructions; [To propose the application, tailoring, and use of a proven highly-cohesive lean-agile framework and its components for new product and service delivery](#)

- ✓ **Book of tactical plays.**
- ✓ **Book of rules or suggestions.**
- ✓ **Book of processes and procedures.**

Now, with the first two major sections of the agile statement of work under our belt, agile business model and agile frameworks, the supplier is free to propose and describe the best proven lean-agile framework they believe is necessary to fulfill the scope and expectations of the agile business model. Notice what has been said here, the supplier will attempt to fulfill the expectations of the agile business model VS. a business requirements specification or integrated master schedule. And, the supplier will propose and describe a proven lean-agile framework VS. a mosaic of best practices from 100 industry standards dating back to 1975. This of course implies that the acquirer and supplier believe and understand that an agile business model is sufficient for the statement of work vs. detailed business requirements or integrated master schedule, and a lean-agile framework is sufficient for satisfying the agile business model. Of course, this implies the acquirer and supplier understand how to apply a lean-agile framework properly, have experience and confidence in doing so, and understand lean-agile thinking values, principles, methods, frameworks, practices, tools, and measures. With all of these assumptions in mind, the supplier now identifies and selects a lean agile framework from the “acceptable” agile frameworks section and begins proposing and describing how to use it to satisfy the expectations of the agile business model and its intended solution. Again, the acceptable agile frameworks for large solutions may include Scaled Agile Framework (SAFe), Disciplined Agile Delivery (DaD), Scrum at Scale (S@S), Enterprise Scrum, Large Scale Scrum (LeSS), Recipes for Agile Governance (RAGE), Solutions for Agile Governance in the Enterprise (SAGE), Spotify, Lean Startup, Startup Way, or a similar proven framework. And, the acceptable agile frameworks for smaller scale project, systems, or point-solutions may include Scrum, Kanban, Scrumban, Extreme Programming (XP), Feature Driven Development (FDD), Crystal Methods, Dynamic Systems Development Method (DSDM), DevOps, DevSecOps, etc. They will propose one of these, resist hybridizing them beyond what is already recommended by the framework, and begin describing its agile mindset, methodology, ceremonies, practices, techniques, roles, specialists, metrics, and tools. Some frameworks, like the Scaled Agile Framework (SAFe) contain seamless interfaces to other methods like Kanban, Scrum, Design Thinking, DevOps, and DevSecOps, while others do not contain seamless interfaces across methods. Both the acquirer and supplier should have a comprehensive understanding of these native interfaces when the supplier writes a proposal and the acquirer performs the source selection (i.e., evaluates proposal). Again, both the acquirer and supplier should resist the temptation to simply hybridize every known method and throw in 100 traditional standards for posterity. Less is more when it comes to lean-agile thinking and hybridizing beyond reason should be an immediate red flag (i.e., combining CMMI, ISO 9001, DevSecOps, Integrated Master Schedules, EVM, and DoD-2167A Data Item Descriptions with a \$50 million configuration management tool from 1975). It is acceptable to combine highly specialized techniques within a broad proven lean-agile framework when necessary (i.e., combining user experience design, application security, Scrum, and DevOps, for instance). However, especially in the case of large problem and solution spaces in agile business models, some of these smaller lean-agile frameworks may not contain the techniques and practices for scaling the delivery process beyond a few teams. For instance, only some of the larger frameworks contain practices for portfolio management, larger solution systems of systems, and larger team of teams-based programs.

9. **Ag-ile • Mind-set** (*ǎj'əl • mīnd'sět'*) Ethos, psyche, attitude, mentality, psychology; [To propose the salient values, principles, and philosophical elements necessary to properly implement the selected lean-agile framework and its components](#)

- ✓ **Lean thinking philosophy.**
- ✓ **Mental attitude or inclination.**
- ✓ **Lean-agile values and principles.**

The agile mindset a critical element of both the acquirer's and supplier's agile statement of work (SOW), agile business model, and agile playbook. The agile mindset is the basic overall philosophical, psychological, behavioral, and management system of values and principles by which the supplier proposes to operate and execute in cooperation and collaboration with the acquirer. It establishes the basic philosophical foundation for the entire playbook as well as the agile (organizational) culture. It establishes fundamental norms such as trust, empowerment, openness, transparency, teamwork, communication, cooperation, collaboration, and establishing a culture of shared discovery, emergence, and even shared risk and reward (successes). These are key tenets of lean-agile thinking which suggest most decision making is decentralized and pushed to front line knowledge workers where decisions can be made quickly, efficiently, and without delay. In the emerging field of business experiments, powerful executives delegate the work of forming, designing, and executing hypotheses to frontline

knowledge workers who are directly responsible for unearthing innovatively new products and services. That is, front line innovators are closest to the customer (data), see opportunities for further market exploration, quickly form and execute hypotheses, and quickly measure these results to verify whether there is an opportunity for innovation. Front-line knowledge workers of large dot coms execute hundreds of thousands of business experiments on the actual product and services fabric directly without management pre-approval (no engineering or configuration control board approval required). Therefore, the agile mindset lays down the philosophical framework for managers to empower front line knowledge workers to explore at-will without preapproval. This isn't to say managers are not responsible for establishing guardrails in which knowledge workers are operating (and even monitoring these boundaries to ensure frontline knowledge workers are not abusing their authority nor causing harm to their firms, products, services, nor customers). There are other important elements of the agile mindset such as openness (i.e., all management decisions and data should be freely available to front line workers). In the Open Book Management paradigm, all financial and budgeting data is freely made available to front-line blue-collar workers so they can appreciate the cost of doing business, identify opportunities for cost optimization, and be rewarded to reduce costs associated with their personal insights through a system of profit sharing. Basically, blue collar workers are empowered with the data necessary to help keep the organization afloat in an era of fierce globalization and competition. In other words, there is no sense in hiding financials behind accounting firewalls where ordinary workers that are closest to the cost centers, customers, and equipment have no control. Similarly, in lean-agile operations, managers should not only empower front-line knowledge workers to make direct constructive changes to the product and service fabric, but the financials too so they can appreciate the cost of doing business (i.e., understand the financial impact and risks of conducting their hypotheses tests). Just as important is the philosophy of lean thinking to limit WIP and allow room for innovation, uncertainty, and operating volatility and risk. In other words, agile frameworks and methods are not just technical practices used by integrated master schedulers to over allocate every second of a knowledge worker's day with administrative tasks which only increases vs. decreases risk. In the agile mindset, everyone eats their own dog food, agile practices are not just for front-line workers, they are for everyone. Executives, directors, middle managers, supervisors, and front-line workers must embody lean-agile values and principles.

10. **Ag-ile • Meth-od-ol-o-gy** (ǎj'ǎl • mĕth'ǎ-dŏl'ǎ-jĕ) Style, process, approach, system, modus operandi; [To propose the application of a proven highly-cohesive lean-agile team-level methodology with known performance properties and outcomes](#)

- ✓ Proven team-level lean-agile approach.
- ✓ Body of methods, rules, and postulates.
- ✓ Cohesive system of lean-agile practices.

In lean-agile thinking, the team is the fundamental unit of operation that will be empowered to make the majority of operating decisions. Therefore, the supplier must now propose a team-level agile methodology that will not only satisfy the agile business model, but the agile mindset as well. This includes lean-agile approaches specified in the agile frameworks section. Some of these were Scrum, Kanban, Scrumban, Extreme Programming (XP), Feature Driven Development (FDD), Crystal Methods, Dynamic Systems Development Method (DSDM), DevOps, DevSecOps, etc. Some of the top ones include Scrum, Kanban, and DevSecOps, so both the acquirer or buyer should beware of selecting an outdated or ineffective methodology. For instance, suppliers proposed using Feature Driven Development (FDD) or Rational Unified Process (RUP) about 10 or 20 years ago but were largely ineffective techniques representing the inexperience of acquirers and suppliers. Selecting an agile methodology is a critically important step and its contribution to overall acquisition success should not be overlooked. Acquirers are prone to suggest hundreds of possible delivery frameworks and suppliers are prone to proposing a smattering of buzzwords to sound impressive (without understanding the risk of hybridizing dozens of techniques, confusing the delivery process, and contributing to acquisition failure). So, let's clear this up by limiting the list of acceptable agile methodologies, honing in on the top two three, and diving down deeper into a basic methodology like Scrum, Kanban, or Scrumban. It was pretty common in the 1990s for firms to propose using the Joint Application Development (JAD) or the Spiral Method in order to get away from traditional linear process and document-intensive approaches like MIL-STD-1521b, DoD-STD-2167a, MIL-STD-498, J-STD-016, ISO/IEC 15288, ISO/IEC 12207, CMMI, ISO 9001, ISO/IEC 15288, or ISO/IEC 122017. This may not have been bad IF the suppliers had a rigorous process definition of JAD or Spiral, rather than using it as an excuse for doing nothing at all. Likewise, acquirers and suppliers should resist using this subsection to propose something they are not willing to do. For instance, if the supplier proposes to use Scrum, then the both the acquirer and supplier should actually do Sprint Planning, Daily Standups, Sprint Reviews, Retrospectives, and Backlog Refinement (no excuses). Scrum was far cry from WIP-laden process and document intensive linear waterfall methodologies from the 1970s that undermine delivery success. However, acquirers and suppliers quickly learned just how rigorous Scrum's ceremonies are and we're back to where we were in the 1990s. Suppliers now propose to use Kanban to get away from Scrum's rigor without having to actually use Kanban (so acquirer and buyer beware of proposing to use Kanban as an excuse to do nothing at all). A better approach would be to apply Scrumban where the acquirer and supplier would get the best of both worlds (i.e., the rigor of Scrum ceremonies with the WIP-limiting capabilities of Kanban). So, watch out for suppliers who propose to use Kanban or Scrumban as a slight-of-hand for doing nothing at all. Of course, lean-agile coaches should examine the underlying causes of failing to apply Scrum, Kanban, or Scrumban. Is it lack of training, skill, and understanding (probably)? Is it misapplication and abuse of Scrum timeboxes (most likely)? Or, is it failure to establish strict low WIP limits to accomplish the tasks of an overburdening integrated master schedule (IMS), brittle brick-n-mortar information technology fabric, or abusive acquirer who wants to flood the supplier with constant stream of daily requests (also very likely)? The buyer and supplier must enforce some level of consistency, compliance, and good judgement when proposing a methodology for successful delivery.

11. **Ag-ile • Roles** (ǎj'ǎl • rŏlz) Job, duty, purpose, function, position; [To propose a small set of general-purpose roles and responsibilities individuals in each team must apply as a forcing function to achieve performance outcomes](#)

- ✓ Lean-agile job function.

- ✓ Lean-agile part or character.
- ✓ Lean-agile leadership responsibilities.

The supplier proposes roles and responsibilities to complement the agile methodology in the agile roles subsection. This may include a product owner, Scrummaster, or Scrum team. Other frameworks may have additional or highly unique roles. This section describes responsibilities as well (i.e., duties within specific ceremonies and what is in and out of scope for their roles). A responsible, accountable, consulted, and informed (RACI) matrix is sufficient for this purpose. The RACI matrix may consist of salient, transient, or traditional roles as well, like executives, business owners, project managers and coordinators, and other functional or shared services leads or department heads (i.e., enterprise architecture, user experience, product management, contracts, business development, administration, testing, configuration management, configuration control board, engineering review board, portfolio or program management office, information technology, information security, facilities, accounting, budgeting, etc.). In the case of Scrum, product owners are responsible for preparing a product backlog, but the Scrummaster may facilitate product planning meeting or backlog refinement (and the Scrum team itself may be involved in co-solutioning the product backlog). The product owner does not develop the product backlog in a vacuum, weigh it down with over scoped user stories, and simply throw it over the wall, while retaining the prerogative to micromanage people's time and jam in new user stories every day. The Scrummaster is often a Scrum ceremony or meeting facilitator and is responsible for enforcing the basic criteria for Scrum meetings. Oftentimes, the Scrummaster becomes the joint product owner and Scrummaster or reverses roles (i.e., product owners serve as Scrummasters and Scrummasters serve as product owners). Scrummasters are responsible for maintaining strict timeboxes, adhering to the basic criteria for Scrum ceremonies, and maintaining optimum workflow. This implies limiting WIP, not over allocating the Scrum team or filling the team to full utilization to get your money's worth out of people or squeezing in too many over scoped user stories, and generally enforcing both bottoms up agile thinking as well as lean thinking principles too. The Scrummaster enforces openness, transparency, communication, impediment removal, and TEAMWORK. A Scrum team is not a group of individuals to whom user stories are assigned to individuals to work in parallel. The Scrum team is a TEAM, so individuals form small (pair programming) teams, self-select user stories, and use one-piece workflow to solution them from cradle to grave every two or three days (i.e., analysis, design, coding, testing, and delivery). If a team cannot solution a user story in two or three days, then the Scrummaster must ask why, what is the impediment (i.e., not enough skills or training, poor development environment, too much uncertainty or lack of information, too much WIP, etc.)? The Scrummaster is responsible for helping the product owner form small, bite-sized user stories, limiting the WIP, optimizing the workflow, and ensuring user stories and business experiments can be completed by the end of the sprint or iteration (i.e., if the Scrum team plans to deliver 5 to 15 user stories by the end of the Sprint, then help them do so). Scrum teams commit to a sprint GOAL vs. a velocity or number of user stories (i.e., perhaps 5 to 15 user stories are estimated to achieve a goal). In this case, the goal may be achieved with 5, 10, or 15 user stories, so the team decides if the GOAL is achieved in collaboration with the product owner and Scrummaster. Agile roles are seen as overhead, therefore it takes special skill to optimize the number of agile roles for larger acquisitions. The supplier should propose that everyone is trained, certified, and experienced in their roles and responsibilities.

12. Agile Ceremonies (*āj'al • sēr'ā-mō'nēz*) Rite, ritual, meeting, session, gathering; [To propose a basic set of short, routine, and strictly time-boxed set of meetings each team must apply as a forcing function for systemic collaboration](#)

- ✓ A strictly time-boxed meeting.
- ✓ Regularly scheduled lean-agile meeting.
- ✓ Collaboration and communication forcing function.

The agile ceremonies subsection is where the supplier proposes a set of complimentary agile meetings corresponding to the proposed agile methodology. If the supplier proposes Scrum, then this section describes how to apply Sprint Planning, Daily Standups, Sprint Reviews, Retrospectives, and Backlog Refinement with some level of consistency, rigor, and compliance (in the context of the agile business model). The goal of the agile business model is not to throw an outdated 15-year-old business requirements specification with 2,000 ambiguously untestable statements over the wall to cut through like a chainsaw with Scrum. The goal of the agile methodology is not to create a detailed integrated master schedule (IMS) replete with an earned value management system (EVM) for measuring individual time down to the minute (as a timesheet). The goal is to help uncover uncertainty, ambiguity, and intangible customer needs using a small number of business experiments. Therefore, there must be some level of excess capacity in the Scrum backlogs. Some agile methodologies specify NO excess capacity and it is common for product owners to fill backlogs with enough user stories to cover every minute of a 40-hour workweek, but then bombard teams with new out-of-scope user stories in on a daily basis by reprioritizing peoples times, reorganizing their personal schedules, and squeezing blood out of turnip. Product owners justify this insane behavior by saying teams must be adaptable daily changes to the backlog! Worse yet, product owners assign user stories to individuals to ensure they are operating in parallel, achieve full utilization, and justify daily changes to the backlog or ability to complete an insane business requirements specification or integrated master schedule (IMS). To add insult to injury, product owners schedule 15 minute daily standup meetings at 9:00 am and then keep the entire team there for six hours, so the daily standup becomes a daily working session to ensure everyone is fully engaged and no one is goldbricking (where's the trust in this common anti-pattern). Much of this is psychology where introverts try to avoid Scrum meetings and extroverts insist upon 6-hour daily standups in order to think out loud. In other words, extroverts have a noisy mind, so they need 6-hour working sessions to bring discipline to their chaotic lives. Still, another anti-pattern is for product owners to do little to no work Monday through Thursday, but then hold an 8-hour daily standup meeting on Fridays to catch up with all of the work and have plenty to say in a weekly status report. What's the point of describing all of these all-too-common Scrum anti-patterns? This is where the supplier can propose how to properly apply the ceremonies of an agile methodology to realize the agile mindset and successfully realize the goals, objectives, and timeframes of the agile business model. IF proper WIP limits are established,

then a Scrum team can select a modest Sprint plan for a two-week period, pairs (teams) of knowledge workers can select a user story from the backlog and solution from cradle to grave it using one-piece workflow principles, standups can be held to 15 minutes, knowledge workers can report how many business experiments were achieved during the Sprint Review, knowledge workers can identify the impediments they encounter during retrospectives, and Scrum teams can reprioritize and groom the backlog mid-week to prepare for the next Sprint. Again, the goal is not to fill every minute of every day of a person's time with individual work and code an immense business requirements specification, over scoped business model, or detailed integrated master schedule (IMS). And, the goal of using an agile methodology is not to serve up daily out of scope user stories from over zealous acquirers or Scrum product owners who were traditional project managers in a previous life.

13. **Agile • Practices** (*āj'al • prāk'tīs'ās*) Rules, customs, patterns, conventions, procedures; [To propose a basic set of general best practices each team must apply as a forcing function for achieving systematic quality and performance levels](#)

- ✓ **Lean-agile conventions for teams.**
- ✓ **Recommended lean-agile procedure.**
- ✓ **Lean-agile collaborative best practice.**

The agile practices section is the place for the supplier to propose a small set of common agile best practices to complement its agile methodology. These may include planning poker (estimating), prioritization techniques, how to capture agile work requests (epics, capabilities, features, or user stories), story mapping, pair programming, information radiators (Scrum or Kanban boards), and a myriad of other agile practices. The number of agile practices ranges in the hundreds or thousands and is growing every day, to the point where it is very challenging to keep up with them. Agile practices may include advanced ones, such as test-driven development (TDD), version control, continuous integration (CI), continuous delivery (CD), behavior driven development (BDD), acceptance test driven development (ATDD), etc. What's unique about some of the practices mentioned here is that they are all common general-purpose BEST practices that can be applied in just about any agile methodology, context, or delivery scenario. Like Scrum ceremonies, agile practices become FORCING FUNCTIONS for collaboration, communication, openness, teamwork, etc. Sprint Planning is a forcing function for bottoms-up team-level iteration or Sprint planning (or it should be). Yes, it is a common antipattern for acquirers, higher level middle managers, product owners, and Scrummasters to plan for teams, but this is an antipattern that must be discouraged. Daily standups are a forcing function for collaboration, transparency, alignment, and accountability. Sprint reviews or demos are a forcing function so that product planners limit the WIP or complexity of user stories and teams actually produce something of value in a Sprint. This is not always the case as product managers often cram feature size user stories into backlogs and individuals fail to form small (pair programming) teams to self-select user stories and complete them within Sprint. Sprint retrospectives become a forcing function for continuous improvement, which is very difficult as autocratic leaders absolutely refuse to accept changes from front line workers, while the workers themselves simply resort to complaining behind the scenes. When front line workers actually proactively improve the process, their first choice is to tailor away the Scrum ceremonies because they refuse to collaborate or openly reveal what they are doing in daily standup meetings. While some of this is due to lack of teamwork in the Western hemisphere, a lot of this is due to too much WIP, lack of collaboration and impediments to communication among acquirers and suppliers, or over scoped requirements that paralyze productivity. Likewise, pair programming is a forcing function for teamwork, cross training, peer reviews, and lean concepts such as limiting WIP and one-piece workflow. Version control is a forcing function for openness, transparency, communication, and common code ownership. Test driven development (TDD) is a forcing function for rigorous testing and quality control. Continuous integration (CI) is a forcing function for testing code early and often, version control, configuration management, testing and quality control, one piece workflow, WIP limits and small batches, DevOps, incremental and iterative development, and continuous delivery. Epics are a forcing function for minimum viable products (MVPs) instead of over scoped business requirements and integrated master schedules (IMSs). Capabilities and features are forcing function for small business experiments. And, user stories are a forcing function for conversations, although most traditionalists treat them like system or software requirements that require no explanation. Of course, Scrum and Kanban boards are a forcing function for openness, transparency, communication, and accountability, so don't hide your plans in integrated master schedules (IMSs) or complex list-driven agile tools.

14. **Agile • Techniques** (*āj'al • tēk-nēkz'*) Skill, art, craft, proficiency, specialty; [To propose a small set of highly-specialized management or technical practices necessary to achieve unique project or product aspects, characteristics, or outcomes](#)

- ✓ **Highly specialized processes and procedures.**
- ✓ **Unique practices requiring specialized training.**
- ✓ **Nuanced formula to achieve specialized outcomes.**

The supplier should propose any highly specialized agile techniques necessary to satisfy the agile business model, in addition to the basic ceremonies of the proposed agile methodology, along with other minimum best practices. Once again, agile ceremonies are specific to the agile methodology proposed by the supplier, but not a given. That is, agile acquisitions are often remiss to routinely perform the basic ceremonies associated with their proposed agile methodology and may even abandon their methodology or commitment to lean-agile thinking if they are implemented poorly or perceived to be an unnecessary burden or overhead. Oftentimes, this is due to misapplication, lack of commitment, and misuse, vs. the inappropriateness of the agile methodology to the context of the agile business model. One basic, and often forgotten value or principle of early agile methodologies is courage (i.e., the courage to forge through the difficult, challenges, and constraints of tailoring the agile methodology to the given context). Teams will often forego lean-agile thinking and agile methodologies for lack of belief, commitment, and courage. Agile practices or best practices are often complementary, but generic add-ons to agile methodologies and their associated ceremonies and are not mandated by the base agile methodology. Once again, these may include planning poker, pair programming, test driven development (TDD), continuous integration (CI), etc. Like the

base agile ceremonies, agile practices are often FORCING functions for teamwork, communication, cooperation, collaboration, quality control, lean thinking, etc., especially in the Western hemisphere where fierce individualism is KING! Agile techniques, on the other hand, are even more specialized context dependent agile practices that are driven by the peculiarities of the agile business model. These may include practices such as design thinking, user experience design, empathy maps, journey maps, experience maps, application security, performance testing, load testing, capacity testing, reliability engineering, safety engineering, mobile applications, microservices, and even cloud computing. For instance, if one is a bank designing a small ecosystem of web services or mobile applications for online banking then user experience design techniques may be in order to maximize usability and user experience. The mobile banking apps may also require a high degree application security as well, therefore, the agile team may need to apply the highly specialized techniques of an application security lifecycle. This may include evaluating and establishing security requirements, identifying the application's attack surface, employing secure application architectures and commercial frameworks, designing specialized security tests, conducting security reviews, adding automated security tests to the deployment pipeline, performing security monitoring, managing security incidents, etc. If the agile business model requires an unusually large number of end-users such as a social networking application or web service, then the agile team may require cloud computing techniques, identification of a commercial cloud provider, training and certification, experience, design of basic cloud services, load testing, performance testing, capability testing, etc. If the agile business model requires unusual uptime, reliability, or availability requirements, then the agile team may require special reliability engineering techniques, technologies, virtual redundant fault-tolerant application architectures, and rigorous reliability testing and modeling. And, of course, in the case of a safety critical agile business model such as a medical device, automobile, human-rated spacecraft, or avionics application, the agile team may need special techniques for safety analysis, hazard analysis, hardware and software interlocks, and other mechanisms to ensure safety.

15. Ag·ile ·Spe·cial·ists (ăj'əl ·spěsh'ə-lĭsts) Guru, expert, master, authority, professional; [To propose a small set of highly-specialized subject matter experts to guide the proper implementation of non-functional requirements](#)

- ✓ **Highly specialized functional skills or abilities.**
- ✓ **Specialists necessary to achieve nuanced outcomes.**
- ✓ **Specialist necessary to optimize a unique characteristic.**

The supplier must also propose a small set of agile specialists to complement agile techniques proposed to address the acquirer's agile business model. This includes agile trainers, agile coaches, agile coordinators, lean thinking coaches, agile thinking coaches, agile teamwork coaches, design thinking experts, user experience design experts, application security experts, reliability experts, cloud computing experts, safety engineering experts, DevOps experts, DevSecOps experts, etc. Let's use an application security lifecycle as an example. The lead security engineer may establish a security community of practice (CoP), train the agile team(s) in security engineering, conduct a security kickoff for a project, release, or sprint, help define security user stories, identify security best practices, help define a security architecture and attack surface, conduct threat modeling, select and apply security development tools, train teams in secure coding and testing, create security documentation, select and execute security tools and tests, develop a response plan, conduct penetration testing, perform a final security review, certify the system, monitor security vulnerabilities, respond to security threats and compromises, and remediate security incidents. The same may be true for user experience design experts, who may develop empathy, journey, or user experience maps, identify pain points, fashion automated user experience wireframes, develop hypothesis tests in the form of capabilities, features, or user stories, and guide these tests through a discovery or exploratory spike to collect user feedback and rinse and repeat this cycle until production requirements can be established. DevOps or DevSecOps engineers may train agile development and information technology operations teams in test-driven development (TDD), version control, continuous integration (CI), continuous delivery (CD), behavior driven development (BDD), and acceptance test driven development (ATDD) best practices; identify, select, establish, and use a basic DevSecOps pipeline or ecosystem; work with application testing functional experts to build in test tools and adequate automated tests; work with governance personnel to empower developers to make deliveries or deployments without manual interventions or approvals; and optimize the pipeline or ecosystem to maximize the number of deployments to achieve the optimal user experience, quality, reliability, security, performance, and safety levels. This may include establishing fault-tolerant blue-green deployment environments for automatic rollback to a known state if deployments fail. Coaches are necessary to promulgate lean and agile thinking at all levels, not just the team level, especially among acquirer groups, supplier leadership groups and functional departments, and of course project management, requirements management, enterprise architecture, administrative, and other support groups. There's simply no sense in proposing an agile methodology if the acquirer is going to demand a 15,000-line integrated master schedule (IMS) and Scrum sprints are simply work packages used to drive over scoped business models and full utilization of individual team members. It's pretty typical for one team on a larger team of teams to achieve optimal performance or become a high-performing team, but it's much more difficult to get all teams on a team of teams to become high performing teams, or better yet, to get the team of teams to behave a single large high-performing team. This is sort of a Shangri La among agile acquisitions but is necessary for overall success. Furthermore, in cross-functional DevOps teams, there's a tendency to have two separate teams, but the real magic is a single high-performing team with well balanced capacity and load to manage fluctuating workloads associated with brittle information technology infrastructures. This is where the real magic begins!

16. Ag·ile ·Met·rics (ăj'əl ·mět'rĭks) Measure, model, formula, equation, calculation; [To propose a small minimum set of lean and agile quantitative and qualitative metrics and models for measuring team performance and product outcomes](#)

- ✓ **Quantitative or qualitative measure.**
- ✓ **A measure of an internal state or property.**
- ✓ **A measure of an external property or outcome.**

The supplier must propose a basic set of agile metrics to compliment the proposed agile methodology. While lean proponents characterize their discipline as more of a science than an art, agile methodologies are more an art than a science. Therefore, neither the acquirer nor supplier must go overboard with the types, kinds, and varieties of agile metrics, nor should they propose too many highly statistical, scientific, or mathematical techniques. Too much agile metrics complexity simply becomes unnecessary WIP and will undermine the goal of realizing the agile business model and overall acquisition success. Basic agile metrics may include, but are not limited to story points, velocity, capacity, load, burndowns, committed vs. actual story points, team morale, sprint goals achieved, etc. Lean proponents may include more statistical or mathematical metrics like WIP limits, lead time, cycle type, Monte Carlo simulations, etc. DevSecOps proponents may include number of commits, tests, deployments, system failures, system up time, mean-time-to-recovery, etc. Traditional proponents may include metrics such as effort, software size, software complexity, productivity, defect density, customer satisfaction, reliability, performance, load, capacity, etc. It's important to note that these are only proxies for team performance. For instance, a team may decide that the average user story size is 5 points, a sprint's worth of user stories is 75 story points, and the velocity is 75 story points give or take 10 story points. While it may be tempting to divide the total sprint capacity of the team in staff hours by the velocity to estimate the hours per story point, this would be mixing apples and oranges, because you'd be mixing qualitative story points with quantitative hours to arrive at a cost, effort, utilization, or earned value management (EVM) measure. A big mistake of traditional acquirers and suppliers is dividing the acquisition budget by the average number of hours per story point and turning this into an integrated master schedule (IMS) with the total number of user stories that can be built per sprint on an annual basis. The failure of this common arithmetic technique is there is a large degree of uncertainty in developing new products and services, loading innovation teams to full capacity, and estimating budgets and timelines by average user story effort simply doesn't work. Agile metrics are qualitative scales used for motivating a team to accomplish its sprint goals and gauge their own performance. Agile contracts are collaborative contracts (i.e., agreements between acquirers and suppliers to communicate, cooperate, and collaborate to tease out intangible market, customer, or end-user requirements using business experiments). It's hard to put a number on communication, cooperation, collaboration, experimentation, and market volatility. Agile methodologies are a game, agile teams agree on qualitative goals and a small number of business experiments with excess capacity to conduct them; they execute the experiments and collect feedback; and they rinse and repeat until qualitative lead measures have been optimized. In the case of Pirate metrics, these may include awareness metrics (impressions, click through rate, attention-minutes, site visits, likes, social shares, social impressions, podcast impressions, etc.), acquisition metrics (new leads, email subscribers, resource downloads, support/sales chats, captured email addresses, etc.), activation metrics (new trial signups, loss-leader product sales, freemium customers, replies, etc.), revenue metrics (customer acquisition cost, trial to paid conversion, first purchase, etc.), retention metrics (customer lifetime value, net promoter score, churn, modified churn, expansion revenue, etc.), and referral metrics (net promoter score, referrals, social shares, awareness metrics, etc.). The point is for suppliers to propose a small number of qualitative agile and leading metrics.

17. **Agile Tools** (*aj'al • toolz*) Device, machine, appliance, instrument, application; [To propose a small ecosystem of simple management and development tools to enhance agile team collaboration, performance, and outcomes](#)

- ✓ **A manual or automated instrument.**
- ✓ **A visual collaboration information radiator.**
- ✓ **An automated tool that performs a specialized function.**

The supplier should also propose a small set of tools to support its proposed agile methodology. These may include manual and automated tools with an emphasis on the former, and they should both be as simple as possible. There is simply no sense in proposing an expensive, multi-million dollar enterprise agile tool suite that becomes an acquisition project or epic-MVP unto itself, which would be a fundamental lean-agile faux pas of violating WIP and complexity limits. In terms of manual tools, simple information radiators are suggested at all times (i.e., Scrum or Kanban boards, white boards, flip charts, post it notes, white board markers, painter's tape, sharpies, and sticky dots for voting). Even the world's foremost Kanban experts suggest manual Kanban boards over complex automated tools, Toyota uses an Obeya (war room) to capture its manufacturing plans, and even mission and safety-critical emergency rooms and hospitals capture their Kanbans on small white boards. The purpose of an information radiator is to limit WIP, collaborate, communicate, maximize transparency, and be a forcing function for teamwork as people stand around the physical Kanban board to update it at regular intervals. A Scrum team should be able to get in a room, construct a story map on a wall or white board with post-it notes, convert this into a simple WIP-limited backlog and Scrum board or sprint plan, and stand around it in daily standups to move post it notes (user stories) from not started, to in-process, and then to testing, done, and deployed. Again, the Scrum or Kanban board is for the team to collaboratively plan and manage its own performance. Post it notes (user stories) are transient, can be pulled off at will (deleted) and recreated constituting a new backlog or sprint plan every two weeks. There is simply no sense in capturing every post it note in a database for posterity, statistical computation of earned value management (EVM), lead times, and cycle times of every post-it note (user story) ever conceived at any point in time. If an acquirer or supplier manager wishes to comprehend team performance, then they are welcome to attend a daily standup, hear the team review the status of their Scrum or Kanban board, or simply walk around and evaluate the Scrum or Kanban boards at their own leisure. This, of course, is the ideal scenario. In today's environment, manual Scrum or Kanban boards have been replaced by extremely expensive automated tools also called agile application lifecycle management (ALM) systems. These include Jira, Rally, Version One, Azure DevOps, Jazz, Trello, etc. There is basically nothing wrong with these tools, unless product managers use them to construct detailed multi-year plans, import business requirements specifications into them, allocate everyone to capacity, count minutes and seconds, export this data into integrated master schedules (IMS), perform earned value management (EVM) on them, and use them as a timekeeping system to ensure everyone performs value adding work every minute of every day. That is, use them as a detailed accounting system to manage dollars and cents down to the penny. Other tools that should be considered are Zoom, Skype, WebEx, similar video collaboration tools, and brainstorming tools

such as Mural. Again, the danger of using agile tools is that the basic mode is a list view, digital Scrum or Kanban boards are rarely used, they are stuffed full of user stories, every entry becomes an immutable commitment, and live ceremonies are skipped. Furthermore, product owners and Scrummasters direct developers to enter their personal user stories into their PCs at their own desk, complete their user stories alone, update their user stories everyday like a timesheet, and do not establish visualizations for the teams such as burndowns, velocity charts, and other charts, graphics, gadgets, or widgets.

18. **AG-ILE • CO-OR-DI-NA-TION** (*ǎj'əl • kō-ôr'dn-ā'shən*) Align, organize, integrate, harmonize, synchronize; [To propose the application, tailoring, and prescriptive components of a proven lean-agile program, project, or delivery framework](#)

- ✓ **Program or project management.**
- ✓ **Managing one or more team of teams.**
- ✓ **Cohesive lean-agile management approach.**

The next major section of the agile statement of work (SOW) describes the supplier's proposed agile coordination approach. The agile playbook section focused on supplier's proposed team-level agile methodology since the fundamental unit of an agile acquisition is the agile team. However, the supplier's proposed agile coordination section focuses on the frameworks, methods, practices, and tools for coordinating the work of multiple agile teams which is necessary in most agile acquisitions. It's pretty rare for an agile SOW to request the services of a small agile team using Scrum, Kanban, or DevSecOps, however, as previously stated the large majority of agile acquisitions require the work of more than one agile team. It's also pretty common for an agile SOW to specify the services of an agile coordination team, which may be called program or project management services, system integration services, working integrated product team (WIPT), systems engineering and technical assistance (SETA), etc. Harkening back to the agile frameworks section, the acquirer should delineate, distinguish, and differentiate between larger lean-agile coordination frameworks and team-level agile methodologies. Again, lean-agile coordination frameworks include, but are not limited to, Scaled Agile Framework (SAFe), Disciplined Agile Delivery (DaD), Scrum at Scale (S@S), Enterprise Scrum, Large Scale Scrum (LeSS), Recipes for Agile Governance (RAGe), Solutions for Agile Governance in the Enterprise (SAGE), Spotify, Lean Startup, Startup Way, etc. There are even a number of agile program and project management methodologies that may suffice for agile coordination purposes, such as radical project management, agile project leadership, extreme project management, adaptive project management, agile project management, etc. While team-level agile methodologies include, but are not limited to, Scrum, Kanban, Scrumban, Extreme Programming (XP), Feature Driven Development (FDD), Crystal Methods, Dynamic Systems Development Method (DSDM), DevOps, DevSecOps, etc. Although the supplier is free to extend the base team-level agile methodologies like Scrum with some additional tools and techniques such as a Scrum of Scrums (SoS) meeting, suppliers are encouraged to propose an actual agile coordination framework from some of the references mentioned here rather than trying to extend a team-level methodology into the agile coordination space. Depending upon the scale, scope, and size of the acquirer's agile business model, agile coordination frameworks may contain mechanisms for enterprise agility, multi-program portfolio management, large solution management for developing systems of systems, and explicit agile program management practices and techniques for coordinating teams of teams. The goal of these agile coordination or scaling frameworks is the promise to make a team of teams, large solution, or even portfolio perform like a single high-performing cross-functional agile team. That is, small lean-agile teams can now achieve performance levels 10 to 100 times greater than large traditional programs, so the challenge becomes employing the work of 6 to 12 teams, or vastly more, while maintaining the productivity levels of a single high-performing agile team. In the case of portfolios or large-solution systems of systems, this challenge may be to ensure multiple groupings of 6 to 12 teams behave as a single high-performing cross functional agile team. In reality, the goal is really to downsize the number of programs, projects, and teams; streamline or strip them down to their bare essential members; and radically limit the WIP with a generous helping of coaching and self-mastery. Regardless of the myths and realities of coordinating the work of high-performing teams of teams, this is the section in which the supplier will propose such a feat.

19. **Ag-ile • A-nal-y-sis** (*ǎj'əl • ə-nāl'ī-sīs*) Plan, organize, visualize, coordinate, optimize; [To propose a set of proven lean-agile problem space identification, analysis, and visualization tools such as value stream mapping](#)

- ✓ **End-to-end product and service supply chain.**
- ✓ **Visualization of one or more enterprise supply chains.**
- ✓ **Set of broad value adding enrichment steps in a supply chain.**

Usually, the first step in any proven large scale agile framework designed for coordinating and ensuring teams of teams perform as a single high-performing cross functional team is some sort of high-level end-to-end enterprise or agile analysis. The suggested lean-agile technique is most often value stream mapping of the end-to-end enterprise, organizational, or business supply chains. The acquirer's agile business model section should contain a high-level representation of operational and even development or technical value streams (teams of teams). That is, the most mature acquirer's will not only perform value stream analysis of their own enterprises, operating units, business units, divisions, portfolios, programs, and projects, but they'll periodically refine their analyses, make constructive changes to their value streams, and keep this data up-to-date. If this is the case, then the supplier uses this section to analyze, understand, and propose to make even more constructive changes to the acquirer's operational and technical development value streams. However, most large public and private sector enterprises are not mature enough to perform value stream mapping nor develop a hierarchy of value stream maps in an ongoing basis. In this latter all-to-common case, it falls upon the supplier to analyze, model, and capture the acquirer's value streams and initiate the creation of the operational and development value streams if they do not exist. In many cases, this is a broad constructive change to the acquirer's fundamental organization and is a bold proposition to a new supplier with little knowledge and relationship capital or trust with the acquirer. Furthermore, there is a deep sense of psychological or hierarchical power distance between acquirers and suppliers, acquirer's view suppliers as nothing more than administrative

assistants, and may not see the wisdom in a supplier-led mapping of the acquirer's value streams. This makes this section of the supplier's proposal even more valuable, as the proposal becomes the contract and acquirers are somewhat obligated to perform value stream mapping if they select a supplier's proposal that contains this stage, practice, or step. Agile analysis in the form of value stream mapping should not be confused with deeper dive methodologies like business requirements analysis, enterprise architecture, or any other broad sweeping business process reengineering methodologies. The goal of agile analysis or value stream mapping is to capture the as-is operational and development value streams in order to focus the agile coordination efforts and resources, and understand the operational and technical constraints, impediments, bottlenecks, and other obstacles to optimal acquisition performance. With a qualitative and quantitative model of the acquirer's value streams, the acquirer and supplier may undergo constructive business process reengineering efforts to improve acquisition performance. For example, let's say an acquirer's agile business model involves the operations and maintenance of some highly brittle brick-n-mortar data centers from the 1970s. In this case, 90% of the acquisition resources are spent restoring global networks when they fail once or twice a week. The acquirer and supplier may jointly decide that the best way to ensure site reliability engineering is to replace the physical data centers with virtual instantiations in public cloud services. This may be a rather significant business process reengineering of the acquirer's information technology infrastructure, but it may reduce operations and maintenance costs by a factor of 10, while improving uptime, reliability, and availability by the same amount. Therefore, both acquirers and suppliers should use agile analysis in the form of value stream mapping to avoid mopping the deck of a sinking ship, putting lipstick on a pig, or wasting resources keeping the lights on in 1970s data centers.

20. **Ag-ile • Man-age-ment** (*ăj'əl • măn'ij-mənt*) Run, execute, implement, coordinate, optimize; [To propose a set of proven lean-agile management execution principles, practices, and tools such as lean, Kanban, and other pull-based concepts](#)

- ✓ **Lean-agile information radiators.**
- ✓ **Kanban or program boards.**
- ✓ **Lean-agile roadmaps.**

The next major subsection of the supplier's agile coordination section is where the supplier proposes an approach to agile management. Agile management is the intake system for the acquirer's requirements or work requests for the acquisition. In traditional acquisitions, the acquirer may allow suppliers to propose a business requirements analysis phase. That is, hire a team of business analysts who will spend months or years assembling several volumes of untestable business requirements. In agile speak, this amounts to several thousand epics or features, an over scoped batch size, or insurmountable WIP. Worse yet, an acquirer may have already paid a supplier to do this analysis a decade or two ago, and simply throws these outdated business requirements specifications over the wall and directs the supplier to bake these untestable epics and features into integrated master schedules (IMSs), agile quarterly plans, and sprint or iteration backlogs. For all intents and purposes, while this seems like an agile management intake process, it's simply inadequate. Why? Because 10-or-20-year-old business requirements specifications do not represent current business market needs, unless your customer still needs rotary phones. Business requirements are not a minimum viable product (MVP) nor business experiment. The real customer, market, or end-user needs are intangible and haven't been discovered or uncovered yet. And, of course, a detailed integrated master schedule (IMS) will crush the acquisition with so much WIP the acquisition will simply fail. Therefore, the supplier should propose a highly rational agile management or intake system such as lean portfolio management, a Kanban system, or some other WIP-limiting agile management approach. A business requirement is treated exactly like what it is (i.e., an epic), it is highly structured and analyzed, it is prioritized, and it is placed on a queue for further solutioning. Furthermore, the intake queue has WIP limits, only a handful of epics are processed at one time, 20-year-old business requirements specifications are shredded or placed in recycling bins, and this is how the acquirer and supplier spend the duration of the acquisition processing work requests. This often replaces outdated traditional governance boards such as engineering review boards (ERBs), architectural review boards (ERBs), and configuration control boards (CCBs). Again, if this is a portfolio management acquisition, then the supplier proposes lean portfolio management. If this is a large solution system of systems or program management acquisition, then there still needs to be a Lean-Kanban intake system on the front end of small epics, capabilities, and features. Agile management serves as a governor or filter on the amount of work and WIP coming into the system, allows teams of teams enough time to finish a few business requirements at a time in the form of epics and features, and also allows the acquirer and supplier to frequently perform inspect and adapt (I&A), continuous improvement, value stream mapping, roadmapping, and other reprioritization of intake requests based upon emerging tacit market, customer, or end-user needs. Lightweight agile roadmaps may replace detailed integrated master schedules (IMSs), but acquirers and suppliers should resist long term agile roadmaps or cramming too many business requirements, epics, and features into these roadmaps in order to squeeze blood out of a turnip with too much WIP. Agile teams speed up by slowing down, that is business requirements, epics, and features move through the system faster when there are fewer of them, and slow down to a crawl or stop when there are too many of them. This explains the failure of traditional management and the success of agile management when done well (i.e., speeding up by slowing down and reducing WIP and batch sizes).

21. **Ag-ile • Ar-chi-tec-ture** (*ăj'əl • ăr'kī-tĕk'chər*) Style, design, framework, blueprint, floorplan; [To propose a set of proven lean-agile architecture and design practices that support modularization, emergence, evolution, and just-in-time delivery](#)

- ✓ **Set-based systems design.**
- ✓ **Model-based systems engineering.**
- ✓ **Emergent and evolutionary architectures.**

The next major subsection of the supplier's agile coordination section is where the supplier proposes an approach to agile architecture, which is the supplier's approach to creating lightweight, near-term emergent and evolutionary just-in-time architectural and design runways. In the case of Scrum, a single agile team may evolve their architectures from iteration-to-

iteration, quarter-to-quarter, and release-to-release, throwing away previous architectures through frequent refactoring. While this may seem wasteful to the traditional architect and designer, agile teams create a just-in-time architecture or design for every epic, feature, or iteration's worth of user stories, the architecture or design remains stable for as long as it is useful, and then thrown away or refactored when it outlives its usefulness. Small agile acquisitions often involve short-term transient point solutions, so the return on investment of evolving a small set of throwaway architectures and designs is optimal compared to the cost of creating a bloated, over scoped, and expensive 5-10-or-15-year architecture that only purports to represent imaginary requirements. This fails to mention that technology evolves so quickly that even if the long-term architecture or design was sound, it would not outlive the next quarter's major technological innovation. For instance, one quarter, a server-based system may be viable, the next quarter a desktop system is better, the next quarter a laptop is more cost effective, the next quarter a mobile app is better, and the next quarter a wearable or embedded nano-computer system is even better. In this scenario, there is no sense investing in 40 quarters worth of server systems when embedded nano computer systems will be needed in little more than a year. NASA's space station and the Air Force's F-35 replaced multi-billion-dollar mission computers with \$500 laptops. The economics of throwing away architectures and designs from quarter-to-quarter is more cost-effective than an expensive obsolete long-term architecture. Many of the 170 or so public sector agencies spent \$500 million creating enterprise architectures at taxpayer expense that were never used at all or obsolete upon completion. However, despite the business case for quarterly throwaway emergent and evolutionary architectures, larger acquisition portfolios and systems of systems solutions involving multiple teams of teams need a slightly larger investment in architectural coordination than a single team evolving quarterly point solutions. There may be a need for a common information technology infrastructure, architectural fabric and interfaces, module specifications, and design guidelines. This is the job of agile architecture, converting or synthesizing intake requests into short, medium, and long-term cohesive architectures. It's not a one-to-one mapping, and it's simply not feasible or desirable to convert business requirements specifications into bloated architectures with thousands of unneeded features and user stories in the form of integrated master schedules (IMs). First of all, the agile management intake system should convert all requirements or work requests into epic, capability, and feature hypothesis statements; these should be prioritized; and short, medium, and long-term roadmaps should be produced by the agile management team of these work requests. Then, agile architecture teams do their job, look at the roadmap, and synthesize a short, medium, and long-term architectural backlog. Of course, agile architecture personnel should have the vision to think broadly, while thinking in a lean and agile near-term fashion, much like a single high-performing cross functional agile team. Another common architectural pattern may be a microservices, cloud computing, or loosely cooperating servers, PCs, laptops, smartphones, and nano-computers requiring a highly cohesive but loosely coupled architecture.

22. **Ag-ile • Prod-uct • Man-age-ment** (*ăj'əl • prŏd'əkt • măn'ij-mənt*) Goods, produce, commodity, component, merchandize; [To propose a set of proven design thinking practices such as empathy, journey, experience, or story mapping](#)

- ✓ **Lean-agile product roadmapping.**
- ✓ **Design thinking and user experience.**
- ✓ **Lean-startup and business experiments.**

The next major subsection of the supplier's agile coordination section is where the supplier proposes an approach to agile product management, which is neither traditional product management nor team-level product ownership. The supplier's agile coordination section exists to address managing the work of multiple teams, teams of teams, and even groups of teams of teams (i.e., large numbers of people or acquisition programs well beyond the scope of a single agile team). Furthermore, these larger acquisitions involve portfolios of programs, large systems of systems, or small ecosystems of loosely coupled but highly cohesive systems. Whereas a single agile team is responsible for designing a small mobile application, larger acquisitions may be responsible for the entire end-to-end telecommunication systems involving back-end billing systems, digital switching systems, cellular towers, smart phone hardware, smart phone operating systems, and hundreds of apps. Therefore, this is well beyond the scope of a single product owner, who couldn't possibly comprehend the scope, magnitude, roadmap, vision, backlog, or story map for something this large. Therefore, a higher-level agile product management team is needed to conceive, plan, and coordinate a larger agile product management plan consisting of all of these components in the form of capabilities and features. More importantly, the agile product management team employs customer centric techniques such as design thinking, user experience design, and business experimentation to smoke out the pain points, identify value adding solutions, and specify capability and feature hypothesis statements for individual product owners. Product owners may still need to develop a vision for the benefit hypothesis if it is a salient component such as an innovative messaging app for a larger smartphone infrastructure and ecosystem. The product owner may still need to facilitate story maps, further design thinking and user experience design, and further business experimentation in the form of user story grouping or minimum viable products (MVPs). However, in this scenario, the individual product owner is governed by the higher-level product management vision, roadmap, user experience design, etc. That is, there are product management constraints on individual product owners. This is the process that supplier's must now propose to solve in the agile product management section. There is plenty of empathy, user experience, and journey mapping to be done at both the agile product management and individual product owner level. Furthermore, this is where the supplier can propose to reduce the economics of staffing by optimizing the number of product managers and product owners across teams of teams and groups of teams of teams. That is, individual agile teams may now be able to share a product owner, since the high-level agile product management team is now bearing a greater burden and degree of design thinking and user experience design for the product owner. Furthermore, product owners no longer operate in a vacuum as their own individual chief executive officer but are now part of the agile product management team. That is, product owners constitute the agile product management panel, perform design thinking and user experience design together, and then efficiently communicate the agile product management intent to the teams of teams in perfect synchronization, coordination, and consistency. This is where team level agilists and larger agile framework agilists beg to differ. Team level agilists believe that each agile team on the team of teams is a highly competitive autonomous

group that is responsible for suboptimizing its own design thinking, user experience design, and backlog refinement. An agile product management panel ensures everyone is a single high-performing systems thinking vs. suboptimizing team of teams.

23. **Agile • Plan-ning** (*ăj'əl • plă'nĭng*) Map, path, route, course, itinerary; [To propose the application of a highly-cohesive lean-agile planning framework for developing quarterly plans aggregated from multiple bottoms up planning teams](#)

- ✓ **Program increment planning.**
- ✓ **Quarterly release planning.**
- ✓ **Big room planning events.**

The next major subsection of the supplier's agile coordination section is where the supplier proposes an approach to agile planning. Once again, the supplier's agile coordination section exists to address managing the work of multiple teams, teams of teams, and even groups of teams of teams (i.e., large numbers of people or acquisition programs well beyond the scope of a single agile team). Therefore, the supplier's agile planning section proposes how all of the teams will perform planning in unison. As noted earlier, a single agile team can often suboptimize vs. become a high-performing cross functional team. The goal of the supplier's agile coordination section is to get all of the agile teams, not to suboptimize and compete with one another, but enable all to become high-performing cross functional teams, and better yet, become a single high performing cross functional team. Much of this involves cadence, synchronization, alignment, and, yes, even consistency. The more or these four basic elements (i.e., cadence, synchronization, alignment, and consistency), the better when it come to establishing single high-performing teams of teams. This is where agile planning plays a big role, synchronizing the planning, ceremonies, vision, backlog, constraints, and communication of all of the teams in the portfolio, large solution, or acquisition program level team of teams. Again, these are not Keystone Cops, competing against one another, and iterating independently, with rogue product owners acting as individual chief executive officers going off of the deep end or out on a limb alone. Agile planning teams are more like a large symphony orchestra in perfect harmony, a synchronized swimming team, or a large marching band in perfect lockstep. So, the supplier's agile planning section describes all of the cadence, synchronization, alignment, and consistency of the agile planning events to achieve coordination among larger teams of teams. First of all, teams of teams may be grouped into large acquisition level programs, sometimes called agile release trains (ARTs), individual ARTs may be grouped together in large solution trains, and individual solution trains may belong to a large portfolio in the Scaled Agile Framework (SAFe). This is simply one example for illustrative purposes. The individual agile teams within ARTs and solution trains will have the same sprint or iteration length, begin and end at the same time, and come together for quarterly or release planning as well as a final system demo, retrospective, and inspect and adapt sessions to fix any major impediments to ART performance. It's important to note that consistency and cadence among individual agile ceremonies within agile teams is also part and parcel to creating high-performing cross functional teams of teams of ARTs and solution trains. Product management teams provide visions, roadmaps, and team of team backlog priorities; individual product owners constitute the product management panel and pre-groom backlogs, story maps, and quarterly plans ahead of synchronized planning; and everyone is in general agreement as to the vision, goals, objectives, and purpose of the teams of teams or ARTs. Individual teams are still responsible for finalizing preliminary story maps, adding user stories or taking some away, adjusting load and capacity based on individual schedules and historical velocity and team performance, devising iteration plans for the quarter, and abstracting team goals into goals and objectives at the teams of teams or ART level. In Scrum, agile teams commit to sprint or iteration goals not user stories, and in a teams of teams context, ARTs commit to iteration, sprint, or quarterly goals (not user stories or scope in spite of the best efforts of traditionalists to use agile methods for hybridization purposes). Team of team plans emerge from agile teams much like in Scrum, not from the top down as many acquirers and suppliers attempt to do.

24. **Agile • Dash-boards** (*ăj'əl • dăsh'bôrdz'*) Board, panel, dials, instruments, visualizations; [To propose the application of a small set of aggregated lean-agile metric, model, and measurement dashboards to visualize overall performance](#)

- ✓ **Project, program, and portfolio visualizations.**
- ✓ **Performance measurement dashboards.**
- ✓ **Team-level performance dashboards.**

The next major subsection of the supplier's agile coordination section is where the supplier proposes an approach to agile dashboards. Agile dashboards are simple collections, aggregations, or groups of charts, graphs, gauges, and dials to measure the performance of both individual agile teams, teams of teams, and even groups of teams of teams at the large solution or portfolio level. Think of automobiles—Almost every vehicle has a dashboard with indicators or visualizations for vehicle speed, fuel level, range on the battery or fuel tank, engine temperature, rotations per minute, odometer, etc. There are even separate controls for the entertainment system, environmental management, and global positioning, mapping, or navigation system to guide the driver to the destination on-time. No automobile driver can live without these basic visual indicators of automobile performance. The driver needs to know if there is enough fuel in the tank or charge in the battery, it's helpful to know if the automobile started and the engine is running, and of course, minding the speed helps the driver stay within the speed limit while arriving at the target destination on time. Similarly, individual agile teams and larger teams of teams need performance dashboards to know how much fuel is in the tank (load and capacity), how fast the team is moving (velocity, lead time, cycle time, etc.), engine temperature (team morale, customer satisfaction, market feedback, leading indicators, pirate metrics, etc.), environmental settings (quality, reliability, security, performance, etc.), and of course global positioning system (a map to the destination, traffic bottlenecks, estimated time of arrival, etc.). Unfortunately, too many agile teams fly blindly in the dark like bats, without the sonar of course, don't use Scrum or Kanban boards, fail to check or maintain agile application lifecycle management systems, and rely on simple lists of bottomless backlogs that are filled to the brim every day by traditional product owners, acquirers, and other middle managers. Instead, suppliers are responsible for proposing simple agile dashboards for individual agile teams like visual Scrum or Kanban boards, burndown charts, velocity

charts, load and capacity charts, and visual indications of whether sprint or iteration goals are satisfied (i.e., agile team predictability). Likewise, suppliers are responsible for proposing simple agile dashboards at the teams of teams level like program or feature dependency boards, feature burndown or progress charts, load and capacity charts, and visual indicators of whether quarterly goals or objectives are satisfied (i.e., program predictability). The scenarios are similar for groups of teams of teams at the capability level and groups at the portfolio or epic level. Likewise, lead and cycle times should be visualized at the team (user story), teams of teams (feature), groups of teams of teams (capability), and portfolio level (epic). Scrum, program, and Kanban boards, roadmaps, and even story, feature, capability, and epic maps should also be available at each of these levels. DevSecOps ecosystems or pipelines will have their own performance measures and dashboards, along with other specialty groups like security, reliability, quality, capacity, load, and performance testing. There will even be information technology infrastructure and cloud computing dashboards to measure and monitor the performance of the product and service platform too. Tools or shared services teams must be established; basic visual dashboards should be established at the team, team of teams, groups of teams of teams, and portfolio level in advance; and basic agile dashboards should not be left to chance by the supplier's proposal so agile acquisitions continue flying blindly in the dark without sonar. Suppliers should propose a hands-on agile dashboard team as part of their proposal just to strike at the heart of this issue.

25. **Ag-ile • Con-tent • Man-age-ment** (*ăj'əl • kŏn'tĕnt' • măn'ij-mənt*) Tool, database, application, version control, or workflow system; [To propose a lean-agile application lifecycle management tool to plan and track the work of multiple teams](#)

- ✓ Collaborative visual planning displays.
- ✓ Agile application lifecycle management.
- ✓ Collaborative/brainstorming workspaces.

The next major subsection of the supplier's agile coordination section is where the supplier proposes an approach to agile content management. This includes proposing the supplier's agile application lifecycle management tools, content management itself which is often built into agile application lifecycle management tools, and collaborative brainstorming workspaces. Agile and lean experts prefer physical information radiators such as Scrum and Kanban boards on a wall, whiteboard, flipchart, etc. They prefer the physical, open, visual, transparent, collaborative, and simple nature of traditional information radiators. They also have lean properties like limiting WIP because of their small size. They also help constrain Parkinson's Law which states that the amount of work will expand to fill the amount of time available (i.e., by corollary amount of WIP is constrained by the amount of wall space available). The value of lean and agile methods is not in the Scrum or Kanban board nor amount of work items, but is in the amount of openness, communication, collaboration, trust, transparency, morale, innovation, and creativity they instill, infuse, or imbue. Traditionalists cannot fathom this basic concept. That is, openness, communication, collaboration, transparency, trust, morale, innovation, and creativity are worth more than a foot-high stack of business requirements or a mile-long integrated master schedule (IMS). Likewise, agile application lifecycle management tools are mainstream now and they're pretty good for capturing, managing, and visualizing the work of distributed agile teams (which is nearly all of them now), but they also come with their own drawbacks. First of all, they're expensive, hard to use, promote individual work, discourage agile collaborative ceremonies, their visualizations require a master's degree in physics to configure, and, most of all, they are subject to Parkinson's Law. Many times, traditional project managers become agile product owners and Scrummasters, order agile teams to sit at their cubicle alone, and create their own individual user stories to work on alone in a vacuum. Furthermore, there are no visualizations (dashboards) of individual or team performance, most of these dashboards are hidden or only available to middle managers, and traditional product owners and Scrummasters simply forego all collaborative ceremonies in lieu of fat fingering in daily progress into agile application lifecycle management tools. Traditionalists will even threaten individuals with termination for failing to create individual task plans in agile application lifecycle management tools, keep their progress updated daily like timesheets, and plan to full utilization and capacity to squeeze all of the value out of every minute of every day. Worse yet, once again, agile application lifecycle management tools fall victim to Parkinson's Law, where acquirers and middle managers add work to backlogs on a daily basis, each entry becomes an immutable requirement, and lead and cycle times approach infinity when individual utilization of knowledge workers reaches 70%, 80%, 90%, 100%, or even 110% or more. That is, agile application lifecycle management tools are bottomless pits of requirements that can elastically expand to infinity that obsessive compulsive acquirers and middle managers are prone to do. So, what is the supplier's proposal supposed to do? Well, the supplier's proposal should promote lean and agile thinking. Furthermore, it should promote agile ceremonies that lead to openness, communication, collaboration, transparency, trust, morale, innovation, and creativity they instill, infuse, or imbue. They should promote collaborative brainstorming workspaces like Mural for distributed agile teams and just say NO to bottomless pits of immutable user stories in agile application lifecycle management tools with infinitely elastic databases.

26. **Ag-ile • In-fra-struc-ture** (*ăj'əl • ɪn'frə-strŭk'chər*) Network, platform, applications, administration, communications; [To propose the use of a low-cost commercial distributed information technology platform for use by management and technical teams](#)

- ✓ Devices and access points.
- ✓ Network and platform infrastructure.
- ✓ Basic automated administrative foundation.

The final major subsection of the supplier's agile coordination section is where the supplier proposes an approach to agile infrastructure. This is not only important for distributed agile teams and virtual firms with a small capital footprint, but traditional firms as well with a small investment in a custom on-premises information technology infrastructure. Many firms are utilizing commercial web services for business functions, including strategic planning, executive management, marketing, sales, accounting, payroll, medical, healthcare, legal services, office productivity tools, software development, experimentation, information technology, etc. The best thing about leasing commercial web services such as these is that the staff always has

access to low-cost, state-of-the-art, and easy-to-use information technologies. Vendors are innovating them with modern DevOps practices with hundreds of deployment updates per day to add new features, fix bugs, and enhance performance. And, of course, they are accessible any time, anywhere, and 24 hours per day (i.e., maximum uptime, availability, reliability, etc.). Therefore, the supplier should propose the most flexible, cost-effective, and innovative commercial web services as possible. More importantly, use of these commercial webservices often involves access points such as inexpensive, rugged, and well-outfitted state-of-the-art devices such as laptops and smartphones. Therefore, the supplier should not only propose to provide the delivery team with low-cost, leased, and state-of-the-art laptops and smartphones, but refresh them early and often. This includes headsets, cameras, Bluetooth devices, and access to commercial teleconferencing services like Skype, Zoom, Microsoft Teams, WebEx, etc. Of course, much like agile application lifecycle management tools, commercial webservices are subject to Parkinson's Law (i.e., the amount of work will expand to fill the amount of capacity available). For instance, since Skype has become the norm for remote meetings, there is a tendency for product owners and Scrummasters to fill everyone's time with meetings from 9:00 a.m. until 5:00 p.m. Then, product owners and Scrummasters fill Fridays with working meetings to make up for all of the administrative meetings they schedule Monday through Thursday. Worse yet, they'll cancel and schedule new meetings without notice and message you during a meeting to join yet another unscheduled meeting. They'll add user stories to your backlog everyday and tell you no collaboration is necessary to understand them, they're hard requirements, just get them done ASAP. Of course, no one observes timeboxing rules, managers are 10 minutes late for meetings, and all meetings go 10 minutes to 45 minutes longer than scheduled. What's the point of regurgitating these antipatterns? The supplier's proposal should also contain a plan for enforcing lean and agile thinking, not substituting collaborative agile ceremonies for bottomless pits of commercial cloud services, timeboxes should be enforced, meetings should be scheduled in advance, meetings should be kept short, and plenty of work time should be built into calendars. There should even be quiet time when Skype meetings cannot be scheduled, like on Fridays, and email servers are turned off. That is, establish WIP limits on bottomless pits of agile infrastructures. Many agile teams are in international locations that do not have reliable power grids, Internet service, web services, offices, and information technology infrastructures. Agile suppliers should remedy this by ensuring that international teams have all of the state-of-the-art information technology equipment and web services like local domestic teams. Furthermore, time zone and meeting etiquette should also be proposed to prevent agile ceremonies from taking place at midnight, early in the morning, Friday afternoons, or late in the evening at any location.

27. **AG-ILE • GOV-ER-NANCE** (*ǎj'əl • gŭv'ər-nəns*) Body, group, board, organization, administration; [To propose the application, tailoring, and prescriptive components of a proven lean-agile portfolio, system of systems, or transformation framework](#)

- ✓ **Rules, controls, policies, and resolutions.**
- ✓ **Standards, guidelines, budgets, and guardrails.**
- ✓ **Guidelines, processes, standards, and procedures.**

The next major section of the agile statement of work (SOW) describes the supplier's proposed agile governance approach. In retrospect, the agile business model focuses on the scope of the acquirer's agile acquisition, the agile framework identifies a short list of acceptable agile approaches, the agile playbook focuses on the supplier's basic team-level agile methodology, and the agile coordination section describes how the supplier will coordinate the work of multiple agile teams or teams of teams. The agile governance section describes the supplier's proposed approach to establishing an agile delivery ecosystem, promulgating its values and principles, devising an agile transformation roadmap for both acquirers and suppliers, describes how the supplier will ensure consistency among agile teams so the entire team of teams performs as a single high-performing cross-functional team, how the supplier will establish a strong lean and agile thinking culture, and, of course, how the acquirer and supplier will assess the effectiveness of the agile playbook, agile coordination, and agile governance itself. This section will describe the formation of a lightweight lean and agile center of excellence (LACE), agile portfolio, program, or project management office (PMO), or some other similar governance board that will ensure the smooth transition of the acquisition to a well-balanced set of lean and agile practices. The supplier should not assume that the supplier has a strong lean and agile culture already established and that the acquisition may have to begin from ground zero. Oftentimes, acquirers not only wish suppliers to help them develop an innovatively new ecosystem of products, services, and service products, but use the acquisition as a training ground to instill lean and agile practices in the supplier's culture itself. In that sense, agile acquisitions are a double-barreled minimum viable product (MVP) to kill two birds with one stone. This is a red flag in most circumstances as the supplier will have enough of a challenge optimizing the delivery of innovatively new products and services without having to convert the acquirer from traditional to lean and agile thinking values, principles, practices, tools, and metrics. Worse yet, the acquirer may expect the supplier to implement or adhere to a hybrid delivery system of both traditional and lean-agile practices such as big up-front requirements, architectures, and integrated master schedules (IMSs) while relegating team-level agile methods to fully scoped work packages on an IMS for the next 5, 10, or 15 years. This is also another red flag that suppliers must watch out for (i.e., having to execute team-level agile methodologies with fully scoped schedules, requirements, and architectures for the entire period of performance). What then should suppliers do in this section? Well, they may need to be prepared to assess the acquirer's culture and develop a roadmap to convert them to lean-agile thinking, be on the lookout for heavyweight hybrid traditional and lean-agile requirements, and not assume the delivery team is mature enough to successfully deliver new products and services without a heavy dose of transformation themselves. So, this section becomes a three-pronged exercise in transforming the acquirer's culture, transforming the acquirer's hybrid practices to full lean and agile thinking, and transforming the supplier's delivery team to a lean-agile mindset too. Lean and agile methodologies are a big departure from traditional ones in that both acquirers and suppliers will see agile teams DELIVERING working new products and services every (two-week) iteration or sprint (instead of every 5, 10, or 15 years). Seeing is believing, and this will go a long way towards transforming acquirer and supplier teams. But nothing should be taken for granted and the supplier needs to propose heavy doses of transformation practices and governance early and often.

28. [Ag·ile·O·ver·sight](#) (*əj'əl • ɔ'vər-sīt'*) Panel, group, board, commission, committee; [To propose the formation, staffing, and deployment of a portfolio, program, or project management office to establish, promulgate, and promote lean-agile thinking](#)

- ✓ [Lean-agile oversight or steering committee.](#)
- ✓ [Lean-agile transformation or process improvement team.](#)
- ✓ [Lean-agile project, program, or portfolio management office.](#)

This subsection of the supplier's agile governance section is where the supplier proposes an approach to agile oversight. The supplier should establish a small, lightweight guiding coalition of lean-agile servant leaders to oversee the acculturation of the agile acquisition team, including acquirers and suppliers. This section describes the formation of a lightweight lean-agile center of excellence (LACE), agile portfolio, program, or project management office (PMO), or a similar governance board to ensure the smooth transition of the acquisition to a well-balanced set of lean-agile practices. The first major element is to decide the composition of the agile oversight team. It should contain at least one member from the acquirer's and supplier's teams, maybe two or three. More importantly, this team should consist of trained, certified, and experienced lean-agile coaches. It would be best if they were all experts in the same lean-agile framework the supplier is proposing to use for coordination purposes. For example, if the supplier selects the Scaled Agile Framework (SAFe), then the coalition should be comprised of certified SAFe Program Consultants (SPCs). This helps guarantee that the agile oversight committee is trained in not only lean-agile thinking, but the particulars of the SAFe framework itself. It's not uncommon for acquirers and suppliers to wing it a bit and adapt some SAFe practices incorrectly. For instance, in SAFe and Scrum, sprint, iteration, and quarterly program increment (PI) goals and objectives are devised bottoms up by the agile teams. However, it's not uncommon for acquisition programs to devise goals and objectives in the form of a 5-10-or-15-year integrated master schedule (IMS) and hand these to agile teams during quarterly planning. However, if the coalition consisted of SPCs, the degenerative practice of handing goals and objectives to teams in the form of schedules could be quickly eliminated or sunsetted at the earliest opportunity, and certainly not tolerated for very long. Another responsibility of the agile oversight coalition is establishing a lean and agile thinking mindset in the form of values and principles. Many agile methodologies and frameworks come with well established values and principles, such as Scrum and SAFe. However, the agile coordination coalition may insist upon doing a brainstorming session with the entire delivery team to devise a set of custom and unique values and principles specific to the agile business model and context of both the acquirer's and supplier's cultures and organizations as well as the agile business model (i.e., scope of the agile contract). The agile oversight coalition should also have a charter defining the roles and responsibilities of the coalition itself, how often it will meet, how it will make decisions and disentangle opposing views, and align the transformation objectives to both the acquirer's and supplier's goals and objectives from its agile business model. This assumes the acquirer's agile business model is well-aligned with its own organizational goals and objectives, which may not always be the case. Again, there aren't many acquirers or suppliers with deep lean and agile experience, not many leaders are trained and certified in a chosen agile framework, and even if coalition members are certified, they still may have diametrically opposing views of how to guide the agile acquisition transformation forward. It may be a little unnerving if the coalition has a SAFe, Scrum, LeSS, DevSecOps, and diehard integrated master scheduling (IMS) expert too. That's a sure recipe for an acquisition consisting of Keystone Cops. It can never be emphasized enough, that the agile coordination coalition must contain members from BOTH the acquirer and supplier. This ensures the acquirer can provide the necessary authorization for lean and agile practices and the acquirer can help keep the agile transformation on the right track too.

29. [Ag·ile·Trans·for·ma·tion](#) (*əj'əl • trāns'fər-mā'shən*) Change, renewal, conversion, alteration, metamorphosis; [To propose the formation, staffing, and use of a small team of coaches to guide the implementation of lean-agile principles and practices](#)

- ✓ [Lean-agile management office.](#)
- ✓ [Lean-agile center of excellence.](#)
- ✓ [Lean-agile strategic coaching team.](#)

This subsection of the supplier's agile governance section is where the supplier proposes an approach to agile transformation. A tried-and-true approach to both traditional and agile transformation is to propose some sort of initial maturity assessment of traditional or agile practices. With this gap analysis in hand, the transformation team can then devise a roadmap, course of action, or get-well plan for improving agile maturity. Of course, the weakness of traditional assessments is that they assess acquirers and suppliers against process and document heavy frameworks (i.e., they encourage the development of linear waterfall development, production of unnecessary processes and products, the development of big up-front plans, requirements, and architectures, etc.). In other words, traditional assessments instantly bog traditional or lean-agile transformations down in unnecessary WIP and large batch sizes which kill acquisitions dead from the start. Using this approach for lean-agile transformations also has its weaknesses as it focuses on individual teams, methodologies, practices, metrics, and tools. They are also training-intensive and essentially force the transformation team to stitch their own agile coordination framework together like a quilt. After a while, it sort of ends up looking like the Winchester Mystery House, if it even gets that far as acquirers tend to lose focus after a quarter or two and kill lean-agile transformations or revert to traditional frameworks amidst the chaos of a mystery mansion or house of mirrors. Therefore, the supplier should propose a cohesive agile framework from the shortlist of preapproved agile frameworks, i.e., Scaled Agile Framework (SAFe), Disciplined Agile Delivery (DaD), Scrum at Scale (S@S), Enterprise Scrum, Large Scale Scrum (LeSS), Recipes for Agile Governance (RAGe), Solutions for Agile Governance in the Enterprise (SAGE), Spotify, Lean Startup, Startup Way, etc. SAFe comes with a built-in transformation roadmap that consists of 12 broad stages (i.e., reaching the tipping point, train lean-agile change agents, train executives, managers, and leaders, create a lean-agile center of excellence, identify value streams and ARTs, create the implementation plan, prepare for ART launch, train teams and launch the ART, coach ART execution, launch more ARTs and value streams, extend to the portfolio, and accelerate). This isn't to say the SAFe implementation roadmap is foolproof and doesn't require a great deal of skill, which it naturally does. There's a bit of take-your-time-but-hurry-

up built into the SAFe implementation roadmap (i.e., you don't want to boil the ocean but you wanna heat up the hot tub in a hurry and get the party started). The strength of proven frameworks like SAFe is they have built-in mechanisms for business agility, lean portfolio management, large solution engineering, and program or team of teams level coordination (in addition to a well-thought out implementation roadmap). A key tenet of the SAFe implementation roadmap is that change comes last not first in SAFe (i.e., the SAFe implementation roadmap is geared toward cold-starting or getting agile teams up and running in a hurry, focusing on delivering value quickly, and letting the results speak for themselves). SAFe even comes with 14 built-in certification training courses to support its various elements. However, this isn't to say SAFe is complete, as there seems to be a great need for training people at all levels, especially acquirer and supplier leadership and management, in basic lean-agile thinking values, principles, and philosophy. Another pain point in SAFe is teaching basic teamwork values, principles, and philosophy, especially in the Western hemisphere where fierce individualism is prized, taught, rewarded, and reinforced. An underlying theme to this section is that the transformation team may be separate from the agile oversight team.

30. **Ag·ile · Scal·ing** (*ăj'əl • skā'ling*) Expand, extend, stretch, increase, enlarge; [To propose the application of an approach to ensure larger teams of teams consistently apply lean-agile methods, practices, and tools as single high-performing team](#)

- ✓ **Project, program, and portfolio consistency.**
- ✓ **How to establish high-performing team of teams.**
- ✓ **How to innovate at scale using lean-agile frameworks.**

This subsection of the supplier's agile governance section is where the supplier proposes an approach to agile scaling. This is where the supplier proposes principles, practices, techniques, training, metrics, and tools to reinforce the agile coordination approach. The supplier's agile coordination framework consists of practices to help teams of teams or groups of teams of teams perform as a single high-performing cross-functional team. However, this is easier said than done, especially in the fiercely individualistic Western hemisphere. Inconsistency is exacerbated if the agile governance coalition has diametrically opposing philosophies (i.e., some favor consistency, while others may favor good old fashioned competitive suboptimization). Uneven team performance is a direct result of the fierce Western individualism of the agile governance coalition itself, i.e., one may favor SAFe, another Scrum another DevSecOps, and yet another integrated master schedules (IMSs). Therefore, this is where the supplier proposes to remedy inconsistencies of suboptimizing teams. Yes, this involves cadence, synchronization, alignment, and consistency. The more cadence, synchronization, alignment, and consistency, the easier it is to establish a single high-performing team of teams. Therefore, this section may propose a consistent approach to business agility, lean portfolio management, large solution or systems of systems management, and program or teams of teams level management. Many agile frameworks directly provide mechanisms to synchronize the planning, ceremonies, vision, backlog, constraints, and communication of all of the teams in the portfolio, large solution, or acquisition program level team of teams. Again, these are not Keystone Cops, competing against one another, and iterating independently, with rogue product owners acting as individual chief executive officers going off of the deep end or out on a limb alone. Agile planning teams are more like a large symphony orchestra in perfect harmony, a synchronized swimming team, or a large marching band in perfect lockstep. So, the supplier's agile scaling section proposes how to ensure cadence, synchronization, alignment, and consistency of all agile teams of teams, along with their roles, ceremonies, practices, tools, metrics, etc. It's important to reinforce that consistency and cadence among individual agile ceremonies within agile teams is also part and parcel to creating high-performing cross functional teams of teams. But, exactly how will consistency be sought, promulgated, and achieved among many disparate teams of teams. First of all, the agile governance, transformation, and scaling teams have to agree that consistency is necessary to result in a single high performing cross functional team of teams. Then, it must propose mechanisms like training and retraining in values, principles, practices, tools, ceremonies, rules, metrics, etc. These elements must be measured, assessed, monitored, and coached. Monitoring can be a bit dicey at times, as individuals within agile teams are very good at purporting to follow consistent agile practices. For instance, one Scrummaster may devise an iteration plan for a team in advance, but not tell anyone. Another may form iteration, sprint, or program increment (PI) objectives for the team in advance, or within a few minutes of ending a multiple day planning event. Still, some Scrummasters may perform coarse user story planning with plenty of capacity, while yet another may do task level planning and aim for full utilization. It may be necessary to assign a pair of lean-agile coaches specifically to smoke out inconsistencies, propose mechanisms for aligning teams, and seek to normalize the performance of a team of teams or groups of teams of teams. This may seem a little antithetical to the notion of self-organizing teams, but some guardrails need to be established to keep groups of agile teams on the tracks.

31. **Ag·ile · Cul·ture** (*ăj'əl • kŭl'chər*) Organizational ideals, values, norms, laws, mores; [To propose the application of a process improvement approach to instill lean-agile thinking, principles, practices, frameworks, and tools as the organizational norm](#)

- ✓ **Portfolio, program, and project commitment.**
- ✓ **Middle management and shared services commitment.**
- ✓ **Group, team, and individual commitment to lean-agile thinking.**

This subsection of the supplier's agile governance section is where the supplier proposes an approach to establishing an agile culture. While the traditional approach to culture change is to train front-line teams in the values, principles, practices, tools, metrics, and philosophy of the new paradigm, this has very limited effects. In this approach, executives, business owners, directors, leaders, and middle managers are not committed to the change, but are committed to training front-line workers to the change. The reason this approach fails is because the minute something changes, typically a new traditional organizational change initiative or some other counter cultural behavior is exhibited, then the front-line workers will quickly abandon their commitment to the new paradigm. Let's say, an executive picks Scrum as the flavor of the day and trains the delivery team in Scrum, but then hires a new traditional director that begins promoting integrated master schedules (IMSs), business requirements analysis, and enterprise architectures. This immediately signals the delivery team that Scrum is no

longer necessary, some may revert to these degenerative practices, other may leave to find a truly agile organization, and yet others may become disillusioned, numb, and skeptical of organizational change at all. They'll just keep their head down and nose to the grindstone, while ignoring every new management change initiative that comes along. All of this is part and parcel to lack of leadership commitment to the new operating philosophy. Edgar Schein of Harvard noted in the 1960s that the key driver of organizational culture is the chief executive officer (CEO). That is, the CEO sets the tone for the organization. If the CEO is traditional, then the culture will be traditional. If the CEO is entrepreneurial, then the organization will also be entrepreneurial. If the CEO is a hardass, then the organizational culture will be hostile. If the CEO is congenial, then the organizational culture will be friendly. Therefore, the supplier should propose to build a leadership infrastructure to support not only lean-agile thinking, but the proposed lean-agile framework itself. This is also true of the acquirer's organization. Although suppliers rarely have direct control over the acquirer, they certainly can influence the acquirer to a great degree. Therefore, the supplier should seek to establish strong lean-agile leaders at all levels (i.e., program manager, key personnel, technical leads, agile governance, agile transformation, agile coordination, etc.). Again, part of this is establishing a joint acquirer-supplier agile governance coalition. This is where the lean-agile leadership messages come out. The messages should be loud and clear, the coalition should eat its own dog food, along with every member of the supplier's delivery team, and there should be no contradictory or degenerative traditional messages coming from the agile governance coalition, i.e., do lean portfolio management AND integrated master schedules (IMSs). A strong lean-agile vision should be in place, and leaders at all levels of the acquisition team should make it their personal agenda to promote that vision early, often, and at all times. Leaders should likewise seek to root out hybrid traditional practices and promote a culture of openness, communication, collaboration, transparency, trust, morale, innovation, creativity, teamwork, shared risks, rewards, and responsibilities. Furthermore, these should not be empty platitudes, but leaders at all levels should personally live up to these ideals. Do leaders insist upon being in on all team communications, then teams should also be in on all leadership communications. What's good for the goose is good for the gander. Do leaders insist upon emotional intelligence, then they should be exemplars when it comes to emotional intelligence. There are far too many lean-agile leaders without emotional intelligence.

32. **Ag-ile • As-sess-ments** (*ăj'əl • ə-sēs'məntz*) Weigh, measure, appraise, evaluate, diagnose; [To propose the application of a simple set of proven highly-cohesive lean-agile surveys, diagnostic tools, and instruments to measure overall performance](#)

- ✓ **Group, team, and individual assessments.**
- ✓ **Portfolio, program, and project assessments.**
- ✓ **Middle management and shared services assessments.**

Finally, this subsection of the supplier's agile governance section is where the supplier proposes an approach to establishing a system of agile assessments. The supplier should propose to use agile assessments at multiple points in the acquisition lifecycle. They might assess agile readiness, assess agile maturity, assess the major acquisition functions, assess the teams, and assess the frameworks, methodologies, practices, techniques, metrics, tools, and even value system. Perhaps, the supplier may propose an initial open-ended assessment of the acquirer's and supplier's leadership teams to gauge their goals and objectives. All of these initial assessments may feed into a broad implementation plan, help the supplier's agile transformation team understand the viability of rolling out the supplier's proposed agile framework, and determine the appetite for change. Again, there's no sense in conducting the assessment if the acquirer's and supplier's leadership teams are not driving the change. It's very difficult to do bottoms up change (i.e., like kicking against the goads). If the supplier's leadership team is willing to undergo a lean-agile transformation, then they may be willing to get the acquirer's leadership team to go along. However, the better scenario is for the acquirer's leadership team to initiate, insist upon, and drive the lean-agile transformation. That's the fundamental purpose of an agile contract statement of work (SOW), i.e., for an acquirer to signal to the supplier ecosystem that lean-agile leadership is required on this contract. Again, change comes from the chief executive officer (CEO), acquirer leadership teams often serve as the acquisition CEO, and whatever they say goes for the most part. For too many decades, acquirer leadership teams have been baking traditional language into the contract statement of work (SOW). This signals to the supplier base that traditional approaches are required and are in fact the superior acquisition delivery approach. It causes a chain reaction of events, and suppliers will invest millions of dollars in excess profits and cash reserves to undergo traditional decade-long transformations that drive their companies and their cultures to hell in a handbasket faster and faster at the supplier's expense. When acquirers take the time to form an agile contract statement of work (SOW) vs. a traditional one, this will begin signaling to the supplier base that it is now better to invest in lean-agile transformations vs. traditional ones. With more agile SOWs, suppliers will invest in agile frameworks, team-level agile methodologies, and other lean-agile principles, practices, roles, tools, techniques, and metrics. Once again, mature agile approaches like the Scaled Agile Framework (SAFe) come with a myriad of prepackaged assessments, not only to get started, but assess agile functions, along with growing agile maturity from quarter to quarter. Historically, it is very difficult if not impossible to institute traditional metrics, measurements, models, and performance assessments. However, agile assessments soften the blow of forcing complex mathematics on ordinary delivery teams, while giving the agile transformation team all of the insight they need to gather the requisite data and information to measure, gauge, and perform continuous improvement. There is a myriad of lean and agile assessments on the open market, most of which are highly inconsistent and skip whole parts of the agile value chain, however, as noted here, some mature agile approaches like the Scaled Agile Framework (SAFe) come replete with plenty of out-of-the-box agile assessments that can be employed with little excess energy, effort, expertise, training, or special certifications. Some traditional paradigms charge in excess of \$100,000 or more to use their multi-million-dollar assessment methodologies, while SAFe assessments are free, easy-to-use, and fast.

Agile SOW Summary

So, what have we learned in this short treatise on how to successfully develop agile contract statements of work (SOWs) to form proper acquirer-supplier relationships for the successful acquisition of innovatively new products and services? Well, we've learned that successful public and private sector acquisition of new products and services begins with a well-formed contract SOW. More importantly, we've learned that first and foremost, SOWs should be based upon the values, principles, and practices of lean and agile thinking. Why is this so? Well, because creating innovatively new products and services is inherently knowledge work, and often, information technology-intensive work. This means that market, customer, and end-user needs, requirements, and pain points are generally unknowable upfront. Therefore, WIP-limited iterative and incremental lean and agile frameworks are needed to tease out hidden, tacit, and inexpressible market, customer, and end-user needs, pain points, and requirements a little bit at a time (often in the form of business experiments). From this statement we learn that it's not a very good idea for acquirers to specify all of the new product and service requirements upfront, which often come in the form of 5-10-and-15-year-old (obsolete) integrated master schedules (IMSs), business requirements specifications, enterprise architectures, and other forms of traditional documentation. Instead, acquirers should alternatively ask suppliers to propose a lightweight lean and agile methodology or framework to jointly, cooperatively, collaboratively, and gradually tease out market, customer, and end-user needs. Therefore, SOWs should be agile contract SOWs, they should be lightweight, forego fixed-scope product specifications, and instead focus on lean and agile thinking principles, design thinking, user experience design, and, most importantly collaboration at all levels.

32 PRINCIPLES AND PRACTICES FOR A HIGHLY SUCCESSFUL AGILE CONTRACT STATEMENT OF WORK (SOW)

1. **AGILE BUSINESS MODEL**—Form a broad skeletal framework of the vision, scope, purpose, and constraints
2. **Agile Customers**—Identify the direct primary and secondary customers, market, or end-users of products and services
3. **Agile Value Stream**—Identify the sequence of beginning to end, cradle to grave, or concept to cash product lifecycle stages
4. **Agile Vision**—Identify a broad overall concept, artist's rendering, or as-is vs. to-be state for new products and services
5. **Agile Solution**—Identify the broad scope, boundaries, parameters, and constraints of new products and services
6. **Agile Roadmap**—Identify a broad sequence of new product and service hypotheses, business experiments, and outcomes
7. **AGILE FRAMEWORKS**—Identify a small set of proven highly-cohesive lean-agile scaling frameworks suppliers can use
8. **AGILE PLAYBOOK**—Propose the application, tailoring, and use of a proven highly-cohesive lean-agile framework
9. **Agile Mindset**—Propose the salient values, principles, and philosophical elements of selected lean-agile framework
10. **Agile Methodology**—Propose the application of a proven highly-cohesive lean-agile team-level methodology
11. **Agile Roles**—Propose a small set of general-purpose roles and responsibilities individuals on each team must apply
12. **Agile Ceremonies**—Propose a basic set of short, routine, and strictly time-boxed set of meetings each team must apply
13. **Agile Practices**—Propose a set of general best practices each team must apply to achieve quality and performance levels
14. **Agile Techniques**—Propose set of highly-specialized management or technical practices to achieve unique product aspects
15. **Agile Specialists**—Propose a set of specialized experts to guide the proper implementation of non-functional requirements
16. **Agile Metrics**—Propose a set of lean and agile quantitative and qualitative metrics and models to measure performance
17. **Agile Tools**—Propose a small ecosystem of tools to enhance agile team collaboration, performance, and outcomes
18. **AGILE COORDINATION**—Propose the application, tailoring, and prescriptive components of a proven agile framework
19. **Agile Analysis**—Propose proven lean-agile problem identification, analysis, and visualization for value stream mapping
20. **Agile Management**—Propose a set of proven lean-agile execution principles, practices, and tools such as Kanban
21. **Agile Architecture**—Propose a set of proven lean-agile architecture and design practices to support just-in-time delivery
22. **Agile Product Management**—Propose design thinking practices such as empathy, journey, and experience mapping
23. **Agile Planning**—Propose the application of a highly-cohesive lean-agile planning framework for developing quarterly plans
24. **Agile Dashboards**—Propose a set of lean-agile metric, model, and measurement dashboards to visualize performance
25. **Agile Content Management**—Propose a lean-agile application lifecycle management tool to plan and track a team's work
26. **Agile Infrastructure**—Propose using a low-cost commercial distributed information technology platform for use by teams
27. **AGILE GOVERNANCE**—Propose use of a proven lean-agile portfolio, system of systems, or transformation framework
28. **Agile Oversight**—Propose a portfolio, program, or project management office to promulgate lean and agile thinking
29. **Agile Transformation**—Propose a small team of coaches to guide the implementation of lean-agile principles and practices
30. **Agile Scaling**—Propose an approach to ensure teams of teams apply lean-agile methods as a single high-performing team
31. **Agile Culture**—Propose process improvement to instill lean-agile thinking, principles, practices, frameworks, and tools
32. **Agile Assessments**—Propose proven lean-agile surveys, diagnostic tools, and instruments to measure performance

It's pretty simple really, but it's taken us decades to reach this philosophical epiphany when it comes to public and private sector contract acquisition of innovatively new products and services. However, the state of public and private sector product and service acquisition contracting is in a rather dismal state, and although it exhibits some light at the end of the tunnel, it is not progressing fast enough (and is even moving backwards at times). The project management and systems engineering disciplines which began solidifying in the 1940s and 1950s with the advent of the electronic computer set us back decades in time, if not centuries. These traditional disciplines insisted upon multi-decade long linear waterfall lifecycles, extended requirements and architecture formulation stages, and large-batch big bang testing 10-20-or-30-years after contract start to verify and validate these over scoped processes and document intensive acquisition lifecycles. Oftentimes, this multi-decade long work was encapsulated in an ironclad integrated master schedule (IMS) replete with earned value management (EVM). Again, all-the-king's-horses-and-all-the-king's-men-couldn't-put-humpty-dumpty-together-again and the problem was that market, customer, and end-user requirements,

architectures, and designs could not be successfully predicted by a small group of analysts, architects, and designers. Technology simply evolves too fast, so do market needs, and, of course, the engineering economics of processing large batches of integrated master schedules (IMs), business requirements, architectures, designs, tests, hardware, software, and documentation are simply untenable. That is, the lean and agile mathematics of long queues, big batches, full-utilization, and overallocation of knowledge workers stops productivity dead in its tracks, which project management and systems engineering did not intend to do.

Traditional disciplines such as project management, systems engineering, and even software engineering was created to reduce the risk of creating innovatively new products and services (vs. increase it). Who knew? Well, mathematicians and operations analysts knew this by the 1950s, but it took them another 50 years to come clean and translate their complex statistical models into practical terms, resulting in the emergence of innovatively new lean and agile thinking approaches by the 1990s. The problem was that by this time, traditional thinking was so deeply embedded in the psychology of project managers and systems engineers that the notion of lean and agile thinking was not only antithetical to conventional wisdom, but downright heretical. That is, the notion that knowledge workers are not interchangeable cogs nor automatons that can be programmed to full utilization, batch size should be reduced, and hidden, inexpressible, and intangible market, customer, and end-user needs should be teased out a little bit at a time using small business experiments. In other words, do a little bit of upfront analysis, form a small hypothesis, develop a rapid small minimum viable product (MVP), quickly expose it to a market, customer, or end-user, and collect quantitative and qualitative data to evaluate the hypothesis. Sometimes, changes to business experiments are not readily apparent, and many business experiments have to be conducted to identify, converge upon, and reveal true market, customer, or end-user needs. In fact, 95% of large-scale business experiments yield no appreciable or measurable improvement in market, customer or end-user satisfaction. This is congruent with data from the 1990s illustrating that in excess of 95% of new product and service functions were never used at all among large systems created in the 1980s and 1990s when computer system complexity exploded.

Okay, well all of this talk about traditional vs. lean and agile thinking, small batches vs. large batches, mathematical queuing theory, hidden inexpressible (psychological) market needs, gradually teasing out these needs using small business experiments, and traditional thinkers overloading systems, queues, and knowledge workers with too much WIP is a bit abstract. What does this have to do with public and private sector contract SOWs? Well, both traditionalists as well as lean-agile proponents have been encouraging contract and procurement personnel to insert their favorite new product and service development methodology into requests for proposals (RFPs), statements of work (SOWs), proposal responses, and the contracts themselves. That is, traditionalists want to see oodles of integrated master schedules (IMs), split second earned value management (EVM), linear systems engineering lifecycles, heavyweight processes and documents, and extended business requirements and architecture phases in order to reduce the risk of public and private sector acquisitions. Better yet, if the acquirer will attach the IMs, business requirements, and architectures directly to the SOW, which some are intent upon doing. Lean and agile proponents on the other hand want to see some mention of lean and agile thinking, values, frameworks, methods, practices, techniques, metrics, and tools. Advanced lean and agile proponent do not want to see large batches of obsolete 5-10-or-15-year-old IMs, business requirements, and architectures attached to SOWs, was they want to use lean and agile approaches to collaboratively tease out hidden market, customer, and end-user needs a little bit at a time in the form of small business experiments and minimum viable products (MVPs). And, in fact, the global market is leaning more and more towards collaborative acquirer-supplier relationships.

A real danger in statement of works (SOWs) is that they are a very specific legal instrument used to establish a contract between an acquirer and a supplier but is not the contract agreement itself (nor should a legal contract agreement refer directly to an SOW). That is, an SOW, especially in the public sector, is a procurement, acquisition, or source selection document. Typically, a public sector agency is under obligation to ensure an open, fair, and competitive procurement for goods and services (innovatively new products and services). A request for information (RFI) may be the first step in the publicly facing acquisition process, because it may take a public sector agency many months, years, and perhaps decades to assemble the final statement of work (SOW), conduct the source selection, and contract with one or more suppliers (i.e., look how long it took DoD to award JEDI). In fact, many public sector acquisitions are very much like icebergs (i.e., 90% of the acquisition is below water, behind closed doors, or takes place in private where there is a lot of under-the-table wheeling and dealing). By the time an SOW is made public, many suppliers have already gathered critical business intelligence about the acquirer, formed their capture teams, and drafted their proposals. In other words, most suppliers don't stand a very good chance of winning a proposal if they are not already in on the ground floor. But, again, the important part is that most suppliers are not legal experts, the SOW is merely a legal instrument to establish an acquirer-supplier contract, and your average bear doesn't know the difference. Supplier capture teams often fashion highly creative out-of-the-box proposals and may get selected on the basis of low-cost or best value (i.e., having an innovative proposal). This of course is where the challenge begins if the acquirer fashions a poorly worded over scoped or fix scoped SOW.

When and if a supplier is selected, the proposal and the final terms and agreement constitute the legal contract, not the SOW. Unfortunately, supplier capture teams are quickly disbanded or get assigned to the next round of SOWs and the proposal is quickly forgotten (i.e., the legal contract between the acquirer and supplier). Proposals are often hidden from the delivery team as closely guarded intellectual property and the delivery team naively assumes the SOW is the contract (which it is not). Delivery teams often decompose the original SOW into new business requirements and place them one at a time on an integrated master schedule (IMS) for the developers to implement. Again, the problem with this approach is that the proposal constitutes the supplier's approach to which the acquirer has agreed (i.e., entered into a legally binding contract to obtain the products and services stated in the supplier's proposal vs. the original SOW). In other words, the acquirer's SOW is the wish list, while the proposal is what the acquirer has agreed to purchase and the supplier has agreed to provide. Worse yet, is that oftentimes, acquirers neither care about the SOW nor supplier's proposal, as they were merely legal instruments to spend money on what they really wanted from the start (i.e., collaborate with the supplier to uncover hidden inexpressible and intangible market, customer, and end-user needs not captured in the acquirer's SOW nor supplier's proposal). All the while, traditional project managers carry around the SOW for the entire period-of-performance as though it is the legally binding contract. If this hypothesis is correct, then the SOW should have merely stated, "Use a collaborative approach to uncover my hidden inexpressible needs!"

A PRIVATE SECTOR CONTRACT STATEMENT OF WORK (SOW) OUTLINE

This Statement of Work ("SOW") is made on [EFFECTIVE DATE] between [CLIENT NAME] ("Client") and [SUPPLIER NAME] ("Supplier") for [PROJECT NAME] pursuant to the [NAME OF MASTER AGREEMENT] dated [DATE OF MASTER AGREEMENT] ("Master Agreement").

1.0 Overview

- 1.1 **Scope and Objectives.** The Scope and Objectives of this SOW are as follows. [DESCRIBE SCOPE AND OBJECTIVES].
 - 1.2 **Period of Performance.** The period of performance for this SOW will be [PERFORMANCE TERM] commencing on [PERFORMANCE START DATE]
 - 1.3 **Place of Performance.** The services will be performed at [LOCATION OF SERVICES].
 - 1.4 **Statement of Work Type.** This is a [Time and Materials - Not to Exceed ("NTE")] [Fixed Price] Statement of Work.
- 2.0 **Plan and Project Schedule.** The overall project plan and project schedule are described in Attachment A.

3.0 Personnel

- 3.1 **Supplier Personnel.** The personnel furnished by the Supplier for performing efforts under this SOW will have adequate qualifications to perform the work.
- 3.2 **Key Supplier Personnel.** The following personnel furnished by Supplier are critical to the success of the completion of this SOW.

4.0 Supplier Tasks and Deliverables

4.1 Supplier Tasks and Responsibilities

- a. The Supplier will provide the necessary resources to plan, implement, and manage work necessary to meet the project objectives and prepare the Deliverables.
- b. The tasks and responsibilities required of Supplier by this SOW consist of; but are not limited to, the following.

4.2 **Supplier Deliverables.** Supplier shall prepare and deliver the following Deliverables to the Client in accordance with the requirements of this SOW and the dues dates ...

5.0 **Limitations on Supplier Authority.** The Supplier shall have no authority to:

- 5.1 Authorize deviations from the project plan or project schedule;
- 5.2 Grant a time extension; or
- 5.3 Approve substitute Deliverables.

6.0 **Client Tasks and Responsibilities.** The following Tasks and Responsibilities required of Client are dependencies to the Supplier's ability to perform its obligations ...

7.0 Project Management

- 7.1 **Governance.** The Supplier and the Client will apply Project Management Institute project management and governance processes.
- 7.2 **Project Managers.** The Supplier and the Client will each appoint project managers who shall serve as the single point of contact for purposes of communications.
- 7.3 **Detailed Project Plan.** The Supplier is responsible for developing and maintaining, with input from the Client, all related scheduling detail itemizing project tasks ...
- 7.4 **Progress Meetings.** The Client and the Supplier will support and provide representation at progress review meetings, as required ...

7.5 Reports

- a. **Monthly Progress Reports.** The Supplier shall prepare a monthly "time on task" summaries including a description of the time spent, by Supplier resources ...
- b. **Final Report.** The Supplier shall prepare a final summary report upon completion of the work performed and results obtained for the entire contract period of performance.

8.0 **Change Management Process.** The Client Project Manager and the Client Purchasing Contact will coordinate all changes to the SOW through Change Orders ...

9.0 Completion Criteria

- 9.1 **Payment for Completed Deliverables.** The Client will authorize payment for complete Deliverables only after complying with the Acceptance Process specified above.
- 9.2 **Payment for Incomplete Deliverables.** In the event any Deliverables are not performed in accordance with this SOW because of factors due to Client's failure ...

10. **Fees.** 11. **Payment Terms.** 12. **Payment of Invoices.** 13. **Expenses.**

U.S. DEPARTMENT OF DEFENSE (DoD) CONTRACT STATEMENT OF WORK (SOW) OUTLINE

1.0 **Scope.** This Section includes a brief statement of what the SOW should cover. The scope paragraph defines the breadth and limitations of the work to be done.

2.0 **Applicable Documents.** Military handbooks, government instructions, service regulations, technical orders, and policy letters, etc.

3.0 **Requirements.** Specific work tasks are called for in SOW Section 3. These tasks, developed to satisfy program needs, are essentially the contractor work requirements. A well-written SOW has the following attributes:

- a. Specifies requirements clearly to permit government and offerors to estimate the probable cost and the offeror to determine the levels of expertise, manpower, etc.
- b. States the specific duties of the contractor in such a way that the contractor knows what is required and can complete all tasks to the satisfaction of the contract, etc.
- c. Written so specifically that there is no question of whether the contractor is obligated to perform specific tasks.
- d. References only the absolute minimum applicable specifications and standards needed. Selectively invokes documents only to the extent required, etc.
- e. Separates general information from direction so that background information and suggested procedures are clearly distinguishable from contractor responsibilities.
- f. Avoids directing how tasks are to be performed and states only what results are required.

SOW Do's – Best Practices

- Select a competent team with an experienced team leader.
- Exclude “how to” requirements since the offeror to ensure cost effectiveness.
- Use the program Work Breakdown Structure (WBS) to outline the required work effort.
- Set SOW objectives in support of the Acquisition Plan (AP), if applicable.
- Explicitly define the tailored limitations of all standards and specifications cited.
- Exclude design control or hardware performance parameters b
- Educate personnel with respect to acquisition streamlining.
- Give priority to commercial items over specification items.
- Give priority to commercial practices as a means of acquisition.

SOW Don'ts - Antipatterns

- Order, describe, or discuss Contract Data Requirements List (CDRL) data.
- Invoke, cite, or discuss a Data Item Description (DID).
- Specify technical proposal criteria or evaluation factors.
- Establish a delivery schedule (May include significant milestones for clarity).
- Specify design control parameters or the performance of hardware.
- Impose Government format on the contractor.
- Over-specify in order let the contractor select the best methods.
- Invoke in-house management instructions.
- Use the SOW to establish or amend a specification.
- Invoke handbooks, service regulations, technical orders, or any other non-DoD document.

Further Reading

- Amirkhanyan, A. A., Kim, H. J., & Lambright, K. T. (2012). *Closer than arm's length: Understanding the factors associated with collaborative contracting*. Binghamton, NY: Binghamton University.
- Blum, M. (2014). *TechFAR: Handbook for procuring digital services using agile processes*. Washington, DC: U.S. Digital Services.
- Book, M., Gruhn, V., & Striemer, R. (2016). *Tamed agility: Pragmatic contracting and collaboration in agile software projects*. Berlin, Germany: Springer International.
- Carter, A. B. (2017). *DoDI 5000.02: Operation of the defense acquisition system*. Washington, DC: Department of Defense.
- Crenshaw, D. (2008). *The myth of multitasking: How doing it all gets nothing done*. San Francisco, CA: Jossey-Bass.
- Denning, S. (2018). *The age of agile: How smart companies are transforming the way work gets done*. New York, NY: Amacom.
- Dolfing, H. (2017). *Agile project management: Budgets and contracts*. Zurich, Switzerland: Data Solutions GmbH.
- Eckfeldt, B., Madden, R., & Horowitz, J. (2005). *Selling agile: Target cost contracts*. Agile 2005, Denver, Colorado, USA, 160-166.
- Edwards, I., Bickerstaff, R., & Bartsch, C. (2015). *Contracting for agile software development projects*. London, UK: Bird & Bird.
- Frydinger, D., et al. (2021). *Contracting in the new economy: Using relational contracts to boost trust and collaboration in strategic business relationships*. London, UK: Palgrave-MacMillan.
- Gottschalk, P., & Solli-Saether, H. (2006). *Managing successful IT outsourcing relationships*. Hershey, PA: IRM Press.
- Hayford, O. (2018). *Collaborative contracting*. Sydney, Australia: Pricewaterhouse Coopers.
- Holsoe, T. (2012). *K03 standard contract for long-term IT projects based on an agile method*. Copenhagen, DK: Kammeradvokaten.
- Home, J. (2017). *Collaborative contracting: Better practice guide*. Canberra, Australia: Department of Defence.
- Hopp, W. J. (2008). *Supply chain science*. Long Grove, IL: Waveland Press.
- Hopp, W. J., & Spearman, M. L. (2008). *Factory physics*. Long Grove, IL: Waveland Press.
- Hubbard, D. W. (2007). *The failure of risk management: Why it's broken and how to fix it*. Hoboken, NJ: John Wiley & Sons.
- Hubbard, D. W. (2010). *How to measure anything: Finding the value of intangibles in business*. Hoboken, NJ: John Wiley & Sons.
- International Standards Organization. (2017). *Collaborative business relationship management systems: Requirements and framework (ISO 44001)*. Geneva, Switzerland: International Organization for Standardization (ISO).
- Knapp, J. (2016). *Sprint: Solve big problems and test new ideas in just five days*. New York, NY: Simon & Schuster.
- MacLean, P. D. (1973). *A triune concept of the brain and behavior*. Toronto, CA: Univ. of Toronto Press.
- MacLean, P. D. (1990). *The triune brain in evolution: Role in paleocerebral functions*. New York, NY: Plenum Press.
- Mahadiwala, R. (2018). *Agile contracts: Get more flexibility with agile contracts*. Troy, MI: Atos-Syntel.
- Modig, N., & Ahlstrom, P. (2018). *This is lean: Resolving the efficiency paradox*. Stockholm, Sweden: Rheologica.
- Moran, A. (2016). *Valuing agile: The financial management of agile projects*. Norwich, UK. TSO Shop.
- Nicoletti, B. (2018). *Agile procurement (two volumes): Adding value with lean processes*. Cham, CH: Palgrave-MacMillan.
- Opelt, A., Gloger, B., Pfarl, W., & Mittermayr, R. (2013). *Agile contracts: Creating and managing successful projects with scrum*. Hoboken, NJ: John Wiley & Sons.
- Pink, D. H. (2009). *Drive: The surprising truth about what motivates us*. New York, NY: Riverhead Books.
- Pound, E. S., Bell, J. H., Spearman, M. L. (2014). *Factory physics: How leaders improve performance in a post-lean six sigma world*. New York, NY: McGraw-Hill Education.
- Reinertsen, D. G. (2009). *The principles of product development flow: Second generation lean product dev.* New York, NY: Celeritas.
- Reynolds, B., & Heffernan, E. (2016). *Agile & digital services delivery: Acquisition execution guide*. Alexandria, VA: Grant Thornton.
- Rico, D. F. (2013). *Agile contract manifesto*. Retrieved March 28, 2013, from <http://davidfrico.com/agile-vs-trad-contract-manifesto.pdf>
- Rico, D. F. (2016). *The 10 attributes of successful teams, teamwork, and projects*. Retrieved September 26, 2016 from <http://davidfrico.com/teamwork-attributes-2.pdf>
- Rico, D. F. (2018). *Lean & agile contracts: 21 principles of collaborative contracts and relationships*. Retrieved June 29, 2018, from <http://davidfrico.com/collaborative-contract-principles.pdf>
- Rico, D. F. (2018). *Using SAFe 4.5 to transform a \$200 million U.S. healthcare portfolio*. Retrieved November 19, 2018 from <http://davidfrico.com/safe-case-study-ii.pdf>
- Rico, D. F. (2019). *Business value of lean leadership: 22 Attributes of successful business executives, directors, and managers*. Retrieved April 27, 2019, from <http://davidfrico.com/rico19a.pdf> or <http://davidfrico.com/lean-leadership-principles.pdf>
- Rico, D. F. (2019). *Piloting the scaled agile framework (SAFe) in a top 10 U.S. energy firm*. Retrieved May 15, 2019, from <http://davidfrico.com/x-safe-case-study-iii.pdf>
- Rico, D. F. (2020). *32 principles and practices of highly successful SAFe implementations*. Retrieved February 16, 2020, from <http://davidfrico.com/safe-principles.pdf>
- Rico, D. F. (2020). *Business value of lean thinking: Capitalizing upon lean thinking principles to rapidly create innovative products and services*. Retrieved January 29, 2020, from <http://davidfrico.com/rico20b.pdf> or <http://youtu.be/wkMfaPAXO6E>
- Rico, D. F. (2020). *Using large solution SAFe in a high-profile U.S. civilian public sector agency*. Retrieved January 29, 2020, from <http://davidfrico.com/y-safe-case-study-iv.pdf>
- Rico, D. F. (2021). *32 principles and practices for maximizing the ROI of SAFe program increment (PI) planning*. Retrieved March 14, 2021, from <http://davidfrico.com/safe-roi-principles.pdf>
- Rico, D. F. (2021). *Business value of organizational agility*. Retrieved March 26, 2021 from <http://davidfrico.com/rico21a.pdf> or <http://davidfrico.com/value-of-business-agility.pdf> or <http://youtu.be/HOzDM5krtes>
- Rico, D. F., Sayani, H. H., & Sone, S. (2009). *The business value of agile software methods: Maximizing ROI with right-sized processes and documentation*. Ft. Lauderdale, FL: J. Ross Publishing.
- Schrage, M. (2014). *The innovator's hypothesis: How cheap experiments are worth more than good ideas*. Boston, MA: MIT Press.
- Stenbeck, J., & Jans, K. (2014). *Agile government contracting: Expert guidance for department, command and agency leaders, contracting officers, procurement professionals, program and project managers, and prime contractors*. Spokane, WA: GR8PM, Inc.
- Strand, K., & Karlsen, K. (2014) *Agile contracting and execution (ACE)*. Oslo, Norway: PROMIS AS.
- Tesfaye, L. (2017). *Agile contracts: Performance work statement (PWS) template*. Washington, DC: U.S. GSA.
- Thomke, S. (2013). *Experimentation matters: Unlocking the potential of new technologies for innovation*. Boston, MA: HBS Press.
- Walle, J. (2010). *PS2000 standard contract for agile software development: An agile tool for the execution of IT projects*. Oslo, Norway: Norwegian Computer Society.
- Weitzenboeck, E. M. (2012). *A legal framework for emerging business models: Dynamic networks as collaborative contracts*. Cheltenham, UK: Edward Elgar Publishing.

CASE STUDIES OF HIGHLY SUCCESSFUL AGILE CONTRACT STATEMENT OF WORK (SOW) IMPLEMENTATIONS

- **Public Sector Government Agency.** *During the heyday of agile frameworks around 2013, a multi-billion U.S. government agency began its journey towards using large team of teams agile frameworks. This agency started using Scrum by 2009, but not without pain. This agency developed a lot of information technology intensive systems. Many of them were data processing intensive systems due to the extremely large data sets they processed. In addition to more than their fair share of traditional brick-n-mortar data centers, they also developed a lot of software dating back to the 1960s and 1970s. By 2005, this agency was steeped in traditional methods, spent decades documenting requirements and architectures, and built immensely over scoped systems. So much so, that they simply could not keep up with the pace of technological change. By the time they began adopting Scrum in 2009, a typical commercial smart phone and its apps could perform most of the system functions created by this agency at a cost of trillions of dollars over many decades, and do it better, faster, and cheaper. This agency was so steeped in traditionalism, that only very senior people with decades of experience were allowed to form voluminous requirements specifications and integrated master schedules (IMS) and simply assign requirements one-by-one to a programmer without much communication, collaboration, or creativity. Programmers were simply taught to turn off their brain, ask no questions, and try not to think very hard. This was certainly true by 2009, when most of its computer systems had already been built (i.e., the heavy lifting had already been done) and most of the programming tasks were simply bug fixes to billions of lines of legacy code. Its developers had a very hard time adopting Scrum in 2009 due to its self-organizing nature, team sprint planning events, and programmer's inability to devise their own sprint goals, tasks, and estimate their own effort (rather than have all of that done for them as they had for the prior 50 years). Although there was talk of insurrection at this agency, as developers simply did not like Scrum nor its imperative to think for themselves as they had become automatons, this agency moved into larger team of teams agile frameworks by 2014. About that time, the agency constructed an agile frameworks SOW to help operate and maintain its entire legacy base of front-end systems. In other words, it evolved an ecosystem of graphical user interfaces (GUIs) to operate to its back-end systems to process large data sets. The SOW referred to a very specific team of teams agile framework. There was no hybridization nor traditionalism in sight. Unfortunately, this agency translated its high-power distance culture into the pure agile framework SOW and assigned all epic, capability, feature, and user story writing to only its most senior managers belonging to the government agency, while relegating computer programming (implementation) of the preplanned and preassigned user stories to the supplier's computer programmers. They were very careful to select an insider who was closely aligned with their traditional culture and reject all suppliers whom they could not control.*
- **Public Sector Healthcare Agency.** *This agency is one of the largest public sector healthcare agencies. It spends billions of dollars developing, operating, and maintaining thousands of nationwide information systems. Many were written in COBOL on mainframes and it was steeped in a culture of integrated master schedules (IMSS) and linear waterfall lifecycles. Lead times for new systems were measured in decades, so over 90% of its personnel and budget were dedicated to doing something other than deliver working software. This included mountains of user documentation, brochures, advertisements, business requirements specifications, homegrown information systems methodologies, and homegrown application lifecycle management tools. It processed large data sets with billions of individual data points. Much of the last five decades was spent developing thousands of incompatible distributed systems spread across the country, so a lot of their effort went into harvesting, translating, and migrating data between disparate nationwide systems. This was largely a manual process, and it required a lot of brainpower to analyze billions of individual datapoints for compatibility issues when moving data between systems. So much so, that many of its personnel were mathematicians, physicists, and other extremely intelligent analysts with master's degrees and PhDs (basically an army of automatons). Even its own homegrown application lifecycle management tools were over scoped and immensely complex and any given processing screen may contain dozens of input fields with menus or pick fields with dozens of options each. Around 2017, this agency decided it wanted to migrate as many of its legacy data warehouses to an on-premises cloud and do it quickly. Therefore, they assembled a hybrid traditional-agile SOW that contained references to hundreds of traditional and a few agile methods and frameworks. An innovative supplier decided to propose a cohesive team of teams framework for overseeing the creation of the on-premises cloud, migration of legacy systems, and development of front end systems for getting the data out. Upon winning the contract, the capture team immediately moved to the next proposal, the proposal was kept secret, and the delivery team converted the SOW into an integrated master schedule (IMS). Unfortunately, the SOW contained more traditional than agile terms and the delivery team abandoned its commitment to the team of teams framework proposed by its capture team. The acquirer didn't mind, since it didn't want to use the supplier's proposed team of teams agile framework, but instead wanted to hire as many low cost administrative personnel as they could along with a few cloud experts. Although the acquirer's division never really abandoned its commitment to traditional practices such as integrated master schedules (IMSS) and business requirements, other larger divisions within the acquirer's agency deepened their commitment to team of teams agile frameworks and spent billions of dollars through the use of hybrid traditional-agile SOWs, although many other contracts insisted upon using them (dragging their suppliers kicking and screaming out of traditionalism all of the way).*
- **Private Sector Energy Firm.** *This was one of the largest energy firms in the world. Being in energy, it was a veritable monopoly and had no major competition for its customers, as they could not easily switch to another utility company. There was little motivation to develop innovative new products and services, except to gobble up more energy firms through mergers and acquisitions. It centralized its information technology staff, although there were many disparate acquisitions, to decimate its information technology budget. Since there were few initiatives for innovatively new products and services, its information technology staff spent its time working from home, working on community projects, and learning about the rest of the corporation (i.e., learning new skills to get out of information technology). Its corporate CIO decided to build a new ecosystem of mobile apps so ordinary consumers could order its services (become clients), pay their monthly bills, check the status of their accounts, and the energy firm could distribute frequent advertising messages through their client's smart phones. At first, a user experience design team was established to develop empathy, journey, and user experience maps. Then opportunities for improvement were identified and point solutions or mobile apps were proposed to close these gaps. Professional UX personnel were hired to develop detailed wireframes and they now had a mountain of them constituting over scoped epics and features. The CIO was happy, because there were experience maps and wireframes plastered all over the walls representing perceived innovation. An agile SOW was established to standup Scrum teams to begin implementing the over scoped wireframes. While there were plenty of developers in the acquirer's organization, most of these personnel were busy with community projects. First, the supplier proposed a few dozen Scrum teams, appointed product owners and Scrummasters, and assembled a few onshore and offshore software development teams. Most of the Scrum team leaders were local, while most of the programmers were overseas. There were many difficulties standing up the agile application lifecycle management tool and other computer programming tools, so that seemed to slow the teams down for a year. One high-performing cross functional Scrum team worked day and night implementing the wireframes, and its product owners labored to squeeze in as many features into a two-week sprint as possible, by forcing offshore programmers to work 60 hours a week and analyzing every minute of their day. The other teams were still trying to get their first development sprint going. Then, a larger team of teams agile framework was proposed to get all of the teams planning and sprinting together. The acquirer said to ignore the high-performing team and get the lower performing teams going, which they did. Within three iterations, the lower-performing Scrum teams leap frogged the high-performing Scrum team while working only 40 hours a week because they properly groomed their backlogs and built in a little excess capacity for uncertainty. The other teams who refused to participate never developed any working code at all.*