32 Principles and Practices for Maximizing the ROI of SAFe Program Increment (PI) Planning

Tra·di·tion·al • Plan·ning (*trə-dĭsh'ə-nəl • plă'nĭng*) is the practice of forming a detailed, sequential, long-term plan for implementing a complex set of business requirements generally one layer at a time. For instance, if you're implementing a building, the steps may include acquire permits; purchase land; clear property; lay foundation; erect frame; install plumbing and electrical; build walls and windows; finish walls, ceilings, and floors; inspect, certify, and occupy building; operate and maintain building; dispose of building at end of life; etc. The same principle holds true for information systems, the steps may include, acquire contract; purchase tools and infrastructure; install and configure platform; build network and operating system; acquire and configure middleware, security layer, and frameworks; design and build applications; test, validate, and accept system; write library of documentation; train users, operate, and maintain system; dispose of system at end of life; etc. The fundamental difference is that buildings are simple visual collections of objects that are easily tracked in a construction schedule (and may take a few months to complete with some level of budgeting accuracy). An information system is composed of millions of invisible parts we call software and may take a few decades and even billions of dollars to complete with NO level of accuracy. In information systems, small teams of planners divine thousands of business requirements meant to represent the needs of mythical users, create multi-decade long architectures and designs, and codify all of these into 15,000+ line integrated master schedules (IMS) as some sort of apocalyptic prophecy. As we've painfully learned over the decades, technology simply evolves too fast to capture in a decade long IMS, user needs are notoriously difficult to predict, and 15,000+ line IMSs are almost NEVER successful!

Ag·ile • Plan·ning (*ăj'əl • plă'nĭng*) is the practice of creating lightweight, skeletal short-term plans for small teams to create innovatively new products and services with high levels of risk, uncertainty, and unpredictability. Agile planners understand that market, customer, and end-user needs exist as hidden, inexpressible, and intangible psychological needs which change quickly; technology is innovating at an exponentially accelerating pace and is a moving target; and developers must rapidly tease out customer needs a little at a time using a small series of business experiments sort of like waving cheese under a mouse's nose in order to delight our customers with innovatively new products and services with the shortest possible lead and cycle time. Agile planners understand that market needs are cyclical, change, and rapidly evolve, so short-term market demand must be satisfied in order to optimize revenues, pivot, and quickly move to the next wave. In order to surf above these market waves instead of sinking to the bottom of the ocean with traditional planning, agile planners must quickly design some features, story maps, and user stories; swarm and implement them together as a team because no individual knows it all; deliver a small minimum viable product (MVP) to a real customer or end-user; collect leading measures and evaluate them; rinse and repeat while we're still above water; and adapt, pivot, and devise new short term plans before the next wave. In order to achieve success with agile planning, the plans themselves should NOT be detailed, they should only be for a few weeks or months, detailed business requirements and integrated master schedules (IMSs) are NOT required, and heavyweight processes and documents are simply waste. Agile planning is like packing light for a quick overnight trip vs. taking three trunk loads of underwear to grandmas for the weekend.

SAFe • Pro·gram • In·cre·ment • Plan·ning (*sāf • prō'grăm' • ĭn'krə-mənt • plă'nĭng*) is an agile planning framework for scaling up its values, principles, practices, tools, and metrics to multiple teams implementing larger and more complex ecosystems of innovatively new products and services. Individual agile teams MAY reach a level of maturity and performance necessary to repeatably apply its values, principles, and practices for rapidly implementing innovatively new products and services. However, it's notoriously difficult for enterprises, organizations, businesses, portfolios, systems of systems teams, and large product and programs and products to get more than one team to reach the performance levels of a single optimizing agile team. Enterprises often ask more than one team to apply lean and agile frameworks, watch them go like the Keystone Cops, laugh at agile planning proponents, and simply revert to traditional planning approaches that have NEVER worked since the 1950s. SAFe recognized this gap, and devised a set of values, principles, practices, tools, and basic metrics for teams of teams called Agile Release Trains (ARTs) to create short-term, lightweight, skeletal, adaptable, and goal-oriented plans for quickly implementing a series of short term business experiments to tease out intangible customer and end-user needs. However, not just create a short term-adaptable plan for teams of teams building a loosely coupled and highly cohesive ecosystem of innovatively new products and services, but actually have all teams iterate together like a single high-performing agile team. That is, have teams of teams look, act, and behave like small high-performance agile teams instead of Keystone Cops bumping heads together at every turn. In doing so, SAFe teams quickly field a series of small minimum viable products (MVPs) resulting in market success for buyers and suppliers.

SAFe was designed from the ground up so that teams of teams can iterate together like a single small high-performing team to successfully create innovatively new products and services. In order for organizations, enterprises, firms, and even public sector agencies to realize the promise SAFe, they must address three distinct phases: (1) Follow the SAFe Implementation Roadmap for standing up Agile Release Trains or ARTs, (2) Conduct fast, efficient, and effective SAFe planning events for creating successful near-term lightweight agile plans in a consistent and repeatable manner, and (3) Consistently execute SAFe ceremonies as a team of teams to successfully deliver high-quality, value-adding, and innovatively new products and services. This whitepaper is going to address step number two (i.e., conducting fast, efficient, and effective SAFe planning events), because quickly creating a smart SAFe plan is critical to successful development execution, implementation, and outcomes. SAFe is designed to address the problems of executing agile values, principles, practices, metrics, and tools at scale—In a team of teams context vs. Keystone Cops. Therein lies the fundamental challenge! Adult humans, especially in the Western world, are notoriously bad team players. Therefore, it's a bit antithetical to ask a large group of Western adults (Keystone Cops) to behave like a perfectly synchronized symphony orchestra. In addition to the challenge of fierce individualism in the Western hemisphere at all levels (i.e., customers, leaders, program managers, coaches, team leads, developers, etc.), new product and service developers tend to apply traditional thinking to create plans that are too detailed, over scoped, expensive, inconsistent, and meant to measure individual performance. Therefore, this whitepaper will attempt to remedy this misconception and get SAFe teams off on the right foot from the start in order to maximize the business value or return on investment (ROI) of using SAFe (and minimize pain, failure, and humiliation).

A Simplified Quick-Start Strategy for Rapidly Deploying the Scaled Agile Framework (SAFe) to Maximize ROI

1. **Be firm, but flexible**—Be goal-oriented, form a personal vision, and sell it to stakeholders fast (strategy, roadmap, schedule, goal, etc.).
2. **Set up a small team**—Form a two to three person mini-LACE, APMO, RTE team, or coaching team to help you (don't try this alone).
3. **Be disciplined**—Apply SAFe/Scrum ceremonies (PI/iteration planning, refinement, standups, demos, retros, retrospectives, etc.).
4. **Scope customer needs**—Perform an initial open-ended interview of key program, project, technical, and agile team leaders.
5. **External coalition**—Form an external group of colleagues, mentors, and coaches for occasional advice, best practices, and ideas.
6. **Dark matter**—Be on the lookout for gray areas, uncertainty, ambiguity, and sticking points in SAFe that are not plainly obvious.
7. **Near-term timeline**—Aim to kickoff SAFe and Scrum ceremonies within 90 days to get a visible value adding transformation going.
8. **ART size**—Select three to five local medium-impact Scrum teams to keep cost, risk, and feasibility manageable (~50 to 70 people).
9. **Visioning**—Develop and socialize a quick-n-dirty ART canvas quickly with key stakeholders, including business owners and managers.
10. **Training**—Follow the SAFe roadmap (train Leading SAFe, SAFe for scrum masters, SAFe for product owners, SAFe for teams, etc.).
11. **Agile ALM tools**—Standup toolset before SAFe kickoff (purchase, configure, dashboards, train, etc.)—Rinse-n-repeat tool coaching.
12. **Agile metrics**—Select basic SAFe and Scrum metrics early (capacity, velocity, burndown, planned/actual, program predictability, etc.).
13. **Product management**—Establish small product management team (to solution product roadmap, backlogs, and features FOR teams).
14. **ART cadence**—Establish synchronized cadence, schedule, and rhythm for all or most SAFe and Scrum events (for consistency).
15. **F2F planning**—Kickoff first SAFe planning event in an in-person large OPEN big room or ballroom-style venue (if possible).
16. **Remote planning**—Apply remote SAFe planning/ceremonies for distributed teams if necessary (Zoom, Teams, Skype, WebEx, etc.).
17. **Initial SAFe plan**—Have SAFe teams produce a physical or digital program board during breakout sessions (keep-it-simple).
18. **Use program board**—Use physical or digital program board for draft and final plan reviews, ART Sync, PO Sync, SoS, etc.
19. **SAFe ceremonies**—Hold regular SAFe ceremonies such as SAFe planning, ART sync, PO sync, SoS, system demos, I&A, IP, etc.
20. **Focus on consistency**—Establish, monitor, and enforce consistent SAFe and Scrum ceremonies, practices, metrics, and tool-use.
21. **SAFe assessments**—Routinely apply basic out-of-the-box SAFe team and program assessments (don't skip, tailor, or replace).
22. **Performance reports**—Data mine SAFe assessments for quick-and-dirty (reliable) measurement data (when tools inevitably fall down).
23. **Limit the WIP**—Viciously and continuously limit the WIP at all levels (transformation, product management, planning, execution, etc.).
24. **Just-do-it**—Come hell or high water, do all of these steps in first 90 days and scale up later (to quickly achieve measurable success).

Footnote. Organization change comes LAST, NOT FIRST in SAFe and Agile—People will change when they SEE RESULTS—Iterating every two weeks with Scrum guarantees results—PI Planning will build trust, transparency, and motivation (however, there will always be curmudgeons who believe in traditional thinking or simply do NOT want to see SAFe and Scrum succeed—These often go hand-in-hand).

Skills, Characteristics, and Attributes of SAFe Release Train Engineers (RTEs) to Yield Fast/Early SAFe ROI

1. Strong or exemplary leadership skills.
2. Exemplifies openness and transparency.
3. Servant leader - Eats their own dogfood.
4. Understands and promotes value of teamwork.
5. Trained and certified SAFe SPC, SSM, or RTE.
6. Has instant personal credibility and face validity.
7. Above average cat herder, facilitator, and motivator.
8. Direct experience standing up and executing ARTs.
9. Natural multi-tasker to do many activities at one time.
10. Can differentiate between traditional and lean thinking.
11. Favors lean thinking over traditional planning activities.
12. Understands design thinking, story mapping, and limiting WIP.
13. Strong SAFe and Scrum facilitation skills (super scrum master).
14. Knowledge of Scrum ceremonies (and expert at concise timeboxing).
15. Doesn't skimp on basic SAFe ceremonies but doesn't overdo them either.
16. Strong hands-on experience with agile lifecycle management (ALM) toolsets.
17. Able to manage high stress levels, demanding customers, and stodgy developers.
18. Great traditional planning skills to organize SAFe planning events, ceremonies, etc.
19. Ability to calibrate coaching plans for individual Scrum teams moving at different speeds.
20. Great customer/people skills, confident, thick-skinned, even keeled, and no-nonsense attitude.
21. Ability to maintain a heavy workload for long periods of time (doesn't easily burnout under duress).
22. Likes sitting in endless face-to-face or remote meetings from dusk to dawn without breaks (i.e., 24x7).
23. Ability to identify and work behind scenes remove real or perceived impediments to keep the ball rolling.
24. Serves as SAFe product manager in absence of a formal product management team to produce roadmaps/backlogs.
25. Exemplary program/project management skills (e.g., contracts, SOWs, CDRLS, IMSs, budgets, hiring, appraisals, etc.).
26. Strong knowledge of basic Scrum and SAFe metrics and enthusiasm for applying and utilizing routine SAFe assessments.
27. Highly motivated self-starter that can cold start, plan, and execute SAFe PI planning ceremonies in traditional environments.
28. Strong creative visualization, collaboration, communication, emotional intelligence, people skills, and is a master of soft skills.
29. Excels at navigating fierce traditional organization politics and is extremely intelligent at planning many moves well in advance.
30. Establishes small, but powerful coalition or SAFe leadership team and values coaching of all customers, managers, and teams.
31. Understands power of lean thinking, slowing down to speed up, viciously limiting WIP, sustainable pace, and work-life balancing.
32. Is a true believer in lean, agile, and SAFe mindset, is in it for the long haul, stays the course, and isn't fickle about SAFe or next fad.

SAFe Planning Principles & Practices

1. Up·front • Train·ing • & • Prep·a·ra·tion (*ŭp-frŭnt • trā'nĭng • ănd • prĕp'ə-rā'shən*) Form, arrange, instruct, assemble, preparation; To lay the basic groundwork or foundation for using SAFe practices

   ✓ Create upfront plan.
   ✓ Train all SAFe ART participants.
   ✓ Follow SAFe implementation roadmap.

   It's important to perform a modicum of upfront training and preparation to get a SAFe rollout off the ground, going on the right foot, and off to a fast start. Obviously, it's important to follow the SAFe Implementation Roadmap as closely as possible, but certainly don't overdo it. Finding that goldilocks zone between just-enough, just-in-time training and preparation and too much of a good thing is kind of challenging to do sometimes. One critical step is to have an open-ended interview to smoke out the scope, scale, size, and intent of the SAFe transformation. That is, identify the program or product team's pain points, gauge their needs, and quickly determine their experience levels with lean, agile, SAFe, etc. From there, key courses like Leading SAFe or even SAFe for Government can be offered to the customers as well as the program or product management's team. It's important to note that the lean, agile, and SAFe community is moving away from (acquisition or contract) programs and projects (with a definitive beginning and end) to development value streams for continually conducting business experiments and evolving ecosystems of interrelated new products and services. Therefore, we may refer to a product team instead of a program or project from time to time. It's important for the lead SAFe coaches to be trained, certified, and experienced SPCs in order to guide the initial SAFe rollout or transformation. While it's important to create more SPCs at this point, it's sort of like the blind leading the blind if there are no currently certified SPCs with experience and enthusiasm for guiding SAFe rollouts. At some point, it is necessary for the initial SAFe coaches to handoff the transformation to internal SPCs for continuous improvement, maturity, and adoption. Scrum masters, product owners, architects, and the teams themselves should also be trained and certified in SAFe as well just before the SAFe rollout. While it would be better if individual teams had prior experience with Scrum or Kanban to aid in the initial SAFe rollout, this is not entirely necessary as SAFe training provides enough basic instruction to cold start teams with basic lean, agile, and SAFe practices. Everything identified here can be done remotely, efficiently, and inexpensively without all of the usual face-to-face investments. Target small to medium-sized ARTs or product teams instead of trying to nuke an entire portfolio or business enterprise with SAFe all-at-once. Getting off on the right foot with SAFe is just such an important critical success factor to ignore without attempting to boil the ocean.

2. Man·age·a·ble • Train • Size (*măn'ĭ-jə-bəl • trān • sīz*) Small, easy, doable, compact, uncomplicated; To constrain a SAFe ART to a size no smaller or bigger than necessary to deliver value-adding features

   ✓ Create tightly scoped Epic MVPs.
   ✓ Build a small to medium-sized ART.
   ✓ Plan to deliver value early and often.

   One of the best-kept secrets is to keep the SAFe ART or product team size as manageable as possible. That is, make a lean and mean fighting machine. Oftentimes, lean, agile, and SAFe transformation coaches want to nuke the whole enterprise or strategic portfolio like information technology operations all at once. Or, transformation consultants want to build an oversized ART or Solution Train (ART of ARTs) with 300 to 700 people a quickly as possible. This is basically a no-no! The best approach is to carve out a tightly scoped, highly cohesive slither, slice, or thread of a value stream consisting of three to seven small-to-medium-sized Scrum or Kanban teams. The smaller, leaner, or meaner, the better! Information technology, chief information office (CIO) functions, or other strategic portfolios often consist of dozens of teams. Furthermore, there's a temptation to nuke every man, woman, and child with SAFe (down to the janitor). To add insult to injury, there are often critically important teams within strategic portfolios building transformation point solutions themselves. That is, once a strategic portfolio begins a digital transformation, a few tiger teams are quickly established to begin building state-of-the-art IT solutions in an attempt to get these strategic portfolios out of the ice age. They are often sort of like scout sniper teams. By the time the transformation coaches can conduct open ended interviews and initial assessments, form roadmaps, and begin architecting product teams or ARTs, it's far too late to incorporate the initial tiger teams into the SAFe transformation process. That is, once the tiger teams get started and restart development after 10 to 20 years of carving out business requirements, UML use-case diagrams, and UX wireframes on the side of their caves, middle managers don't want anyone to interrupt their progress (even if they are off to a bad start, going down a tangent, or burning the midnight oil 60 to 80 hours a week on traditional death marches). Basically, traditional middle managers don't know the different between a small, highly effective lean-agile Scrum or Kanban team and a death march. That's okay, what this means is that there are dozens of teams placed on the backburner roarin' and ready to go. Resist the temptation to roll out all of these middle to bottom tier teams onto the product team or ART. Select the next three to six teams in the queue, even if the strategic vision or burning platform is yet to be established for them. You'll be surprised what small product teams or ARTs of medium performers can do.

3. Vi·cious·ly • Lim·it·ed • WIP (*vĭsh'əs'lē • lĭm'ĭ-tĭd • wĭp*) Slop, dross, waste, excess, materials; To obsessively minimize the materials and work necessary to quickly deliver value-adding features

   ✓ Teach and reinforce Lean Thinking.
   ✓ Keep ART scope and size lean and mean.
   ✓ Learn to deliver value quickly with limited WIP.

   The key to high performance is lean thinking, lean startup cycles, and rapid low-cost business experiments. In lean-speak, viciously limit your scale, scope, size, and WIP! It never ceases to amaze people that lean, agile, and SAFe teams speed up

by slowing down; dramatically reducing complexity or batch size; and eliminating traditional tools like integrated master schedules (IMSs), enterprise architectures, business requirements, etc. Instead, have product teams or ARTs perform small business experiments, hypothesis tests, epic-MVPs, and other informal probes. Traditional thinkers believe they can read their customer's minds, predict the future, and specify over scoped solutions that a small army can successfully build in 5, 10, or 15 years! Furthermore, all that's necessary is to divinate a stack of business requirements, a 15,000+ line integrated master schedule (IMS), and torment people with earned value management (EVM). Analyzing some freshly harvested quivering livers may not hurt at this point! Of course, this is all nonsense! That's not science, that's alchemy—All of it! First, admit no human can predict the future, read their customer's mind, or determine where the market will be in a few months, years, or decades. Certainly, do your homework, collect some analytical data, and form some hypothesis statements. Then design a series of small experiments, rapidly and inexpensively solution them without killing your developers with overtime, have your customers or end-users try them out, and collect some real feedback (measurement data). This is called a "business experiment" and you probably need a control group in there to determine if the treatment can be attributed to the measurement losses or gains. Now that's science! Talent helps to quickly build high-quality realistic business requirements, but even teams of mediocre performers can simulate the effect of a prima donna or two. Persistence is also necessary to perform a stream of small business experiments to tease out the data. It's more of an art than a science to declare victory, stop, or continue experimenting before pivoting to a new set of hypotheses. The key here is don't over allocate, aim for full-utilization, overload iterations or program increments with too much scope, and certainly don't load them down with unnecessary processes, documentation, and tools (and don't measure individuals). Simply form a few simple, quick, and low-cost hypothesis, create simple story maps, swarm on solutions, get feedback fast, and rinse-and-repeat forever at a sustainable pace.

4. Lean • Prod·uct • Man·age·ment (*lēn • prŏd'əkt • măn'ĭj-mənt*) Goods, produce, services, commodity, merchandise; To apply lean design thinking to create consumer-grade market-oriented products and services

   ✓ Use lean and design thinking.
   ✓ Maximize user experience design.
   ✓ Limit WIP, emergence, and empathy.

   Lean product management is an emerging discipline, not only critical to SAFe product teams or ARTs, but is often missing entirely. Part of that is that it is a new discipline, while the other part is that it's simply misunderstood. At least in the public sector, RFPs, SOWs, contracts, business requirements, enterprise architectures, work breakdown structures, and integrated master schedules (IMSs) are mistaken for (lean) product management. That is, customer stakeholders or their representatives took the time to form detailed SOWs or business requirements and jam ALL of these into overloaded contracts. The only thing left to do in the traditionalist's mind is to apply lean, agile, and SAFe to drink from the firehose and implement all of the requirements on a firm fixed price contract or schedule. Other than the fact that it's simply too much WIP, the worst sin is that the requirements are incorrect, invalid, or not needed at all. Many times, business requirements are written 5, 10, or 15 years before a contract is released, and it will take another 5, 10, or 15 years to solution them, so the requirements will be 30 years out-of-date IF they are implemented at all. This is certainly NOT product management NOR lean product management. So, if you skipped traditional or lean product management on the way to implementing SAFe, STOP EVERYTHING, you missed a critical piece of the puzzle to success! First, establish a small lean product management team, TRAIN and certify them in SAFe Agile Product Management (APM), and have an experienced SAFe APM coach lead the team through its first few cycles. This often includes using design thinking; creating empathy, customer, journey maps, etc.; building a lean canvas for a product or small ecosystem of highly cohesive products; forming lean product roadmaps; identifying a few product hypothesis tests in the form of SAFe epic, capability, or feature statements; and feeding a small set of strategic hypothesis tests to a small to medium-sized SAFe product team or ART. This queue of hypothesis tests is called a portfolio, solution, or ART-level program (or product) backlog. This is NOT to be confused with team-level Scrum product backlog. This usually results in an instant civil war, because SAFe SPCs cannot easily distinguish between an ART and team level product backlog. An ART-level program or product backlog contains all of the hypothesis tests for a SINGLE product, family, or ecosystem of products. Each lean-agile Scrum or Kanban team usually builds ONE feature or aspect of that product, not the entire product itself!

5. Cross • Func·tion·al • Fea·ture • Teams (*krôs • fŭngk'shə-nəl • fē'chər • tēmz*) Team, pair, squad, group, assembly; To form small teams of people with the skills necessary to deliver value-adding features

   ✓ Form small, cohesive teams.
   ✓ Ensure all skills are included in team.
   ✓ Ensure teams can deliver vertical features.

   Each lean-agile team on a small to medium-sized SAFe product team or ART should have all of the skills necessary to create a vertical slice of functionality. That is, the lean-agile team must have the capability to solution one of the hypothesis tests from the SAFe ART-level program or product backlog. Perhaps, the ultimate product is a new cloud-based mobile app for buying low-cost medical insurance for a self-employed consultant with no healthcare benefits. Maybe one of the features is account services so customers can sign up and create accounts. Perhaps another feature is identify a low-cost plan with the requisite benefits for the consultant's family. The next feature may be purchase a plan and arrange for payments. Another feature may be manage claims and disputes. Yet another feature may be distribute newsletters, plan benefits, and other plan changes. Finally, the last feature may be close out existing plans, resolve final payments, and discontinue the account when it is no longer needed. In this scenario, lean product management, in collaboration with one of the lean-agile teams has agreed to prioritize the hypothesis test for identifying low-cost plans with the requisite benefits, budget, and healthcare needs for the consultant's small family (ANONYMOUSLY)! Remember, empathy and journey maps have already determined the market is chock full of honey pots to capture people's phone number and email addresses to spam you with high-cost plans for the next

10 years. The lean-agile team must have cloud, network, operating system, database, middleware, GUI, and UX knowledge ALL ON ONE TEAM (vs. a separate team for each horizontal layer). This eliminates unnecessary dependencies between horizontal layers that result in contract, financial, functional, political, and sociological barriers to fast and successful solutioning. A SINGLE team with all of these horizontal specialties must work together under ONE product owner and scrum master to build a vertical functional feature slice in one or two weeks for evaluation by a live customer. Remember, it's only a hypothesis test, NOT the final function (gotcha!). Of course, it must be a high-quality, high-fidelity test. The hypothesis test must be evaluated by multiple consultants and feedback must be collected. At this point, the lean product management team in collaboration with the lean-agile team must decide whether to continue, stop, or pivot. This isn't to say horizontal shared services teams aren't needed to create a broader architectural runway of epic, capability, feature, and story enablers.

6. Build • Com·pe·tent • Ag·ile • Teams (*bĭld • kŏm'pĭ-tənt • ăj'əl • tēmz*) Skilled, capable, qualified, talented, endowed; To prepare all cross-functional teams in a SAFe ART to rapidly deliver value-adding features

   ✓ Ensure all ART teams are SAFe teams.
   ✓ Build more than one high-performance team.
   ✓ Distribute and balance high-technical performers.

   Each lean-agile sub-team on a small to medium-sized SAFe product team or ART should perform a vital SAFe function in a technically competent way. Simply put, each team should perform a vital, necessary, value-adding (often-vertical) function! There should probably be a lean-agile program management team (at least on public sector acquisition contracts). There should be a lean-agile product management team to apply design thinking, create empathy and journey maps, identify lean product roadmaps, and build ART-level product or program feature (hypothesis) backlogs. There should be a Lean-Agile Center of Excellence (LACE) or small coalition of lean-agile strategists at the ART product or program level. There may be a team of competent SAFe coaches. A product or system architecture team creates the vision for an architectural runway. The majority of the ART consists of cross functional feature teams to quickly solution hypothesis tests in 2-to-12-week iterations or increments. Of course, there may be a few exploratory, architectural runway, or shared services teams to establish, maintain, and even polish the train tracks for the cross functional feature teams. However, there is NO dross; each team performs a vital, visible, and value adding function; each team is as small as possible; all members are trained, certified, and self-actualizing (attaining to self-mastery); and, most importantly, the SAFe ART is NOT upside down (which is typical in most public and private sector industries. Upside down? What does that mean? It often means the ratio of (middle) management teams to cross functional teams is 10:1—In other words, most people are creating WIP, requirements, creating non-value adding processes and documents, micromanaging feature teams, and not solutioning hypothesis tests. Only the lean product management team should be establishing hypothesis tests. You don't need a small army of middle managers to micromanage the cross functional feature teams. Program managers administer the contract; product managers establish the backlog of feature (hypotheses); coaches train, assess, and facilitate ART-level performance improvements and course corrections; cross functional feature teams solution hypotheses; and a small number of collaborative horizontal architecture and shared services teams build and provide vital service products for the entire SAFe ART. In other words, there should be more value adding developers than non-value adding middle managers. Everyone rolls up their sleeves; everyone gets their hands dirty!

7. Team • Lev·el • Pre·plan·ning (*tēm • lĕv'əl • prĭ-plă'nĭng*) Ready, draft, sketch, prepare, compose; To develop a preliminary or draft team-level program increment plan in-advance of SAFe planning

   ✓ Create lightweight PI plans.
   ✓ Begin capacity planning very early.
   ✓ Collaborate with product management.

   Cross functional feature teams, architectural runway teams, shared services teams, and even SAFe coaching teams should collaborate with the lean-product management team. That is, representatives from each of the teams should participate in lean product management to apply design thinking, construct empathy and journey maps, identify lean product roadmaps, and fill up and prioritize the ART-level product or program backlog with features (hypotheses). There should be no surprises, lean-product management should rely on each of the team's expertise, individual teams should proactively help to design features (hypotheses), and prioritize ART-level product or program backlogs. With this product or program backlog feature intelligence in-hand, individual lean-agile teams should apply program or product increment backlog refinement for each of their teams. For instance, the lean product management coalition may collectively decide to construct a feature hypothesis test for a video streaming service similar to YouTube. It's acceptance criteria may include a 1 terabyte video size, 100 megabit per second streaming rate, a 99.9999% reliability rate, and the ability to start, pause, and stop videos midstream for later playback. Although this feature hypothesis test may be for the NEXT product or program increment, the cross functional feature team to which it belongs may begin preplanning to solution this hypothesis well BEFOREHAND. This may include commercial or OTJ training and skills, buying tools, configuring tools, getting access rights, and even pre-planning or prepopulating the product or program or team board with user stories. The first user story may be get OTJ training, the second create a demo spike, the third perform a design review, the fourth do an early demo, and the final one prepare for early market and customer testing. There are several illustrative points to this scenario: (1) there should be a lean-product management coalition; (2) cross functional feature teams should be a part of the coalition; (3) lean-agile teams should know what's coming; (4) lean-agile teams should do some pre-planning; (5) lean-agile teams should NOT be starved, overloaded, or surprised by features (hypotheses); (6) product or program increment planning should consist of productively fleshing out lightweight pre-plans; and (7) team-level product owners, scrum masters, and other team members should create sustainable plans that achieve the goals of a hypothesis test (e.g., gathering measurement data from customers, regardless of whether it is good or bad).

8. Per·form • Sto·ry • Map·ping (*pər-fôrm' • stôr'ē • măp'ĭng'*) Chart, sketch, roadmap, blueprint, architecture; To create a

cohesive business requirements architecture, functional flow, story board, or wireframe

- ✓ Obtain features in-advance.
- ✓ Collaboratively solution features.
- ✓ Create lightweight story maps in-advance.

Story mapping is one of many techniques lean-agile teams may use for team level pre-planning. Generic preplanning may consist of a simple sequence of steps like get training, configure tools, construct spike, perform design review, complete initial solution, show early demo, or finalize solution. However, story mapping is slightly more sophisticated than a simple linear risk reduction plan. For one thing, the simple plan leaves a lot to the imagination. You know what they say, you should never surprise your manager or customer. There's nothing wrong with evolutionary architecture and emergent design. For one thing, they're creative, encourage out-of-the-box thinking, and result in waste-free innovative designs. Furthermore, if the lean product management team STOPS or PIVOTS, little is lost. In that vein, story mapping also suggests a slightly more thoughtful business architecture or design. Let's say the feature (hypothesis) is an Internet outage mobile app. That is, any consumer can monitor the current state of the Internet to find downed links, ISPs, cloud services, and other bottlenecks. This may be a helpful if you can't login to your corporate network, favorite email client, or other cloud service. Instead of assuming you've made a mistake, got locked out, or didn't pay your bill, you may want to do a quick check of current Internet outages. You may determine, AWS is down again, that's why I can't get to YouTube, E-Bay, or Twitter. It's sort of like checking the weather. In this scenario, the lean-agile team may want to build a story map. The activities consist of creating accounts, displaying outages, customizing reports, getting notifications, and cancelling services. The lean-agile team may identify 5 to 15 user stories for each of these activities, prioritize them, and review them with lean product management and architecture to identify the MVP. They may determine only one user story is necessary for end-users to evaluate the hypothesis (better yet, it's only 5 story points) and can be done by two developers in a day. The rest of their time could be spent swarming with other teams, refactoring and reducing technical debt, or training for the next go around—Quite a sustainable pace. IF the end-users like the solution and ask for more features, then the story map can be revisited, refined, and reprioritized. The lean-agile team may be able to perform two or three more unplanned hypothesis tests in the same product or program increment. That's quite a good way to deliver value-adding features fast! A little lean-agile story mapping-based preplanning goes a long way!

9. Sim·plest • Pos·si·ble • Tools (*sĭm'plĭst • pŏs'ə-bəl • to͞olz*) Computer, instrument, automation, application, workflow system; To use intuitive, easy-to-use, automated workflow tools with the least possible overhead

- ✓ Use simple and inexpensive tools.
- ✓ Capture as little information as possible.
- ✓ Don't use tools for detailed traditional planning.

A key and counterintuitive best practice of lean-agile teams is to use simple tools. In the early days, simple physical visual aids such as white boards, flip charts, and post-it notes on the wall were all that was needed to plan a two-week iteration, 90-day program increment's worth of work, or an entire multi-quarter lean-agile product or project release plan or roadmap. If we go back to the beginning of formalized lean practices, Toyota pinned up all of their short, medium, and long-term production plans on walls (oftentimes in a dedicated conference or war room called an Obeya). Its leaders could meet at regular intervals such as daily, weekly, bi-weekly, monthly, quarterly, or even annually; examine the current plan; make any updates; and continue refining their production plans (on a physical, visible, and highly visual wall). This makes sense for Japanese who place greater emphasis on visual stimulation, graphical data, and body language. Obeyas provide other benefits, like only high-level strategic and tactical plans could be created since there wasn't enough room in the war room for a 15,000+ line integrated master schedule (IMS). Therefore, the plans were SIMPLE, VISUAL, and could be immediately grasped by anyone. Japanese are highly consensus oriented and collaborative people, so teamwork is greatly prized in Japan. The entire team gathers around the Obeya, collaboratively designs and debates it in real-time (politely), it is updated and tracked, and it is regularly communicated and immediately visible to EVERYONE. There is NO HIDING lean plans in Japan. Conversely, (automated) IMSs are so complex; it takes an accountant, mathematician, or scientist to design and understand them; they're easily hidden in complex IMS tools that require DBAs to operate; and only the high IMS Western priest views the divine oracle (IMS). Since the advent of Western lean-agile methods in the mid-1990s, firms created vast ecosystems of automated life cycle management (ALM) workflow tools, use them to dump 15,000+ line IMSs into them and only the high ALM priest knows how to write SQL queries to get the data out of them. However, just like IMSs, it's garbage-in-garbage-out, humans are simply unable to predict the future, and 99% of the data in an IMS is wrong the minute it is codified. What's the bottom line? Lean-agile teams must resist using complex IMS and ALM tools for creating detailed long-term plans, bring the lean-agile plans out-of-the-closet and onto the walls, create simple visual plans, and gather around as a team (not an individual high-SQL priest).

10. Vi·su·al • Col·lab·o·ra·tion • Tools (*vĭzh'o͞o-əl • kə-lăb'ə-rā'shən • to͞olz*) Open, graphic, explicit, obvious, pictorial; To utilize graphically-intensive planning and collaboration workflow tools to simplify and communicate data

- ✓ Use physical information radiators.
- ✓ Create physical team iteration boards.
- ✓ Minimize use of detailed analytical workflow tools.

Visual tools and aids also work best for operating lean-agile teams in addition to mere simplicity! Once again, a whiteboard, flipchart, and post-it notes on a wall work best. They also serve to limit the WIP, because of the limited physical space. Another important point is to plan with goals and objectives vs. implementation details. That is, each of the post-its should be a goal-oriented statement on a post-it note, index card, or even a bullet (vs. a detailed technical implementation plan). This is where lean-agile teams get in-trouble, because they know how to speak, think, and write in technical implementation steps

(vs. goal-oriented market, business, or customer language). Teach lean-agile technical teams how to perform GOAL-oriented visual planning and you'll have plenty of left-over space on the whiteboard, flipchart, or wall. Unless of course, you have a left-brained traditional project manager or coordinator in the room who wants to fill all of the available space with goals and objectives, ensure full allocation and utilization, and get every hour, minute, and second out of every developer each day. If you have to use an automated tool, ensure it's a simple, visual, collaborative, unstructured, and right-brained brainstorming tool. Many first-generation automated whiteboards simply didn't work very well, but they've come along way now. Oftentimes, people had to take photographs of whiteboards, email them around, and then erase the boards to continue brainstorming. This was the quickest way to defeat because no one ever looked at the dark fuzzy photos anyway. Now, technologies have improved, teams can put post-its on large touch screen displays, many people can do this physically or virtually, and, best of all, it's anonymous. So, this limits the control of the tool by one domineering micromanager. Of course, this also necessitates the use of a trained and certified facilitator to guide the collaboration process using a lean-agile planning framework like Scrum, Kanban, SAFe, Design Sprints, and Design Thinking approaches. The important point is to use visually intensive collaboration boards, ensure visual icons and graphics are used to capture data (like physical or virtual post-it notes). Follow lightly structured brainstorming and innovation frameworks like Scrum, Kanban, Design Sprints, Design Thinking, SAFe, Story Mapping, etc. And, be sure to remember that the output is a small set of design goals for teams with limited WIP vs. detailed plans for 100% utilization of individuals. Simple visualizations are great for normalizing collaboration and communication.

11. Com·mer·cial • Cloud • Ser·vic·es (*kə-mûr'shəl • kloud • sûr'vĭs'əs*) Web, online, Internet, networked, commercial; To use innovative, publicly-available, and commercial Internet-based collaboration tools and services

- ✓ Use simple visual collaboration tools.
- ✓ Use commercial cloud workflow tools.
- ✓ Minimize use of analytical workflow tools.

In many cases, the best simple visual collaboration tools for creating and tracking lean-agile plans are available as low-cost and often free commercial cloud services such as Mural. People simply navigate to Mural anonymously; communicate with VOIP services like a smartphone, Skype, or Zoom; and a facilitator guides the small team through the loosely-structured steps of an innovation framework like Scrum, Kanban, Design Sprints, Design Thinking, SAFe, Story Mapping, etc. The facilitator may start by showing the team around Mural. It's quite simple to use but does require a little practice to drive. In some cases, the facilitator may prepare a few story boards in advance for exercises such as ice breakers, skills identification, brain storming, dot voting, training, lean-agile innovation exercises, and actual plan construction. Once again, the strength of Mural is that it is low-cost for the facilitator, free for the team, and quite simple to use. More importantly, it's highly visual and can be used from the office, home, or other remote location. It's quite unstructured and good for collaborative brainstorming exercises and creating lightweight lean-agile plans. Notice we said the word, LIGHTWEIGHT. This is NOT a 15,000+ line integrated master schedule (IMS), nor would you want one in the first place! Its not a highly structured workflow tool like Jira, ADO, CLM, etc., but its great for creating the lightweight plans that feed into these tools (if you really need them at all in the first place)! Other visual collaboration tools like iObeya have direct real-time APIs into ALM tools like Jira. That is, teams can use their fingers to create post-its on large touchscreen television monitors, which get recorded as Jira issues, tickets, epics, capabilities, features, user stories, and even tasks. Vice versa, teams can enter issues into Jira which pop up as visual icons on iObeya television touch screens. Parts of the team can be in multiple global locations gathering around touchscreens, while other specialists may be sequestered at their home with Jira watching issues show up in backlogs, correcting them, and making their own additions and modifications too. While low-cost commercial cloud services like Mural have been around for years, many lean-agile teams and specialists have not had the luxury of using these tools. Oftentimes, they're former traditional project managers steeped in using left-brained analytical IMS and ALM tools and can't comprehend the purpose or value of using unstructured Internet-based commercial cloud visualization services for creating simple visual lean-agile plans.

12. Al·ter·na·ting • Lo·cal • & • Re·mote (*ôl'tər-nā-tiŋ • lō'kəl • ănd • rĭ-mōt'*) Vary, change, switch, rotate, exchange; To alternate between local face-to-face and remote SAFe ceremonies such as PI planning

- ✓ Initial F2F SAFe planning.
- ✓ Alternate remote SAFe planning.
- ✓ Always include all SAFe ART participants.

SAFe ceremonies are designed to be held in-person, face-to-face, or live. This is reinforced the agile principle of, "The most efficient and effective method of conveying information to and within a development team is face-to-face conversation!" That is, the most effective means of collaboration consist of rich, high-context face-to-face conversations (both formal and informal). Yes, facilitated, highly structured lean-agile planning and other brainstorming meetings are very effective. However, critical conversations and trust-building bonding occurs during meals, watercooler conversations, bathroom breaks, and even texting and emails between formal ceremonies. Trust takes time, and although highly structured meetings do build quick trust, deep trust takes many iterations, program increments, and years to form (in between the seams). The global workforce is distributed now, telecommuting is the norm, and global pandemics are a danger now. Therefore, remote SAFe planning events have come front and center. The textbook approach to virtual teams is to have the initial few SAFe planning events in-person to the maximum extent possible, have one in the middle of a typical 18-month roadmap or release, and then have the ART or product team come together near the end for a final system demo, celebration, and send off. If you have the luxury, simply alternate F2F and remote, with the initial SAFe PI planning event in a collocated fashion. Recent global pandemics forced SAFe ARTs or product teams to hold remote SAFe planning events all of the time. At first, this seemed like a daunting task, heavy-lift, or practical impossibility, but humans are remarkably resilient and we found a way. It was little awkward at first, as we all learned how to optimize the use of Skype, Zoom, Microsoft Teams, WebEx, and other cloud-based video

conferencing services. SAFe ARTs and product teams have now mastered the “art,” pun intended, of not only conducting remote SAFe planning events efficiently and effectively, along with the rest of the SAFe ceremonies, but even exceeded the performance of face-to-face teams. Some of these best practices include routine daily standups to maintain social and psychological connectiveness, but iteration planning, backlog refinement, demos, retros, and other ceremonies as well. Teams can schedule collaboration sessions to problem solve as well, including heads down working time too. One best practice is to break the entire team down into very small teams of two or three “pairs” to work together on a daily basis.

13. Big·Room • Plan·ning • Ven·ue (*bĭg-rōom • plă'nĭng • vĕn'yōo*) Space, chamber, ballroom, meeting room, conference room; To use a large-enough virtual or physical meeting space to accommodate the entire SAFe ART team

- ✓ Use a large room for SAFe PI planning.
- ✓ Bring everyone together to openly collaborate.
- ✓ Minimize cubicles, breakout rooms, and other obstacles.

When organizing planning events at the SAFe portfolio, large solution, or ART-level product or program level, it’s best to have everyone gather and plan TOGETHER in a single large room! This way everyone can meet, collaborate, socialize, plan, gel, and build trust together! Everyone can hear customers and business owners explain visions; solution and product managers explain roadmaps; and architects explain technical constraints. That is, EVERYONE on the Solution and ARTs hears the same message, meets and sees everyone, and can plan and collaborate together. We prefer large round tables or circles of chairs for individual teams. Ideally, teams stand around whiteboards, flip charts, walls, or touch screens to create their plans. Hopefully, product owners and scrum masters have facilitated a little preplanning with high-level story boards, objectives, constraints, and capacity planning allocations. Team members should have participated in preplanning events, so any story maps, boards, or skeletal plans should NOT be a surprise to them. At this point, lean-agile teams have heard the visions, product plans, and technical constraints; they now know their capacity, skills, and resources limitations by who and who is not present in the big room; and they can begin fleshing out the high-level story maps and boards with some additional details (but not too much more). They determine their operating capacity for the next quarter, sequence iteration boards, allocate user stories, determine delivery dates, and identify critical dependencies or risks. They finalize high-level iteration and program increment goals in customer or business language and assemble draft and final PI plans. Teams also populate SAFe program boards as early as possible for the entire solution or ART team to see (including the management team). More importantly, there should be team building exercises, lightning talks, cross-training, dependency discussions, and other cross-fertilization activities ACROSS the solution or ART teams. This is only possible in a large open room venue. Some people don’t like the noise and distraction, while others enjoy the excitement of a wedding-like atmosphere. It’s even possible to perform big-room solution and ART-level planning events remotely (virtually) using Zoom, Skype, Mural, iObeya, etc. It takes several licks to get to the center of a tootsie-roll tootsie-pop, so it may require two or three events for everyone to collaborate. Remember the purpose of big room planning is a simple set of agreed upon goals and objectives (vs. a detailed IMS).

14. Ac·tu·al • Cus·tom·er • Par·tic·i·pa·tion (*ăk'chōō-əl • kŭs'tə-mər • pär-tĭs'ə-pā'shən*) Buyer, client, consumer, end-user, business owner; To include customers in SAFe planning to communicate vision, needs, and validation

- ✓ Ensure business owner participate.
- ✓ Have business owners communicate vision.
- ✓ Have business owners assign business value.

A critical element of SAFe is that business owners and customers should participate in SAFe ceremonies, especially planning events. This supports the agile principle, “Business people and developers must work together daily throughout the project.” Within SAFe planning, business owners and customers present the vision, listen to solution and product managers describe the product roadmap (features), and listen to architects describe the technical constraints. Business owners and customers are smart and understand the implications of product roadmaps and technology stacks. They should agree with these high-level artifacts and even validate them before SAFe planning events. Wouldn’t you wanna agree with the blueprint of a custom house? Business owners and customers should participate in the SAFe process itself as “the voice of the customer.” This includes business executives, middle managers, program and project managers, and functional managers. Business owners and customers don’t just pop in for a 15-minute pep talk and jet set off to their next meeting. They participate in the full SAFe planning event to absorb its full implications, determine the value of the investment, answer questions by technical personnel, and ask questions of developers too. Business owners and customers sit in on breakout sessions; eat with the ART team; and participate in draft plan reviews, management problem solving, final plan reviews, confidence votes, retrospectives, and happy hours! Of course, program predictability is a simple, vital SAFe metric that reflects the satisfaction of business owners and customers. Therefore, program increment (PI) objectives should be written for them, business owners and customers should ask questions about these objectives, and then SCORE them on a simple scale of 1 to 10. This is often the most SKIPPED ceremony in SAFe planning (i.e., business owners and customers hanging around long enough to score business objectives). Of course, it doesn’t help if SAFe planning is long and boring and SAFe teams write developer-focused PI objectives. One anti-pattern is to expect business owners and customers to attend every daily standup, iteration planning, backlog refinement, demo, and retro—This is simply NONSENSE! Business owners and customers help form and communicate the vision, roadmap, tech stack, score PI objectives, and validate system demos (and the degree to which PI objectives are satisfied). They do NOT attend daily standups for three to nine SAFe teams along with all of the other routine lean-agile ceremonies.

15. Max·i·mize • Col·lab·o·ra·tion (*măk'sə-mīz' • kə-lăb'ə-rā'shən*) Assist, cooperate, synergize, teamwork, communicate; To work together in close-knit tightly-cohesive groups to deliver value-adding features

- ✓ Form working collaborative teams.

- ✓ Focus on communication and teamwork.
- ✓ Eliminate power-distance as much as possible.

A critical element of SAFe is maximum collaboration up and DOWN (vertically), as well as left and right (horizontally) across the food chain, organization, product team, or program. Successful SAFe ARTs are fully dependent upon maximum communication, cooperation, and teamwork of EVERYONE involved. Maximum transparency, information sharing, and shared responsibility must be sought, achieved, and realized. A SAFe ART is a single small, medium, or large team in lockstep together (e.g., synchronization, cadence, and delivery). A SAFe ART is NOT a loose collection of independently suboptimizing teams, political or territorial constituencies, or functional departments in it for themselves. A SAFe ART is not merely a random group of lean-agile teams selecting their own ceremonies, practices, tools, and measurements; stopping and starting; and exhibiting erratic behavior. We might expect that from small groups of independent lean-agile teams. BUT, a SAFe ART is NOT a group of independent teams doing their own thing, it is part of a greater collective or super cross-functional team building highly-cohesive product or services (or ecosystem of products and services). Therefore, there is NO command and control, hierarchy, power-status, functionally siloed departments, cliques, gangs, rogue elements, or completely independent terrorist cells. Again, a SAFe ART is not your typical collection of ragtag agile teams (i.e., a SAFe ART is NOT a militia of volunteers, it is a regularly polished professional army, trained, certified, in-uniform, and following carefully scripted goal or objective-oriented plans, ceremonies, practices, tools, and measurements). As such, it's necessary for ALL members of the ART including business owners, customers, end-users, solution and product managers, architects, product owners, scrum masters, and developers to OPENLY, regularly, and routinely communicate, cooperate, and collaborate in an egalitarian fashion (like one big happy family). Therefore, it's extremely important for everyone, especially lean-agile leaders (and we just don't mean coaches), to exhibit superior social skills, emotional intelligence, humility, and create and maintain a positive, fun, and enjoyable atmosphere. Again, SAFe is not a rigid command and control hierarchy with elevated levels of power-distance, master-slave, or buyer-supplier relationships. Managers, developers, subcontractors, and developers must strive to achieve a "one-team, badgeless" sense of psychological safety (i.e., teams that span organizational boundaries).

16. Es·tab·lish • Psy·cho·log·i·cal • Safe·ty (*ĭ-stăb'lĭsh • sī'kə-lŏj'ĭ-kəl • sāf'tē*) Haven, cover, shelter, refuge, protection; To create an environment of open, honest, egalitarian, and creative freedom of expression

- ✓ Optimize creativity and innovation.
- ✓ Gather creative inputs from everyone.
- ✓ Expect people to tell the truth at all times.

It's critical to establish a strong culture of psychological safety. Much of this is achieved by creating an environment where everyone on the SAFe ART participates in planning. This includes product management, architecture, planning, ceremonies, etc. SAFe plans are created by and for the teams themselves. In traditional planning, impossibly complex, over scoped plans are thrust upon teams in the form of integrated master schedules (IMSs). Project managers, coordinators, team leads, and schedulers torment developers on a daily basis to complete tasks on the IMS to meet short dates, deadlines, milestones, cost, budget, and scope goals. In SAFe, business owners, customers, portfolio, solution, and product managers create business experiment-oriented visions and roadmaps and allow teams to form loosely structured goal and objective oriented program boards, iteration boards, and story maps to swarm and implement the smallest possible minimum viable product (MVP) as quickly as possible (and gather measurable customer, market, and end-user feedback). Much of the visioning at the portfolio, large solution, and product roadmap level is done through bottoms up participatory planning. This means you're going to ask developers for their opinion, inputs, and direction. Typically, developers may not have the power and status of executives, directors, middle managers, product managers, and other technical leads. Oftentimes, developers are quiet introverted worker bees who sit at the bottom of the typical organizational hierarchy. That is, loud extroverted people often float to the TOP of the organizational hierarchy, while quiet introverted worker bees often sink to the bottom of the organizational hierarchy. This creates a deep power-status psychological divide where organizational leaders command, berate, and humiliate developers into achieving impossible IMSs on a short deadline. Developers, on the other hand, tend to keep their mouth shut, do what they're told, and work long hours to accomplish impossible deadlines to the best of their ability. Once again, SAFe turns the hierarchy upside down and typical organizational managers are servant leaders, while the once silent developers contribute to visions, roadmaps, architectures, program objectives, and program increment plans. They even identify estimates, risks, dependencies, and other typical management data. So, it's necessary for lean-agile servant leaders to create a culture, environment, and atmosphere where developers with little power and status confidently make important planning decisions.

17. Con·cise • Plan·ning • E·vent (*kən-sīs' • plă'nĭng • ĭ-vĕnt'*) Ready, prepare, schedule, organize, strategize; To hold the shortest possible SAFe planning event necessary to organize a value-adding program increment

- ✓ Keep SAFe PI planning short.
- ✓ Execute all SAFe PI planning ceremonies.
- ✓ Balance efficiency, effectiveness, and collaboration.

Although it's tempting to organize long SAFe planning events, the real magic is keeping it short, sweet, and to-the-point. It's a challenge to bring 50 to 125 people together in a large room. Business owners, customers, solution and product managers, and architects must communicate visions, roadmaps, and technical constraints. Teams must gel; assemble iteration boards, program boards, and product or program increment objectives; and identify critical high-level risks and dependencies. Teams must also eat, socialize, collaborate, explore dependencies, perform team building exercises, and get cross-training. That's a tall order for a two-day SAFe planning event and it's tempting to extend this to three or four days. Instead, master the art of holding the shortest possible planning events without sacrificing these goals. A little preplanning wouldn't hurt (i.e., assemble read aheads with visions, roadmaps, technical architectures and constraints; last quarter's performance data; and even

lightweight story maps, program increment objectives, program boards, dependencies, and risks prepared in advance by each team). A little homework never hurts and it's a great way to capture salient, tangible, and explicit planning data for people who can't attend, need a little time to think, and wish to walk away with valuable data, information, and knowledge. Definitely hold a socialization breakfast, present salient but quick-n-dirty opening presentations, and have two short breakout sessions (not to exceed 1.5 to 2 hours each). Present updated draft plans with objectives, program, and team boards as early as possible; have another socialization lunch; and don't forget dependency management, lightning talks and cross training, dinners, management reviews, and final presentations. Done well, this can be accomplished in about 1.5 days (one day for a small ART if the stars align). Hold all SAFe planning meetings, but don't extend them and then skip essential ones (e.g., eating, management reviews, or retrospectives). Use visual collaboration tools (iObeya, Mural, etc.) vs. ALM tools like Jira, ADO, CLM, etc. This is a good way to capture outputs, involve remote team members, and minimize non-collaborative heads-down time. Gather anonymous feedback as much as possible in the form of objectives, stories, estimates, risks, dependencies, critiques, and retrospective data (Kahoot, Poll Everywhere, etc.). Big room anonymous brainstorming and retrospectives using smartphones is an awesome way to swarm planning events with large groups in real-time. Take your time but hurry up!

18. Main·tain • Strict • Time·box·es (*mān-tān′ • strĭkt • tīm-bŏks′əs*) Time, span, space, period, interval; To obsessively limit, restrict, and respect the number, frequency, and duration of SAFe ceremonies

- ✓ Maximize use of development time.
- ✓ Keep SAFe ceremonies lean and mean.
- ✓ Strictly timebox all intervals and ceremonies.

Maintaining strict timeboxes for SAFe ceremonies is an extremely critical success factor! SAFe's goal is to plan as many rapid-fire business experiments (hypothesis tests) in the shortest period of time. Dot Coms like Google, Yahoo, Facebook, Amazon, Microsoft, Apple, etc. conduct 100,000+ small business experiments (hypothesis tests) per year. The difference between traditional and lean-agile thinking is that traditionalists believe they can predict the future, divine a mountain a business requirements, and then create mile-long integrated master schedules (IMSs) to be measured down to the second. The major goal of traditional thinking is to read the minds of non-existent customers and markets and then plan over scoped product solutions that require decades, millions of dollars, and armies of people to complete. They often justify this by filling up everyone's capacity to the brim with ambiguous requirements through which you can drive a Mack truck. They not only take a million years; but they can't be tested, debugged, or delivered; and of course, mythical customers don't need them. This fails to mention that the technology will be instantly obsolete before it can be delivered. Again, the expense of over scoped traditional systems is justified by compensating for every minute of a developer's time! Conversely, SAFe teams plan small tightly scoped business experiments (hypothesis tests) or features, quickly implement skeletal wireframes, and then have real customers evaluate them. With objective data in hand, more small, interrelated hypothesis tests can be conducted to zero in on valuable products and services. We've known since the 1980s that 95% of business requirements or product features are not needed nor used, but now we have damning evidence. That is, 95% of hypothesis tests conducted by Dot Coms FAIL to yield positive results. So, if SAFe teams are going to conduct small rapid-fire hypothesis-tests to smoke out the 5% that will yield positive results, there is no need to have a detailed set of business requirements, enterprise architectures, system designs, integrated master schedules, and project plans to fill up every minute of every day for 5, 10, or 15 years to get your money's worth out of your developer's salaries! As such, you don't need to have long SAFe ceremonies, you don't need a lot of them, and you don't need to saturate Microsoft Outlook schedules with endless meetings. You also don't need a three-or-four-day SAFe planning event to build a detailed traditional IMS or project plan (vs. a small number of business experiments)!

19. Full • Team • Par·tic·i·pa·tion (*fo͡ol • tēm • pär-tĭs′ə-pā′shən*) Entire, complete, everyone, whole team, all-encompassing; To invite members of the entire SAFe ART to PI planning to gather inputs and disseminate plans

- ✓ Invite entire team to planning.
- ✓ Encourage bottoms up participation.
- ✓ Delegate planning to technical engineers.

It's VERY important for the entire SAFe ART to participate in SAFe planning events. SAFe is an inherently "bottoms-up" paradigm (i.e., the best plans come from the people doing the work). In traditional paradigms, business development, product management, or project management develops the plans for the team in the form of integrated master schedules (IMSs). A work breakdown structure (WBS) or deliverable oriented IMS comes from a chief engineer or systems architect, since the goal is to capture ALL mythical technical scope of the final product or service. SAFe asks teams of DEVELOPERS who understand the customers, end-users, market, technical staff, and their own capabilities best to devise the plan for themselves. Even lean-agile teams are traditional thinkers too, so they are predisposed to create deliverable oriented SAFe plans to predict and capture the scope of a mythical product or service that the market, customers, or end-users DO NOT NEED (which is a major no-no). Again, the purpose of SAFe is to create a goal-oriented plan of a MINIMUM viable product (MVP), or a "just-enough" "skeletal" implementation, to test a hypothesis. The epics, capabilities, or features should also be in the form of hypotheses, vs. detailed business, architecture, design, or implementation requirements. That is, the lean-agile product or service roadmap is a set of hypotheses created to satisfy a vision for an ecosystem of innovatively new products and services. Therefore, lean-agile teams create skeletal business requirements themselves, oftentimes in the form of story maps or story boards. The goal is to then select a small set of these business requirements or user stories constituting the MVP in order to quickly gather measurable just-enough, just-in-time feedback from real end-users or customers. Think of it like a hungry shark that swims towards the smell of blood. The shark may meander for a while, but once it gets a whiff of blood, it zeros in on the target to devour its lunch. There may be a few false positives at first, before it finds a blubber laden whale carcass somewhere around it's territory. Therefore, include the entire ART or product team, including the development staff of each of the subteams in as

many SAFe product management, pre-planning, and planning events in order to get the best possible SAFe plans. And, remember, the purpose is for the team to write agile or SAFe objectives written in the language of the customer or business (i.e., the ART level goals the teams will pursue to gather the feedback necessary to satisfy the SAFe product roadmap).

20. Heads·up • In·for·ma·tion • Ra·di·a·tors (*hĕdz-ŭp • ĭn'fər-mā'shən • rā'dē-ā'tərz*) Map, wall, chart, board, poster; To use simple highly-visible status, program, Kanban, or iteration boards to track value-adding work in progress

- ✓ Use physical planning boards.
- ✓ Use visual aids as much as possible.
- ✓ Use physical planning boards throughout PI.

Remember to maximize the use of heads-up information radiators during SAFe planning. These include whiteboards, flipcharts, walls, and windows. Reduce the amount of heads-down technology in SAFe planning events as much as possible. SAFe wants EVERYONE full engaged. This means teams standing around iteration boards, program boards, draft and final PI plans, product objectives and risks, etc. Limit the use and presence of heads down technology like laptops and desktop computers, smartphones, and analytical agile lifecycle management tools like Jira, IBM CLM, ADO, etc. These latter tools have their place, but NOT in SAFe planning where we need people talking, socializing, visiting, communicating, collaborating, learning, listening, participating, and planning together. The best and most timely, valuable, and important information and data is in people's heads (brains) and the most efficient method of conveying information is through communication (speaking) and creating plans together. The output of SAFe PI planning is INTANGIBLES like trust, cooperation, learning, socialization, cooperation, relationships, information sharing, inclusion, psychological safety, and FUN. Of course, TANGIBLE outputs of SAFe include a VERY SMALL NUMBER of product objectives written in the voice of the business or customer, program board, dependencies, risks, confidence votes, and agreed upon team boards (i.e., very loosely populated story maps). Again, this is an goal or objective-oriented plan to satisfy a SMALL number of business experiments, hypothesis tests, or features. A SAFe planning event does NOT produce a 15,000+ line integrated master schedule (IMS), work breakdown structure (WBS), work packages, tasks, or detailed implementation steps that traditionalists use to build over scoped, gold fleeced, bloated, and unneeded products and services that require a million years to implement. SAFe involves teamwork, swarming, pairing, collaboration, cooperation, shared responsibility, and crowdsourcing. The purpose of SAFe is NOT to implement a fixed-scope system in one fell-swoop. SAFe's fundamental purpose is to create innovatively new, value-adding, and high-quality products and services with the least possible lead and cycle time. SAFe does this by implementing a small set of business experiments, hypothesis tests, and features to tease out tacit customer, market, and end-user needs until the optimal value point is achieved. This is done with lightweight heads-up vs. detailed heads-down information radiators.

21. Light·weight • It·er·a·tion • Plans (*līt-wāt • ĭt'ə-rā'shən • plănz*) Easy, thin, slim, slender, skinny; To create skeletal team-level iteration plans during SAFe planning for implementing minimum viable products (MVPs)

- ✓ Create lightweight iteration plans.
- ✓ Use high-level story maps/story boards.
- ✓ Minimize use of detailed traditional planning.

The fundamental purpose of SAFe planning is for the ART or product teams to form extremely lightweight iteration plans (often in the form of skeletal story maps). Remember, SAFe is intended to implement small series of business experiments, hypothesis tests, and features to tease out intangible customer, market, and end-user knowledge. In doing so, SAFe teams can identify the optimal value point for innovatively new products and services with the least possible lead and cycle time. Again, SAFe is NOT intended to create nor implement a 15,000+ line integrated master schedule (IMS) in the form of a detailed product roadmap, release plan, program board, or set of iteration boards. SAFe product managers should perform lean-agile product management, and product owners and scrum masters properly perform program increment preplanning, story mapping, and/or apply closely related design thinking practices and tools. Therefore, individual SAFe lean-agile teams should have as input into program increment planning a loosely structured story map they participated in pre-building. As such, the output of program increment planning should be an agreed-upon story map with a little more updated information. Remember, story mapping, although timeboxed, results in a very detailed set of possible business requirements. It's a rapid brainstorming technique where a small group of people quickly identify a large pile of user stories. The real magic behind story mapping is in deconflicting these ideas, identifying patterns, and distilling these ideas to a simplified possible grouping of essential user stories. From there, the SIMPLIFIED version is further simplified into a minimum viable product (MVP)—The smallest number of user stories necessary to implement a business experiment, hypothesis test, or feature (not a gold fleeced, over scoped subsystem, component, or module). The smallest possible story map should consist of a single story, or at least a handful of stories. While SAFe recommends loading iterations, other than the innovation and planning iteration, to 80%, this is simply not sufficient. True innovation teams should be loaded at around 50% to 60%. That is, the story map or iteration plan should not consume more than half of an individual lean-agile team's capacity (possibly less). This is especially true for extremely innovative products and services with high levels of risk, uncertainty, and doubt. Therefore, SAFe teams must strive to create lightweight goal and experiment-oriented iteration plans vs. detailed integrated master schedules (IMSs).

22. Build·in • Ex·cess • Ca·pac·i·ty (*bĭld-ĭn • ĕk'sĕs' • kə-păs'ĭ-tē*) Space, margin, border, reserve, cushion; To create a conservative team plans to allow for variability, creativity, innovation, and unplanned work

- ✓ Aim to create lightweight MVP.
- ✓ Create lightweight business experiment.
- ✓ Create enough of an MVP to measure feedback.

It ALMOST goes without saying that lean-agile teams MUST build-in excess capacity into their SAFe plans. Remember, the

goal of SAFe is to rapidly field a small number of simple business experiments, hypothesis tests, and features to tease out intangible market, customer, and end-user feedback. Again, in traditional planning, the goal is to jam as many over scoped business requirements into a 15,000+ line integrated master schedule (IMS) as you can to fill up each developer's every waking hour several times over. Conversely, the ultimate purpose of SAFe is an innovatively new product or service with a high degree of risk, uncertainty, and doubt. Therefore, it's mathematically impossible to estimate the time it takes to tease out intangible customer needs, much less account for every day, hour, minute, and second of a developer's time by cramming their schedules with every conceivable business requirement. This, of course, fails to mention that traditional managers don't stop adding new business requirements to developers during planning, but reserve the right to dream up even more business requirements every day, even while they sleep, and add more and more work on top of developers each and every day. Traditional managers believe it is their God given right to break the back of developers by tormenting them with out-of-scope business requirements because they're paying them a lot of money do so! However, the basic purpose of SAFe is to conduct a small series of tightly scoped business requirements, rapidly implement and field them, gather end-user feedback, and test the hypothesis as quickly as possible (at a sustainable pace). In order to gather this feedback, experiments have to be small, quick, and inexpensive. This has to take place in days, hours, minutes, and seconds (almost real time). However, don't get the message that we can simply cram 60 business requirements into every minute. Instead, managers and developers cooperate to plan a small number of simple experiments and PLAN for UNCERTAINTY, creativity, innovation, sustainable pace, longevity, self-mastery, continuous improvement, and even fun! SAFe recommends loading teams to 80% capacity, but this is a bit unrealistic. Even computers can't be loaded to 80% capacity, and humans are certainly not machines. Instead, teams should plan for 30% to 60% capacity and reserve as much time for creativity, innovation, and uncertainty as possible. That is, plan to gather measurement data quickly, leave room for uncertainty, and then implement another experiment quickly.

23. Con·sis·tent • Plan·ning • Prac·tic·es (*kən-sĭs'tənt • plă'nĭng • prăk'tĭs'əs*) Similar, standard, uniform, identical, homogeneous; To use standard ceremonies, practices, metrics, and tools across all SAFe PI planning teams

- ✓ Execute uniform planning practices.
- ✓ Create repeatable planning ceremonies.
- ✓ Enforce consistent ceremonies and estimating.

Synchronization, cadence, quality, and consistency are king in SAFe! Yes, SAFe is a creative problem-solving framework. However, SAFe is also a repeatable framework for creating innovatively new products and services. It's used to conduct business experiments AT-SCALE. An individual lean-agile team might conduct business experiments IN-THE-SMALL, while a SAFe ART is a team of teams. That is, SAFe ARTs are highly cohesive cross functional teams that conducts larger groups of business experiments to build ecosystems of new products and services. Therefore, not only does an individual lean-agile team have to be consistent in SAFe, but ALL teams on the ART must be consistently consistent! Teams vision, roadmap, architect, plan, implement, tool, measure, deliver, and continuously improve TOGETHER, in cadence, synchronization, and UNISON. SAFe is more like a fairly large symphony orchestra coordinated by a maestro (solution and release train engineer). SAFe is less like a noisy smattering of disorganized individual one-person street bands on the boardwalk playing a variety of eclectic music on homemade instruments. Therefore, all lean-agile teams on the SAFe ART use consistent planning practices, teams don't select from obscure collections of homegrown, opinionated practices, tools, and metrics. They don't determine the levels of planning necessary. One team doesn't plan for 5 minutes, another for 30 minutes, and another for 4 hours. They don't use different formats for business requirements (i.e., user stories, use cases, UML, SysML, wireframes, EA, DoDAF, etc.). And, they don't pull estimating practices and metrics out of left-field (i.e., inches, feet, kilometers, pounds, grams, etc.). One team shouldn't be estimating in seconds, while another team estimates in story points, and yet another team estimates in COSMIC Function Points. One team shouldn't equate story points to 15 minutes, while another team equates them to 3 days. This just doesn't make sense, especially in SAFe. It's pretty typical for a disorganized lean-agile transformation to have multiple teams with inconsistent practices. One team may do standups, another sprint planning only, another demos only, another retrospectives only, and another all Scrum ceremonies. One team may use ALM tools, another uses whiteboards and flipcharts, while another team's plan is in the product owner's head. SAFe demands, expects, and coaches all teams to use consistent, uniform, and repeatable practices. It's simply NOT appropriate to choose ad hoc ones.

24. Con·sis·tent • Es·ti·mat·ing • Prac·tic·es (*kən-sĭs'tənt • ĕs'tə-mā'tĭng • prăk'tĭs'əs*) Similar, standard, uniform, identical, homogeneous; To use standard units of metrics and models for measuring performance

- ✓ Apply uniform capacity estimating.
- ✓ Create light weight team estimates.
- ✓ Use consistent estimating framework.

SAFe ARTs should use consistent estimating practices to the maximum extent possible! Consistent estimating in SAFe appears at multiple levels throughout its framework. Let's start with the program or product team level. All teams should have the same iteration length, cadence (begin and end iterations together), plan, inspect and adapt, and innovate together. SAFe is NOT a ragtag group of ALMOST AGILE teams doing whatever they please, whenever they want to do it, and using whatever ceremonies, practices, tools, and metrics they want. That's simply chaos, disorder, and anarchy! Some SAFe coaches believe agile teams are empowered to create their own rules, but that's suboptimizing. One team may move fast, while another team never gets formed or is dysfunctional. This will simply unbalance the SAFe ART and sink the ship. The lean-agile teams on a SAFe ART are more like a naval battlegroup and less like a flashmob. Beyond iterating and planning together, SAFe teams also estimate together. They typically use the same units of measure (i.e., story points), they typically us the same estimating practices (i.e., planning poker), and they typically use the same scale (i.e., story points are generally about the same level of size, complexity, and effort). A story point shouldn't be one hour, a day, or a week to different teams.

SAFe coaches and scrum masters should not only teach, coach, and expect consistent estimating practices, but actually perform consistent estimating during pre-planning, program increment planning, iteration planning, and backlog refinement. One team's velocity shouldn't be 0 points per iteration, while another's is 25, 50, 75 or 300. It simply doesn't make sense. One may argue that the iteration goal or program objective is the most important measure (along with business value and program predictability), and this much is true. If a team's velocity is 0 story points, then that team is stuck in first gear. If another team's velocity is 25 to 75 story points, at least we have something to work with. If another team's velocity is 300, then they're probably planning in tasks, assigning tasks to individuals, and working at maximum capacity. That's simply traditional thinking. Story points are a creative measure, not a method of timekeeping. If a team is measuring in weeks, days, hours, or minutes, watch out! Again, the goal is not to compare teams, micromanage teams, or dismiss low-performing teams. The objective is to coach a SAFe ART to successfully deliver upon as many goals or objectives as possible (with moderate capacity limitations).

25. Dem·o·a·ble • It·er·a·tion • Boards (*dĕm'ō'ə-bəl • ĭt'ə-rā'shən • bôrdz*) Show, exhibit, present, describe, illustrate; To communicate team-level iteration plans during SAFe PI planning in addition to PI plans and program boards

- ✓ Create visible iteration boards.
- ✓ Demo iteration boards if possible.
- ✓ Demo planned iteration load estimates.

SAFe teams are asked to swarm around information radiators to create iteration plans for up to a 90-day period (some of which are called team-level iteration boards). Teams determine the number of developers that will be available for the program increment period, their availability, and allocate stories to iterations. There is a final iteration called an Innovation and Planning (IP) Iteration that should NOT have any user stories allocated to it at all. The IP iteration is a planning buffer, safety net, and time to wrap up any loose ends. The IP iteration may be used for some minimal carryover IF necessary, but this is an antipattern if you're planning too much work for the final iteration. For example, a team may have about 75 user stories in the backlog to implement over the next 90 days. Teams decide which user stories go into each iteration, any dependencies or risks, and, more importantly, iteration goals! User stories should be written using the INVEST criteria: "I" ndependent, "N" egotiable, "V" aluable, "E" stimable, "S" mall, and "T" estable. Treat each user story as a fully separate or independent business experiment. One or more user stories may be highly cohesive (i.e., a small series of interrelated business experiments). However, each business experiment should be implementable without the other. Also, IF, teams have done program increment pre-planning, devised a user story map, and have draft plan in-hand (as they should), then allocating the user stories from the story map to the iteration boards should be a CINCH! Furthermore, it is a Scrum best practice, no imperative, that each iteration should have a goal, purpose, output, or business value (so iterations are NOT random collections of user stories). Iteration one may be to explore the worthiness of a cloud service, iteration two may be to stress test the basic capabilities of the cloud service, iteration three may be to migrate a few legacy applications to the cloud and query user data from them using a metasearch streaming service, etc. These iteration goals become SAFe program increment (PI) objectives. So, in summary, story maps become iteration boards; user stories are fully independent business experiments; user stories are loosely coupled and highly cohesive; and each iteration has a goal, theme, or objective in the business or customer language. Don't SKIP iteration boards because you're lazy. Make them highly-visible for everyone to see, to optimize transparency, collaboration, communication, and TRUST (don't hide 'em in ALM tools or secret team rooms)!

26. U·til·ize • Pro·gram • Boards (*yo͞ot'l-īz' • prō'grăm' • bôrdz*) Plan, strategy, approach, roadmap, blueprint; To proactively construct SAFe program increment dependency boards and illustrate them

- ✓ Create program boards early.
- ✓ Use program boards throughout PI planning.
- ✓ Demo program boards during management and final plan review.

Program boards are a critical element, information radiator, communication device, and output of SAFe planning. Program boards are used to illustrate the overall strategy for a program increment (90-day period). Program boards emerge bottoms-up from SAFe teams and they are NOT integrated master schedules (IMSs) pushed down on teams by traditional managers! Program boards are a lightweight high-level skeletal structure of highly cohesive business experiments to be performed by SAFe ARTs expressed as features. Program boards consist of a two-dimensional grid with iteration columns and team rows. The top row may consist of dates and perhaps another row for critical milestones like trade shows, demos, and other external dog-n-pony events; solution train demos, solution tests, and other critical integration points; or even regulatory requirements (SOX, NIST, or ISO compliance by a particular date). SAFe product management teams create minimalistic, but prioritized feature roadmaps and backlogs, and teams determine the minimal user stories that constitute each feature (to satisfy the acceptance criteria). Product managers, product owners, and the teams negotiate whether the team's implementation plan satisfies the feature's acceptance criteria. Communication, collaboration, teamwork, and transparency are KING in SAFe (and a little due diligence goes a long way)! So, teams determine WHEN (what iteration) they plan to complete the user stories constituting each feature, place the feature in the appropriate column and row, and identify any critical dependencies between other teams (columns and rows). Teams should probably collaborate on a draft program board in advance for all stakeholders to see, inspect, and analyze; teams should populate and finalize the program board throughout SAFe planning; and, most importantly, teams should demo their portion of the program board during draft and final program increment plan reviews. That seems logical, but teams often SKIMP on SAFe planning practices, game planning events, skip the program board, and don't understand its critical role in SAFe planning. The program board IS the central information radiator in SAFe planning. All stakeholders are invited to SAFe planning and program boards ARE a necessary tool to communicate, collaborate, and synchronize. Therefore, SAFe teams must bring the program board front and center, proactively populate it, and display it for all to see early and often. Needless to say, program boards are the central information radiator used for ART syncs and SoSs.

27. Con·cise • Pro·gram • Boards (*kən-sīs′ • prō′grăm′ • bôrdz*) Outline, sketch, skeletal, blueprint, lightweight; To create a high level SAFe planning board illustrating essential features and dependencies

- ✓ Create high-level program boards.
- ✓ Exhibit minimalistic critical dependencies.
- ✓ Abstract/rollup detailed story-level dependencies.

SAFe program boards represent high-level, minimalistic, and lean-agile product roadmaps or strategic plans for a 90-days. SAFe program boards should be as concise as possible to be meaningful, useful, and valuable. SAFe lean-agile product roadmaps and feature backlogs also represent minimalistic sets of business experiments to evaluate some hypotheses, construct some control and treatment tests, execute those tests, and gather feedback. Hypothesis tests are supposed to evaluate theories, not prove or disprove them (i.e., "Does 'one-click' checkout for online book shoppers increase sales"?). Therefore, a minimalistic feature hypothesis test would be constructed to implement a 'one-click' checkout feature as quickly as possible. It might be rapidly deployed to a live online shopping website; usage statistics and economic data collected to determine if sales increased, decreased, or stayed the same; and the feature evaluated and refined as necessary. A SAFe team may plan a minimalistic story map to evaluate the 'one-click' checkout feature. The position in the program board (iteration) in which the 'one-click' feature is ready to be deployed and tested is noted in the form of a blue index card or post-it note on in the appropriate row and column. Perhaps an architectural runway (enabler) feature is necessary, like a blue-green or dark release website instance has to be set up on a cloud platform to deploy the new feature. This way, a select group of frequent shoppers, end-users, or customers can be instantaneously switched to the ecommerce website instance with the new feature for a few hours, days, or weeks; data can be collected; and then switched back to the production website. Product managers evaluate whether the dark release impacted sales, why or why not, collect additional feedback, and a propose a new feature hypothesis test to optimize sales. Again, a SAFe program board is a lightweight, bottoms-up lean-agile plan from developers vs. a detailed minute-by-minute integrated master schedule (IMS) for achieving full-developer utilization, multi-tasking hell, and building over scoped systems on a shoe-string budget. Therefore, the SAFe ART's program board should not be too busy looking (it's strategic)! Furthermore, only critical dependencies should be highlighted, and it should not look like a Black Widow's spider web. Developers shouldn't use it for user story or task planning nor use dependency strings as a proxy for tasking or punishing people for troubling them with a commitment to implement a hypothesis test on a short deadline!

28. Cus·tom·er • Fo·cused • Ob·jec·tives (*kŭs′tə-mər • fō′kəst • əb-jĕk′tĭvz*) Market, consumer, commercial, end-user, business; To write SAFe program increment planning objectives in the language of the customer or business

- ✓ Write clear business objectives.
- ✓ Speak the customer's language.
- ✓ Plan and manage to PI objectives.

In SAFe, customer-focused program increment objectives in the voice of the customer, end-user, or business are very important. SAFe empowers development teams to listen to the product vision, evaluate the product roadmap, preplan feature hypothesis tests, and construct the bottoms up implementation plan for a 90-day period all by themselves. SAFe empowers developers to use their own brain to creatively implement solutions to feature hypothesis tests. Of course, they should do this collaboratively and not in a vacuum. There's an old saying in traditional projects that developers should never surprise their managers, and that is still true in SAFe—Don't surprise your stakeholders. Instead, if managers involve developers in the planning process, then developers should involve managers early and often throughout the implementation process in the form of frequent demonstrations at the team, system, and program increment level. In both SAFe and Scrum, the primary output of a lean-agile plan is the iteration or program increment goal or objective, NOT a pile of user stories, story points, hours charged by individual developers, nor amount of output (i.e., business requirements, system architecture diagrams, tests, software code, documentation, etc.). That is, each iteration in SAFe and Scrum should have a purpose, goal, or objective. The goal of iteration one is to stand up a test server for dark releases, the goal of iteration two is to prototype a feature spike on a new cloud service, the goal of iteration three is to conduct a load test on the new cloud service, the goal of iteration four is to implement a new 'one-click' checkout feature on the new cloud service for 50,000 users a minute to see if abandoned shopping carts decrease and sales increase, etc. Therefore, hours charged per person, capacity and load per person, velocity per person, documentation per person, tests per person, code person, and/or story points per person are NOT the primary measures in SAFe and Scrum. The primary measure is whether the purpose, goal, or objective is meaningful to customers, business owners, and product managers; these stakeholders agree with goals; and to what degree teams achieve their goals! In BOTH SAFe and Scrum program increment and iteration goals and objectives are the plan, not the detailed user stories, story points, tasks, hours, burndown, and velocity. Therefore, SAFe teams should focus on mastering the art and science of writing, communicating, using, and measuring program increment and iteration goals and objectives.

29. Just·in·Time • Train·ing (*jŭst-ĭn-tīm • trā′nĭng*) Fast, quick, timely, punctual, opportune; To provide small amounts of quick on-the-job training on an as-needed basis to satisfy targeted context-specific needs

- ✓ Train teams early and often.
- ✓ Train teams to solve specific problems.
- ✓ Reward people for sharing knowledge and skills.

SAFe is built on the four core values of program execution, alignment, transparency, and build-in-quality when implementing innovatively new products and services at-scale! It's also built on core principles like synchronization, cadence, consistency, and repeatability for teams of teams! That is, unlike an organization with ad hoc lean agile teams running around like the Keystone Cops, SAFe teams play beautiful music together like highly talented professional orchestras! This takes, practice,

coaching, teaching, training, and careful coordination. Musicians dedicate their lives to mastering one or more instruments. This involves lifelong training, practice, education, hard work, and elbow grease. Groups of musicians also have to be trained (i.e., strings, brass, percussion, woodwinds, etc.). And, of course, the entire symphony orchestra has to practice and train together in unison day after day for a long period time. When an audience pays to hear a world-class symphony orchestra, it is easy to forget the blood, sweat, and tears that go into the lifetime of training, mastery, and education of those musicians. Sometimes, the training is even apropos, like a team lead providing some just-in-time coaching tips to someone who is a little out of tune or pace, suggesting a better-quality instrument, or even demonstrating to an individual musician on how to play a particular piece. The basic message is that training, training, and more training is necessary for world class symphony orchestras to reach a point of near perfection and the SAME is true of SAFe teams. Unfortunately, little time is applied to the art and science of training in SAFe teams. Sometimes, people are hired who have been trained and certified in advance. Sometimes the entire ART is subjected to the training requirements of the SAFe Implementation Roadmap. Sometimes individuals take it upon themselves to take training classes for self-mastery, certification, or career advancement. The point is that the critical importance of training cannot be overstated. Conduct a few training exercises and lightning talks during SAFe planning events, breakout sessions, and even along side the events themselves (just before or after). SAFe planning takes place within the Innovation and Planning (IP) iteration (i.e., the IP is designed to acquire just-in-time training to improve performance, quality, and gear up for the future). This is a good time to show developers how to master lean-agile thinking, SAFe ceremonies, agile ALM tools, new technologies, and even emotional intelligence. A little OTJ training goes a long way!

30. Vi·cious • Cross • Train·ing (*vĭsh'əs • krôs • trā'nĭng*) Share, educate, prepare, instruct, pollinate; To disseminate the technical skills of all specialists to every member of the team to improve productivity

- ✓ Disseminate all technical knowledge.
- ✓ Ensure skill redundancy as much as possible.
- ✓ Openly share data, knowledge, skills, and information.

A little just-enough and just-in-time training goes a long way in SAFe, and so does VICIOUS cross-training! Who wouldn't show employees how to log into an antiquated enterprise portal? Which manager wouldn't show employees how to enter their hours into the time-reporting system on an annual basis (for legal reporting purposes)? Which SAFe coach wouldn't take an ART through the SAFe Implementation Roadmap and its requisite just-in-time training courses? However, many of these training examples are impersonal, institutional, and done at the enterprise or group level. GREAT SAFe teams take this to the NEXT-LEVEL? Teams and individuals must take it upon themselves to viciously cross-train one another on a daily basis. SAFe's purpose is to smoke out tacit, intangible, and uncertain market, customer, and end-user needs quickly—at the speed of light—because today's enterprises are engaged in a global battle where millions of firms entice your customers away with superior products and services at better prices! SAFe teams don't have the luxury of implementing over scoped integrated master schedules (IMSs), business requirements, and system architectures over years and decades while mobile wearable devices are innovating in seconds, days, and hours (while we sleep). SAFe teams only have a few hours to evaluate a small hypothesis test to yield enough revenues to stay solvent for the next quarter. While it may have taken an individual a lifetime to master a discipline like cloud computing, security, architecture, programming, testing, tooling, or SAFe itself, teams must be willing to CONVEY that lifetime of knowledge to a teammate in a few minutes. Someone has to sit down for a few minutes, show people how to login into a new tool, navigate to essential features, configure them, use them, analyze results, and improve. Gone are the days when you can just tell them to go and get a master's certificate and come back when they're done. Convey your knowledge to a teammate in a few minutes and the WHOLE team can tease out critical market knowledge in a few minutes. Withhold individual knowledge and the entire team is paralyzed. It's said that any person on the team can execute 80% of a function with only 20% of the knowledge and skills before reaching the point of diminishing returns (and two people working together can solve a hard problem in 15-minutes, while an individual working alone will require 6 to 9 months)! Let's viciously cross-train one another, solve hard problems in 15-minutes, and delight our customers faster than competitors.

31. Ba·sic • Ag·ile • Meas·ure·ments (*bā'sĭk • ăj'əl • mĕzh'ər-məntz*) Key, vital, simple, minimum, essential; To collect and utilize only the bare-minimum metrics and measurement necessary to capture the essence of performance

- ✓ Use bare minimum metrics.
- ✓ Use basic Scrum/agile metrics.
- ✓ Use SAFe program predictability.

While it doesn't appear so on the surface, SAFe has had a recommended set of BASIC agile measurements since its inception! Software measurement has been a very controversial discipline since its earliest days with no two people agreeing on the basic process and product measurements necessary for success. Others, mostly developers, have rebelled against any productivity or information technology measurements—Instead arguing that knowledge is pure psychological thought stuff that defies measurement. So, there you have it, two diametrically opposing camps—One says measure every minute of every day, along with every ounce, while others declare that new product and service development measures are for fools. Worse yet, traditional management is a pantheon of measurement demons in the form of earned value management (EVM), parametric metrics and models, and analytical workflow timekeeping and task tracking systems used to measure individual productivity in seconds, minutes, and hours. Yes, Fredrick Taylor and even Watts Humphrey recommended that people measure their time in minutes and seconds with stop watches, wow! Agile lifecycle management (ALM) tools aren't much better, encouraging people to enter individual user stories to fill every minute of every day, allocate hours to them on daily basis, and ensure every task has an output attached to it like documents, code, tests, designs, etc. More importantly, no two lean-agile coaches can agree on a common set of measures, and generally recommend dozens each day (the more the better)! This is a MAJOR MISTAKE. SAFe teams must apply a basic set of agile measurements. Tracking work in user stories,

story points, burndowns, velocity, and team happiness are sufficient (perhaps lead and cycle time if you're a Kanban team). For SAFe in particular, have real business owners, customers, or other vital stakeholders such as product managers score program increment objectives during SAFe planning and Inspect and Adapt (I&A) events. Measure program predictability at a very minimum, and reject the natural instinct to identify and enforce dozens of traditional, agile, and lean metrics. Garbage in, garbage out, if teams don't use basic measures, and track them in simple tools, then a SAFe ART is simply flying in the dark! Instrument and configure metrics in advance, train and coach people how to use the metrics, and teach people how to apply them. Keep data private and anonymous, because seeing is believing, and metrics are key to continuous improvement!

32. Rou·tine • Ag·ile • As·sess·ments (*roo̅-tēn' • ăj'əl • ə-sĕs'mənts*) Test, analysis, appraisal, evaluation, measurement; To use simple out-of-the-box SAFe assessments early and often to gather qualitative performance data

- ✓ Use basic SAFe assessments.
- ✓ Minimized custom assessments.
- ✓ Apply assessments early and often.

Conducting routine agile assessments is a SAFe success factor! SAFe has gone the extra mile and created a slew of basic assessments to help SAFe teams collect performance data in a non-threatening manner. SAFe teams forget how challenging it is to acquire a complex agile lifecycle management (ALM) tool; train people how to use it; configure it properly; coach teams to apply it on a daily basis; and data mine the performance data from it with complex SQL queries, filters, Javascripts, Ruby, PowerBuilder, APIs to better visualization systems, etc. Without basic measurement data, SAFe teams are flying in the dark. So, yes, acquire and configure a simple toolset IN-ADVANCE, TRAIN and COACH people how to use it, keep it simple and anonymous, and GIT-R-DONE with agile metrics. However, in the meantime, we don't need to wait for an over scoped, long lead time, WIP-laden agile ALM toolset—Start using basic SAFe assessments early and often TODAY! Here's the secret—Use the out-of-the-box SAFe assessments—It's THAT SIMPLE! Don't SKIP them because you don't have a clue what an assessment is and you're afraid of them! Don't tailor them until the cows come home because you're not familiar with the SAFe vernacular (i.e., iteration planning, standups, demos, retros, etc.)! And, more importantly, don't bring in your own SAFe assessment from your last consulting gig because of the NOT-INVENTED-HERE (NIH) Syndrome, you like the one you stole from your last client, you wanna look like the hero for using a custom assessment, and you don't want another SAFe coach to take credit for suggesting basic SAFe assessments. If the SAFe coaches can't agree to use the basic out-of-the-box SAFe assessments, then they probably aren't SAFe coaches. SAFe coaches must form a coalition to socialize SAFe values, principles, practices, measurements, assessments, tools, and metrics. If a house is divided against itself, then it cannot stand! Take courage, TRUST SAFe, get the SAFe assessments, apply them as-is, and do it early and often, because the agile ALM team will take a year or two to get their act together due to their cost, complexity, and overhead. Using agile ALM tools is sort of like renting an elephant for a birthday party (i.e., gotta find one, gotta rent a Mack truck, gotta get it through the gate, gotta feed it, gotta cleanup it's poop, and you gotta scrape up the bodies of those it trampled)! So you have a choice, accept the consequences of renting elephants for birthdays or conduct simple out-of-the-box SAFe assessments—You make the call!

SAFe Summary

So, what have we learned in this short treatise on how to maximize the ROI of using SAFe planning and its associated values, principles, practices, metrics, and tools? Well, we've learned that a little bit of short-term adaptable planning goes a long way with respect to successfully implementing innovatively new products and services. There is a stark difference between traditional and lean-agile thinking when it comes to planning. In traditional thinking, the goal is for a small group of managers to build a large backlog of business requirements to be codified into long-range integrated master schedules (IMSs) representing gold-fleeced, over scoped market, customer, and end-user needs, that quite frankly, are expensive, fraught with failure, and not needed at all. In lean-agile thinking, innovators believe that true market, customer, and end-user needs exist as hidden, inexpressible, and intangible needs deep within their subconscious psychological minds. Therefore, rather than building multi-decade long, multi-billion-dollar business requirements documents and IMSs in a vain attempt to read people's minds and divine the future, it's necessary to plan and implement a series of short-term business experiments to gradually tease out these hidden market, customer, and end-user needs. While basic lean-agile methods designed for small teams building small point-solutions have proven effective for iteratively teasing out these needs with some success, there is a need for lean-agile scaling frameworks like SAFe so that larger teams of teams can plan, iterate, and innovate together to bring high-quality products and services to market with the least possible lead and cycle times. SAFe planning events were designed specifically for this latter purpose and have proven successful at doing so, time and time again in multiple public and private settings but require careful attention at-scale.

32 Principles and Practices for Maximizing the ROI of SAFe Program Increment (PI) Planning

1. **Upfront Training & Preparation**—Lay the basic groundwork or foundation for using SAFe practices
2. **Manageable Train Size**—Constrain SAFe ART to size no smaller or bigger than necessary to deliver value-adding features
3. **Viciously Limited WIP**—Obsessively minimize the materials and work necessary to quickly deliver value-adding features
4. **Lean Product Management**—Apply lean design thinking to create consumer-grade market-oriented products and services
5. **Cross Functional Feature Teams**—Form small teams of people with the skills necessary to deliver value-adding features
6. **Build Competent Agile Teams**—Prepare all cross-functional teams in SAFe ART to rapidly deliver value-adding features
7. **Team Level Preplanning**—Develop a draft team-level program increment plan in-advance of SAFe planning
8. **Perform Story Mapping**—Create a cohesive business requirements architecture, functional flow, story board, or wireframe
9. **Simplest Possible Tools**—Use intuitive, easy-to-use, automated workflow tools with the least possible overhead
10. **Visual Collaboration Tools**—Use graphical planning and collaboration workflow tools to simplify and communicate
11. **Commercial Cloud Services**—Use innovative, public, and commercial Internet-based collaboration tools and services

12. **Alternating Local & Remote**—Alternate between local face-to-face and remote SAFe ceremonies such as PI planning
13. **Big Room Planning Venue**—Use a large-enough virtual or physical meeting space to accommodate the entire SAFe ART
14. **Actual Customer Participation**—Include customers in SAFe planning to communicate vision, needs, and validation
15. **Maximize Collaboration**—Work together in close-knit tightly-cohesive groups to deliver value-adding features
16. **Establish Psychological Safety**—Create an environment of open, honest, egalitarian, and creative freedom of expression
17. **Concise Planning Event**—Hold shortest possible SAFe planning event to organize value-adding program increment
18. **Maintain Strict Timeboxes**—Limit, restrict, and respect the number, frequency, and duration of SAFe ceremonies
19. **Full Team Participation**—Invite members of the entire SAFe ART to PI planning to gather inputs and disseminate plans
20. **Headsup Information Radiators**—Use simple highly-visible status, program, Kanban, or iteration boards to track work
21. **Lightweight Iteration Plans**—Create skeletal team-level iteration plans during SAFe planning for implementing MVPs
22. **Build-in Excess Capacity**—Create conservative plans to allow for variability, creativity, innovation, and unplanned work
23. **Consistent Planning Practices**—Use standard ceremonies, practices, metrics, and tools across all SAFe planning teams
24. **Consistent Estimating Practices**—Use standard units of metrics and models for measuring performance
25. **Demoable Iteration Boards**—Communicate team plans during SAFe planning in addition to PI plans and program boards
26. **Utilize Program Boards**—Proactively construct SAFe program increment dependency boards and illustrate them
27. **Concise Program Boards**—Create a high level SAFe planning board illustrating essential features and dependencies
28. **Customer Focused Objectives**—Write program increment planning objectives in the language of the customer or business
29. **Just-in-Time Training**—Provide small amounts of quick on-the-job training on an as-needed basis to satisfy targeted needs
30. **Vicious Cross Training**—Disseminate technical skills of all specialists to every member of the team to improve productivity
31. **Basic Agile Measurements**—Collect and utilize bare-minimum metrics necessary to capture and measure performance
32. **Routine Agile Assessments**—Use out-of-the-box SAFe assessments early and often to gather performance data

Once again, there are many stark philosophical differences between traditional and lean-agile thinking when it comes to planning and implementation values, principles, practices, metrics, and tools. In traditional thinking a small group of managers creates a plan, typically in a vacuum, for a larger group of developers. Traditional thinking is a DIVIDE-and-CONQUER approach based on the theory is that the managers know more than the developers, an infinitely complex problem space can be divined in advance and captured in IMSs (by one or more planners), each of tasks can be assigned or pushed to one or more low-paid, ignorant developers who can implement these work packages, and then later integrated by higher-paid engineers to build a final solution. Worse yet, traditional firms have evolved large and complex ecosystems of planners and managers and squeezed out the population of ignorant, low-paid developers. That is, traditional firms typically follow the Pareto Principle, where 80% of the workforce are planners and less than 20% of the workforce are developers. This exacerbates the problem when this small army of planners and managers divine too many business requirements and IMSs representing centuries of work for a small group of low-paid developer slaves to implement (like making bricks without straw). Furthermore, let's put all of the planners and managers in the Western hemisphere and all of the low-paid developers in developing nations (pun intended)! Conversely, SAFe suggests that the future cannot be divined, smaller short-term plans must be created for no more than a few weeks or months, and the developers should create them. SAFe plans are goal and objective oriented roadmaps, representing a small number of rapid business experiments in which to tease out hidden market needs, not tasks, work packages, nor modules to be integrated later.

While this all sounds fine and dandy, the failure of long-term divination vs. short-term adaptive planning, it's certainly a little challenging to achieve—That is, realize the promise, payoff, and ROI of lean-agile planning frameworks like SAFe. Even smaller lean-agile frameworks are fraught with pitfalls. Humans are humans, especially hardheaded, fiercely individualistic adult humans in the Western hemisphere. It's quite challenging, dare we say impossible, to get a small team of Western planners and developers to communicate, collaborate, and cooperate together in the short, medium, and long-term at a sustainable pace, because, quite frankly, every man, woman, and child is in it for themselves in the West. Again, teams of adult planners and developers in the Western hemisphere is kin to herding cats or watching an old movie about the Keystone Cops in 1915. Typically, when a public or private sector organization charters multiple small lean-agile teams, it's quite difficult to get more than one of them to optimize their performance, and when team does get good for a little while, it has a suboptimizing effect. Let's say a small development team becomes high-performing developing hundreds of innovative software modules a day, but sales, marketing, deployment, infrastructure, support, maintenance, etc. are stuck in the stone ages. Then, does it really matter how many software modules the high-performing, sub-optimizing development team produces per day? Not really—If a tree falls in the forest, but no one is around to hear it, does it make any noise? SAFe was built to bring all closely interrelated teams on a portfolio, large solution, or single product or program team together, have them create a feasible near-term team of teams plan, and, of course, execute it to the benefit of the market, customers, end-users, supplier firm, and, yes, the team of teams itself.

Part of the solution to successfully applying SAFe planning for rapidly implementing innovatively new products and services lies in understanding the fundamental difference between traditional and lean-agile thinking. If you don't understand the difference between their basic philosophies, then you'll simply attempt to use SAFe to gather a large team of managers to firehose a small team of foreign developers with too many requirements for a decade. You can also turn the SAFe plan into an IMS and use EVM to manage the SAFe team for all intents and purpose. Use SAFe for traditional thinking and you'll get the same result—Failure! Instead, apply SAFe the way it was meant to be used—as a framework for innovation—Treat developers equally and allow them to create their own near-term plans, keep these plans lightweight, and construct business experiments instead of an army of interchangeable cogs, and things will work out just fine. Once again, use SAFe badly, and its payoff, business value, and ROI will be hard to achieve. Conversely, learn and master SAFe, select a manageable sized-ART, train and certify people appropriately, setup a proper lean-agile LACE that eats-its-own-dogfood as well as lean-agile product management team, and develop a small, skeletal near-term bottoms-up plan together, and things will turn out just fine. That is, you'll quickly realize the payoff, promise, and ROI of using SAFe planning. The SAFe Implementation Roadmap comes replete with training wheels to help any team of teams get started and achieve near-term ROI but SAFe requires CAREFUL experience, training, practice, and continuous improvement.

A Short List of Economic Benefits for Applying Scaled Agile Framework Planning Events

- Faster planning & replanning
- Faster measurement & course correction
- Focus on earlier & faster value delivery
- Faster alignment & realignment
- Participatory planning, budgeting, and design
- Better adaptability to change
- Optimal teamwork & collaboration
- Optimal planning & communication speed
- Faster team formation & trust building
- Better ownership & accountability
- Valid, just-in-time technical planning
- Bottoms up estimating & commitments
- Improved communications
- Improved cooperation
- Improved business value
- Fewer requirements defects
- Reduced uncertainty & risk
- Improved planning accuracy
- Improved morale
- Improved visibility
- Improved transparency
- Improved trust
- Improved empowerment
- Improved lead and cycle times
- Early retrospectives & continuous improvements
- Fast transition to lean-agile thinking
- Measure performance earlier
- Minimize tyranny of the urgent
- Bottoms-up decision-making & innovation
- Improve business outcomes
- Better conflict management
- Clear roles & responsibilities
- Faster decision-making
- Communicate clear vision
- Standardize architecture assumptions
- Avoid team suboptimization
- Ensures holistic systems thinking
- Steady development rhythm
- Better scheduling of key events
- Adapt to changing priorities
- Avoid program & team overload
- Greater psychological safety
- Better stakeholder alignment
- Match capacity & demand
- Ensure commitment, ownership, & engagement
- Resolve dependencies earlier
- Quickly resolve impediments & roadblocks
- Standardize ALM, DevOps, & technical platform
- Improve end-to-end quality & reliability
- Deliver business value fast
- Minimize delays, bottlenecks, & frozen queues
- Easily enforce lean thinking values & principles,
- Limit WIP, optimize workflow, & reduce lead times
- Get faster market, customer, & end-user feedback
- Minimize & manage technical debt
- Maintain velocity, sustainable pace, & work-life balance
- Rapid business experimentation & hypothesis testing
- Rapid problem identification & solving w/design thinking

Costs, Benefits, and ROI of Using Scaled Agile Framework (SAFe) Planning Events

Cost Item	Cost Terms	Small	Medium	Large	Solution
		50	**75**	**125**	**225**
Training	Materials + Hours x Rate	$165,000	$247,500	$412,500	$742,500
Planning	People x Days x Hours x Rate	$160,000	$240,000	$400,000	$720,000
Tools	People x Annual Cost	$6,000	$9,000	$15,000	$27,000
Facililty	Day Rate x Days x Rooms	$1,000	$1,500	$2,000	$4,500
Food	Breakfast + Lunch + Snacks	$3,750	$5,625	$9,375	$16,875
Materials	Flipcharts + Postits + Markers, etc.	$500	$750	$1,250	$2,250
50% Travel	Airline + Hotel + Meal + Uber, etc.	$43,125	$64,688	$107,813	$194,063
F2F Costs	1 Years	$1,022,500	$1,533,750	$2,554,250	$4,601,250
	3 Years	$2,737,500	$4,106,250	$6,837,750	$12,318,750
	5 Years	$4,452,500	$6,678,750	$11,121,250	$20,036,250
3 Year ROI	50% ROI @ Low End	$12,262,500	$18,393,750	$30,662,250	$55,181,250
Remote Costs	1 Years	$829,000	$1,243,500	$2,072,500	$3,730,500
	3 Years	$2,157,000	$3,235,500	$5,392,500	$9,706,500
	5 Years	$3,485,000	$5,227,500	$8,712,500	$15,682,500
3 Year ROI	70% ROI @ Low End	$18,843,000	$28,264,500	$47,107,500	$84,793,500

Further Reading

- Campbell-Pretty, E. (2016). *Tribal unity*: *Getting from teams to tribes by creating one team culture*. Richmond, Vic: Pretty Agile.
- Campbell-Pretty, E., & Wilson, A. L. (2019). *The art of avoiding a train wreck*: *Practical tips and tricks for launching and operating SAFe agile release trains*. Pretty Agile Pty.
- Dalton, J. (2019). *Great big agile*: *An os for agile leaders*. New York, NY: Apress Media.
- Hopp, W. J. (2008). *Supply chain science*. Long Grove, IL: Waveland Press.
- Kane, L. (2018). *SAFe PI planning*: *A step-by-step guide*. Torrance, CA: Agile One Media.
- Kaye, B. & Jordan-Evans (2014). *Love em or lose em*: *Getting good people to stay*. San Francisco, CA: Berrett-Koehler.
- Knaster, R., & Leffingwell, D. (2018). *SAFe distilled*: *Applying the scaled agile framework for lean enterprises*. Boston, MA: Pearson Education.
- Leffingwell, D. (2007). *Scaling software agility*: *Best practices for large enterprises*. Boston, MA: Pearson Education.
- Leffingwell, D. (2011). *Agile software requirements*: *Lean requirements practices for teams, programs, and the enterprise*. Boston, MA: Pearson Education.
- Leffingwell, D. (2018). *SAFe reference guide*: *Scaled agile framework for lean enterprises*. Boston, MA: Pearson.
- McConnell, S. (2019). *More effective agile*: *A roadmap for software leaders*. Bellevue, WA: Construx Press.
- Pearson, C., & Porath, C. (2009). *The cost of bad behavior*: *How incivility is damaging your business and what to do about it*. New York, NY: Penguin Group.
- 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.
- Qvist-Sorensen, O., & Baastrup, L. (2020). *Visual collaboration*: *A powerful toolkit for improving meetings, projects, and processes*. Hoboken, NJ: John Wiley & Sons.
- Reinertsen, D. G. (2009). *The principles of product development flow*: *Second generation lean product development*. New York, NY: Celeritas.
- 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). *Lean & agile organizational leadership*: *Establishing vision, guidance, & trust to unleash enterprise competitiveness*. Retrieved February 20, 2018, from http://davidfrico.com/rico18g.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). *32 attributes of successful continuous integration, continuous delivery, and devops*. Retrieved September 27, 2019, from http://davidfrico.com/devops-principles.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
- Rico, D. F. (2019). *Business value of lean thinking*: *Capitalizing upon lean thinking principles to rapidly create innovative products and services*. Retrieved November 11, 2019, from http://davidfrico.com/rico19d.pdf
- Rico, D. F. (2019). *Evolutionary architecture*: *24 principles of emergent, organic, and highly-adaptive design*. Retrieved September 3, 2019, from http://davidfrico.com/evolutionary-architecture-principles.pdf
- Rico, D. F. (2019). *Lean & agile enterprise frameworks*: *Using SAFe 4.6 to manage U.S. government agencies, portfolios, and acquisitions*. Retrieved June 12, 2019, from http://davidfrico.com/rico19b.pdf
- Rico, D. F. (2019). *Lean leadership*: *22 attributes of successful business executives, directors, and managers*. Retrieved August 28, 2019, from 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. (2019). *Top SAFe videos*, Retrieved April 7, 2019, from http://davidfrico.com/top-safe-videos.htm
- 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 CI, CD, & DevOps(Sec): Scaling up to billion user global systems using containerized microservices. Retrieved January 26, 2020, from http://davidfrico.com/rico20a.pdf or http://youtu.be/qrWRoXSS9bs
- 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). *SAFe 5.0 videos*, Retrieved January 21, 2020, from http://davidfrico.com/safe-5.0-videos.htm
- 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., 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.
- Ries, E. (2017). *The startup way*: *How modern companies use entrepreneurial management to transform culture and drive long-term growth*. New York, NY: Crown Publishing.
- Rothman, J. (2016). *Agile and lean program management*: *Scaling collaboration across the organization*. Victoria, Canada: Practical Ink.
- Smith, P. G. (2018). *Flexible product development*: *Agile hardware development to liberate innovation*. San Francisco, CA: Jossey-Bass.
- Soukath-Ali, M. M. (2017). *Get SAFe now*: *A lightning introduction to the most popular scaling framework on agile*. Chennai, India: Ali Publications.
- Thompson, K. W. (2019). *Solutions for agile governance in the enterprise* (*SAGE*): *Agile project, program, and portfolio for development of hardware and software products*. Vancouver, CA: Sophont Press.
- Yakyma, A. (2016). *The rollout*: *A novel about leadership and building a lean-agile enterprise with SAFe*. Boulder, CO: Yakyma.

Case Studies of Maximizing the ROI of Scaled Agile Framework (SAFe) Program Increment (PI) Planning

- Global Energy Firm. *The first case study involves a multi-billion-dollar global energy firm. One of its recent acquisitions was a 100-year-old regional energy operating unit. It had a very antiquated information technology department steeped in traditional thinking with a very low-level operating budget. It wasn't in a hurry to implement new information technology systems, because energy is a virtual monopoly (i.e., there is no market competition or impetus for change). To add insult to injury, the information technology department had a laissez faire culture and it wasn't in a big hurry to create innovatively new products and services. Most of its projects had 10-or15-year integrated master schedules (IMSs), it had a telecommuting policy, empty all-glass high-rise buildings, and much of its workforce were busy with community-building and other public welfare activities. After their acquisition, their corporate-level CIO asked this sleepy information technology department to undergo a digital transformation initiative, adopt the latest lean-agile frameworks, and reinvent its underlying information technologies to be more customer centric. The employees took this to heart, stood up a design thinking group, built some journey maps, and planned a series of new mobile apps to help customers create online accounts and manage their utility billing services. That was the easy part. Now the real work began of creating a lean-agile department at the portfolio, large solution, product team, and even individual application levels. The first wave of digital transformation consisted of helping the design thinking group create some near-term product roadmaps and standup individual lean-agile teams to begin implementing solutions to satisfy these UX goals. The next wave of digital transformation consisted of implementing Essential SAFe to herd all of the individual lean-agile teams together to implement a broader ecosystem of innovatively new products and services. Basically, one or two teams out of dozens were able to begin implementing individual applications in a lean-agile fashion, but the other two dozen teams were a bit slow to get started. Perhaps using Essential SAFe, this sleepy information technology department could get all of the lean-agile teams iterating together and a larger ecosystem of innovatively new products and services could be implemented quickly to satisfy the corporate-level CIO's vision. Three of the 20 or so teams were selected for an initial SAFe pilot, the teams were trained in basic SAFe practices, and SAFe planning was started quickly. A big room venue was identified, materials were obtained, along with food and refreshments, and everyone was assembled. The teams stormed their individual team boards, devised program increment objectives, created a joint program board, identified risks, etc. SAFe planning was just what the doctor ordered, 75 people now had a common vision, purpose, and plan, forgotten middle managers and developers felt confident they were getting their money's worth, and people felt empowered, included, and valuable. The new SAFe ART joined the single high-performing lean-agile team, except the SAFe ART worked at a sustainable pace vs. 80 hours a week like the other.*
- Public Sector Network Agency. *The second case involves a large multi-billion-dollar public sector network agency. The agency had been operating in the public sector for more than 60 years and created custom telecommunication systems for federal agencies. It was conceived in an era where public sector agencies did not acquire and apply public services such as commercial telecommunication systems but contracted with commercial firms to build custom systems. Much of the public sector has been moving towards the acquisition and use of commercial telecommunications systems more and more, especially over the prior two decades. This particular situation was a long-forgotten leaf-node program operating 40-to-50-year-old traditional brick and mortar networks for moving public sector data around the globe. With the speed of global innovation, especially in information technology networking, where billions of Internet hosts and devices exchange trillions of packets of information and transactions, it was simply impossible for this program to keep their public sector networks operating properly, reliably, and securely. This fails to mention that this agency wanted to move towards commercial cloud services to replace brick-n-mortar data centers, but they were simply moving to slowly. Therefore, this program decided to make one last-ditch effort to keep up with the speed-of-change by using Essential SAFe to operate and maintain a national family of traditional brick-n-mortar data centers. They hoped that a lean-agile framework such as SAFe could help replace their 5-10-and-15 year integrated master schedules (IMSs) with lightweight adaptable plans to keep up with the pace of change. SAFe coaches were engaged, SAFe planning was organized, 80 people were trained, and SAFe planning was initiated within 90 days. This was clearly a SAFe cold start, that is, most teams on this ART were not using lean-agile methods, nor doing them very well. SAFe training focused on basic lean-agile values, principles, practices, tools, and metrics, including those of SAFe. All teams created lightweight, adaptable plans, synchronized their cadence to iterate together, and executed their first iterations. The teams spent the first 90 days getting used to the lean-agile practices and cadence, the second 90 days focusing on delivery, and the third 90 days on optimizing their performance. Customers began seeing their first deliveries in a decade, the SAFe ART satisfied 85% of its program increment objectives by the third 90-day period, and everyone was impressed by the increased team motivation, participation, productivity, transparency, etc. The teams had a hard time adapting to the lean thinking mindset, limiting their WIP, and resisting the temptation to use SAFe planning for creating detailed 90-day integrated master schedules (IMSs). Of course, they were battling a very brittle network in a high-speed global cybersecurity war and didn't quite understand how overplanning was hindering their ability to adapt to exponentially escalating cybersecurity threats, attacks, and vulnerabilities. Customers quickly discovered that the long-term solution was in combining SAFe with commercial cloud services.*
- Public Sector Cloud Migration. *The third and final case involves a large-multibillion dollar public sector healthcare agency. The agency was actually two agencies in one, for two entirely different markets. The smaller one evolved an ecosystem of several hundred legacy applications to collect national healthcare data, analyze national healthcare needs for their market segment, and properly administer federal funds for supplying needed healthcare products and services. Much of its portfolio was comprised of mainframe and application systems technologies from the 1960s and 1970s. It only evolved several hundred applications instead of thousands, because of its ironclad traditional thinking approach to information systems planning and management. That is, it was steeped in a culture of creating 5-10-and-15-year integrated master schedules (IMSs), therefore it hadn't produced any new information systems in over 15 years. In fact, their leaders simply didn't want to create any new information systems due to the immense difficulty of doing so given the complexity of their problem space and the inability of their information technology and planning paradigm to enable rapid development. As cloud technologies came front and center, their leaders decided to acquire the services of a commercial cloud vendor, migrate their critical applications to the cloud, and perhaps even create new ones. The goal was to build APIs between current and future applications and operate them as a single large application instead of disconnected point solutions. They tried using lean-agile frameworks for small teams, but simply couldn't get any productivity out of these projects. For one thing, the agency invested millions of dollars in creating mountains of vague business requirements through which one could drive Mack truck. The agency was happy producing multiple new business requirements documents each year, since they were incapable of producing new information systems. The other problem was that their suppliers were steeped in traditional thinking and resisted the transition to small lean-agile teams in lieu of 15-year project cycles. Their director needed to get their legacy applications to the cloud fast and build new ones too, so he chartered a cloud project and the use of Essential SAFe to help him finish his pilot cloud project as quickly as possible. A SAFe LACE was established, everyone was trained, a product management team was established and created a cloud roadmap, and SAFe planning began in earnest. Although the cloud team was using Essential SAFe, the agency's director also used solution pre-and-post planning events for the public and private sector leadership teams to develop a joint vision for moving to the cloud, feed this vision to Essential SAFe planning events, and synchronize the expectations of the solution space or small portfolio. The cloud team completed multiple MVP business experiments, including enablers for architectural runways, migrations of legacy applications to the cloud, and real-time streaming services for APIs, data mining, and queries. Following the SAFe implementation roadmap, hiring an aggressive RTE, and standing up SAFe planning quickly helped a lot.*