SAFE 5.1: Leading SAFe REFRESHERER

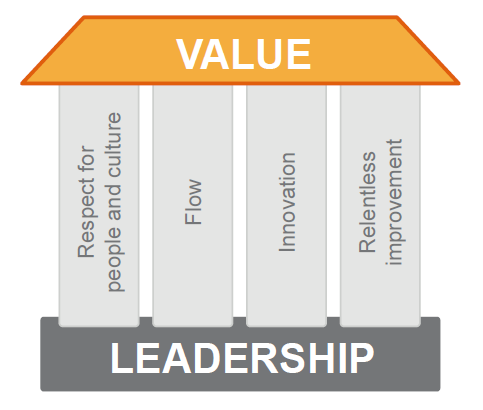
1. We need a dual operating system for Business Agility: Value Stream network + Functional hierarchy
2. Achieving a state of Business Agility means that the entire organization—not just development—is engaged in continually and proactively delivering innovative business solutions faster than the competition.
3. Business Agility: Technical Agility + Business Level Commitment + involvement from everyone + use Lean, Agile Practice
4. SAFe: proven, integrated principles, practices and competencies for achieving Business Agility by implementing Lean, Agile, and DevOps at scale.
5. Business Results: 30% Happy Customer; 50% Faster GTM; 35% Productivity; 50% less defects
6. SAFe Configuration: Essential; Portfolio; Large; Full;
7. 7 Core Competencies:
   1. Team Technical Agility:
8. Agile Teams
9. Teams of Agile Teams (Agile Release Train)
10. Built in Quality
    1. Agile Product Delivery:
11. Customer Centricity and Design Thinking;
12. Develop on Cadence & Release on Demand;
13. Devops and Continuous Delivery Pipeline : Continue explore; integrate; deploy; release
    1. Enterprise Solution Delivery;
14. Apply Lean System Engineering Practice
15. Coordinate Train and Suppliers
16. Continually Evolve Live System
    1. Lean Portfolio Management;
17. Align, Strategy, Funding and Execution
18. Optimize Operations across Portfolio: Agile Portfolio Operation
19. Lightweighted governance and Decentralized decision making : Lean Governance
    1. Organization Agility;
20. Create Lean Agile Mindset Organization wide
21. Map & continuously improve Business process
22. Respond quickly to Opportunity& Treats
    1. Continuous Learning Culture;
23. Learn and Grow Together
24. Creativity and Innovation Culture
25. Relentless Improvement (Plan/Do/Check/Act)
    1. Lean Agile Leadership:
26. Lead by example, inspire others
27. Lean Agile Values & Principle
28. Lead the Change

Customer Centricity in Core

1. Measure and Grow is the way each Portfolio evaluate progress towards Business Agility

**Chapter 2: Becoming Lean Agile Leader**

1. SAFe Core Values: 4
   1. Alignment
      1. Communicate the mission, vision, and strategy
      2. Provide briefings and participate in PI Planning
      3. Participate in backlog review and preparation
      4. Organize around Value Streams
      5. Constantly check for understanding
   2. Transparency
      1. Visualize all relevant work
      2. Take ownership and responsibility for errors
      3. Admit your own mistakes
      4. Support others who acknowledge and learn from their mistakes—never punish the messenger
   3. Built in Quality
      1. Refuse to accept low-quality work
      2. Support investments in technical debt reduction
      3. Ensure UX, architecture, operations, security, compliance, and others are part of the flow of work
   4. Program Execution
      1. Participate as an active Business Owner
      2. Celebrate high quality and predictably delivered PIs
      3. Aggressively remove impediments and demotivators
2. Lean Thinking
   1. Identify Value Stream of each Product
   2. Make value flow without interruption
   3. Let Customer pull value from producer
   4. Pursue perfection
3. SAFe house of Lean (Lean Thinking Mindset):-
   1. Top: Value – Best Quality & High Morale, Customer Delight
   2. Pillars: Respect People and Culture- Generative Culture; Build Long Term Partnership; To change culture, change the Organization9
   3. Pillar: Flow- Optimize sustainable value delivery; Build in quality; Understand, exploit and manage variability; Move from Project to Products
   4. Pillar: Innovation- Innovative People; Provide time & space for innovation; Go See; Experimentation & Feedback;
   5. Pillar: Relentless Improvement: Constant sense of Danger; Optimize the whole; Reflect at Key Milestone; Base improvement on facts
   6. Base: Leadership: Lead by example; Adopt Growth Mindset; Exemplify the values and principle of Lean-Agile & SAFe; Develop People, Lead the Change; Foster Psychological Safety



1. The Agile Manifesto
   1. 4 Values of Agile Manifesto:-
      1. Individual and Interactions
      2. Working Software
      3. Customer Collaboration
      4. Respond to Change
   2. The Agile Manifesto Principles: 12
      1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
      2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
      3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference for the shorter timescale.
      4. Business people and developers must work together daily throughout the project.
      5. Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done.
      6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
      7. Working software is the primary measure of progress.
      8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
      9. Continuous attention to technical excellence and good design enhances agility.
      10. Simplicity—the art of maximizing the amount of work not done—is essential.
      11. The best architectures, requirements, and designs emerge from self-organizing teams.
      12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.
   3. SAFe Lean Agile Principles: 10

Lean Agile Transformation Delivers substantial benefits; Understanding principle assure the change moves the Enterprise in right direction

#1 Take an economic view

* + - * Deliver Early & Often
      * Deliver Value Incrementally
      * Early Delivery has higher Value
      * Solution Economic Trade-off : Sequence job for maximum benefit; Do not consider money already spent; Make economic choices continuously; Empower local decision making: Value- Lead Time- Product Cost- Development Expense : Risk

#2 Apply systems thinking

* + - * Optimizing one component doesn’t optimize the system
      * Higher level understanding of System behavior and architecture is required
      * The value of system passes through its interconnections
      * A System can evolve no faster than its slowest integration point
      * Optimize full value stream: Most problem with your process with surface as delay; Reducing delays is the fastest way to reduce Time to Market
      * Focus on the delays: Flow Efficiency

#3 Assume variability; preserve options

* + - * Development occurs in uncertain world: Requirements must be flexible to make economic design choices; Design must be flexible to support changing requirements
      * Preservation of options improve economic results
      * Flexible specification; Design Set -> Cone of Uncertainty
      * Set Based Approach; require less rework against Point based approach

#4 Build incrementally with fast, integrated learning cycles

* + - * Apply Fast Learning cycle: Iterative learning Cycle (Plan, Do, Check, Adjust): Improving learning efficiency and reduces time between action and effect; Reducing the cost of risk by truncating unsuccessful path quicky; Shorter the Cycle, Faster the Learning
      * Integration point control production development: Development can proceed no faster than slowest loop; Integration point accelerate learning; Synchronization of design loops
      * Integration point reduces risks

#5 Base milestones on objective evaluation of working systems

* + - * Phased Gate Issue: force design decision too early; assume “point” solution exists, can be build correctly first time; huge batch, long queue; fix requirement, design too early and late adjustment too costly
      * Apply objective milestone: Program Increment (PI) System Demo
      * Iterate to objective solution: facilitate learning, adjustment to optimum solution

#6 Visualize and limit WIP, reduce batch sizes, and manage queue lengths

* + - * Visualize to increase understanding
      * Reduce batch size for higher predictability: reduce variability
      * Small batches go through the system faster with lower variability
      * Finding optimal batch size: Total cost= Transactional cost + Holding cost; Higher holding cost reduces Optimal batch size
      * Reducing Transactional cost reduces total cost, lowers optimal batch size
      * Longer queues: longer lead time, increases risk, more variability, lower quality
      * Reduce queue length: wait time= queue length/processing rate

#7 Apply cadence, synchronize with cross-domain planning

* + - * Candence:
        1. Convert unpredictable events into predictable occurrences & lower cost
        2. Make wait time for new work predictable; Limit batchsize to a single interval
        3. Support regular planning and cross functional coordination
        4. Provides scheduled integration points
        5. Slowest component drags the train
      * Synchronization:
        1. Cause multiple events to happen simultaneously
        2. Provides routine dependency management
        3. Provides multiple feedback perspective
        4. Synchronize to assure delivery
      * Control variability with planned cadence
      * Synchronize with cross domain planning

#8 Unlock the intrinsic motivation of knowledge workers

* + - * Managing knowledge workers, workers must be heard and respected
      * Autonomy
      * Mastery
      * Purpose

#9 Decentralize decision-making

* + - * Openly discuss how decisions are made
      * Provide clarity on Organization objectives, coach and cultivate decision making
      * Except decision for Long lasting, infrequent and significant economy of scale- decentralize everything

#10 Organize around value

* + - * Value doesn’t follow silo; political boundaries can prevent cooperation, communication across silos difficult
      * Organize around Development Value Stream
      * Recognize opportunity through release and validation
      * Value flow across the Organization boundaries

**Chapter 3: Establishing Team and Technical Agility**

Agile Team and Teams of Agile teams create and support the business solution that delivers value to Enterprise’s Customers.

3.1 Forming Cross Functional Agile Team: Agile teams are cross functional, self organizing entities that can define, build, test and deploy increment of value

- Optimized for communication and delivery of value

- Delivery value in every 2 weeks

- 2 roles: Scrum Master, Product Owner

- 5-11 team members

- Create and refine stories and acceptance criteria’s

- Develop and commit to team PI objectives and Iteration Goal

1. Scrum Master:

- Coaches the Agile Team in self-management

- Helps the team focus on creating increments of value each Iteration

- Facilitates the removal of impediments to the team’s progress

- Ensures that all team events take place, are productive and kept within the timebox

1. Product Owner:

- Contributes to Vision and Roadmap

- Acts as Customer for Team’s questions

- Prioritize team backlog

- create, communicate and accept user stories

- Team visualize flow with Kanban and helps optimize

- Extend int Business with Agile Business Teams

3.2 Build in Quality: ensure every increment in the Solution reflects quality standards

- Required for sustainable development velocity

- Agile quality practice apply to every team Business or Technology

* + Establish Flow
  + Peer Review and Pairing
  + Collective Ownership and Standards
  + Automation
  + Definition of Done

- Built -in Quality practices for Software Teams

* TDD : Test Driven Development – Agile Team Unit test
* BDD: Behavior Driven Development – Product Owner- Development Team
* Lean UX: Product Owner- Customer
* Agile Testing
* Refactoring, Code Quality, Agile Architecture

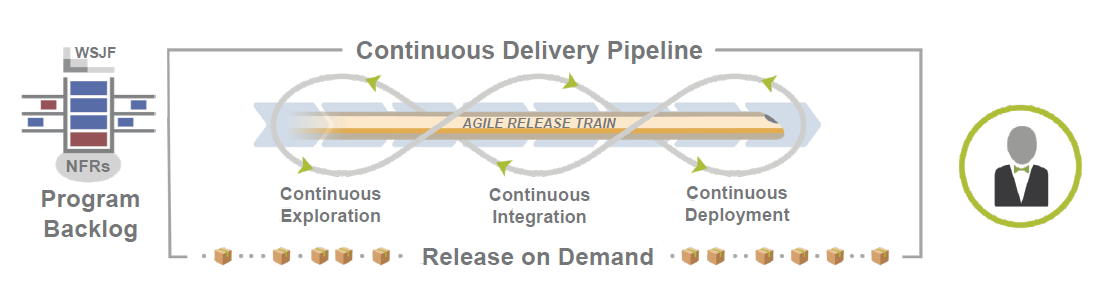
- Built-in Quality practices for Hardware Teams: Support Hardware testing by exploratory, early iteration, frequent system level integration, design verification, Model Based System Engineering (MBSE), Set based design

3.3 Organizing Release Trains around the Flow of Value

- ART a virtual organization: 5 -12 Teams (50 – 125+ Individuals): cross functional

- synchronized on common cadence and program increment (PI)

- align to common mission by a single Program Backlog



- Stream Aligned Team: deliver value to Customer/end use

- Complicated Subsystem Team: deep speciality skills and expertise for specific subsystem

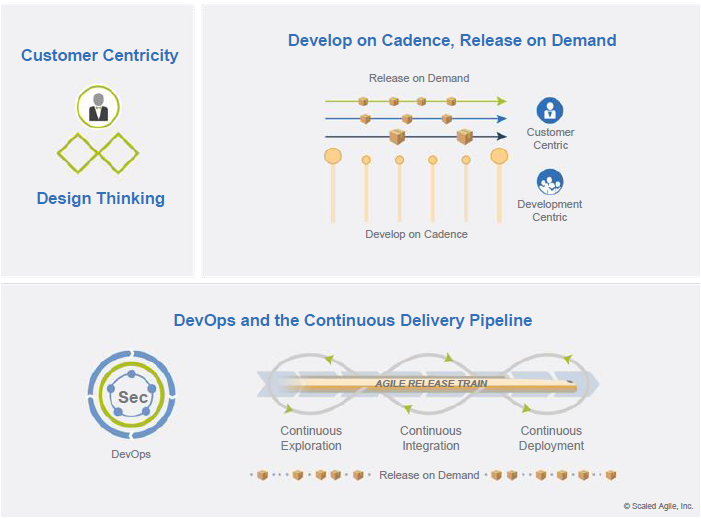
- Platform Team: development & support of platform

- Enabling Team: organized to assist other teams

- Roles on Agile Release Train:-

* Release Train Engineer (RTE) acts as Chief Scrum Master
* System Architect/Engineering provides architectural guidance & technical enablement to team
* Business Owners
* Product Management: owns, define, prioritize the Program Backlog
* The System Team provides tools and process to integrate and evaluate the asset early & often

**Chapter 4: Building Solutions with Agile Product Delivery**



**4.1: Customer Centricity and Design Thinking**

- Customer Centric Businesses generates-

- Greater Profits

- Increased employee engagement

- More satisfied customers

- Customer centric government and Non Profits create resiliency, sustainability and alignment to fulfil mission

Customer centric Enterprise deliver whole product solution designed with deep understanding of Customer need.

* Understand the Customer needs
* Think like Customer
* Focus on Customer
* Know Customer Lifetime Value
* Build whole product solution

Design Thinking is clear and continuous understanding of target market, customers, the problems they are facing and the jobs to be done

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Epic and Features | Understand the problem (Problem Space) | Discover: Diverge | Gemba Walks |  |
| Define: Converge | Personas, Empathy maps |
| Design the right solution (Solution Space) | Develop: Diverge | Journey Maps, Story Mapping, Prototyping |
| Deliver: Converge | Prototyping |

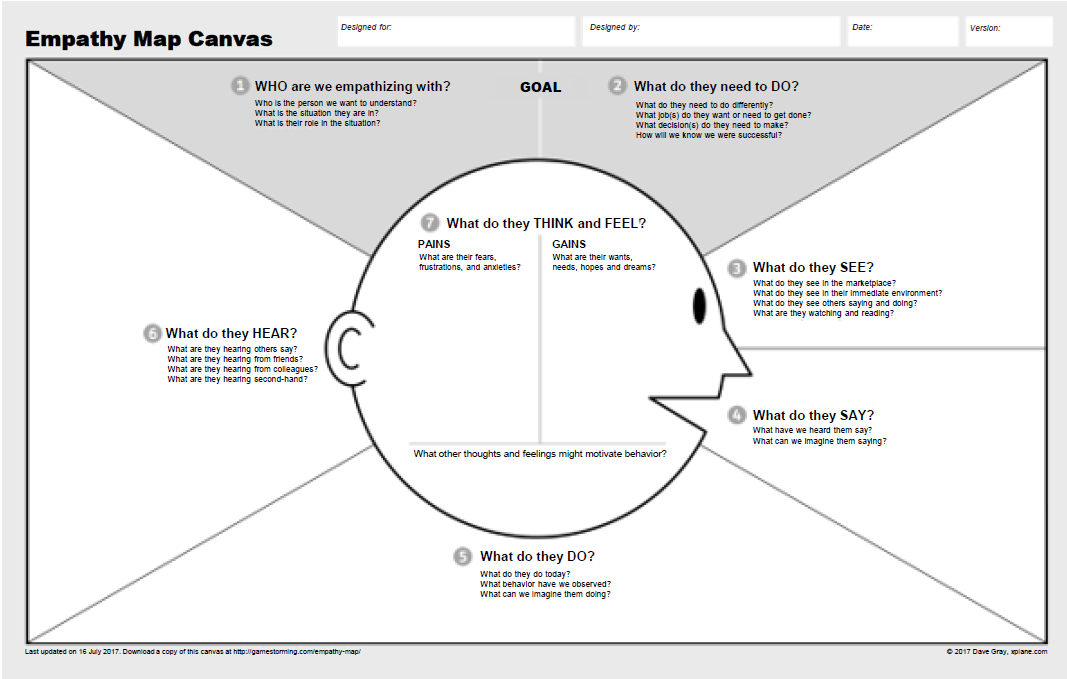
DESIRABLE – VIABLE – FEASIBLE: SUSTAINABLE

Personas are fictional characters that represents the different people who might use your product. They:-

1. convey the problems they are facing in context and key triggers for using the product
2. capture concise information that inspired great products without unnecessary details

Empathy Map : tool helps team develop deep, shared understanding and empathy of the Customer; use it to design better user experience and value streams

1. who are we empathising with
2. what do they need to do
3. what do they see:
4. what do they say
5. what do they do
6. what do they hear
7. what do they think and feel: Pains; Gains



User Journey Map to design End to End Customer Experience

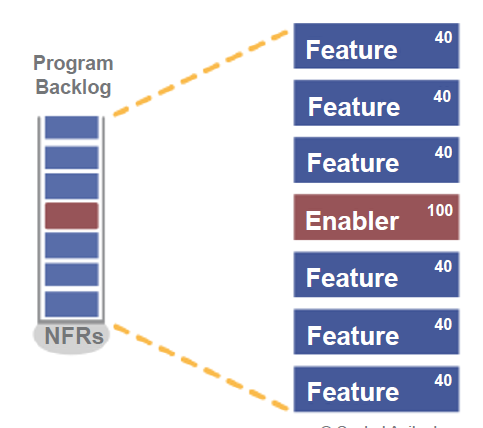
1. Decide: Assess Options
2. Learn: Clarify Goals
3. Choose: Choose what to buy
4. Apply: ID & Financial Check
5. Purchase: Negotiate Sale
6. Complete: Celebrate

Use Story Maps to capture workflows: Stories within an activity are prioritized from Essentials to Delight

**4.2 Prioritize Program backlog**

- Features are managed through Program Backlog

* Program Backlog is the holding area for upcoming features that will address user needs and deliver business benefits for a Single Agile Release Train ART



- Vision aligns everyone on the product directions: description of the future state of Product

* How will out product solve Customer problem
* Features does it have
* USP- how will it differentiate
* What NFR does it deliver

- Feature represents the work for Agile Release Train (ART)

* Feature benefit hypothesis justifies development cost & provide business perspective for decision making
* Acceptance criterias defined during Program Backlog refinement
* Reflects Functional and Non Functional Requirements (NFR)
* Fits in 1 PI

- Features are implemented by Stories

* Stories are small increments of values that can be developed in days and are relatively easy to estimate
* Story User: voice form captures role, activity and goal
* **Features fit in 1 PI for 1 ART; Stories fit in 1 iteration for one team**
* Business Feature ; Enabler Story; User Story

- Estimate Stories with relative Story Points

* A Story Point is a singular number that represents:
  + Volume
  + Complexity
  + Knowledge
  + Uncertainty
* Story points are relative, not connected to any specific unit of measure

- Apply estimating Poker for fast and relative estimating

* Estimating poker combines expert opinion, analogy and disaggregation for quick but reliable estimates
* All member participates

- Estimation is a whole team exercise: increase accuracy by including all perspectives; builds understanding; creates shared commitments

- Prioritize features for optimal ROI

* Job Sequencing is the key to improve economic outcomes
* To prioritize based on Lean economics, need to know 2 things:

1. Cost of Delay (CoD) in delivering value
2. Cost to implement valuable thing

*If you only quantify one thing, quantify the cost of delay*

WSJF = Cost of delay/ Job Duration used to prioritize epic, capabilities and Feature

WSJF: Weightage Shortage Job First

Cost of delay: User Business Value + Time Criticality + Risk Reduction/Opportunity Enablement

Job Duration: Job size;

**4.3 PI Planning**

Program Increment (PI) planning is a cadence based event that serves as the heartbeat of ART, aligning all team on the ART to shared mission and vision

- Occurs for 2 days in every 8-12 weeks

- Everyone plans together

- Product management owns feature priorities

- Development team own story planning and high level estimates

- Architects/Engineers, UX work as intermediaries for governance, interfaces and dependencies

The benefits of PI Planning:

* Establishing communication across all team members and stakeholders
* Aligning development to Business Goals, context, vision, and Team/PI Objectives
* Identifying dependencies and foster cross team/ART collaboration
* Matching demand to capacity and eliminate excess Work in progress (WIP)
* Fast Decision Making

Create Alignment with PI Objectives:

* Objectives are the business summary which team intended to deliver in PI
* Often directly related to Feature in backlog
* Aggregation of set of features; Milestone like trade show; Enabler feature supporting implementation

- Maintain predictability with Uncommitted Objectives

* They are planned and aren’t extra things team can do “just in case if you have time”
* They are not included in commitment, thereby making commitment more reliable
* If a low confidence in one of PI Objective, it should be moved to Uncommitted
* Too many unknowns -> Uncommitted
* Uncommitted objectives are count when calculating Load

- Prepare to experience a simulated PI Planning Event

Simulation flow:

1. You are represented the program vision
2. You will be involved in planning two iterations considering stories and features
3. You will be drafting PI Objectives based on Program Vision
4. Collaborate with business owner to assign Business Value to PI Objectives

- Outcome of PI Planning Simulation:

1. Communication

2. Estimate Capacity

3. Objectives

4. Manage Risks

ART Roles: Executive, Product Manager, System Architect/UX/ Development Manager, RTE

Simulation: Why are we here; gain alignment and commitment to common mission and objectives

|  |  |
| --- | --- |
| Day1 | Day2 |
| Business context | Planning Adjustment |
| Product/Solution Vision | Team Breakouts: Final plan, Impediments, Business owner assign Business Values |
| Architecture Vision and development  Practices | Final Plan Review/Lunch |
| Planning context and lunch | Program Risk |
| Team breakouts: Draft plan , identify risks & impediments | PI Confidence Vote |
| Draft Plan Review | Plan rework if required |
| Management Review and Problem Solving | Planning retrospective and moving forward |
|  |  |

* Simulation -Briefing: Executive, Product Manager, System Architect
* Simulation -Planning Guide:
  + Product Owner- have the content authority to make decision on User Story Level
  + Scrum Masters- manage the timebox, the dependencies and ambiguity
  + Agile Team- define user stories, plan them in iteration and work-out interdependencies with other teams
* Simulation: Planning Requirements
* Simulation: use historical data to calculate velocity i.e. average output from previous iterations
* Simulation: Calculate your capacity

- Activity: Scrum of Scrums (SOS)

* Conducted by RTE
* Each team’s scrum master provides the team’s current status and addresses questions from RTE
* RTE holds the meet-after after the sync (limited to 1-2 topics in simulation)

- Management Review and problem solving: at the end of the day1, management meets to make adjustments to scope and objectives based on the day’s planning

- Day2: Make planning adjustments-

* Based on previous day’s review and problem solving meeting, adjustments are discussed
* Possible changes: Business priorities, Adjustment to vision, Changes to scope, Realignment of work & teams

- Day2: breakout:

* Business owners circulate and assign business values to PI Objectives
* Team finalize Program Increment (PI) plan
* Team consolidate risk, impediments, and dependencies
* Uncommitted objectives provide the capacity and guard band needed to increase the reliability of cadence-based delivery

- Program Board: Feature, delivery, dependencies, and Milestones

Blue: feature; Red/Pink: Significant dependency; Orange: Milestone/Event; Red String: Dependency

- Final Plan Review: Teams and Business Owner review; Changes to capacity and load, Final PI objective with Business Value, Program risk and impediments

- Business owners are asked if they accept the final plan

- Risk: Address Program Risks:

* ROAMing risks: Resolved, Owned, Accepted, Mitigated

- Confidence Vote: team and program

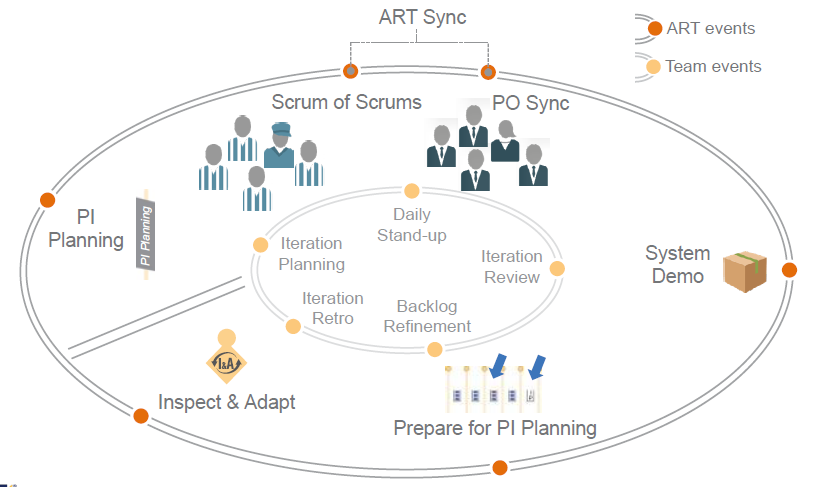
* Commitment : team agrees to do everything in their power to meet the agreed-to objectives
* Escalate immediately if tea feel its unachievable to take corrective actions



**4.4 Develop on Cadence; Release on Demand**

- Manage the flow of work with Program Kanban Board

- ART Events drive the train: creates a closed loop system to keep the train on track



- ART Sync is used to coordinate progress

|  |  |
| --- | --- |
| **Scrum of Scrums** | **PO Sync** |
| ▸ Visibility into progress and impediments  ▸ Facilitated by RTE  ▸ Participants: Scrum Masters, other select team members, SMEs if necessary  ▸ Weekly or more frequently, 30–60 minutes  ▸ Timeboxed and followed by a meet-after | ▸Visibility into progress, scope, and priority adjustments  ▸ Facilitated by RTE or PM  ▸ Participants: PM, POs, other stakeholders,  and SMEs as necessary  ▸ Weekly or more frequently, 30–60 minutes  ▸ Timeboxed and followed by a meet-after |

- Demo the Full System increment in every 2 weeks

- Innovation and Planning (IP) Iteration; estimating guard band for predictable delivery

* Without IP Iteration
  + Lack of delivery capacity buffer impacts predictability
  + Little innovation; tyranny of the urgent
  + Technical debt grows uncontrollably
  + People burn out
  + No time for teams to plan, demo and improve together

- Improve results with **Inspect and Adapt** Event: 3-4 hours per PI; Teams and Stakeholders- 3 parts:

* PI System Demo; End of the PI;
  + led by Product Management, Product Owners and the System Team; attended by BO, ART Stakeholders, RTE, Scrum Masters and Team
* Quantitative, Qualitative measurement;
  + prior to System Demo team review the business value achieved for each of the PI objectives
    - self assess the objective with Business owner
    - planned v/s actual Business value is then rolled up to predictability measure (%)
  + Measure ART Predictability : compares the actual vs planned Business value achieved to
    - Effective process control range
    - Predictability sufficient to run the business
    - Handles common variation
    - Special causes may still cause excess variation
* Problem Solving Workshop
  + Agree on the problems to solve
  + Apply Root cause analysis (RCA): 5 whys: People, process, tools, program and Environment
  + Identify biggest root cause with Pareto Analysis
  + Restate the new problem for biggest root case
  + Brainstorm Solution
  + Identify Improvements as Backlog

**4.5: Building a Continuous Delivery pipeline using DevOps**

DevOps is mindset, culture, set of technical practices -comms, integration, automation among all the people: plan, develop, test, deploy, release and maintain the Solution

* Development: measured on Business Value delivered
* Operation ITSM: Health and Stability to the Production Environment

1. DevOps is just about automation - **Myth**

2. DevOps is a cultural change - **Fact**

3. You don’t need Lean-Agile to have a successful DevOps implementation - **Myth**

4. Agile is for development not operations - **Myth**

5. The deployment pipeline is used to deploy environments as well as solutions - F**act**

6. DevOps tries to bridge the gap between new Features and stable solutions- **Fact**

7. Measurements are an important part of DevOps - **Fact**

8. Automation of testing reduces the holding cost - **Myth**

9. DevOps is only for small software companies - **Myth**

10. Chaos monkey was developed by Netflix - **Fact**

Maximize speed and Stability- Architecture, Business , Security, Development, Compliance and Operations

CALMR Approach to DevOps:

1. Culture of Shared Responsibilities
2. Automation of continuous delivery pipeline
3. Lean flow: batch size small, limit WIP, provide visibility
4. Measurement of the flow
5. Recovery: establish Fast recovery, reversion, fast fix forward- enable low risk release

- Building the Continuous Delivery Pipeline through DevOps

* Each ART builds, maintains, shares a pipeline
* Continuous Exploration: understand Customers need
  + Hypothesis
  + Collaboration, Research: Product Management
  + Architect
  + Synthesis: WSJF, PI Planning
* Continuous Integration: Develop, Build, Test end-end, Stage
* Continuous Deployment: Deploy, Verify, Monitor, Respond

- Decouple release elements from total solution

- Separate Deploy from Release: Timing of Release to be Business Decision; Deploy in production- Enable testing in Prod, Hide new functionality/features

* Release on Demand: Release, Stabilize & Operate, Measure, Learn
* Architect for releasability: Architectural Runway- Enables Build the runway, Feature Consumes it.

**Chapter 5: Lean Portfolio Management**

1. 5.1 Define SAFe Portfolio

- Collection of development value stream: each stream build, deploy and maintain solutions

- Solutions delivered to the Customer: Internal or External to Enterprise

- Enterprise may have Single or Multiple Portfolios as per Size

* Define Portfolio with Portfolio Canvas; Defines the domain of portfolio and key elements; It’s a template to identify specific SAFe Portfolio

- Value Propositions; Key Partners, Activities, resources, Cost Structure, Revenue streams

* Map Solution by Horizon: Evaluating, Emerging, Investing, Extracting, Retiring

1. 5.2 Connecting Portfolio to Enterprise Strategy

3 Phases: Strategy and Investment funding , Lean Governance, Agile Portfolio Operations

* Strategy and Investment funding: collaboration and responsibilities to ensure the portfolio is aligned and funded to create/maintain the solution needed to meet Business targets

Involves: Enterprise Executives, Business Owner, Enterprise Architect

* + Connect the portfolio to Enterprise strategy
  + Maintain a Portfolio Vision
  + Realize Portfolio Vision through Epics
  + Establish Lean Budgets and Guardrails
  + Establish portfolio flow
* Elements of Enterprise Strategy Formation:
  + Vision, mission, core value
  + Enterprise Business drivers
  + Distinctive competence
  + Financial Goal
  + Competitive Environment
  + Portfolio Context: KPIs, Qualitative data, Lean Budget Gaurdrails (policies, practices for budgeting)
  + Portfolio Budget
  + Strategic Themes
* Connect portfolio with Enterprise Strategic Themes
  + Strategic Themes are:
    - Collaboration between Lean Portfolio Management and Large Enterprises
    - Drive Future state of Portfolio
    - Provide context to Portfolio vision and lean budgeting
  + Influence of Strategic Themes
    - Portfolio vision
    - Portfolio Kanban, Portfolio Backlog
    - Vision for Solution, Program and Team Backlog
    - Value Stream Budgets
    - Lean Budget Gaurdrails

1. 5.3 Maintain Portfolio Vision

* SWOT: way to uncover current situation of your value stream, portfolio, product
* TOWS: to identify strategic options to create better future state
* Envision the Future State:-
  + - Portfolio Canvas -> SWOT -> TOWS-> Future state Portfolio canvas -> Identify Epics/Enablers
* Express the future state as vision

1. 5.4 Realize Portfolio vision through Epics

Epic is a significant Solution Development Initiative. Two types:-

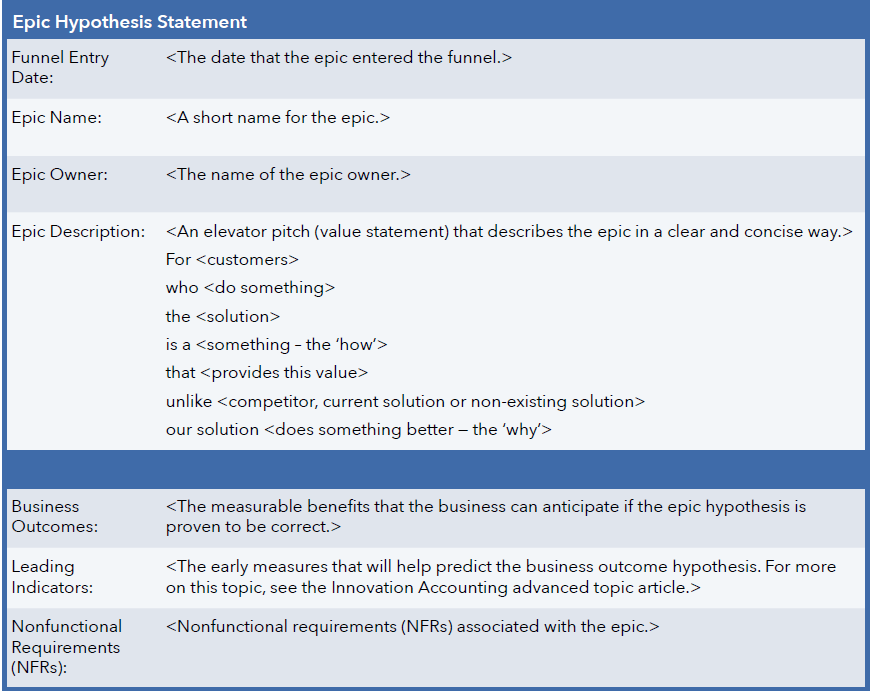
1. Business Epic: directly deliver business value
2. Enabler Epic: support Architecture Runway for future business functionality

- Portfolio epics are cross cutting and typically cut across multiple Value Stream and PIs

- Epics need Lean business case and the definition of Minimal Viable Product (MVP) and approval by LPM

- Epics are initially described as Epic Hypothesis statement: 4 major Fields

* The Value Statement: For Customer- Who do something – The solution..
* Business Outcome Hypothesis: measurable Business benefits
* Leading Indicators: early measure to predict the business outcome
* Non Functional Requirement (NFR)



1. 5.5 Establishing Lean Budget and Gaurdrails
   1. Cost Center Budgeting

- Traditional project based, cost center budgeting creates overhead and friction, lower velocity

- Project requires collaboration of cost centers, assignment of people, budget and schedule

- multiple budgets in a single project budget

Result:

- Slow, Complex budgeting process

- Leads to utilization based planning and execution

- low program throughput

- Moves people to work

* 1. Project overruns cause re-budgeting and increases cost of delay

- wait for new budget approval

- costly variance analysis, blame game, threatens transparency

- resource scramble reassignments

* 1. Solution: Fund Value stream, Not Projects: provides full control of spend

- No costly and dela inducing project cost variance analysis

- No Resource reassignment, No blame game

* 1. Budgets are not affected by Feature Overruns or changing priorities
  2. Maintain the Guardrails :-
     1. Apply Investment Horizons
     2. Utilize Capacity Allocation
     3. Approve Epic Initiative
     4. Continuous Business Owner Engagement

1. 5.6 Establishing Portfolio Flow
2. Govern Epic with Portfolio Kanban
   1. Makes largest business initiatives visible
   2. Brings structure to analysis and decision-making
   3. Provides WIP limits to ensure the teams analyze responsibly
   4. Helps prevent unrealistic expectations
   5. Helps drive collaboration among the key stakeholders
   6. Provides a transparent and quantitative basis for economic decision-making
3. Epics flow through the Portfolio Kanban
   1. Funnel
      1. New Business Opportunities
      2. Cost Savings
      3. Marketplace Changes
      4. Merger, Acquisitions
      5. Problems with Existing Solutions
   2. Reviewing
      1. Refine
      2. Create Epic Hypothesis statements
      3. Preliminary cost estimation, WSJF
      4. WIP limited
   3. Analyzing
      1. Solution Alternatives
      2. Refined cost estimates and WSJF
      3. Define MVP
      4. Create Lean Business Case
      5. Go/NoGo Decision
      6. WIP Limited
   4. Portfolio Backlog
      1. Epics approved by LPM
      2. Sequenced using WSJF
   5. Implementing
      1. MVP:
         1. Build and Evaluate MVP
         2. Pivot or Persevere Decision Made
         3. Pulled by teams
      2. Persevere

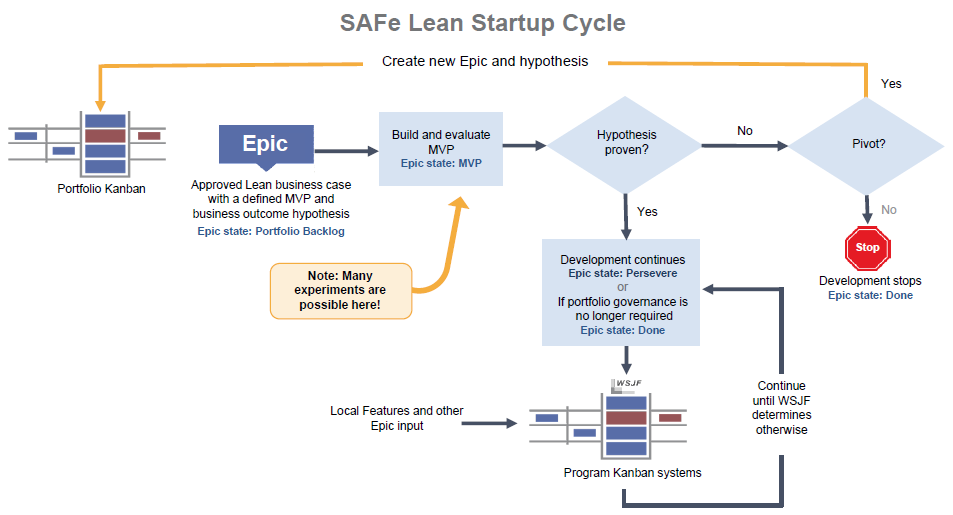
1. Affected ARTs or Solution Train reserve capacity for Epic

Continue feature implementation until WSJF determine otherwise

* 1. Done: when LPM governance no longer required

- Feed the Portfolio Tunnel

- MVPs foster innovation and control scope (SAFe Lean Startup Cycle)



**Chapter 6: Leading the Change**

Lean Agile Leadership

1. 6.1 Leading by Example
2. Authenticity
3. Emotional intelligence: self awareness, self regulation, motivation, empathy and social skills
4. Lifelong Learning: do yourself and encourage others on learning journey
5. Growing Others:
6. Decentralize decision making : move the decision making authority to where the information is
7. Leading the Change
8. Establish sense of urgency
9. Create a powerful guiding coalition
10. Develop the vision and strategy
11. Communicate the vision
12. Empower employee for broad bases action
13. Generate short term wins
14. Consolidate gains and produce more wins
15. Anchor new approach in the culture

SAFe Implementation Roadmap

|  |  |
| --- | --- |
| 1. Waterfall/Adhoc Agile 2. Train Lean-Agile Change Agents 3. Train Executives, Managers and Leaders 4. Identify Value Streams, ARTs 5. Create Implementation Plan 6. Prepare for ART Launch 7. Train team and launch ART 8. Coach ART Execution 9. Launch more ARTs and Value Streams 10. Extend the Portfolio 11. Accelerate |  |