SAFE Agile Exam

**Scaled Agile Framework** – Synchronizes alignment, collaboration, and delivery for large number of teams

**Core Values**

1. Built-in Quality
2. Program execution
3. Alignment
4. Transparency

**SAFE House of Lean –** Achieve sustainably shortest lead time

Leadership -> Respect for people and culture, flow, innovation, relentless improvement -> VALUE

Respect for People and culture – People do all the work, culture change comes last, not first. To change culture, you first have to change the organization

Flow – Always optimize continuous and sustainable throughput of value

Innovation – Producers innovate, customers validate.

Relentless Improvement – Constant sense of danger, optimize the whole, consider facts – then act quickly

Leadership – Lead the change, emphasize life-long learning, develop people, decentralize decision making

**Agile Manifesto**

* *Individuals and interactions* over process and tools
* *Working software* over comprehensive documentation
* *Customer collaboration* over contract negotiation
* *Responding to change* over following a plan

1. Highest priority to satisfy customer through early and continuous delivery of valuable software
2. Welcome changing requirements
3. Deliver working software frequently
4. Business and Tech must work together daily
5. Motivated individuals
6. Face to face conversation for most effective and efficient way to transfer information
7. Working software is the primary measure of progress
8. Sustainable development
9. Continuous attention to technical excellence
10. Simplicity
11. Self organizing teams
12. Retrospective and continuous improvement

**SAFE Lean-Agile Principles**

1. Economic view
   1. Cycle Time, Product Cost, Value, Development Expense, and Risk
2. Apply systems thinking
   1. Value is only added if integrated projects and work streams are synchronized. System can evolve no faster than its slowest integration point
   2. Reducing delays is the fastest way to reduce time to market.
3. Assume variability and preserve options
   1. Set based - the beginning, have variety of options, as time goes on narrow the options to the best fit need
   2. Point based – start with one option, adjust as time goes on –NOT GOOD
4. Build incrementally with fast integrated learning cycles
   1. Plan, Do, Check, Act
   2. Shorter the cycle, the faster the learning
   3. Integration points control product development and reduce risk
   4. Improvement comes through synchronization of design loops and faster learning cycles
5. Milestones are based on objective evaluation of working systems
   1. Phase Gate – Forced early design decisions, encourages false feedback, assumes point solution, creates huge batches and long queues, centralizes requirements an design
      1. Requirements and design are set too early, adjustments can be very costly and late.
   2. PI Demos delivery objective progress, product, and process metrics
6. Visualize and limit WIP limits, batch sizes, manage queue
   1. BVIR – Big Visible Information Radiator
   2. Small batches go through the system faster with lower variability
   3. Large batch sizes increase variability
   4. High utilization increases variability
   5. Most important batch is the **transport(handoff) batch**
   6. Optimum Batch Size = lowest total cost between holding and transaction costs
   7. Reducing transaction costs reduced total costs, and shifts optimum batch size lower
7. Apply cadence, synchronize with multiple teams and projects
   1. Cadence-based planning limits variability to a single interval.
   2. Synchronization – causes multiple events to happen at the same time, facilitates cross-functional trade offs
   3. Design cycles must be synchronized
   4. Synchronize with cross-domain planning
8. Intrinsic motivation
   1. Rewards actually increase error and mistakes
   2. Task can provide its own intrinsic award
9. Decentralize decision making
   1. Centralized
      1. Infrequent, long lasting, significant economies of scale
   2. De-centralized – Frequent and common, time critical, and require local information

**Agile Release Train**

* Organization of 5-12 teams (50-125 individuals), that plans commits and executes together
* **Program Increment(PI)** – fixed timebox, default is 10 weeks
* Synchronized iterations and PI’s
* Program backlog – common mission
* Operated under architectural and UX Guidance
* Synchronization is key to assure delivery and executed integration points
* Cross Functional Agile Teams around **value**
  + Organize for larger purpose
  + Organize team around *features and components*

**Agile Release Train Roles**

1. Release Train Engineer – Chief Scrum Master for the train
2. Product Manager – Product owner on steroids, owns, defines, and prioritizes program backlog
3. System Architect/Engineer – Provides architectural guidance and technical enablement of teams on the train
4. System team – provides processes and tools to integrate early and often
5. Business Owners- Key stakeholders on the ART

PM/PO team steers the release train

**PI Planning** – PI Planning meetings are the pacemaker of the Agile Enterprise

* 2 day ceremony every 8-12 weeks
* Everyone attend in person if possible
* Produce Management owns feature priorities
* Development teams own story planning and high level estimates
* Architect/Engineering and UX works as intermediaries for governance, interface, and dependencies

Input – Vision and top 10 features

Output – Team and program PI objectives are program board

PI Objectives – Business point of view of what each team intends to deliver in the upcoming PI.

* Aggregation of a set of features
* Milestone – trade show
* A major refactoring

Stories -> Features - >Themes/Epics

**Enabler Stories** – Represent different types of work: Exploration, Architecture, Infrastructure

Story Point estimates, volume, complexity, knowledge, and uncertainty – they are relative and not connected to any specific unit of measurement

**SWOT** – Strengths, weakness, opportunites, threats

**Planning Guidance**

Product Owners – content authority to make decisions at the user story level

Scrum Masters – Manage the time box , the dependencies, and the ambiguities

Agile Team – Define and deliver user stories, features, and work out interdependencies

**Stretch Objectives** – Planned, but not committed. Help teams that have a low confidence in PI objective if the item has many unknowns.

**Program Board** – Used to identify feature delivery, dependencies, and milestones

**ROAM-ing Risks**

Resolves – Has been addresses, no longer a concern

Owned – Someone takes responsibility

Accepted- Nothing more can be done, risk occurs, release may be compromised

Mitigated – Team has plan to adjust as necessary

**Program PI Objects** – Synthesis of each teams PI Objectives

**Solution vision** communicates strategic intent

* Where are we headed, what problems do we solve, who are we providing it to?
* User Stories = Business Value
* Features = Benefits

***EPIC -> CAPABILITY -> FEATURE -> STORY***

**Features** – have benefits and acceptance criteria,

**Weighted shortest job first(WSFJ)**

WSFJ – COD(cost of delay)/duration

COD(Cost of Delay) = User business value + Time criticality + RR |OE value)

***Job with the highest WSFJ provides the greatest economic benefit***

**Built in quality** – Software/Hardware quality ensured sustainable development

**Balance between emergent and intentional architecture is required for speed of development and maintainability**

* **Emergent Design** – Teams grow the system design as user stories require
* **Intentional Architecture** – Foster team alignment and defines architectural runway

**Architectural Runway -**  existing code, hardware components, that technically enable near-term business features

* Enablers build up the runway, features consume it
* Runway must be continuously maintained

**Continuous System integration** – Integrate frequently

**Team iteration demo** – provides true measurement of progress by showing working software, functionality, hardware, etc.

**Kanban** helps improve flow by establishing WIP limits, and buffers. Measure flow using the CFD(Cumulative flow diagram)

**Agile Release Train Synchronization**

Scrum of Scrums – Visibility in to progress and impediments

* Facilitated by RTE
* Participants – Scrum Master, other team members, SME’s if necessary
* Weekly or more frequently – 30 – 60 minutes
* Timeboxed and followed by a “meet after”

PO Sync

* Visibility in progress, scope, and priority adjustments
* Facilitated by RTE or PM
* Participants – PM, PO, any other stakeholders, and SME as necessary
* Weekly or more frequent, 30-60 minutes
* Timeboxed followed by a meet after

**System Increment** – Every two weeks features are put together to see if they work in the system

**System Demo** – Demonstrate full solution increment to stakeholders every iteration

**Innovation and Planning iteration**

* Innovation – Opportunity for spikes, hackathons, infrastructure improvements
* Planning – Provides for cadence-based planning
* Estimating guard band for cadence-based delivery

**Inspect and Adapt**

1. PI System Demo
   1. At the end of the PI, teams demonstrate solution to appropriate stakeholders to see if system works as a whole
2. Quantitative measurement
   1. Teams assess planned vs actual business value as part of Solutions demo.
   2. **PI Predictability Measure** – assess whether achievements fall into an acceptable process control band
3. Problem-solving workshop
   1. Teams conduct short Retros to systematically address larger impediments limiting velocity.

***Release value on Demand*** – Chapter 6.6

* Real value only occurs when end users are successfully operating the solution
* Team Increment - > System Increment -> Solutions Increment -> Release
* Definition of Done
  + Team Increment – Stories meet acceptance criteria
  + System Increment – Stories completed by all teams in the ART and integrated
  + Solution Increment – Capabilities completed by all trains and meet acceptance criteria
  + Release – All capabilities done and meet acceptance criteria

**Building an Agile Portfolio – Chapter 7**

*Strategic Themes* – Connect each SAFe portfolio to the enterprise business strategy

*Strategic Themes* – Output of a disciplined strategy formulation process

* Influence ART funding, Portfolio Backlog, Program Vision, and Roadmap

*Value Stream* – Sequence of steps used to deliver value to the customer

* Contains people who do the work, the systems, and the flow of information and materials

*Epics* – Epics are the enterprise initiatives sufficiently substantial in scope as to warrant analysis, understanding ROI, lightweight business case, and approval

Portfolio Kanban – manages flow of epics

Portfolio Backlog - holds Epics approved for implementation

**Lean-Agile Budgeting Chapter 7.4**

Traditional Waterfall – Budget by cost centers – created overhead and lowers velocity; Funding projects can increase cost of delay when over runs happen.

Agile Lean Budgeting – Fund value streams not projects. Increase flexibility and be flexible eto adjust budgets based on changing demands and needs

**Chapter 8 – Building really big systems**

3 Value Stream Roles:

1. Value Stream Engineer – Servant leader that facilitates and guides work of all ARTS and suppliers. RTE on steroids
2. Solution Management – Main content authority guiding value stream, main priority is value stream backlog
3. Solution Architect/Engineer – Technical responsibility for overall architectural engineering and design of the solution

**Solution/Solution Context**

* Solution is uniquely associated with one value stream, defined by solution intent
  + Solution Intent- Single source of truth as to intended and actual behavior of solution (DOORS for FM)
* *Capabilities* describe the higher level behaviors of a solution
* Customer – Critical aspect of development, engaging with customer throughout the process
* Supplier – Overall value streams agility is dependent on supplier’s agility. Significant to role in solutions development

**Chapter 8.2 Coordinate and integrate multiple ART’s and Suppliers**

* Attended by : Customers ,VSE , Solution MGMT, Solutions Architect/Engineers , Value Stream stakeholders and representatives from ART’s and supplier
  + Pre – Helps build an aligned plan for next PI and match solution to demand to ART capacities
    - Input – Results of previous PI, outcomes of solution demo
    - Output – Set of features for every ART
    - Updates to the ART visions
  + Post – Review , recap, communicate and provide feedback
    - Goal – Understand resultant PI plan for entire Value Stream
    - Input – Program PI objectives from all ART’s, value stream planning board, unresolved program risk and confidence vote
    - Output – Consolidated Value Stream PI objectives, adjustments for ART plans

**Solution Demo** – Major event in life of the solution. Entire value stream map fully integrates in the solution.

* Senior managers and high profile stakeholders review the progress, action and investment decisions are based on this objective evidence
* Requires frequent solutions integration and testing to provide best objective evidence

**Value Stream Inspect & Adapt** –

1. Solution Demo
2. Retrospective
3. Problem-solving workshop

**Chapter 9 – Lead the change**

* Life long journey
* Power of “Ba” – Self organization teams/nature
* Leadership styles
  + Expert
  + Conductor
  + Developer
* Create environment of mutual influence for learning, growth, and mutual influence