Table of Contents

SAFe 4.5 Update by SAFe 3
Action 3
Reference 3
Notes 3
Configurability 3
Continuous Delivery Pipeline 3
Continuous Exploration 4
Continuous Integration 4
Continuous Deployment 4
DevOps 4
Lean UX 5
Implementation Roadmap 5
Large Solution level (old value stream level) 5

SAFe 4.5 Update by SAFe

Listen to update pitch of SAFe 4.5

Action

- What are we setting up in terms of telemetry for the AtC
- See implementing SAFe for training for ART
- Read compliance article

Reference

Started at 5 Reasons to Update

Notes

Configurability

Improve big picture usability Too complex Create more white space and reduce duplication

Menu structure improved Why SAFe ← from SAFe distilled

New name - SAFe for lean enterprises To incorporate whole business

Configurable framework Full SAFe - portfolio and large solutions Portfolio SAFe - multiple solutions but not a lot of dependencies (default) Large solution SAFe - don't need portfolio - e.g. DoD Essential SAFe - start here - sometimes it's all you need

Spanning palette icon is smaller set than rest You don't have to have metrics for example But not essential Can still use of course

Continuous Delivery Pipeline

4 phases of the continuous delivery pipeline Continuous exploration Continuous integration Continuous deployment Release on demand Can use parts of it - don't have do it all at once

Drive this through hypothesis, build, measure, learn

Continuous Exploration

Epics are big Chance you need all of it are slim Use lean startup cycle to determine Met MVP or keep going

Changed epic value statement Now have hypothesis statement Includes intermediate measures associated with the anticipated outcomes Being clear “we don't know” upfront

Epic Then continuous exploration

Continuous Integration

Too much focus story level and this is not just about technical problem Need to integrate across whole solution And this is culture as well

Continuous Deployment

Decouple deployment from release Want to deploy to production Use feature toggles etc Get feedback, check you haven't broken anything yet Turn on when necessary Prevent production emergencies

Should be release incrementally Idea of “streamlet” - part of a value stream E.g. Security updates vs release update

New program Kanban Want to go beyond getting ready for implementation So includes steps as it is released as well - more part of total value stream

DevOps

No standard definition of DevOps So have SAFe definition Bridge gaps between dev and ops so deliver value faster and more reliable Reduce large batch

CALMR approach to DevOps Culture of shared responsibility Automate everything Lean flow accelerates delivery Measure everything - application telemetry (business a/b testing, but also the system) Recovery - enable low risk solutions (canary releases etc)

Culture first - interesting not culture last of rest of SAFe Both have to be equally responsible Need input from operations Need tolerance to risk - we have good safety nets (e.g. Test first) Need to share discoveries - learning across silos

Automation still important but need culture as well Match staging environments to production Match development environments to extent the same way Automate deployment - everything under version control, auto build environments, automate the actual deployment process

Lean flow Need to identify bottlenecks, and balance WIP (not through over wall to ops), reduce batch size, manage and reduce queue length

Measure everything Data collected on business, application, infrastructure, client layers Different telemetry for different stakeholders Compare to event - when did we deploy and when did we find the problem

Recovery Stop the line Plan for and Rehearse the failure

Lean UX

Smaller cycle for features Innovation happens at every level of SAFe Feature now have benefit hypothesis As don't really know if we are getting the benefits

Implementation Roadmap

Toolkit executive workshop for tipping point Some minor things as well - get more information Just documenting process that we know works Shows training

Implementing 1-2-3 has been deprecated

SPC is part of big picture Can bring experience from external

Foundation is now SPCs Values Principles Implementing SAFe Lean agile leadership

Large Solution level (old value stream level)

Train of agile trains Large complex systems And a lot of dependencies across the trains And need to be aligned Trains build solutions themselves - this is only large solutions

Compliance Part of solution intent Schedule and manage specific compliance activities Stored in solution intent

Team level Better align with scrum guide Iteration review (vs team demo) Dev team is “anyone contributes to the solution”

Terminology changes Note MVP is part of SAFe

Webinar, SAFe, Video, Learning

From:
https://www.hanssamios.com/dokuwiki/ - **Hans Samios' Personal Lean-Agile Knowledge Base**

Permanent link:
https://www.hanssamios.com/dokuwiki/safe_4.5_updates_by_safe

Last update: **2020/06/04 11:12**