Table of Contents

Essential SAFe - Dean Leffingwell 3
Premise 3
Presenter 3
Attendees 3
Reference 3
Notes 3
Lean agile principles 4
Agile teams and release trains 5
Cadence and synchronization 5
Critical Team and train roles 6
PI planning 7
System demo 7
Inspect and adapt 7
IP iteration 8
Arch runway 8
Lean agile leaders 8
Additional 8

Essential SAFe - Dean Leffingwell

Premise

Essential SAFe Milwaukee Scaling Agile Meetup Sponsored by Icon At Northwestern Mutual

Presenter

Dean Leffingwell

Attendees

Looked like 300 people in the room, friendly audience

Reference

Dean Leffingwell - Essential SAFe

Notes

Local “Cow” story Did visual. When there was a major impediment they put cow a visual cow on a visual track so that it was visible

Leffingwell's history Learn and like agile Started with XP Had failed implementation when started to get more people involved “Can do it 50 people but what about 2000”

Seeing SAFe implementations but some are not successful Question is what are the things that don't make work Some things we've seen include:

- No Inspect and adapt
- No planning and innovation iteration
- Individual art teams not themselves cross functional
- No system Demos

Also hearing

- SAFe is flexible - but we don't use arts
- … didn't include the teams training as “Our teams don't have time for training”
- and so on

“This shouldn't be funny”

Sometimes when Dean says things that aren't comfortable to organization he doesn't get invited back Says it's “a form of WIP limit” for him

But leads to the question “What is essential about SAFe?” Also leads to simplification / more incremental approach if that makes sense Single agile release train

And “Less of eye chart” Doesn't discount Big Picture but that is not appropriate for all

10 essentials of SAFe are:

1. Lean agile principles
2. Agile teams and trains
3. Cadence and sync
4. Critical team and program roles
5. PI planning
6. System demo
7. Inspect and adapt
8. PI iteration
9. Arch runway
10. Lean agile leaders

Rest of notes go through some of these

Lean agile principles

Originally developed as a result of discussion from large scale system engineering. Response to “sure it works for IT / software but what about our stuff for a Satellite system”

On transformation in general we see “Pace of learning is fast; pace of adoption is slow” and we need to be careful about changing SAFe too much

If you are doing an exec presentation, concentrate of the principles as this explains SAFe without much else

Question - “do you see batch sizes; do you see queues” Answer is yes, you now see batch size; queues We know have conversation about WIP These are all changes as a result of how we think about things

If you find a practice isn't working for you, stop here first (i.e. principles), figure out which principles apply and work the issue that way

Book - DevOps handbook - Gene Kim / Jez Humble (new) - https://www.amazon.com/gp/product/B01M9ASFQ3?storeType=ebooks Talks about the same kinds of principles. First half of the book is about the stream of work.

Agile teams and release trains

Idea behind teams and release trains is that if you do it (and based on value streams) you will Invert Conways law (software reflects organizational structure)

Without teams and trains

- Teams are isolated
- Lack of collaboration
- There is over specialization
- Not aligned
- Release late, Big Bang
- No arch / ux integrity
- Too much WIP, multiplexing, low productivity

The set up allows for “Resource pool” (people as resource, this is a lean concept even though we don't like calling people resources) The excess capacity allows work to flow through the system

Concept of centralized and decentralized decision making Idea that some centralized decisions makes sense But also note that concept of central vs decentral is dependent on point of view At my level we might have a central decision process, but if you move up a level, that same decision making process could be decentralized

Cadence and synchronization

Benefits of Cadence

- Transforms unpredictable
- Makes wait times predictable
- Facilitate planning, provide more efficient use of resource

Benefits of Synchronization

- Causes multiple events to happen in the same way
- Sync events facilitate cross functional trade offs of people and scope

Discussion of sprint Sprint is scientific method / plan-do-check-act cycle Program increment is same at higher level

Idea is to apply Scientific method / plan-do-check-act at all levels Scientific method - https://en.wikipedia.org/wiki/Scientific_method?wprov=sfsi1

Idea - have you noticed that teams flex to the work of the ART Perhaps you started with a core technology (component) team, but then found that team did not have any work in the PI Those team members perhaps joined other (feature) teams If we suddenly found we have new work on the core technology area, do we reform the core team? Typically not - we just send the work to one of the teams Result is that team structure is more fluid

Without cadence and synchronization you will find

- Entropy increases
- Get the right people to meeting is impossible
- Integration comes late
- Tendency is to try and shove everything in current timebox
- No forced integration and evaluation people
- Teams are agile, but system not iterating

Discussion on entropy increasing unless you do something - The second law of thermodynamics states that the total entropy of an isolated system always increases over time (https://en.m.wikipedia.org/wiki/Second_law_of_thermodynamics) Planning event is a singulatory Note that you will start getting Entropy immediately after planning The next planning event brings it back together again

Example was 400 people that had a combined velocity of 0 Problem was that while everyone was saying they were working on the work, in reality everyone was working on something else

Critical Team and train roles

Team roles

- Product owner needs to say “no” (add to list of things we want)
- Scrum master means you have discipline
- Team gets things done

Plus Train roles

- RTE
- Product management
- System arch/engineer
- Customer
- Business owner

Without roles

- Responsibly not clear
- Meeting waste of time not clear outcomes
- Teams find hard to integrate
- … (and others)

PI planning

This is the magic It is far more than just a planning event - synchronization, alignment, common understanding, etc

Quote - “PowerPoint animation should be a controlled substance in our company”

Without PI planning

- Teams cannot come together
- “Building software is problem of building a social framework, with people that that aren't social”

So planning helps develop that social network.

System demo

If you don't have a system demo

- You risk waterfall-Ing the PI
- You impact trust - no trust
- Business unclear of results
- There is no forcing function for ci and test automation
- You have individual system behaviors, but your train is not agile

Goal is to have system sprint, not just teams “Show me the system”

Inspect and adapt

A lot of people say this event costs too much and you can't capitalize What proportion of the PI is the I&A - 0.7% The question you have to ask yourself is “why can't we afford to spend 0.7% of time fixing the stuff that lowers our velocity?”

Problem is that if you do not allocate time to do this you won't do it as there is too much pressure to get the next feature

Without the I&A

- Can't improve systemically
- Improvement effort address symptoms not root clauses
- Centralized improvements do reflect reality

Eventually, if this continues, since you are not showing improvement, management will tell you what to do to improve because you are not improving. And it won't be effective as they are too far away from the work

IP iteration

Statement - “Agile not optimized for free association; Agile is optimized for value delivered”

If you have to deliver every sprint there is no time to innovate

So we need to provide capacity for innovation

Without the IP iteration

- No capacity to innovate so it doesn't happen
- No innovate as delivery option

Arch runway

False velocity Arch deteriorates under pressure of now Solution robust reduces over time

Lean agile leaders

Successful transform are based on education management first. They become lean thinking manager - teachers who lead rather than follow the transformation

Concept is that senior management have nice cushy floor with middle management, while the teams have a ceiling beyond which they cannot get the changes they need to become more effective. This means Managers (middle) has to be lean agile leaders - helping both sides, leading, teachers etc.

Additional

In the process of creating SAFe implementation roadmap, beyond the 1-2-3 - coming soon

They are also in the process of developing an RTE course Expect to the hardest course, harder even than SPC, with harder test to qualify They will run a first alpha version in January

Concept Development value stream and operational value stream

On of the key values of safe is that it provides a common taxonomy which improves our ability to communicate Common taxonomy is needed for social network

Meetup, scaling, SAFe, Leffingwell

~~LINKBACK~~ ~~DISCUSSION~~

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

Permanent link:
https://www.hanssamios.com/dokuwiki/essential_safe_-_dean_leffingwell?rev=1477319387

Last update: **2020/06/02 14:28**