33rd Annual INCOSE international symposium

hybrid event

Honolulu HI USA

Mixing Code-Centric and Evidenced-Based development

Visualizing AGILE inside the V

Agenda

- Clashing Cultures
- Analysis Methods
- Conclusions

Presented by:

Robin Mikola and David Hetherington

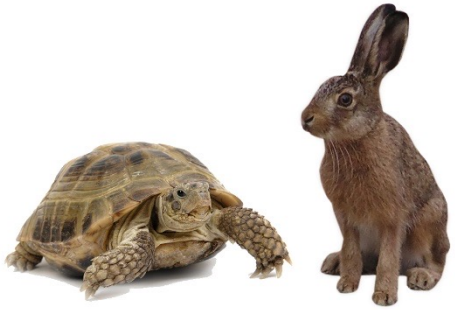
Co-Authors:

Robin Mikola, David Hetherington, Sonya Bixby, and Robert Peters

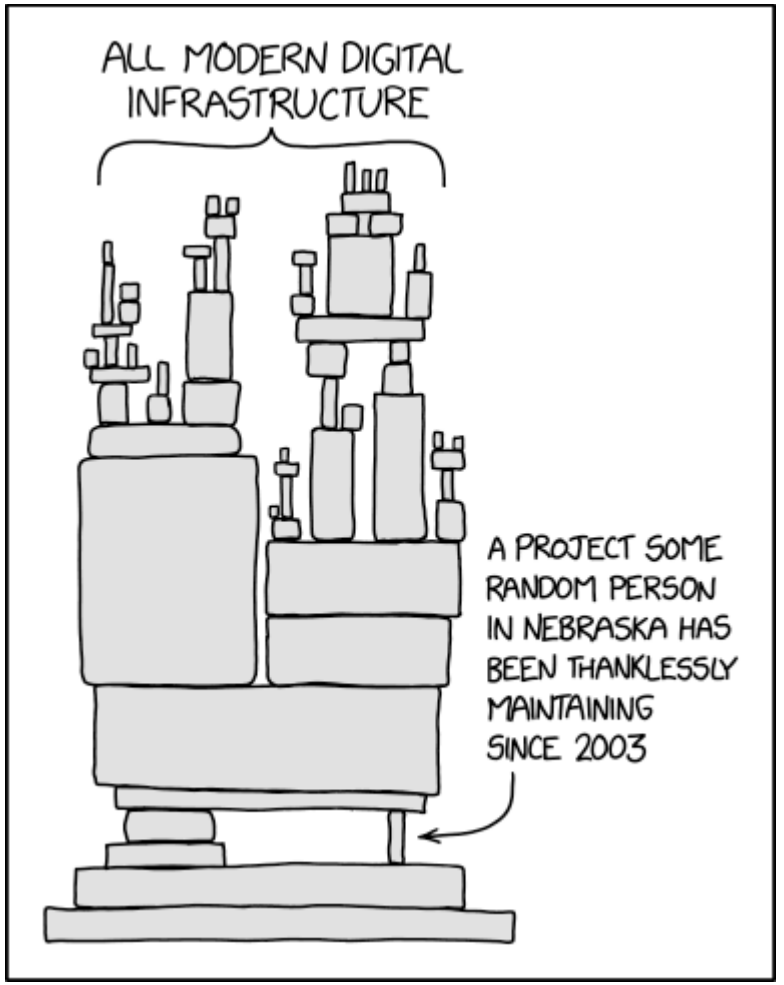
- www.systemxi.com

The Problem

Too many differing perspectives, language and priority



https://www.linkedin.com/pulse/technical-debt-non-tech-people-alex-di-mango



https://xkcd.com/2347/

Why is this so difficult to synchronize?

Clashing Cultures

Not Every Application Needs Careful Planning

If you are developing a "Cat Eating Cheeseburger" social media application, overly formal requirements, modeling, and documentation approaches would be counter-productive.

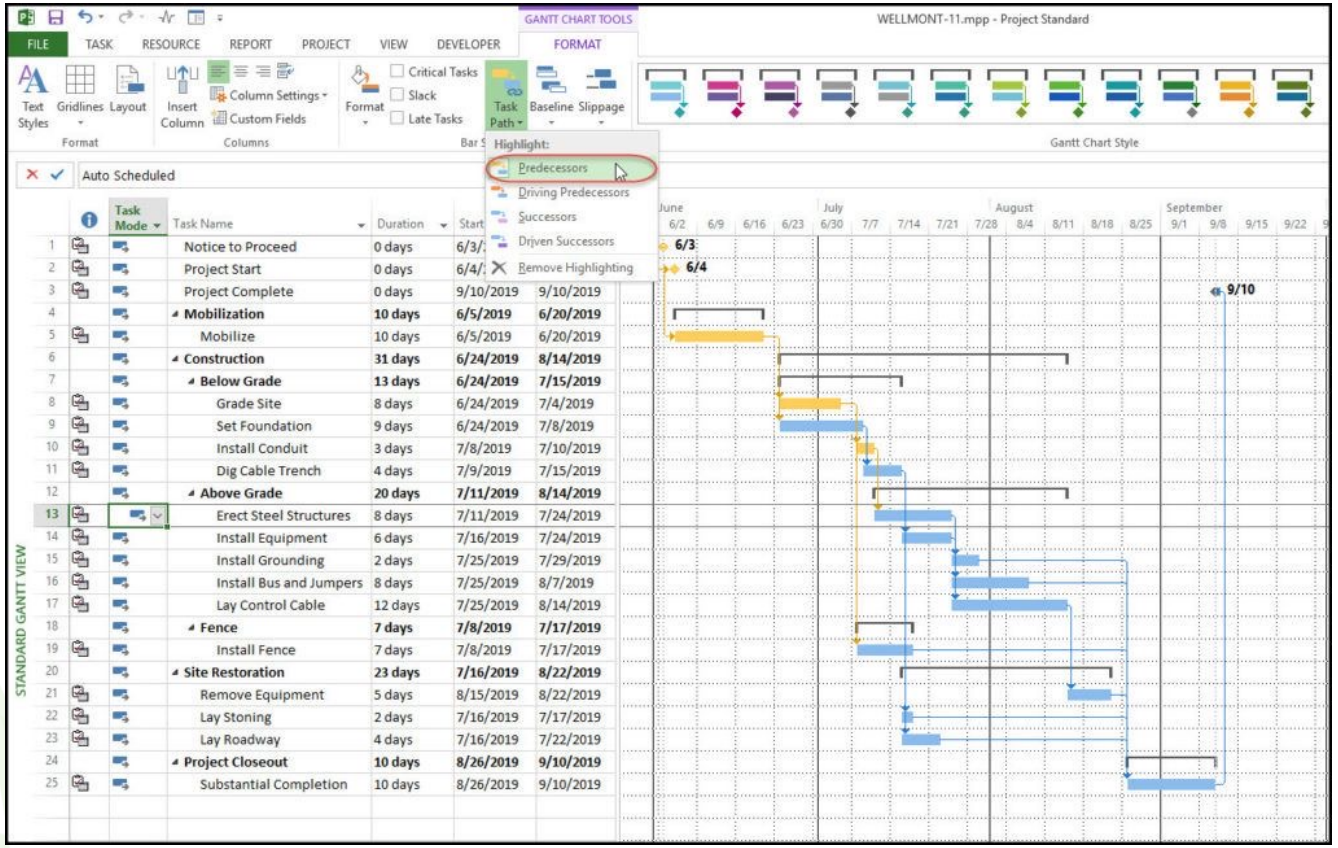
Some Applications Need Careful Planning

If a careless mistake in your requirements, coding, or verification can incinerate a family of five, casual software development approaches are simply not morally acceptable.

Video Licensed from Shutterstock by David Hetherington on 17 Feb 2020

Waterfall/Vee Weakness – The Illusion of Certainty

- It looks wonderful in Microsoft Project
- The project manager prints out 120 sheets of the grand master plan and covers an entire hallway, carefully taping them all together so that the arrows line up…
- The problem is that this sort of chained plan is an assumption built on an assumption built on an assumption built on…
- Even the simplest Markov chain analysis of the probability that any one of the activities will actually happen anywhere near the projected timeframe produces ridiculously tiny probabilities.
- Oh, and then there are “surprises”.
- The project manager’s beautiful plan is out of date before he finishes pinning it to the wall.

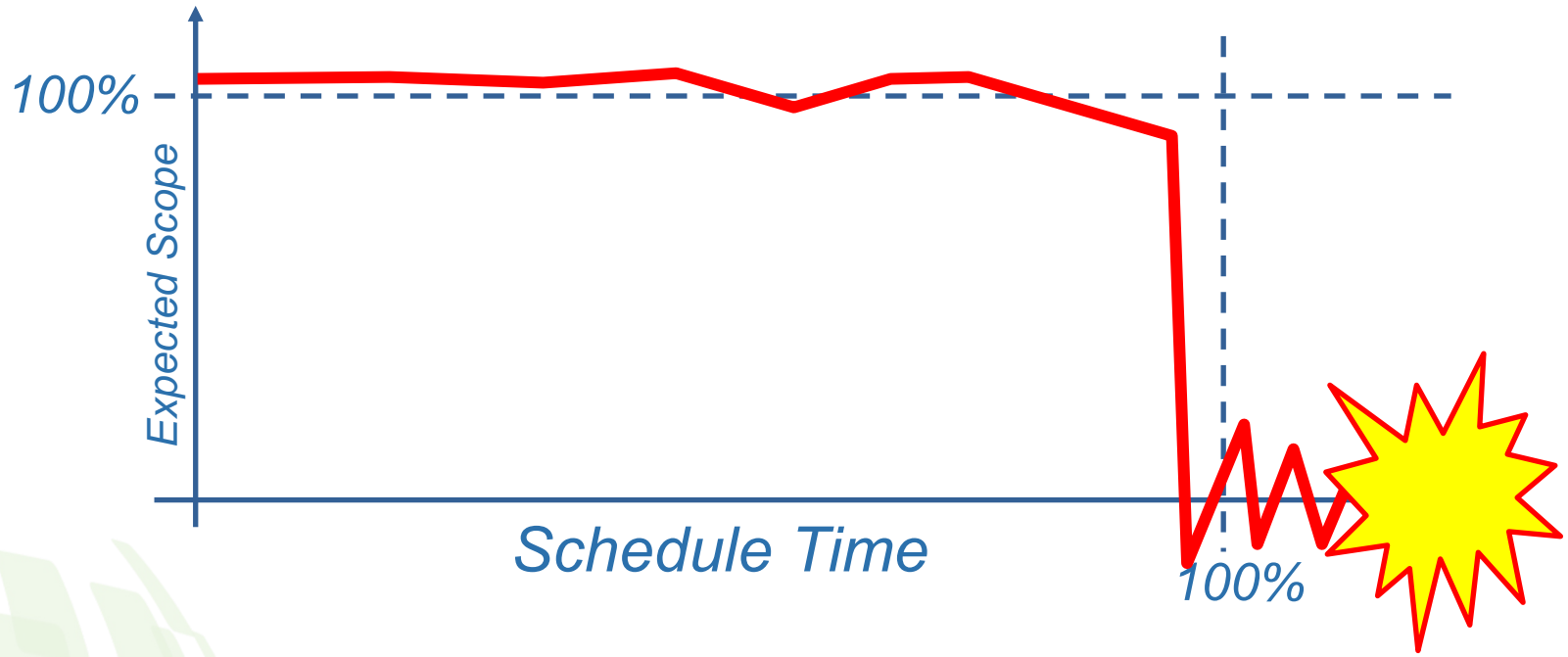


Agile Weakness – Eyeball Rolling about “Waterfall”

- This is a 20,000-ton drill ship.
- You don’t pick up the steel for it at *Home Depot.*
- It has about 10 decks.
- It gets built from the keel upward, one deck at a time.
- As each deck gets built, pieces of large equipment get craned into position.
- There is no such thing as a *“minimum viable drillship”* that can be seat-of-the-pants iterated into satisfactory product.
- This is a Waterfall/Vee project. Period.
- Having the software developers sitting around rolling their eyeballs and moaning about *“Braindead Waterfall Dinosaurs who just don’t get it”* is not helpful.

Waterfall/Vee Weakness – Scope/Risk Management

- Waterfall/Vee projects tend to have difficulty thinking about scope and risk.
- Wildly risky and unrealistic objectives – usually ignoring the reality of previously failed projects – are bought into with rousing cheers of *"This time it is going to be different!"* and *"We will overcome the problem with team spirit!"*
- This group fiction is maintained religiously by all team factions until shortly before the project deadline…
- At which time a catastrophic scope collapse occurs.



Agile Weakness – Small Team Assumption

- During World War II many bomber prototype/production lines were organized with the engineers in balcony offices above the production floor.
- Often the planes were still being designed as the first production units were moving down the line.
- Engineers could walk down the stairs and have a quick chat in front of the nose of the aircraft.
- **It was a perfect implementation of Agile's: *"Individuals and interactions over processes and tools"***
- After World War II, the aircraft became so complex that no geometry of building could house the team this way.
- Document-based systems engineering was born.
- **Agile Weakness:** Beyond a certain system and team size, you can't just *"have a cup of coffee and talk it over".*

Waterfall/Vee Weakness – Inability to Prioritize

- Waterfall/Vee projects tend to have difficulty thinking about requirements priority.
- Since the requirements are often decided through a political consensus process, it is extremely difficult for leadership to admit that any of them are anything other than “Critical Must Have” requirements.
- This phenomenon is basically weak leadership.
- Weak leadership that does not have the fortitude to explain to stakeholders that not more than 10% of the requirements can be “Critical Must Have” simply sets up the catastrophic scope failure mentioned previously.
- (Better leadership will work harder on contingency plans for all medium and higher requirements. What is plan B if we can’t have it, or it arrives too late?)
- (The Agile process makes the prioritization inherent and helps sidestep the emotional problems)

(After much gnashing of teeth, we managed to identify two requirements that we might possibly be able to imagine living without.)

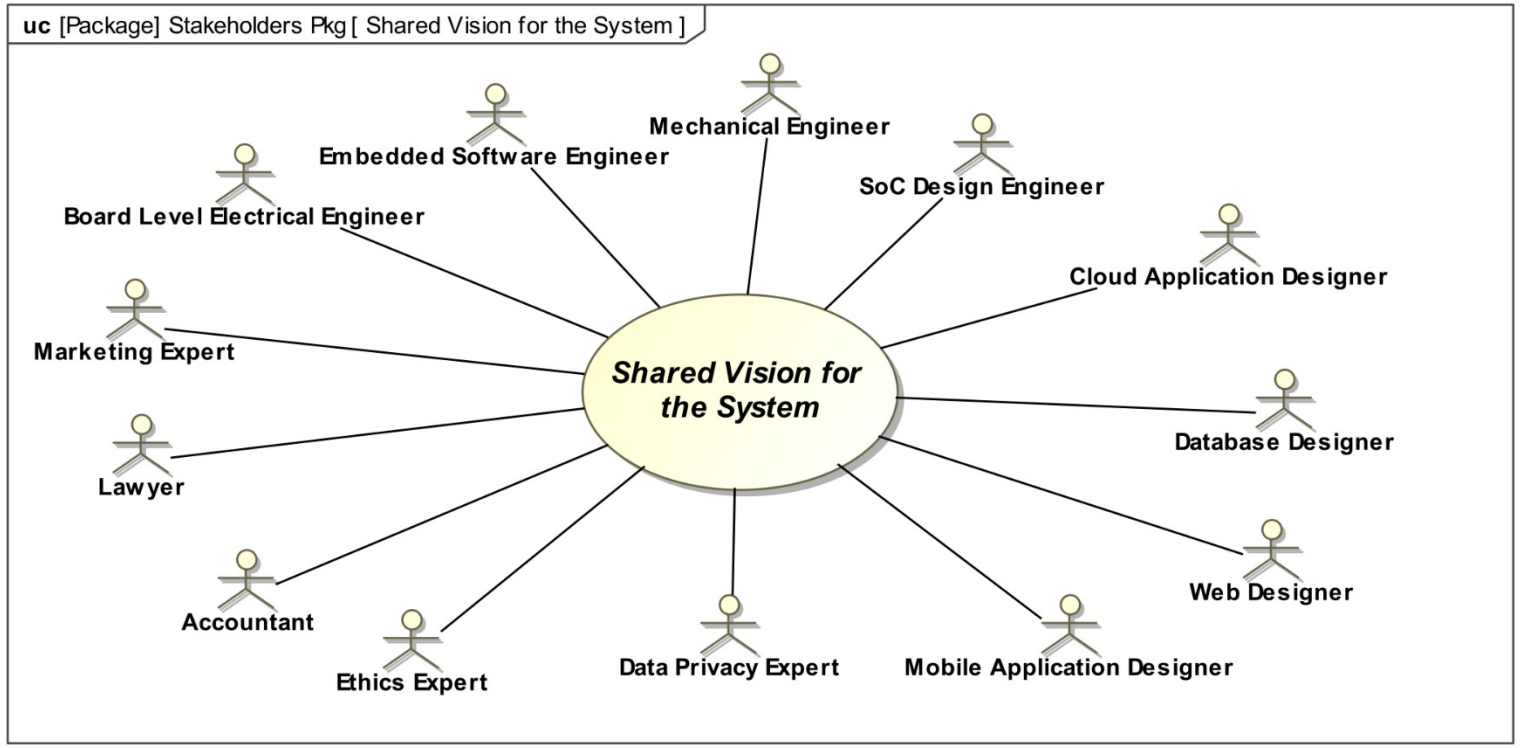
Agile Weakness – Software Belly Button Gazing

- *"The code is the documentation"*
- *"Models should be drawn on whiteboards and the whiteboards should be erased when the team leaves the room."* (Martin Fowler)

These software team slogans make a lot of sense… if software coders are the only people in the universe. Maintaining models and elaborate design documentation only for the software coders makes no sense at all. Indeed, after the initial problem analysis, the whiteboard should be erased, and the source code should be the authoritative source of the truth.

The problem with engineered systems is that the software coders are NOT the only people in the universe. It takes a large and diverse team to put an engineered product on the market – especially one that is cybersecure, complies with global rules and laws, and does not kill its users.

In this sort of environment, the software team has to understand that the source code is NOT the only critical work product. Rigorous models and solid design documentation are critical and cannot simply be ignored.

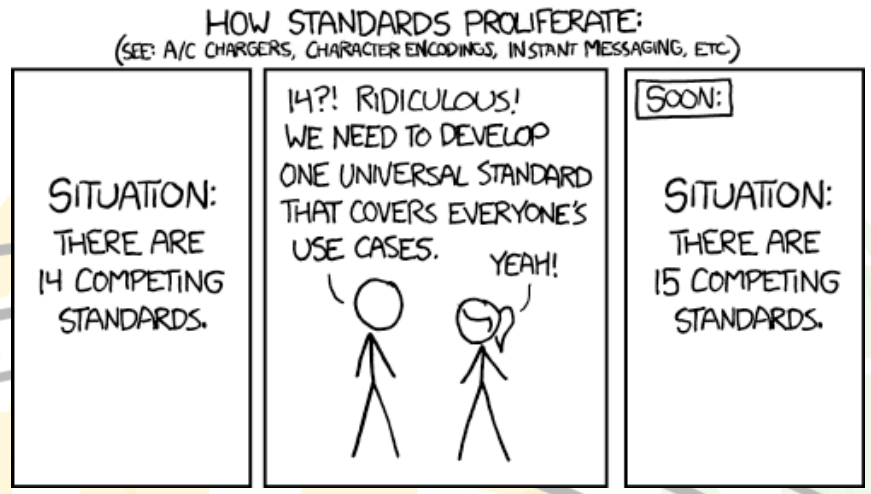


How do we bring this together to achieve project resilience, quality and accountability?

Merging Worlds

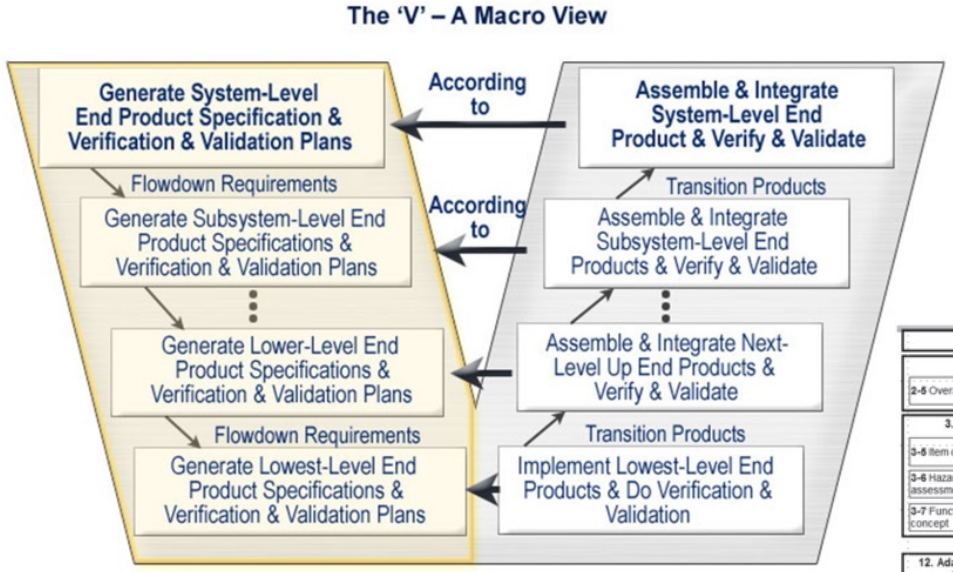
Not Another Standard!

"In this presentation we will present techniques for bringing these worlds together and achieving both project resilience, and the high levels of quality and accountability required for safety-critical and similar systems."

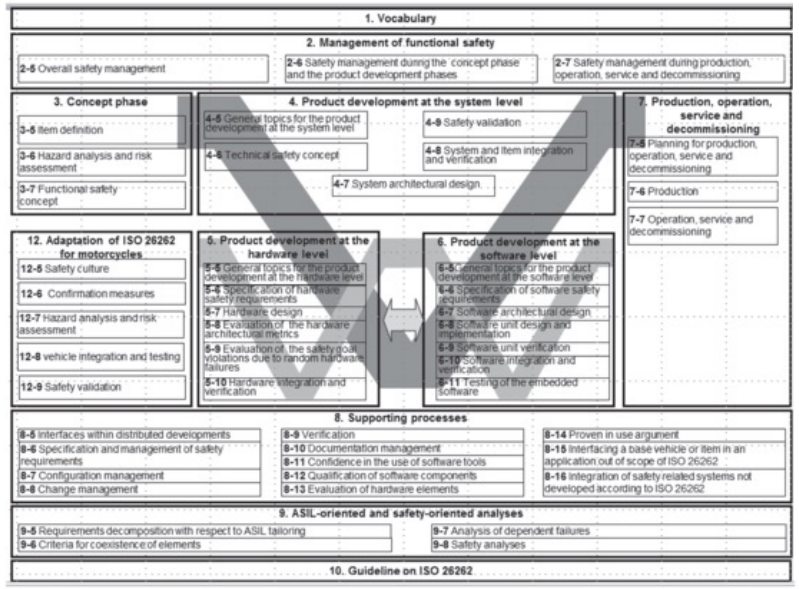


https://xkcd.com/927/

Vee Model Processes



https://sebokwiki.org/wiki/System_Lifecycle_Process_Models:_Vee



https://www.iso.org/standard/43464.html

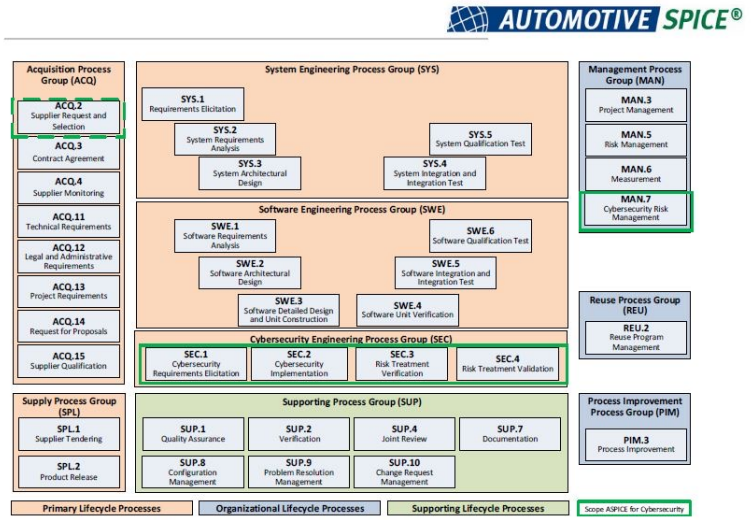
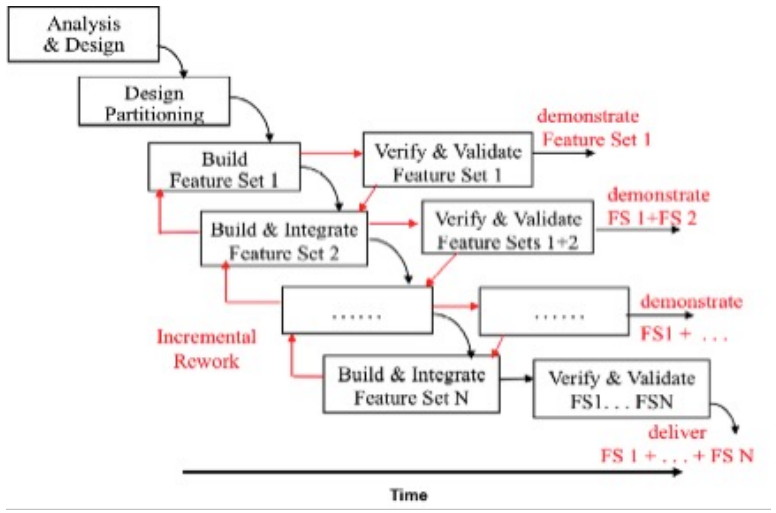
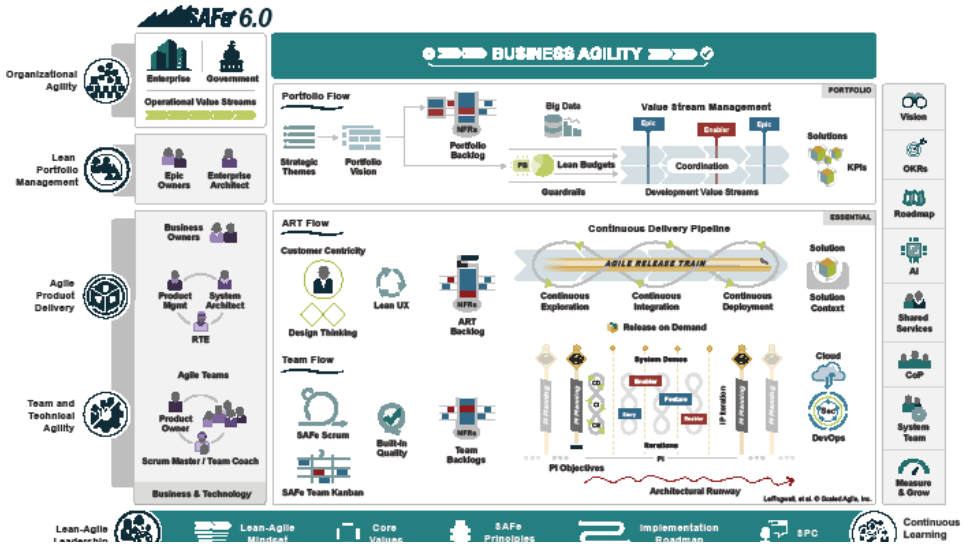


Figure 2 — Automotive SPICE and Automotive SPICE for Cybersecurity Process Reference Model – Overview

https://www.automotivespice.com/fileadmin/software-download/AutomotiveSPICE_for_Cybersecurity.pdf

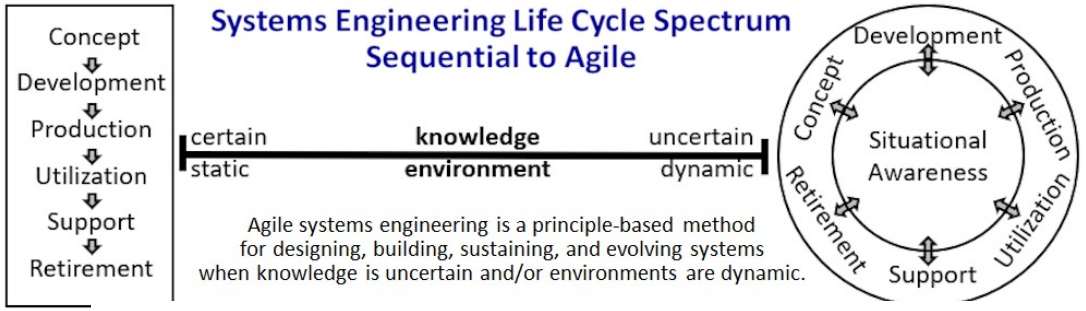
Agile Processes



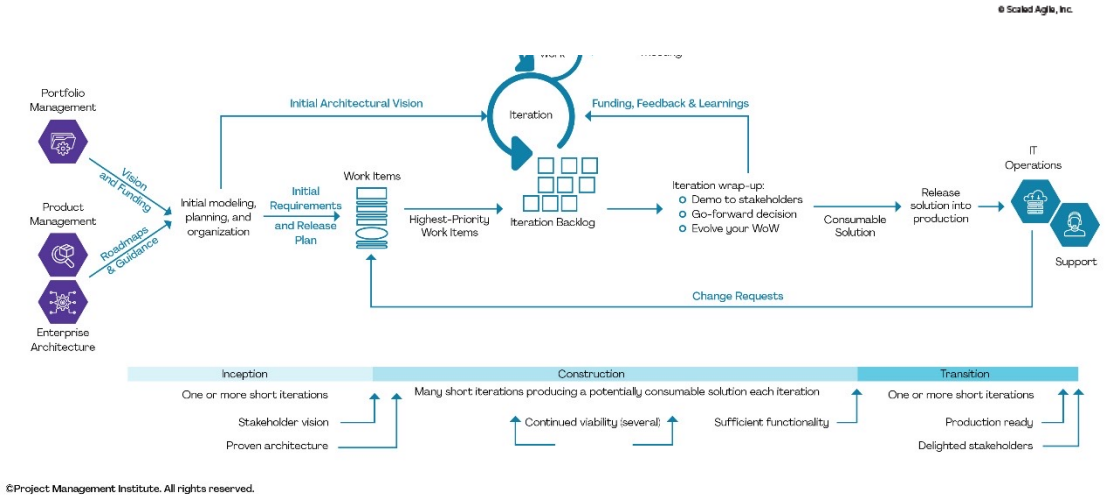


https://scaledagileframework.com/wp-content/uploads/delightful-downloads/2023/03/Portfolio_SAFe_6.0_US_Letter.pdf

https://sebokwiki.org/wiki/System_Lifecycle_Process_Models:_Incremental



https://www.incose.org/incose-member-resources/working-groups/transformational/agile-systems-se



https://www.pmi.org/disciplined-agile/lifecycle/agile-lifecycle

Product Development Lifecycles

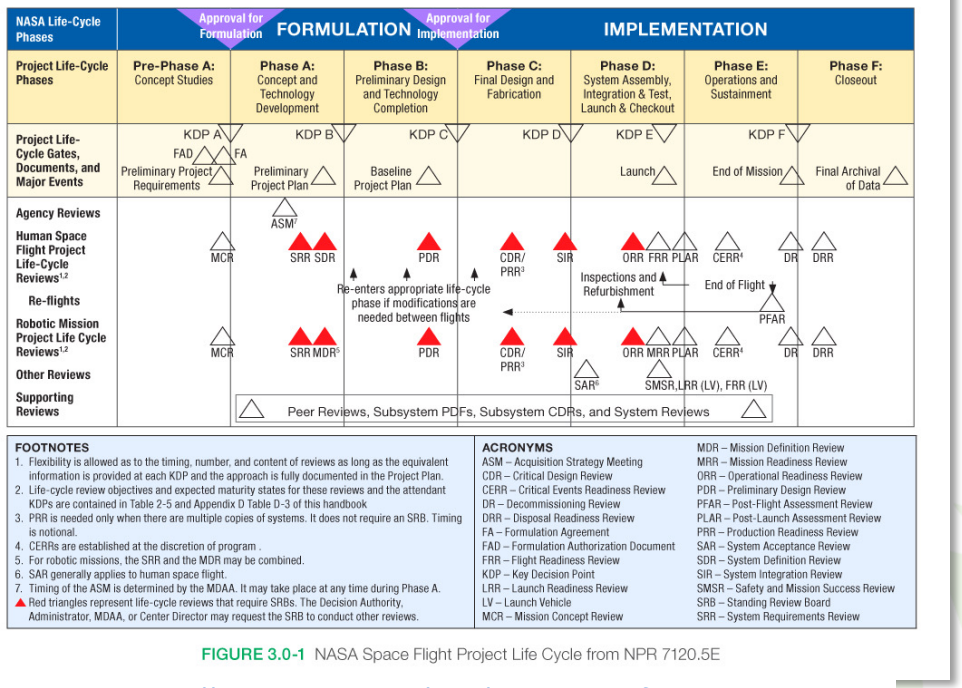
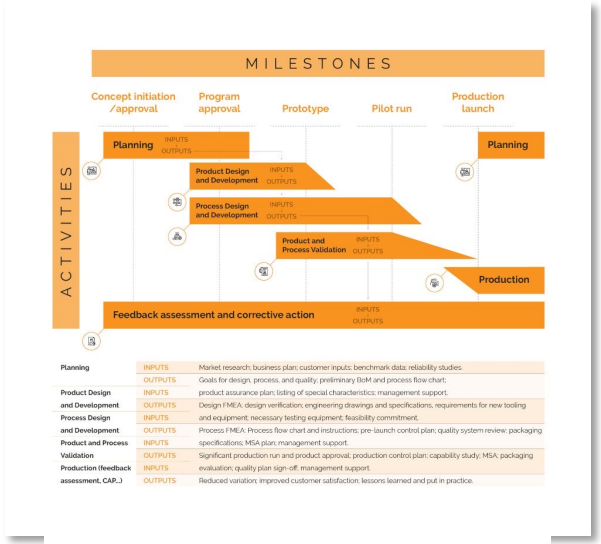
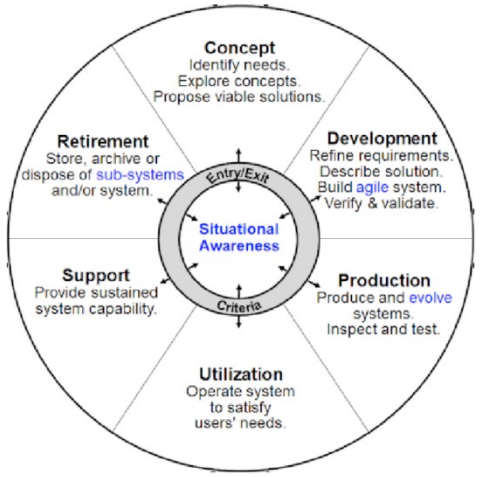


FIGURE 3.0-1 NASA Space Flight Project Life Cycle from NPR 7120.5E

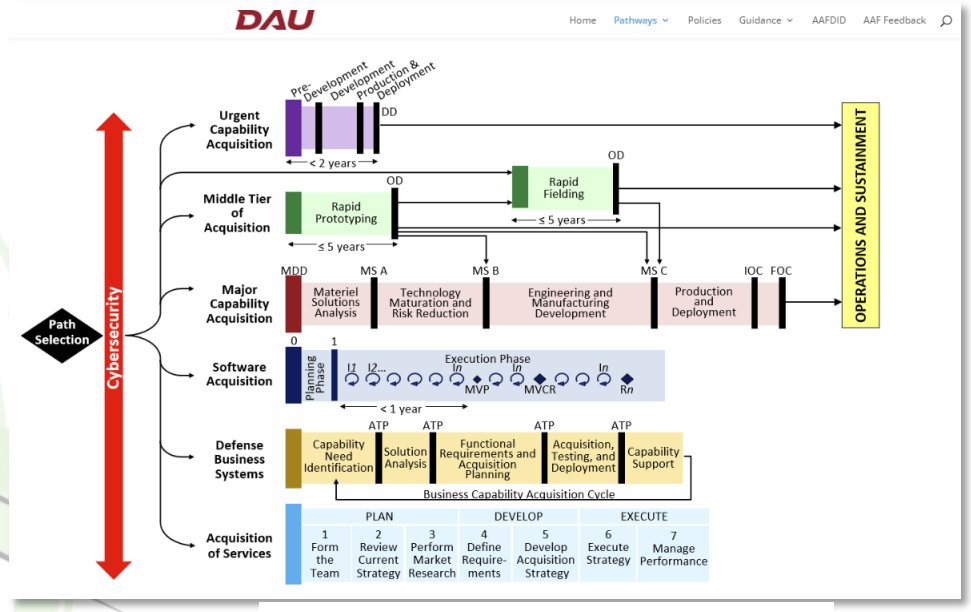
https://www.nasa.gov/seh/3-project-life-cycle



https://qualityinspection.org/advanced-product-quality-planning/



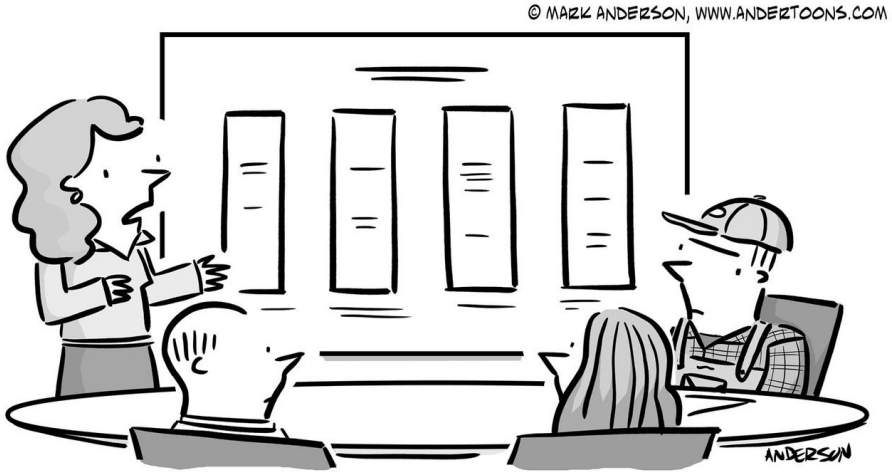
https://sebokwiki.org/wiki/System_Life_Cycle_Process_Models:_Agile_Systems_Engineering



https://aaf.dau.edu/aaf/aaf-pathways/

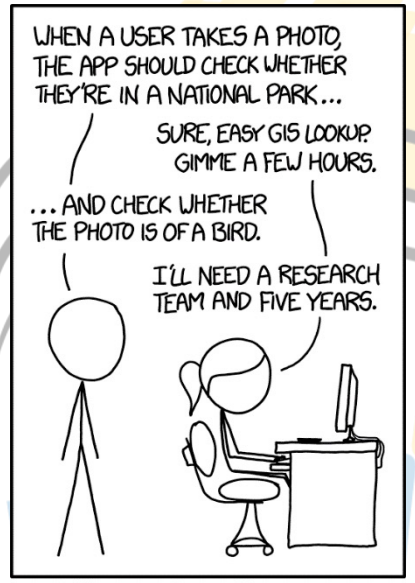
Team and Organizational Structures

https://giphy.com/explore/eight-bosses



"I know not all of us are excited about the prospect of breaking down silos..."

https://andertoons.com/silo/cartoon/7431/i-know-not-all-of-us-are-excited-about-breaking-down-silos



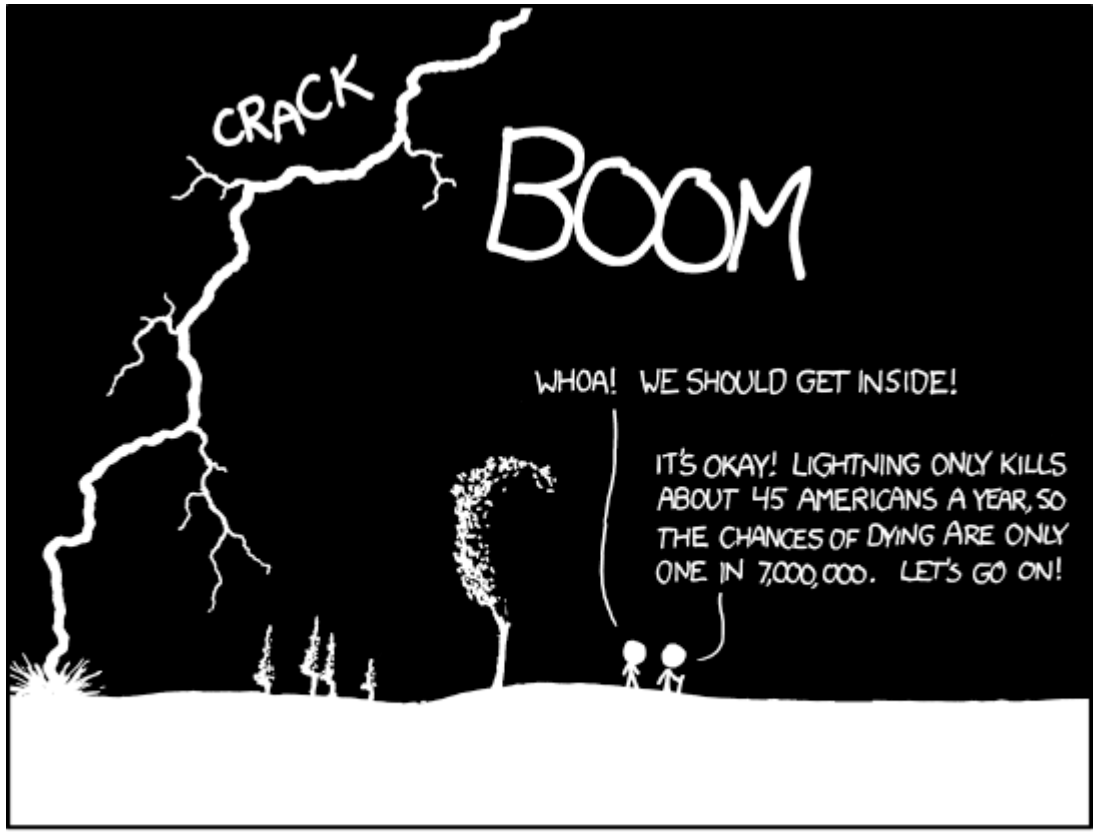
https://xkcd.com/1425/

Cyber Security, Safety and Quality

It's all about Risk Management!

What do we need to do?

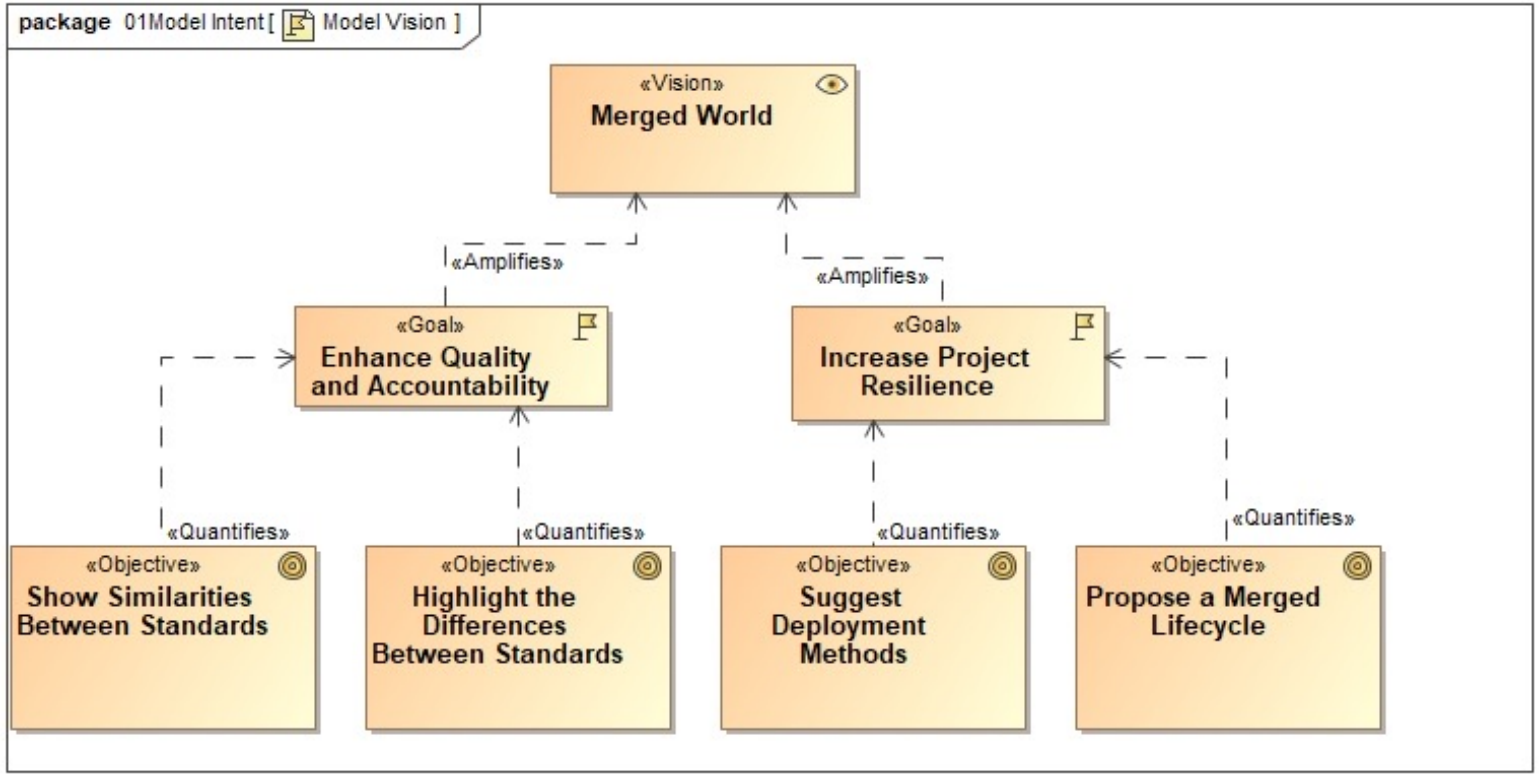
- Identify, categorize, and implement mitigation
- Contain Risk
- Recover from Risk Realization
- Control Risk
- Monitor Risk
- Top Leadership needs data and evidence to be able to accept the risk and own the realizations.



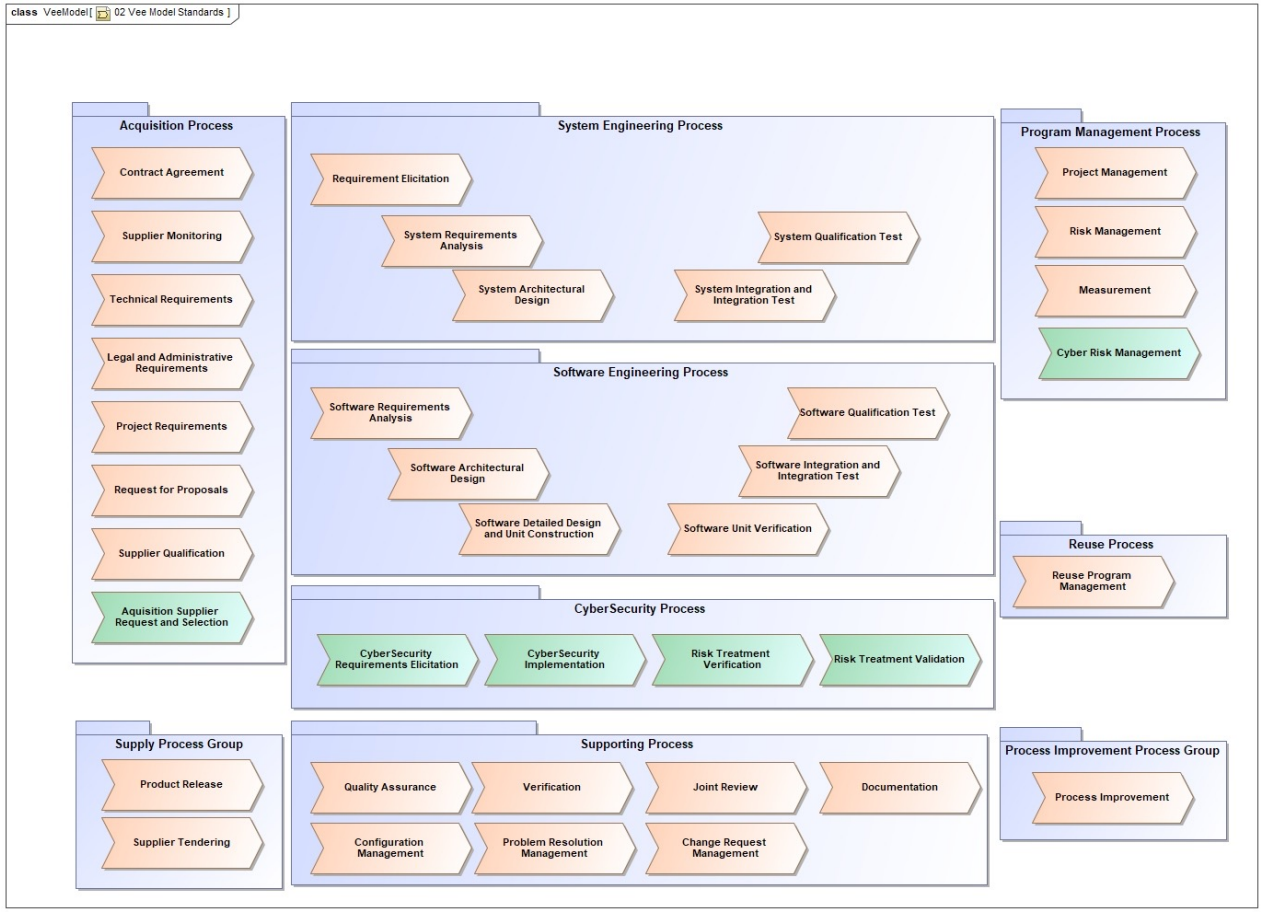
https://xkcd.com/795/

The Analysis Method: BPMN Model

- Process Areas, Objectives, and Activities
- Evidence and Data Types
- Team Structure
- Lifecycles



Modeling the Processes – Vee Model



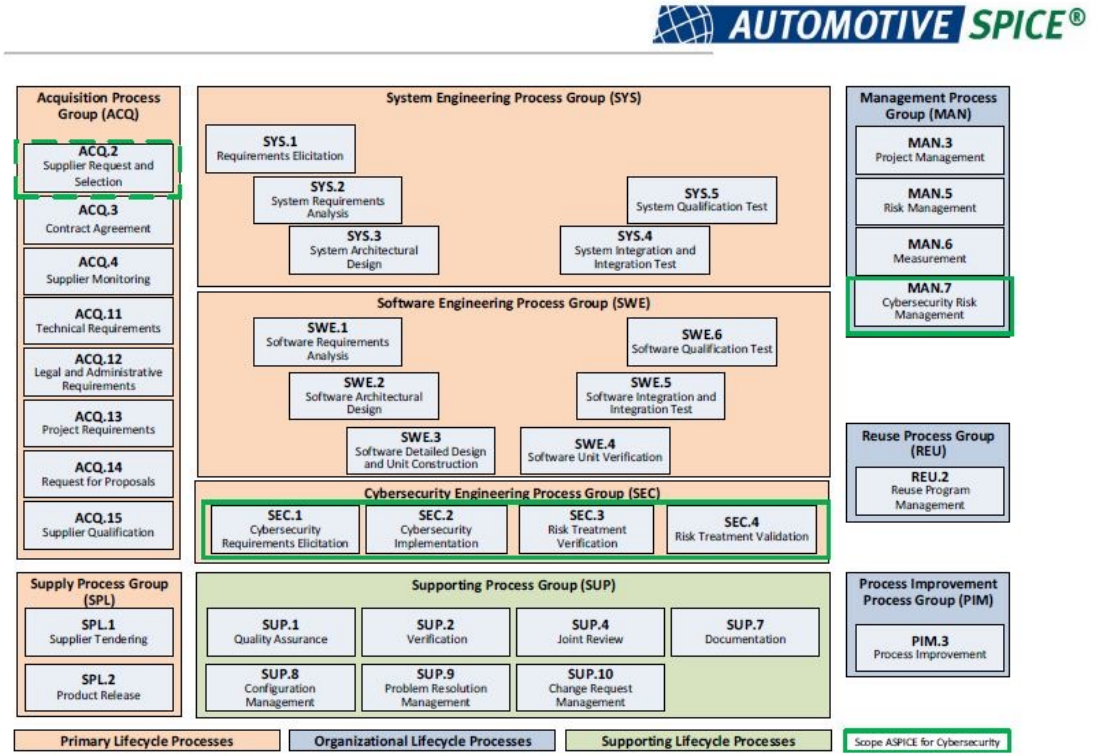
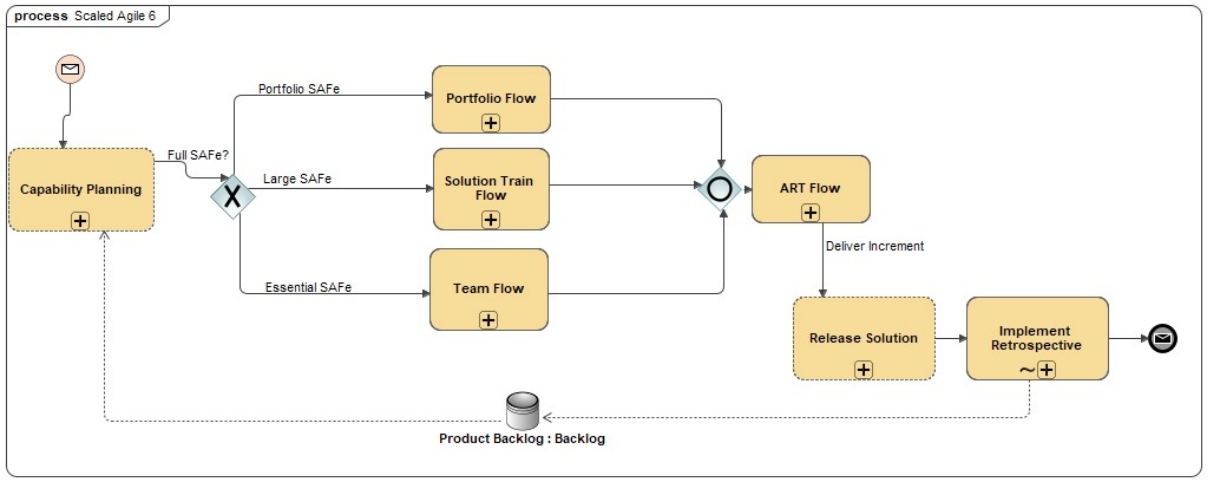
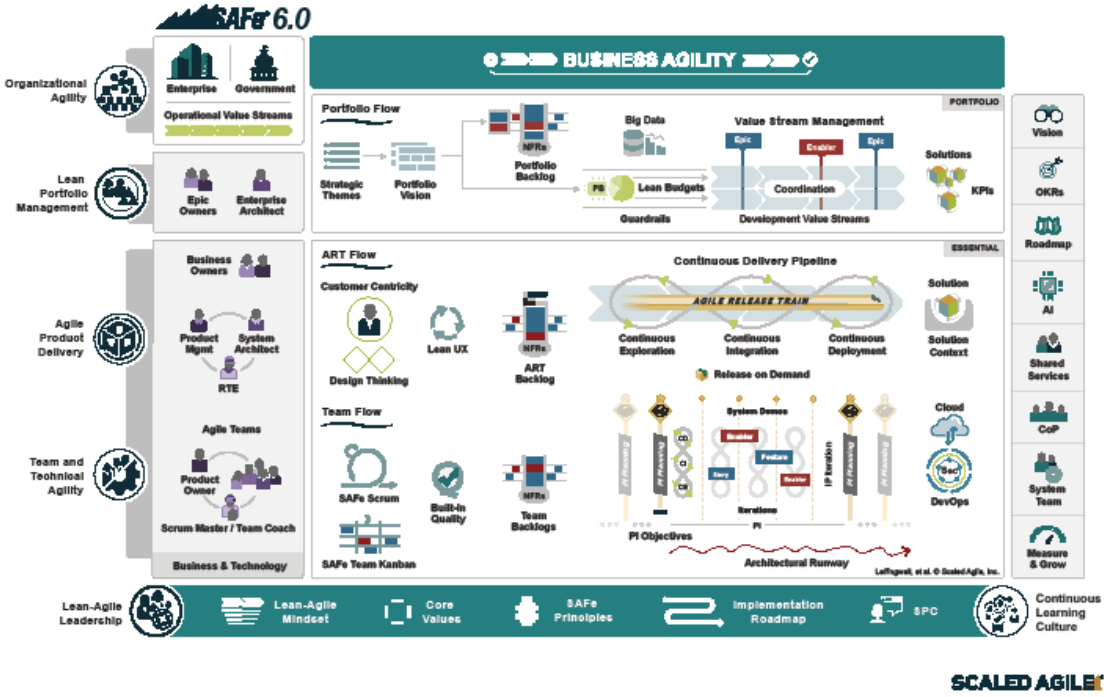


Figure 2 — Automotive SPICE and Automotive SPICE for Cybersecurity Process Reference Model – Overview

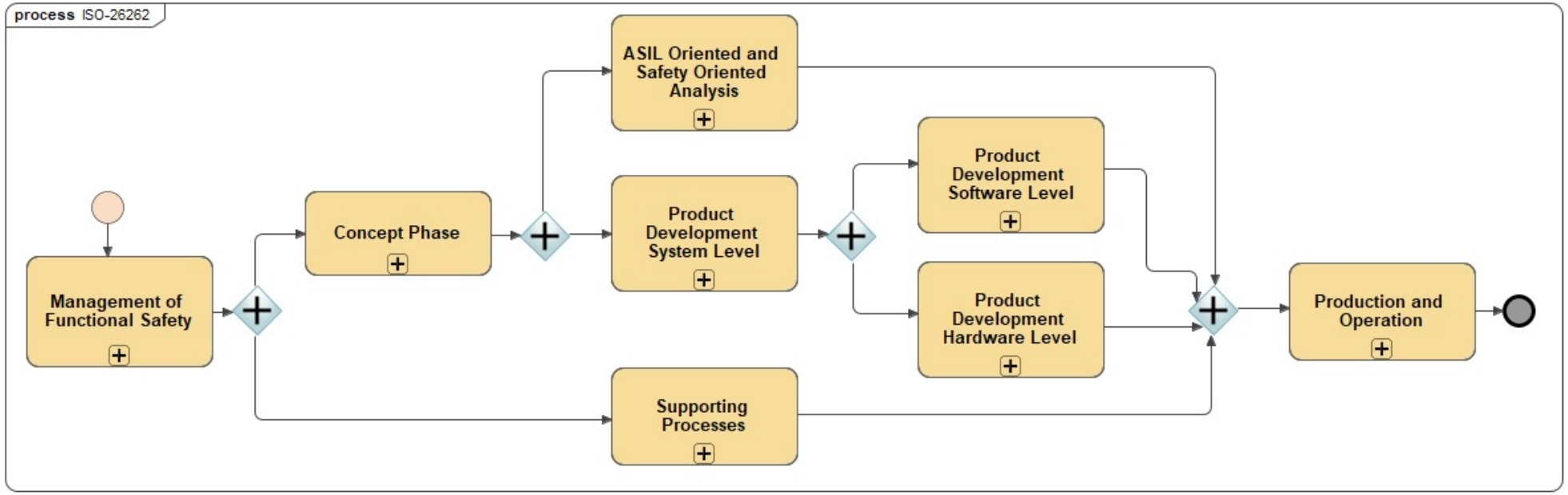
Modeling the Processes – Agile

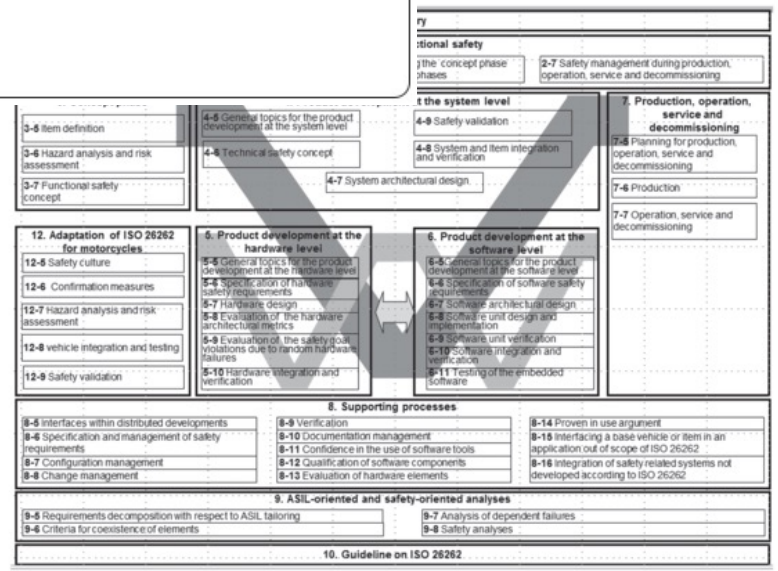




https://scaledagileframework.com/wp-content/uploads/delightful-downloads/2023/03/Portfolio_SAFe_6.0_US_Letter.pdf

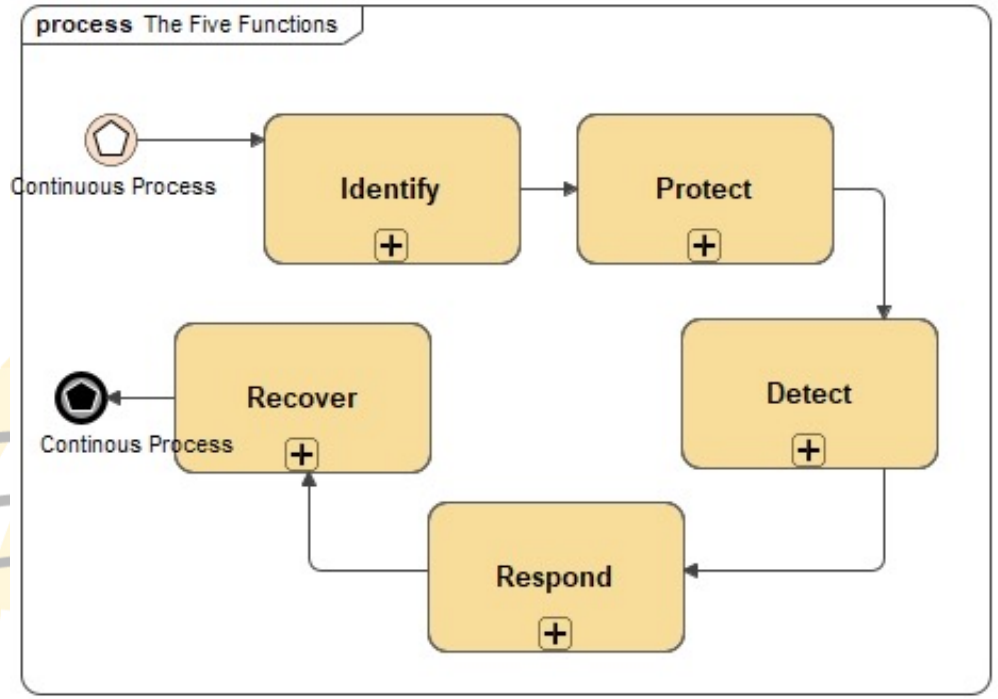
Modeling the Supporting Process: Safety ISO-26262

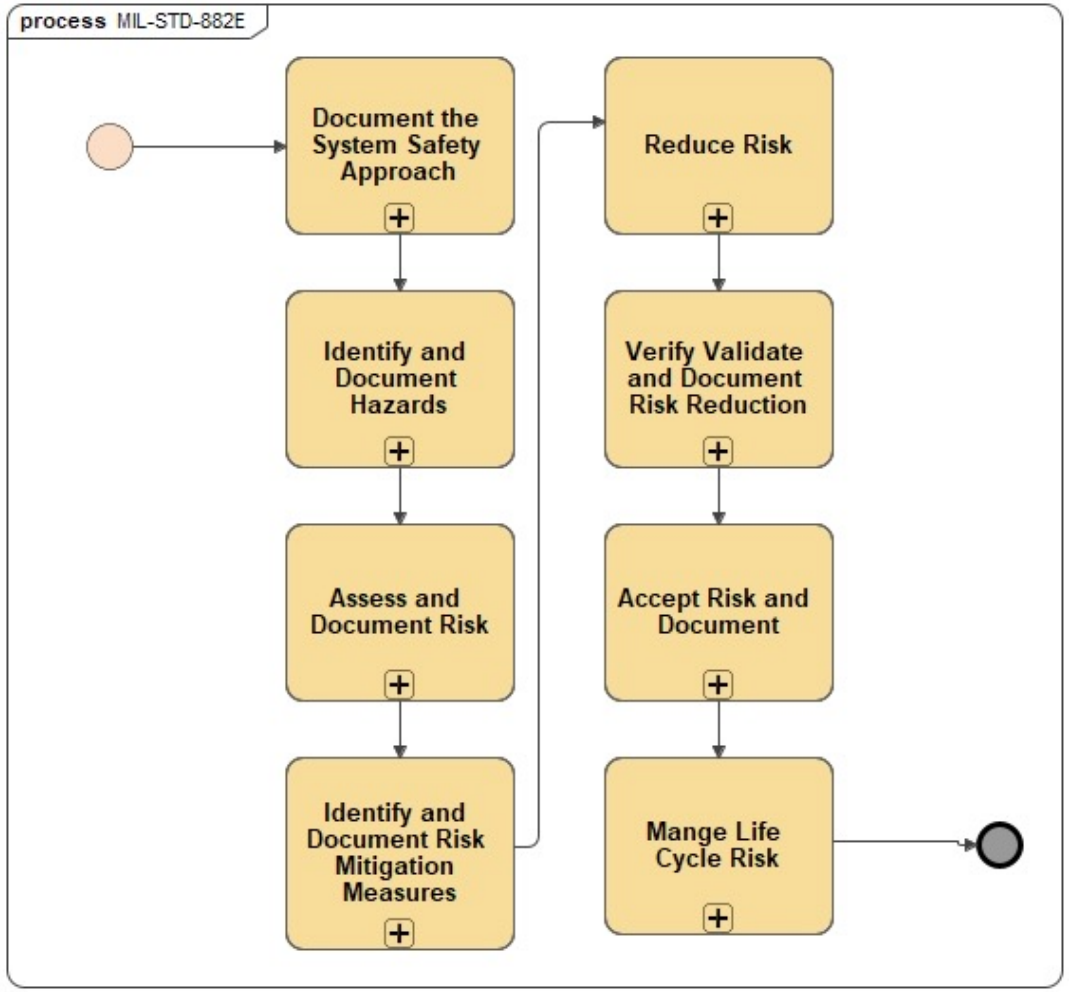




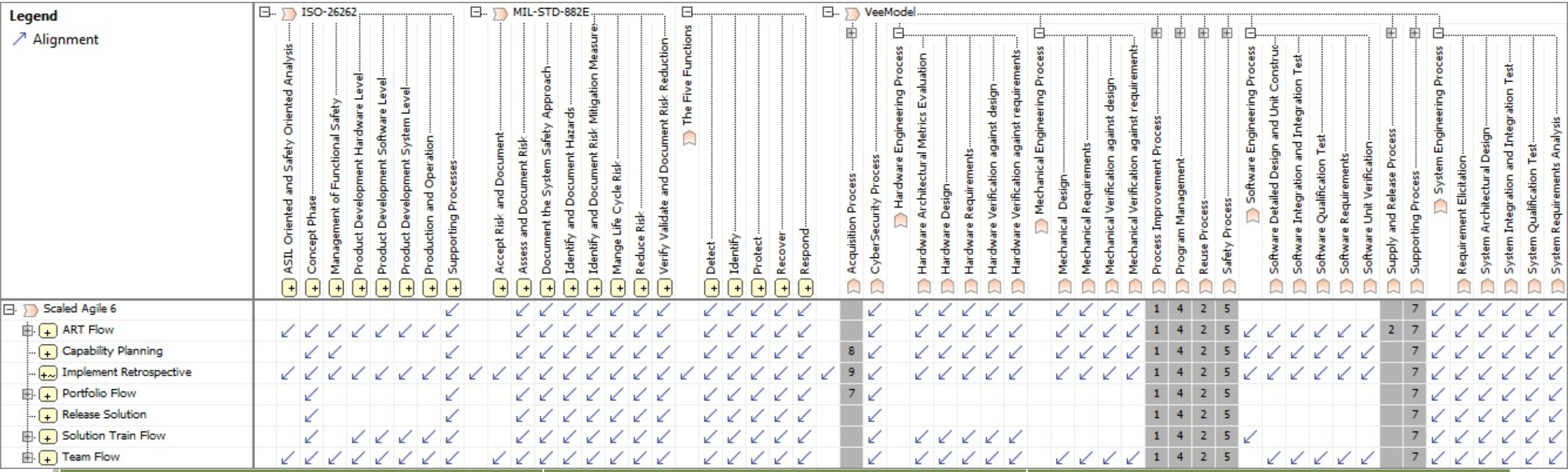
https://www.iso.org/standard/43464.html

Modeling the Supporting Process: NIST, MIL-STD-882E



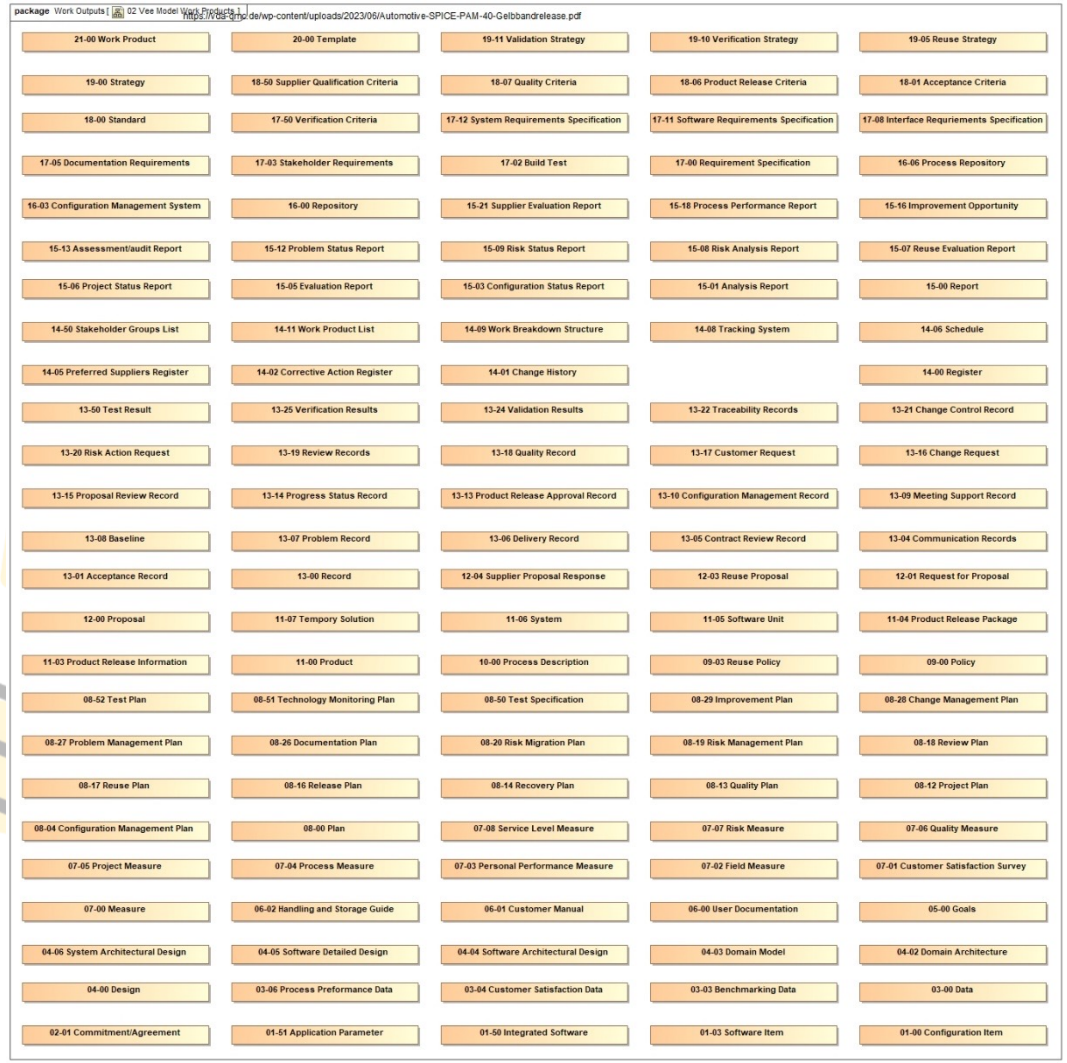


Comparing Processes

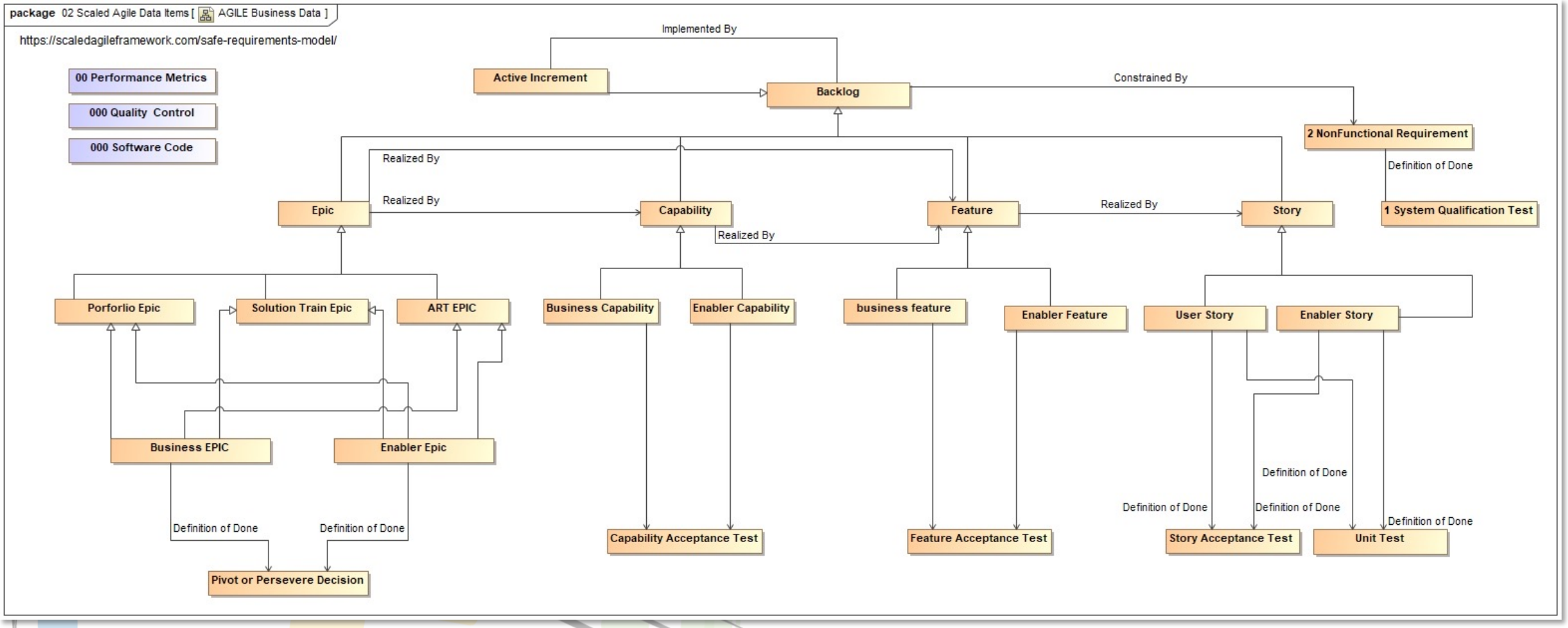


Similarities	Differences	Narrative
Same objectives, context and activities	Names of activities, sequencing, roles included, and definition of done.	In Agile it covers multiple process areas in a modified parallel workflow. In Vee Model each process is seen as a workflow state and required complete to transition.

Modeling the Evidence: Vee Model

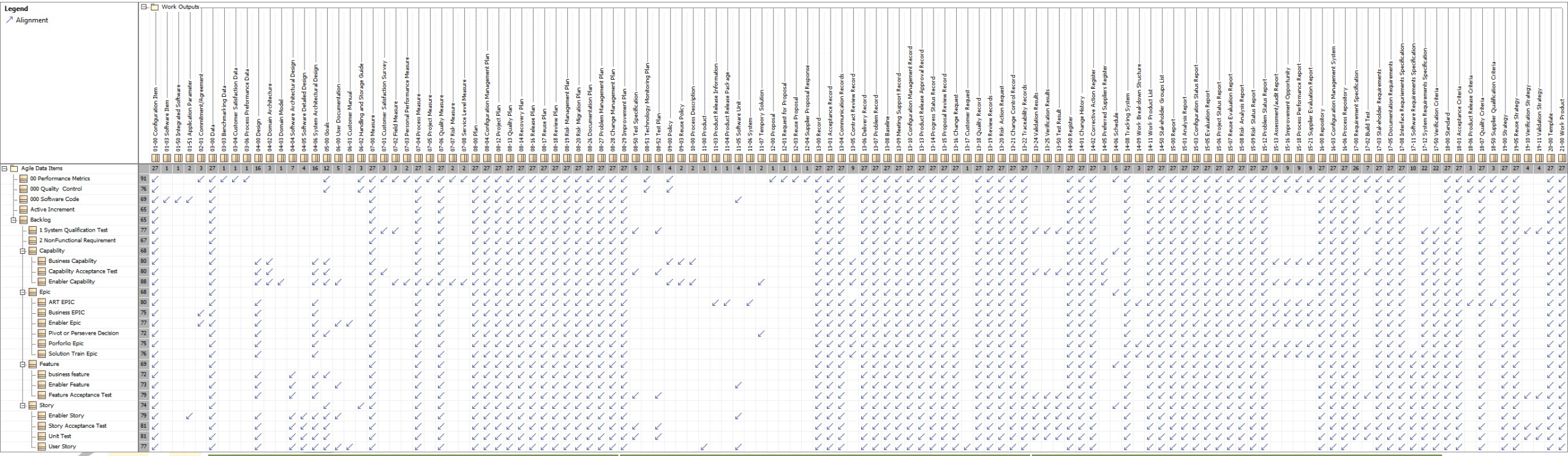


Modeling the Evidence: Agile



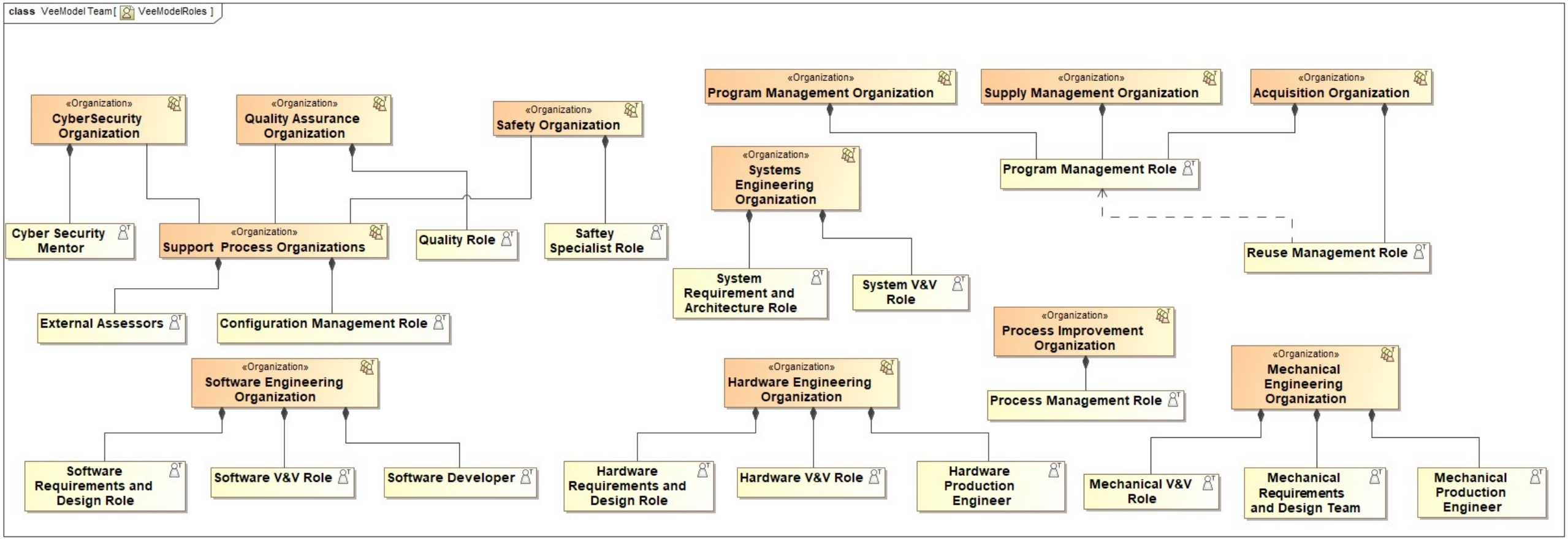
https://www.incose.org/incose-member-resources/working-groups/transformational/agile-systems-se

Comparing Evidence

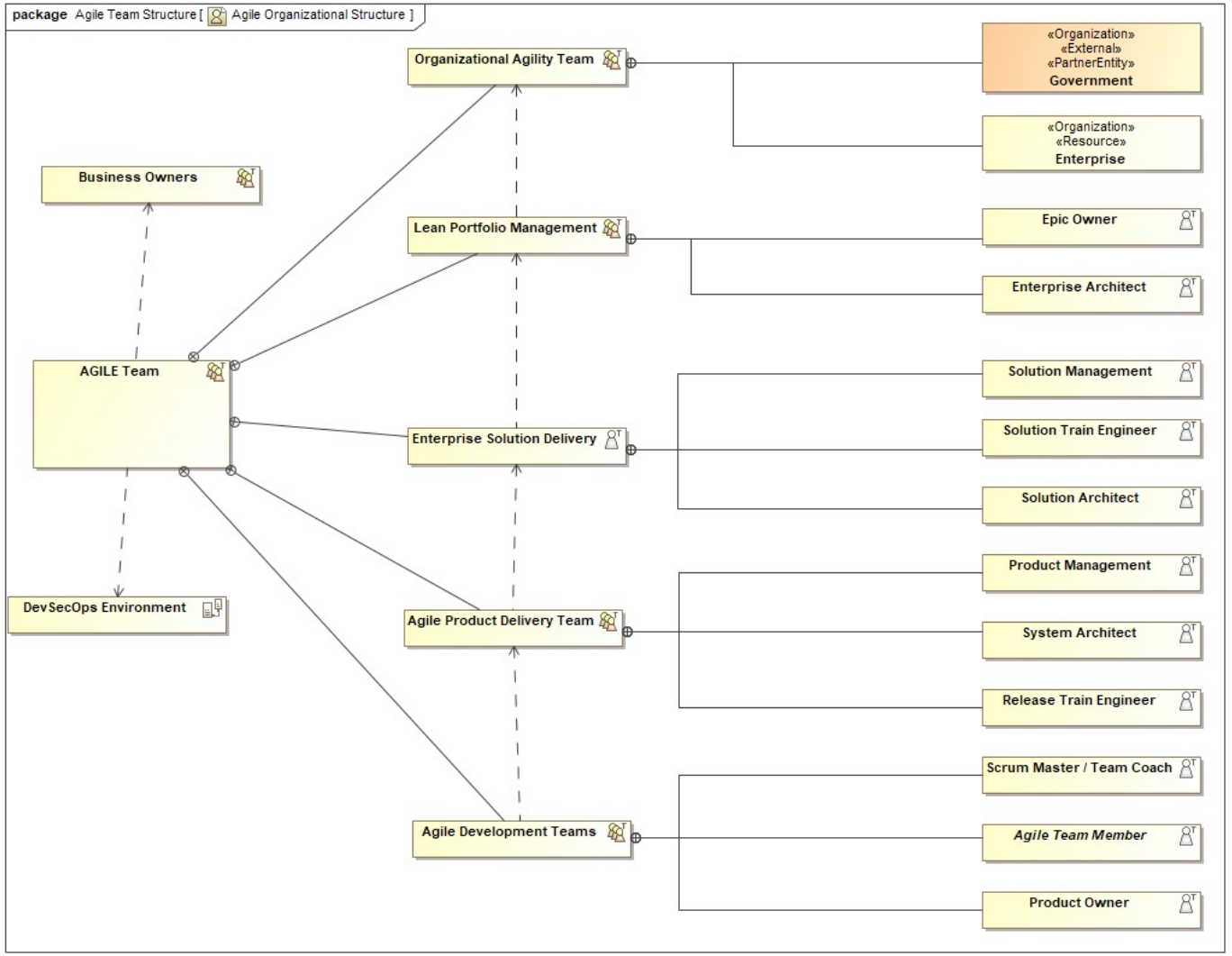


Similarities	Differences	Narrative
Same objectives, context and required evidence.	When it is complete.	In Agile evidence is complete when the capability is complete. In Vee Model the evidence is complete when the artifact is complete.

Modeling the Team Structure: Vee Model

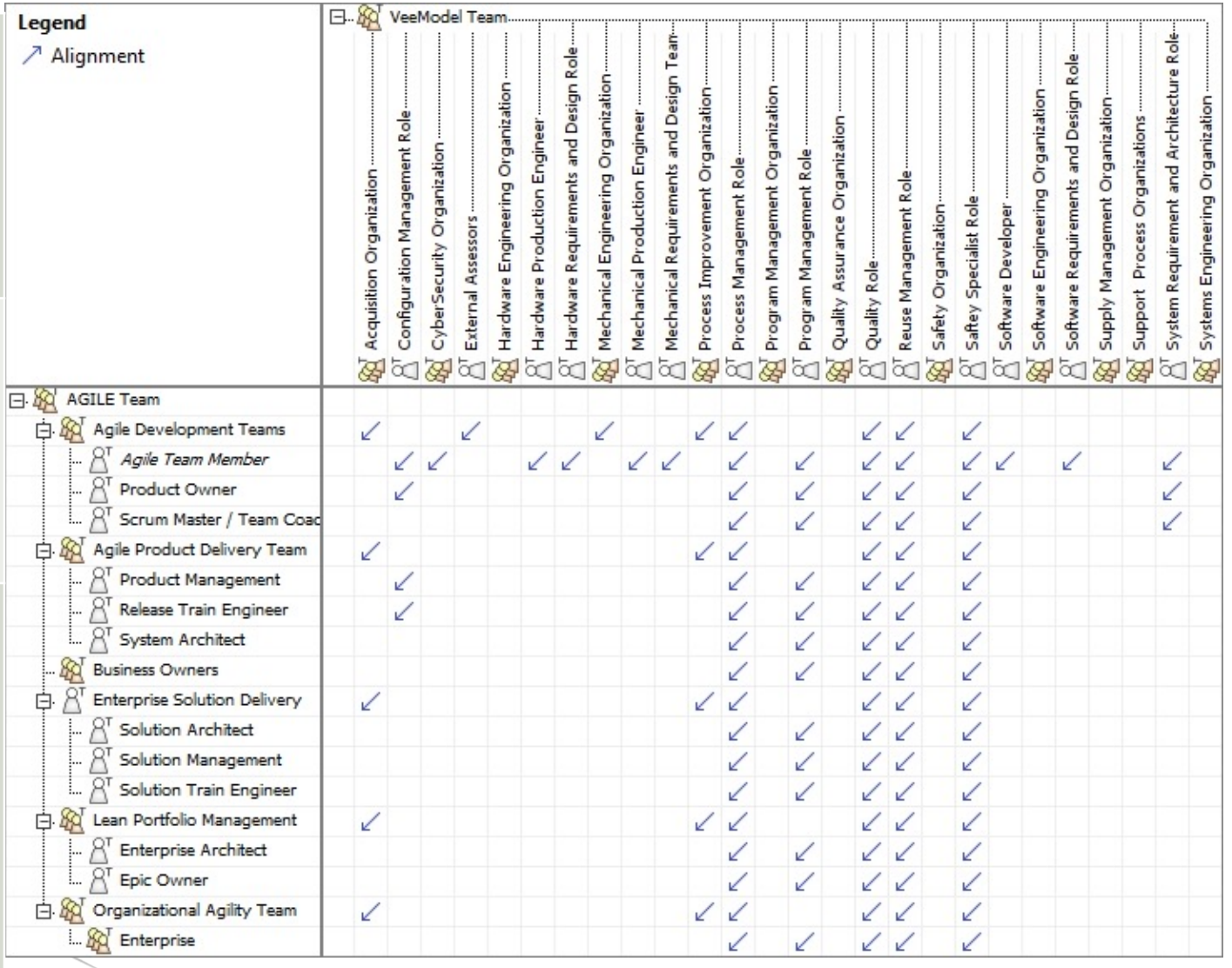


Modeling the Team Structure: Agile

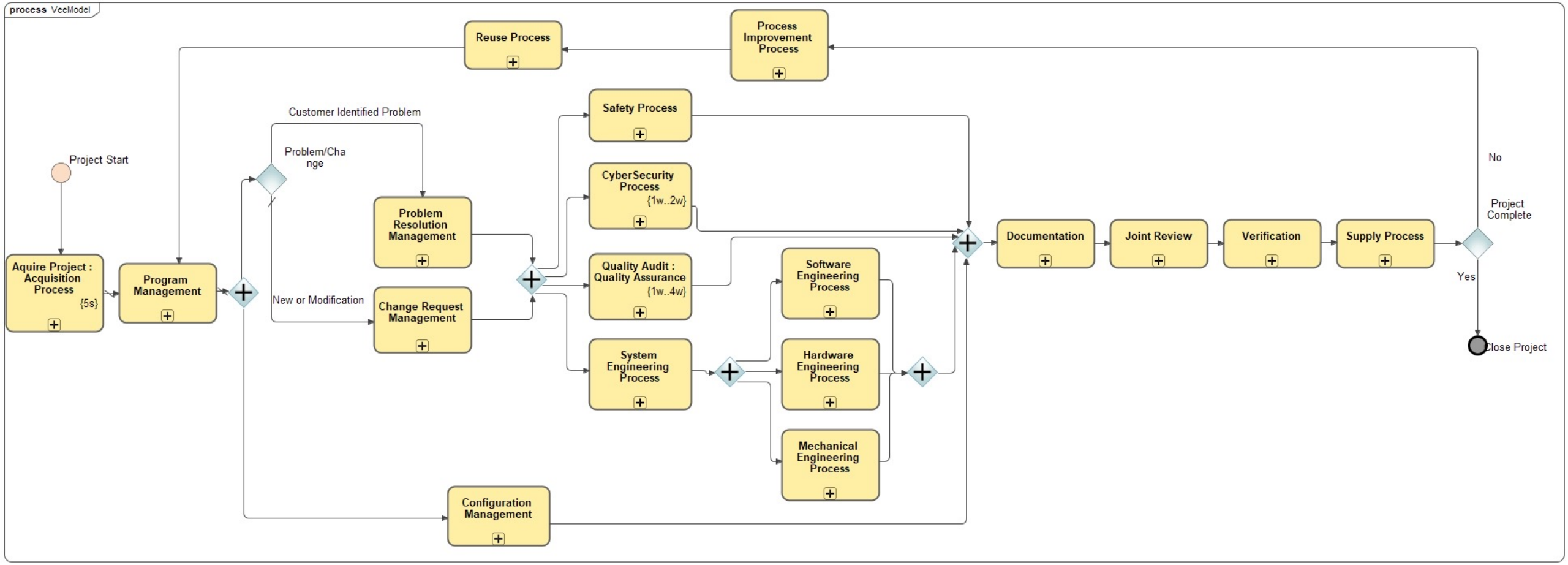


Comparing Team Structures

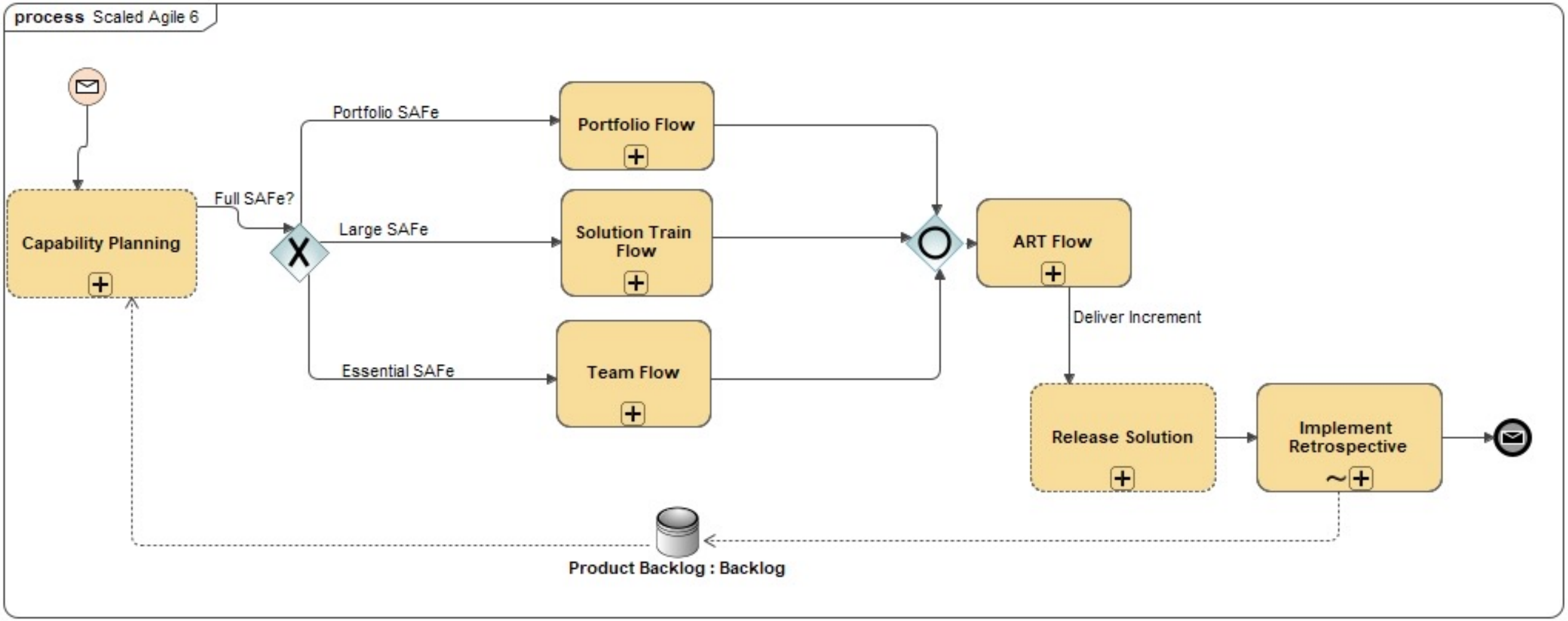
Similarities	Same objectives, context and required skills.
Differences	When to perform which roles and when to collaborate between roles.
Narrative	In Agile, Roles are more hats you wear when you have the required skills to collaborate on the solution. In Vee Model, Roles are your position in the hierarchy of the project organizational structure and Silo'd away from the project team.



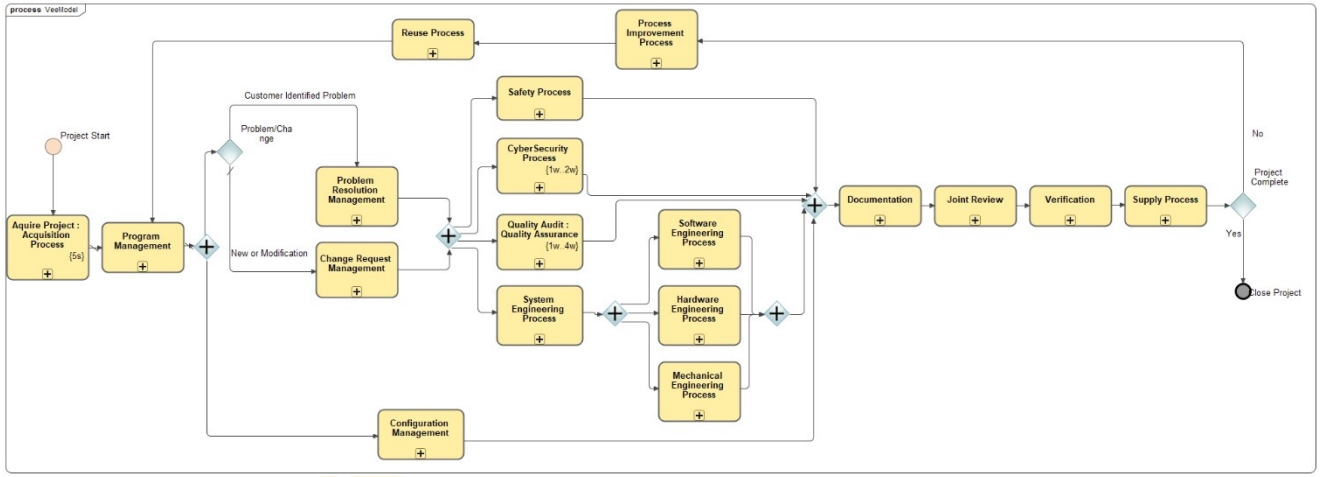
Modeling Lifecycles: Vee Model

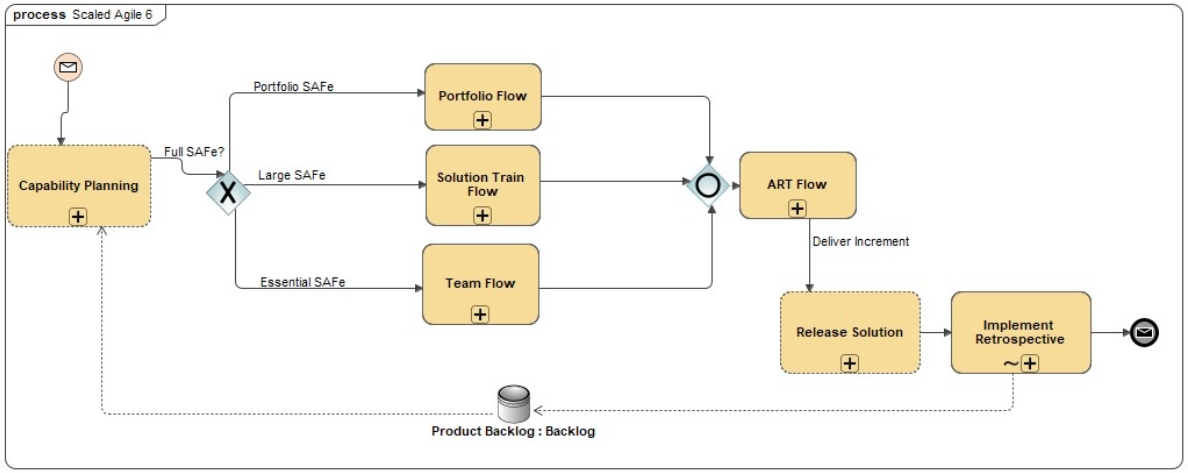


Modeling Lifecycles: Agile



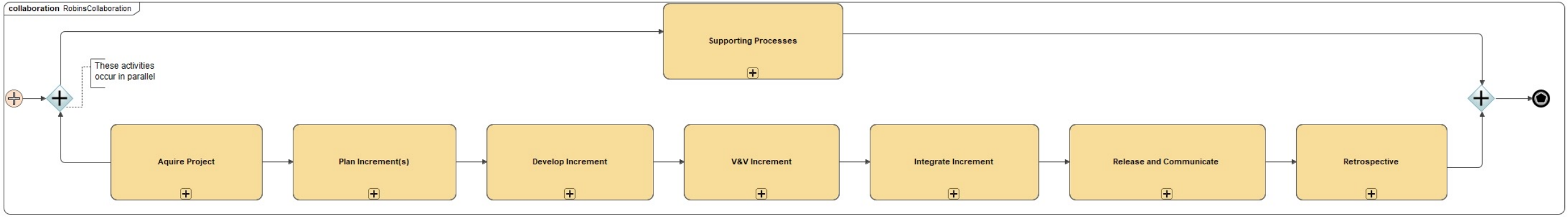
Comparing Lifecycles

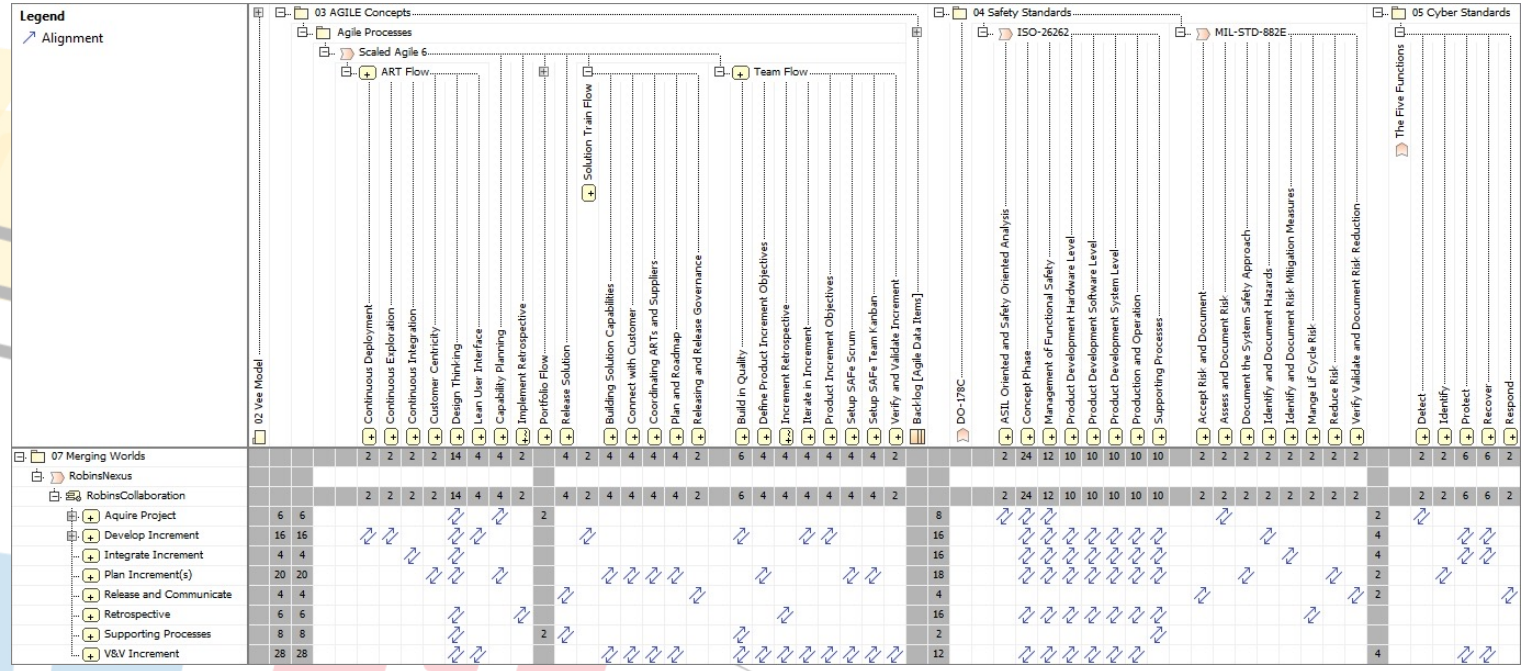




Similarities	Differences	Narrative
Both lifecycles can meet the process and evidence requirements and objectives.	Agile covers process areas in a workflow for a solution. Vee Model covers solutions by silo'd process areas.	Agile is collaborative across silos by definition of the team. Solutions are found together. Vee Model requires continuous hand offs between separated process areas relying on checkpoints or phase gates for integration while forcing silo'd structures and development.
Both could be benefited by the other.	Capability vs Process Focus. Wait Times, Lean Value Stream	Agile aligns all activities at the beginning and end of each increment and integration event. Vee Model aligns at phase gates which can be months apart.

Robins Example Simplified Merged Lifecycle Model





Why is knowing this important?

Conclusions

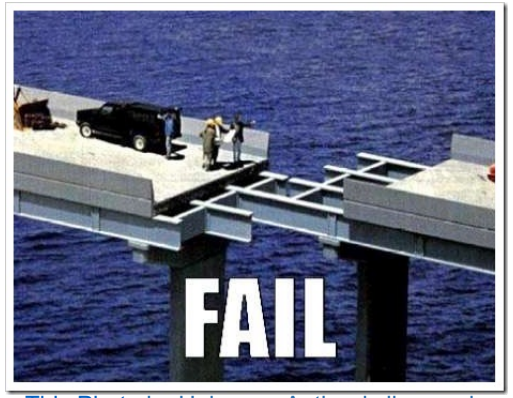
Why?

Techniques for bringing these two worlds together and achieving both

i. Project resilience

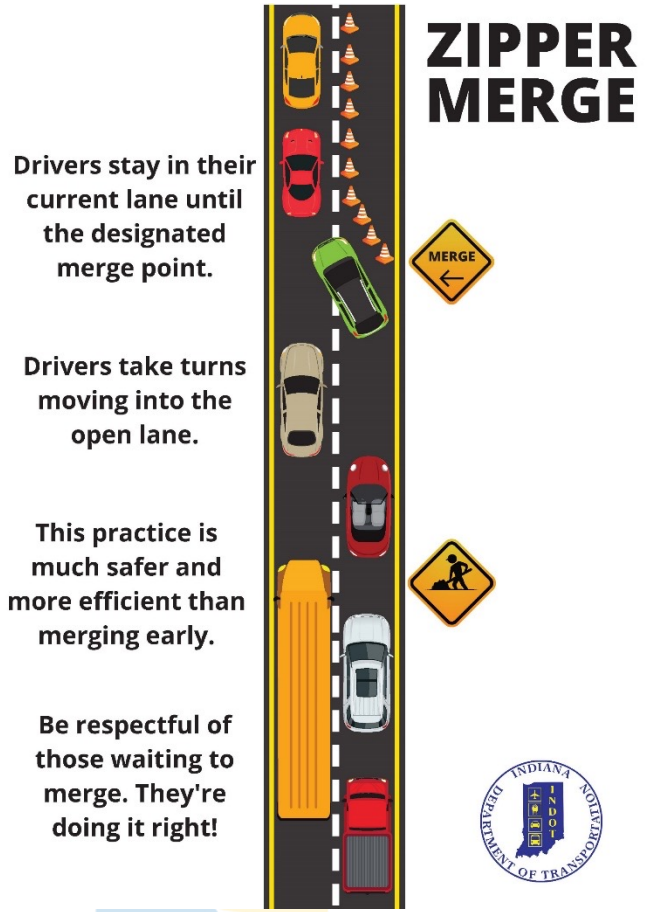
ii. The high levels of quality and accountability required for safety-critical and similar systems.

https://www.reddit.com/r/ProgrammerHumor/comments/u5pyji/incoming_flood_of_merge_conflicts/



https://cheezburger.com/3216448768/planning

Conclusions



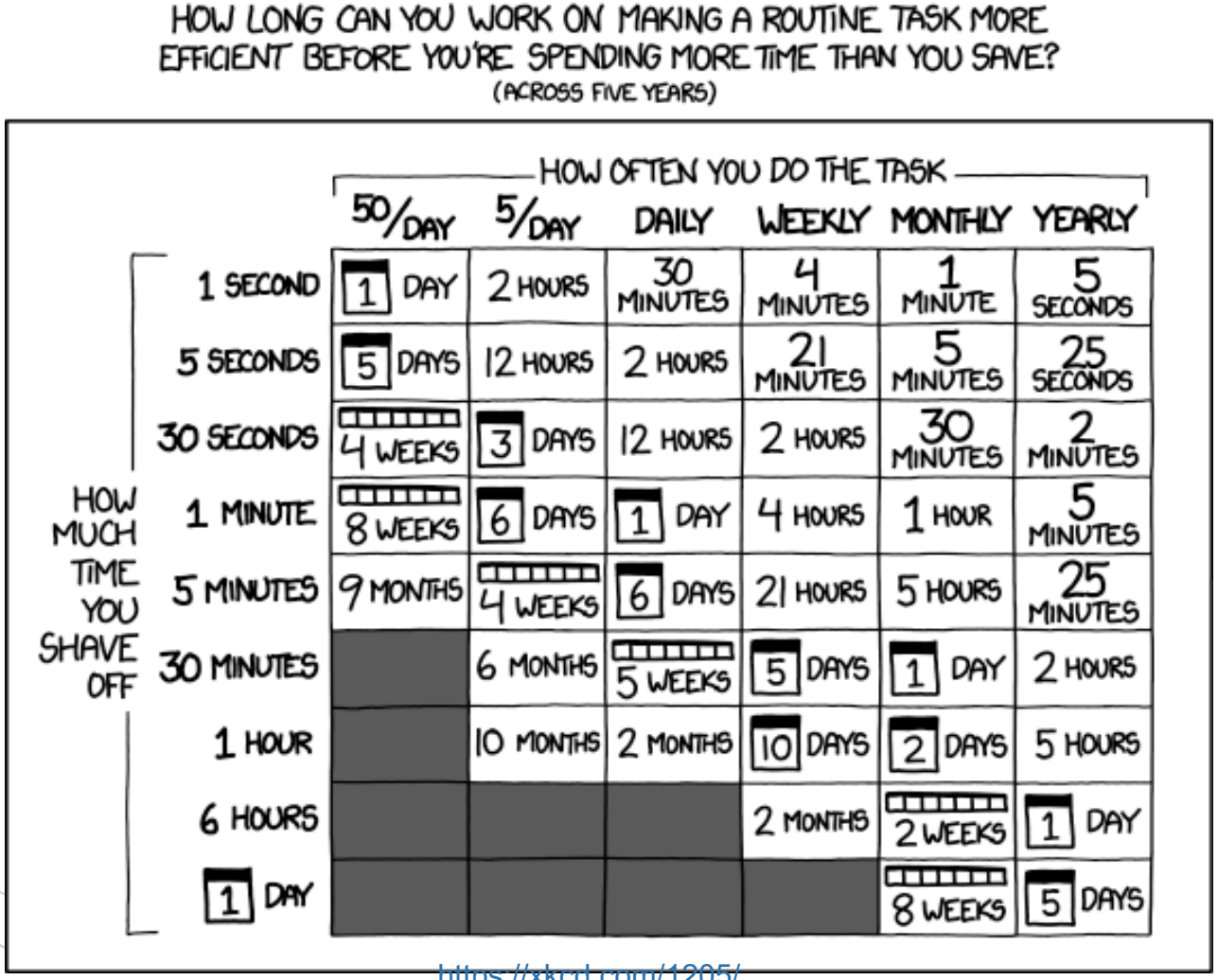
https://www.in.gov/indot/safety/zipper-merge/

Key Values Achievable with Merging:

- Shared acceptance of goals, objectives and risks and the methods for tracking and sharing them.
- Focus on the process objective, capability and customer need instead of document and section number in the document tree.
- Know the dependencies and priorities for just in time development.
- Shared risk and resolution
- Reduced team member spool up
- Continuous and integrated improvement.

Recommendations for Deployment to a project team.

1. Start by capturing your processes.
2. Take credit for existing good practices.
3. Create tasks for areas of improvement.
4. Ensure the tasks are in the project backlog
5. Plan them like any other backlog task for teams.
6. Collaborate and find solutions to problems together!
7. Share, accept, and mitigate all risks
8. Continuously Improve
9. Share Knowledge with Lessons Learned.



https://xkcd.com/1205/

How does merging increase resilience and accountability?

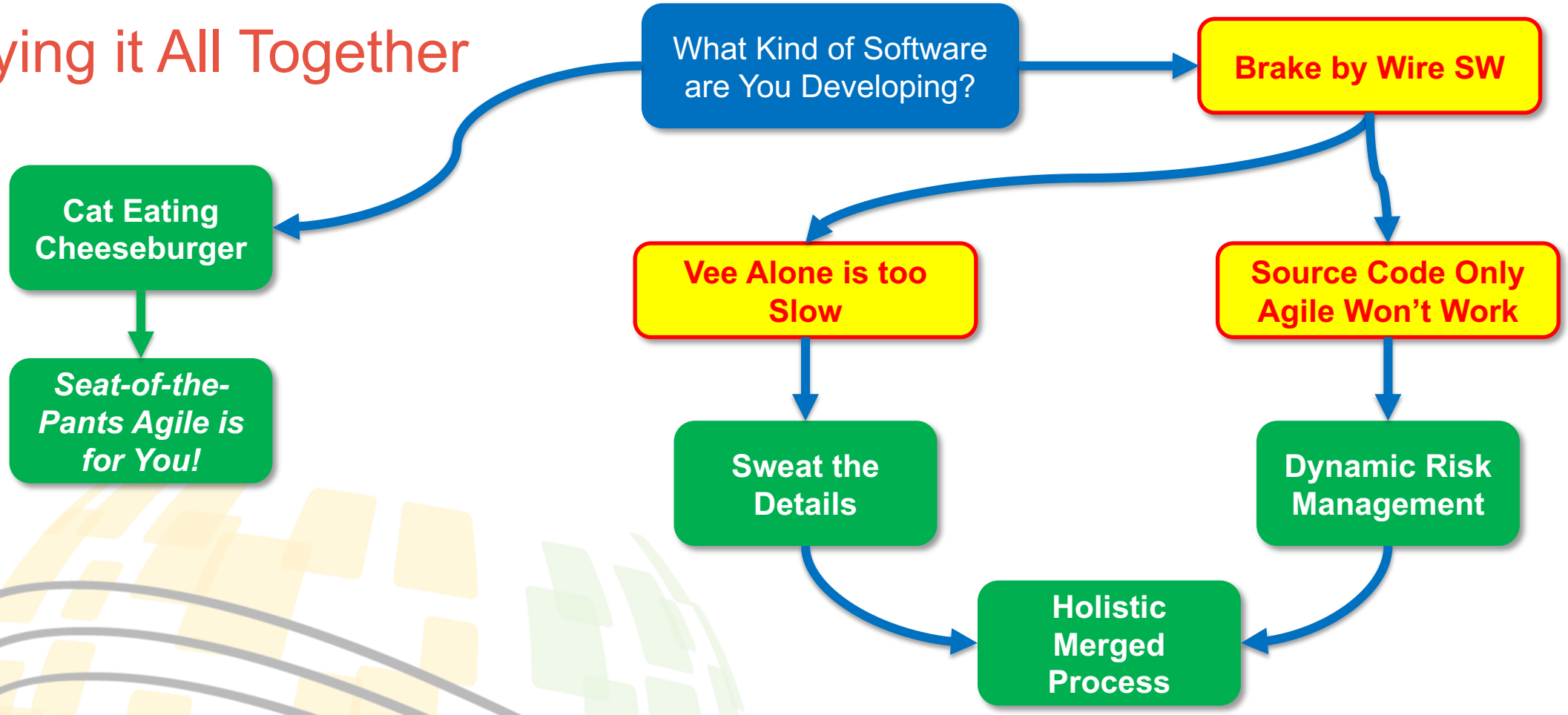
Resilience

- *Coverage of the standards is clear and accessible.*
  - *The Why is clear, accepted, and accessible to everyone*
- *Teams have more ability to adapt*
  - *Teams build and improve the How you go.*
- *Collaboration focuses on solutions and not team structures.*
  - *Solutions that meet the needs of the user*
- *Forces teams to share data and tools dynamically and bidirectionally*
- All decisions and activities are collaborative and tracked in backlogs accessible and fully linked.

Accountability

- Coverage is clear and only done one time
  - Infinite reuse, maintainability and adaptability as processes improve.
- Ability to track maturity over time as a solution.
- *Decision making is done with a broader view of the evidence for a solution not the individual pieces instead of a roll up of the pieces.*

Tying it All Together



- 3-5 week sprints aligned with project phase gates
- Vee is filled in as a mosaic, not a linear sequential slog
- Holistic project backlog management. Work on items that immediately reduce program risk.

References

- *Adaptive Acquisition Framework Pathways*. DAU. (n.d.). https://aaf.dau.edu/aaf/aaf-pathways/
- *Agile Systems & System Engineering*. INCOSE. (n.d.). https://www.incose.org/incose-member-resources/working-groups/transformational/agile-systems-se
- Anjoran, R. (2022, October 19). *What is the APQP? (Advanced Product Quality Planning Process)*. QualityInspection.org. https://qualityinspection.org/advanced-product-quality-planning/
- *DAD life cycle – Agile (scrum based)* . PMI Disciplined Agile (DA). (n.d.). https://www.pmi.org/disciplined-agile/lifecycle/agile-lifecycle
- Fairley, D., Forsberg, K., Madachy , R., & Marbach, P. (n.d.). *System lifecycle process models: VEE*. Guide to the System Engineering Body of Knowledge. https://sebokwiki.org/wiki/System_Lifecycle_Process_Models:_Vee
- Forsberg, K. (n.d.). *System lifecycle process models: Incremental*. Guide to the System Engineering Body of Knowledge. https://sebokwiki.org/wiki/System_Lifecycle_Process_Models:_Incremental
- *ISO 26262-1:2011*. ISO. (2018, December 17). https://www.iso.org/standard/43464.html
- Lean-Agile Mindset Core Values Safe - scaled agile framework. (n.d.). https://scaledagileframework.com/wp-content/uploads/delightful-downloads/2023/03/Full_SAFe_6.0_A4.pdf
- Maag, K. (n.d.). *AGILE SPICE Version 1.1 ready for piloting* . International Assessor Certification Scheme. https://www.kuglermaag.com/fileadmin/05_CONTENT_PDF/model_intacs_agile-SPICE.pdf
- Marbach, P. (n.d.). *System life cycle process models: Agile Systems Engineering*. Guide to the System Engineering Body of Knowledge. https://sebokwiki.org/wiki/System_Life_Cycle_Process_Models:_Agile_Systems_Engineering
- Shea, G. (2019, February 6). *3.0 NASA program/Project Life Cycle*. NASA. https://www.nasa.gov/seh/3-project-life-cycle
- VDA QMC Project Group 13 / Automotive SIG. (2017, November 01). Automotive SPICE Process Assessment / Reference Model v3.1
- VDA QMC Project Group 13. (2021, July 16). Automotive SPICE® for Cybersecurity Process Reference and Assessment Model. https://www.automotivespice.com/fileadmin/software-download/AutomotiveSPICE_for_Cybersecurity.pdf

Thank you!

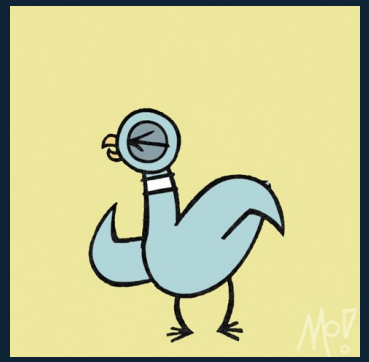
https://media.giphy.com/media/l0IyjeA5mmMZjhyPm/giphy.gif

Email Us!

33rd Annual INCOSE international symposium

hybrid event

Honolulu HI USA



https://media.giphy.com/media/P6q7wQz0wlcj6Nhzhd/giphy.gif