



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



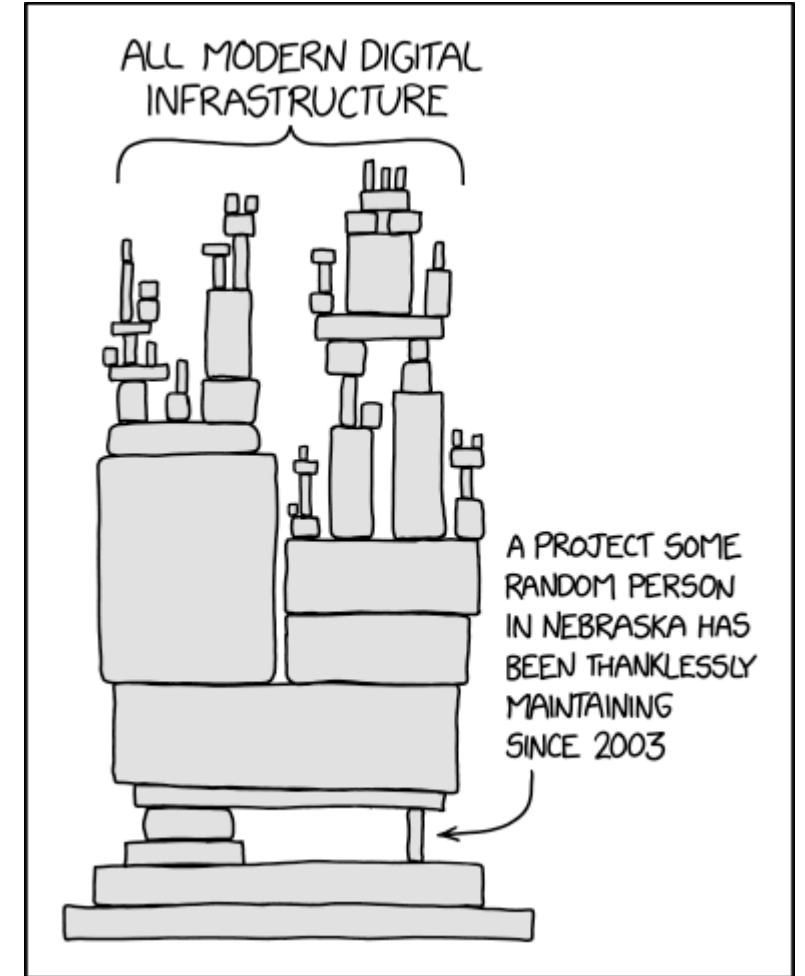
This Photo by Unknown Author is licensed under [CC BY-SA-NC](#)



This Photo by Unknown Author is licensed under [CC BY](#)



<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



Copyright 2016 Asatte Press, Inc – Used with Permission

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

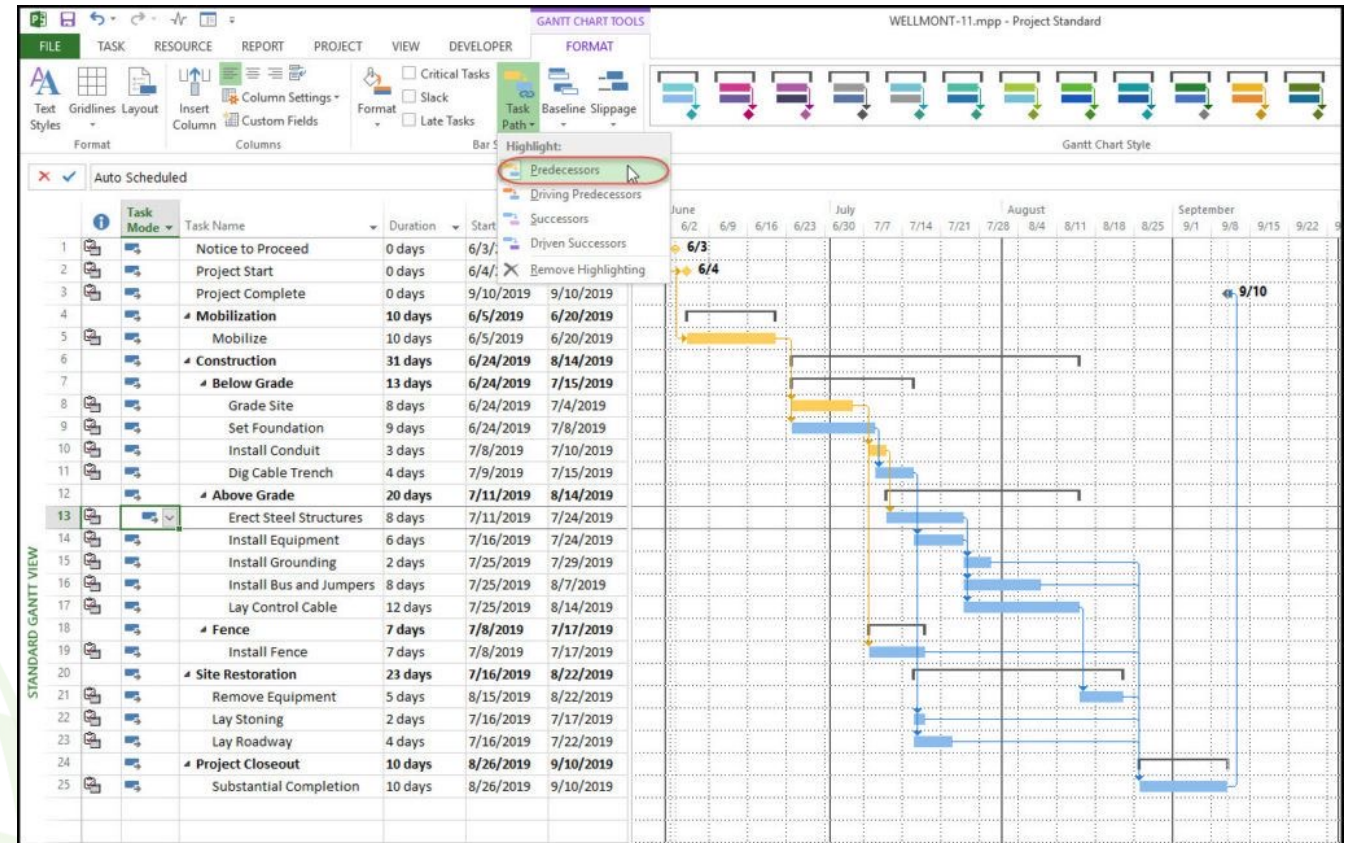
If a careless mistake in your requirements, coding, or verification can incinerate a family of five, casual software development approaches are simply not morally acceptable.



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- The project manager’s beautiful plan is out of date before he finishes pinning it to the wall.



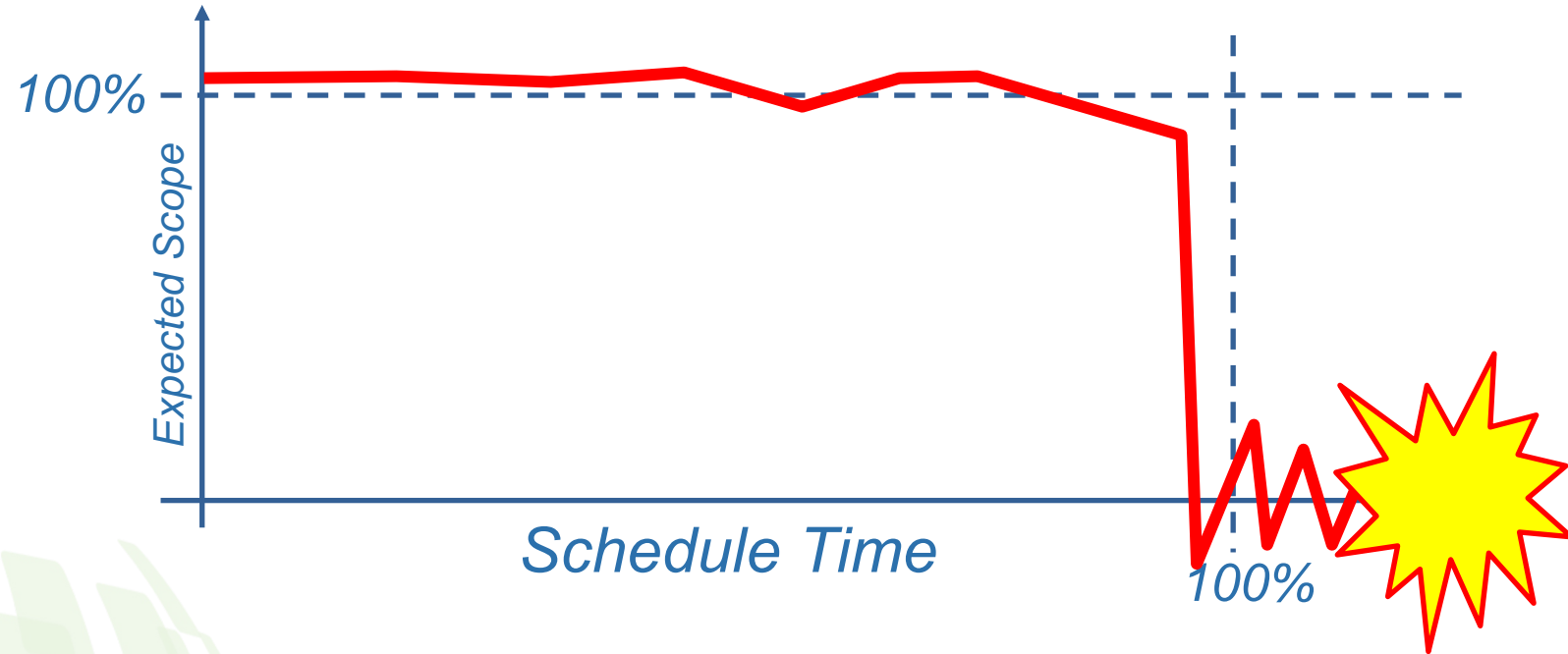
Agile Weakness – Eyeball Rolling about “Waterfall”

- 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)

1st Priority
“Critical Must
Have”
Requirements

2nd Priority “Super
Important” Requirements

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

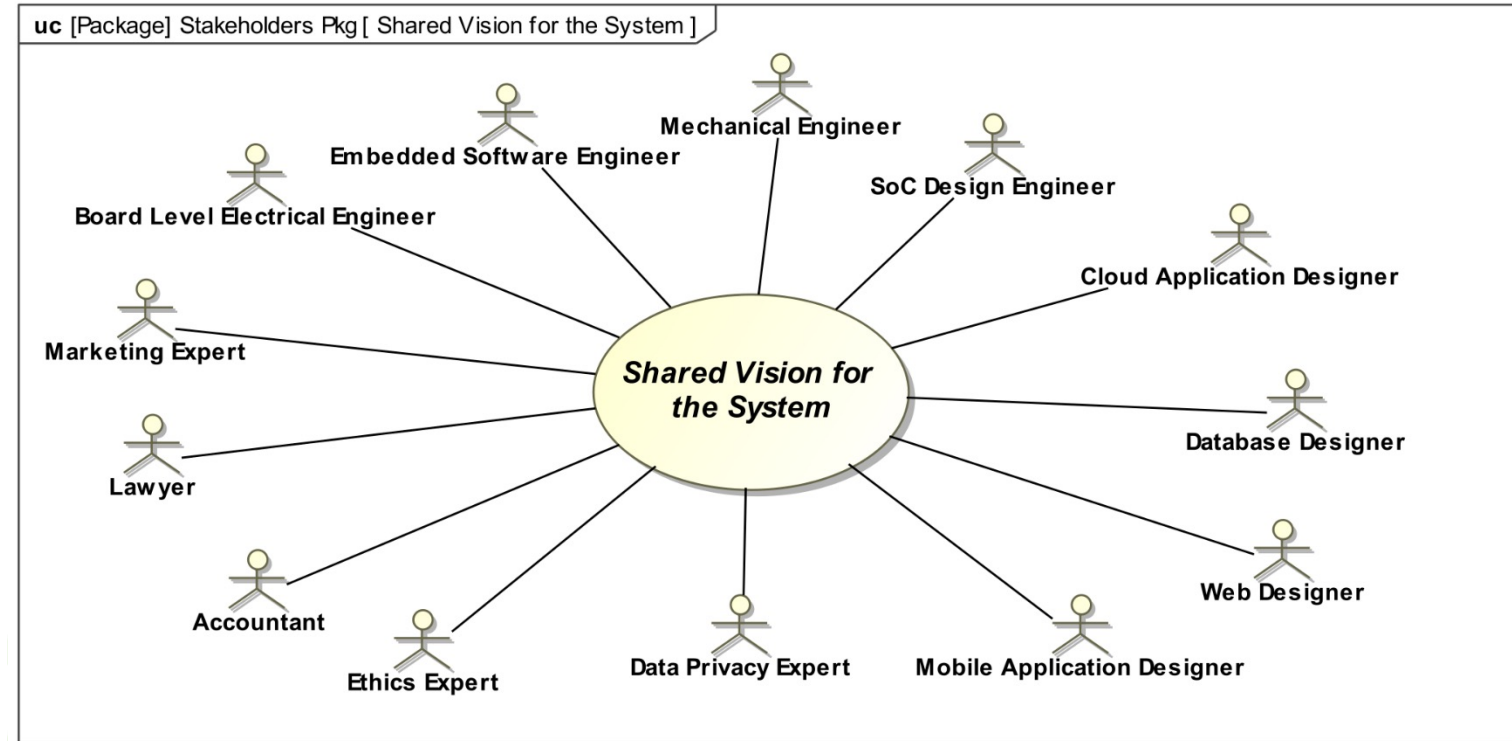
Agile Weakness – Software Belly Button Gazing

- *“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.



Copyright 2022 Asatte Press, Inc – from “Simple SysML for Beginners: Using CATIA No Magic Products”
Used with Permission



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.”

HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC)



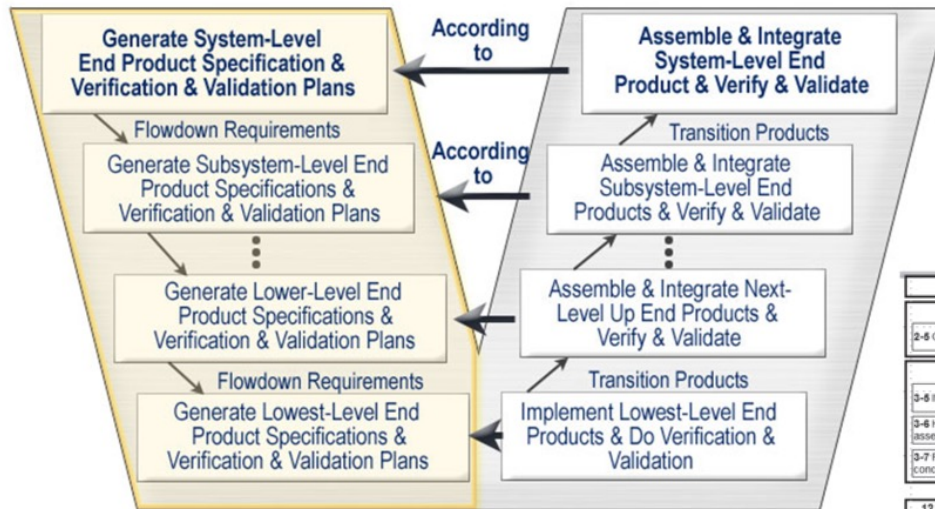
<https://xkcd.com/927/>



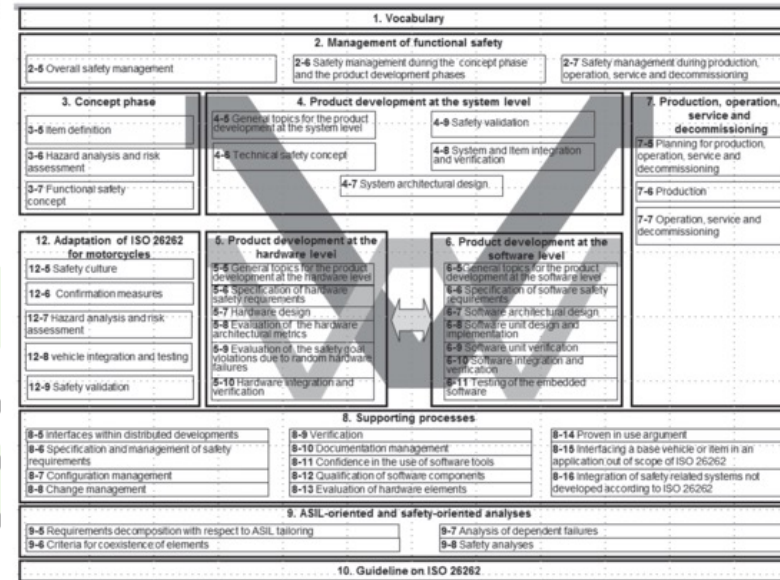
Vee Model Processes



The 'V' – A Macro View



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



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

www.incose.org/symp2023

AUTOMOTIVE SPICE®

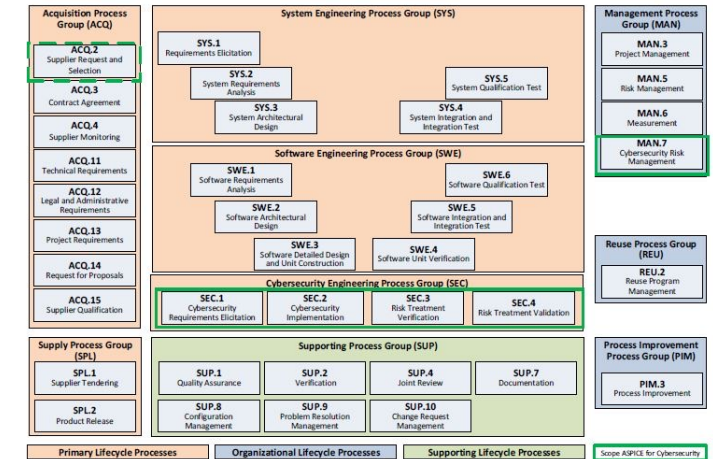
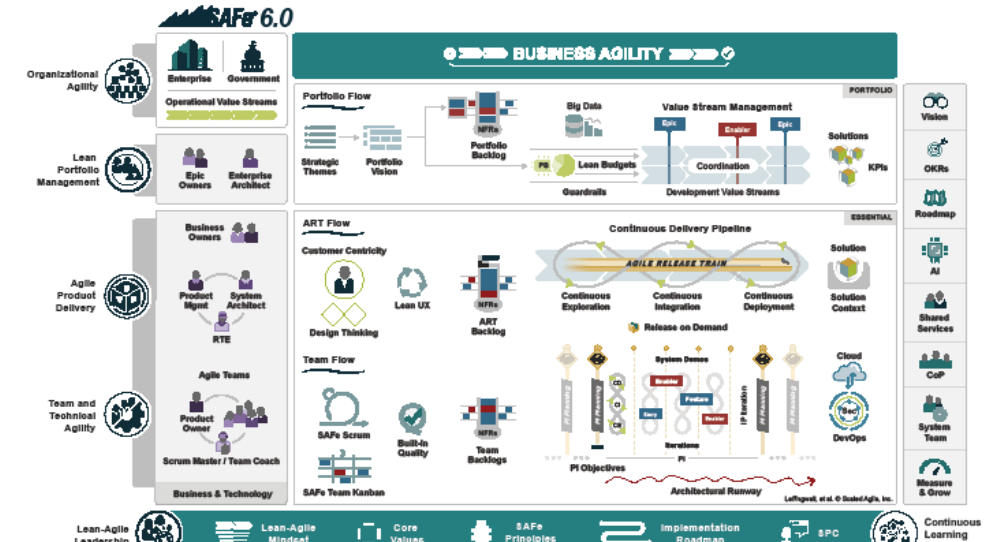
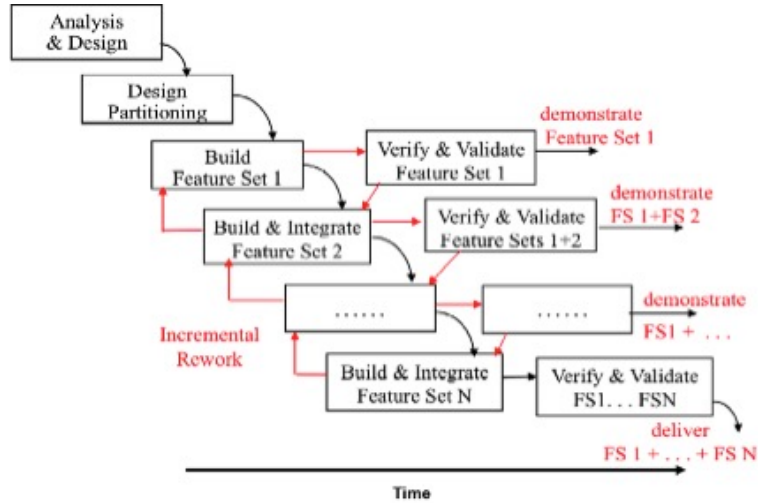


Figure 2 — Automotive SPICE and Automotive SPICE for Cybersecurity Process Reference Model – Overview

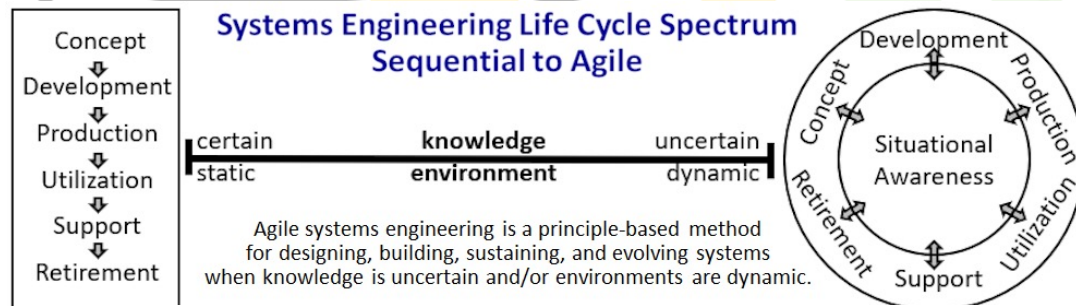
https://www.automotivespice.com/fileadmin/software-download/AutomotiveSPICE_for_Cybersecurity.pdf

Agile Processes

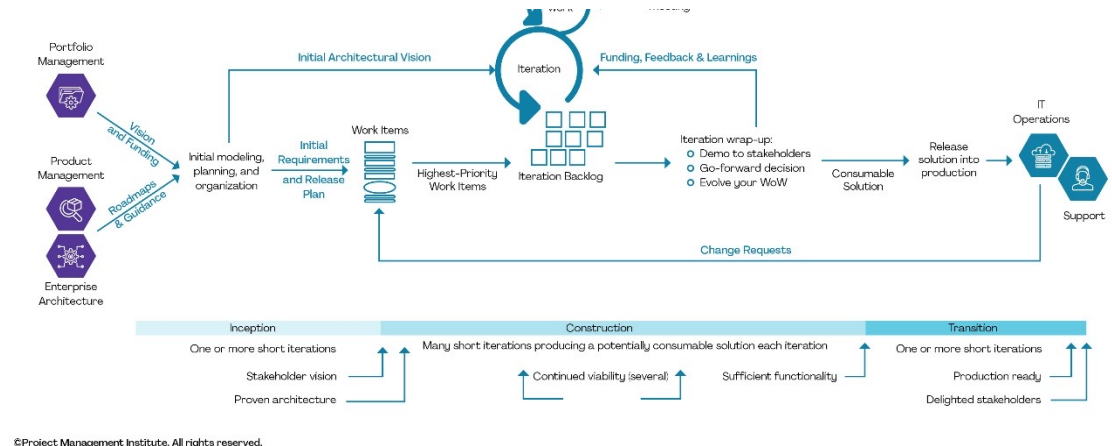


https://scaledagileframework.com/wp-content/uploads/delightful-downloads/2023/03/Portfolio_SAFe_6.0_US_Letter.pdf

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

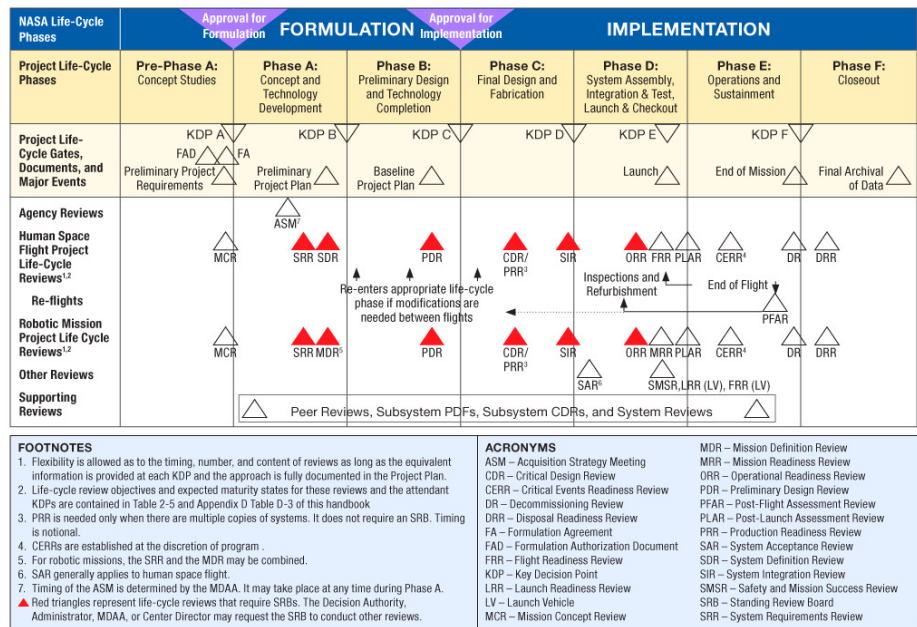
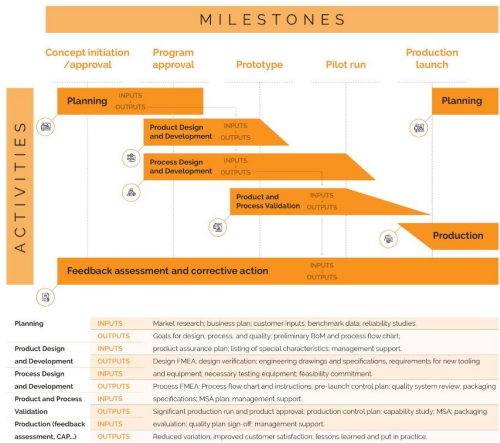
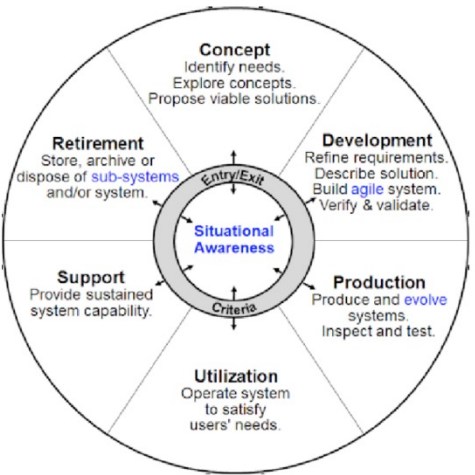


FIGURE 3.0-1 NASA Space Flight Project Life Cycle from NPR 7120.5E

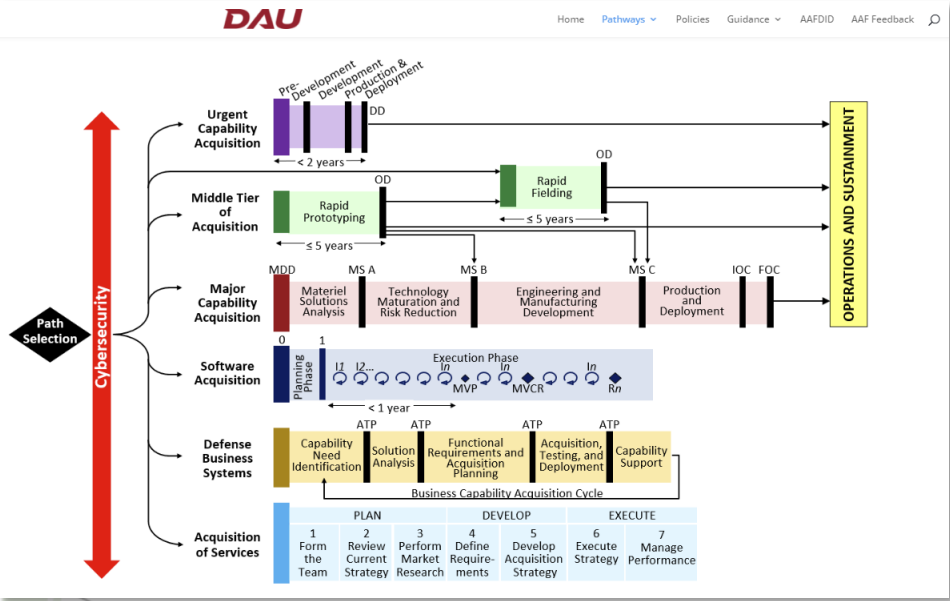
<https://www.nasa.gov/seh/3-project-life-cycle>



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



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

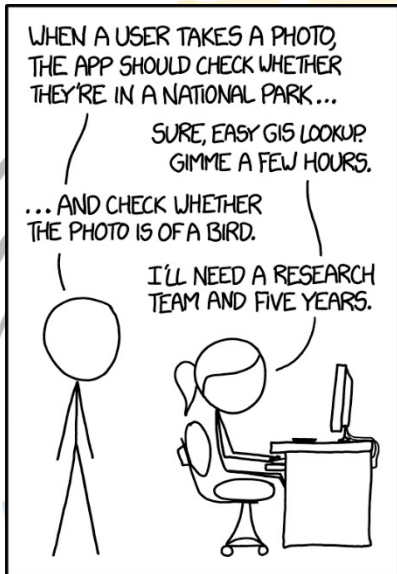


<https://aaf.dau.edu/aaf/aaf-pathways/>

Team and Organizational Structures



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



IN CS, IT CAN BE HARD TO EXPLAIN THE DIFFERENCE BETWEEN THE EASY AND THE VIRTUALLY IMPOSSIBLE.

<https://xkcd.com/1425/>



"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>

www.incose.org/symp2023

Cyber Security, Safety and Quality

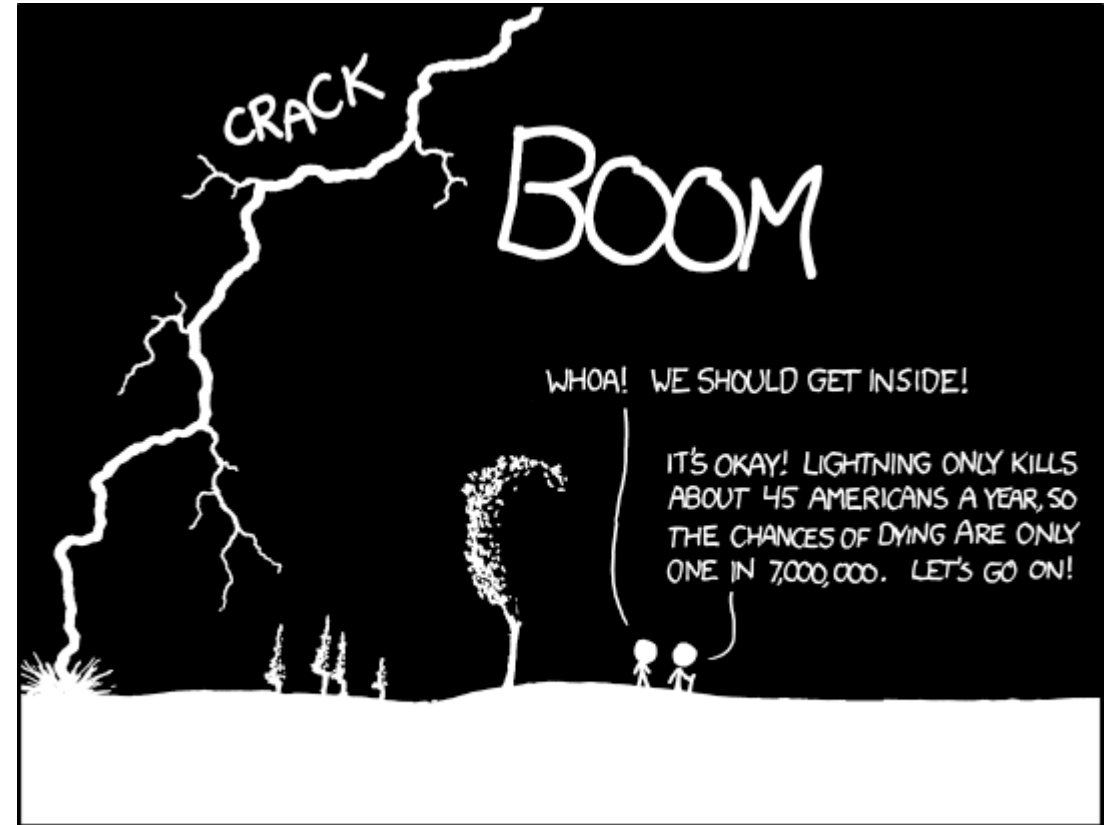
It's all about Risk Management!

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.



www.incose.org/symp2023



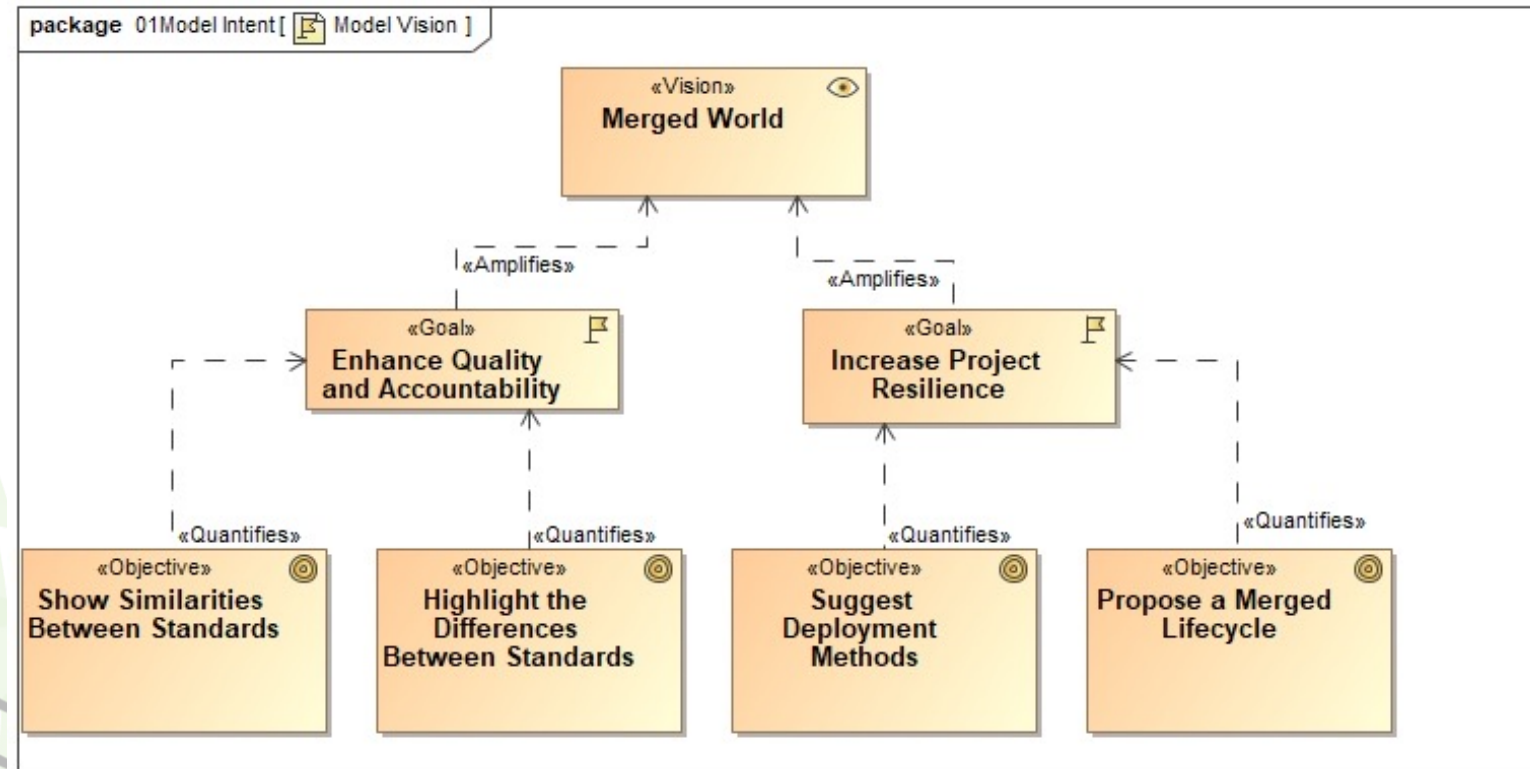
THE ANNUAL DEATH RATE AMONG PEOPLE WHO KNOW THAT STATISTIC IS ONE IN SIX.

<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

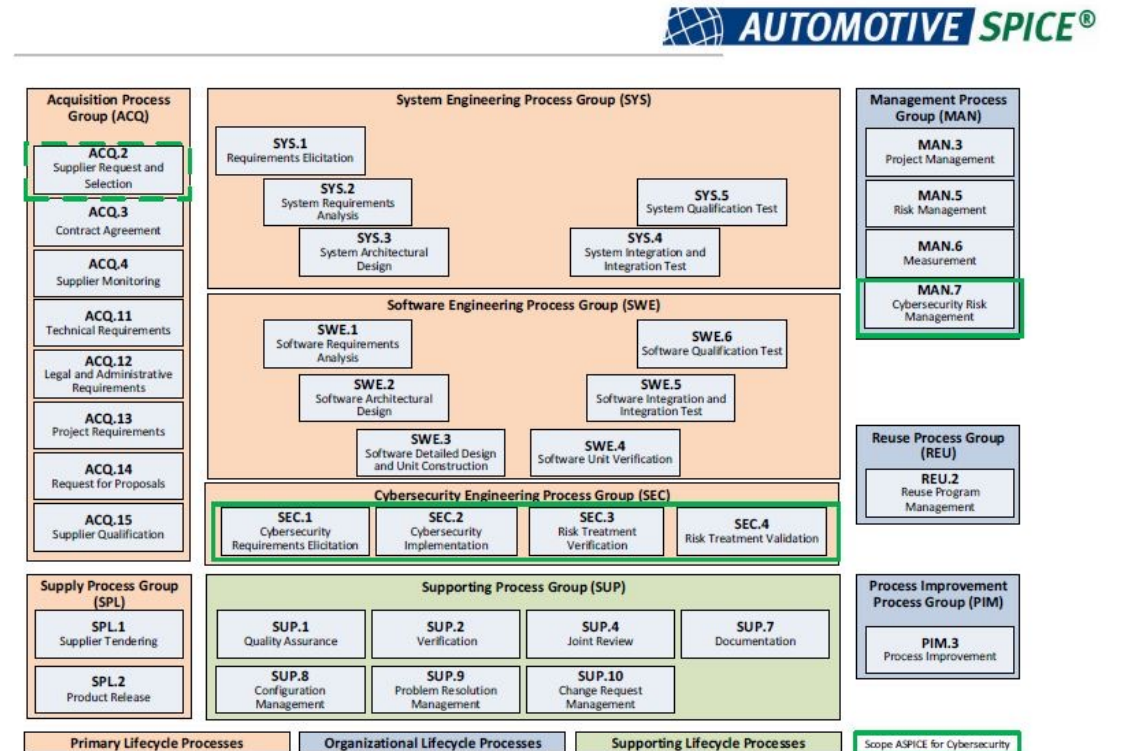
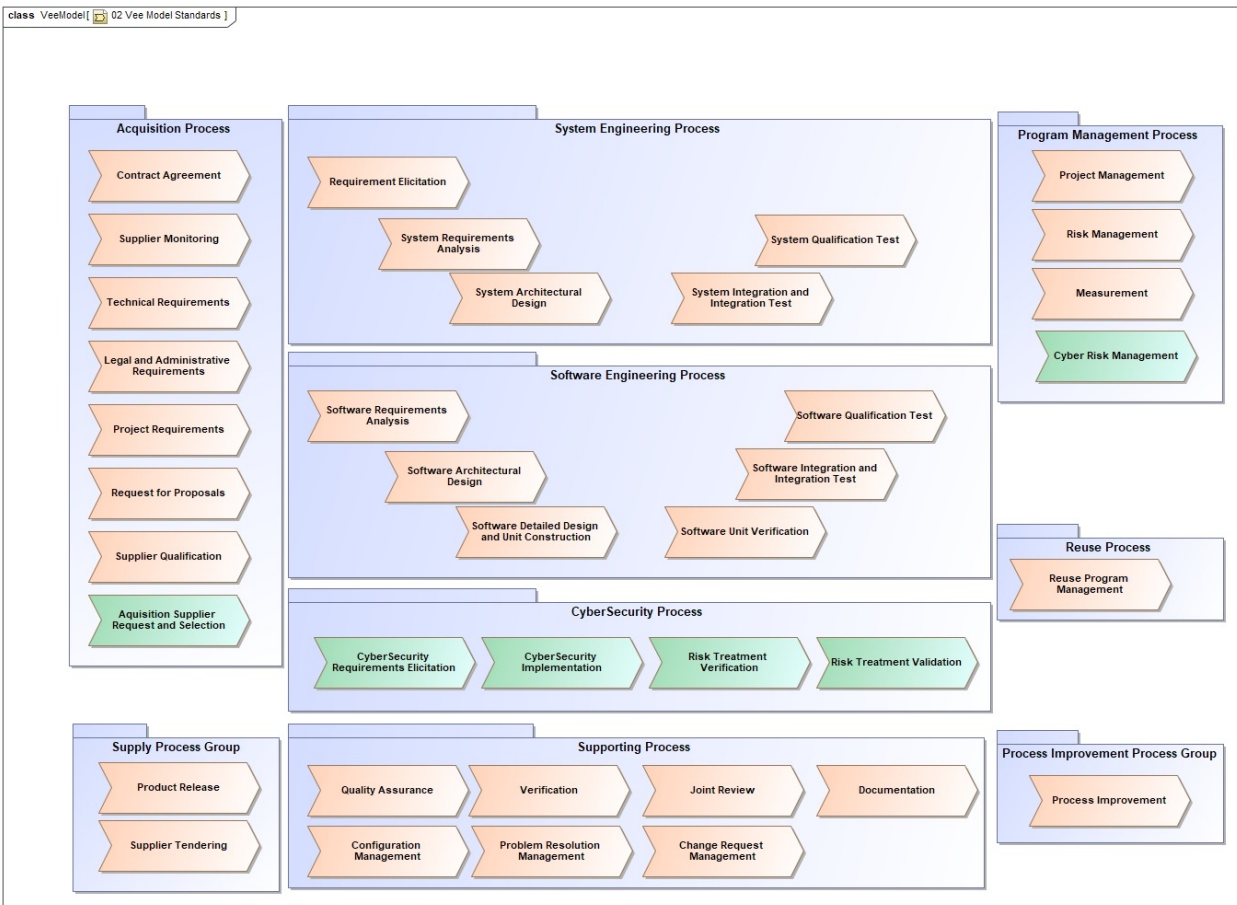
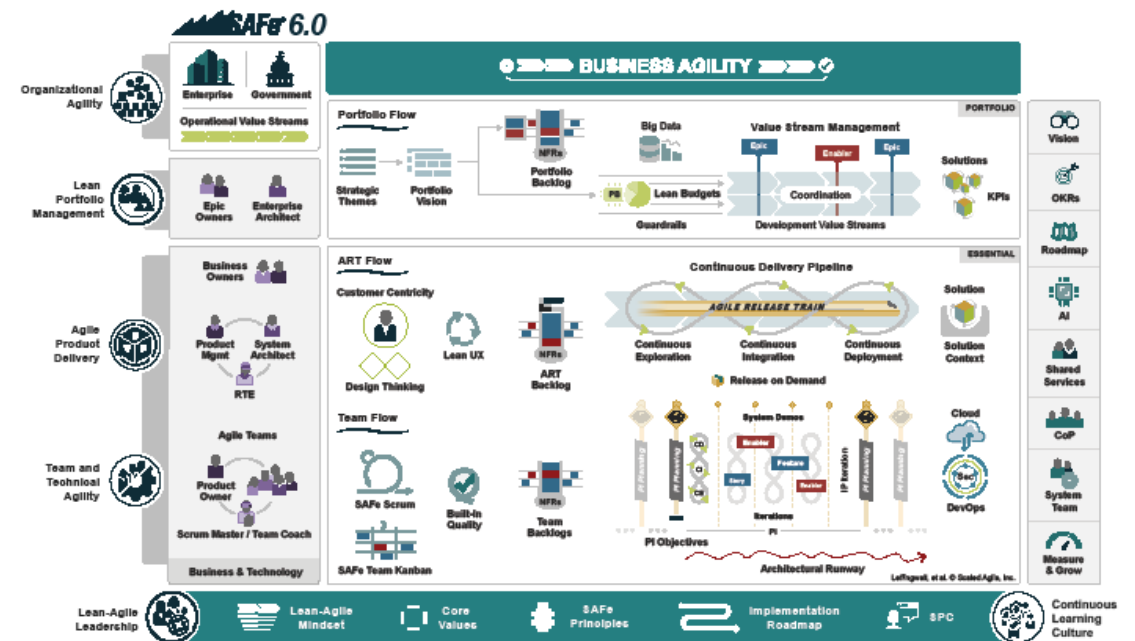
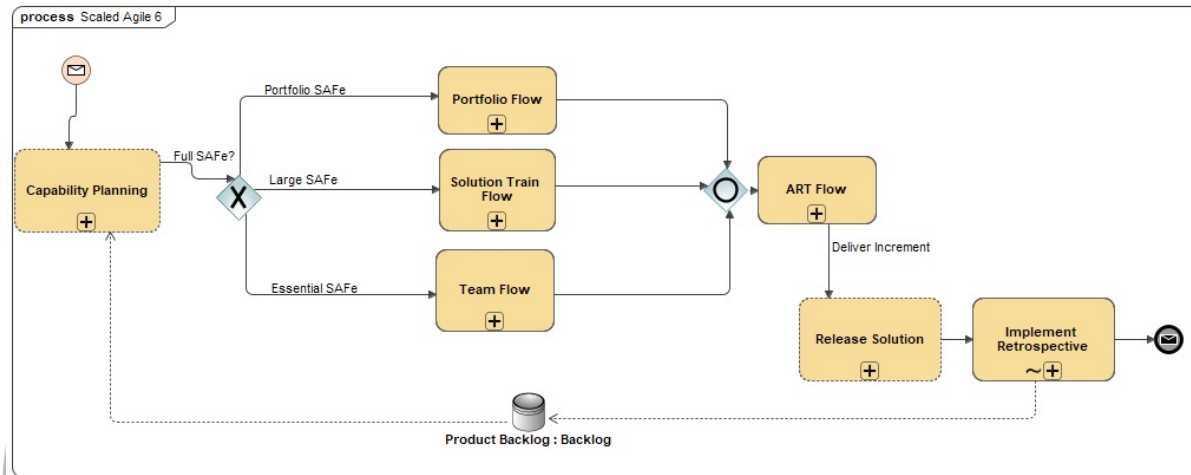


Figure 2 — Automotive SPICE and Automotive SPICE for Cybersecurity Process Reference Model – Overview

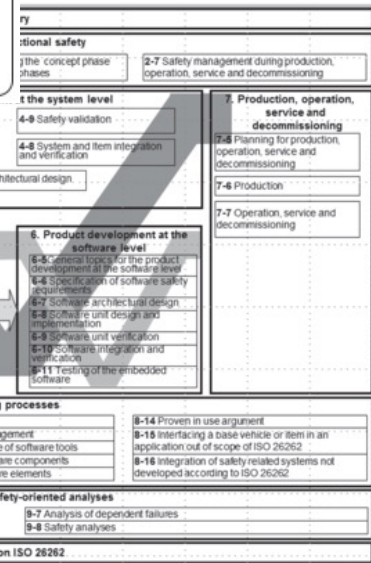
Modeling the Processes – Agile



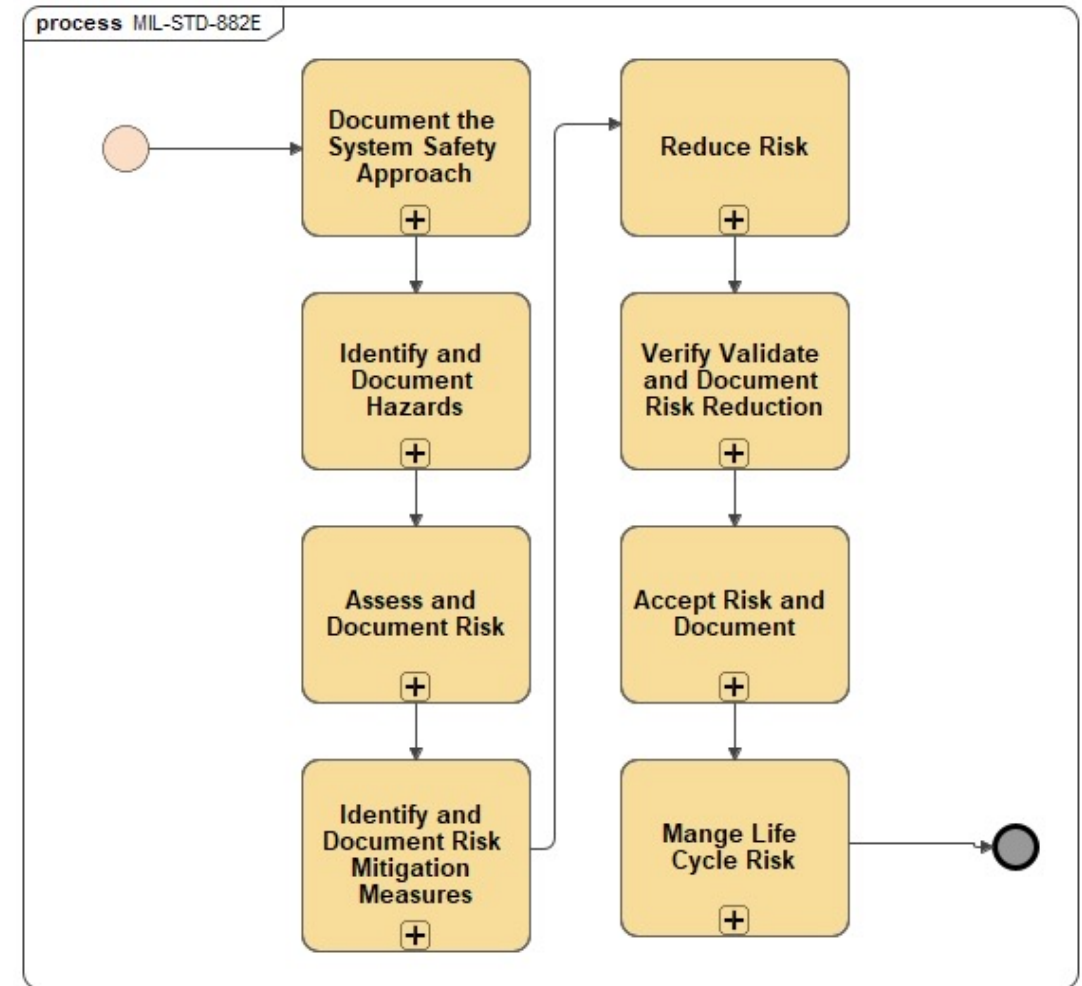
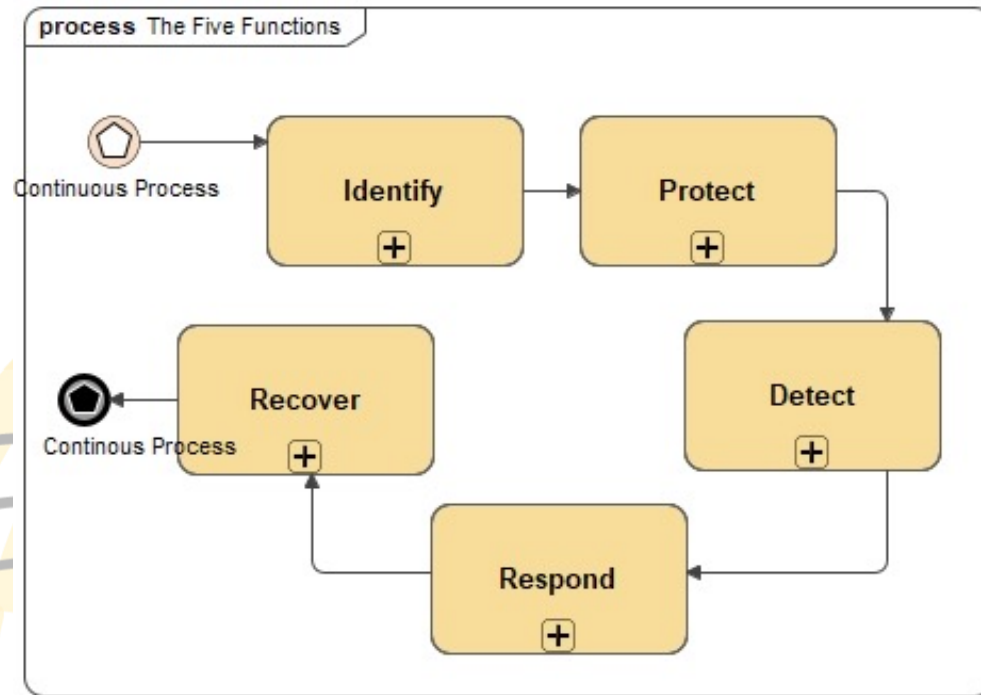
https://scaledagileframework.com/wp-content/uploads/delightful-downloads/2023/03/Portfolio_SAFe_6.0_US_Letter.pdf

© Scaled Agile, Inc.

process ISO-26262



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



Comparing Processes

Legend																								
Alignment																								
	ISO-26262	MIL-STD-882E	The Five Functions	VeeModel																				
	ASIL Oriented and Safety Oriented Analysis	Accept Risk and Document	Detect	Acquisition Process																				
	Concept Phase	Assess and Document Risk	Identify	CyberSecurity Process																				
	Management of Functional Safety	Document the System Safety Approach	Protect	Hardware Engineering Process																				
	Product Development Hardware Level	Identify and Document Safety Hazards	Recover	Hardware Architectural Metrics Evaluation																				
	Product Development Software Level	Identify and Document Risk Mitigation Measures	Respond	Hardware Design																				
	Product Development System Level	Manage Life Cycle Risk		Hardware Requirements																				
	Production and Operation	Reduce Risk		Hardware Verification against design																				
	Supporting Processes	Verify Validate and Document Risk Reduction		Hardware Verification against requirements																				
				Mechanical Engineering Process																				
				Mechanical Design																				
				Mechanical Requirements																				
				Mechanical Verification against design																				
				Mechanical Verification against requirements																				
				Process Improvement Process																				
				Program Management																				
				Reuse Process																				
				Safety Process																				
				Software Engineering Process																				
				Software Detailed Design and Unit Construc																				
				Software Integration and Integration Test																				
				Software Qualification Test																				
				Software Requirements																				
				Software Unit Verification																				
				Supply and Release Process																				
				Supporting Process																				
				System Engineering Process																				
				Requirement Elicitation																				
				System Architectural Design																				
				System Integration and Integration Test																				
				System Qualification Test																				
				System Requirements Analysis																				

Similarities

Same objectives, context and activities

Differences

Names of activities, sequencing, roles included, and definition of done.

Narrative

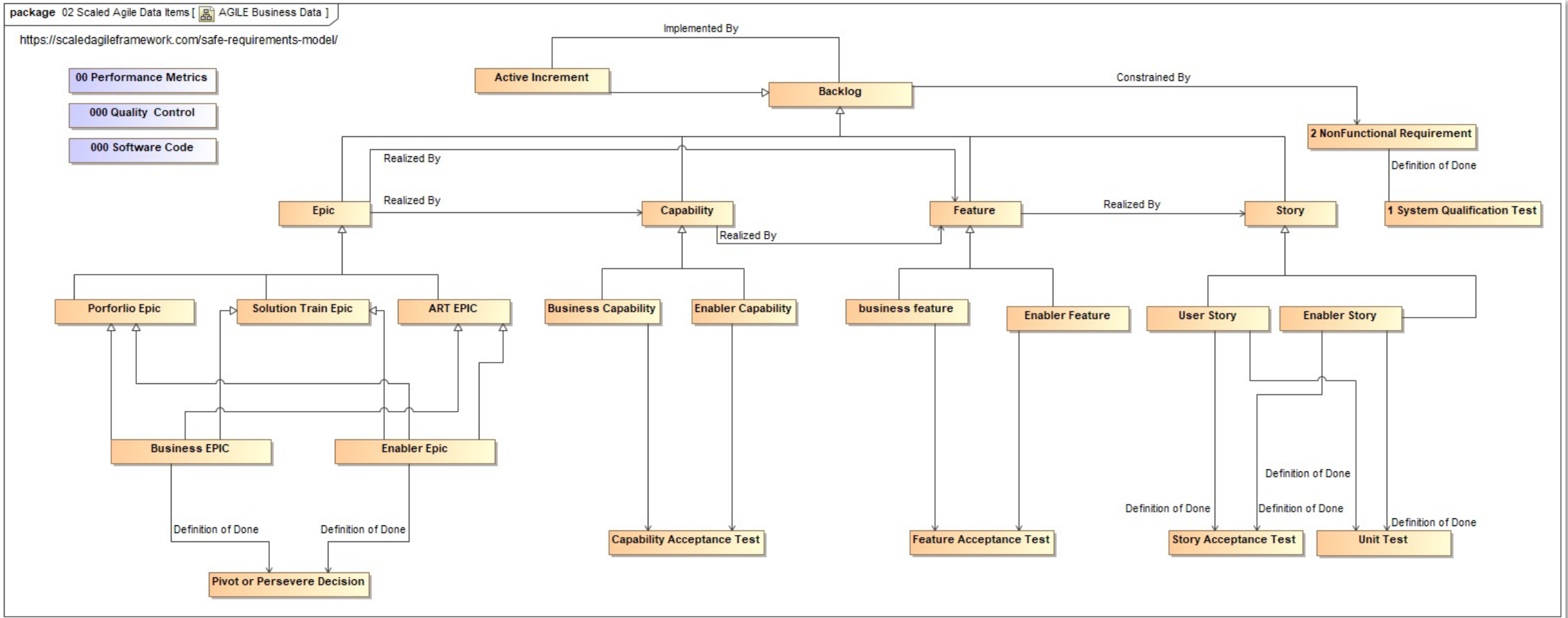
In Agile it covers multiple process areas in a modified parallel workflow.
In Vee Model each process is seen as a workflow state and required complete to transition.

Modeling the Evidence: Vee Model

package Work Outputs | 02 Vee Model Vee Products | de/wp-content/uploads/2023/06/Automotive-SPICE-PAM-40-Gelbandrelease.pdf

21-00 Work Product	20-00 Template	19-11 Validation Strategy	19-10 Verification Strategy	19-05 Reuse Strategy
19-00 Strategy	18-50 Supplier Qualification Criteria	18-07 Quality Criteria	18-06 Product Release Criteria	18-01 Acceptance Criteria
18-00 Standard	17-50 Verification Criteria	17-12 System Requirements Specification	17-11 Software Requirements Specification	17-08 Interface Requirements Specification
17-05 Documentation Requirements	17-03 Stakeholder Requirements	17-02 Build Test	17-00 Requirement Specification	16-06 Process Repository
16-03 Configuration Management System	16-00 Repository	15-21 Supplier Evaluation Report	15-18 Process Performance Report	15-16 Improvement Opportunity
15-13 Assessment/Audit Report	15-12 Problem Status Report	15-09 Risk Status Report	15-08 Risk Analysis Report	15-07 Reuse Evaluation Report
15-06 Project Status Report	15-05 Evaluation Report	15-03 Configuration Status Report	15-01 Analysis Report	15-00 Report
14-50 Stakeholder Groups List	14-11 Work Product List	14-09 Work Breakdown Structure	14-08 Tracking System	14-06 Schedule
14-05 Preferred Suppliers Register	14-02 Corrective Action Register	14-01 Change History		14-00 Register
13-50 Test Result	13-25 Verification Results	13-24 Validation Results	13-22 Traceability Records	13-21 Change Control Record
13-20 Risk Action Request	13-19 Review Records	13-18 Quality Record	13-17 Customer Request	13-16 Change Request
13-15 Proposal Review Record	13-14 Progress Status Record	13-13 Product Release Approval Record	13-10 Configuration Management Record	13-09 Meeting Support Record
13-08 Baseline	13-07 Problem Record	13-06 Delivery Record	13-05 Contract Review Record	13-04 Communication Records
13-01 Acceptance Record	13-00 Record	12-04 Supplier Proposal Response	12-03 Reuse Proposal	12-01 Request for Proposal
12-00 Proposal	11-07 Temporary Solution	11-06 System	11-05 Software Unit	11-04 Product Release Package
11-03 Product Release Information	11-00 Product	10-00 Process Description	09-03 Reuse Policy	09-00 Policy
08-52 Test Plan	08-51 Technology Monitoring Plan	08-50 Test Specification	08-29 Improvement Plan	08-28 Change Management Plan
08-27 Problem Management Plan	08-26 Documentation Plan	08-20 Risk Migration Plan	08-19 Risk Management Plan	08-18 Review Plan
08-17 Reuse Plan	08-16 Release Plan	08-14 Recovery Plan	08-13 Quality Plan	08-12 Project Plan
08-04 Configuration Management Plan	08-00 Plan	07-08 Service Level Measure	07-07 Risk Measure	07-06 Quality Measure
07-05 Project Measure	07-04 Process Measure	07-03 Personal Performance Measure	07-02 Field Measure	07-01 Customer Satisfaction Survey
07-00 Measure	06-02 Handling and Storage Guide	06-01 Customer Manual	06-00 User Documentation	05-00 Goals
04-06 System Architectural Design	04-05 Software Detailed Design	04-04 Software Architectural Design	04-03 Domain Model	04-02 Domain Architecture
04-00 Design	03-06 Process Performance Data	03-04 Customer Satisfaction Data	03-03 Benchmarking Data	03-00 Data
02-01 Commitment/Agreement	01-51 Application Parameter	01-50 Integrated Software	01-03 Software Item	01-00 Configuration Item

Modeling the Evidence: Agile



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

Comparing Evidence

Legend		Work Outputs	
Alignment			
Agile Data Items			
00 Performance Metrics	91		
000 Quality Control	76		
000 Software Code	69		
Active Increment	65		
Backlog	65		
1 System Qualification Test	77		
2 NonFunctional Requirement	67		
Capability	68		
Business Capability	80		
Capability Acceptance Test	80		
Enabler Capability	88		
Epic	68		
ART EPIC	68		
Business EPIC	75		
Enabler Epic	77		
Pivot or Persevere Decision	72		
Portfolio Epic	75		
Solution Train Epic	76		
Feature	69		
business feature	72		
Enabler Feature	73		
Feature Acceptance Test	78		
Story	74		
Enabler Story	79		
Story Acceptance Test	81		
Unit Test	81		
User Story	77		
01-09 Configuration Item	27		
01-09 Software Item	1		
01-50 Integrated Software	1		
01-51 Application Parameter	2		
02-01 Commitment/Agreement	2		
03-00 Data	27		
03-03 Benchmarking Data	1		
03-04 Customer Satisfaction Data	1		
03-06 Process Performance Data	16		
04-00 Design	3		
04-03 Domain Architecture	1		
04-04 Software Architectural Design	7		
04-05 Software Detailed Design	4		
04-06 System Architectural Design	16		
05-00 Goals	12		
06-00 User Documentation	5		
06-01 Customer Manual	2		
06-02 Handling and Storage Guide	3		
07-00 Measure	27		
07-01 Customer Satisfaction Survey	2		
07-03 Field Measure	3		
07-09 Personal Performance Measure	3		
07-04 Process Measure	2		
07-06 Project Measure	27		
07-08 Quality Measure	2		
07-07 Risk Measure	2		
07-08 Service Level Measure	2		
08-00 Plan	27		
08-04 Configuration Management Plan	27		
08-10 Project Plan	27		
08-13 Quality Plan	27		
08-14 Recovery Plan	27		
08-16 Release Plan	27		
08-17 Reuse Plan	27		
08-18 Review Plan	27		
08-19 Risk Management Plan	27		
08-20 Risk Migration Plan	27		
08-25 Documentation Plan	27		
08-27 Problem Management Plan	27		
08-28 Change Management Plan	27		
08-29 Improvement Plan	27		
08-30 Test Specification	5		
08-51 Technology Monitoring Plan	2		
08-52 Test Plan	5		
09-00 Policy	4		
09-03 Reuse Policy	2		
09-08 Process Description	2		
1-00 Product	1		
1-03 Product Release Information	1		
1-04 Product Release Package	4		
1-05 Software Unit	1		
1-06 System	2		
1-07 Temporary Solution	1		
1-08 Proposal	1		
1-01 Request for Proposal	1		
1-03 Reuse Proposal	1		
1-04 Supplier Proposal Responses	1		
1-06 Record	27		
1-01 Acceptance Record	27		
1-04 Communication Records	27		
1-06 Contract Review Record	9		
1-07 Delivery Record	27		
1-07 Problem Record	27		
1-08 Baseline	27		
1-09 Meeting Support Record	27		
1-10 Configuration Management Record	27		
1-13 Product Release Approval Record	27		
1-14 Progress Status Record	27		
1-15 Change Request	27		
1-16 Change Request	27		
1-17 Customer Request	1		
1-18 Quality Record	27		
1-19 Review Record	27		
1-20 Risk Action Request	27		
1-21 Change Control Record	27		
1-22 Traceability Records	27		
1-24 Validation Results	7		
1-25 Verification Results	7		
1-26 Test Result	27		
1-40 Register	27		
1-41 Change History	27		
1-42 Corrective Action Register	27		
1-45 Preferred Suppliers Register	3		
1-46 Schedule	5		
1-48 Trading System	27		
1-49 Work Breakdown Structure	3		
1-11 Work Product List	27		
1-45 Stakeholder Groups List	27		
1-50 Report	27		
1-01 Analysis Report	27		
1-03 Configuration Status Report	27		
1-05 Evaluation Report	27		
1-06 Project Status Report	27		
1-07 Reuse Evaluation Report	27		
1-08 Risk Analysis Report	27		
1-09 Risk Status Report	27		
1-12 Problem Status Report	27		
1-13 Assessment/Audit Report	9		
1-16 Improvement Opportunity	9		
1-18 Process Performance Report	9		
1-21 Supplier Evaluation Report	27		
1-00 Repository	27		
1-03 Configuration Management System	27		
1-06 Process Repository	27		
1-09 Requirement Specification	26		
1-02 Build Test	7		
1-03 Stakeholder Requirements	27		
1-05 Documentation Requirements	27		
1-06 Interface Requirements Specification	27		
1-11 Software Requirements Specification	10		
1-12 System Requirements Specification	22		
1-50 Verification Criteria	22		
1-01 Acceptance Criteria	27		
1-06 Product Release Criteria	3		
1-07 Quality Criteria	27		
1-05 Supplier Qualification Criteria	3		
1-05 Strategy	27		
1-05 Reuse Strategy	27		
1-10 Verification Strategy	4		
1-11 Validation Strategy	4		
20-00 Template	27		
1-00 Work Product	27		

Similarities

Same objectives, context and required evidence.

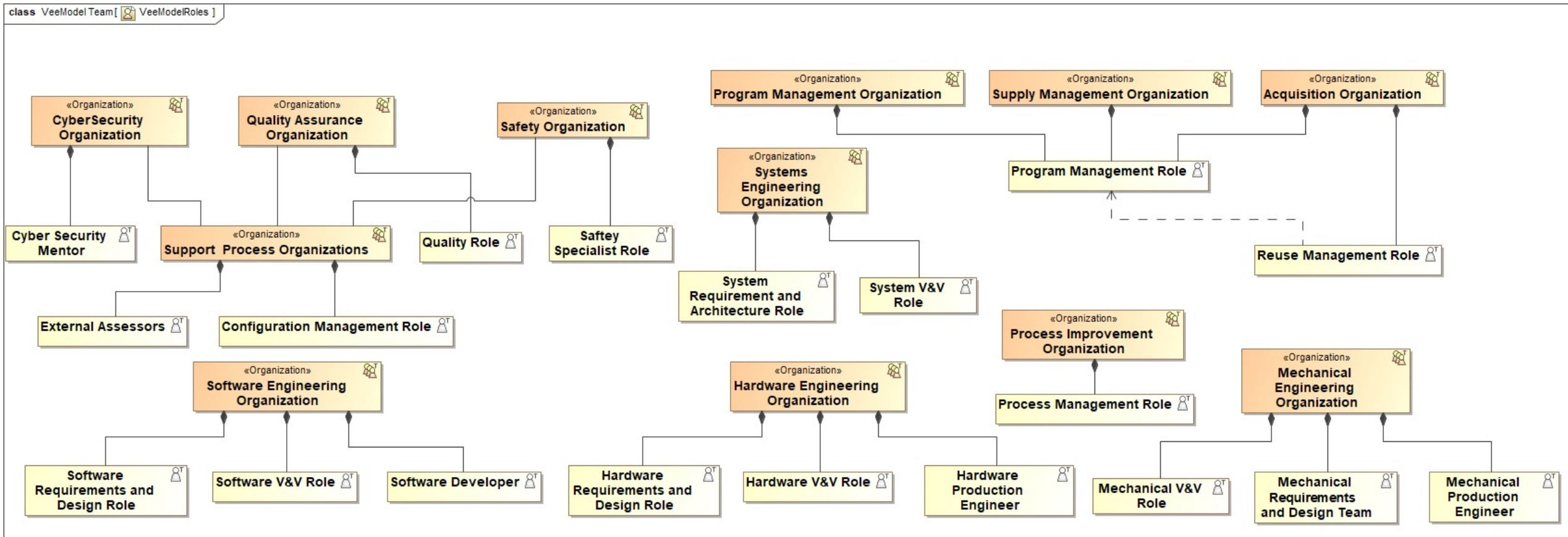
Differences

When it is complete.

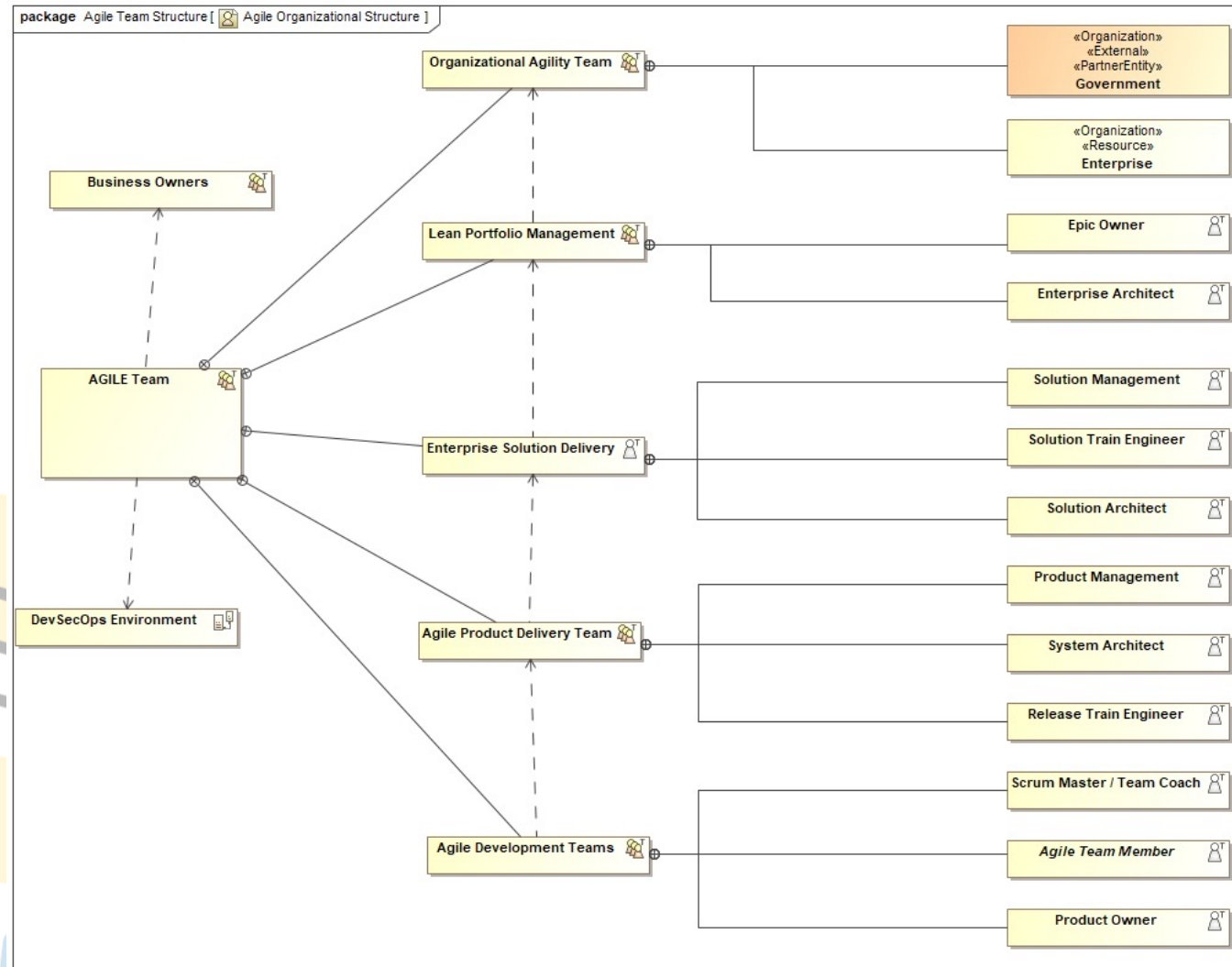
Narrative

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

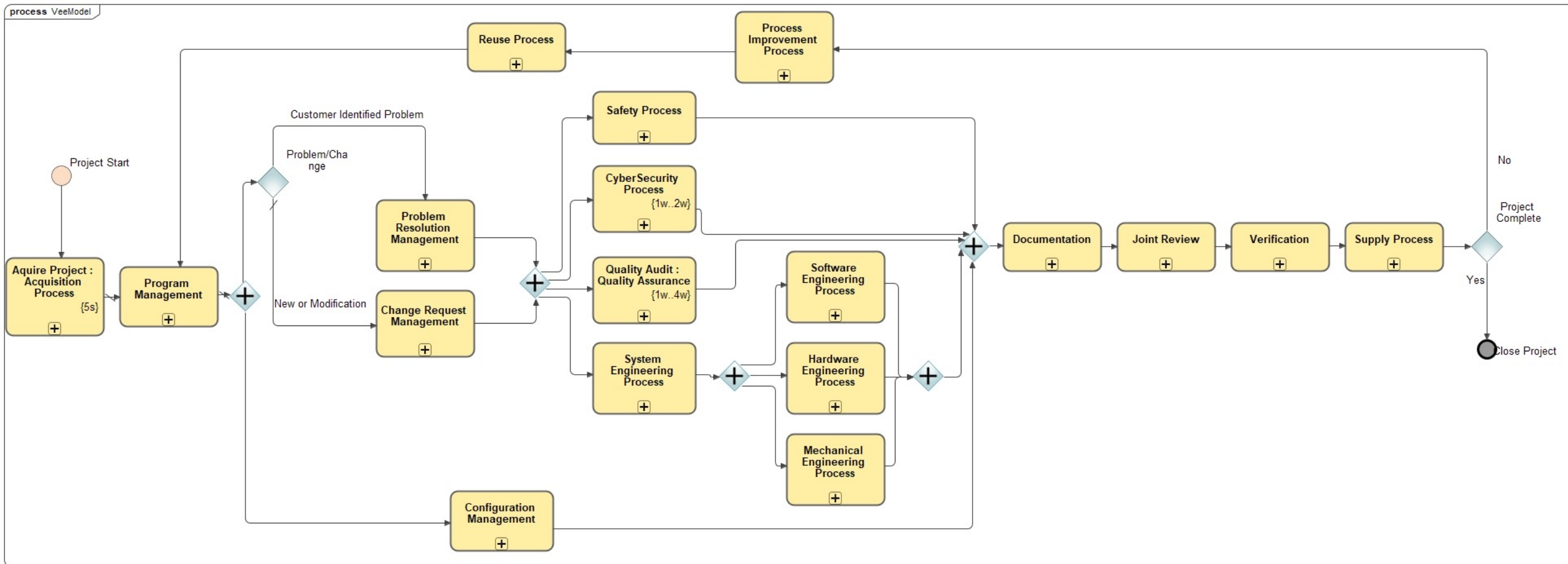


Comparing Team Structures

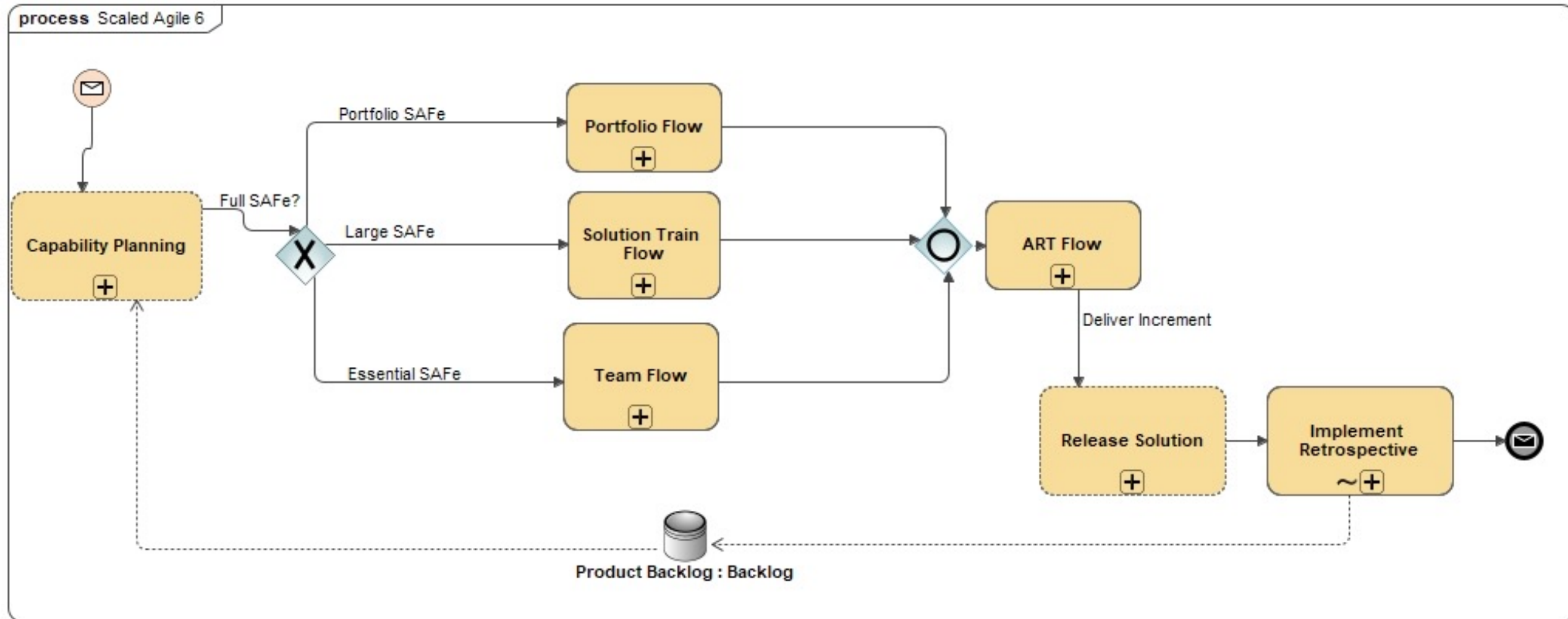
Similarities	Same objectives, context and required skills.
Differences	When to perform which roles and when to collaborate between roles.
Narrative	<p>In Agile, Roles are more hats you wear when you have the required skills to collaborate on the solution.</p> <p>In Vee Model, Roles are your position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

Legend	VeeModel Team																							
Alignment	Acquisition Organization	Configuration Management Role	CyberSecurity Organization	External Assessors	Hardware Engineering Organization	Hardware Production Engineer	Hardware Requirements and Design Role	Mechanical Engineering Organization	Mechanical Production Engineer	Mechanical Requirements and Design Team	Process Improvement Organization	Process Management Role	Program Management Organization	Program Management Role	Quality Assurance Organization	Quality Role	Reuse Management Role	Safety Organization	Safety Specialist Role	Software Developer	Software Engineering Organization	Software Requirements and Design Role	Supply Management Organization	Support Process Organizations
AGILE Team																								
Agile Development Teams	✓		✓					✓			✓	✓				✓	✓		✓					
Agile Team Member		✓	✓			✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓
Product Owner		✓									✓	✓	✓	✓	✓	✓	✓	✓						✓
Scrum Master / Team Coach											✓	✓	✓	✓	✓	✓	✓	✓						✓
Agile Product Delivery Team	✓										✓	✓	✓	✓	✓	✓	✓	✓						
Product Management		✓									✓	✓	✓	✓	✓	✓	✓	✓						
Release Train Engineer		✓									✓	✓	✓	✓	✓	✓	✓	✓						
System Architect											✓	✓	✓	✓	✓	✓	✓	✓						
Business Owners											✓	✓	✓	✓	✓	✓	✓	✓						
Enterprise Solution Delivery	✓										✓	✓	✓	✓	✓	✓	✓	✓						
Solution Architect											✓	✓	✓	✓	✓	✓	✓	✓						
Solution Management											✓	✓	✓	✓	✓	✓	✓	✓						
Solution Train Engineer											✓	✓	✓	✓	✓	✓	✓	✓						
Lean Portfolio Management	✓										✓	✓	✓	✓	✓	✓	✓	✓						
Enterprise Architect											✓	✓	✓	✓	✓	✓	✓	✓						
Epic Owner											✓	✓	✓	✓	✓	✓	✓	✓						
Organizational Agility Team	✓										✓	✓	✓	✓	✓	✓	✓	✓						
Enterprise											✓	✓	✓	✓	✓	✓	✓	✓						

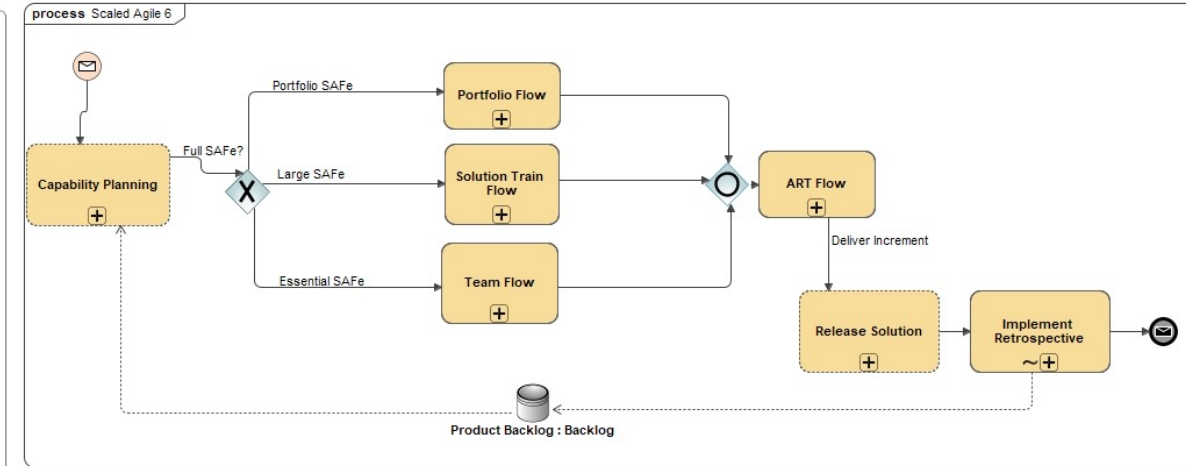
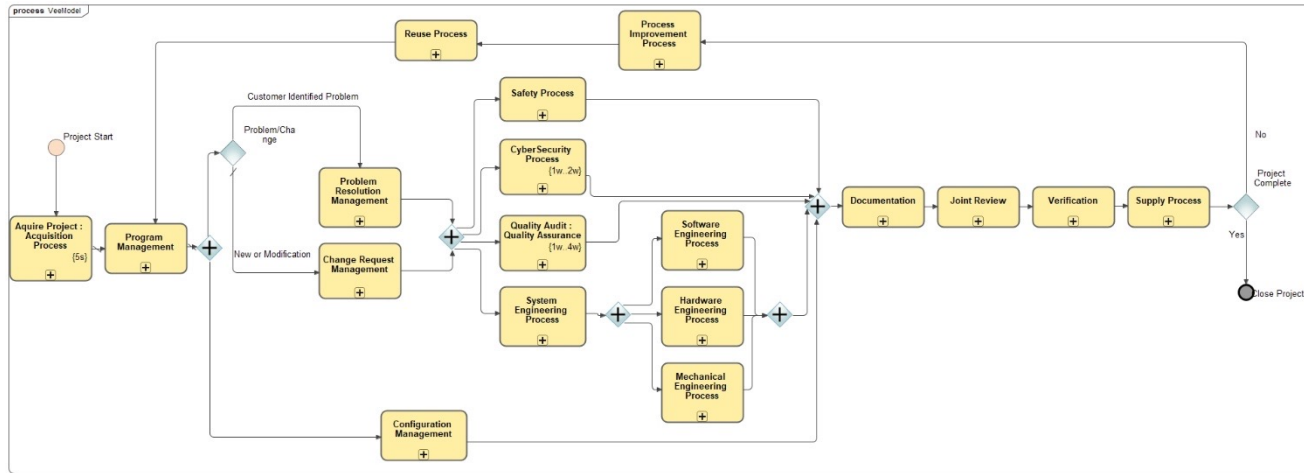
Modeling Lifecycles: Vee Model



Modeling Lifecycles: Agile

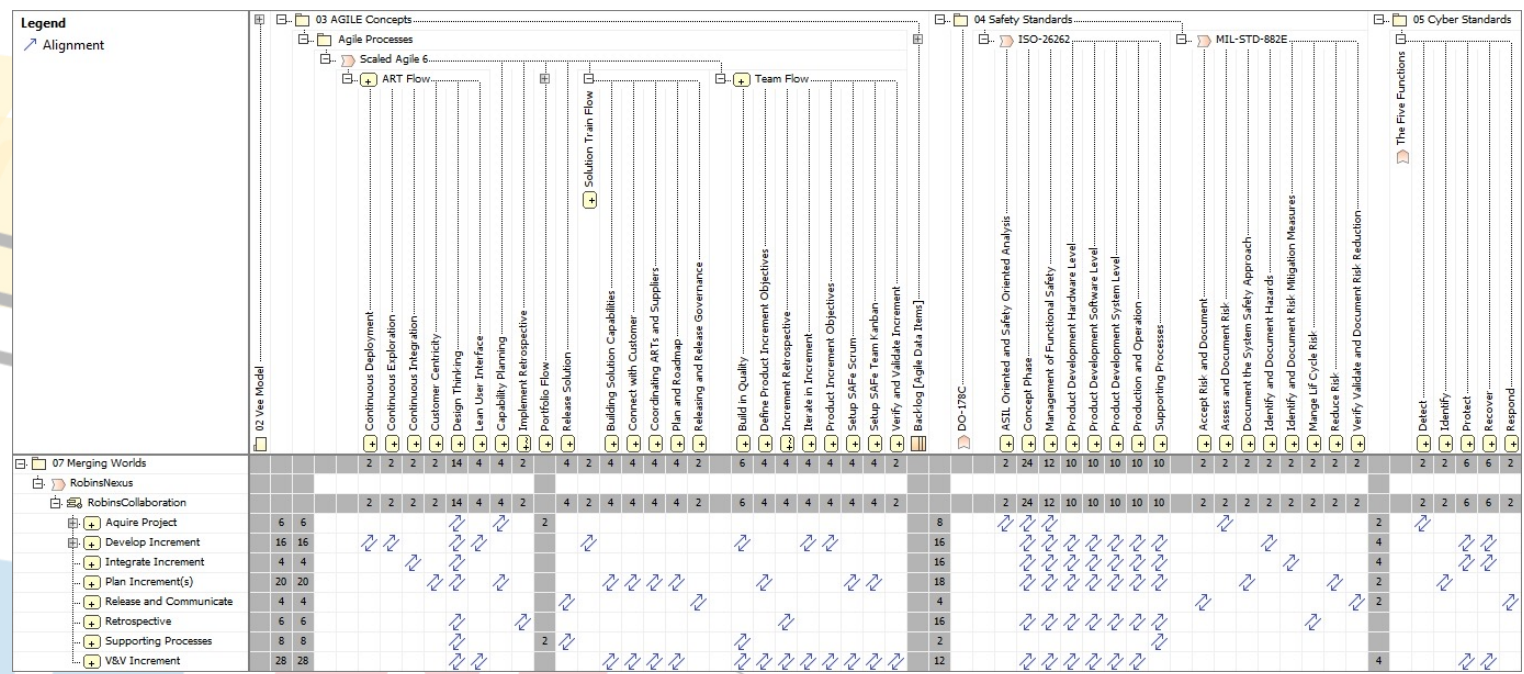
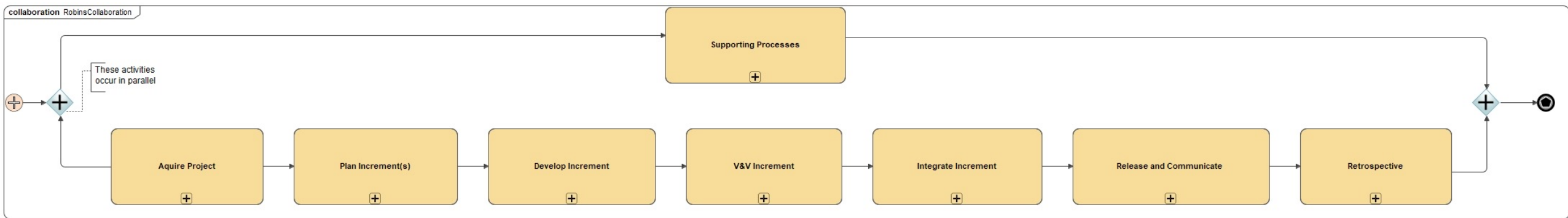


Comparing Lifecycles



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

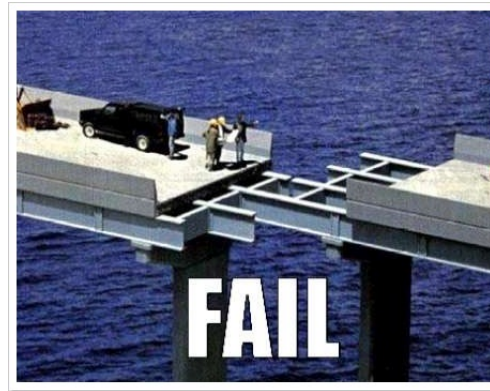
Why?

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/



This Photo by Unknown Author is licensed under [CC BY-ND](#)



This Photo by Unknown Author is licensed under [CC BY-SA-NC](#)



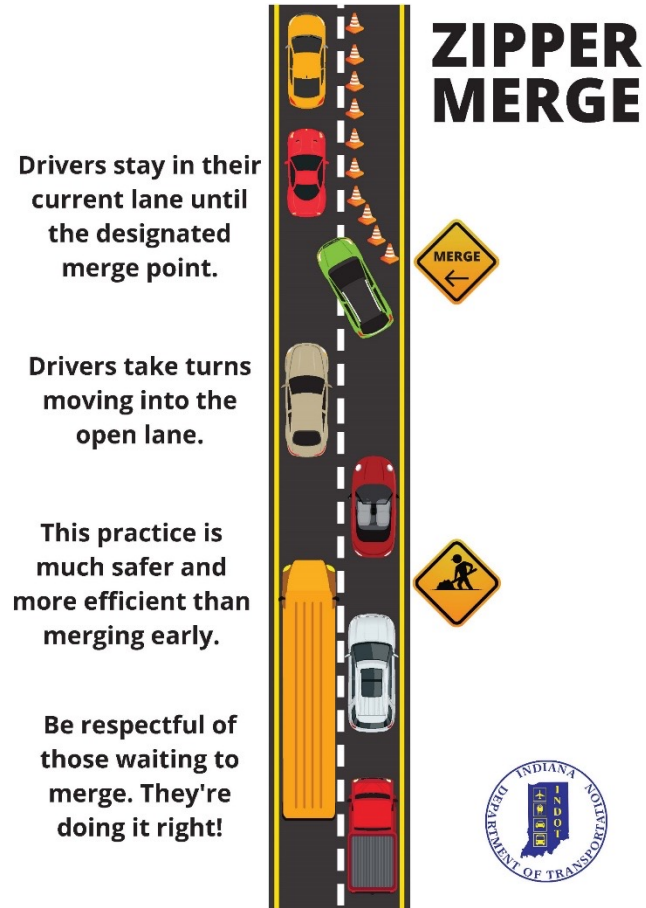
PLANNING

Still a good thing to do first.

VERY DEMOTIVATIONAL .com

<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.

HOW LONG CAN YOU WORK ON MAKING A ROUTINE TASK MORE EFFICIENT BEFORE YOU'RE SPENDING MORE TIME THAN YOU SAVE?
(ACROSS FIVE YEARS)

		HOW OFTEN YOU DO THE TASK					
		50/DAY	5/DAY	DAILY	WEEKLY	MONTHLY	YEARLY
HOW MUCH TIME YOU SHAVE OFF	1 SECOND	1 DAY	2 HOURS	30 MINUTES	4 MINUTES	1 MINUTE	5 SECONDS
	5 SECONDS	5 DAYS	12 HOURS	2 HOURS	21 MINUTES	5 MINUTES	25 SECONDS
	30 SECONDS	4 WEEKS	3 DAYS	12 HOURS	2 HOURS	30 MINUTES	2 MINUTES
	1 MINUTE	8 WEEKS	6 DAYS	1 DAY	4 HOURS	1 HOUR	5 MINUTES
	5 MINUTES	9 MONTHS	4 WEEKS	6 DAYS	21 HOURS	5 HOURS	25 MINUTES
	30 MINUTES		6 MONTHS	5 WEEKS	5 DAYS	1 DAY	2 HOURS
	1 HOUR		10 MONTHS	2 MONTHS	10 DAYS	2 DAYS	5 HOURS
	6 HOURS				2 MONTHS	2 WEEKS	1 DAY
	1 DAY					8 WEEKS	5 DAYS

<https://xkcd.com/1205/>

How does merging increase resilience and accountability?

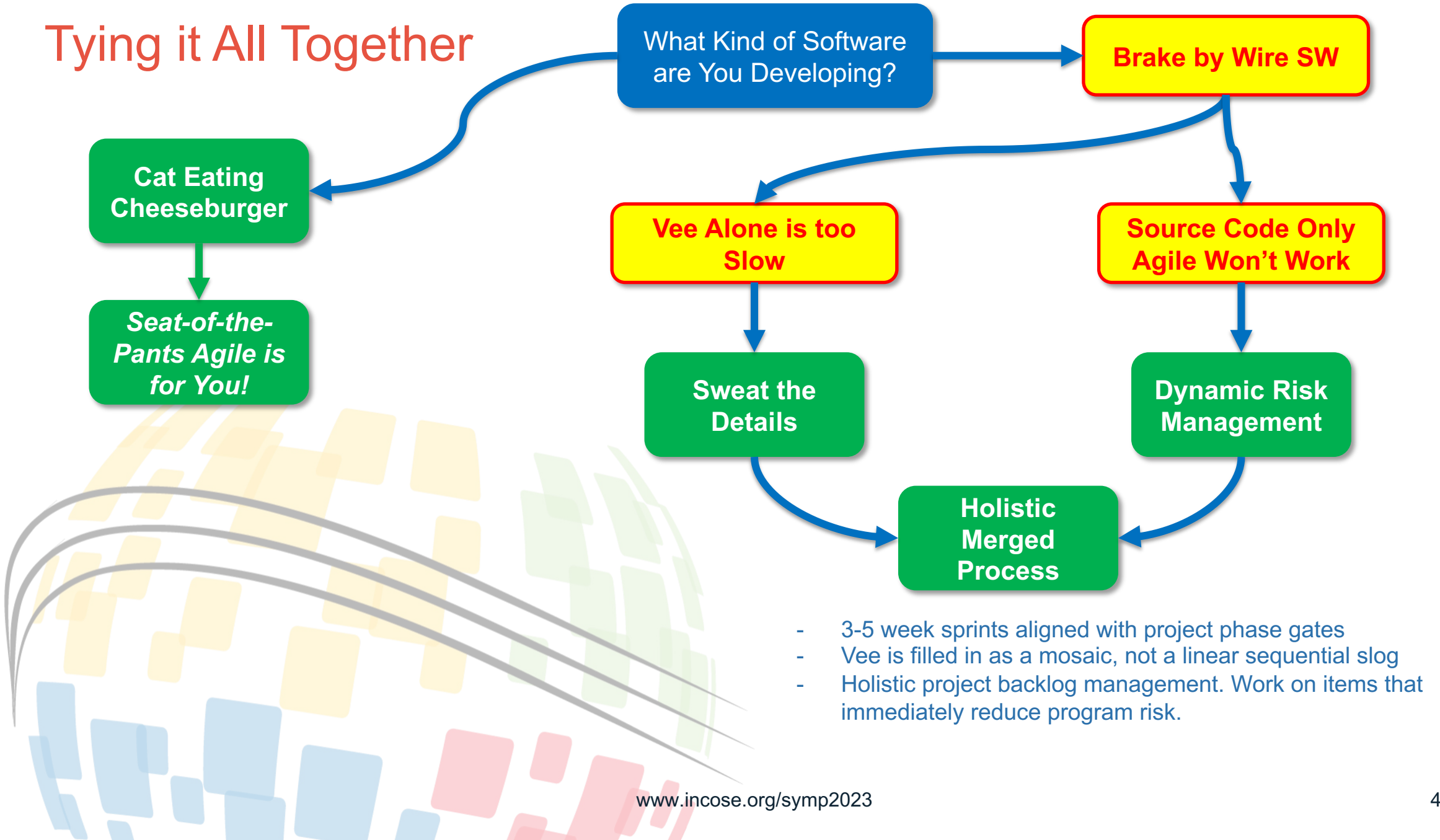
Resilience

- *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



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/l0lyjeA5mmMZjhyPm/giphy.gif>

Email Us!

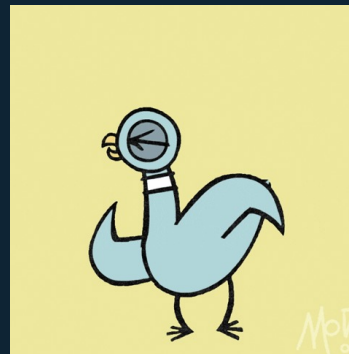




33rd Annual **INCOSE** international symposium

hybrid event

Honolulu HI USA



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