Building Large, Hardware-Reliant Systems with SAFe

Harry Koehnemann

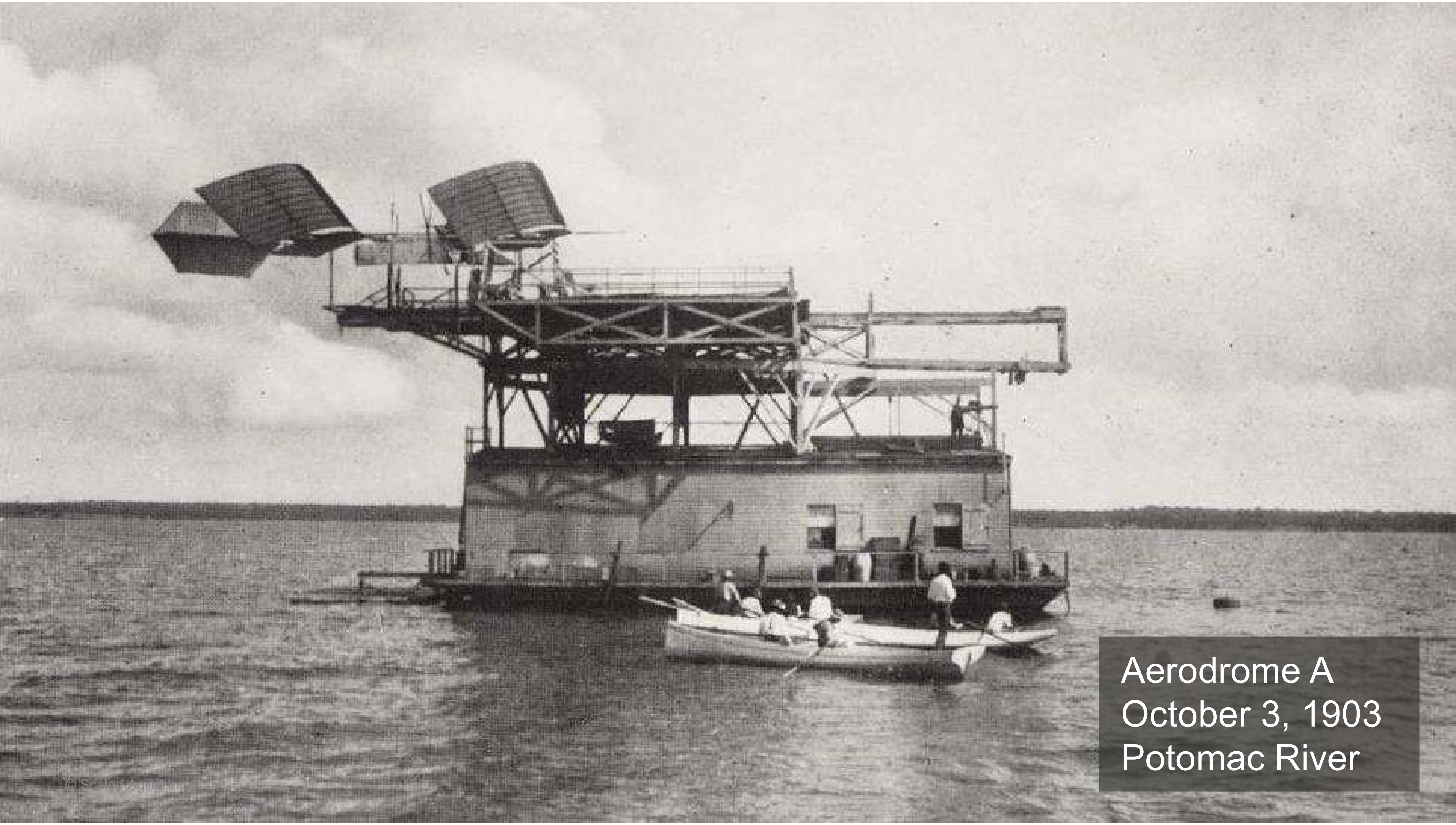
Methodologist and SAFe Fellow

Scaled Agile, Inc.

Agenda

- Why change to Lean-Agile development?
- Thriving in the Digital Age for big system builders
- Apply Lean-Agile practices to large, hardware-reliant systems
  - Organize around value
  - Specify the system incrementally
  - Apply multiple planning horizons
  - Design for change
  - Frequently integrate the end-to-end solution
  - Manage the supply chain
  - Continually address compliance concerns
  - Shift learning left
  - Move to Lean-Agile Management

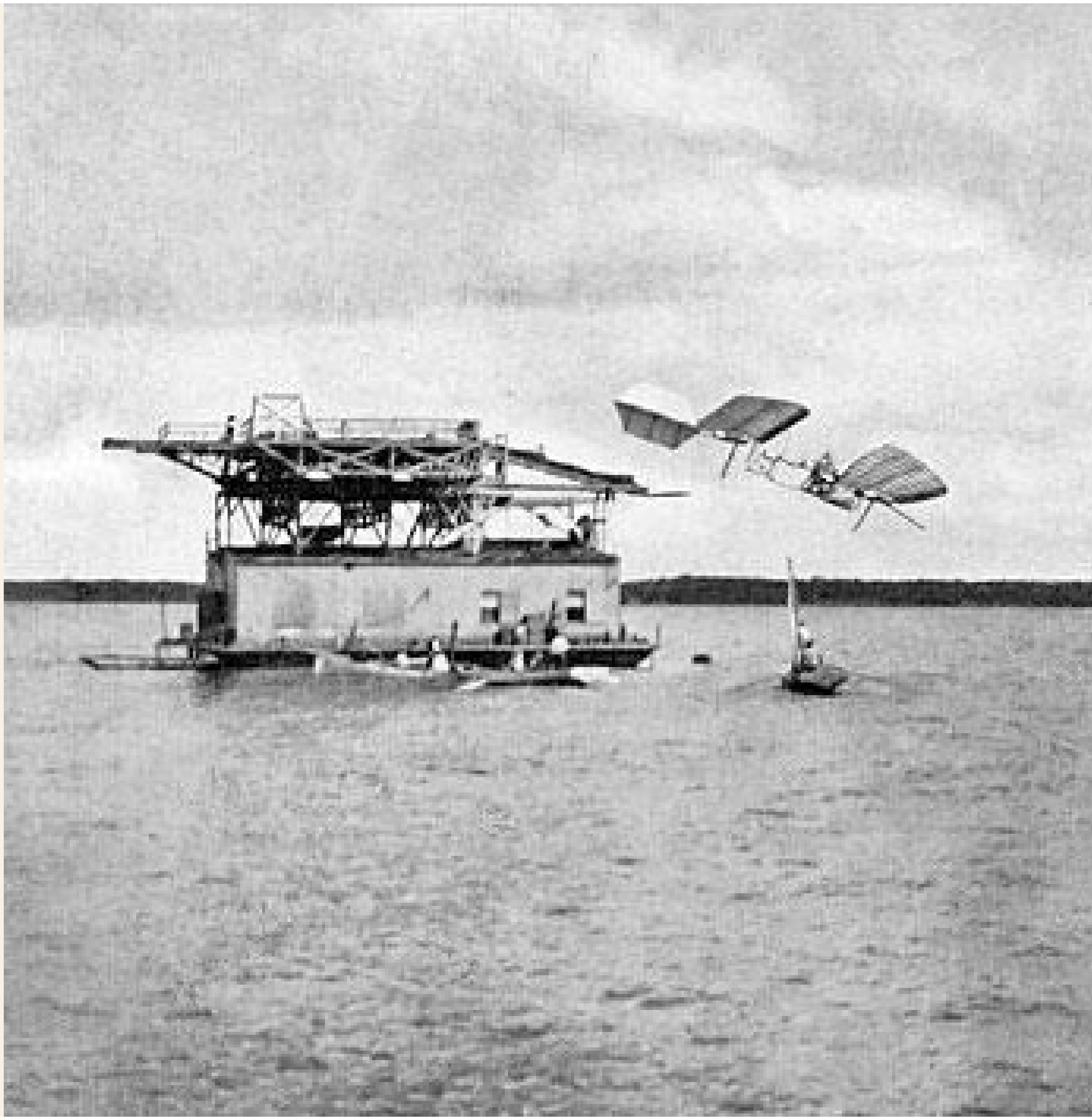
Why change to Lean-Agile development?

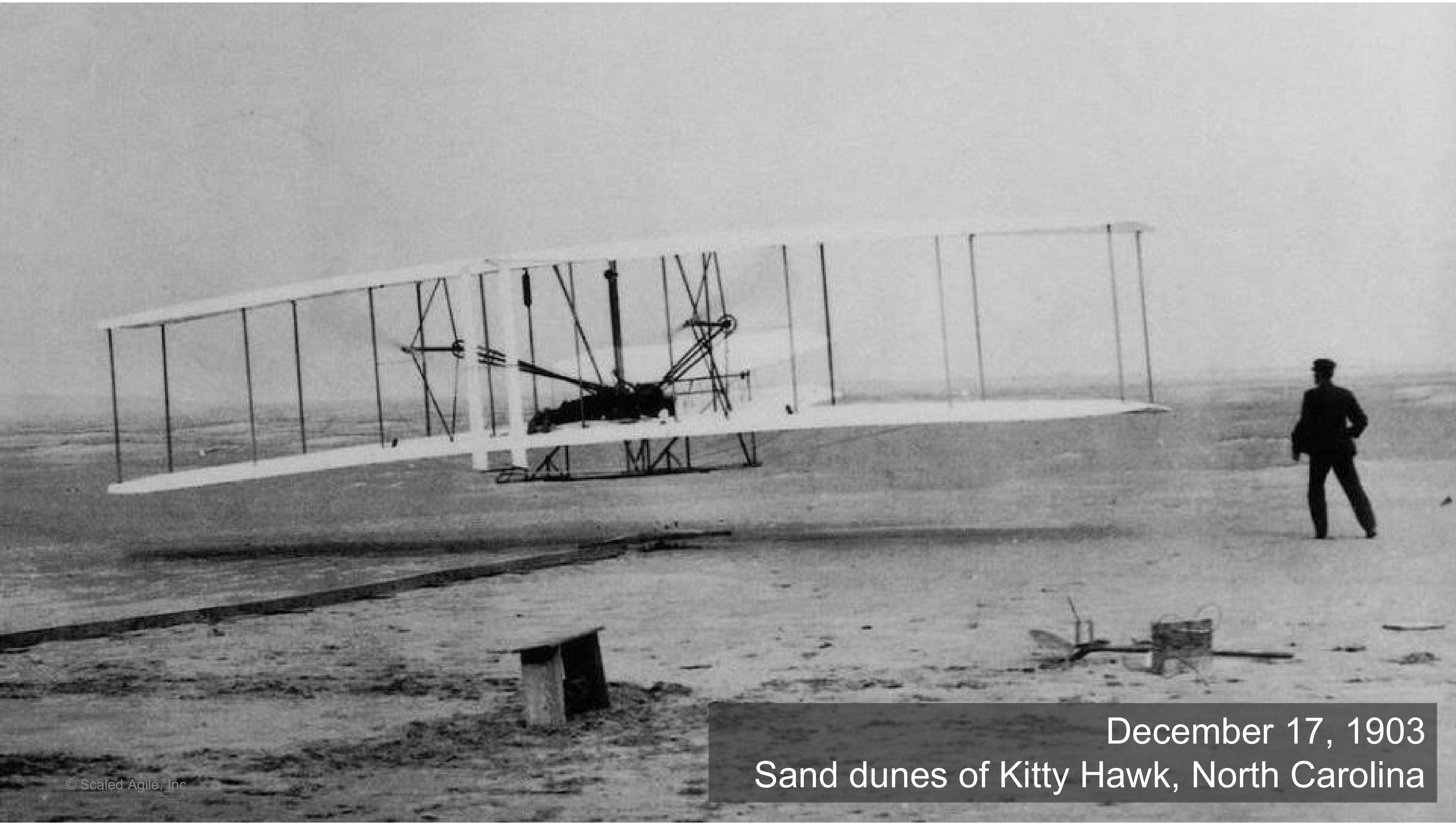


Aerodrome A
October 3, 1903
Potomac River

Samuel Langley

- Well-funded: $50,000
- Designed a single-point flying machine
- Never flew due to fundamental design flaws





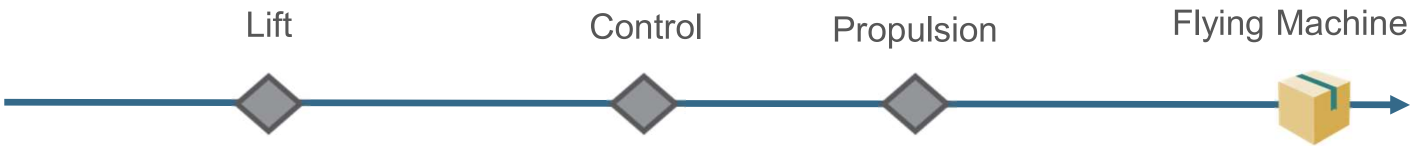
December 17, 1903
Sand dunes of Kitty Hawk, North Carolina

Wilbur & Orville Wright

- Spent less than $1,000 US
- Iteratively learned about barriers to flight
- Rapid experiments with home-built wind tunnel
- Created the first flying machine

The Wright Brothers applied incremental learning

Establish learning milestones



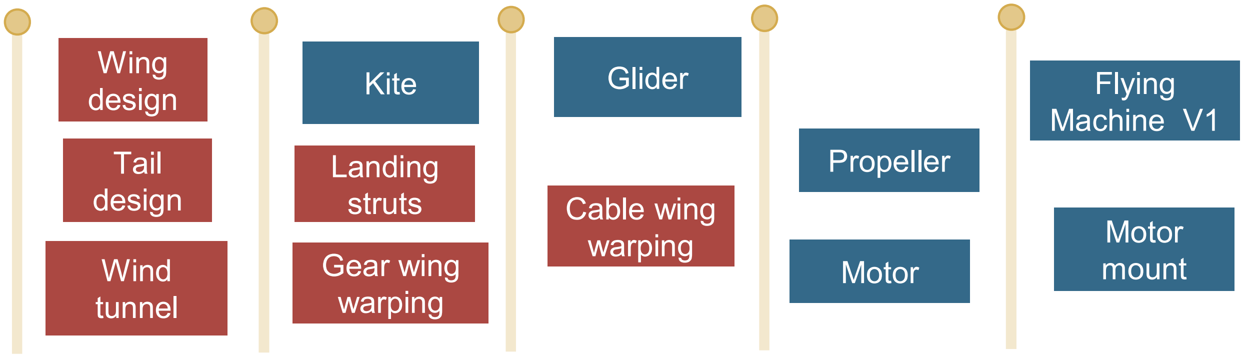
Lift

Control

Propulsion

Flying Machine

Create backlog of work needed to gain knowledge



Wing design

Tail design

Wind tunnel

Kite

Landing struts

Gear wing warping

Glider

Cable wing warping

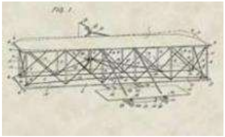
Propeller

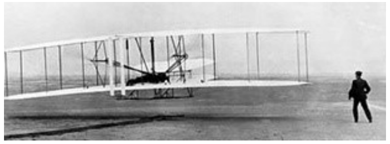
Motor

Flying Machine V1

Motor mount

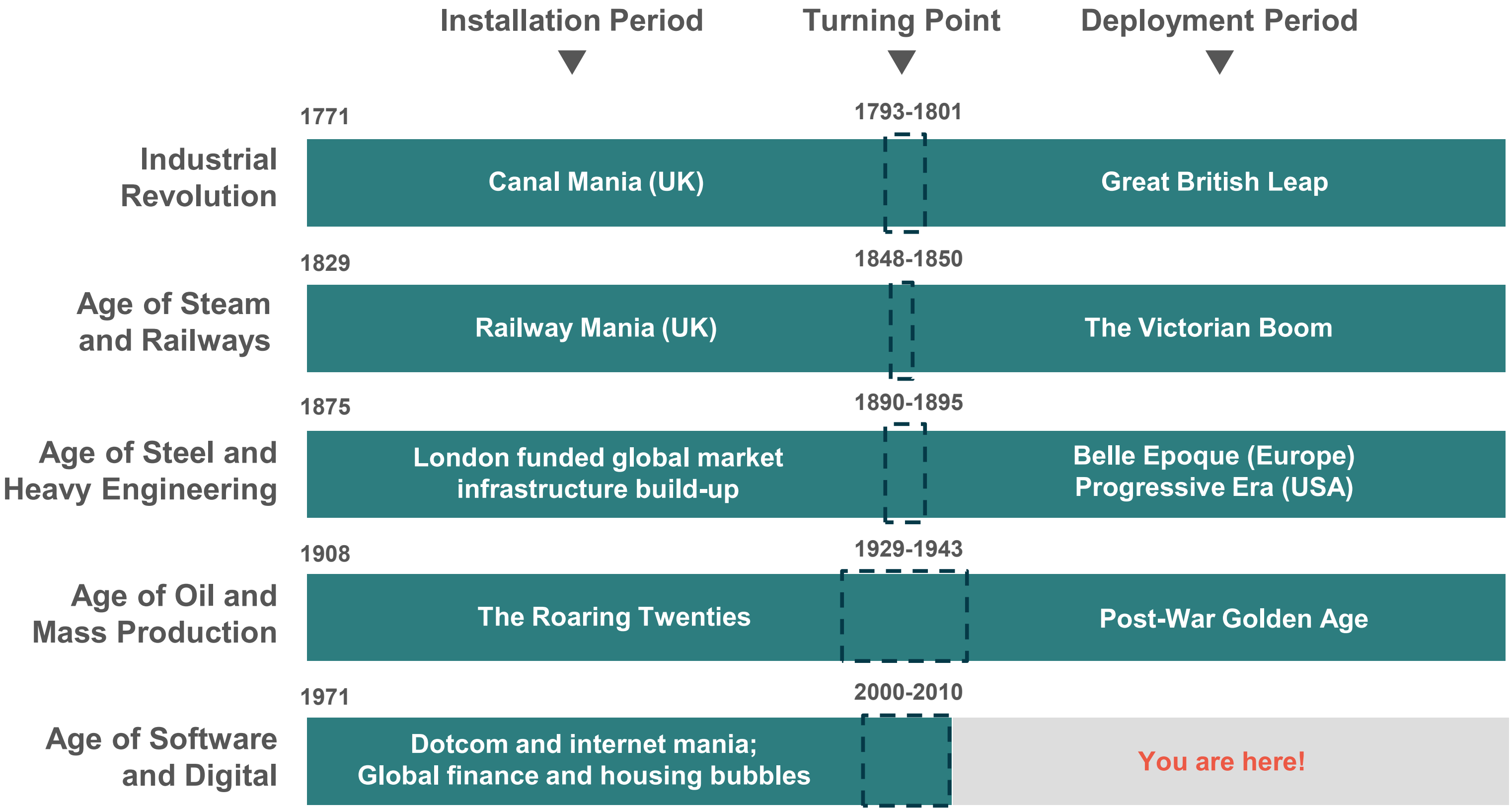
Evolve the solution based on learning





Thriving in the Digital Age

Technological revolutions periodically create a new economic order



	Installation Period	Turning Point	Deployment Period
Industrial Revolution	1771 Canal Mania (UK)	1793-1801	Great British Leap
Age of Steam and Railways	1829 Railway Mania (UK)	1848-1850	The Victorian Boom
Age of Steel and Heavy Engineering	1875 London funded global market infrastructure build-up	1890-1895	Belle Epoque (Europe) Progressive Era (USA)
Age of Oil and Mass Production	1908 The Roaring Twenties	1929-1943	Post-War Golden Age
Age of Software and Digital	1971 Dotcom and internet mania; Global finance and housing bubbles	2000-2010	You are here!

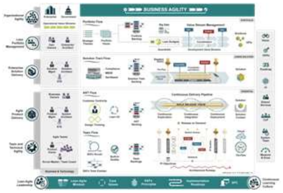
Source. The five technological revolutions, adapted from *Technological Revolutions and Financial Capital* by Carlota Perez (2002).

…and new methods for managing work and people

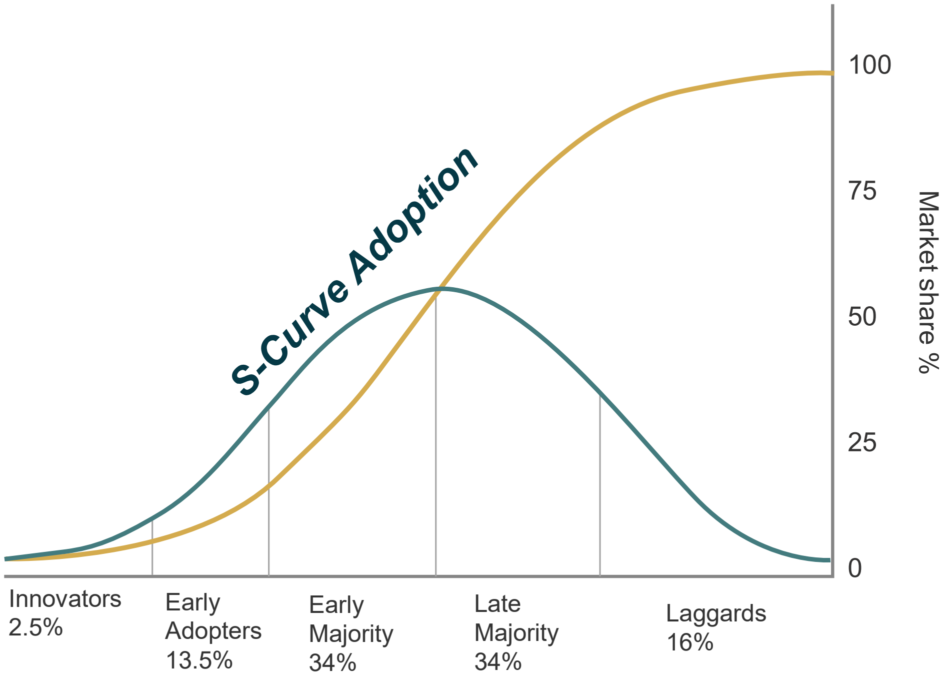
		Installation Period	Turning Point	Deployment Period	Management Method
Industrial Revolution	1771	Canal Mania (UK)	1793-1801	Great British Leap	Factory Systems
Age of Steam and Railways	1829	Railway Mania (UK)	1848-1850	The Victorian Boom	Subcontracting
Age of Steel and Heavy Engineering	1875	London funded global market infrastructure build-up	1890-1895	Belle Epoque (Europe) Progressive Era (USA)	Taylorism
Age of Oil and Mass Production	1908	The Roaring Twenties	1929-1943	Post-War Golden Age	Fordism
Age of Software and Digital	1971	Dotcom and internet mania; Global finance and housing bubbles	2000-2010	You are here!	Continuous Learning (Lean-Agile)?

Project Management

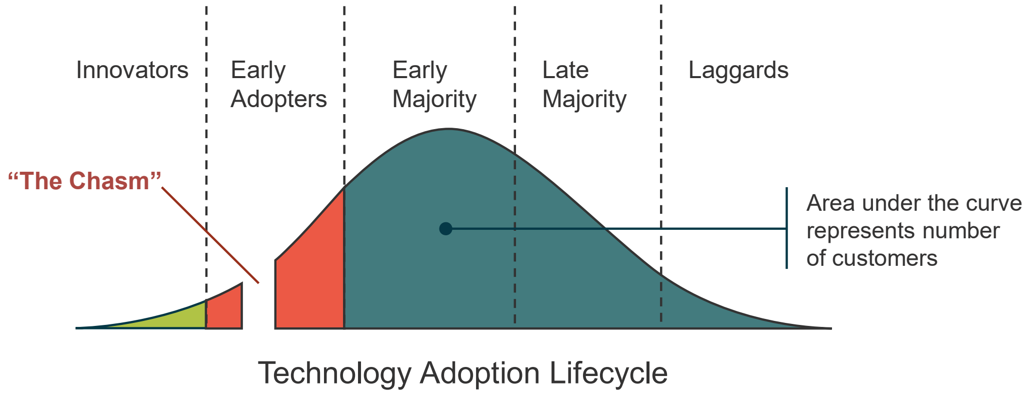
Source. The five technological revolutions, adapted from *Technological Revolutions and Financial Capital* by Carlota Perez (2002).



Diffusions of Innovation and Crossing the Chasm

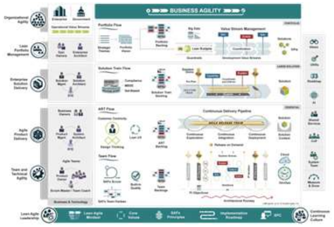


Diffusions of Innovation

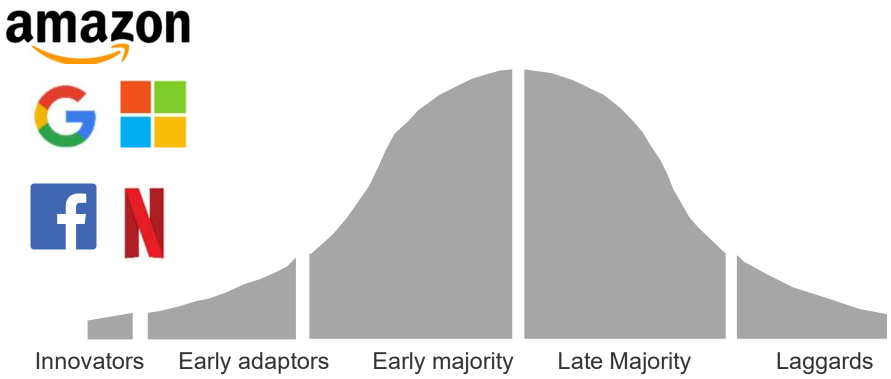


Crossing the Chasm

SAFe helps larger enterprises achieve software agility

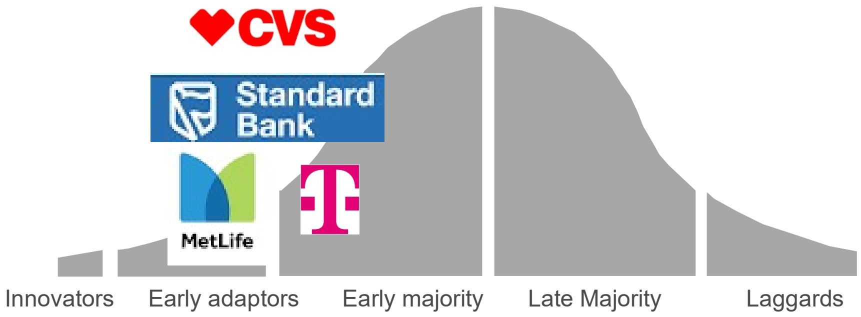


Agile adoption (circa 2000-2010)

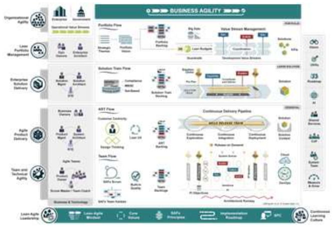


Agile adoption (circa 2010-2020)

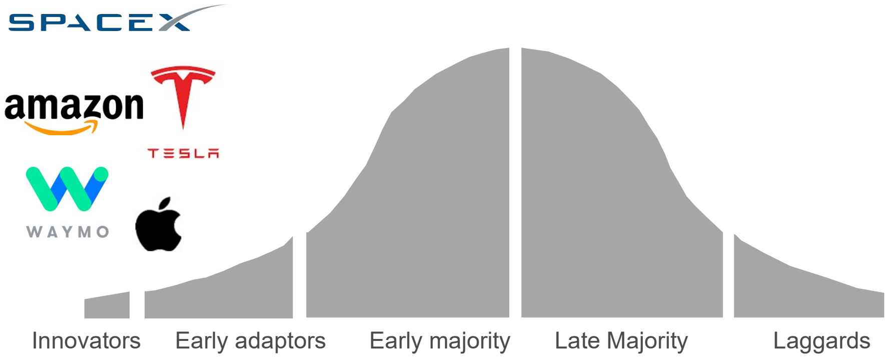
75% of the Global 2000



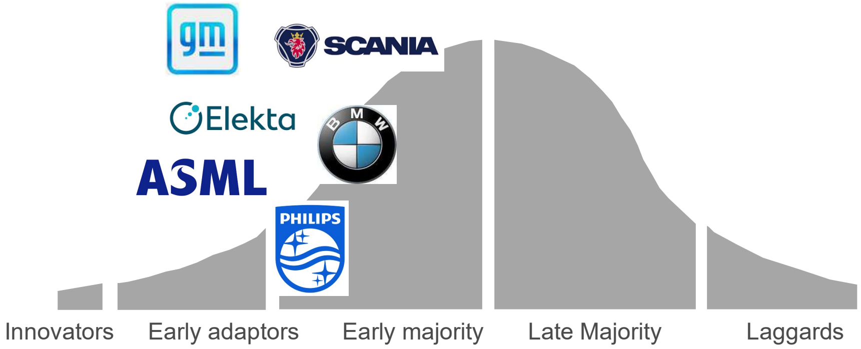
Agile is penetrating other industries



Agile hardware (circa 2015-2021)



Agile hardware (today)



Digital Age early adopter characteristics

Cultural

- Inquisitive, growth mindset
- Create passion to provide amazing solutions for customers
- Delegate decisions (with guardrails)
- Teaming, working without boundaries
- Learning organization, growing T-skills
- Permission to fail

Process / Technical

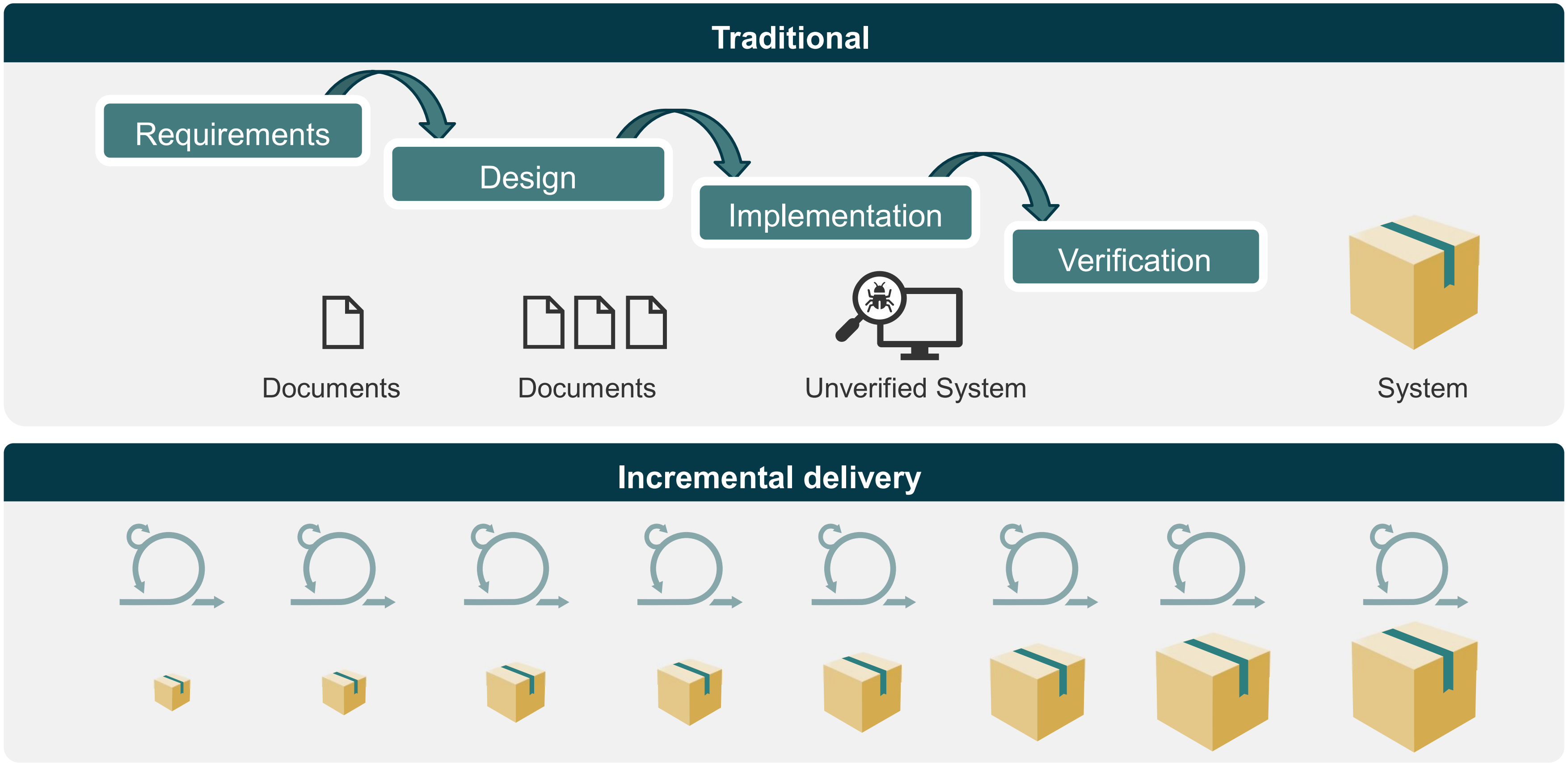
- Deliver fast (MVP), get feedback, adjust
- Architect systems for fast change
- Leverage virtualization
- Invest in the ‘machine that builds the machine’ (CI/CD pipeline)

Scaling Agile for Hardware-Reliant Solutions

1. Organize around value
2. Specify the system incrementally
3. Apply multiple planning horizons
4. Design for change
5. Frequently integrate the end-to-end solution
6. Manage the supply chain
7. Continually address compliance concerns
8. Shift learning left
9. Move to Lean-Agile management

Organize Around Value

Agile product development goal: Deliver quick for fast feedback



Traditional

Requirements

Design

Implementation

Verification

Documents

Documents

Unverified System

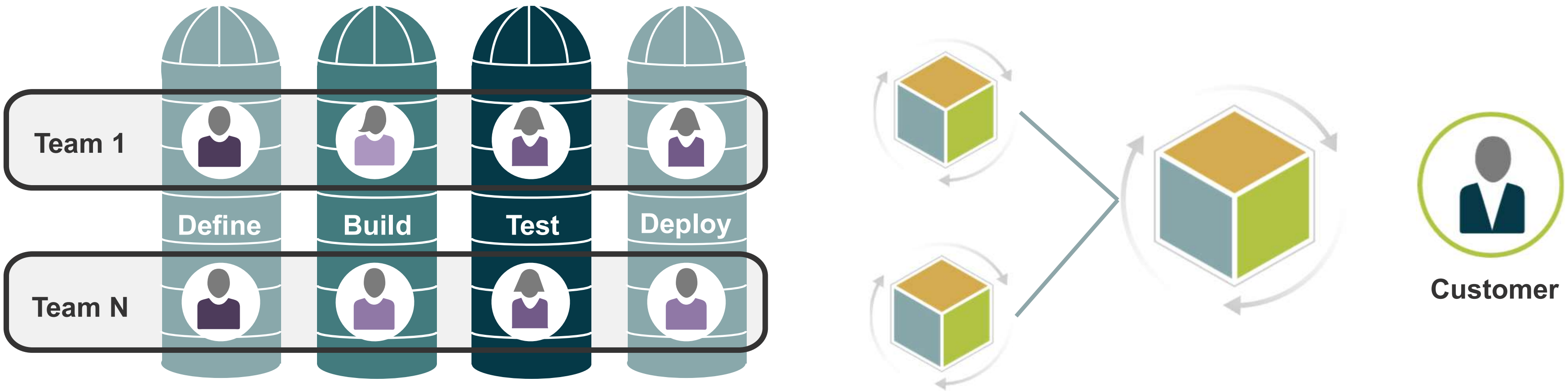
System

Incremental delivery

Agile Teams are optimized to deliver quickly

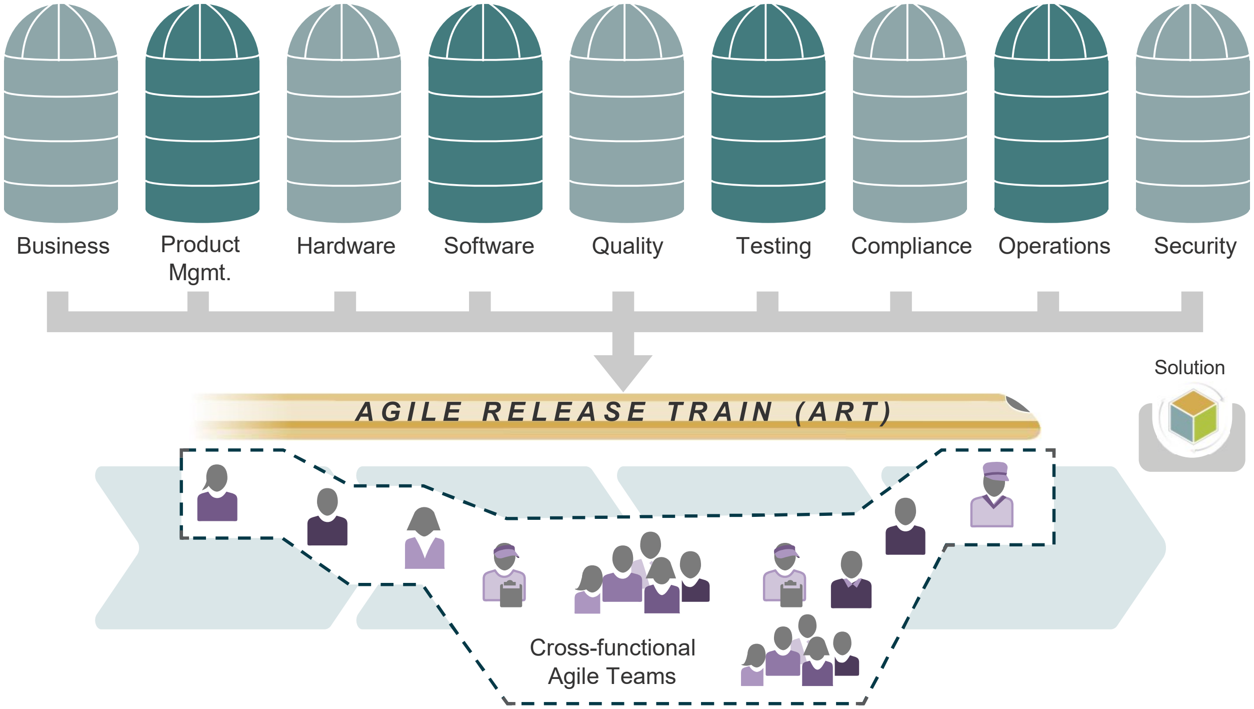
Agile Teams are cross-functional, self-organizing entities that can define, build, test, and where applicable, deploy increments of value.

- All requisite skills and authority
- Well-known set of roles, events, artifacts, and practices

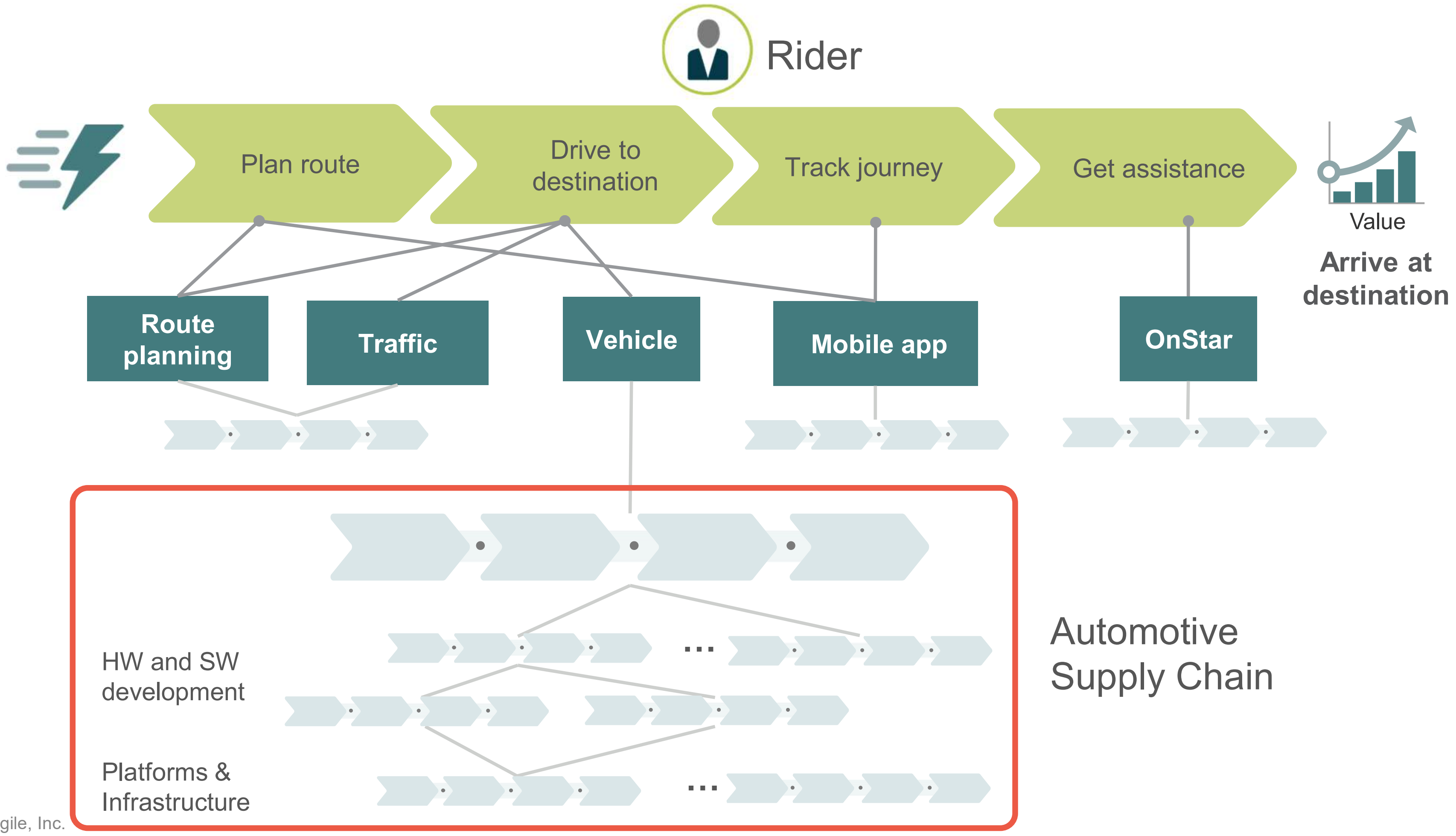


Bigger systems require a team-of-Agile teams

- Contain all the skills and authority necessary to deliver a solution
- Scales the well-known set of roles, events, artifacts, and practices



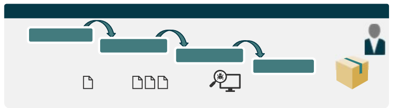
Big systems are built by a network of DVSs



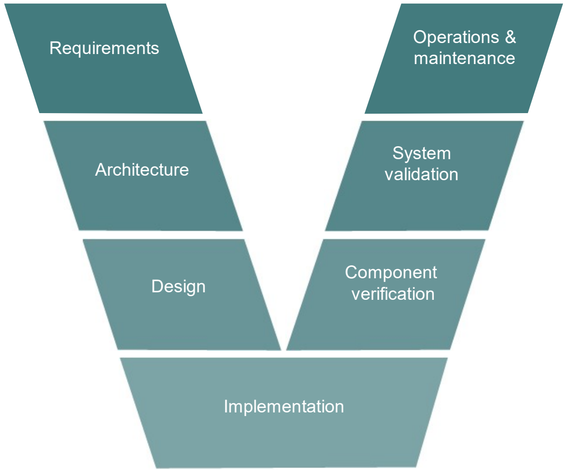
Automotive Supply Chain

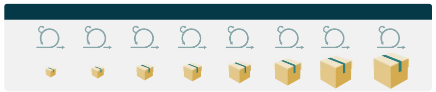
Specify the System Incrementally

Lower the specification batch size



Traditional ‘V’





Continuous Flow

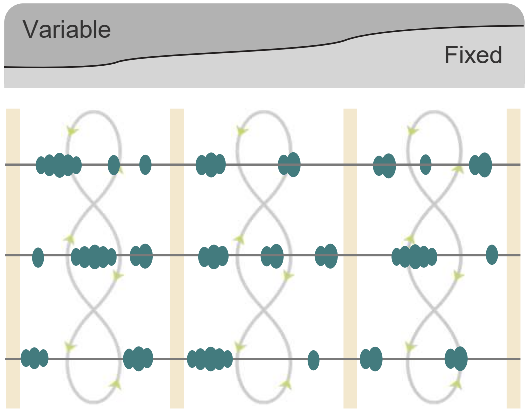
- *100 Shall statements*
- *Many questions*

- *1000s of Shall statements*
- *Many decisions*

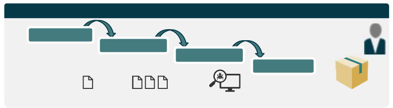
Evolve the Solution Intent

Manage Capabilities and Features

Adjust Roadmaps

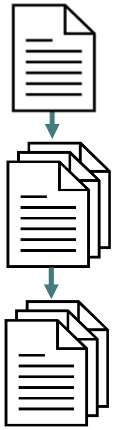


Replace detailed specifications with backlogs and roadmaps



Traditional requirements

Provide no opportunity to adjust based on feedback



System requirements

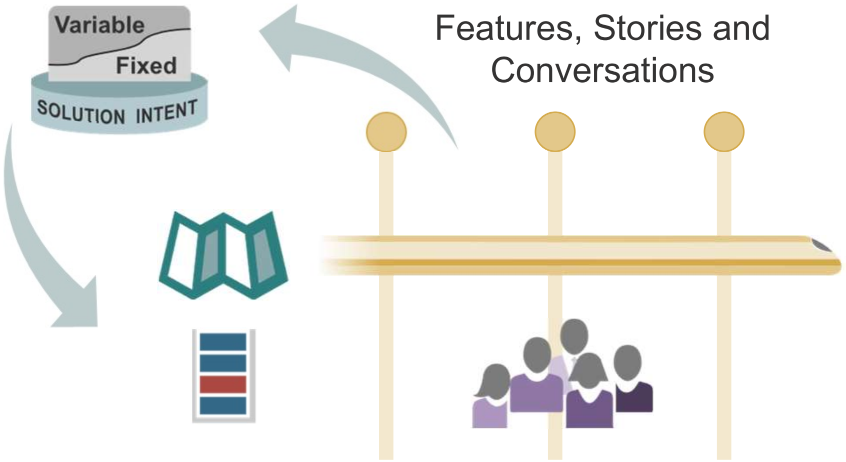
Subsystem requirements

Component requirements

Lean-agile requirements

Evolve continuously from variable to fixed based on learning

System requirements

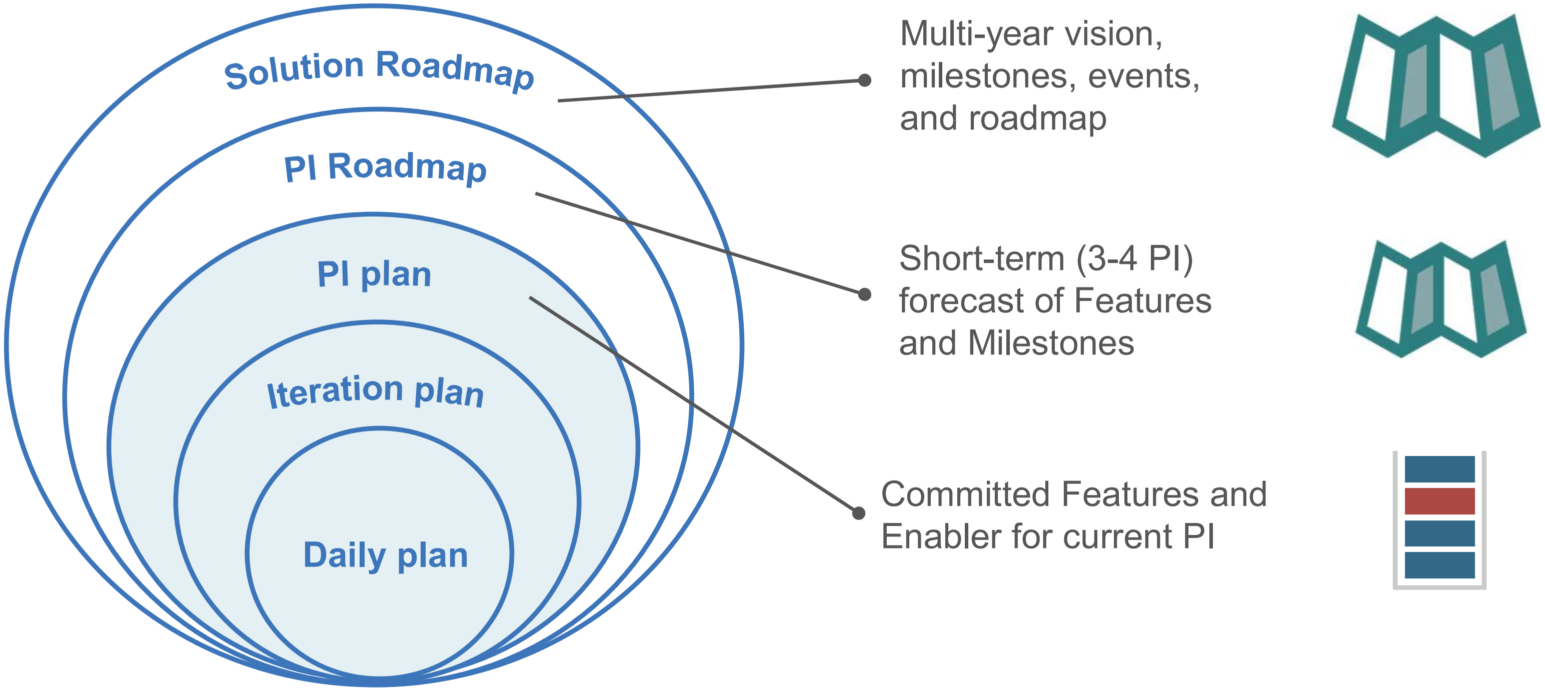


Features, Stories and Conversations

Apply Multiple Planning Horizons

Planning Occurs at Multiple Levels

- Roadmaps replace fixed schedule with forecasts for planning and adjustment



Solution Roadmap

PI Roadmap

PI plan

Iteration plan

Daily plan

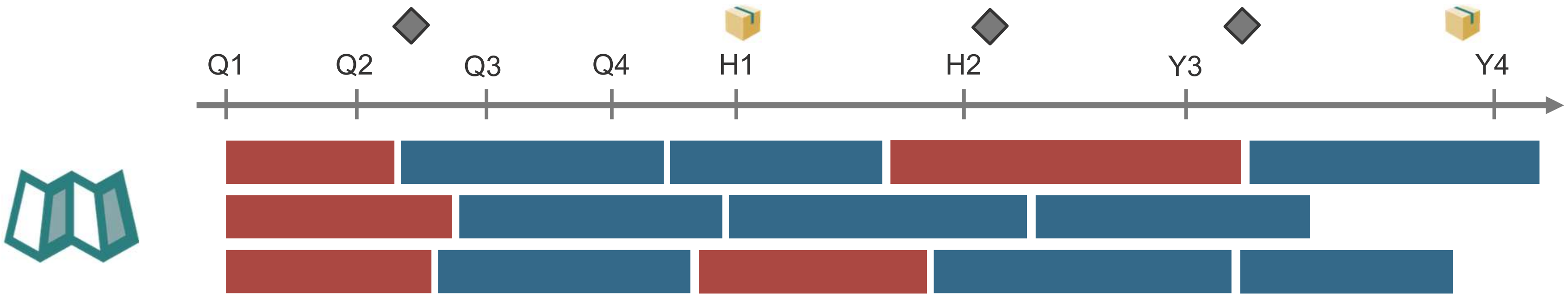
Multi-year vision, milestones, events, and roadmap

Short-term (3-4 PI) forecast of Features and Milestones

Committed Features and Enabler for current PI

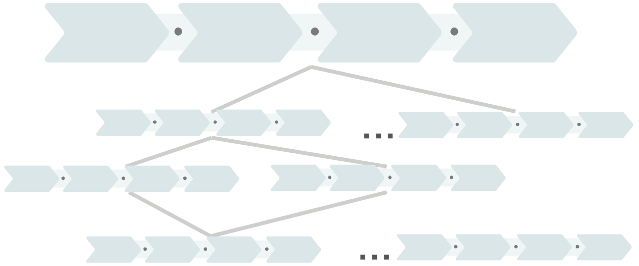
Use Solution Roadmap to forecast work and milestones

- Shows Epics sequenced over time
- Depicts highly-visible milestones and releases
- Describes a forecast, not a commitment

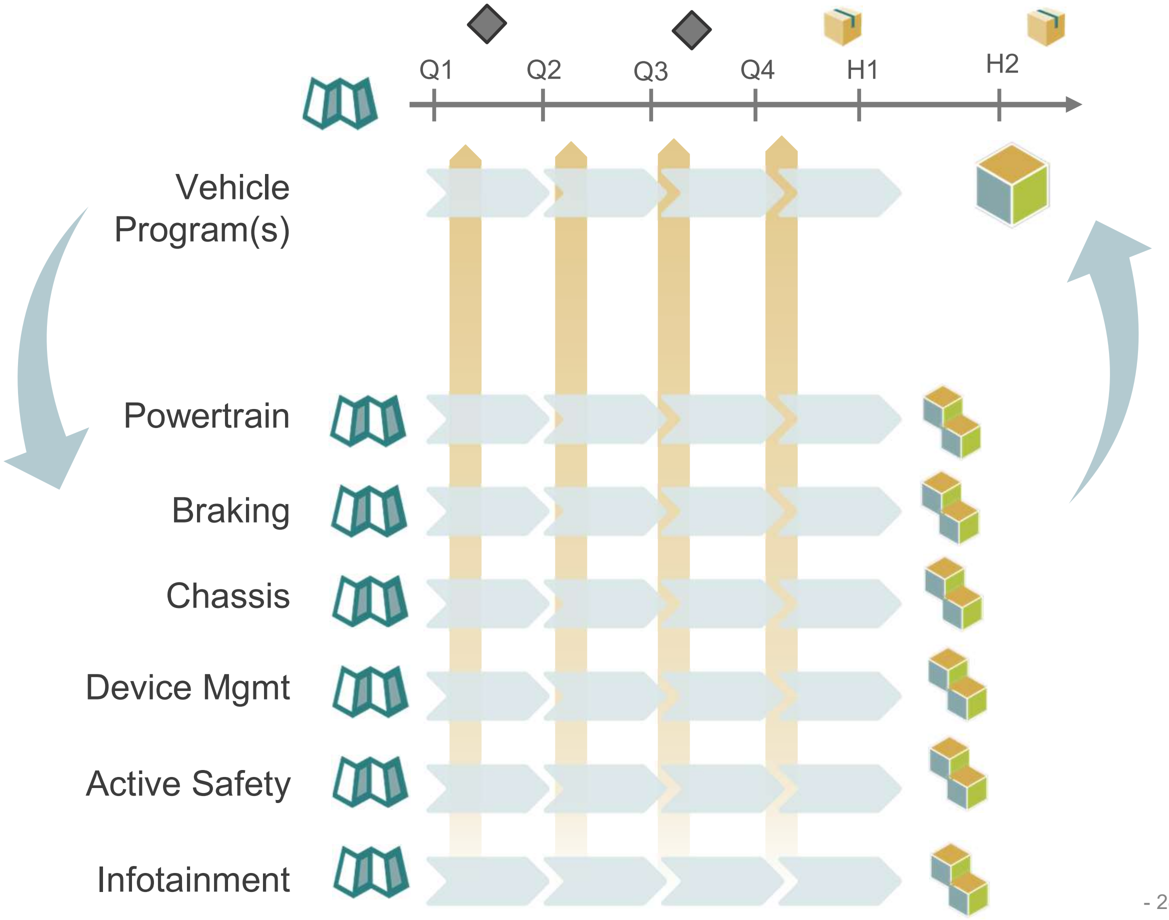


© Scaled Agile, Inc.

Automotive alignment example



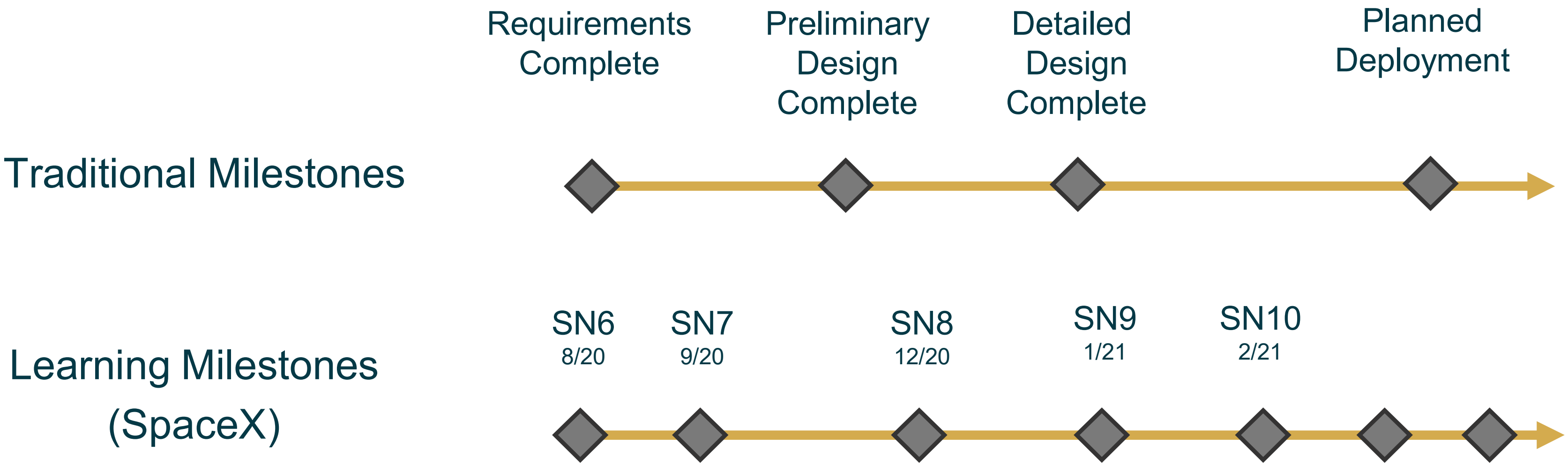
Automotive supply chain



Base milestones on objective evaluation of working systems

Development is more dependent on what needs to be learned than on what tasks must be completed to exit a gate.

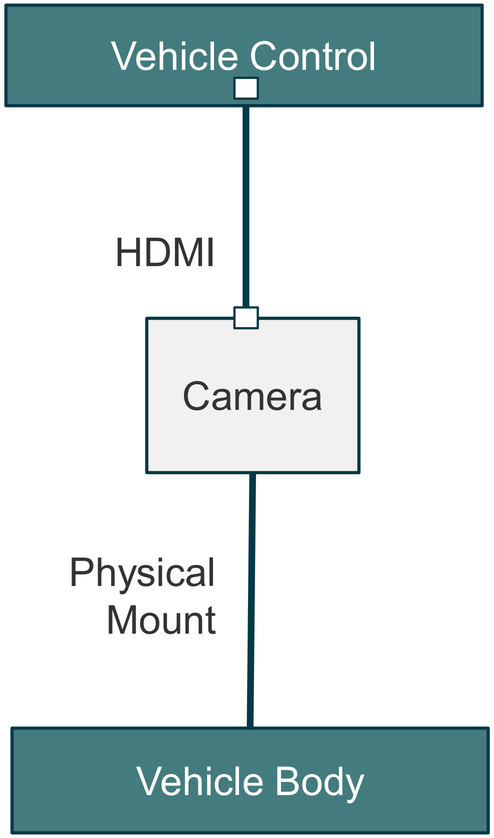
— Allan Ward, Lean Product and Process Development



Design for Change

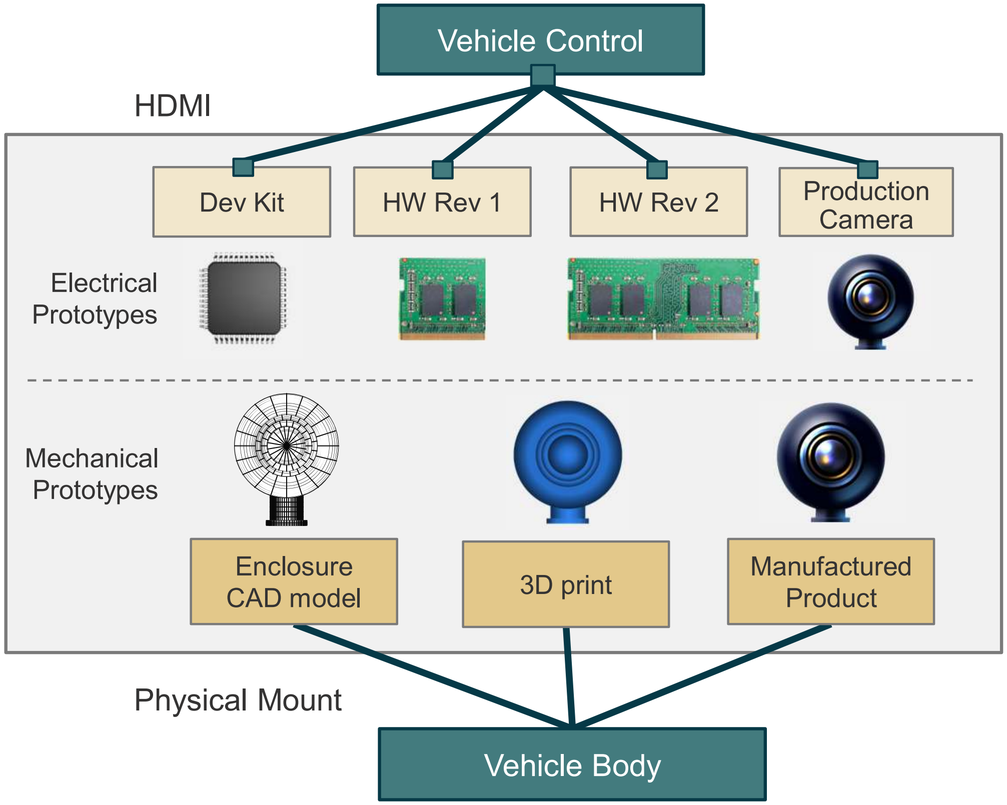
Modular designs with defined interfaces support efficient change

- To evolve designs, define interfaces first
- Interfaces include software APIs, signals, and physical connections
- Interfaces accelerate changes in both development and production environments
- Interfaces enable set-based design



Interfaces enable frequent, independent design iterations

- Allow teams to independently evolve their designs
- Support exploration of independent design sets (SBD)
- Enable frequent integration

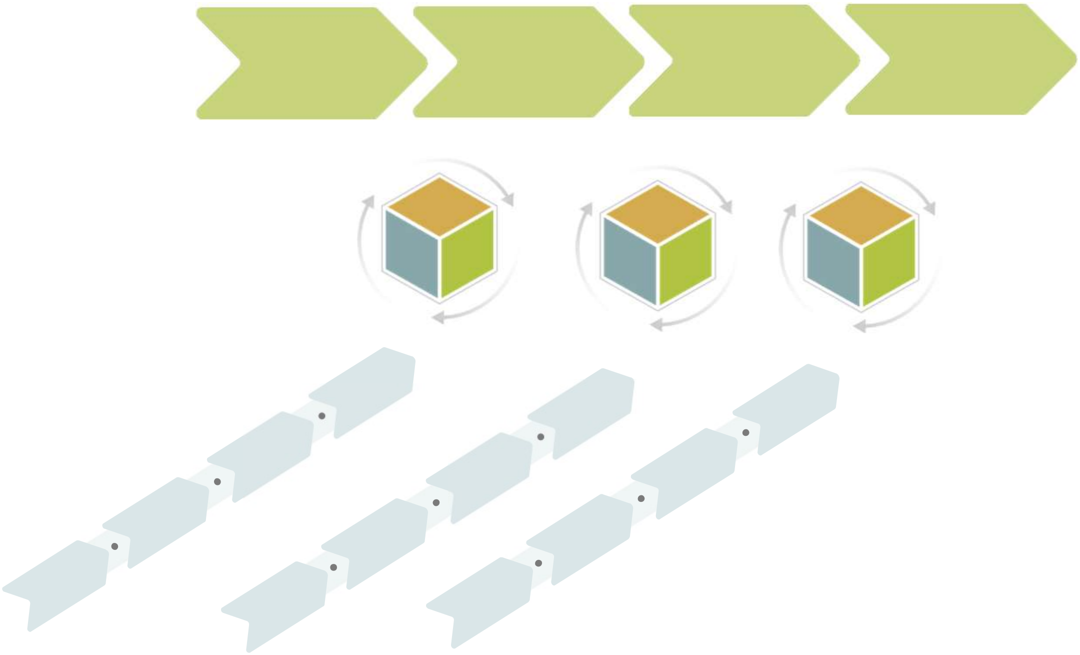


Design decisions must balance ALL costs

Unit, manufacturing, and other operational costs

Optimal Solution Design

Cost of delayed delivery and feedback



Ensure design decisions include the user and business value for costs of delay and total cost of ownership

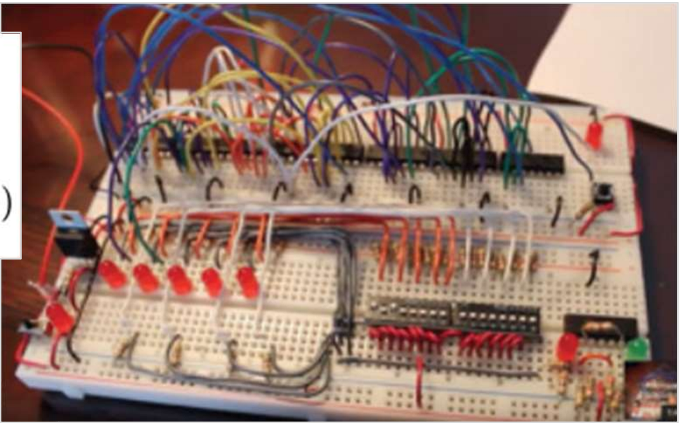
Frequently Integrate the End-to-End Solution

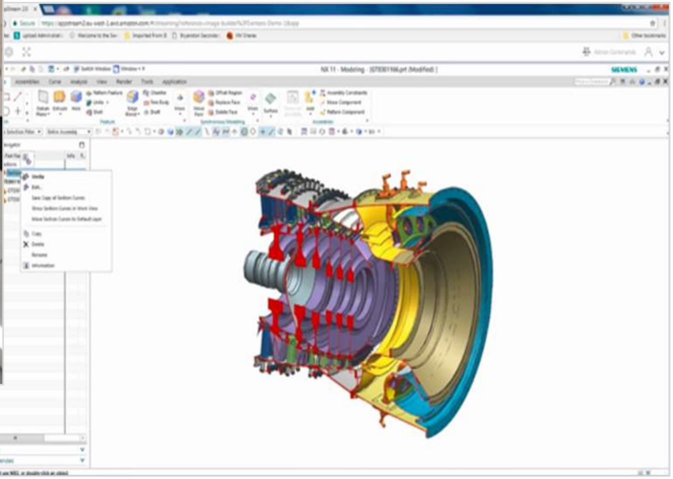
Hardware domains experiment during development

- Provides knowledge and feedback earlier in product lifecycle
- Mitigates risk by validating assumptions sooner

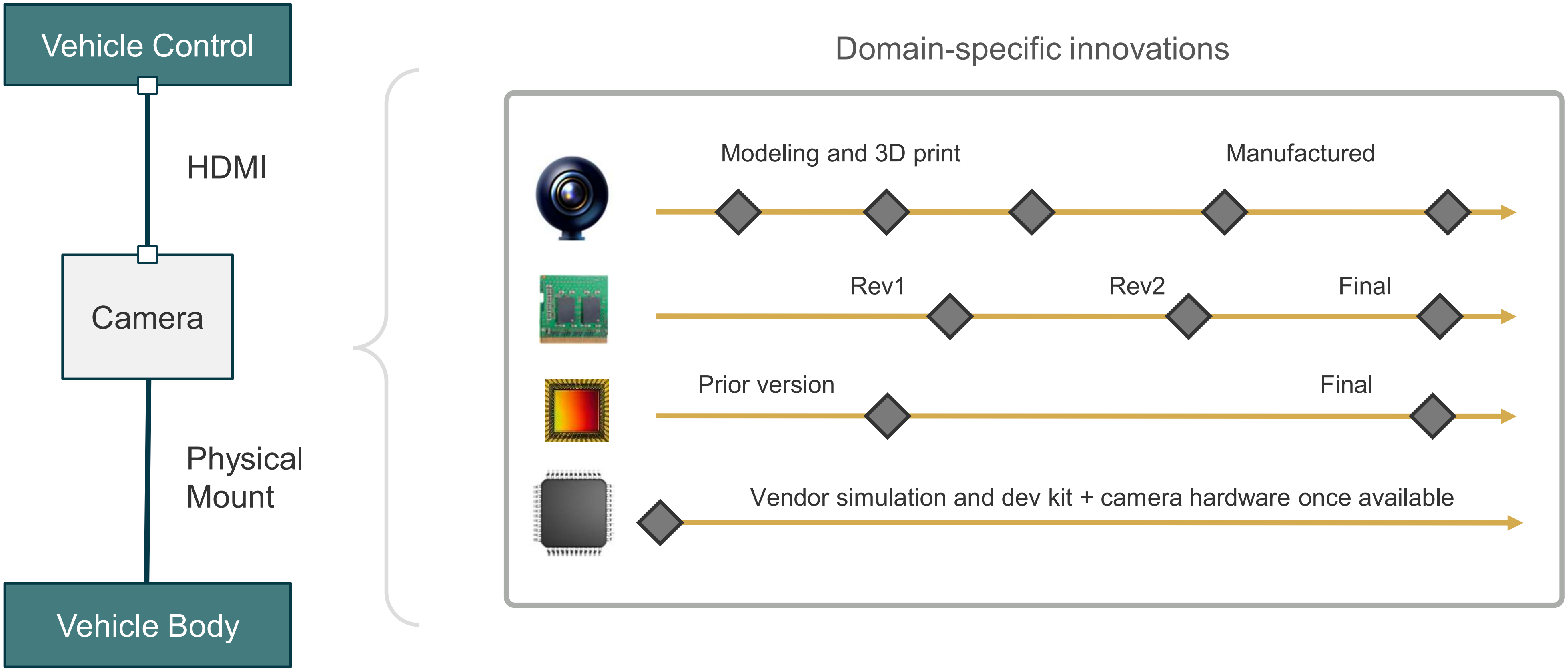
$$\hat{H} = \sum_{n=1}^{N} \frac{\hat{p}_n^2}{2m_n} + V(x_1, x_2, \cdots x_N)$$

$$= -\frac{\hbar^2}{2} \sum_{n=1}^{N} \frac{1}{m_n} \frac{\partial^2}{\partial x_n^2} + V(x_1, x_2, \cdots x_N)$$





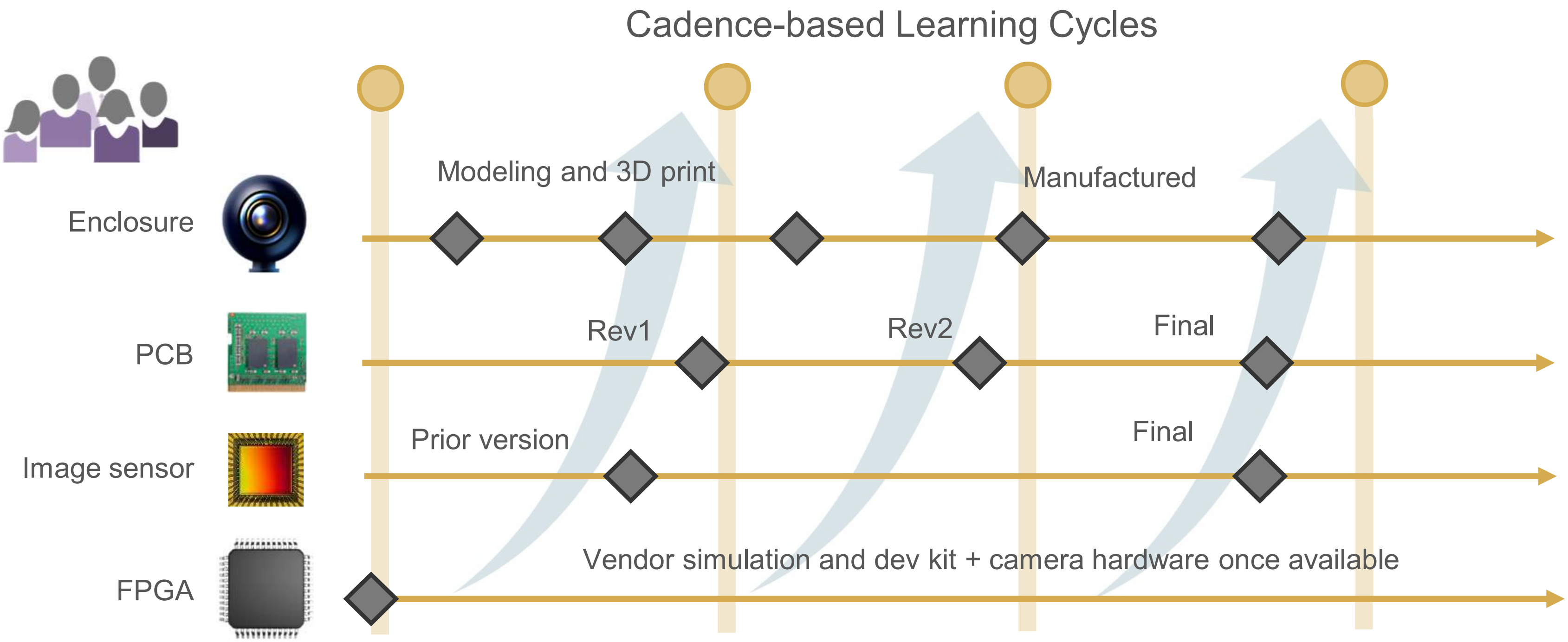
Learning is often done in a local context



Ensure that the entire system is learning

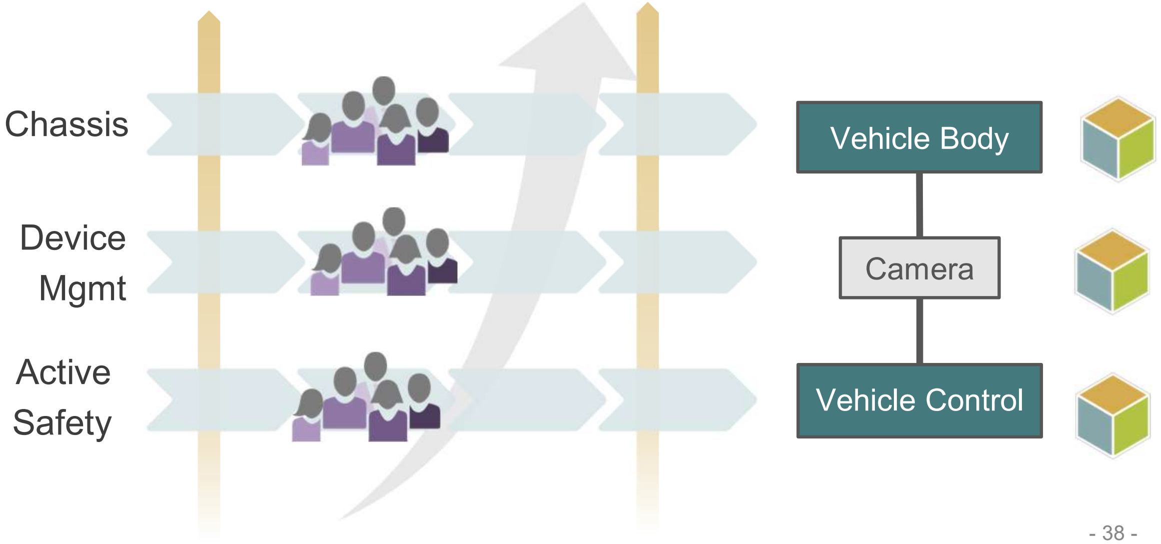
“Integration points control product development and are the leverage points to improve the system. When timing of integration points slips, the project is in trouble.”

— Dantar P. Oosterwal

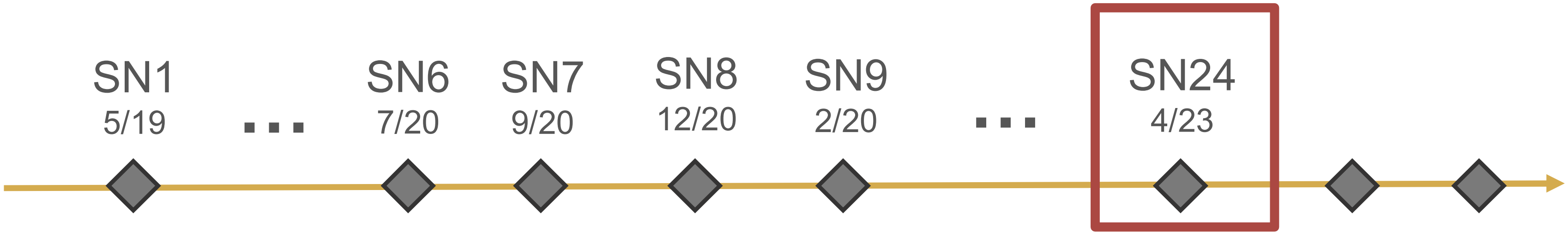


Continuously integrate the end-to-end system

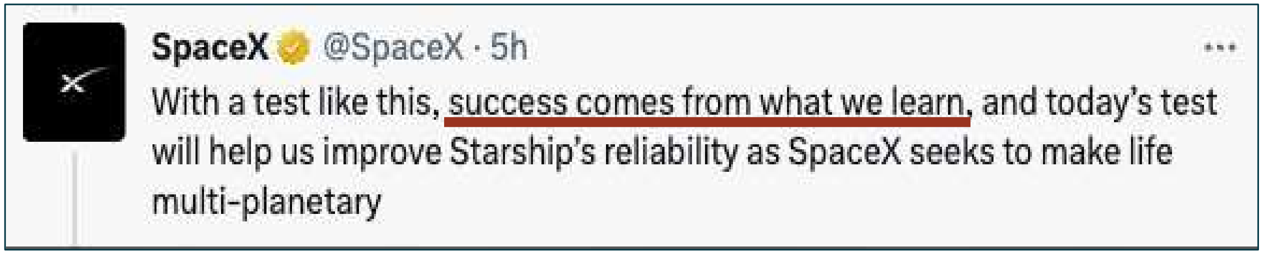
- Frequently integrate changes into richer contexts for early verification and validation
- Common cadence provides regular learning cycles



Provide the psychological safety to learn faster



RUD – Rapid Unscheduled Disassembly



SpaceX @SpaceX · 5h

With a test like this, success comes from what we learn, and today's test will help us improve Starship's reliability as SpaceX seeks to make life multi-planetary

Shift Learning Left

The factory is the product

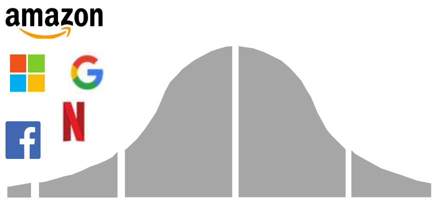
This is the machine that builds the machine and it's the latest version of the machine that builds the machine. The factory is the product.

— Elon Musk

- Products are never one-and-done
- Change the mindset to build quickly and evolve instead of build once and maintain

By Steve Jurvetson - Flickr: Tesla Autobots, CC BY 2.0, https://commons.wikimedia.org/w/index.php?curid=24819239

Where did the early digital innovators investing back in the early 2000s?



Products

Google AdWords

amazon.com

YouTube

twttr

[thefacebook]

NETFLIX

Products that build products

Microservices

Cloud computing

Infrastructure-as-code

Containerization

Monitoring

Service virtualization

Test and deployment automation

API management

Where are today's hardware digital innovators investing?

- Digital engineering – model and sim environments, integrated
  - Tesla
- Reduce manufacturing time - 3D printing, *giga-press*
  - SpaceX manufactures rocket components on-site in TX
- Smart manufacturing
  - SpaceX welds and xrays at the same time. Telemetry data fed to ML to improve welding process
- Automated testing and quality

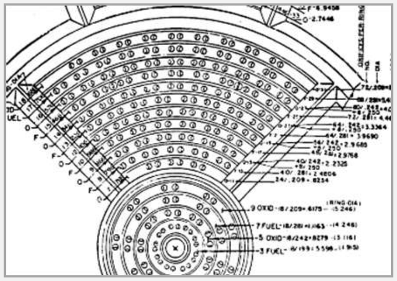
Learning occurs in three environments in hardware

Virtual Environment

Prototype Environment

Operational Environment

Blueprint



3D model

Modeling
Simulation
Digital Shadows
Digital Twins

3D printed prototype

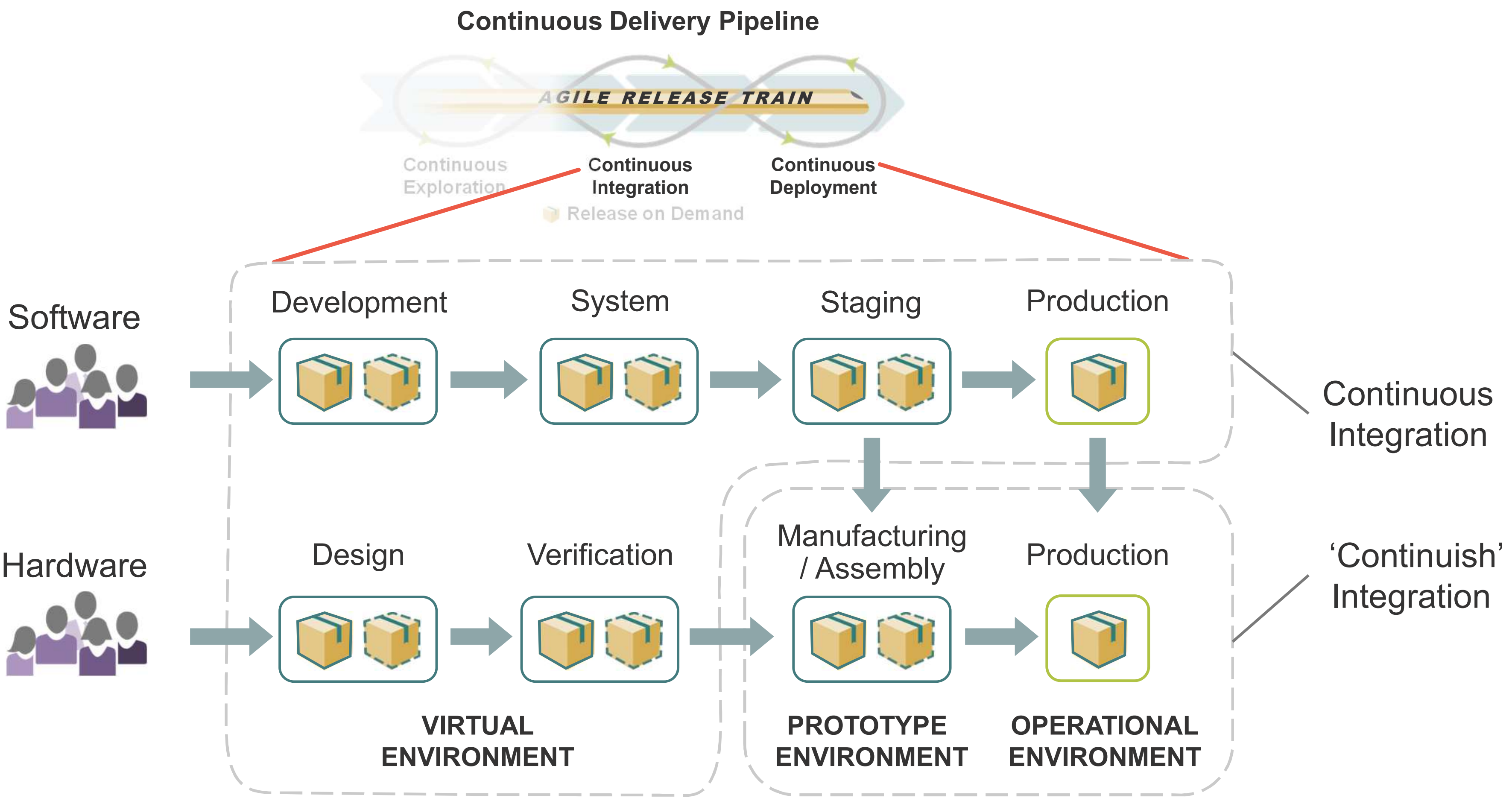
Testing prototype

Mockups
3D Printing
'PCB-in-a-day' services

Real world testing

Image Attribution - Derek Muller, Petr Lebedev, and Emily Zhang. (Aug 12, 2021). The Genius of 3D Printed Rockets. YouTube. www.youtube.com/watch?v=kz165f1g8-E

Build the machine that builds the machine



Learning and innovation creates business value

	Falcon 9 Block 1	Falcon 9 Block 2	Falcon 9 Block 3	Falcon 9 Block 4	Falcon 9 Block 5
Year	2010-13	2013-15	2015-17	2017-18	2018-20
Engine	Merlin 1C	Merlin 1D	Merlin 1D	Merlin 1D	Merlin 1D
Innovation	Tried Parachute recovery (failed)	60% More Thrust	17% more thrust First reusable 1st stage	Improved 2nd Stage Engine Thrust upgrades	Solve reuse & reliability
SpaceX NASA Launches	5	15	25	11	27
All Other NASA Launches	23	18	14	11	2

Questions?

harry@scaledagile.com