Agile HW testing

an enabler for knowledge-based product development

Dirk Holste – Think Flow

Think
Flow

Why agile in HW development

Volkmar Denner (CEO Robert Bosch GmbH)

“For Bosch agility is crucial, it allows us to adjust to the increasing speed of change around us. Agility allows us to remain in a position as an innovation leader.“

Hebert Diess (CEO Volkswagen Group)

“The big question is: Are we fast enough? If we continue at our current speed, it is going to be very tough.”

Elon Musk (CEO Tesla)

“Pace of innovation is all that matters in the long run.”

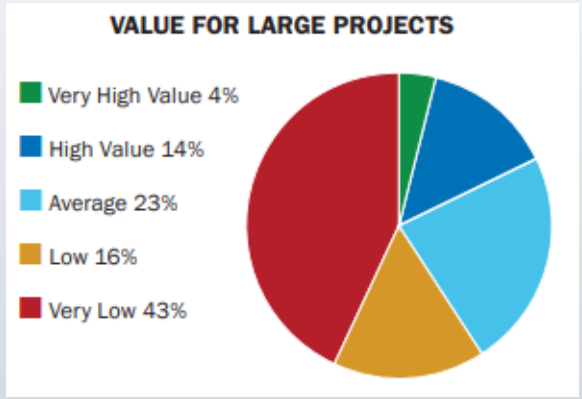
Jim Bridenstine (NASA Chief Administrator)

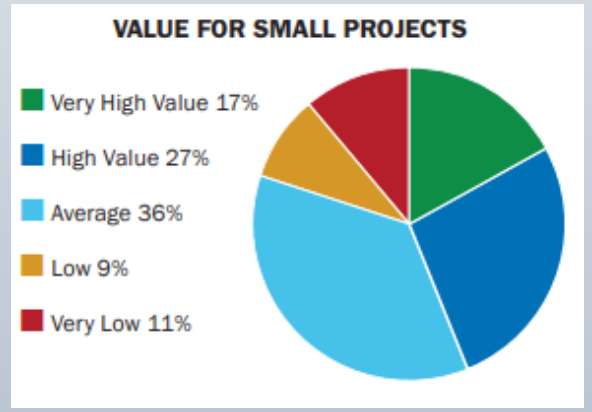
“Every piece of every rocket is full qualified, and everything must go perfect on every launch. And that slows us down. The willingness to fail is something NASA has frankly lacked, but it’s what enables SpaceX to move so fast. To rapidly iterate and improve.”

Challenge of large waterfall projects

CHAOS RESOLUTION BY AGILE VERSUS WATERFALL

SIZE	METHOD	SUCCESSFUL	CHALLENGED	FAILED
All Size Projects	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Large Size Projects	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Medium Size Projects	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Small Size Projects	Agile	58%	38%	4%
	Waterfall	44%	45%	11%





Attributes of success: OnTime, OnBudet, OnTarget, Value and satisfaction
Project was resolved within a reasonable estimated time, stayed within budget, and delivered customer and user satisfaction regardless of the original scope.

Source: CHAOS Report from Standish Group

Think
Flow

Scrum for HW design

Existing challenges

1. Modularity – is your hardware modular enough
2. Longer design cycles in hardware
3. Hardware has higher functional interdependence
4. Hardware has poor refactoring capabilities
5. Higher cost of change
6. Broader range of operating conditions than software
7. Different testing demands

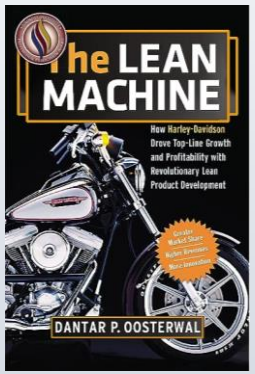
… all of these can be overcome!

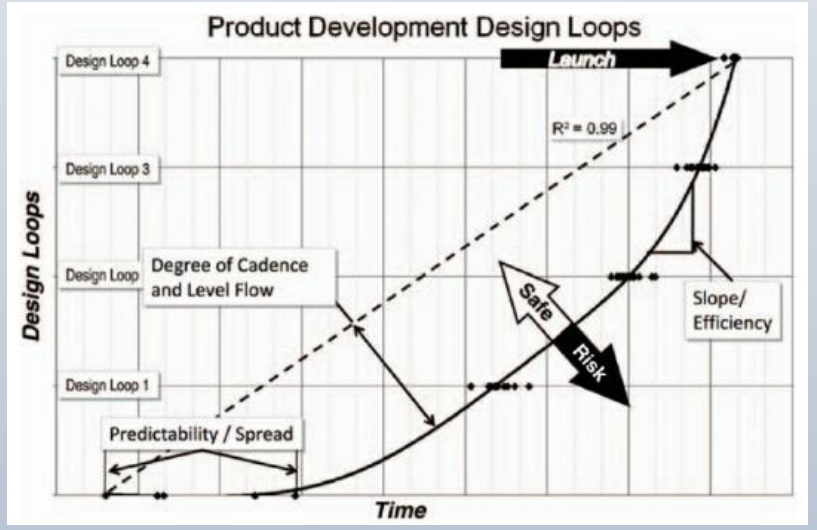
Think
Flow

Learning based HW development

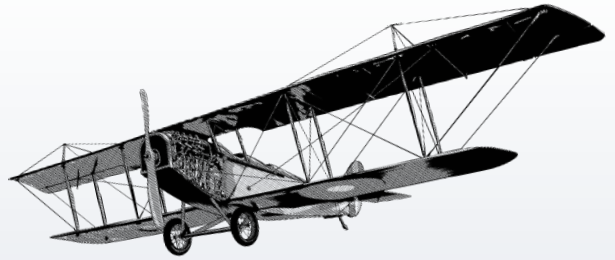
- Relies on creation of re-usable knowledge by learning cycles based on evaluation, definition and design

- Each cycle is followed by a convergence point intended to eliminate concepts that do not work or will not be ready in time for production





First manned flight



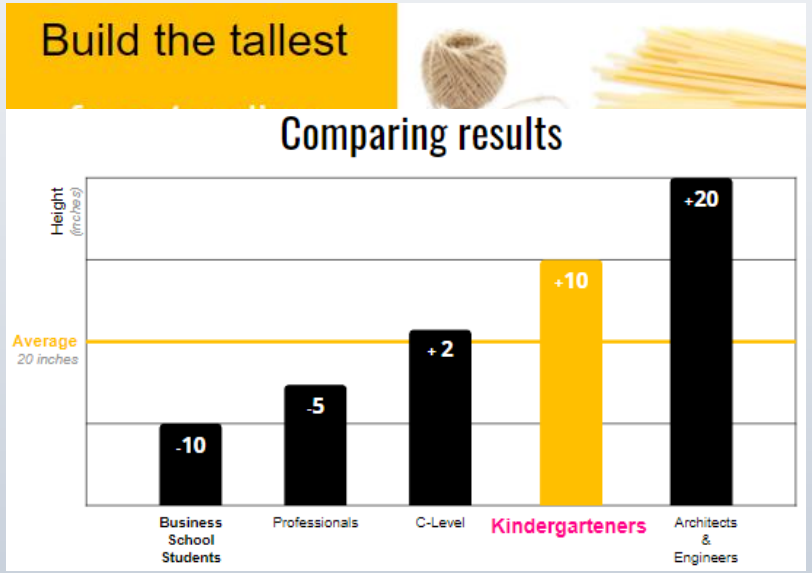
Professor Samuel Pierpont Langley (1834-1906)

- Leading scientific figure for his astronomical research
- Successful flights of his previously designed unmanned flying machines
- Received funding of a total sum of $50,000 for building the first human piloted flying machine
- Despite having an excellent engine, the Aerodrome, met with disastrous results, crashing on takeoff in October 1903 over the Potomac river

- **Wright brothers**
  Orville (1871 – 1948) and Wilbur (1867 – 1912)
- Two American aviators, engineers, inventors, and aviation pioneers
- Building various design alternatives for gliders and kites to gain knowledge
- In-depth observation and repeated testing of prototypes allowed them to explore the trade-offs between different variables at the time
- In less than two years of testing alternative designs and less than $1,000 dollars to fly the first in history piloted flying machine

Marshmallow contest

- What is the challenge?
- What groups perform best?
  - Professionals
  - C-Level
  - Architects & Engineers,
  - Business School Students,
  - Kindergarteners
- What is the difference in the process they use?



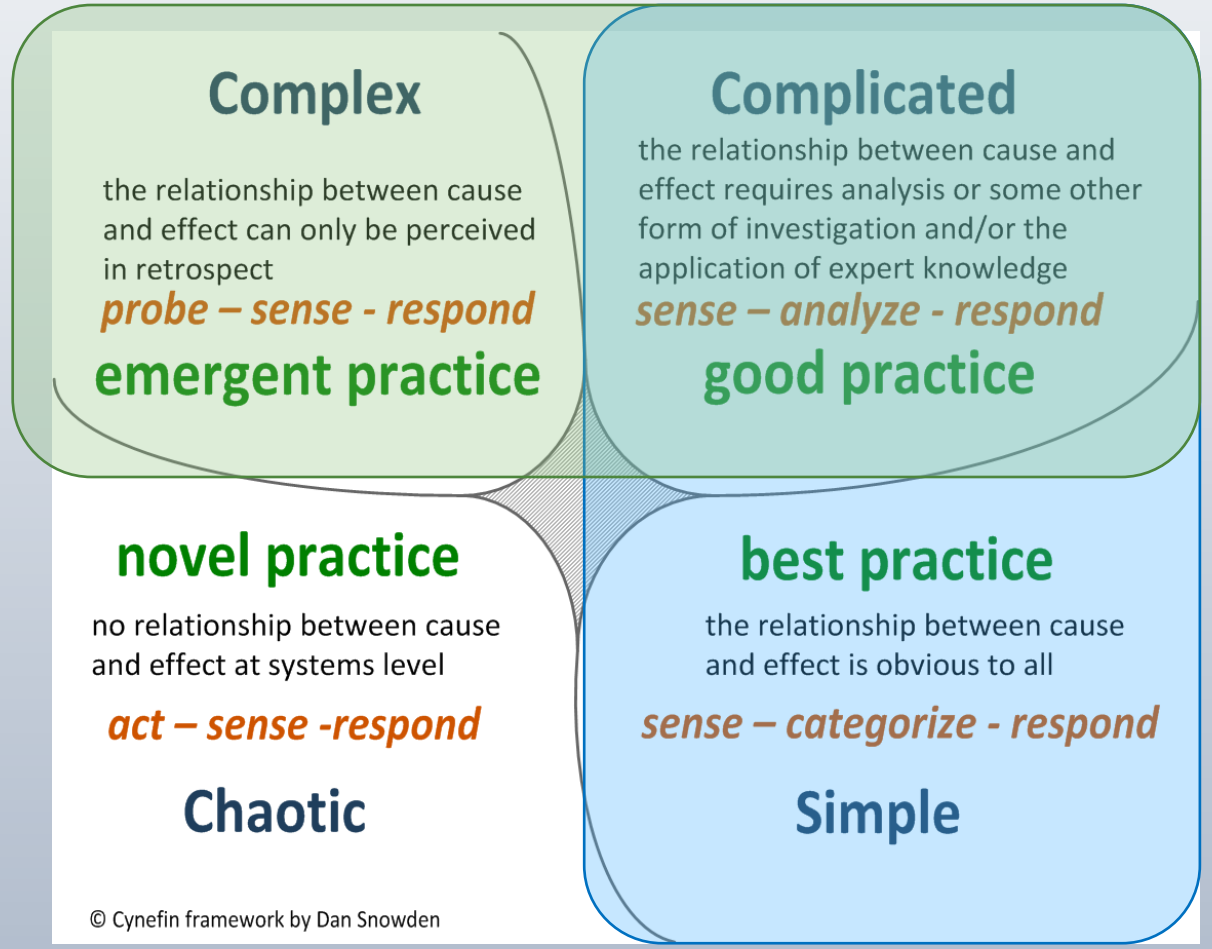
Think
Flow

Verification Mindset: Tests should not fail

- The purpose of tests are to verify that the system works as expected
- Failures are seen as negative and should be avoided
- There is a verification mindset not a learning mindset

But we should know – shouldn't we?

Learning based product development with emerging solution

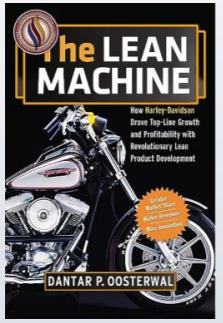


Traditional product development with predicted solution

Learning based Development

- Product development, is more dependent on what needs to be learned than on what tasks must be completed to exit a gate.
- Projects failed because designs selected in early phases could not go through enough test and redesign loops in time for launch.

Change of mindset



Design then test

learn then design

Think
Flow

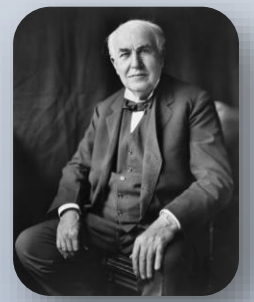
Love failure & deviations

- Embrace failure and deviations to learn and progress faster
- Dare to try!
- Be guided by:
  - Is it safe enough to try?
  - What can we learn?

- Note: This doesn’t mean you should take unnecessary risks!

“I have not failed. I've just found 10,000 ways that won't work.”

Thomas A. Edison



Think
Flow

Visualize and improve your HW test / learning flow

Think Flow

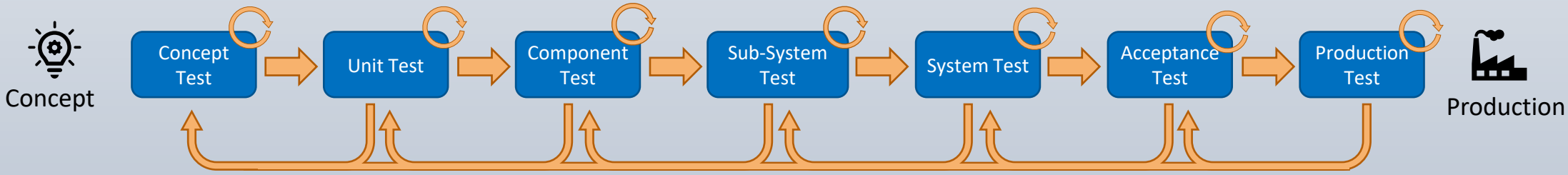
Product Development Value Stream

- A product development value stream is the flow from concept to release to customer
  - It contains the object to be delivered, people and systems needed, the flow of information and knowledge aggregation via analysis and test

- Aim is to optimize the lead time from concept to cash

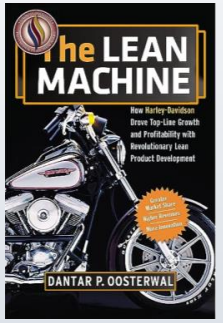
Test Value Stream / Learning Loops

- In complex system including HW components running tests and analyzing the results (learning loops) is an important factor in the whole product development value stream

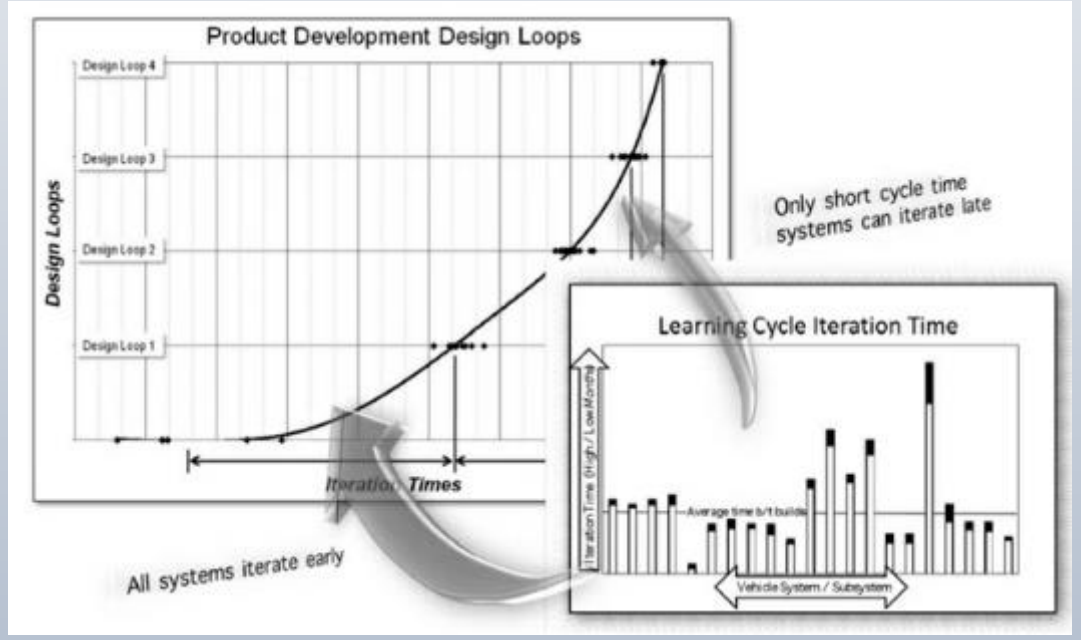


Think
Flow

The threat of long design loops



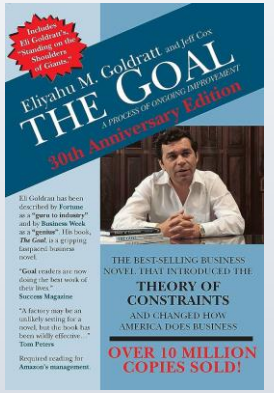
- Systems/module with long design loops have a heavy impact on project success
- Designs chosen early in the project can not be verified and redesigned in time for launch
- Systems with faster design loops are several design loops ahead with risk that parts do not fit together, if not integrated early and often
- Product development can not progress faster than the slowest system in the cycle.
  They are a constraint to product development!

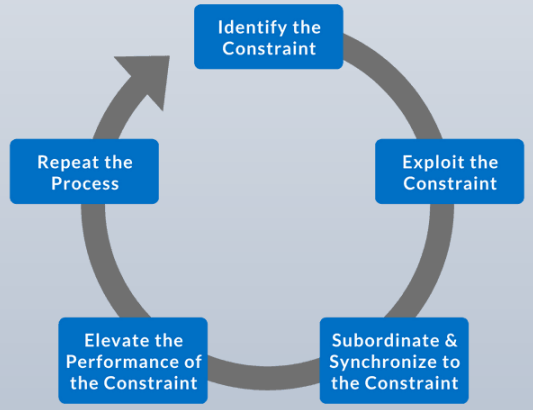


Think Flow

Theory of constraints

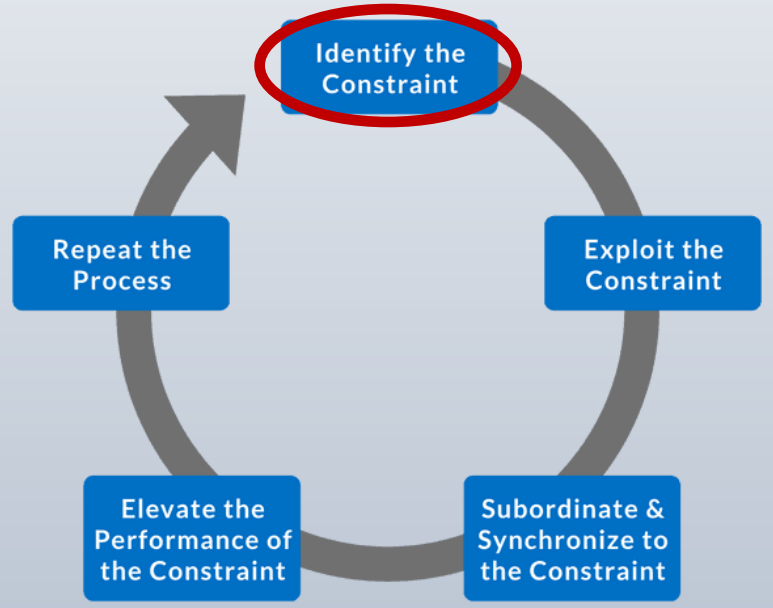
- Identify the constraint – find your “Herbie”
- Make quick improvements to the throughput of the constraint using existing resources (i.e., make the most of what you have).
- If the constraint still exists (i.e., it has not moved), consider what further actions can be taken to eliminate it from being the constraint





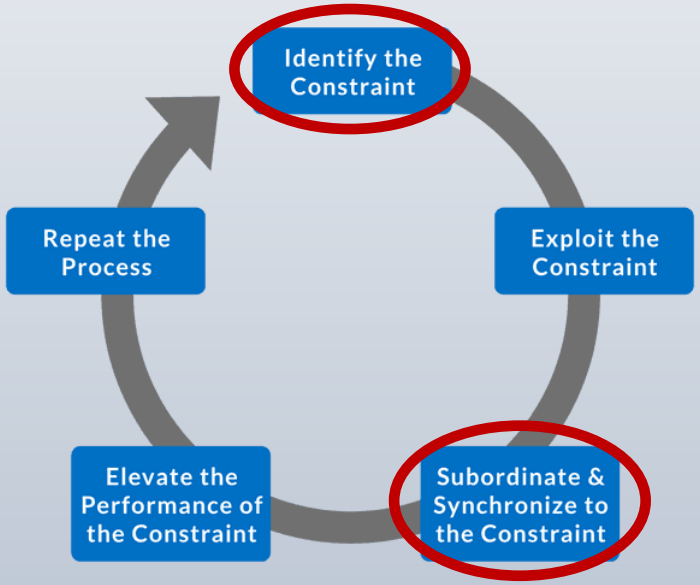
Task: Identify your “Herbie” in the product development value stream

- Identify the system/module with:
  - Longest design loops
  - Heavy interconnection and dependencies to other modules
  - High impact on the product / project



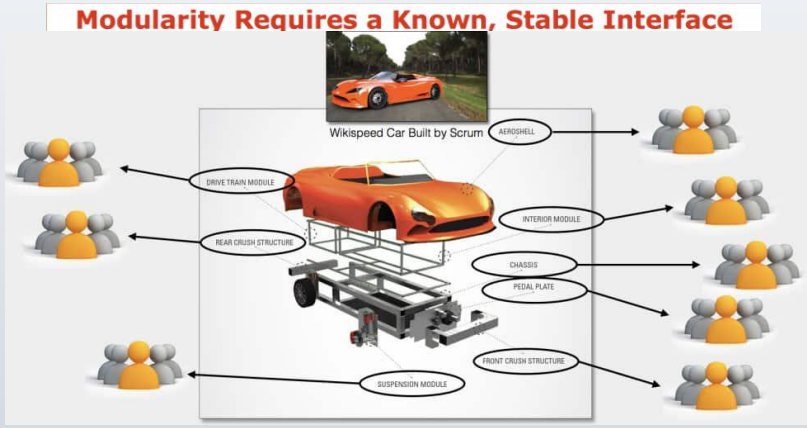
Task: Exploit and subordinate your “Herbie” in the product development value stream

- Find quick improvements to mitigate the constraint using existing resources
- Explore other possibilities to remove dependencies to or support learning loops and integrations points with the constraint

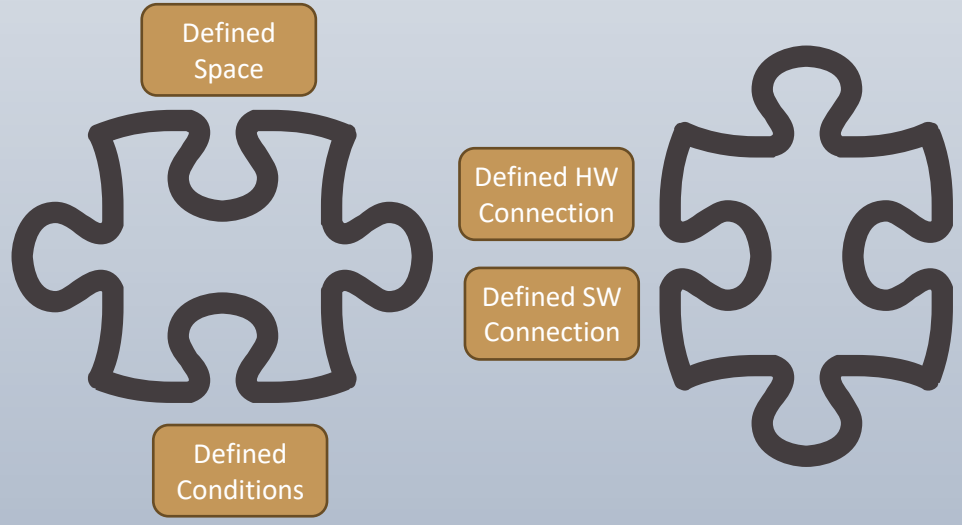


Modularization with defined interfaces

- Interfaces accelerate changes in both development and production environments
- Enable frequent, independent design iterations
- Create modules that define
  - Physical parameters (e.g. weight, space)
  - Environmental parameters (e.g. heat)
  - Interfaces to other modules
- Specify interfaces that define
  - Physical connections (e.g. mounts)
  - Electrical connections
  - Software connections

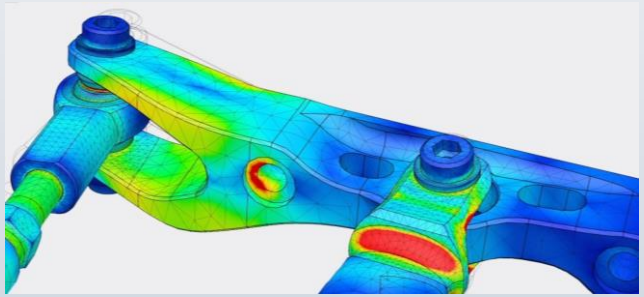


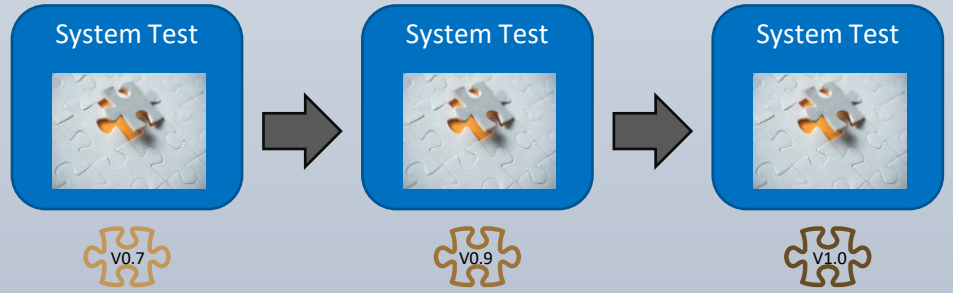
https://www.scruminc.com/scrum-in-hardware-guide/



Integrated testing

- Model based system engineering (MBSE)
  - Puts models at the center of system design
  - Integrate models and simulation to learn in the virtual world
  - Validate models in the physical world
- Protype based integrated testing
  - Defining tests and prototypes to verify the most critical learning needed
  - Integrate early prototype in the whole system to get early feedback on critical questions
  - Evolve with more mature prototypes

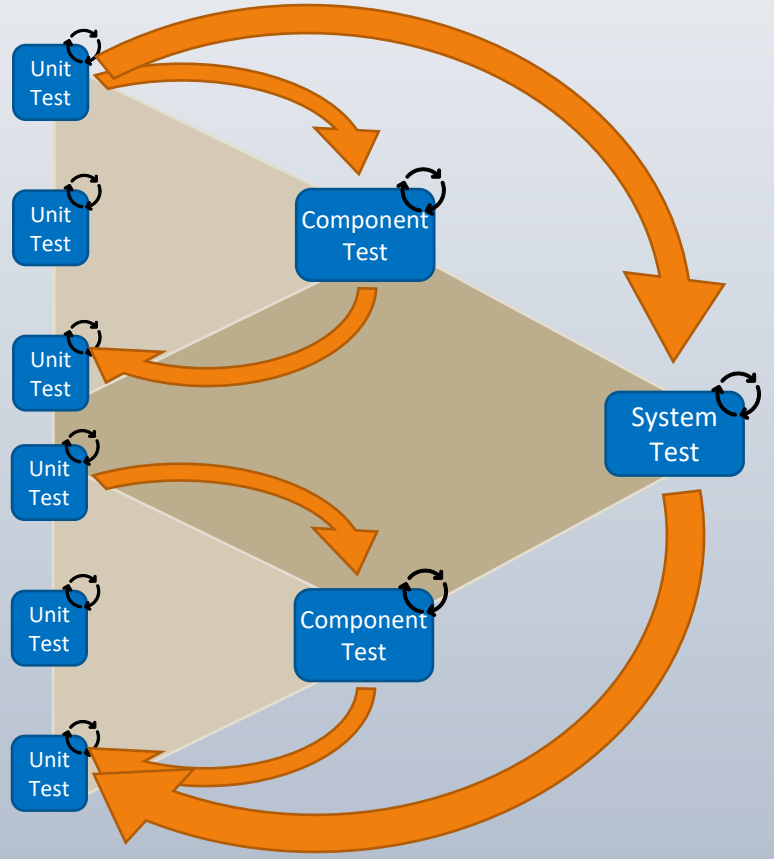




Think
Flow

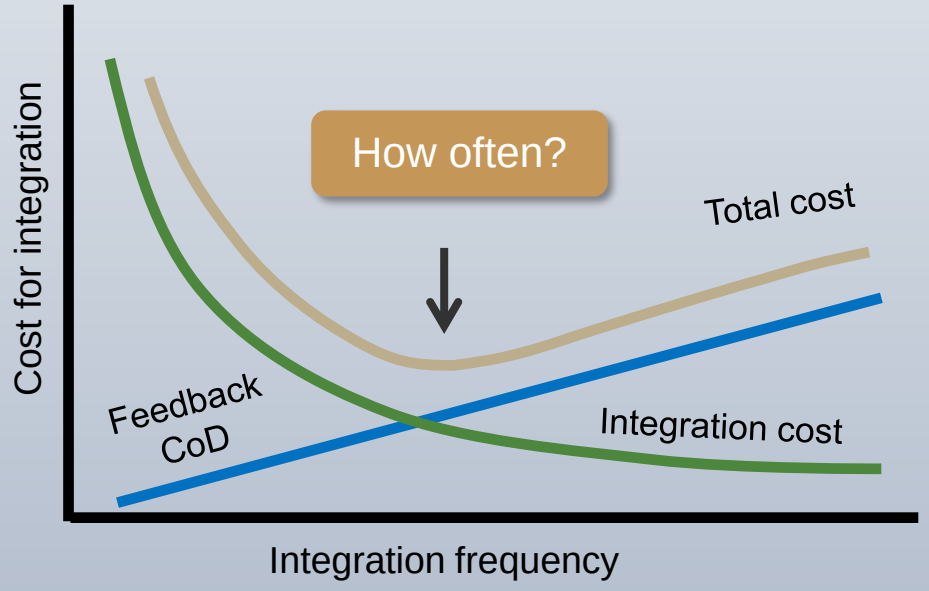
Amplify learning loops

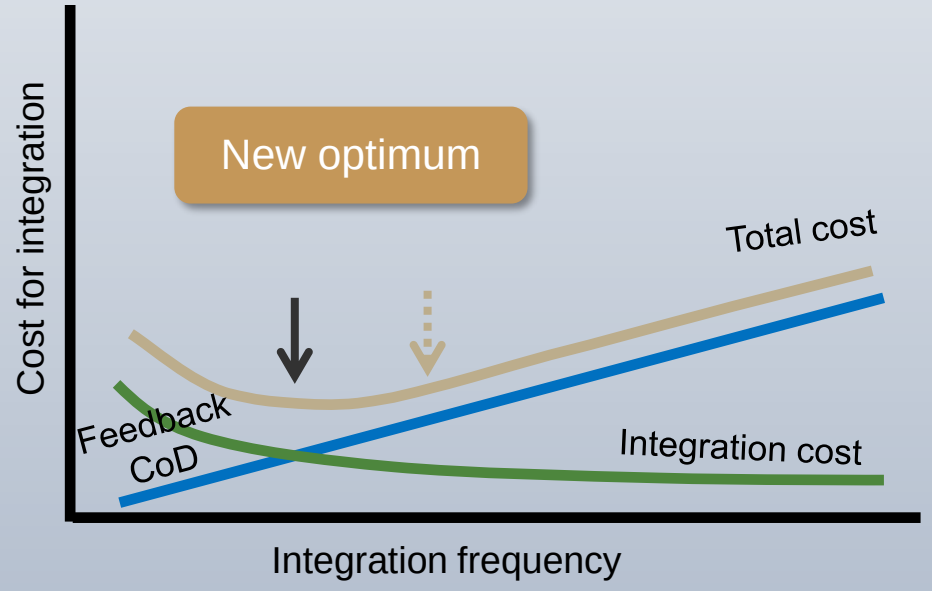
- On each test level iterate, integrate and evaluate
- Later stages use latest working elements and integrate them
- These loops are interconnected and nested into each other
- Feedback results to earlier stages to amplify learning



Strive for more frequent integration

- Invest in infrastructure to simulate and automate to:
  - Minimize dependencies on manual builds / tests
  - Reduce batch size and integrate and learn more often





Principles of Product Development Flow, Don Reinertsen

Thank you!

Think Flow

Dirk Holste

dirk.holste@thinkflow.se

www.thinkflow.se

We are curious about your challenges and together with you apply methods from our (lean & agile) toolbox to address them.

Make a difference in your flow

Mikael Broomé

Martin Teljeby

Anders Fresk

Andreas Johansson

Anders Jonsson