

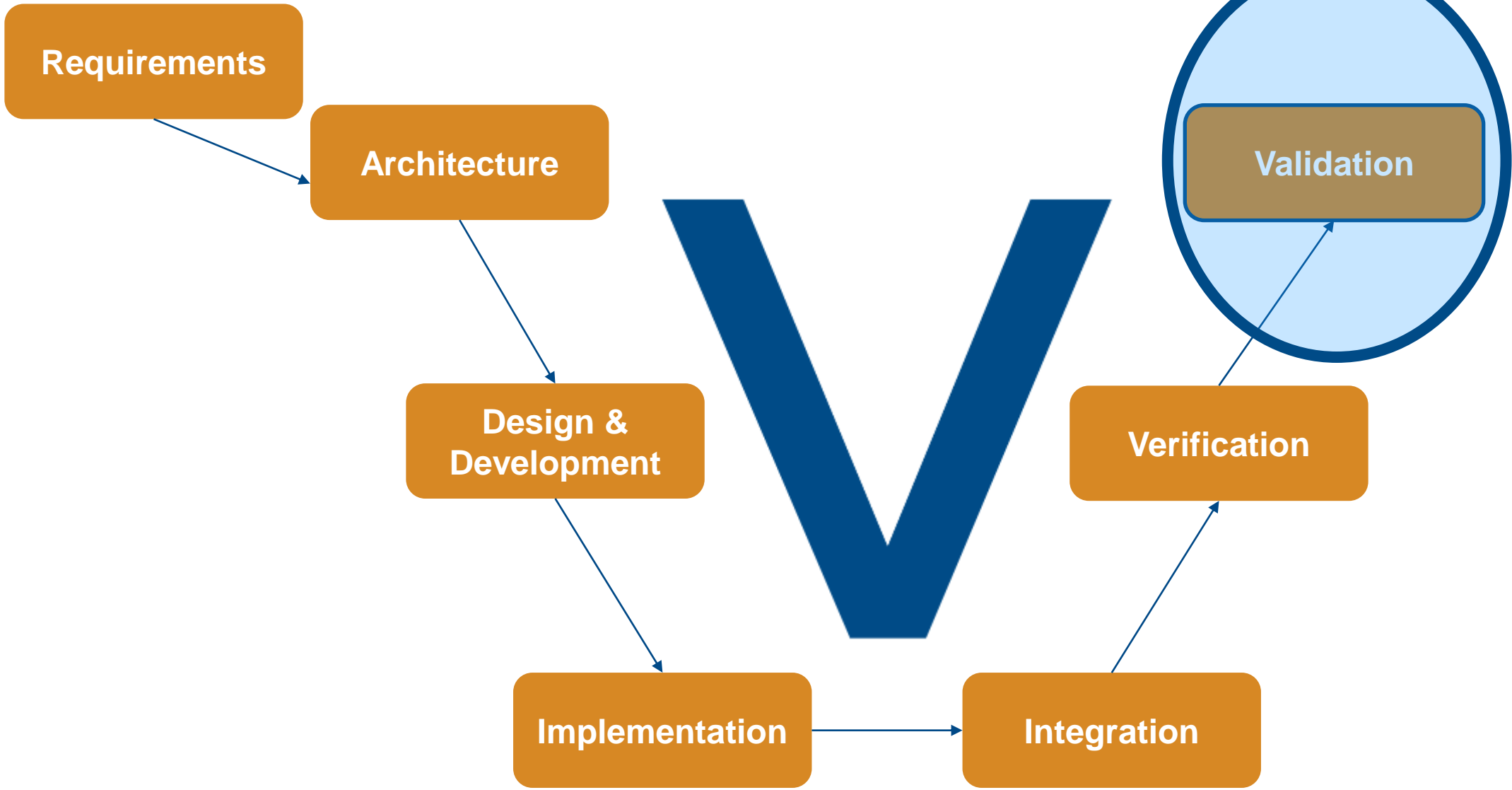
MathWorks
**AUTOMOTIVE
CONFERENCE 2022**
North America

**Turning the Tables on Validation with
Agile Model-Based Design**

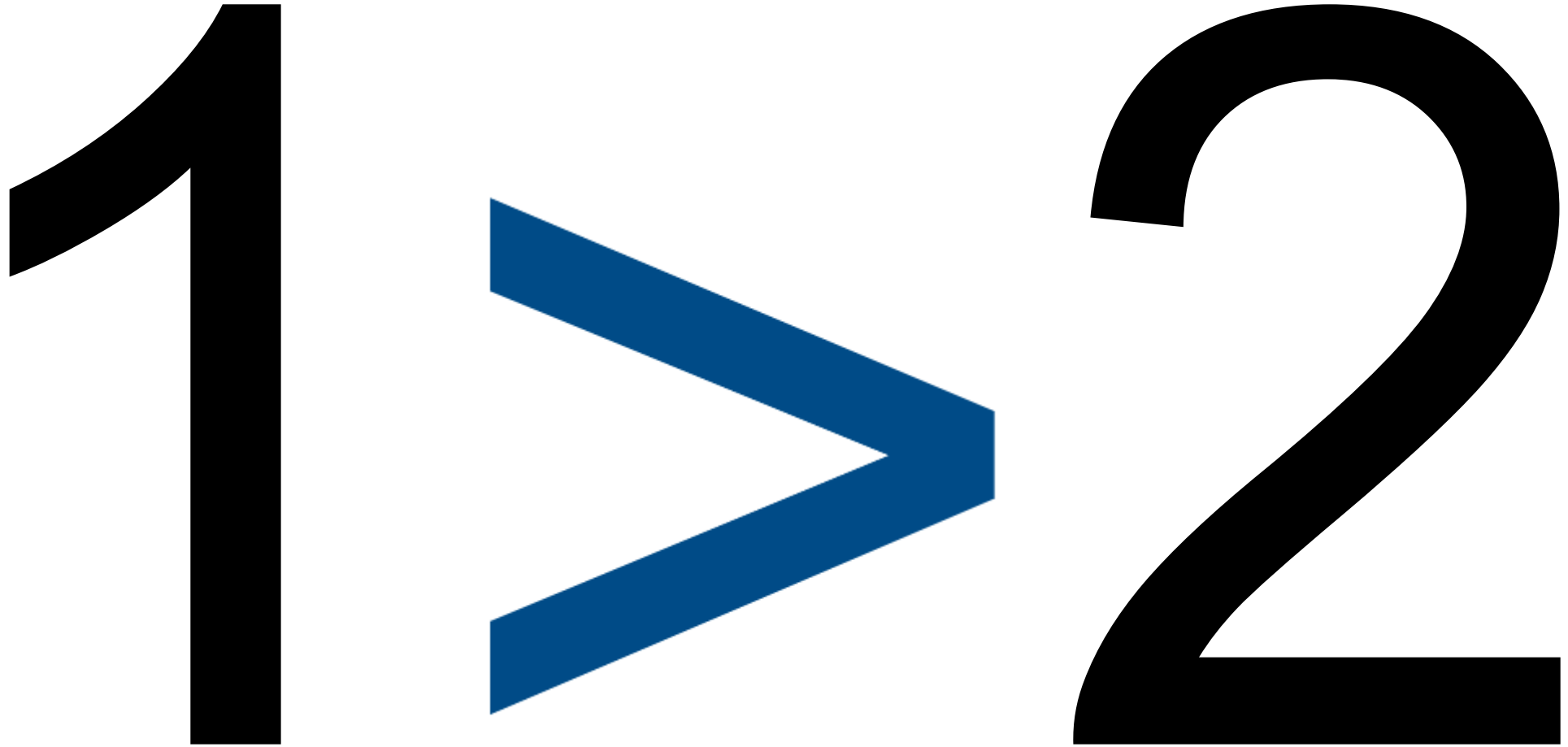
Jim Ross, MathWorks



Simplicity--the art of maximizing the amount of work not done--is essential.

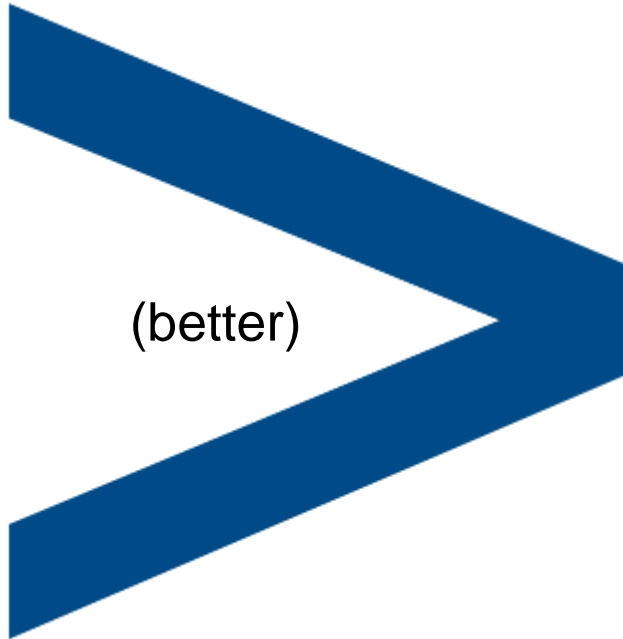






Simplicity--the art of maximizing the amount of work not done--is essential.

1



2

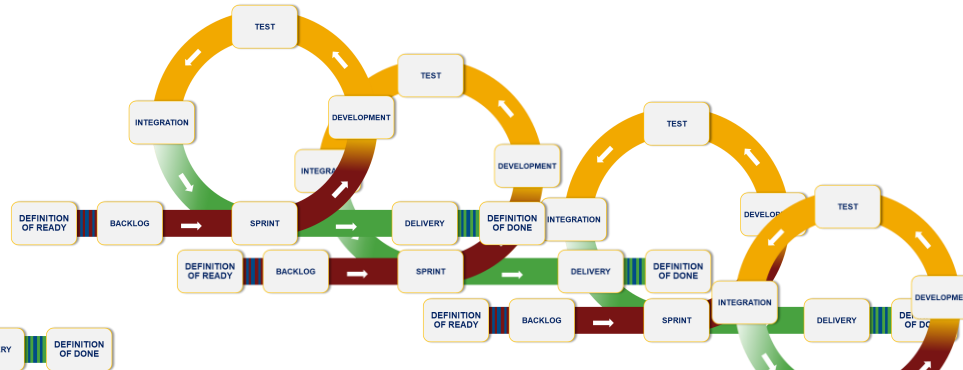
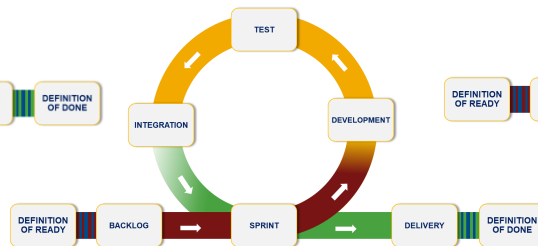
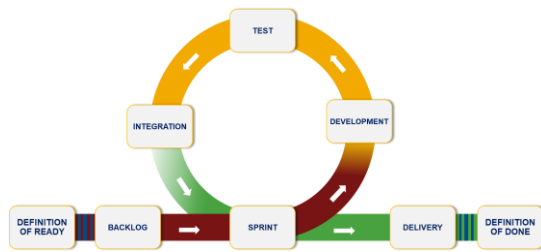
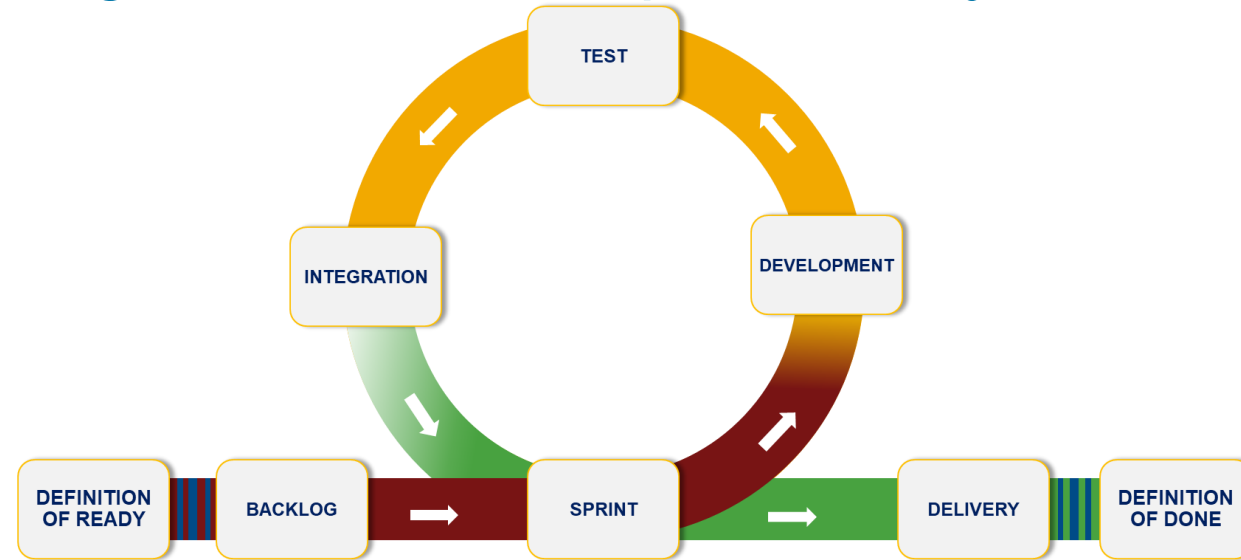
1 (time) > 2 (times) > 3 (times) > ...

Have you ever had a problem that kept you awake at night?
I want to tell you about one time that happened to me.

Have you ever had a problem at work keep you awake at night?
I want to tell you about one time that happened to me.



Let's see if an Agile twist can help this story.

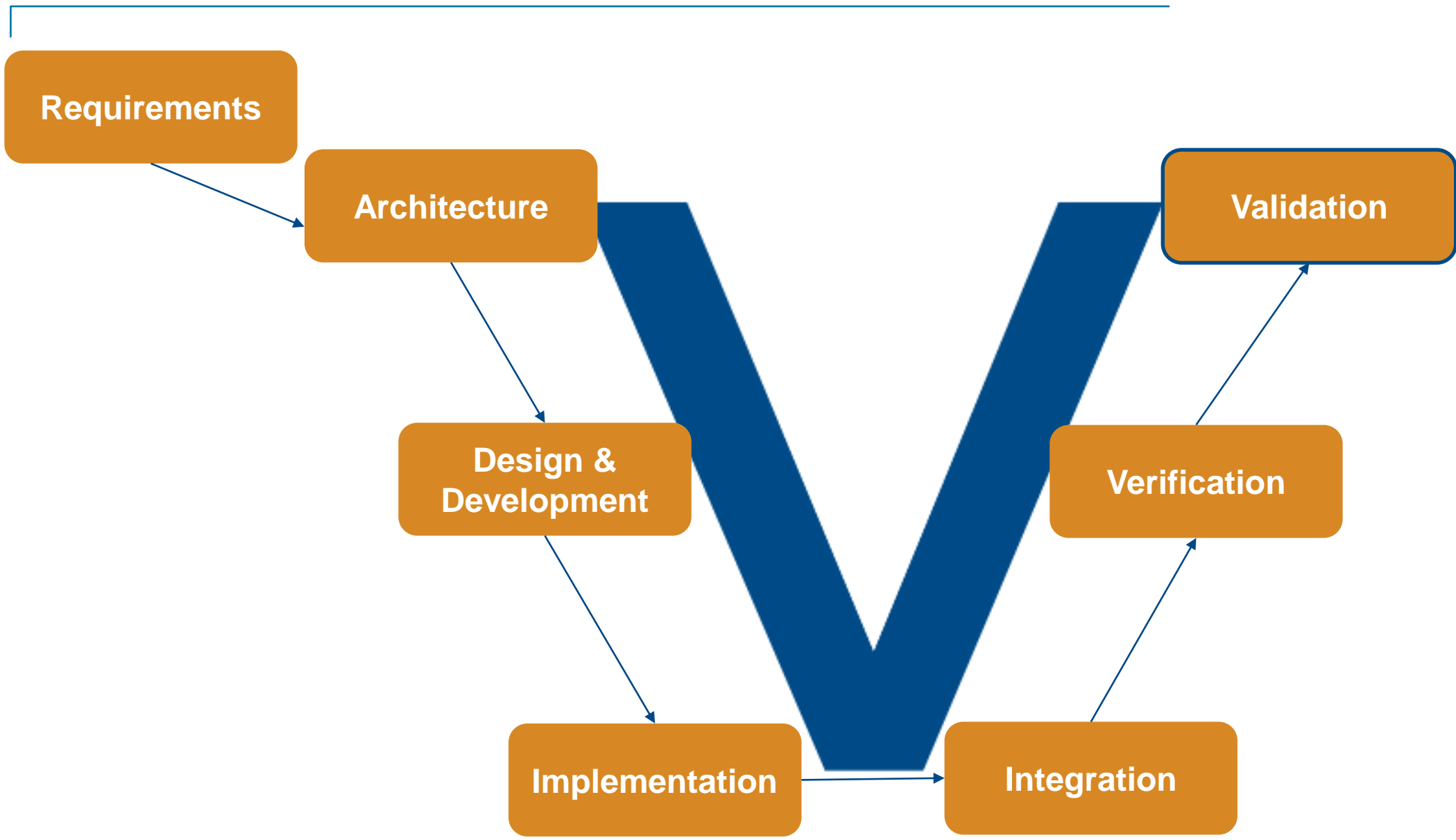


Release Fan Control Software

Apply Fan Control

Apply Fan Control





Design

V

Validate

Implement & Verify

Design

Validate

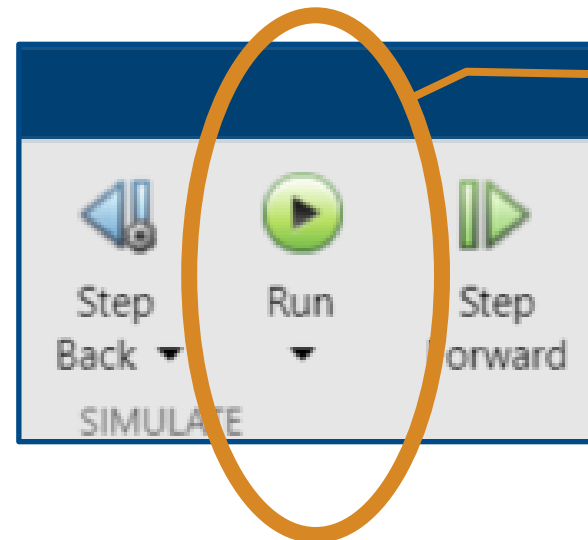
Implement & Verify

The BIG Question

How do I validate early?

The big question – how do I validate early?

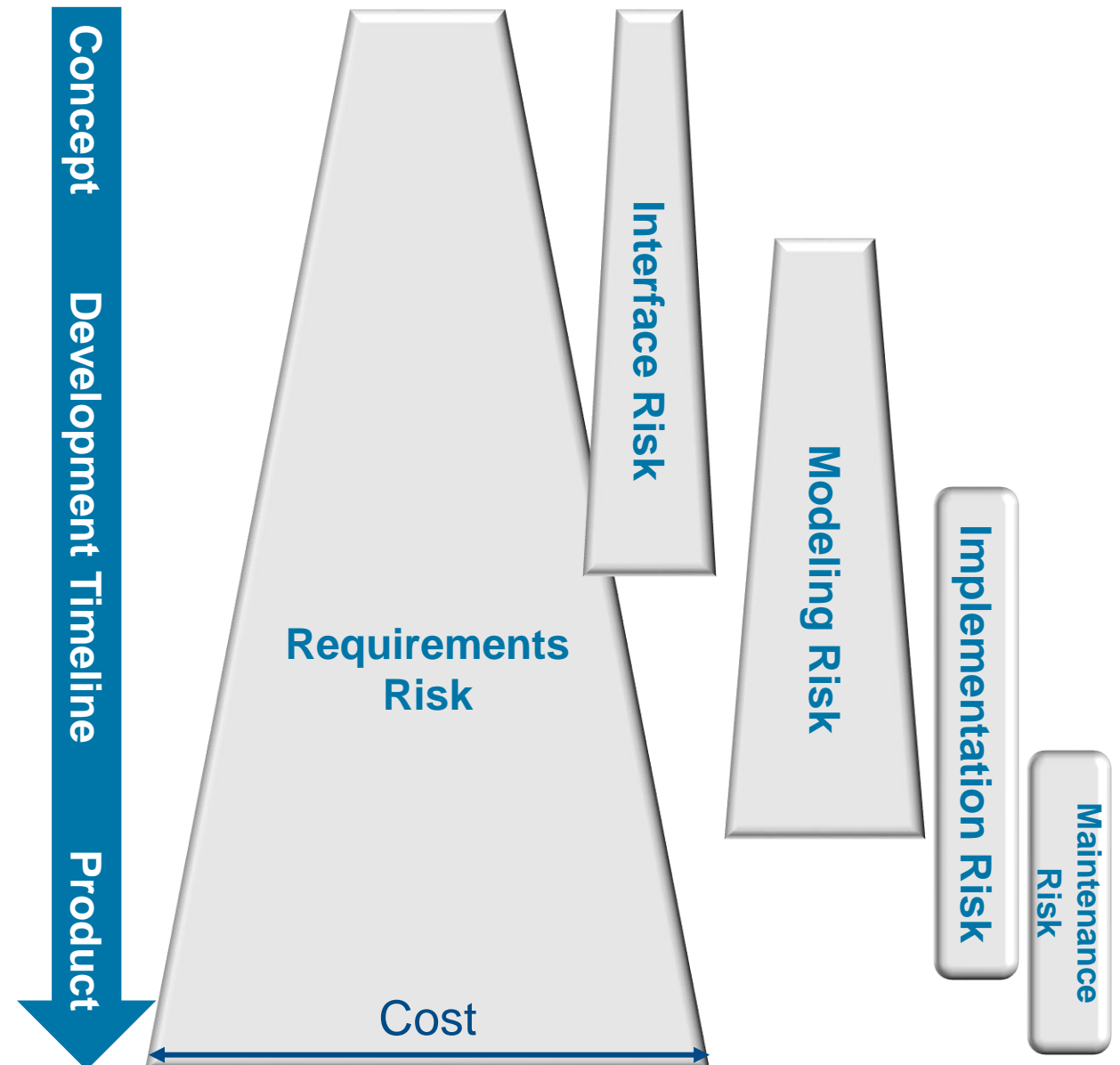
SIMULATION



“The most valuable button in Simulink”

**Sarah Dagen
MathWorks**

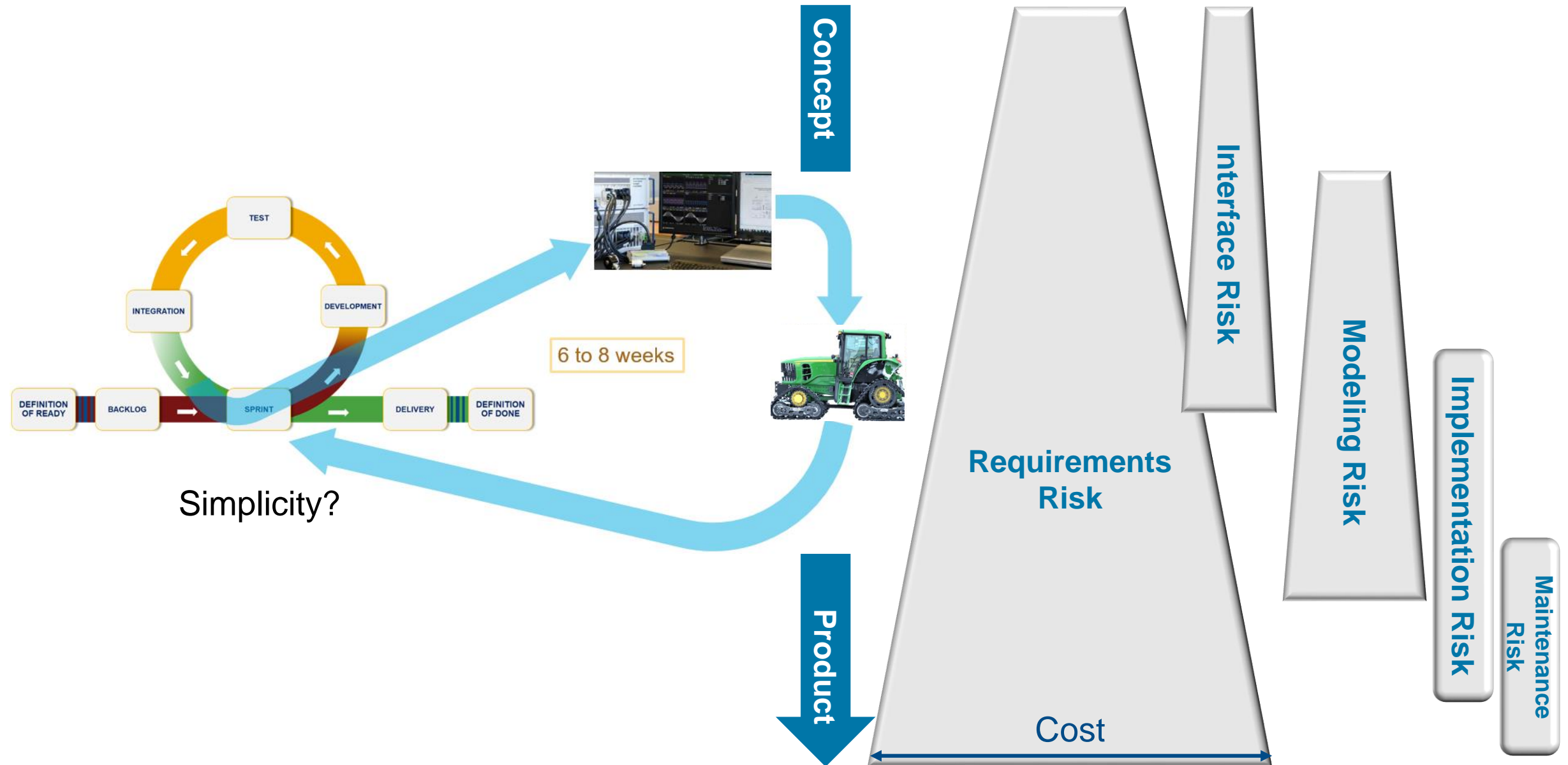
Verification and Validation activities are intended to reduce risk.



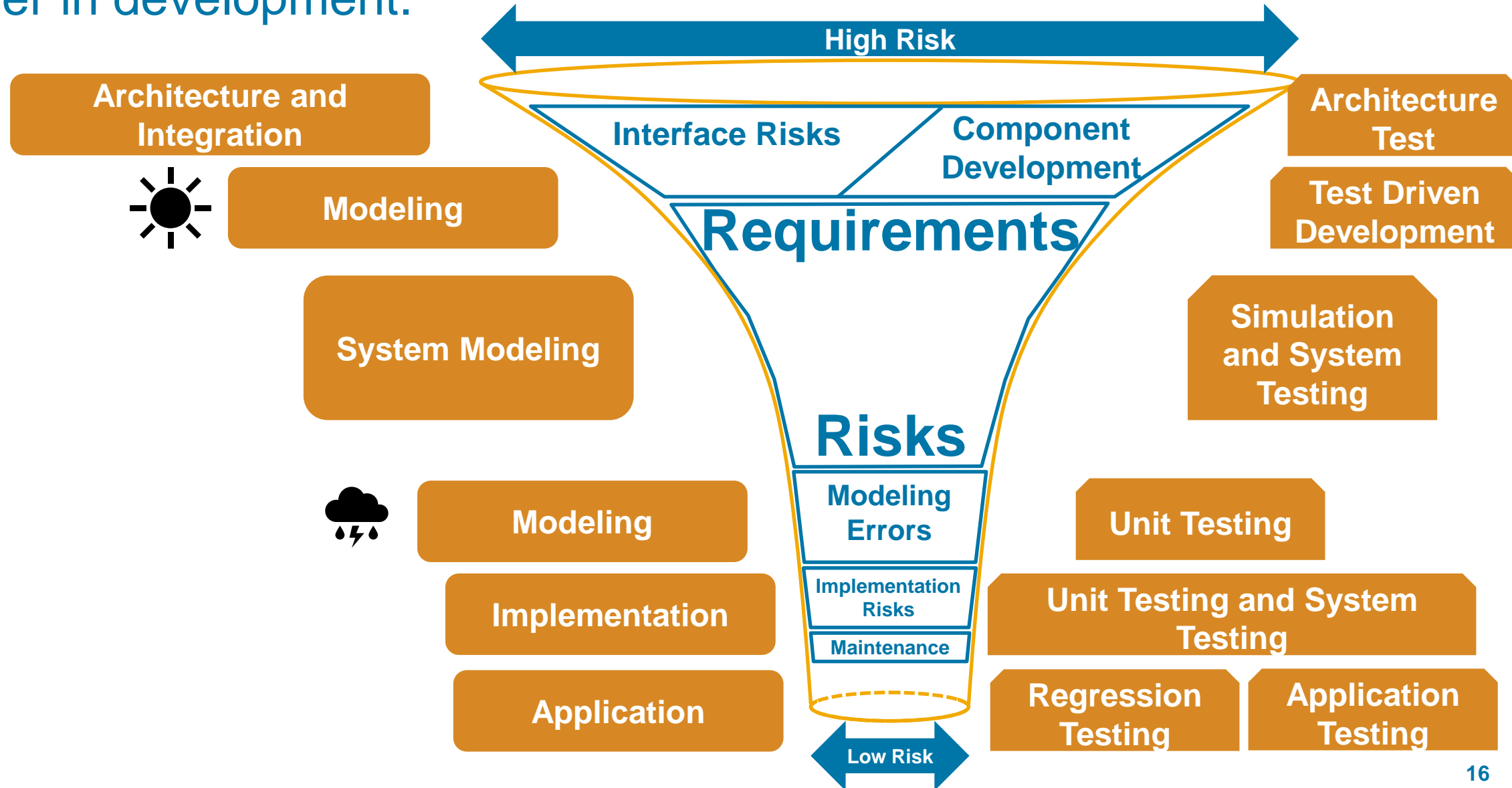
Another Agile Principle

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

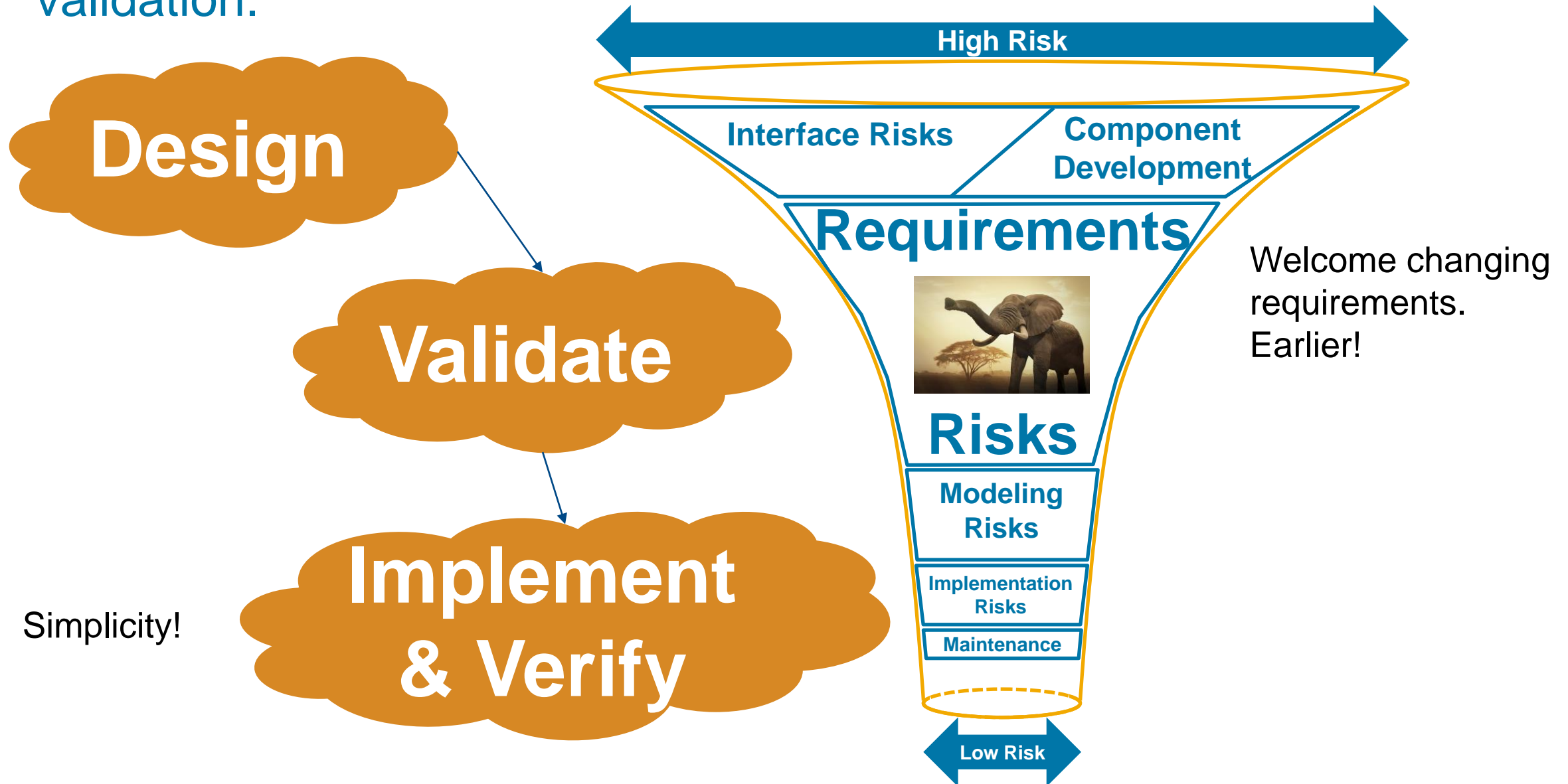
An Agile approach might address these risks along the timeline.



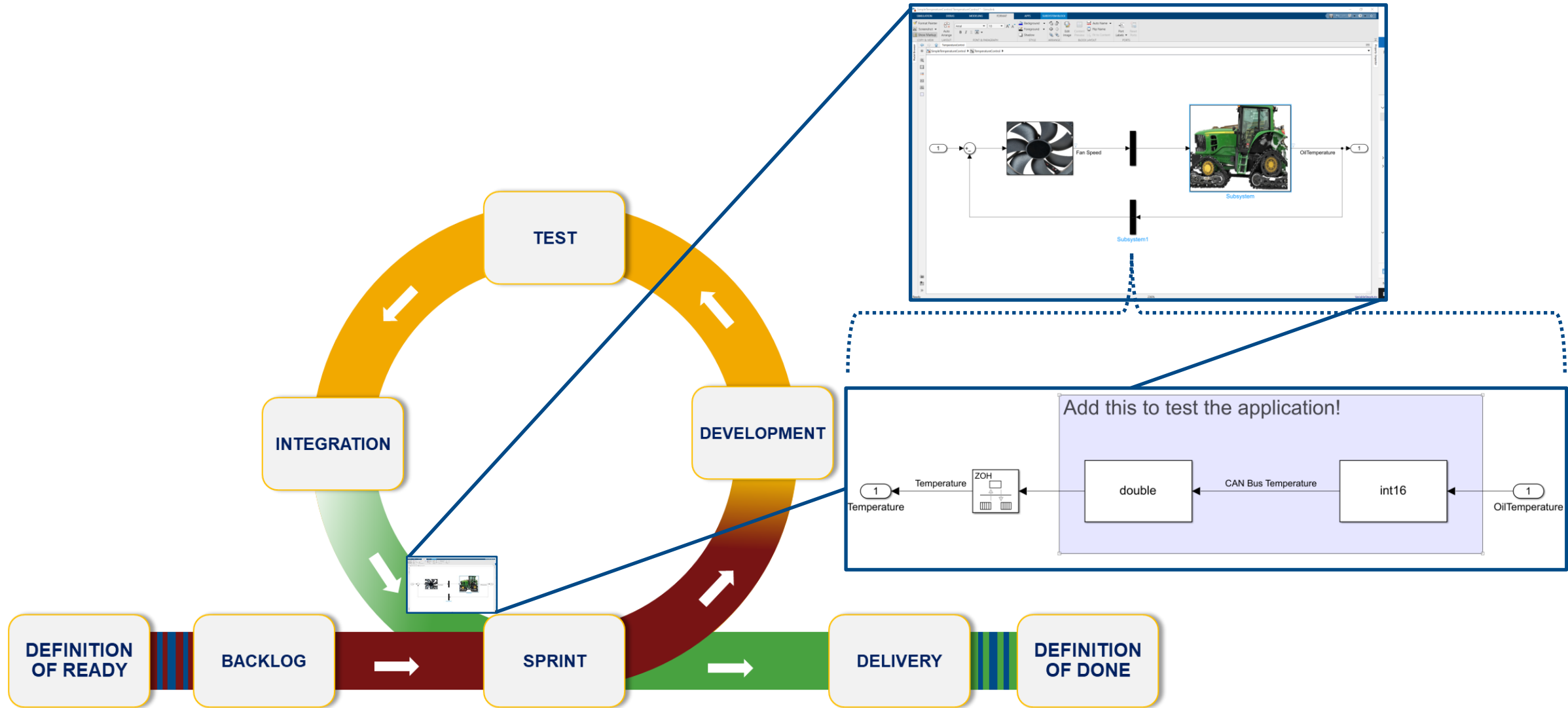
Agile Model-Based Design allows you to address the biggest risks earlier in development.



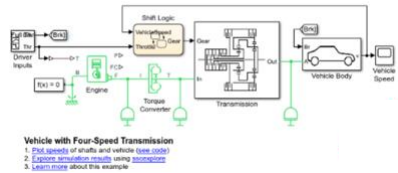
Agile Model-Based Design means up-front and continuous validation.



Let's re-imagine that tractor with Agile Model-Based Design.



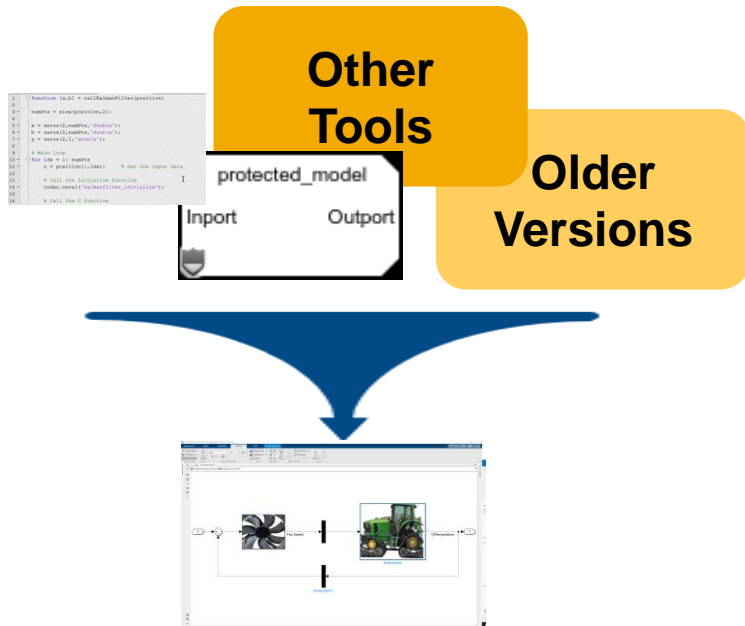
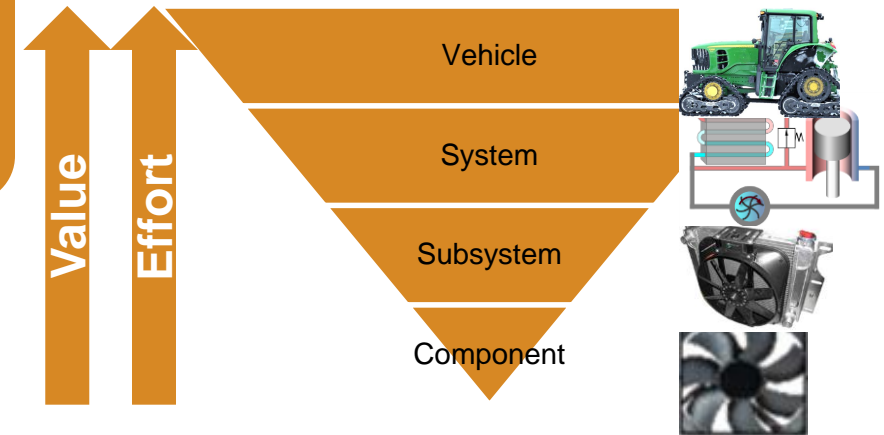
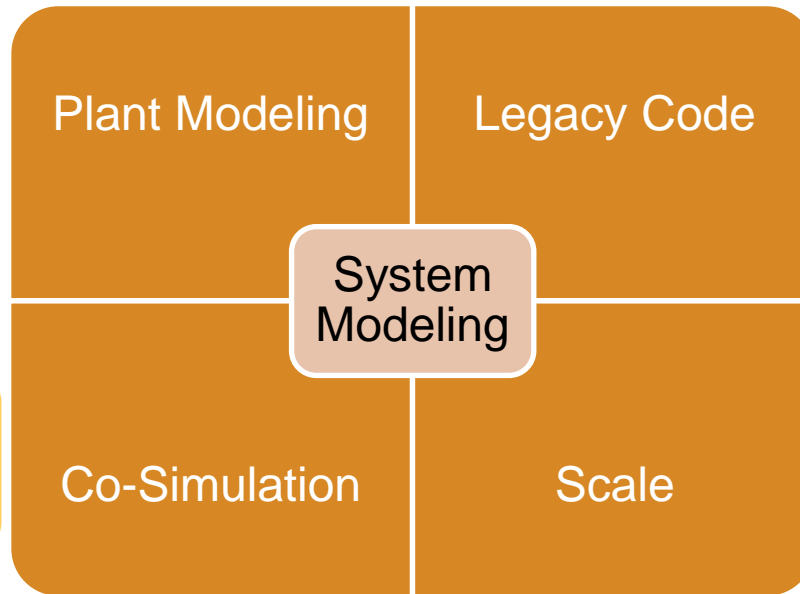
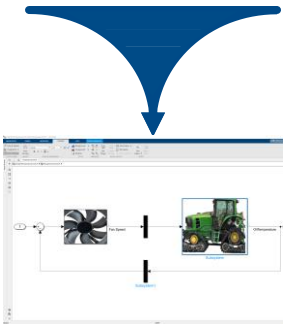
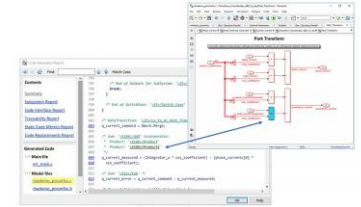
What issues might arise along the way?



```

1 function [a,b] = walkKalmanFilter(position)
2
3 numPts = size(position,2);
4
5 a = zeros(2,numPts,'double');
6 b = zeros(2,numPts,'double');
7 y = zeros(2,1,'double');
8
9 % Main loop
10 for idx = 1:numPts
11     z = position(:,idx); % Get the input data
12
13     % Call the initialize function
14     coder.ceval('kalmanfilter_initialize',
15               % Call the C function
16


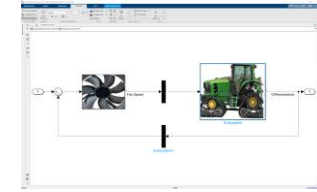
```



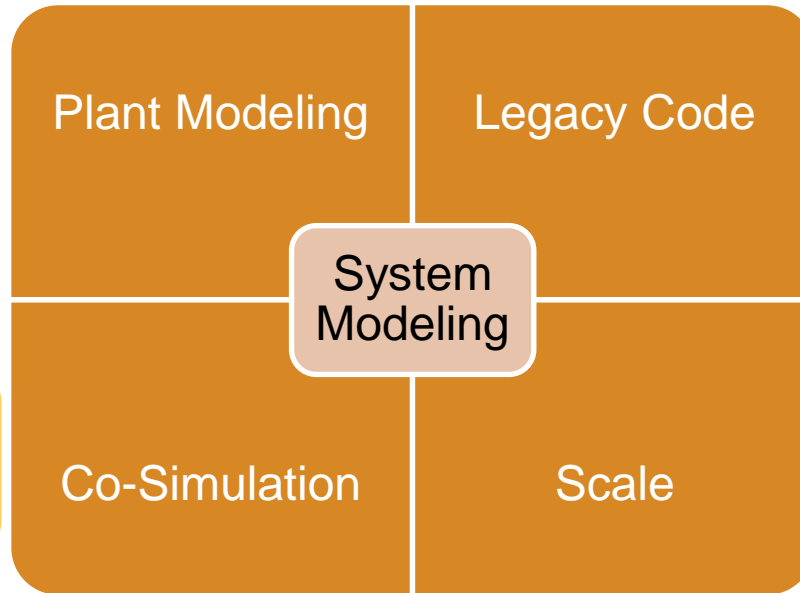
Let's consider some methods for overcoming these barriers.
 First, remember that Agile principle: Simplicity

```

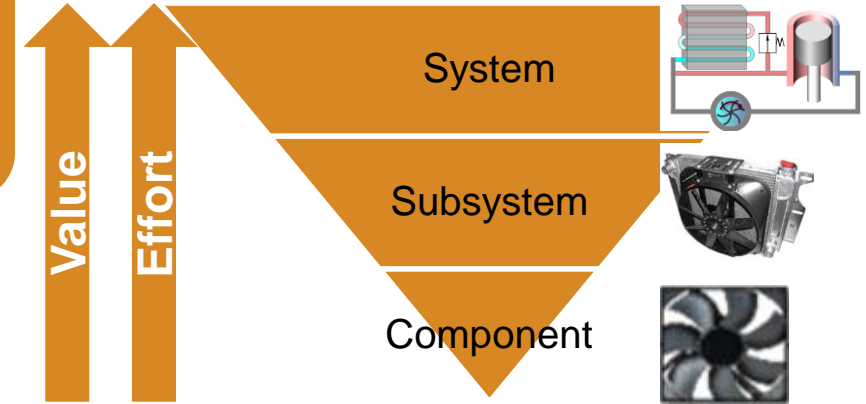
1 function [a,b] = myFunction(x)
2
3 numPts = size(x,1);
4
5 a = zeros(1,numPts);
6 b = zeros(1,numPts);
7
8 % Main loop
9 for i=1:numPts
10     x = x(i,:);
11
12     % Call the sub-function
13     [a(i),b(i)] = mySubFunction(x);
14
15 % Call the sub-function
16
    
```

$$\tau = I\alpha$$

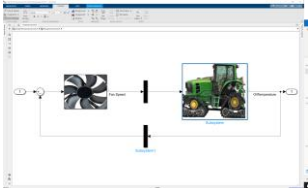


Other Tools
Older versions

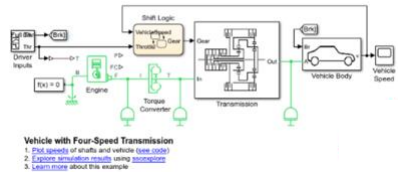


```

1 function [a,b] = myFunction(x)
2
3 numPts = size(x,1);
4
5 a = zeros(1,numPts);
6 b = zeros(1,numPts);
7
8 % Main loop
9 for i=1:numPts
10     x = x(i,:);
11
12     % Call the sub-function
13     [a(i),b(i)] = mySubFunction(x);
14
15 % Call the sub-function
16
    
```

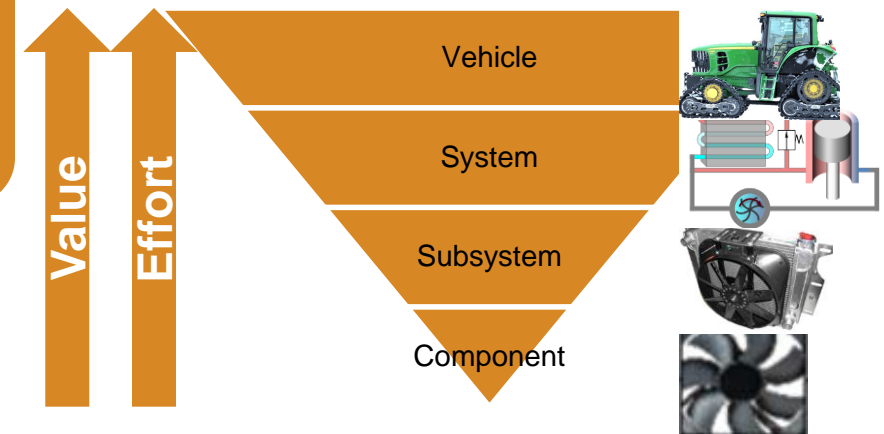
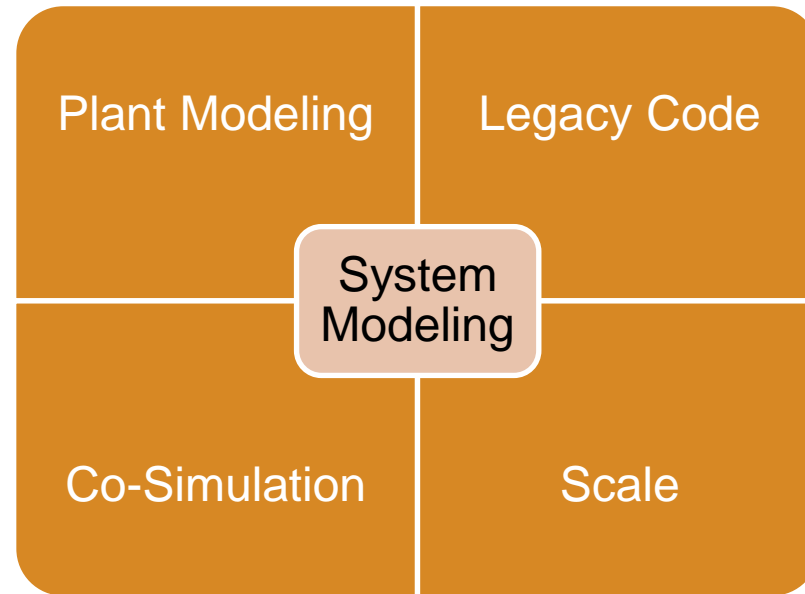
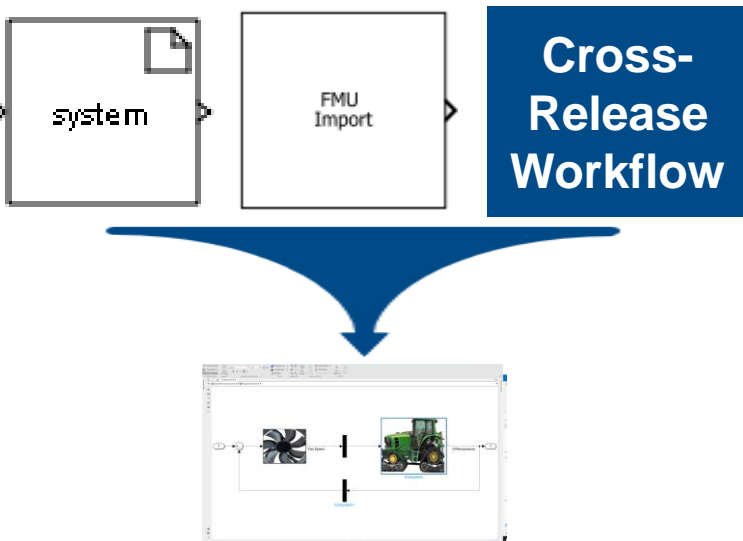
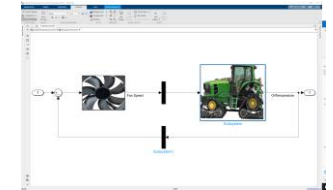
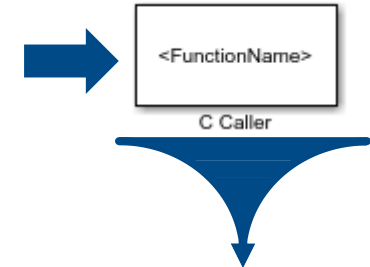


Look to MathWorks for help.



```

1 function [a,b] = walkKalmanFilter(position)
2
3 numPts = size(position,2);
4
5 a = zeros(2,numPts,'double');
6 b = zeros(2,numPts,'double');
7 y = zeros(2,1,'double');
8
9 % Main loop
10 for idx = 1:numPts
11     z = position(:,idx); % Get the input data
12
13     % Call the initialize function
14     coder.ceval('kalmanfilter_initialize');
15
16     % Call the C function
    
```



See [MathWorks Consulting](https://www.mathworks.com/help/consulting) or contact me: jimr@mathworks.com

MathWorks
**AUTOMOTIVE
CONFERENCE 2022**
North America

Thank you

