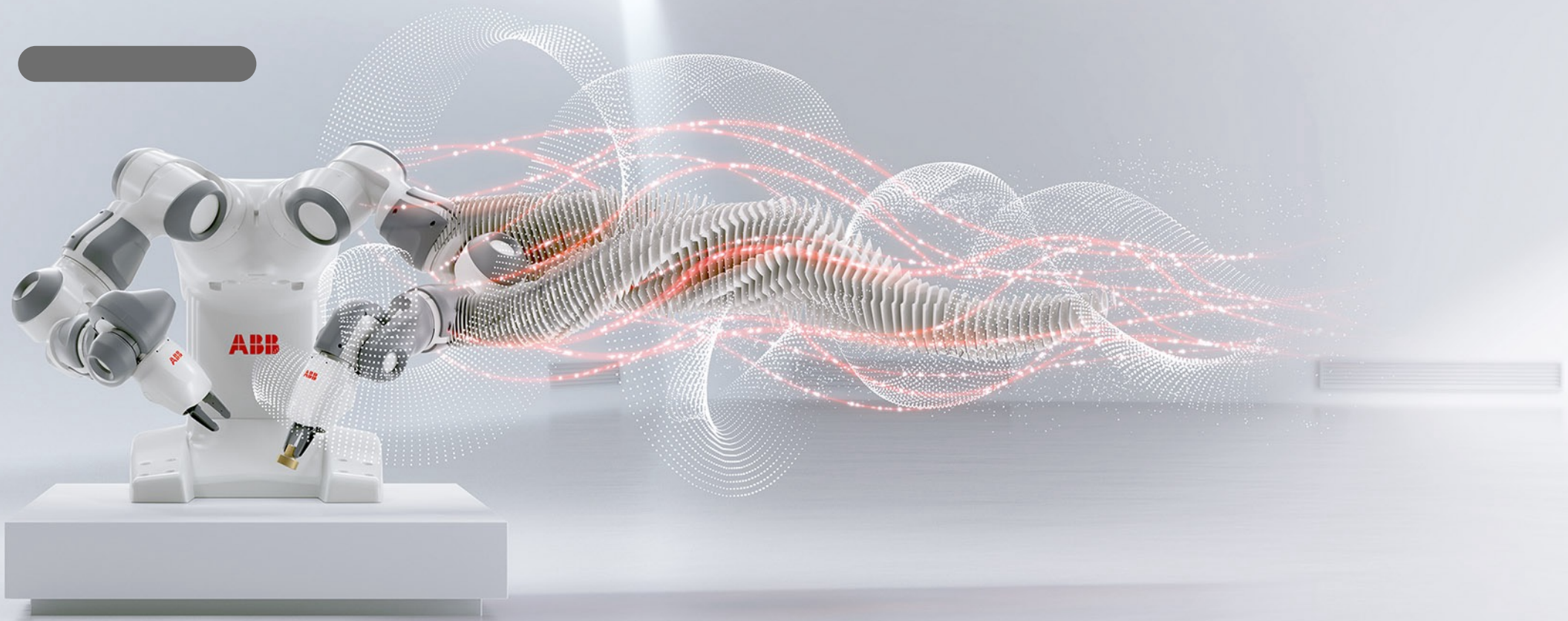


JAN 11TH 2023

How to apply agile to HW development

Mikko Kaijärvi

TMO, Motion Business

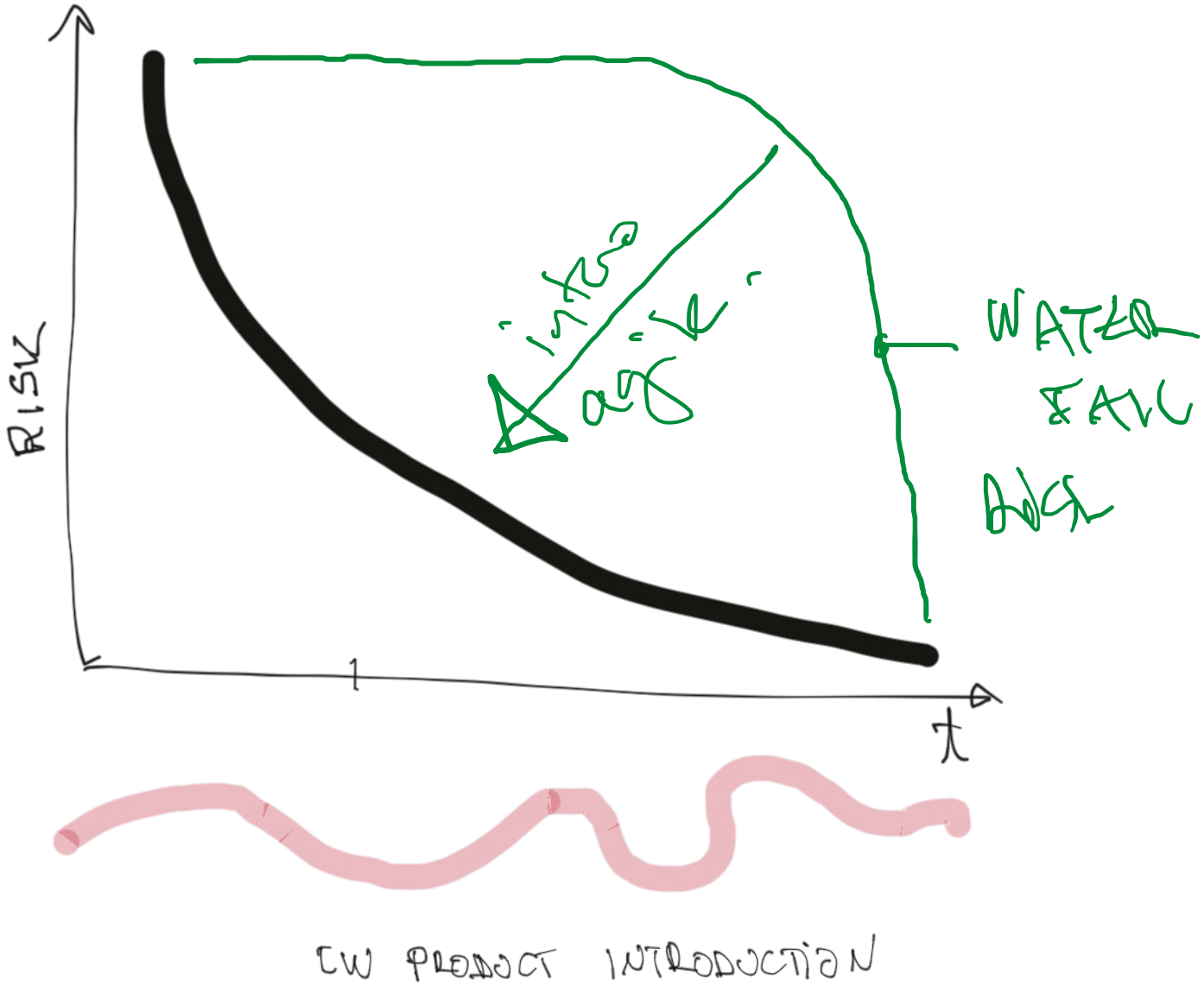
Intro

Risk – System – Small things

New product introduction End to End

Agile needs new habits
These can be helped with changing the system

Agile as an approach to managing risk



Agile things
- Experiment with system changes

System change
- Frequently consider new ways of working

Start

Uncover a problem

Identify real valueable thing to solve

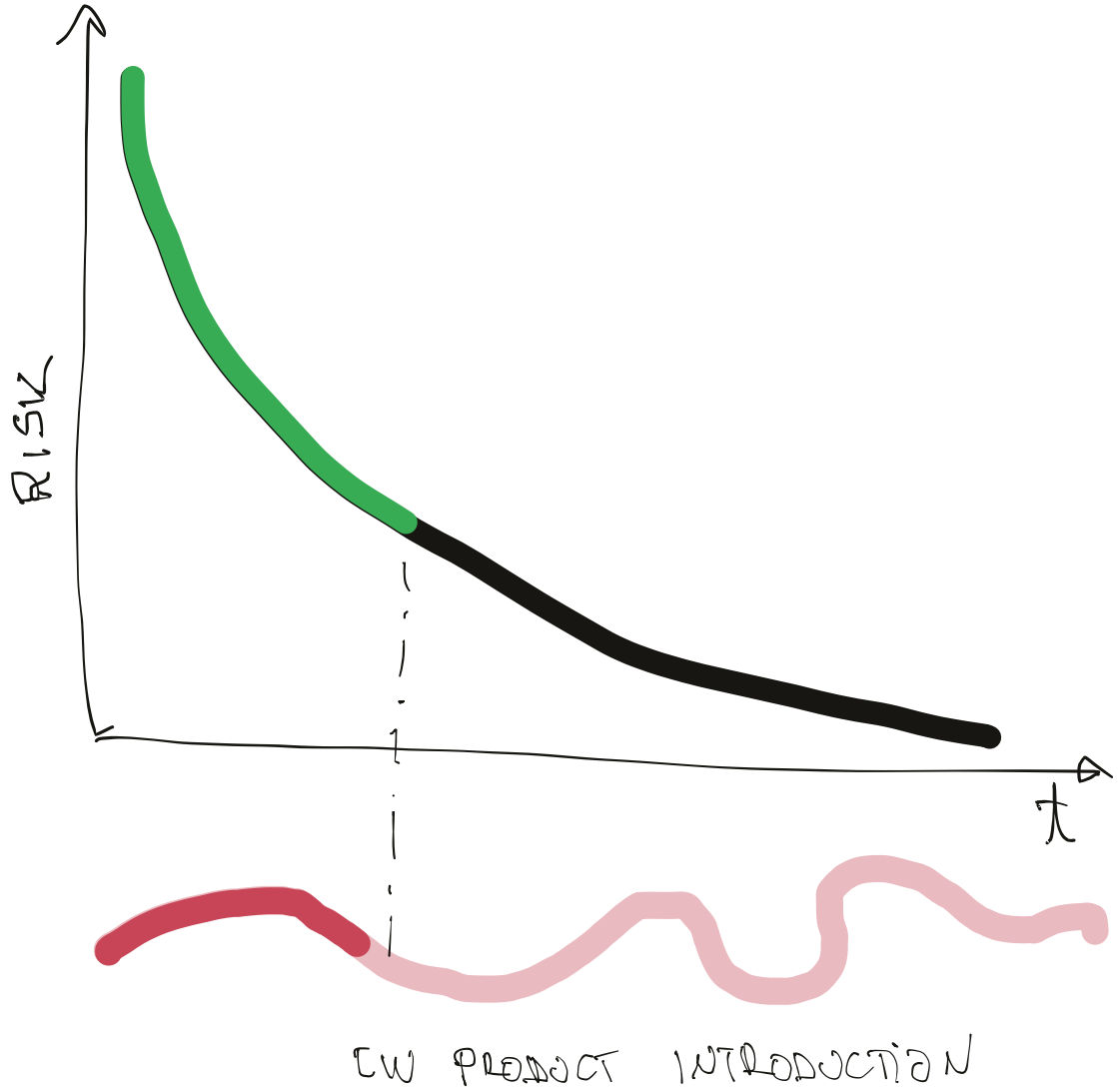
Talk to customers (Customers know how they work)

Iterative (Several takes to understand)

Diversity (people have a unique angle)

Agile things
- Customer collaboration
- Iterative
- Our highest priority is to satisfy the customer through early and …

System change
- Team involve
- Focus on seeing the problem



Solution

Concepts

Simulation (system)

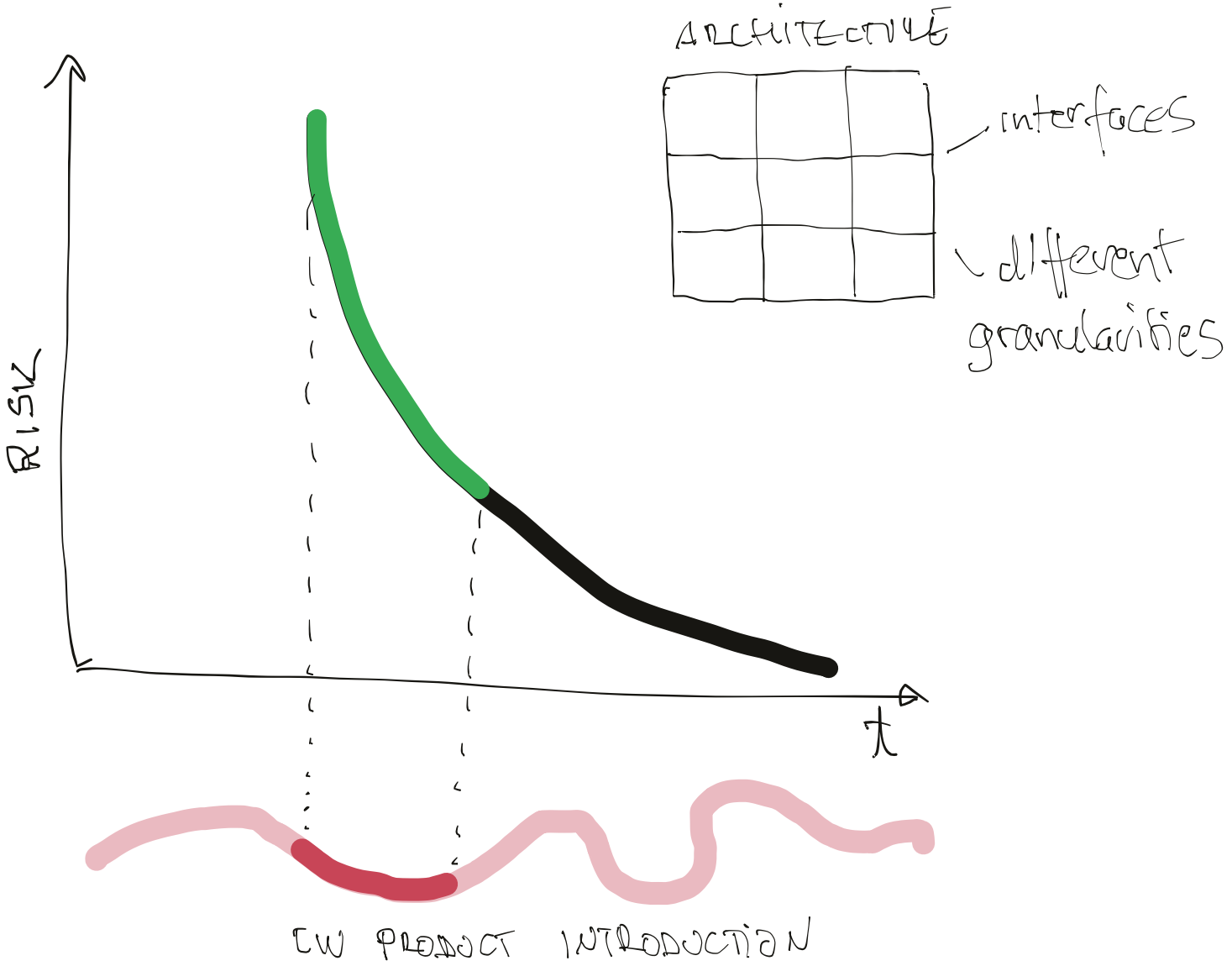
Complexity of a product (physical, cloud, algorithms,..)

Architecture (System, product, discipline, …)

Diversity (people have a unique angle)

Agile things
- Frequent delivery of value
- Maximize work left undone
- Business /dev must work together

System change
- Architecture focus
- Simulation capability
- Team composition



Design

Fast feedback to accelerate learning

Fast feedback (simulation, fast prototyping, pairing…)

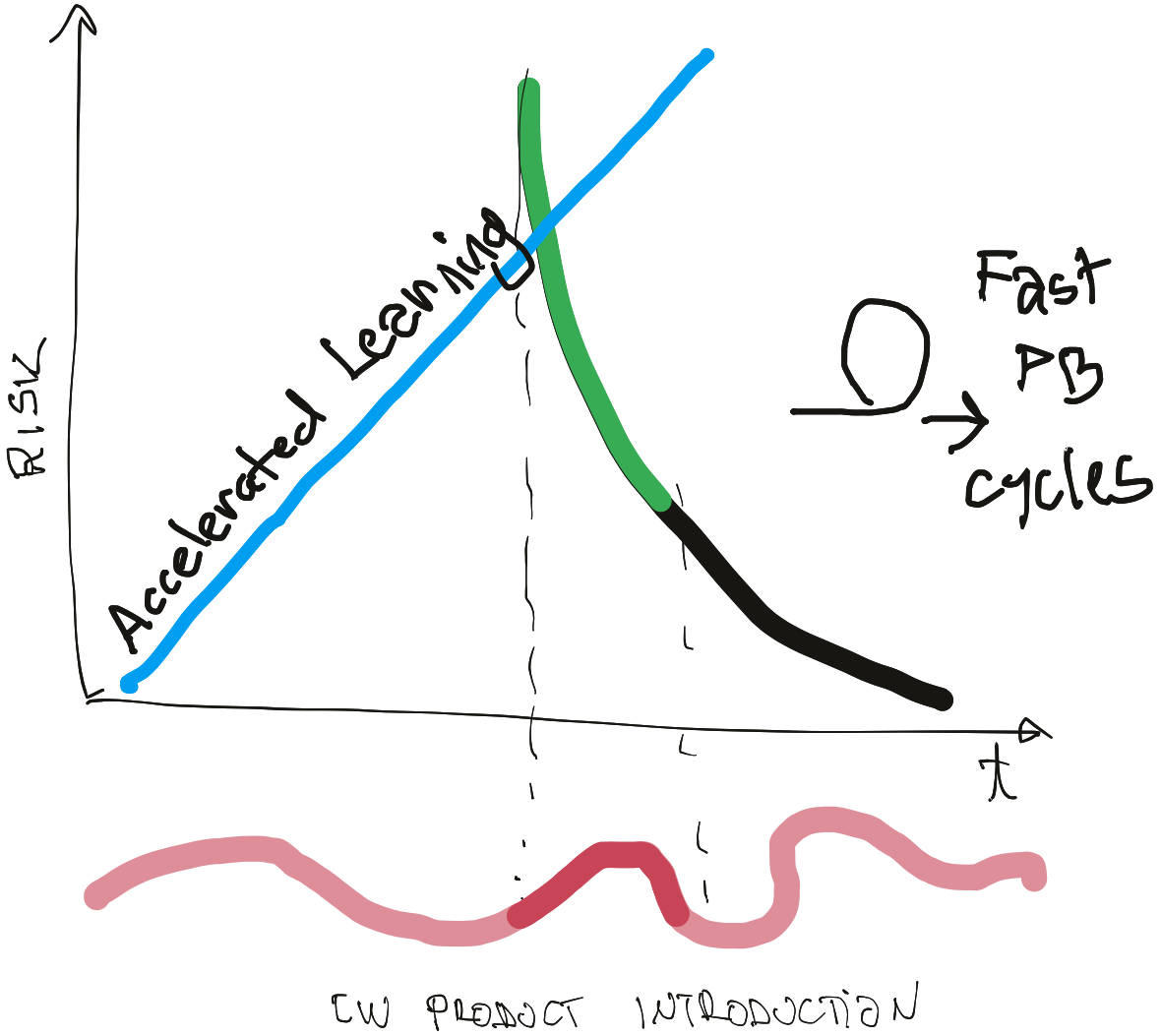
Engineering practices (automate, remove bureaucracy, …)

Visibility/transparency (War rooms, access, same floor area…))

Diversity (people have a unique angle)

Agile things
- Continuous attenetion to tech…
- Maximize work left undone
- Business /dev must work together

System change
- Architecture focus
- Simulation capability
- Team composition



Manufacturing

Agile?

Fast lines (easy start, flexible procuring, …)

Modular robotic lines (easy to modify even volumes,)

Part of design team (allows better solution)

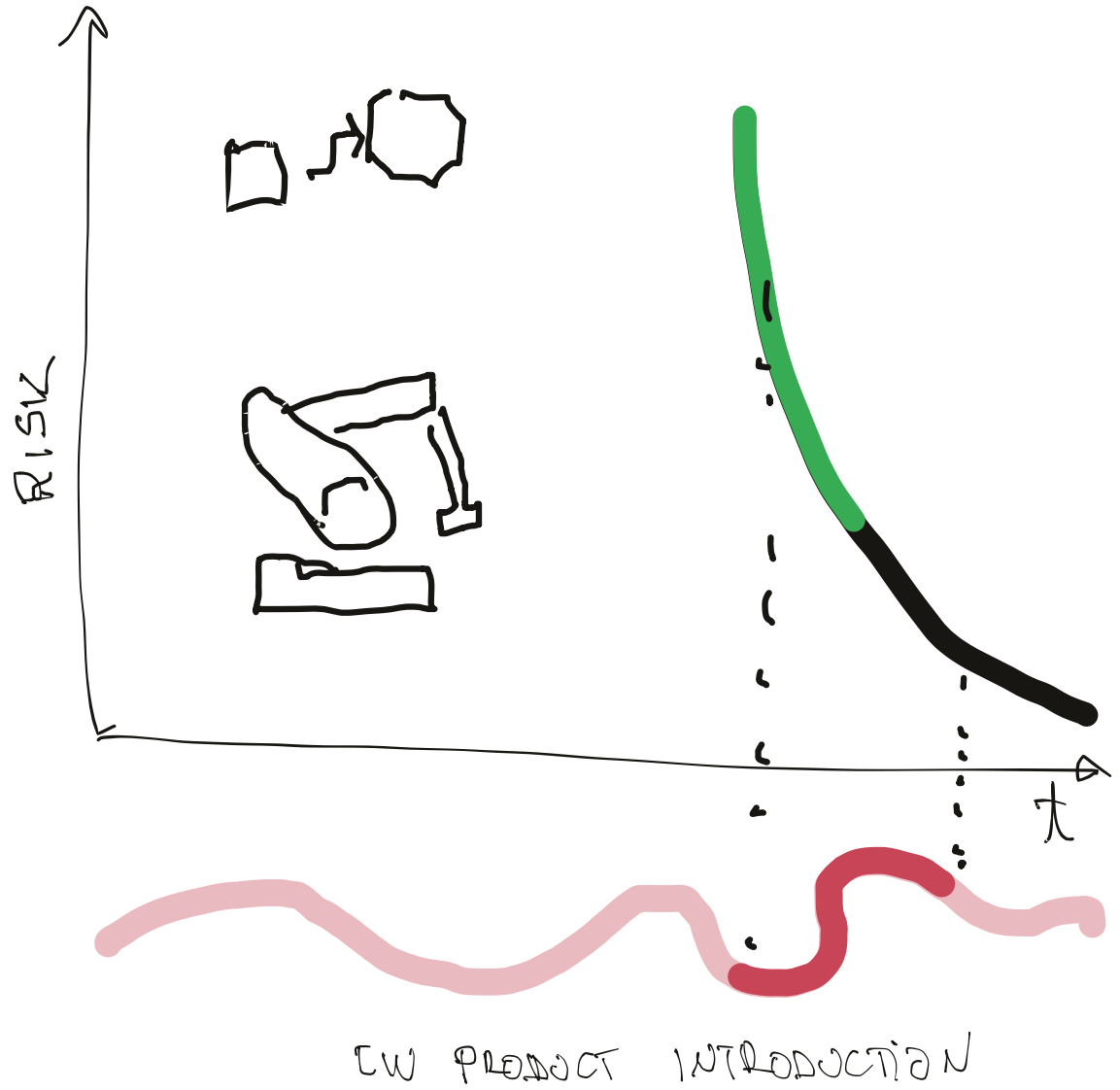
Diversity (people have a unique angle)

Agile things

- Continuous attenetion to tech…
- Maximize work left undone
- Business /dev must work together

System change

- Architecture focus
- Simulation capability
- Team composition



Market Introduction

Gradual

Select Customers (small volume, fast FB, early revenue)

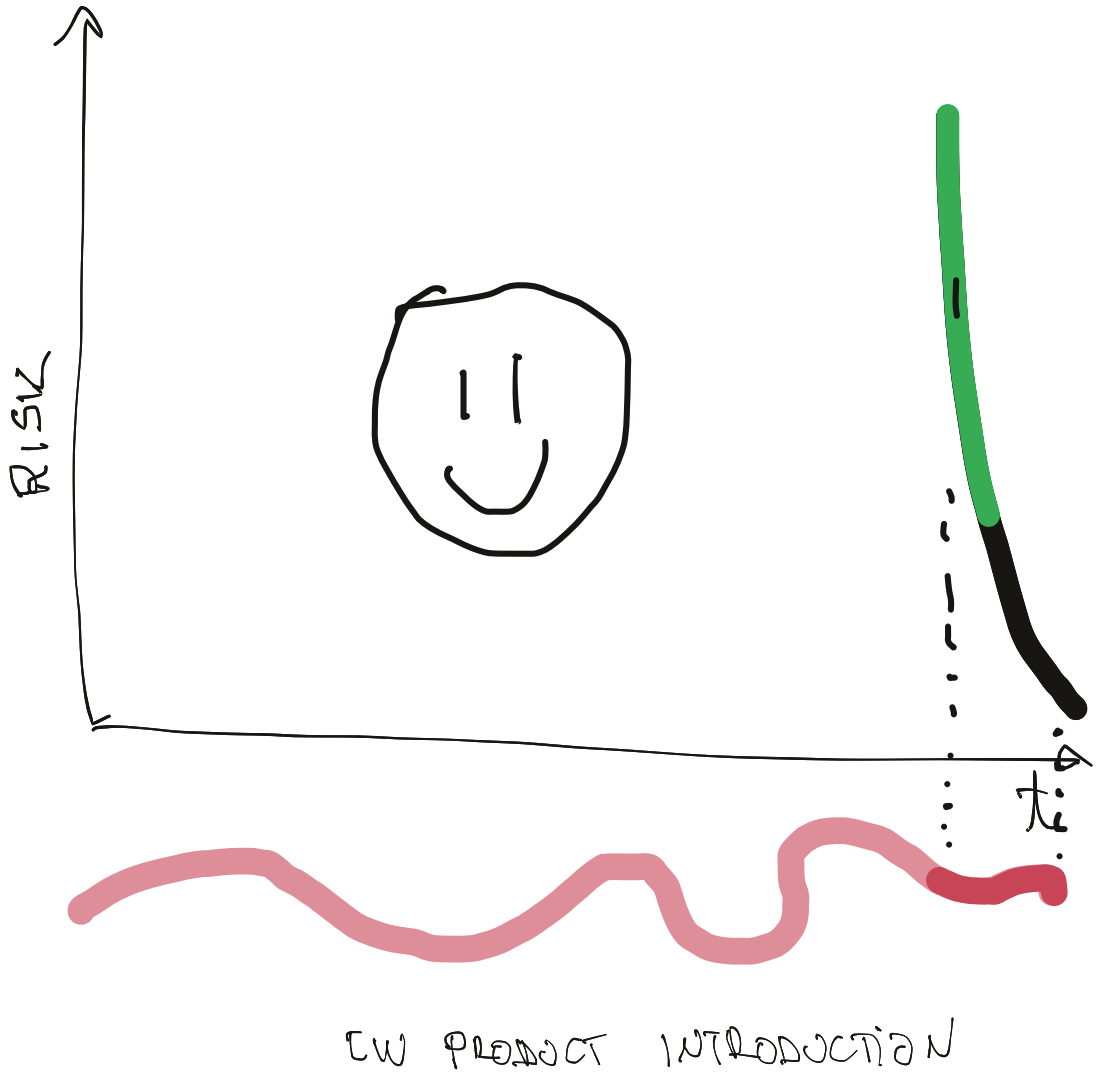
Areas (selected Areas first, limit volume, fast FB)

Updates (MVP, next functionality, …)

Service (field support must be available)

Agile things
- … Valueable products
- Business/dev must work together
- Simplicity

System change
- Allow early installation
- Collaboarate with customers
- Service part of the dev team

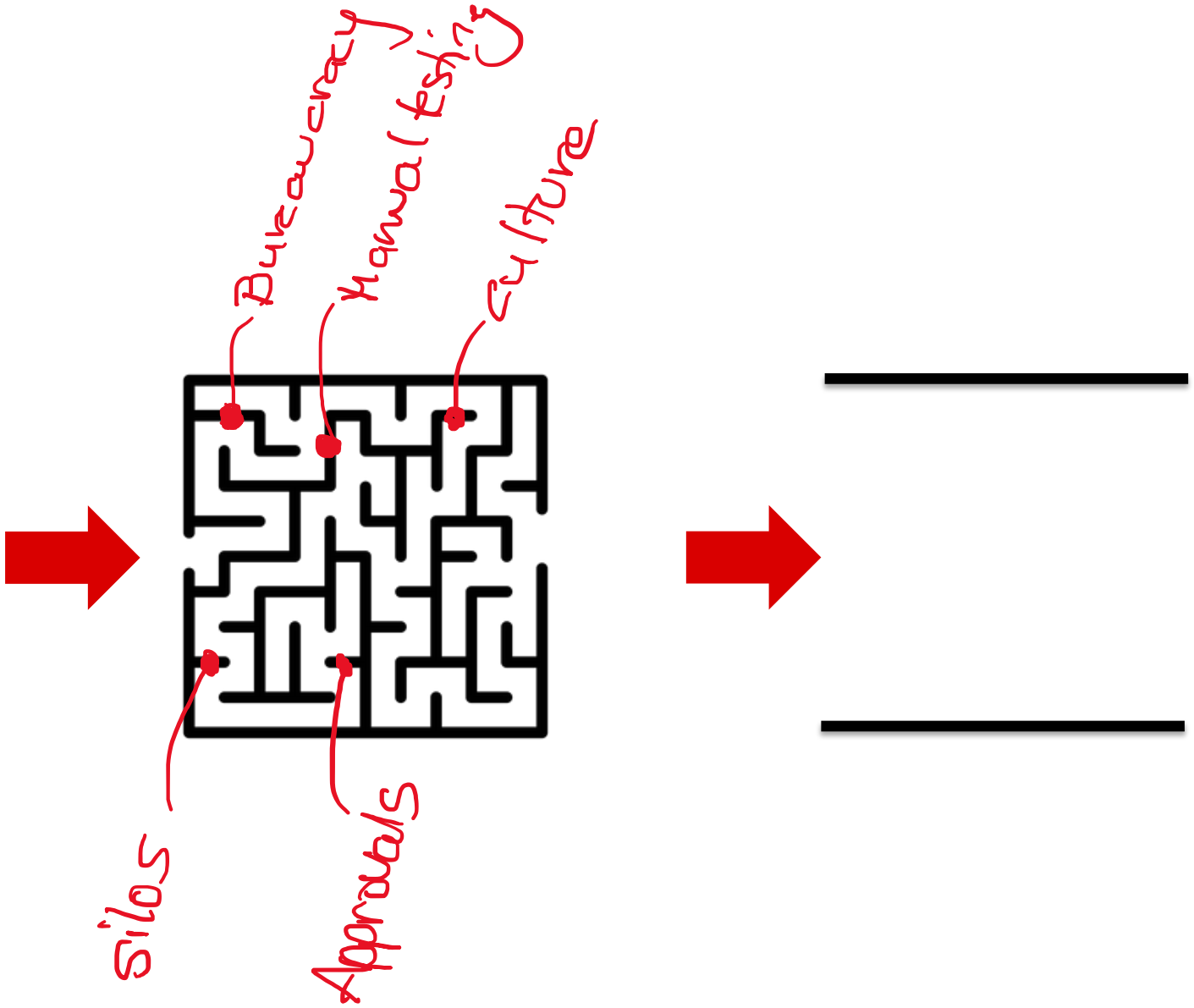


What changes

Gradual

Faster delivery
Better fit
Accelerated learning

Slack (this is a good thing)
Ownership
Happier employees



Basics

Agile and LEAN

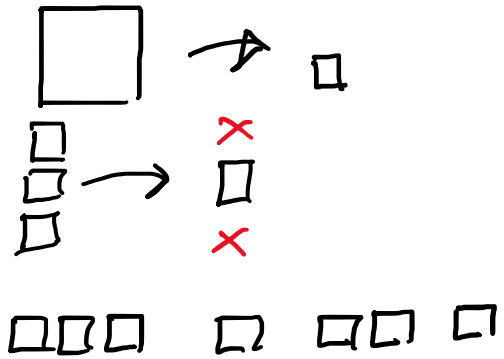
Smaller batch size

Smaller WIP

Slack in system

Faster FLOW

Reduced risk



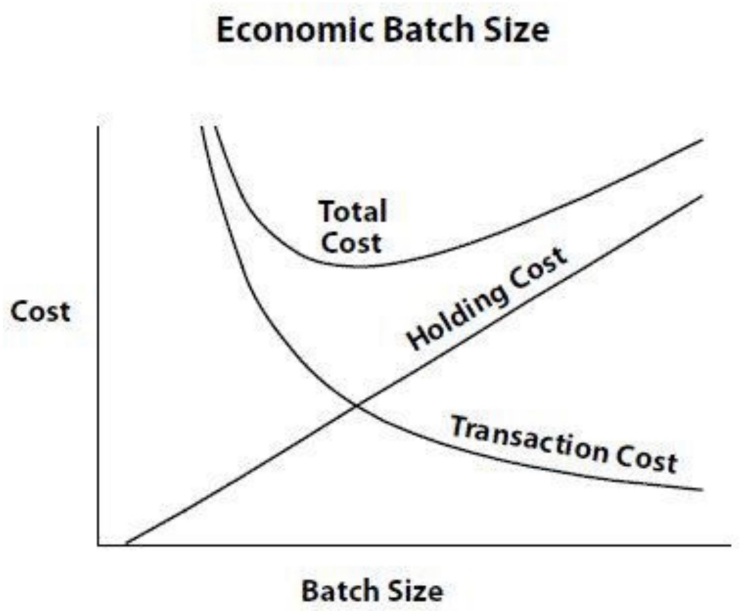


Figure 2-2 Economic batch size is a U-curve optimization. We do not need to find the perfect optimum to capture most of the value.

What has worked for us

Some personal observations

Organic change (allow teams to PULL the chnages they want)

Many small changes seem to be faster and better than one big program

Allow teams to grow change positive view

Support exgineering practices with needed investment (test automation)

Allow teams to take ownership and drive changes – help and coach

Talk – listen – listen - listen (rinse and repeat)


ABB