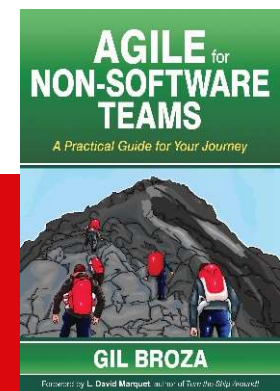
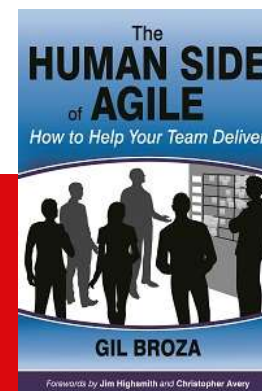
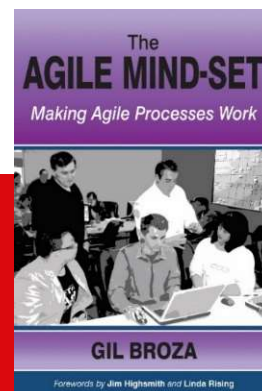


## How Product Management Takes an Active Role in Technical Agility

**Join the raffle for all 3 books:**  
<https://3PVantage.com/PMTA>

Gil Broza



Product management needs and expects Agility.

Teams appear to have it.

Yet, over time:

- Features take longer to build
- Changes become riskier to make
- Quality gets more difficult to maintain

**cost of change**



**safety to make changes**



## Technical Agility:

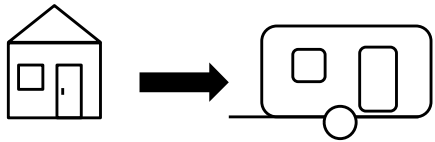
Doing the work in a way that enables overall Agility

Our goals today:

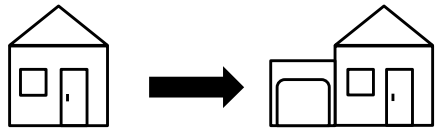
- Explain why good Technical Agility is *not an option*.
- Make cost-of-change and safety regular factors in your decision-making.
- Start creating the conditions that make Technical Agility a reality (or strengthen the foundation you already have)

# Cost of change

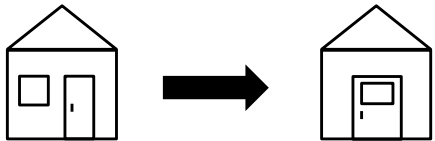
## Four kinds of change



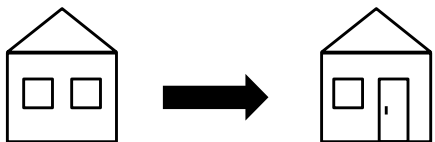
pursue a different outcome or replace the solution



add / change features, elements, or aspects of the solution



make the solution more efficient, robust, or general



correct mistakes or choices made in the preparation of the solution

**Activity (3 min):** Recall a specific solution your team/organization built, that it later decided to change, and that change was costly.

## Those costs...

### **Often include**

- Effort & time
- Lost opportunity
- Lower quality
- Reduced morale

### **Are often due to**

- Complexity
- Reckoning with existing sol'n
- Coordination / dependencies
- Alignment
- Expectation management
- Change management
- Moving targets

# Safety

**engaging without fear  
of harm or retribution**

**doing one's work  
without fear of failure**

## Delivery team members may feel unsafe to do their work

- *“We don’t know whether our changes broke something else – or when we’ll discover that.”*
- *“We don’t really understand the full problem.”*
- *“This is taking much longer than we promised.”*

### **Activity (5 min):**

Reflect on a big change your team is carrying out.

What do they say (individually / as a team) that indicates feeling unsafe?

Write two such statements, each as a full sentence in the first person.

Pair up with a neighbor and choose the two most impactful statements.

“Quad up” with a pair of neighbors and choose the top two statements.



## Other examples

- *“The PO can change their mind easily and quickly; we’re always slower.”*
- *“I hope the PR/code review goes well. I never know what to expect.”*
- *“I don’t know the impacts of my choices (especially on security).”*
- *“This task requires big technical changes. I don’t know how to explain them to Product without looking like I’m making excuses.”*
- *“I’m working on a platform I don’t entirely know or trust.”*
- *“What merge conflicts will I have when I finish my task?”*
- *“I’m pretty sure my team doesn’t understand [this problem] the same way.”*
- *“This change reaches everywhere; what are we missing?”*
- *“If we later decide to undo the change, what will we break?”*
- *“I’m not comfortable showing half-baked work.”*

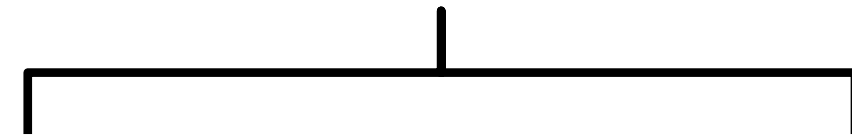
Technical Agility:

**Doing the work in a way that enables overall Agility.**

It is a particular subset of the Agile **principles**, implemented by specific **tactics**, all meant to minimize cost and maximize safety in making changes.

# Technical Agility drivers/principles

## Development



### Rapid feedback

Learn and make informed decisions quickly and frequently

### Small safe steps

Proceed in tiny steps that are easy to test and undo

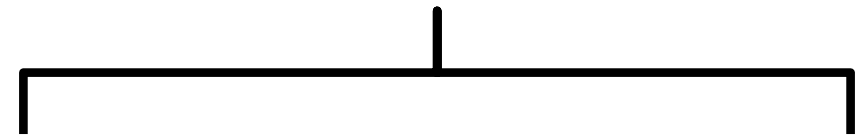
### One thing at a time

Don't start a task while doing another; sequence your tasks

### Clean work

Make it easy to keep working on the deliverable

## Product + Development



### Evolution

Design and develop iteratively and incrementally

### Simplicity

Achieve the outcome in the simplest viable no-regrets way

### Reliability

Finish what you start; don't start what you can't finish

### Shared responsibility

for everything about the work



## Causes of low Technical Agility (TA)

Low TA = low implementation of these drivers/principles

**Discuss with your neighbor (4 min):**

Do you consider your team's TA low?

If so, what can explain it? What prevents higher TA?

# Technical Agility requires intention & mutual commitment

**Everyone** (product, delivery, management) must be **intentional** and **committed** to Technical Agility because it's in everybody's interest.



# To implement & demonstrate commitment to Technical Agility, how should people behave?

	<b>Do</b>	<b>Don't</b>
<b>Product/ Business/ Management</b>		
<b>Delivery</b>		
<b>Everyone</b>		

# Some suggestions that seem to work well

	Do	Don't
<b>Product/ Business/ Management</b>	Factor TA principles in conversations about scope/estimation/dates/splitting Help Dev justify large refactorings	Impose constraints without considering the trade-offs
<b>Delivery</b>	Frequently show & explain how they apply the principles – and their effect Include Clean Work in definition of Done	Act as if “they” don’t get it or care Dramatize issues (“X is built all wrong!”)
<b>Everyone</b>	Be strategic about technical debt Demonstrate (and expect) a high level of communication and transparency Reaffirm explicit collective commitment	Assume that skill & experience are enough for TA Make one-sided assumptions about the product’s future