

How the P's Can Mind the Q's

Program managers,
Project managers,
Product managers

“QA” people,
Testers



James Bach, Satisfice, Inc.

james@satisfice.com

www.satisfice.com

+1 (360) 440-1435

The Tester Heuristic

For good testing, make it someone's responsibility.

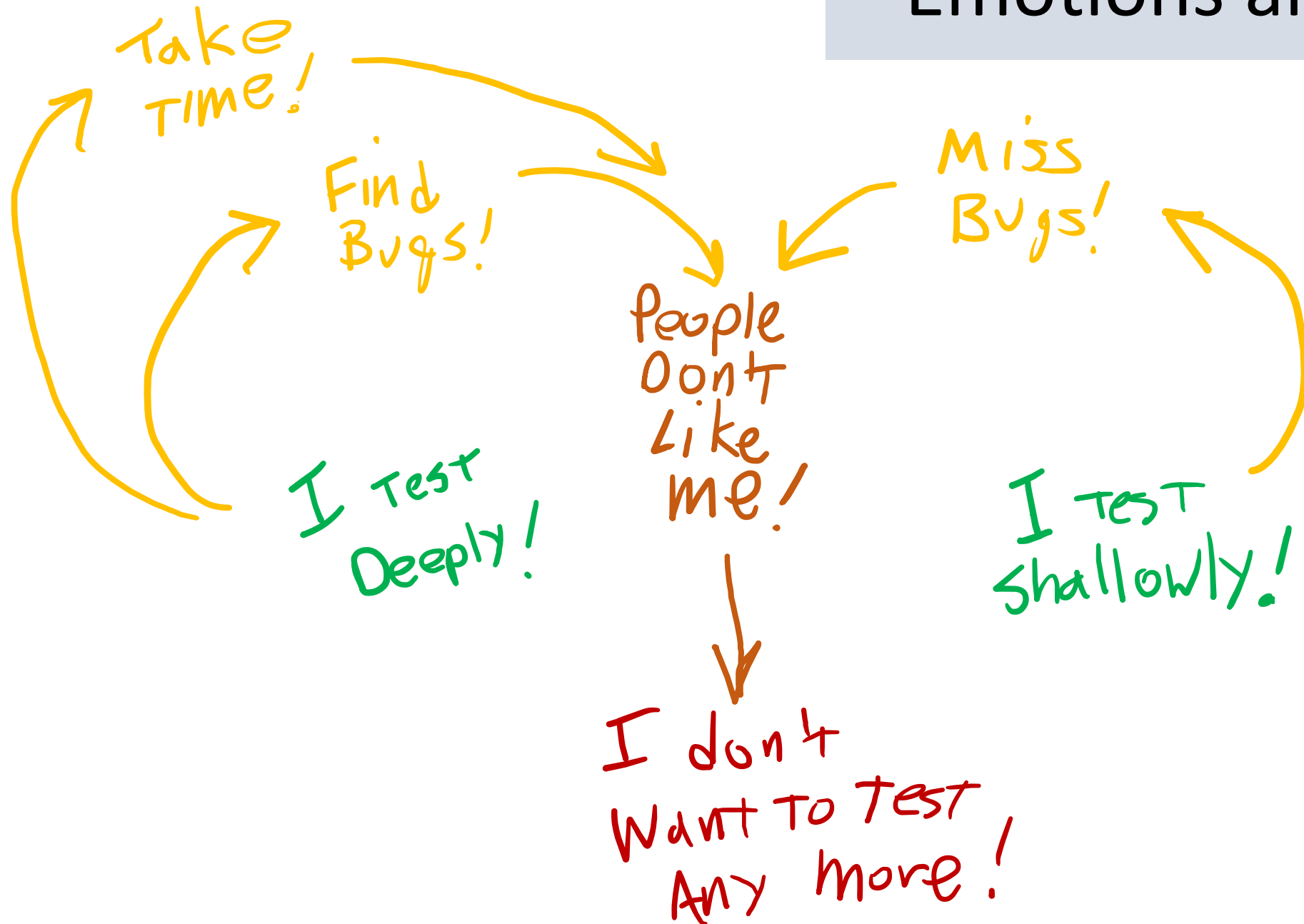
For poor testing, make it everyone's responsibility.

Bandwidth? When testing is “everyone’s responsibility” that means everyone has many responsibilities other than testing. In that situation, testing is never the priority, and what testing happens will be chopped up into small tasks that are frequently interrupted.

Accountability? When testing is equally on everyone, no specific person will feel accountable for the quality of testing. This eliminates any feeling of urgency. “Social loafing” results.

Identity? When testing is a specialty, it also becomes a focus of pride and craftsmanship.

Emotions and Testing



What can you do? Creating a nurturing atmosphere.

- **Empathize** with the plight of the bad news bear-ers.
- **Appreciate the humanity of testers.** Automation is helpful, but nothing substitutes for a tester mindset.
- When they report **good** bugs, **thank them**.
- When they report **bad/invalid** bugs, **thank them** (while rejecting the report).
- **Distinguish shallow from deep testing**, and encourage discussion about what that looks like during planning.
- **Listen to testers explain their strategy**; they need to practice this and they won't unless someone sounds like they care.
- Encourage testers to **advocate for testability** and prompt them to discuss it.

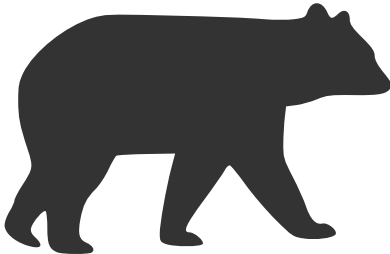
Five Kinds of Testability

- **Epistemic.** If it's already tested, then it's easy test.
example: being careful about change management to minimize need for regression testing
- **Subjective.** If you make the tester better, the product will be more testable.
example: tester learns about the architecture and implementation to test smarter
- **Project-Related.** A well-managed project supports testing.
example: the developers are easy to access and talk to.
- **Value-Related.** It's easier to test if we know what users want and what hurts them.
example: customer behavior is understood and realistic data is available.
- **Intrinsic.** It's easier to test a product that has been designed not to have bugs or hide bugs.
example: the product has comprehensive logging that the tester can access.

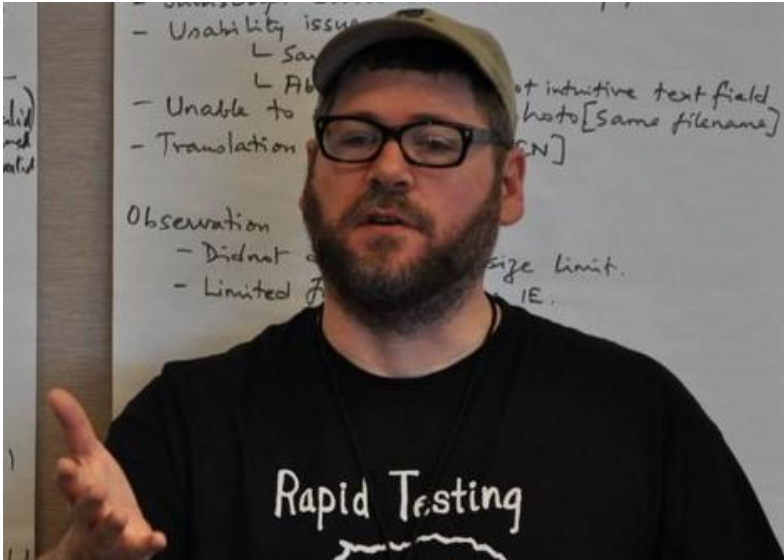
Make Biases Work For You!

- Celebrate the **cleansing fire** of testing. (Aren't we tough?)
- Celebrate the **learning** that comes from mistakes. (Victims? Not us!)
- **Challenge the testers** to find better bugs. (Appeal to pride of identity.)
- **Respect testers**; suggest that the price of respect is excellence. (Ditto.)
- If the product is too good, **put bugs into it on purpose**, so that the testers always believe there is something to be found. (Gives hope.)
- **Be a tough customer** of testing. (When our customers care, we do.)

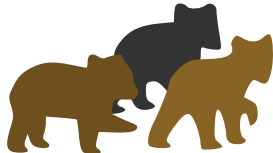
mother bear



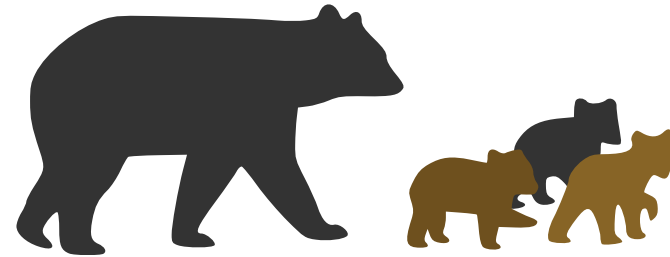
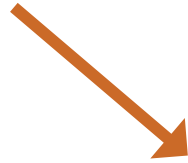
threatening creature



cubs



This is known as “shift left”



development

tester

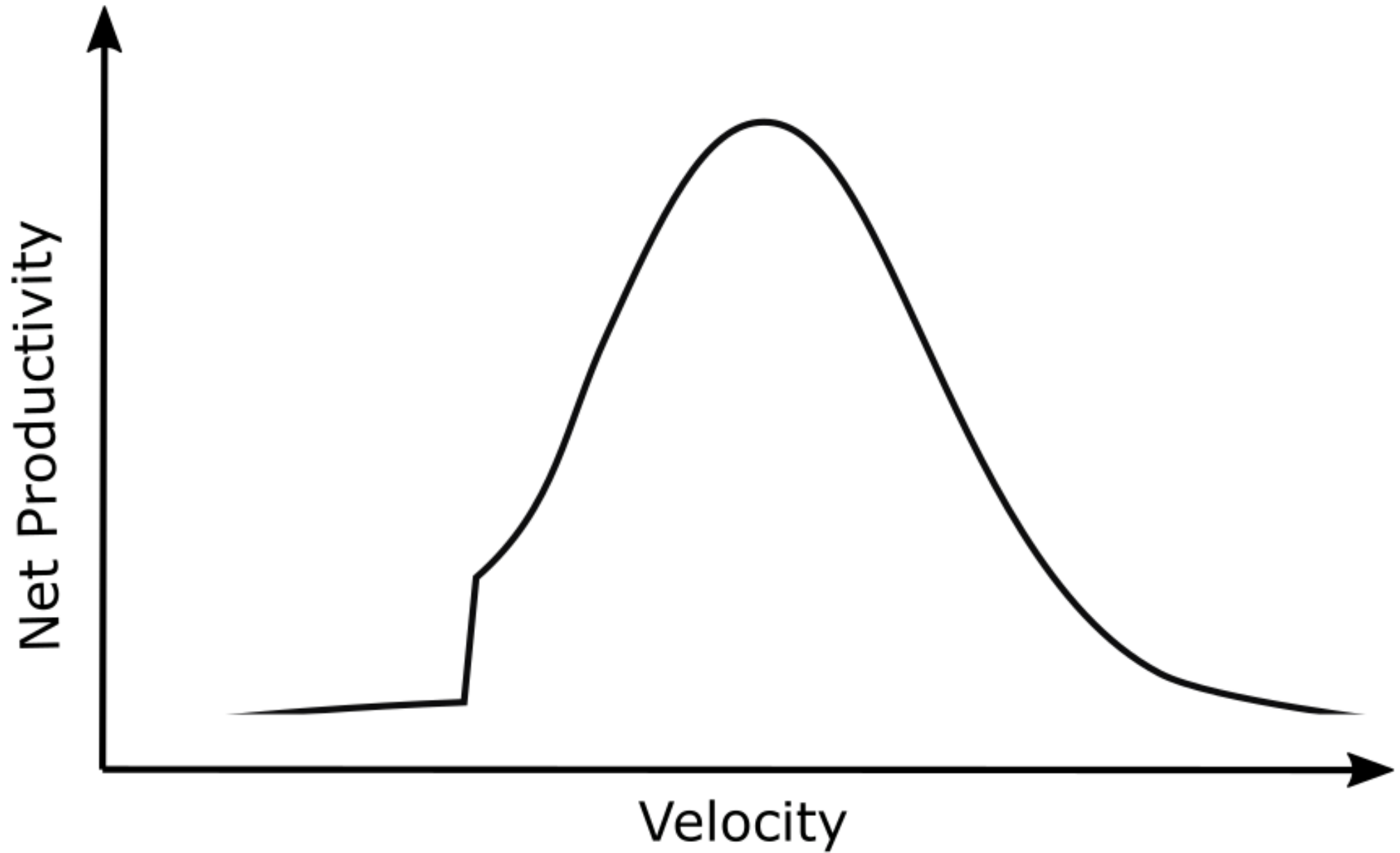
product for delivery

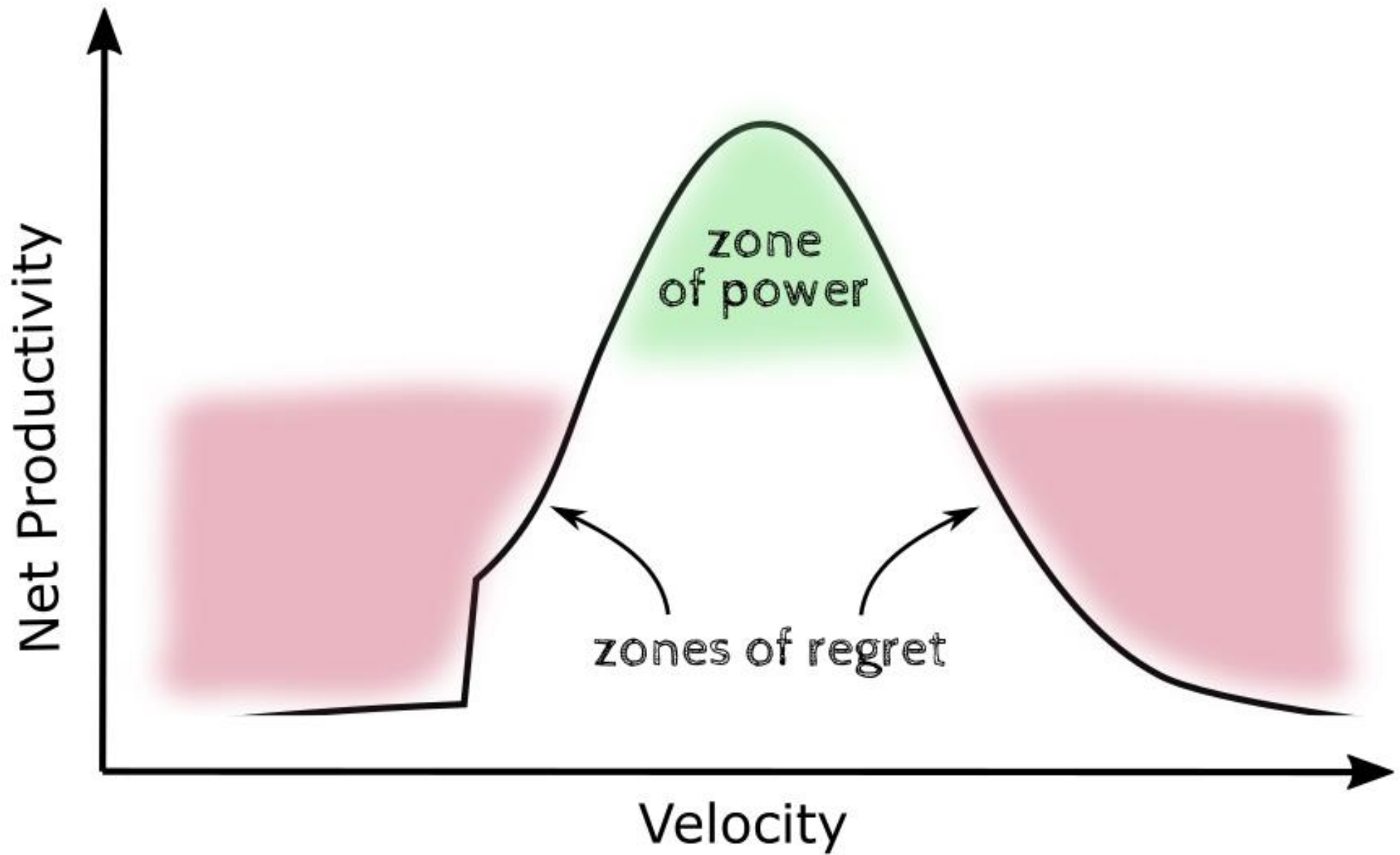
Deep testing before release is a business decision

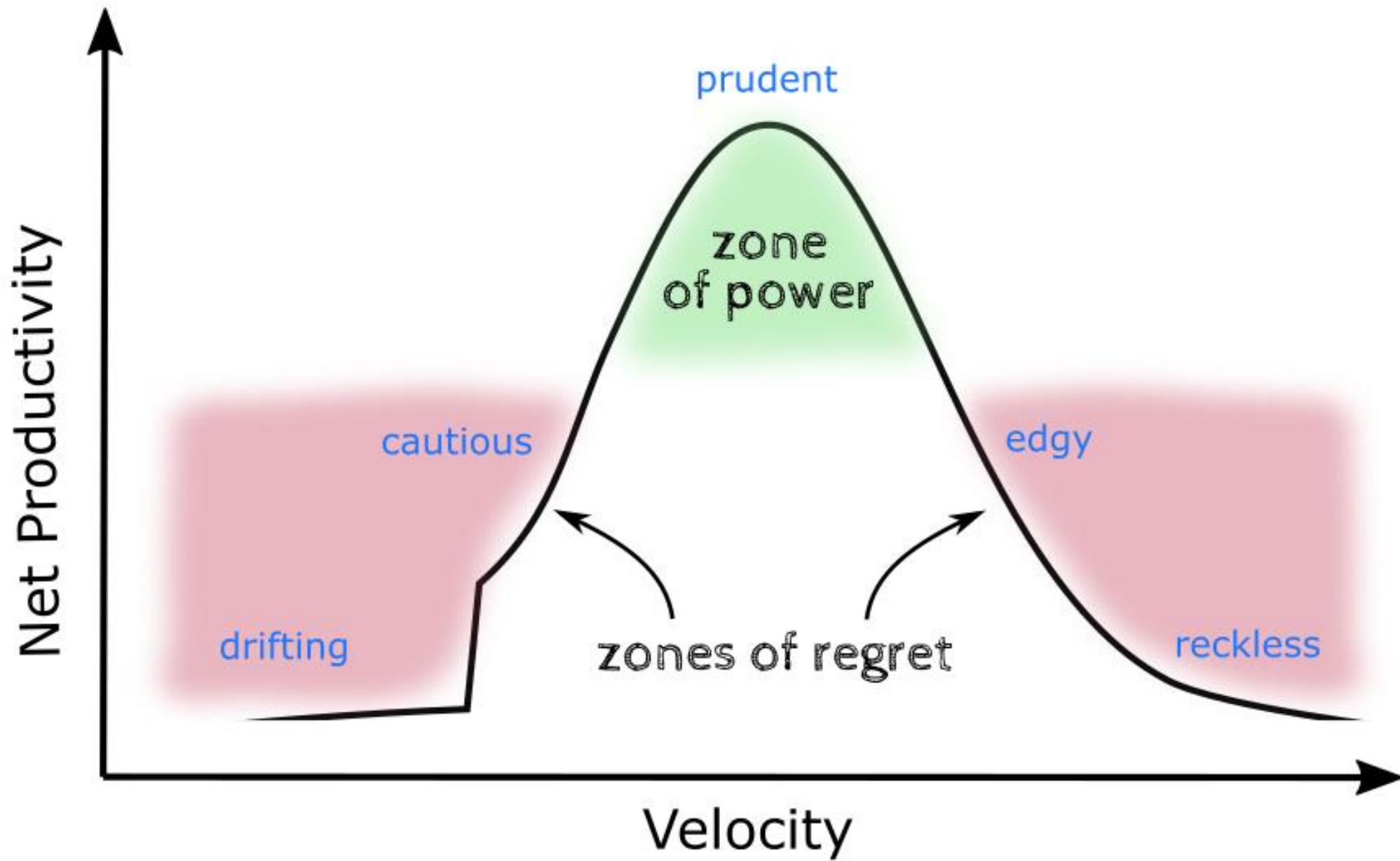
- You give us time to test only if the business senses risk; **this is reasonable!**
- Help us learn to speak about **risk** in specific and compelling terms.

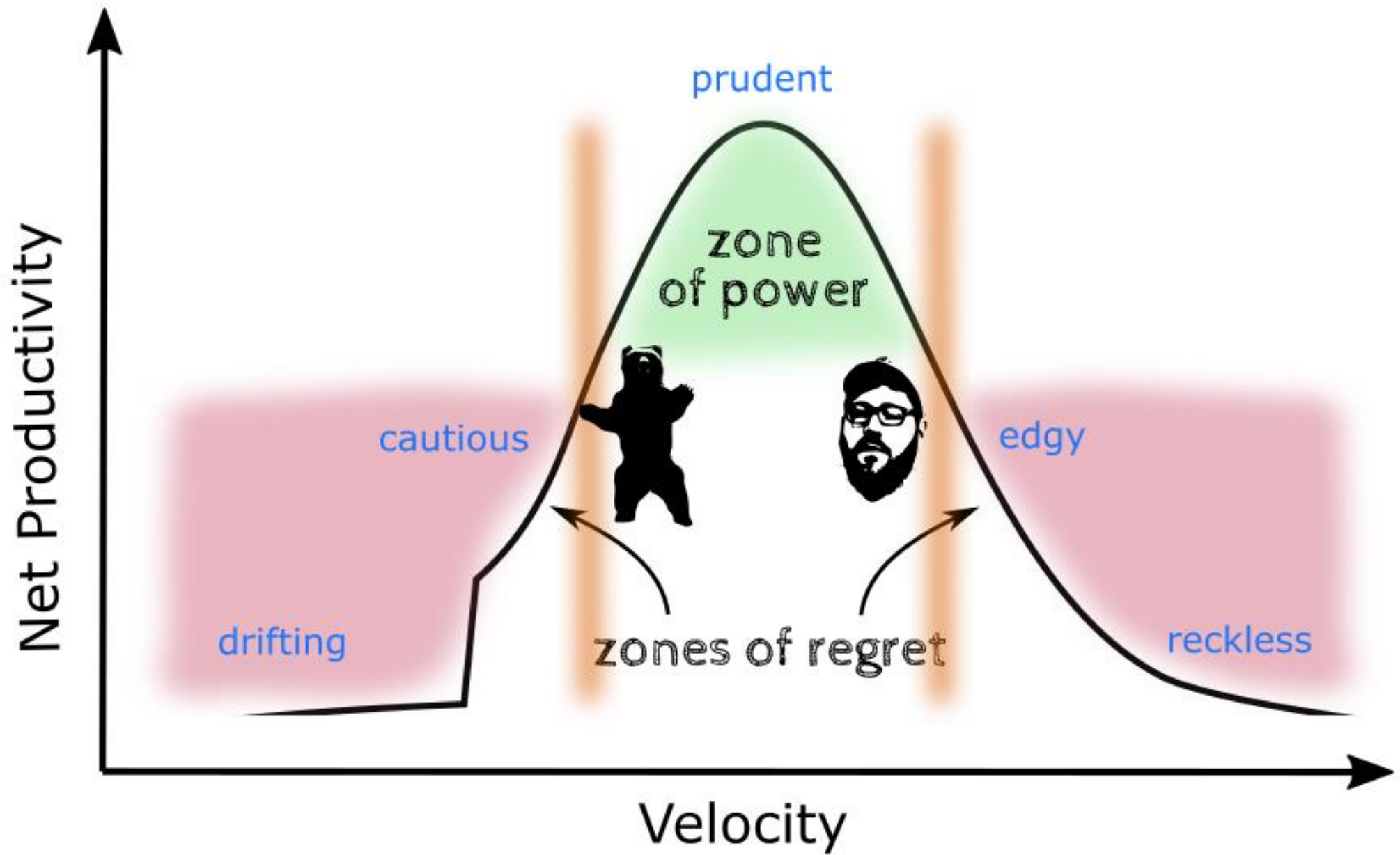
- What sorts of problems will escape purely mechanical output checking?
- What are examples of such problems?
- What makes such problems possible?
- What's the worst that could happen? Who gets hurt?
- What would we do if the worst happens?

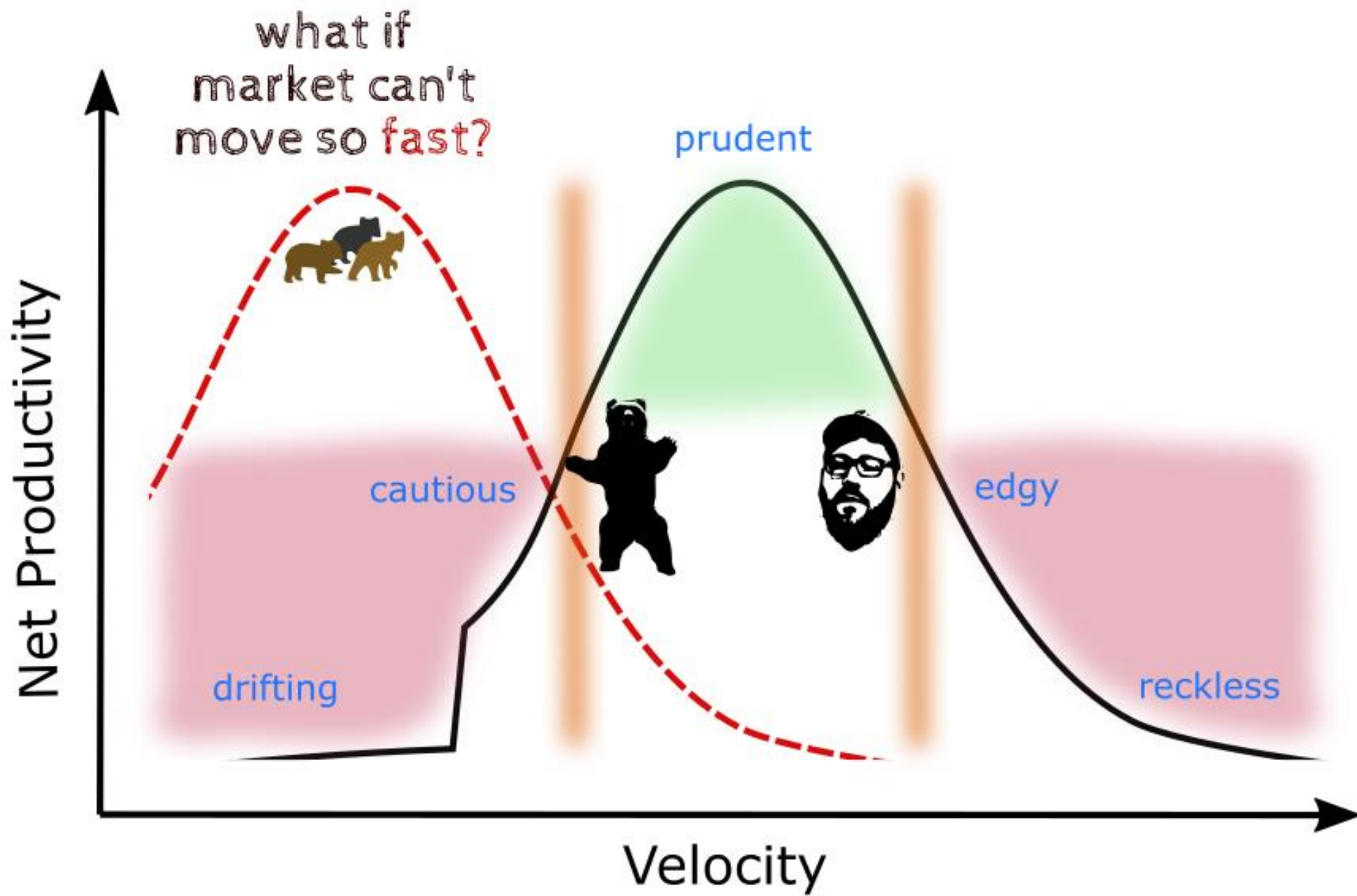
- Risk stories are rarely a matter of reasoning with facts and numbers; but usually the feelings elicited by **vivid and plausible imagery**.
- This is why testers should be **collecting real stories** of real bugs.
- And institute **market surveillance** to see what real bugs escape.

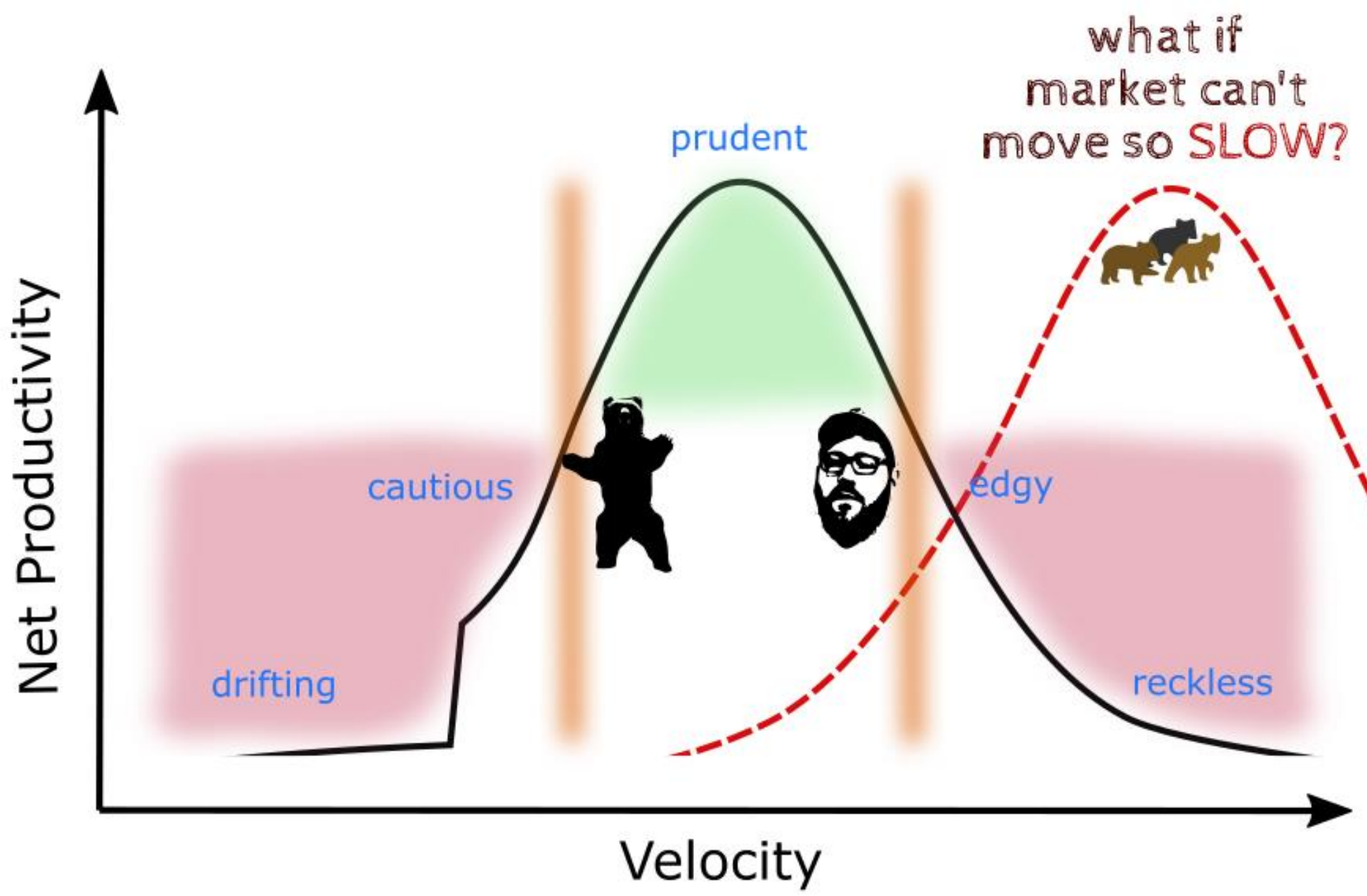












Net Productivity

Velocity

what if market can't move so SLOW?

prudent

cautious

edgy

drifting

reckless