# Agile Release Planning — Context and Basics

April 28, 2025

Joe Little — LeanAgileTraining.com

# Before I forget…

- I will send you the link for the PDF of the slide deck.
- Options:
  - You know I have your email
  - Add to Chat
  - Send me an email: info@LeanAgileTraining.com
- Then: I'll send the link.

# ARP Workshop

- For years, I have done a ONE DAY ARP Workshop.
- We cover all of this content in an Agile Release Planning workshop.
- Often in-house (or public).
- Both ideas, and **practice** doing it with **real work** (maybe your project).
- People say: “should be required”, “essential”.

# Program - 1

- Intro
- Why Planning? — Pros & Cons
- Some Context — 6 Months
- How Much Do We Know?
- Planning Helps Us Learn
- The Context of Project Portfolio Planning
- Key Ideas
- What's changed?
- Some actions!
- Questions?

# Caveat

- “Don’t believe half the lies they tell about me.” Yogi Berra
- “In theory, there’s no difference between theory and practice. In practice, there is.” Yogi Berra
- We make the ideas specific.
- Recommend: 6 months, 7 people, 1 day, etc.
    - You have to use common sense and apply the specific ideas in your (different) situation.
    - Still, they are likely good suggestions.

# Why Planning? (“Cons”)

- No one knows the future.
- The customers don’t really know what they want.
- We don’t know everything!
- Change will happen.
- Stuff happens.
- Changes in direction. YAGNI
- Estimates are dodgy (at best).
- They will punish us!

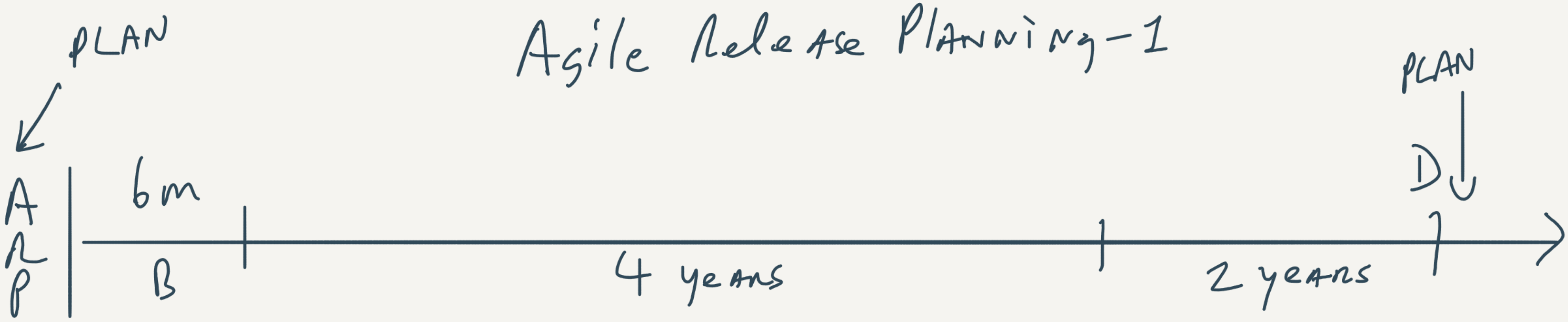
# Why Planning? (Pros)

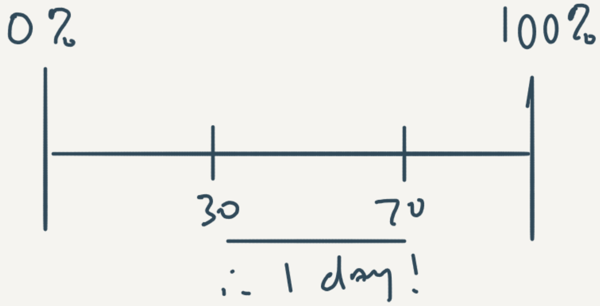
- We must.
- A failure to plan is a plan to fail.
- *"In preparing for battle I have always found that plans are useless, but planning is indispensable." Eisenhower.*
- Planning helps us LEARN.
- Planning helps us adapt later.
- The Team becomes motivated.
- We find out who we are, what we know.

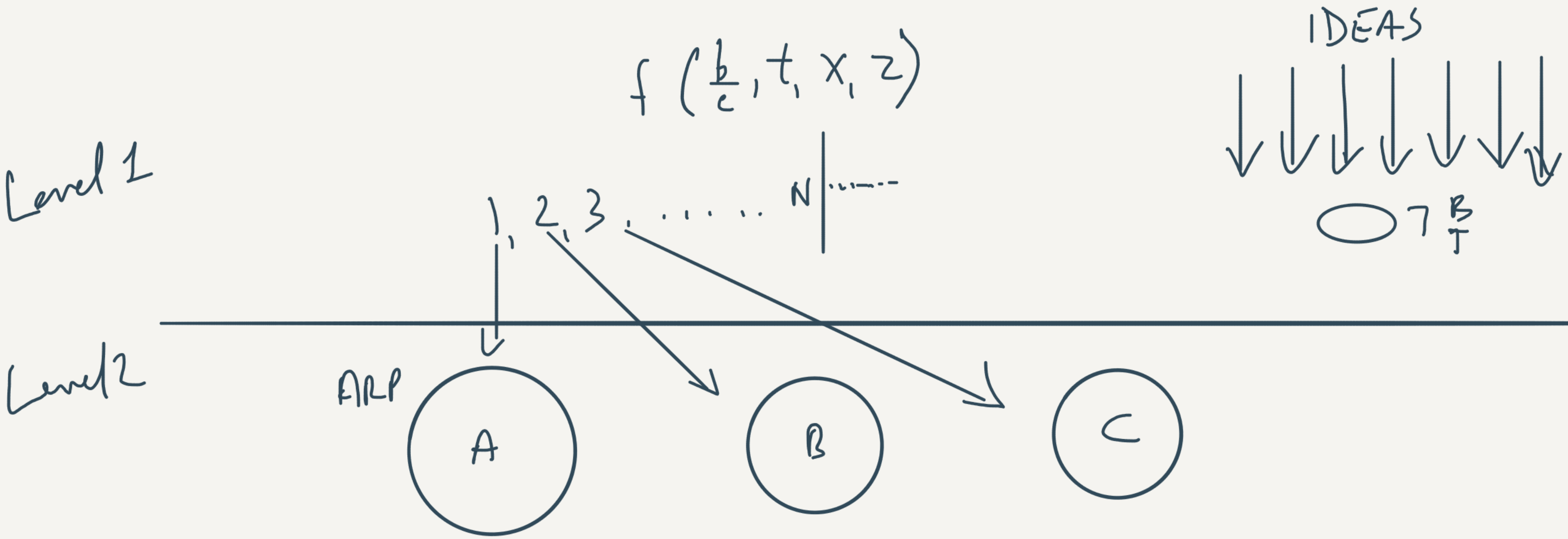
# Context: As small as possible

- 6 months
- Vision can be accomplished
- Multiple releases in that time
- Why?
  - More digestible
  - Better learning because enough and not too much (e.g., tedious)

# Agile Release Planning-1







Now — Some more ideas.

The theory of planning

# How Much Is Known?

- Not 100%.
- Not 0%.
- My guess? 30% to 70% of “useful” knowledge about future
- So… how long to do the *initial* plan?
- We work (think/learn) only 1 day up-front, AND you need to <u>continuously</u> re-plan.

# Learning

- Planning is about Learning
- In all 18 useful domains.
- And about the Future
- “We are chasing the tail of change.”

# Context: Organization Planning

- The “upper level” folks always (?) do planning — usually of the project portfolio.
- Yes, it can be more complex…
- Usually there are many problems between what your organization should do and what it currently does.

# Key Ideas — 1

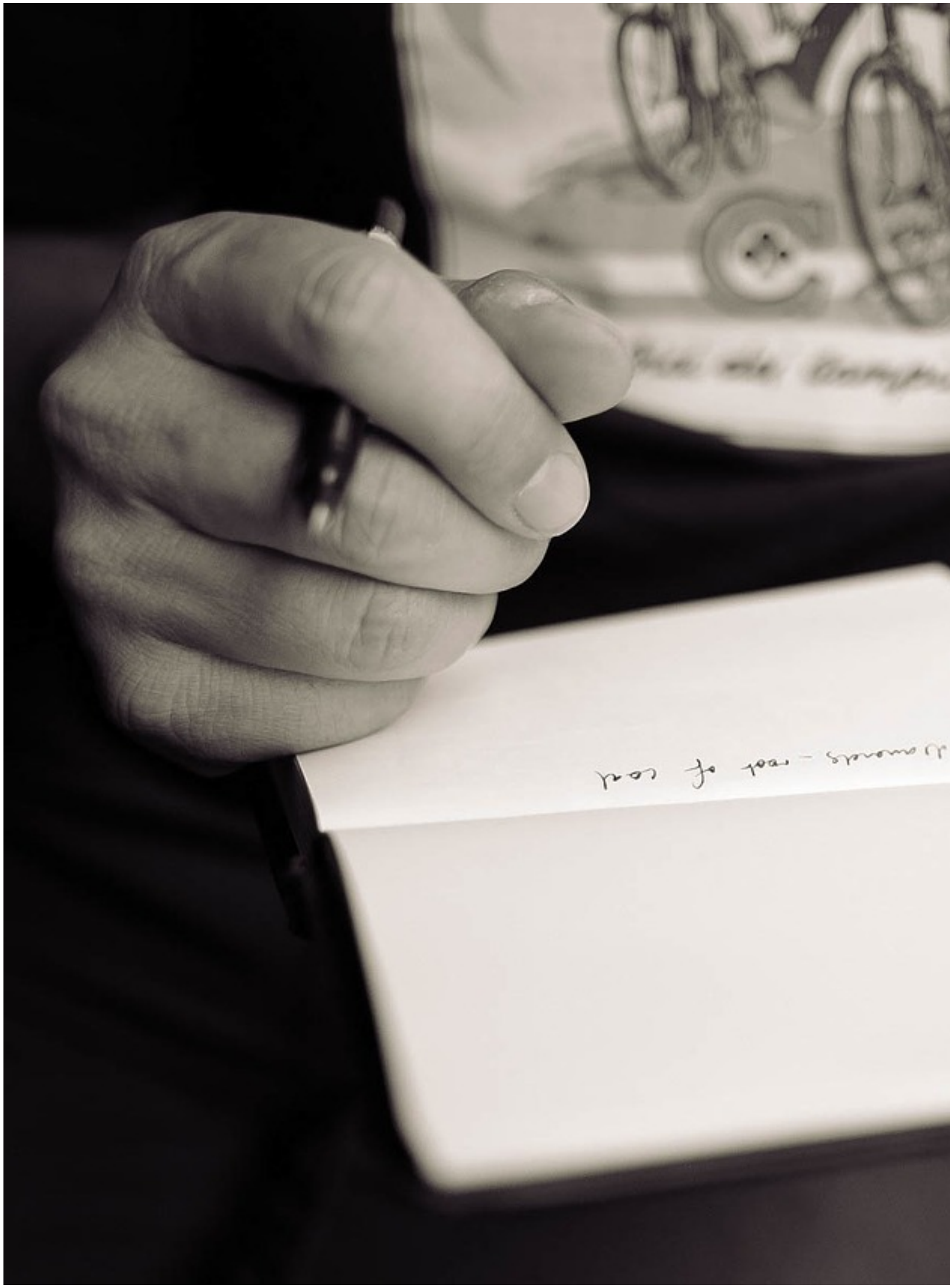
- The work must be important.
- The steering committee estimates quickly.
- Usually **estimates** a scope-date-budget.
- They should try to inspire us.
- We should be able to say “not my thing.”
- The team should (re)estimate (scope - date - budget).
- They should expect our estimate to be … less **inaccurate**.

# A Slight Distraction

- ***"To predict is difficult, particularly of the future."*** —Yogi Berra
- ***"Everybody's got a plan until they get punched in the mouth."*** —Mike Tyson
- ***"The best laid schemes of mice and men oft go awry."*** —Robert Burns
- ***"Let's treat them as if they were real people."***
- ***"People are remarkably good at doing what they want to do."*** —Joe Little
- ***"You live, you learn. You love, you learn."*** —Alanis Morissette
- ***"The future ain't what it used to be."*** —Yogi Berra

# Key Ideas — 2

- We are all ***human***.
- Their estimate is worse than ours.
- Our initial estimate is not so good either.
- We prioritize our stupidity and actively get smarter in the better places.
- You learn faster from action than from thinking in the clouds.



# Key Ideas — 3

- We do the best we can within a time-box. Then we observe and **improve**.
- We improve the plan iteratively and incrementally.
- We improve it as…
  - We get smarter
  - The customer gets smarter
  - Good change happens
  - Bad change happens
  - We improve our process

# Old vs. New

- **Old:** “Stick to the dang plan! Suck it up, buttercup.”
- **New:** “You folks have been getting smarter for 2 weeks. So, of course you can change the plan. What’s the new plan look like now?”
- And the new plan can be BETTER!

# How or Why Better?

- We do less stupid.
  - **Ex:** We don’t build the wrong things as much.
- We improve our Velocity.
- We adapt to change better.
- More BV by the same date.
- We get better at mining the gold-platinum-diamonds only.
- We release earlier (with less?).
- We learn from feedback.
- 7 heads are better than 1!

# Changes In Latitudes, Changes In Attitudes…

- Night and day difference.
- We use planning in a totally different way.

# Goals for the 1 Day

- Get them all on the same page.
- The Big 3:
    1. They all see the same elephant.
    2. They all are more motivated.
    3. They all have shared (more of) the tacit knowledge.

# 5 Actions

1. Change attitudes.
2. Get more clarity on how important this new "project" is.
3. Build the team's initial plan in 1 day.
4. Give initial feedback.
5. Start iterating (start doing real work and learning.)

Now — the Day itself

The practice (suggested)

# Program - 2

- Introduction
- We walk through the day of Agile Release Planning
- Conclusions
- Questions?

# Introduction

- Now I want to talk in a practical way that makes it more concrete.
- To do that clearly, we need a relatively simple situation.
- *Therefore, we will take 1 team of 7 people and look at their work for 6 months (best guess).*

# Introduction (cont.)

- We want them to be engaged the whole time, contributing and also learning from one another.
- They learn better if they are engaged.
- Make it a game, as much as possible.
- Everyone has some pieces to the puzzle. To share.
- Everyone can ***learn.***

# Idea: Quickly

- We are doing the ARP in 1 day only, and really for only about 6 hours of work. (Yes, good breaks, too.)
- So, it's not a lot of time.
- Do they know everything on Day Zero? ***No!***
- "Shut up and drive." (Cf. Rihanna).
- We will revise and improve as we get smarter (every Sprint).

# Quickly (cont.)

- We do the best we can with what we know today.
- Quickly!
- Then we can prioritize our stupidity as a group (team)…
  - And learn!
  - And revise!

# Why So Simple?

- Why use such a simple situation?
- **First:** It is NOT so simple.
- **Second:** It should be at least this simple.
- **Third:** It’s hard to learn if not a simple situation.
- **Fourth:** We rely on the Group to take the learnings and apply them appropriately.

# Preamble

Assume that:

- A manager has given the Scrum Team a set of work.
- The work is expected to be for 1 team of 7 people — about 6 months of work (best guess).
- We have this team of 7 plus 4 Business Stakeholders.
- We have organized the day. Typically in about 3 days everyone can meet. Collocate (I recommend).

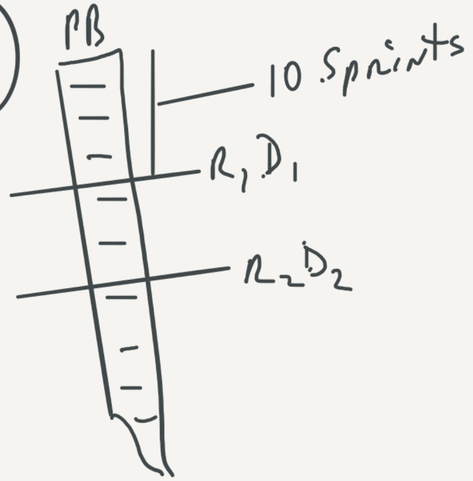
# Agile Rel. Planning - 2

ST-7
+
BSH-4

- VISION
- PROD BKLOG
  - ROLES (5-7)
  - USER STORY WORKSHOP (50)

breaking up by R

- BUS VALUE
  - DRIVERS (3-5)
  - PRIORITY POKER → BVP
- EFFORT $\hat{R} = BVP / SP$
  - DOD
  - PLANNING POKER → SP
- Risks, Depend, Learning, MMFS/MVP, other



- FIX IT PLAN
- comm plan
- contingency

(RE) ORDER THE WORK

Vel = $X_{EST}$

Scope
date

$ $40,000

# Vision

- We create the vision together in 20 minutes on one flipchart sheet.
- It is rather abstract and not so well understood (yet).
- But everyone participates (or at least everyone can).
- Everyone agrees. That is it!

# Product Backlog 1

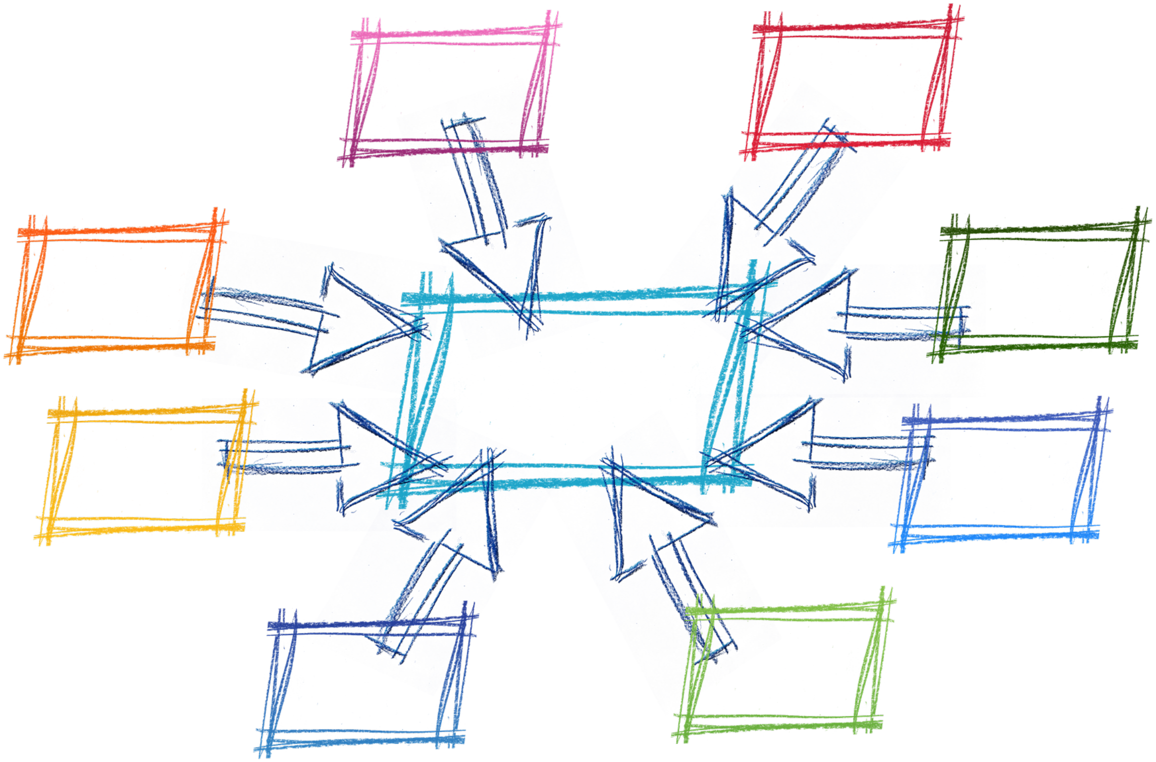
- We identify 5-7 roles.
- These are the actors or personas or roles that we will use in the user stories.
- Our “customers”
- These are NOT the roles in the Team.
- This should only take about **10 minutes.**
- Yes, worth more time later.

# Product Backlog 2

- We then start to create the Product Backlog together.
- Everyone participates.
- Start creating User Stories (3 parts).
- We need about 50 (60) stories.
    - 12 Sprints x 8 stories = 96 —> 50 stories today
- This takes 15-30 minutes.

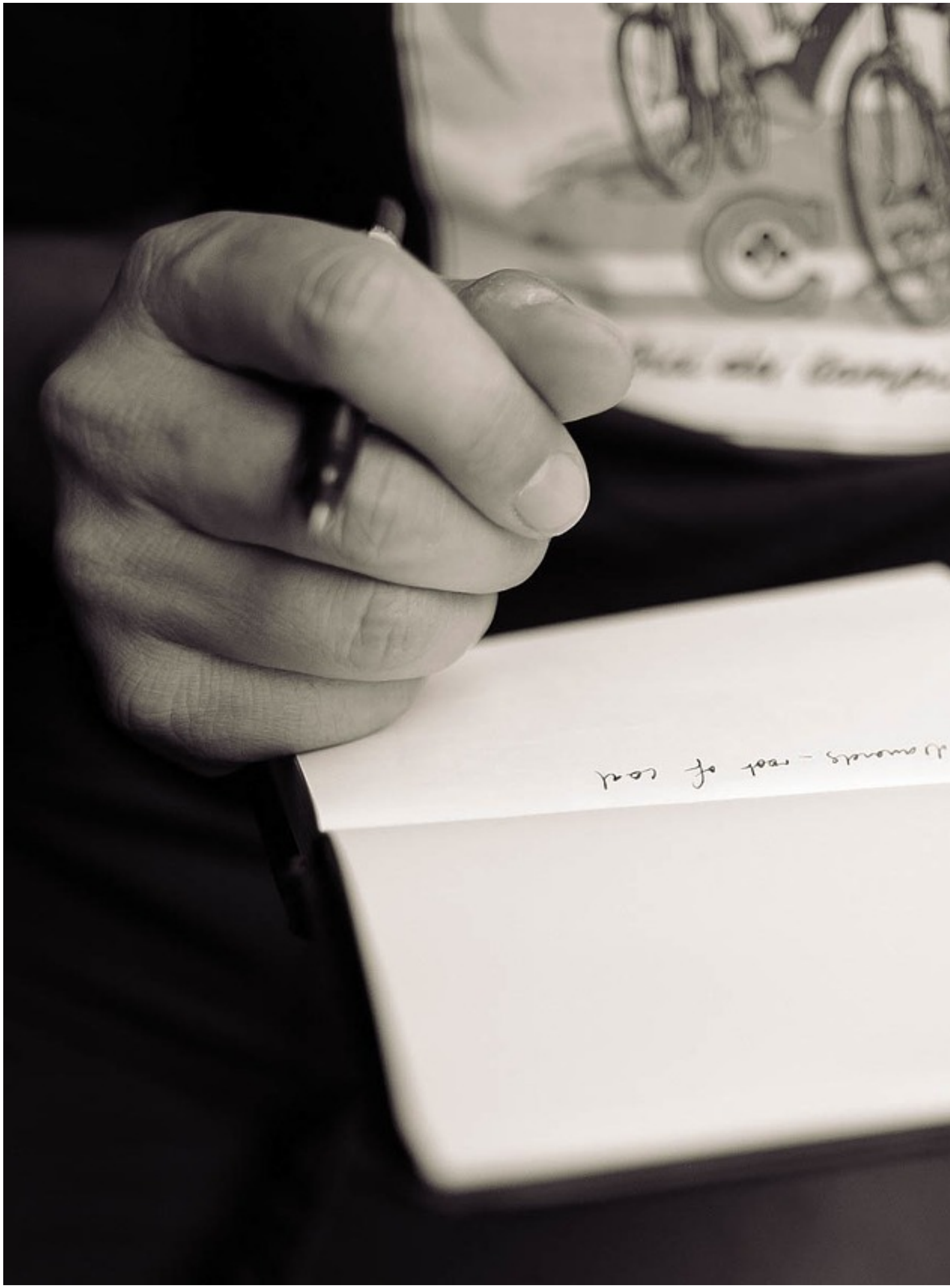
# Business Value 1

- First, we identify the Drivers. 3-5 Drivers.
- The Drivers are ***why*** we are doing this project.
- They are metrics or KPIs or **OKRs**.
- With them we can prove later that we hit the bullseye (if we did).
- They make it much clearer, in numerical form, why we are doing this project or product or work. **Motivation**.



# Business Value 2

- We then do Priority Poker.
- Or, the 5 best people for Business Value use the Fibonacci cards to estimate Business Value Points (BVPs).
- These people should be the 4 Business Stakeholders and the PO.
- The Implementers watch and learn.
- They identify the highest BV story. Give it 100 BVPs.
- All other cards are compared to that reference story, and voted.
- This should take about 1 hour.



# Work 1

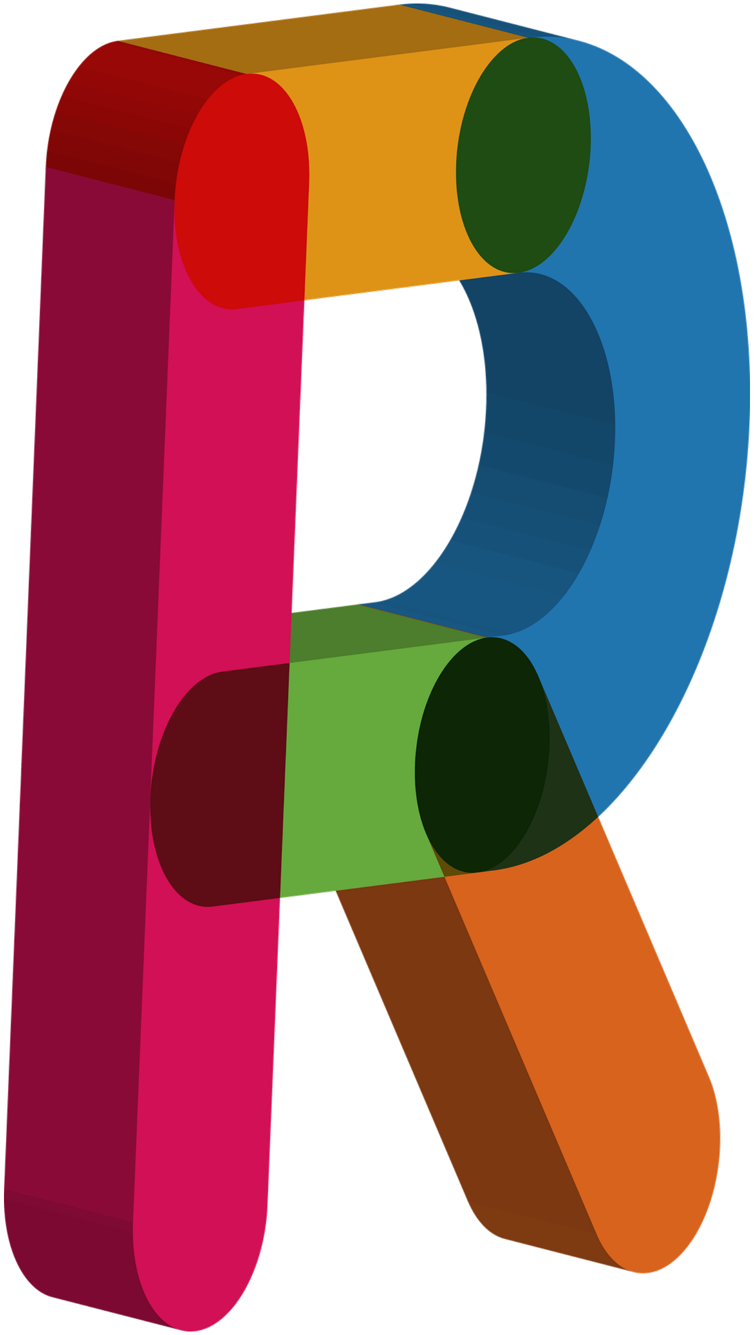
- We review the DOD.
- The DOD identifies the work that happens in the Sprint that we must estimate with the Story Points.
- This helps the estimators (of effort) estimate more consistently.
- Note: The estimators of BV are completely different than the estimators of Work.

# Work 2

- The SM and an outside expert pick the reference story for work. It should be small, typically about 1 “ideal day” of work, in their opinion.
- These details are NOT discussed with the team.
- The 5 Developers use the Fibonacci cards to estimate the work. Everyone votes regarding the full effort.
- This takes about 1 hour plus, maybe 75 minutes.

# R Factor 1

- For each story, we calculate BVP/SP.
- This is called the **R Factor.**
- It represents:
  - ROI
  - Bang for the Buck
  - Cost-Benefit Analysis
  - Identifying the low-hanging fruit



# R Factor 2

- It takes the team about 4 minutes to calculate the R Factor.
- And it takes them about 4 minutes to organize the Product Backlog in R order.
- Everyone participates! (Well, usually about 4 do it.)
- Now the Product Backlog should, for the moment, be strictly in R Factor order.

# Reflection

- The Product Backlog will not look “right.”
- You will see that it will need to change, but in some ways it will make sense.
- Then…

# Re-ordering the Product Backlog

- We want to re-oder the Product Backlog based on other factors, which I propose are:
  - Risks
  - Dependencies
  - Learning
  - MMFS/MVP
  - and other factors
- It will take between 5-30 minutes to re-order the Product Backlog (this time).

# Re-order - How?

- Anyone can propose that an item be moved forward (or backward).
- That person must explain why.
- Anyone can disagree, but must explain why.
- Discussion ensues (often).
- If no agreement is reached fairly quickly, the PO must decide (for now).

# Scope-Date Trade-off

- The Product Owner “draws the line” where the first [2?] releases are. (Commonly 2 releases)
- They must think through which stories must be in the first release — MVP (first guess).
- This is expected to change, but first guess.

# Agile Rel. Planning

ST-7

+

BSH-4

- VISION
- PROD BKLOG
  - ROLES (5-7)
  - USER STORY WORKSHOP (50)

12 Sprints

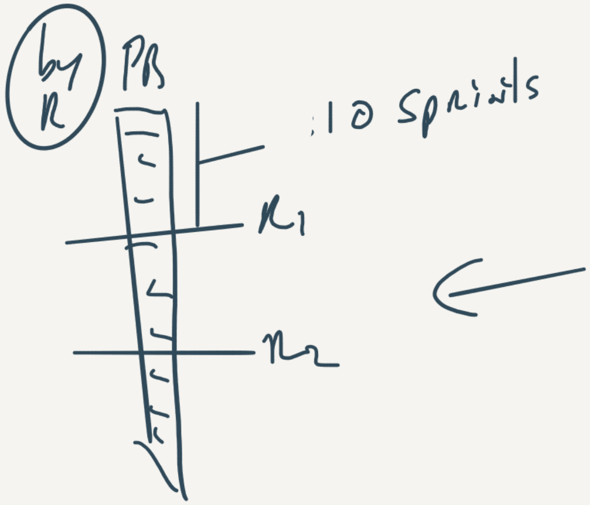
8+ Stories

96 → 50

breaking up

- BUS VALUE
  - DRIVERS (3-5)
  - PRIORITY POKER → BVP
- EFFORT
  - DoD
  - PLANNING POKER → SP

$R = \frac{BVP}{SP}$



- RISKS, DEPEND, LEARNING, MMFS/MVP, other

(RE) ORDER THE WORK

$Vel = X_{EST}$

Scope

date

$

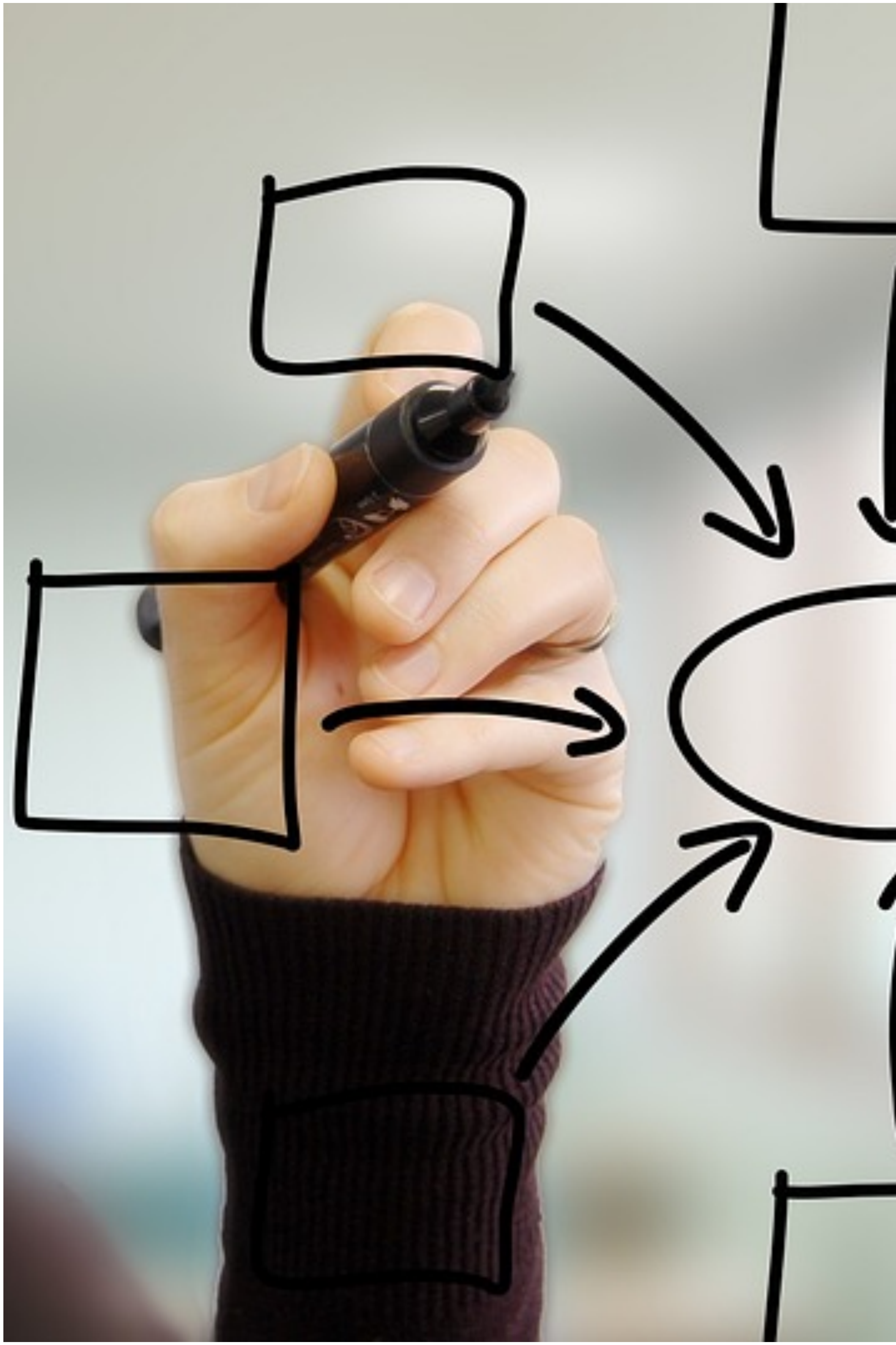
$40K

FIX IT PLAN

- comm plan
- contingency

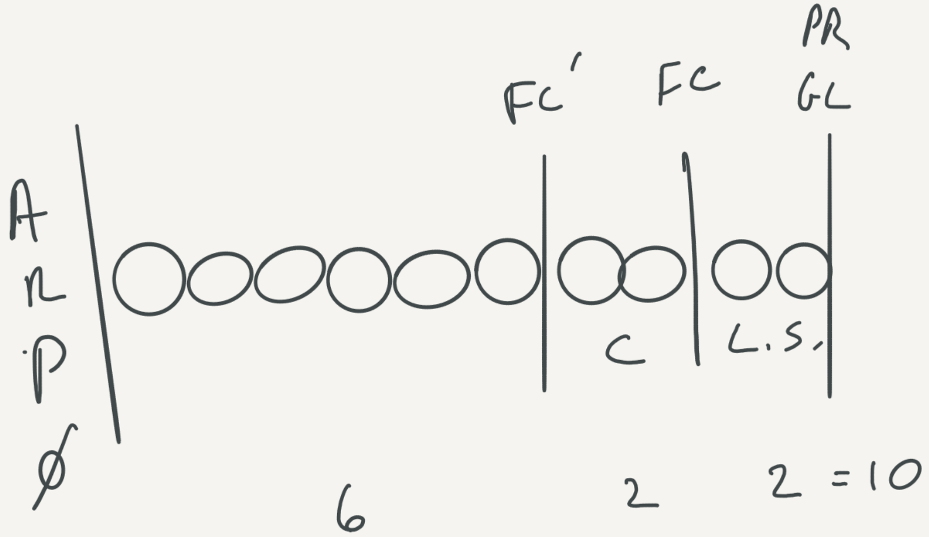
# How Long?

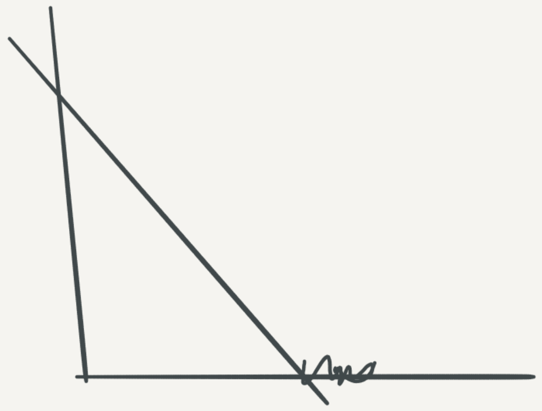
- We estimate the Velocity of the team (a bit of a fancy calculation).
- A SWAG.
- Then we use that to estimate how many Sprints to complete the first release.
- Include contingency and "landing strip!"



30 Stories, 4SP = 120SP

Vel = 20SP ⟹ 6 Sprints





↓

## Final Testing

- fuller INT/REG ↑
- UAT
- Sec Testing
- Load
- Perf.
- UX Testing
- Exploratory

TRAINING

APP'L
DATA MIGRATION

# The Budget

- We can take a first guess at the budget.
- $40K per Sprint x 10 Sprints = a budget of $400K for the first release.

# Finish Up

- Two key discussions:
  - Contingency
  - Communicating the plan changes
- Then: the “Fix-It Plan.”

# Contingency

- At Day Zero, you always need a significant amount of “contingency.”
- Contingency because:
  - Life steals people from us.
  - We ***never*** have all the stories for R1.
  - The stories we have are typically under-estimated.
  - Stuff hitting the golden fans.

# Communicating

- The plan will be changing. They should not judge us by ANY plan.  They should Trust us (Trust is validated).

- So, how do we educate the managers and the customers so they are comfortable with the changes?

- And so that the managers sing that song at the end …

- *“You are the champions, my friends….”*

# Continuous Improvement

- This is Release Plan Refactoring — a.k.a. Product Backlog Grooming or Product Backlog Refinement.

- I recommend 2 meetings per Sprint (assuming a 2-week Sprint).

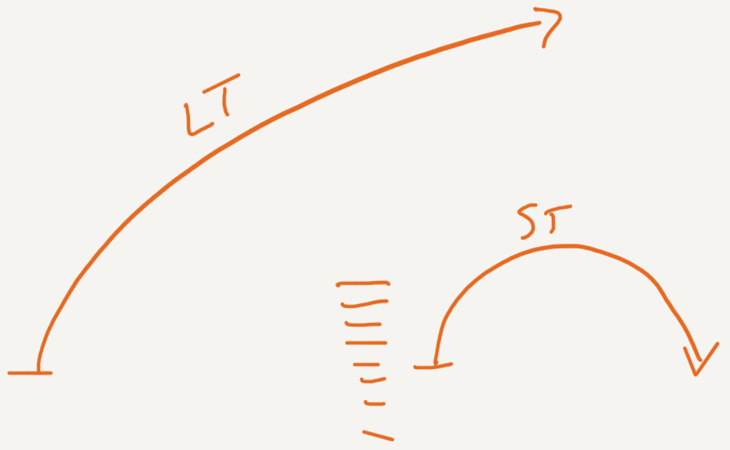
- During the Sprint, the team and the Business Stakeholders are learning things that force the plan to change.

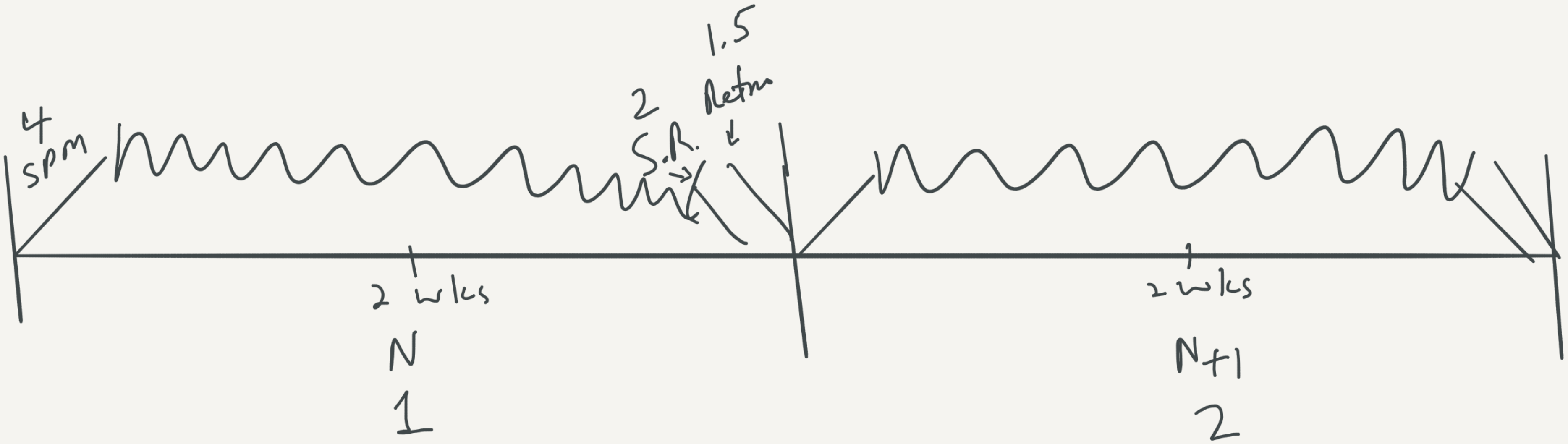
- The changes can be good, bad or both.

# Rel. Plan Refactoring

AKA PB Grooming
PB Refinement

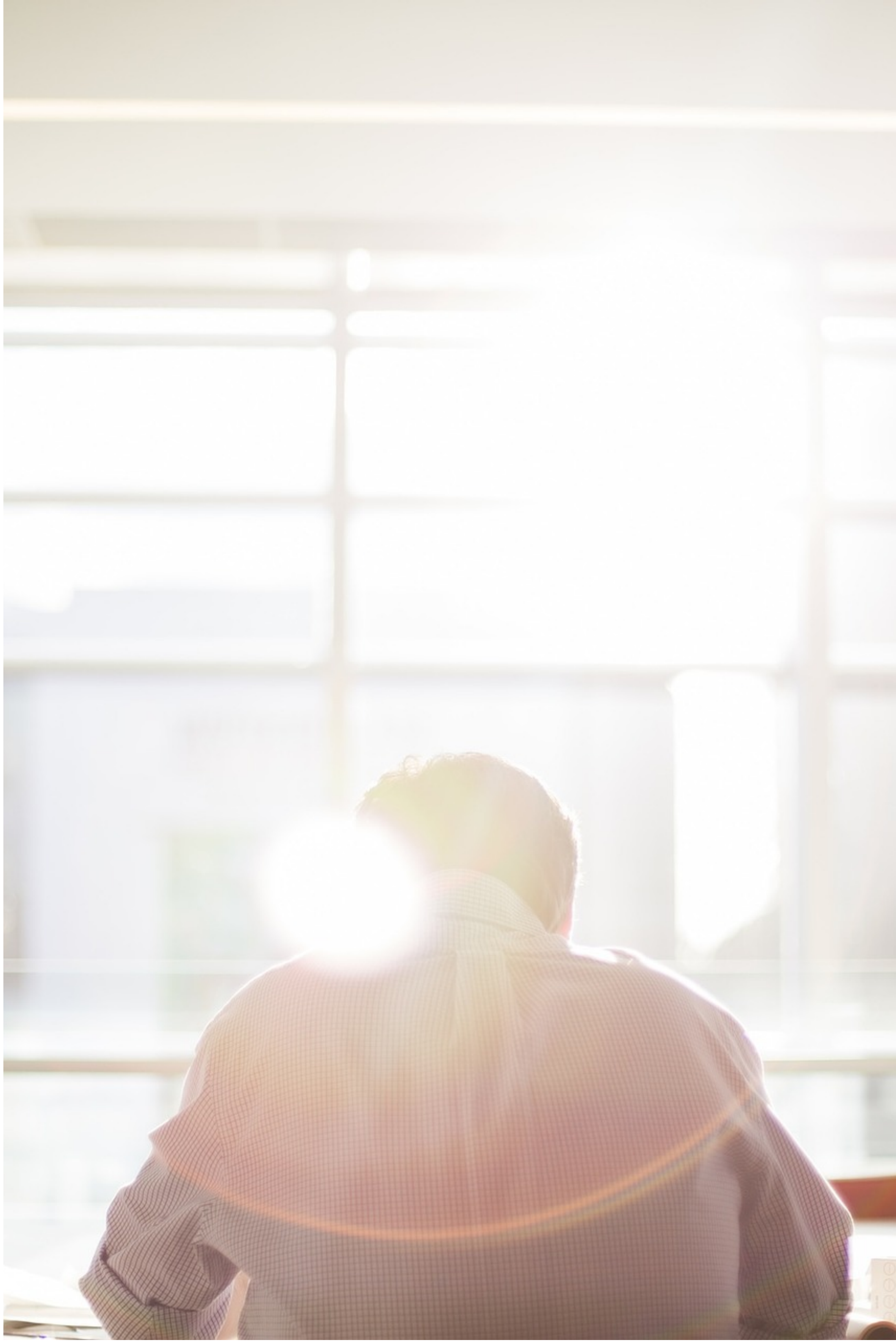
R.P.R.





# The Keys

- We prioritize our stupidity.
- We improve the plan as we get smarter.
- And we are always trying to get smarter.
- Can things get worse? (late, fewer features, more $, etc.) Of course they can.



# How Can Things Get Better?

Better in that…

- Fewer features are essential for the next release — or fewer story points in total, at least.
- We identify a better direction or just better features.
- The Velocity goes up.
- We fix another impediment that improves things a different way (e.g., higher quality).

# Goals for the ARP Day

- They all are on the same page - going the same direction.
- Specifically:
  - They all see the same elephant (in multiple dimensions).
  - They all are more motivated.
  - They all have shared MUCH more of the key tacit knowledge. This is BIG!

# Questions?

- What are your questions?
- What's your biggest take-away?  PLEASE SHARE.
- With your voice.
- Or in the Chat.

Thank you!

Please contact us if we can help you more.

info@leanagiletraining.com

**www.LeanAgileTraining.com**

We cover this area almost every week in our 1-day Agile Release Planning workshop.
***Please join us!***

# Discussion

- Who wants to start?

# Contact Info

- Joseph Little
- www.LeanAgileTraining.com
- jhlittle@LeanAgileTraining.com
- Please use and please give attribution!