

Notes for Improving Scrum Clinic - Part 2 - Ver 30 (Work-in-Progress)

Joe Little — June 10, 2024
Webinar — [LeanAgileTraining.com](https://leanagiletraining.com)

© Joseph Little 2024

Intro



Reminder

- You can get a copy of the slide deck.
- Put your email in the Chat
- Or send me an email (info@leanagiletraining.com)



Contents - Part 2

- Intro - slide 2
- Getting Started 2 - slide 5
- Diagnosis In-Flight - slide 26
- Start Sprinting - slide 52
- Commit - slide 59
- Getting Started 3 - slide 69
- Sprint Zero? - slide 74
- Discussion, Questions - slide 84



Getting Started - 2



List 1

- Why Sprint ASAP? (2-25)
- Agree Team Goal (earlier) (31-33)
- Agree Product Goal [later?]
- Evaluate Team skill sets (61)
- What does the Team need to go? (2-9)
- Business side commitment (2-10)
- Define Product Success [later! ARP.]
- Better automated testing (2-16)
- DOR/DOD (2-17)
- Agile Release Planning day (2-20)



List - 2

- An Impediment List
- A Risk List
- List: What your Team needs to go
- Business side commitment (2nd step)
- Define Product success
- Do Initial Agile Release Planning Day
- Initial discussion of adaptive planning
- Commit to Continuous Improvement (2nd step)



First Impediment List

- Top 20, prioritized
- Any show stoppers?
- Any major impediments?

- Impediments are: (a) negatives to remove, or (a) opportunities to become better
- How do we double our Productivity?
- I suck, we suck, they suck.



First Risk List

- List the risks
- Impact and Probability
- Discuss whether you want to start taking action
- If risk is huge, you must discuss



What is needed to be ready to go?

- There is commonly a work list of things that we could do to get started better.
- Examples:
 - Start automated testing
 - Improve Continuous Integration
 - Get servers (environments) just for us
 - Get laptops and access
- Do a short workshop on how to attack this work
- Add to Impediment List as appropriate
- Often we can start Sprinting before much of this is fixed. Usually Recommended to start Sprinting,



Get commitment from Business Side

- It is VERY common that the Business Side is NOT used to working in an agile way, and do not commit enough
- Business Side should provide a FT PO for a Team of 7
- Future: 4 good Business Stakeholders, who will show up and provide good feedback at Sprint Review. Have enough time
- Future: Business side will provide “minions” who will help PO develop the “details” and answer Questions quickly
- Explain more! And explain why it is important that they do this!



Get commitment from Business Side - 2

- Get 4 good Business Stakeholders. Evaluate. Request improvement.
- Who will show up and provide good feedback at Sprint Review
 - Business value
 - Details
- Product strategy
- Help PO
 - Ordering PBL
 - Deciding MVP



Get commitment from Business Side - 3

- “The Minions” - 1 to 20?
- Got some. Request improvement.
- Business side provides “minions” who will help PO develop the “details” & answer questions quickly
 - Minions might be called lots of things: BA, SBA, Manager, supervisor, smart person, someone who does the real work, SME (Subject Matter Expert), etc.
 - Minions need to commit enough time to develop details (commonly they do not - this will likely be an impediment)
- It is VERY common that the Business Side is NOT used to working in an agile way, and do not commit enough - re Minions
- Again: Explain more!



Define Success for Product

- The Drivers, 3-5
- Measurable

- And measure later
- Within 9 months (I suggest); faster feedback



Define Success for Team

- Here are my suggestions:
 - More Fun! Or Happiness
 - Double Velocity in 6 months
 - Fewer hours (~40 / wk)
 - Higher quality
 - Less stress
 - More creativity / innovation
 - More BV per SP
 - “I never want to leave this Team!”

Repeated here to
enable comparison



What does it mean?

- Discuss with Team what the “definition of Team success” means for them
- Important.
 - Has to be said many times
 - They won’t really believe it
 - *Say it MANY times: “Work smarter, not harder.”*



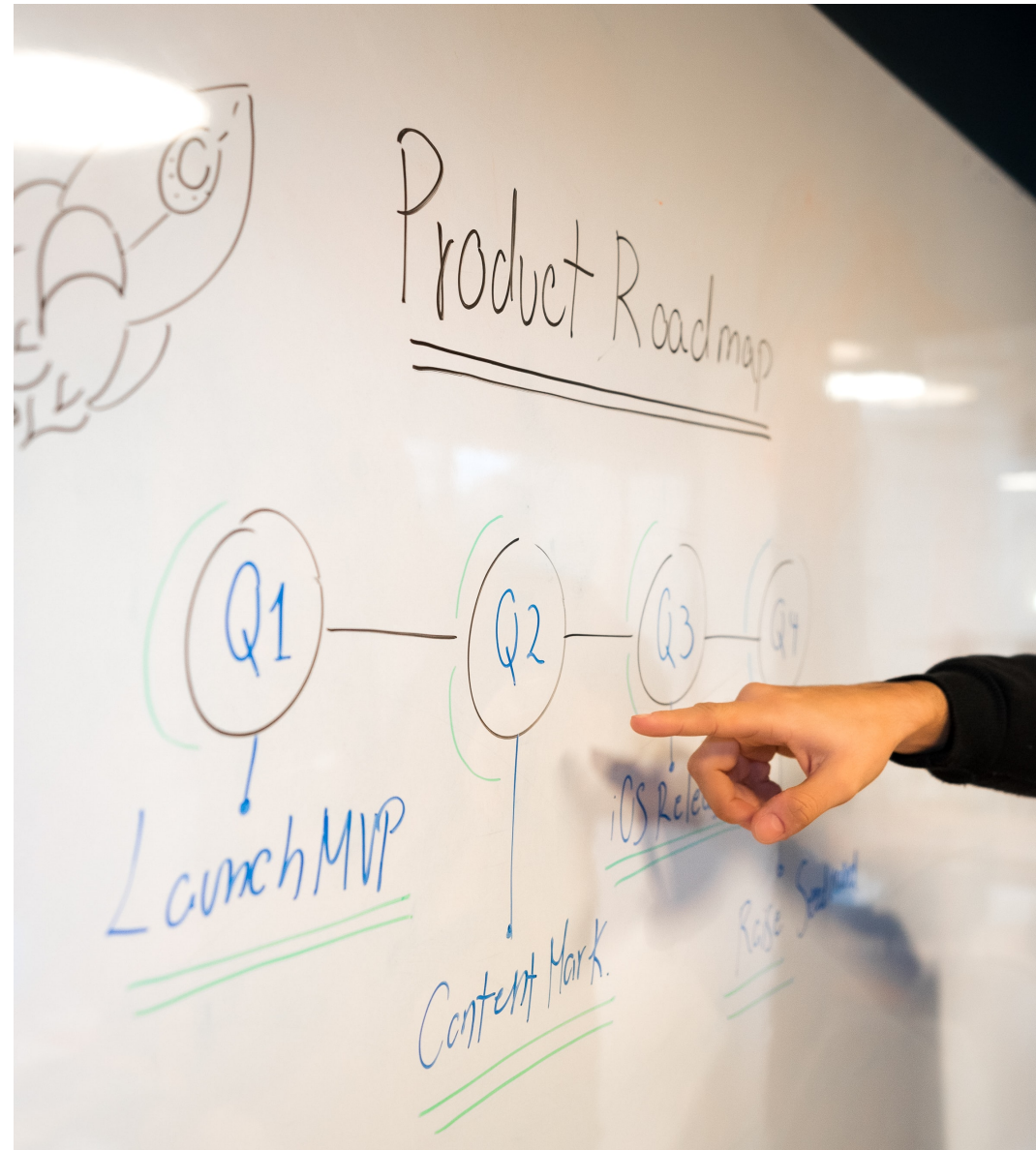
Commit to Automated Testing

- On this new work, all new stories will have automated tests. No other choice IMO
- Trouble, work, expensive — still the right thing
- Issue: How fast to build automated versions of manual tests
-



DOD

- Agree on a Definition of Done (DOD)
- Needed: Can't really estimate stories without it. Can't make Sprint commitment without it



DOR

- Agree on a Definition of Ready (DOR)
- What do the “Developers” need to build it right the first time
- Jeff Sutherland calls this “an enabling spec”. I think it it as many pieces of information that tell us what we need (and only what we don’t have).
- Needed: To deal with “unclear requirements”



Minions and DOR Process

- The PO is responsible for this, but many can contribute.
- Minions are the good people (part timers mainly) who will help on some details or some stories, etc.
- DOR Process is a very simple process for how this will work



Do Agile Release Planning

- A bunch of things, see my book, workshop, etc.
- Gather “full” Product Backlog quickly
- Get the stories to be smaller (initially for 6 months, twice as big as sprint-sized stories).
- Prioritized: By BV, ROI, and other factors
- Organize into Sprints for first release, based on a guess at Velocity



Agile Release Planning - 1

- 1 day with Team and BSHs
- Vision
- Product Backlog
- Business Value
- Effort
- R factor
- Other factors



Agile Release Planning - 2

- Estimate Velocity
- Lay out stories into Sprints
- Add Contingency and “Landing Strip”
- Figure out implied date

- PB Refinement
- Revising Plan
- Changing the Culture



Continuous Improvement of the Plan

- New plan every sprint
- Always learning to improve the plan



Commit to Continuous Improvement

- More broadly (eg, not just improving the plan)
- That we will spend 20% (?) of Team time on improving (eg, mostly the SM)
- And that we expect 100% improvement in Velocity by [1 year] because things are less stupid (the “process” runs in a less sucky way).
- It must have more fun / happiness.
- It is also like eating your broccoli every day. (I do not like broccoli.)



Diagnosis In-Flight



WARNING! ACHTUNG!

- No worries! Just for fun...
- I will ask you what else we should look at.
- Start making notes of **your own top 3 ideas** now!
- Type them in the CHAT window



First

- Often someone says either:
 - This team is not doing well
 - This team will not get the Release done by the deadline
 - OR: This Team is doing well
- Don't believe any of them quickly. Usually.
- The real questions: How much better can this Team become?
AND: How do we get them on a faster improvement path sooner?
- Four things to improve:
 - More Fun
 - More Motivation / Satisfaction
 - More productivity ("work") per Sprint (More SPs)
 - More BVPs per SP.
- Related goal (commonly): Less stress on the Team
- Related: Able to learn more per Sprint



Ask Them!

- It is their Team
 - I am assuming I am an outsider, such as an Agile Coach
- Ultimately, the Kaizen culture is more important than fixing N impediments or issues
- They know better
- They will act better if they participate in deciding what to do
- They, together, usually come up with a better specific solution



To diagnosis a Team - 1

- Collect and review available data
- Watch the Team in the Meetings (and as they collaborate)
- Talk to the Team (and to each team member)
- Talk to stakeholders
- Ask for or start to collect additional “data”
- Ask them which are the biggest “opportunities for improvement”
- Ask them to identify impediments (usually no existing Impediment List)
- Discuss the “success” we want
 - Product
 - Team



To diagnosis a Team - 2

- Does the Business Side give the Team good support? [PO, BSHs, Minions]
- How does the Business Side support? Money, influence, thankfulness?
- Prioritize what you want to work on (can be revised)
- Do one thing and at a time. And watch. Transparency. Inspect. Adapt.
- Let's be clear: You are always doing diagnosis -> the patient can always be healthier



Questions - 1

1. What do we measure? Ex: Average velocity over last 3 Sprints?
2. How's the current motivation of the Team? Of each person? Are they all inspired by this Prod Goal?
3. What's demotivating the Team?
4. How many Team Members? Which roles? How's the match vs skill-sets needed? All 100% allocated?
5. Who are the Chickens? And how do they help? Where doing well? Badly?
6. How many Bus Stakeholders? Do BSHs come to Sprint Review every time? Is their feedback useful?
7. Is there a concept of JIT knowledge creation & usage? When does knowledge go stale quickly? When not?



Questions - 2

1. How unclear are the Reqs? The details (for stories entering the Sprint)?
2. How fast do questions (re stories in Sprint) get answered?
3. Is Scrum seen as a Game? How much does the Team win?
4. Is the Team having more fun?
5. Does the Team have Drivers? And goals for each driver? Measure “success” later?
6. Do the Managers help fix impediments?
7. Does the Business Side feel agile is important to them? Is it working for them?



Questions - 3

1. How is the Team managed?
2. How is Tech Debt handled? Is Tech Debt in the Prod Bklog?
3. How are legacy Bugs handled? In Prod Blog?
4. How happy is the Customer? How happy do they expect to be?
5. Has the Team identified the 18 key knowledge domains and skill sets?
6. How well do the Devs match the needed skill sets? Depth in needed places?
7. How much interrupt work does the Team get each Sprint?



Questions - 4

1. How much does the PO say “no” to some of the interrupt work?
2. How much does each team member collaborate? How well with each person?
3. Is there a hum in the room, commonly?
4. Does the Team have a team room? That they use a lot?
5. Does the Team have a Kaizen mindset? How does this mindset come out in practice and in improved results?
6. Do the Developers vote on the quality of the info for each story? Do they reject some? Does the DOR lead to more productivity? And more “fit-for-purpose”?
7. Is the PO (with the Team and others) trying to do 80-20? As you approach having done 20% of the SPs, do you check what % of BVPs are now done?



Questions - 5

1. How often does the Team fix all the bugs (identified in that Sprint) during the same Sprint? What exceptions?
2. Is stress in the eu-stress zone? Or is there over-stress? How much? Efforts to mitigate?
3. How is the Team learning? Is knowledge shared with all team members?
4. Is the PO getting R&D done outside the Team?
5. Do team members feel empowered to self-manage the Team? How does that play out? Should it be more or less?
6. How many stories per Sprint? How long is the Sprint? How close are all sprint stories to the same size?
7. How well does the Team do “single-piece continuous flow” inside the Sprint on Stories & Tasks?



Questions - 6

1. Does everyone want to be on this Team and attacking this Prod Goal? Or would some prefer to be individual contributors?
2. Does the Team ask manager(s) to say “yes” to fixing some impediments? People, money, approval?
3. What is the learning plan for each Team member? How will the skill-set gaps be mitigated?
4. Are team members willing to do whatever it takes to be successful? Or in what ways not?
5. Is there a trend of continuous improvement? How much? How would someone outside the Team see that?



Available data

- Data might be real data
- Get opinions also (but keep an open mind)
- Ask manager(s)
- Speak to SM privately
- Speak to PO privately
- Even the absence of data is to some degree “data”



Some possible data - 1

- How long has the Team been together?
- How many changes in the Team since it started? How stable a team?
- Is the Team on one mission? Or is there mission “confusion”?
- How many Sprints so far?
- Are the Sprints considered successful? Why?
- How much is the Team winning or losing? What does losing mean? Is the Team learning from losing?



Possible data - 2

- Existing Impediment List (if any)
- How well prioritized? Looking retrospectively.
- List of Impediments fixed
- Velocity per Sprint (all Sprints). Look at trend
- Does the Team have a DOR-like concept?
- What is the quality of the incoming Reqmts? How is this “controlled”? Process? How well does it work?
- How many ‘committed’ stories do not get done per sprint?
- How many Bugs not fixed in Sprint where identified?
- How many Code review problems identified but not fixed in the Sprint



Possible data - 3

- Happiness metric
- How many hours per person per week?
The honest answer.
- How much interrupt work added to the Sprint?
- Plan vs actual re delivery of current release
- How much BV have we delivered to customer so far
- Any attempt to track ratio of BVPs to SPs to date or in this release?



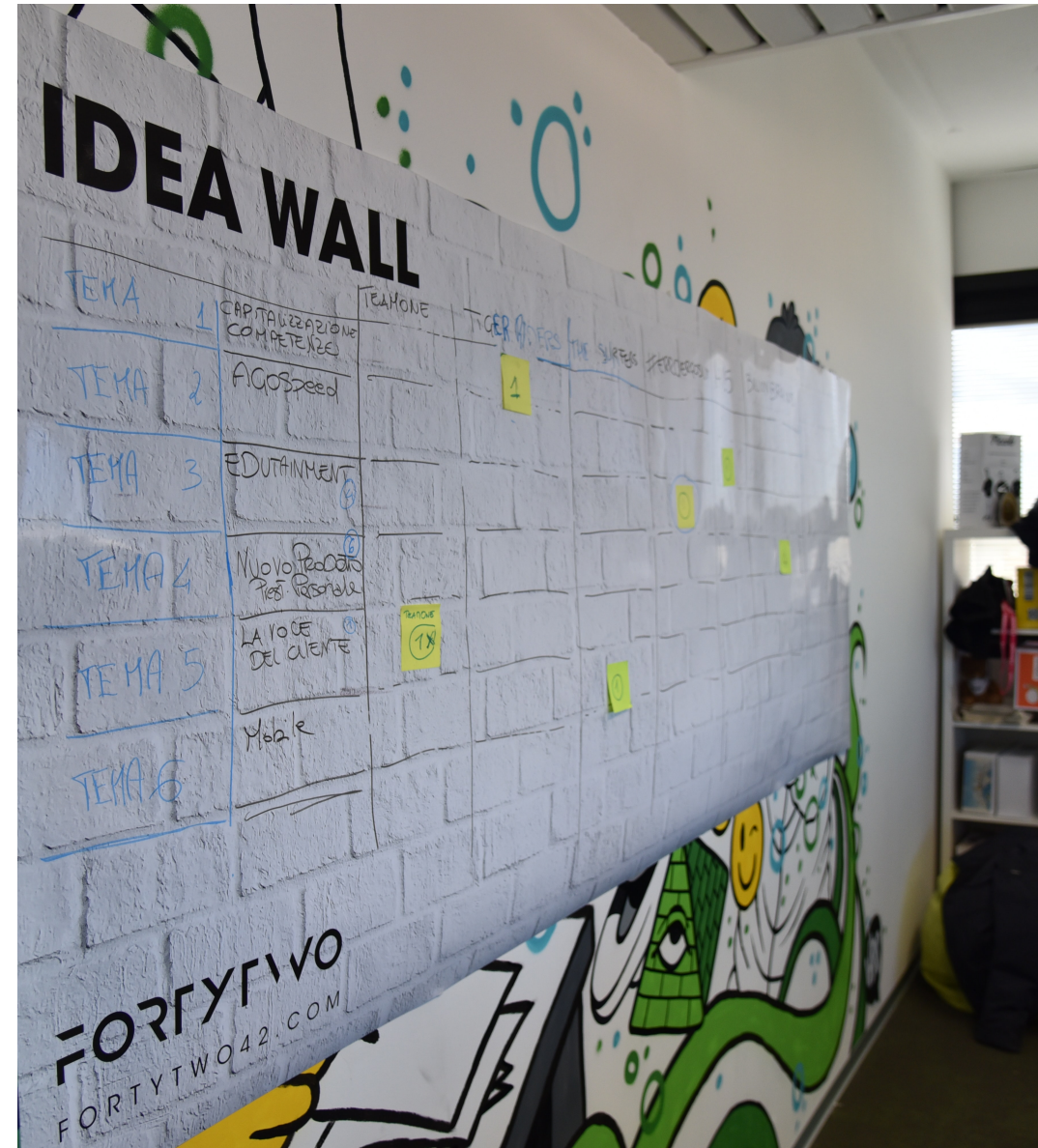
Watching Meetings - Basics

- Start and stop on Time
- Ask at end: Was this meeting useful? How many hands go up?
- How much is talking spread out amongst people? How evenly?
- Is one person too talkative?
- Do people listen to each other?
- Does each Meeting have a “goal” and everyone can state it? Do people think the goal is accomplished?
- Is the “most important thing” acted on promptly?
- Are there some celebrations?
- Can each person articulate the value (or logic) behind each meeting?



Talk to stakeholders

- People outside Team:
 - Manager(s)
 - Business side
 - Minions
 - Customers
 - Other people / groups
- Learn:
 - Their view of Team
 - What do they say is “Stop, Start, Continue”?



Additional data

- Hard to identify now.
- Depends what they have / had
- Depends what you've learned so far
- Issue: Value of new “data” vs difficulty (cost) to get



Identify Impediments

- Start from scratch
- Identify what we mean by impediment (anything that would make us more effective - removing or mitigating a bad thing, or adding a good thing). ANYTHING!
- Review types of impediments. Open their minds about it
- Write them
- Share and briefly discuss
- Prioritize
- Don't assume the new List is that great, but I bet it's a lot better. And a good place to start.
- Psychologically, fixing their biggest thing gets them more "in"
- Business side stuff is more important than they usually think



Prioritize

- Let the Team prioritize the Impediment List. Yes, you can influence them (fairly)
- Accept some wins; the perfect is the enemy of the good. Get the ball rolling downhill.
- We need some “quick wins” - to some degree.
- ROI mainly. Maybe dependencies, political capital cost, risk



Agree on Outcomes

- Depends partly on the culture. Start to change the culture, but don't fight the culture too much.
- I suggest:
 - “Business” outcomes. Typical customer, business, product things.
 - Team outcomes. Often not considered or not given much weight.



Business Outcomes

- I suggest “Drivers”.
 - 3-5
 - Measurable, “analog”, aspirational
- Motivation of the Team is key
 - We want everyone on the Team to find these motivating. Or at least each person is motivated (ok if Drivers are not the thing)
- Linkage to Managers, Business side, etc.
 - They are important in that context too.



Team Outcomes / Goals

- My suggestions:
 - Fun / Happiness up
 - ~ 40 hrs / week
 - Less stress
 - Higher Velocity (50% or 100%)
 - “Working smarter, not harder”
 - Higher quality
 - More BV per SP
 - “Beautiful!”
 - End: “I never want to leave this Team”



Business Engagement

- Why important?
 - Is this hard to explain?
- Signs:
 - PO is good and has time
 - BSHs are good and have time
 - Business side supports agile and the Team
 - Business side responds quickly with answers to questions
- How much has it improved?
- Best guess: How much will it improve in next 6 months? Why?



Business - Honestly?

- Commonly, engagement sucks.
 - Yes, sometimes more, sometimes less
- We do not try enough to pull them in
- We do not know how to talk with them
- It can improve
- It must improve
- Business Agility is probably more important than “agile in the teams”



What else?

- Every situation is different
- What do you smell? What do your spidy senses tell you?
- What else should you add?
- “..., every unhappy family is unhappy in its own way.” Tolstoy
- Type your ideas into the CHAT window now. 3 ideas each. A word or phrase to remind you what you mean.



One step at a time

- Fix (mitigate) one thing at a time.
- Check if you got results
- Transparency. Inspect. Adapt.
- For you and the Team: the process, to some degree, will reveal more to you and the Team.
- Success (in removing impediments) will cause them to say - “you know, I’m gonna go ahead and mention this now...”
- Build up team spirit, “all for one, one for all”, fun. Build up the sense that anything could change (if it helped) — then they will reveal more.
- They will use vague words. Help them draw out the impediment they are hinting at



“Give me a kiss to build a dream on...” Louis Armstrong

- <https://youtu.be/82Ff4ncMCWU?feature=shared> Louis Armstrong.
- Some Teams start slowly, and go faster later.
- Other Teams start fast, and then slow down.
- Patience, perseverance
- Take the win. Keep going.



Start Sprinting!



Start Sprinting!

- So important
- When?
- Hard to decide. Pressures both ways. When in doubt...
- Start now!
-



Why?

- You learn so much by doing Sprints
- You also are making some progress toward the goal (typically the first release)
- Definitely as we go: Improve things more for your Team.
- And: get the first Sprint started pretty soon!
-



Start Sprinting

- Yes, everything is NOT ready
- Nonetheless, start sprinting
- Sprints will tell you the most important thing to fix next
- Sprints help the Team form
- Sprints force the Team to address some key issues
- Sprints start to tell you the Team's real Velocity



Learn from the Sprints

- Lots of learning:
 - In the Scrum meetings
 - In doing the work together
- Don't just learn; take action or adapt
 - Focus on 1 impediment at a time
 - People can also fix their own impediments



Continuous Change to the Impediment List

- You should see continuous change in the Impediment List
 - Some are getting fixed
 - Better impediments are being added
- And soon, continuous-ish increase in Velocity. Maybe 1 SP per sprint
 - With no extra effort
 - With MORE fun



Problems

- Telling the truth...
- Team members don't want to admit they are not perfect.
- The "organization" does not want to talk about the dirty linen.
- People blame Scrum for the problems Scrum reveals.
- Hence: "Courage"

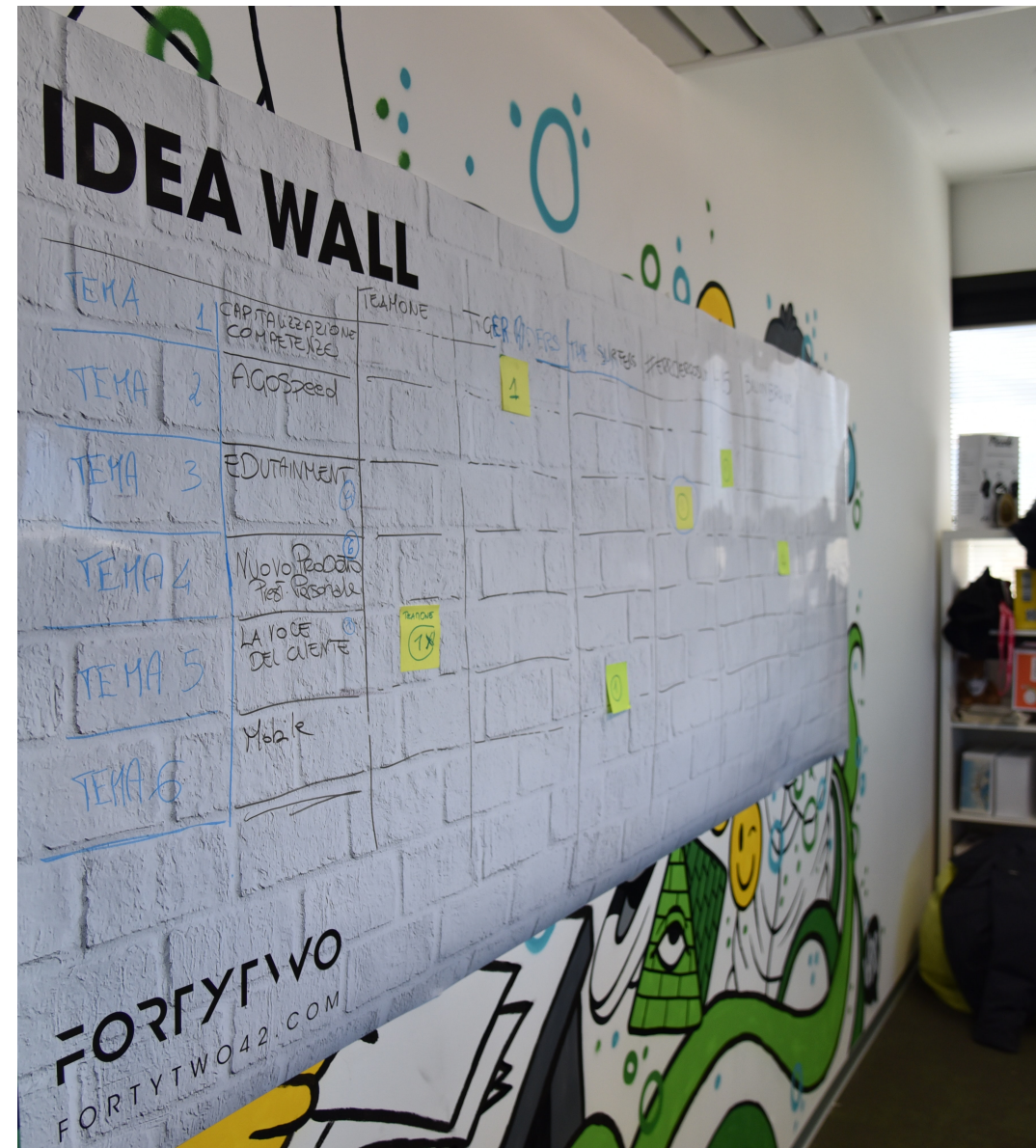


Commit



Commit - I want to do this (I think)

- Don't bring a knife to a gun fight
- Also: It is important that "I want it"
- "A dead SM is a useless SM"
- And get others to commit. Not always the first day. Some will hesitate, but become committed later.



Not perfect now

- No one is perfect at first. Don't worry about that.
- But commit to starting as well as you can (given the whole situation)
- And commit to getting better



Commit to doing “Aggressive Scrum”

- Get them to agree to TRY “Scrum by the book” or complete Scrum
- For the Sprint stuff and the 4 meetings in the Sprint (SP, DS, SR, Retro).
- Use Velocity
- Having FUN!
- They will not do it well. Like a bunch of 8 year olds learning football. Be patient.



Become a Change Agent

- You already are
- But start to identify this, be aware of it, and learn how to do it in a professional way
- It does not mean: You get to rule the world
- Not: I am King of Everything!
- BUT: I want to try to lead us to more success (and then define success)



How are we changing?

- Start learning (more) about making change happen
- Come up with approaches to change
- Gather the initial “change team”
- Think about changing the culture
- Make an initial plan

- Manns/Rising: More Fearless Change (~60 patterns)



When are you “doing Scrum” correctly?

- Not an easy question
- What is your (current) vision of perfection?
- May include Scrum stuff plus generic agile stuff plus other stuff



When are you “doing Scrum”?

- One answer: When you have (some) working product by the end of the Sprint reliably
- Another: When you are doing everything in the Scrum Guide
- Better answer? You will always be trying to do Scrum, and to understand the agile-scrum mindset, better. Always the climb.
- Excelsior!



When are you “doing Scrum”?

- Another answer: When you are doing most of the Scrum stuff, and you are getting more success from that
- Another answer: When the Team and those around the Team do agile/Scrum with a good mindset, and therefore become more successful
- Where does “business agility” fit in?



Recommendation

- Try: To do all of Scrum at least for 3 months, and test how much it helps you (or maybe not)
- Do not obsess with agile or scrum. The real goal is for everyone to have a better life. And some of that is, for example, better metrics as a tool to becoming better.
- Again, focus more on the outcome than the tool (the means).



Getting Started - 3



Managers identify key Impediments

- The managers should identify them first
- Call them impediments, risks, showstoppers, whatever — things that if they happen as badly as we fear, this Team will not have success
- Consider mitigation. Usually some already clearly need to be fixed or mitigated.
- Probably: start taking action



Other things

- Team Room
- Team Norms
- Team Training (of 2 types)
 - Agile-Scrum
 - Other skill sets
- IAD: Infrastructure, Architecture, Design
- What else for “set-up”?
- Probably: start taking action



Infrastructure, Architecture, Design (IAD)

- These are key things in software development.
- And often these names or similar are used in other work
- Sometimes they exist or are “obvious” and fairly easy
- Sometimes the “set-up” is lots of work
- If so, then we (the Team) or someone must do this work
- If the Team does this work...



IAD

- Let's discuss and define briefly:
 - Infrastructure
 - Architecture
 - Design
- or the equivalent for your product
- And similar things

- Yes, it is true that some of these are “necessary”
- But almost always: more imp to start soon



Sprint Zero?



Sprint Zero - Wait! What?

- The simple idea is that we do everything “to get started” in one Sprint Zero.
- Yes, there are some things to do before your Team starts sprinting
- Context is very important.
- Both: what you need for your product and how Sprint 0 might be done in your company vary widely.
- Can get to “the meat” sooner if we understand the situation, at least basically?



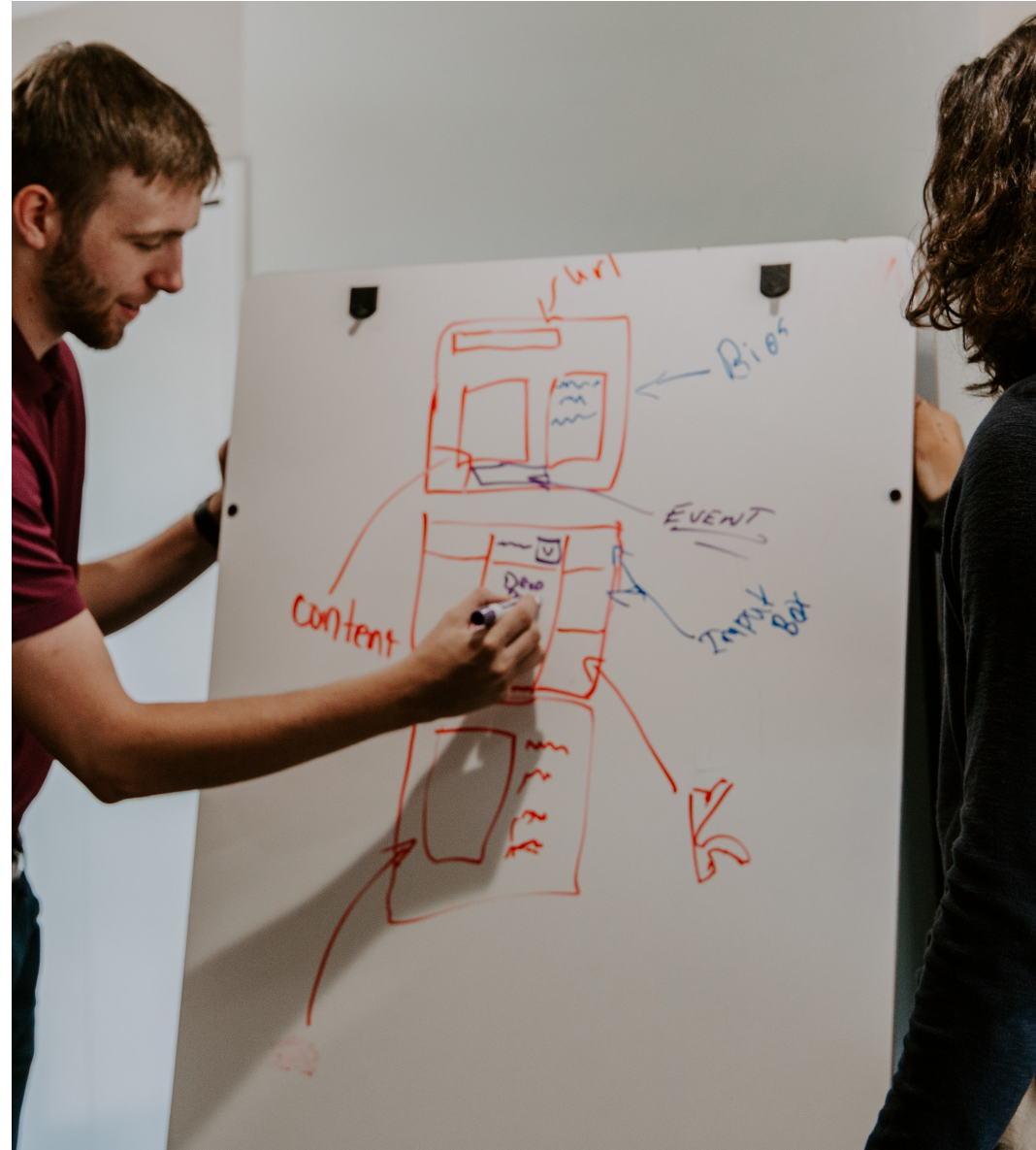
Issues

- Sprints should be of consistent length (eg, 2 weeks). But...
- The work is hard to estimate; hard to hit the Sprint size well
- The BSHs need to start giving feedback; but typically are not good re IAD
- The Team needs feedback!!
- The “manager” often does not understand “no real progress - we’re getting ready to start”



IAD work

- IAD (Infrastructure, Architecture, Design) — we need some...but what is needed specifically for us?
- A wide range of different levels of effort. Depends, a lot
- Define the “cards” (work) between our Team and any other groups
- Assuming sizable and for us (the Team), how do we attack?
-



Fix all the Impediments??

- This is often a request.
- NO! (to fixing all the impediments — Some impediments? Depends more...)
- OK, we can and probably should quickly fix one or two impediments.
- But...
 - We can fix impediments as we go
 - We can see the biggest impediment better
 - We can fix things as we see quickly how much difference it made
 - What needs to be fixed most will change



BUFR?

- BUFR = Big Up-Front Requirements.
- In Agile the bias is no.
 - Delay; you can start “learning by doing” sooner. And probably deliver sooner.
 - Change
 - YAGNI (You ain’t gonna need it)
- But is “no” or “none” reasonable? In some cases, we think not.



How about for Requirements? Analysis? Design?

- This might include a kind of prototype.
- Situations vary a lot.
- The willingness of the business side to fund the building varies a lot. The amount of funding needed varies a lot. Politics varies.
- So, maybe. See next slide.
- Use common sense. Common sense is very uncommon.
- If we do some “up-front thinking”, this should be much shorter than what we used to do in waterfall.



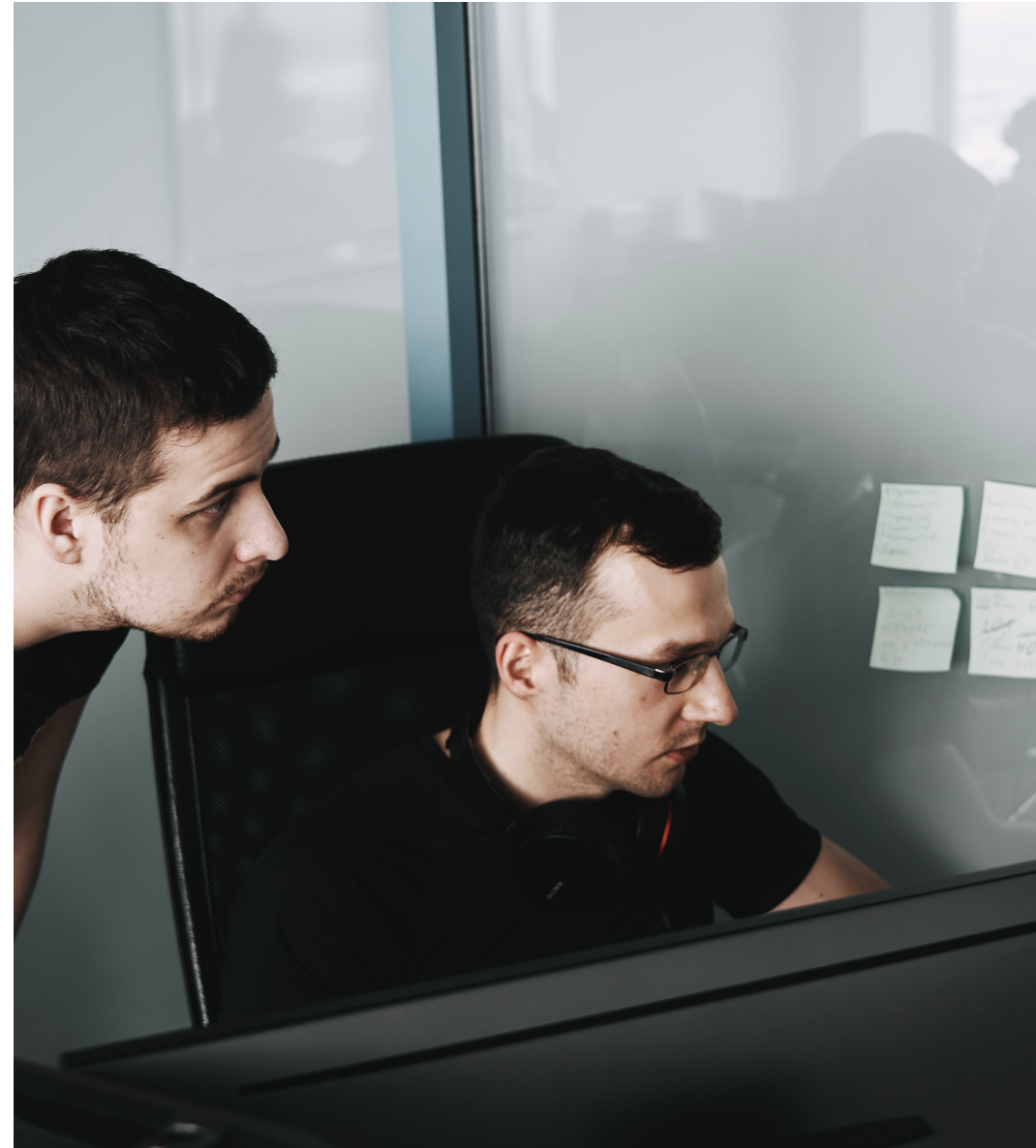
Proportionality

- BUFX for 1 sprint is one thing if the total release timing is 3 Sprints.
- Very different to take 1 Sprint out of 20 Sprints.



How about for Requirements? Analysis? Design? - 2

- Ways to think, questions to ask.
- How much delay? Compared to what?
- How much learning is gained? Or confidence gained?
- How much is risk reduced, if....?
- What is the minimum we can talk the business side into?
- How much will things change once we spend 1-2-3 sprints doing this stuff?
- The more these things will be changing, the more we want to do this “research” while we are building.
- Do not let people think all the learning has occurred. It has not.



Other things?

- Are IAD & other work similar?
- Maybe
- What other things?
 - Make your specific list
-



Discussion, Questions



Discussion, Questions

- Don't be shy!
- Turn off the mute
- OR: Type them in the Chat



Other webinars

- Two Types:
 - About courses and workshops (30 mins)
 - About agile questions or issues (60 mins)
- Where:
 - [LeanAgileTraining.com](https://leanagiletraining.com)
 - [MeetUp](https://www.meetup.com)



More Info about Us

- Courses & Workshops:
leanagiletraining.com/lean-agile-and-scrum-courses
- Joe Little
- jhlittle@leanagiletraining.com
- (704) 376-8881
- leanagiletraining.com

