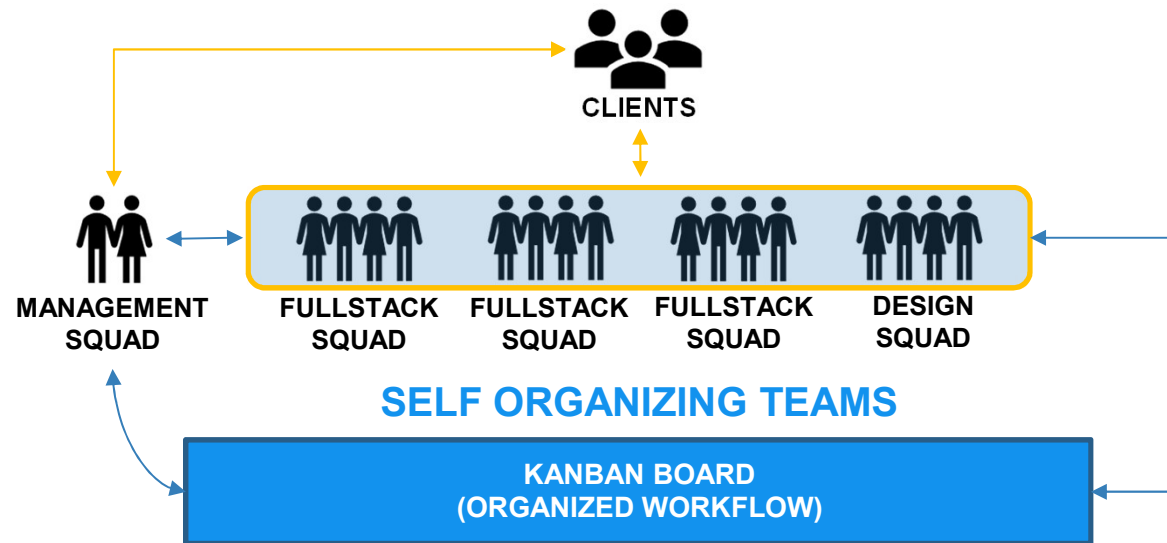




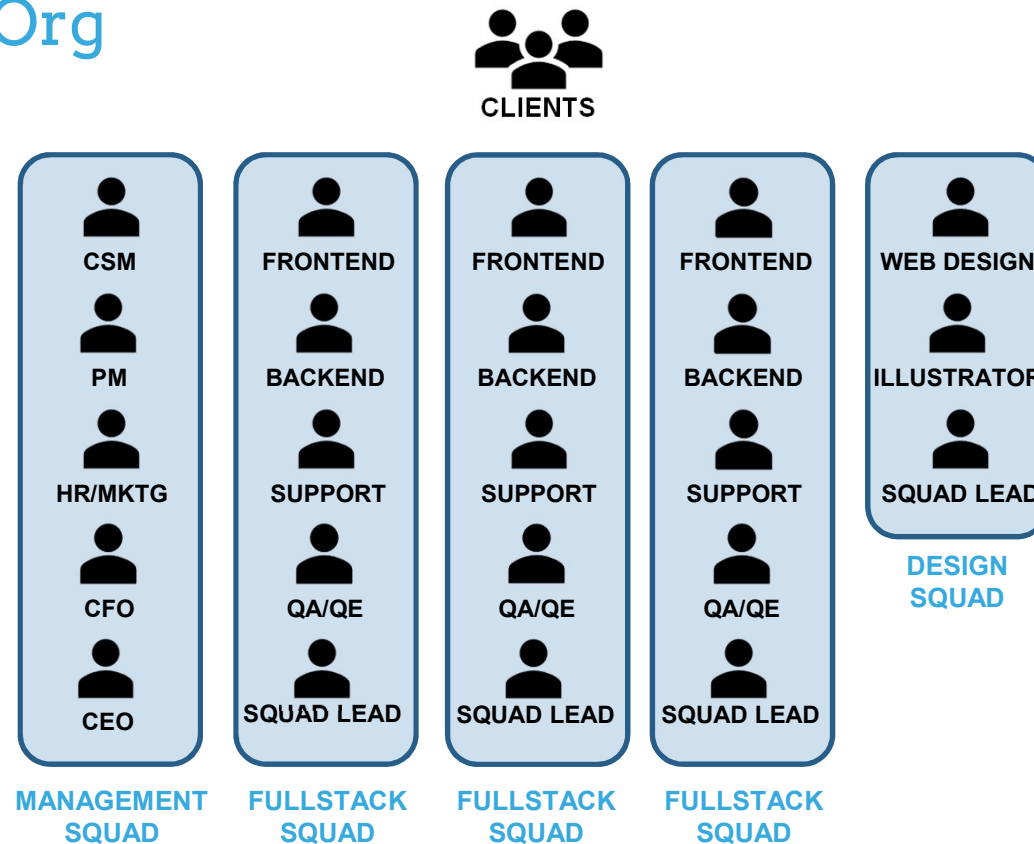
Fullstack HQ Agile Training

Oct 21, 2019

FSHQ Agile Model



Optimized Org Structure



SELF ORGANIZING TEAMS

Optimized Org Structure

What are Squads?

- A Squad is small, cross-functional, self-organized team. The team members have end-to-end responsibilities, and they work together towards their project mission. With Squads, the key driver is **autonomy**.
- Each Squad has the **autonomy** to decide **what to build, how to build it, and how to work together while building it**.

People work with **autonomy, mastery, and purpose**. Autonomy is motivating, and motivated people **build better stuff, faster**. Autonomy makes us **faster** by allowing **decisions** to be made **locally** instead of **solely** by **managers and committees**.



Optimized Org Structure

Squad Roles:

- **Squad Lead:** Communicates the details of the project that needs to be worked on. Acts as a **liaison** between the client and the team.
- **Squad Member:** Collaborates with **both** the **client & team members** to find the best solution. Has the **autonomy** to decide **as a team**, rather than **wait** for a **management** or **leader's** decision.

“Loosely coupled, tightly aligned”



Better Workload Management

Workload Sizing:

- **Estimation:** In software development, an “estimate,” in the usual sense, consists of a quantified evaluation of the **effort** necessary to carry out a given development **task**; this is most often expressed in terms of **duration**.

The intent is to aggregate many such individual estimates, so as to obtain an indication of the **overall duration, effort or cost of a software project**.

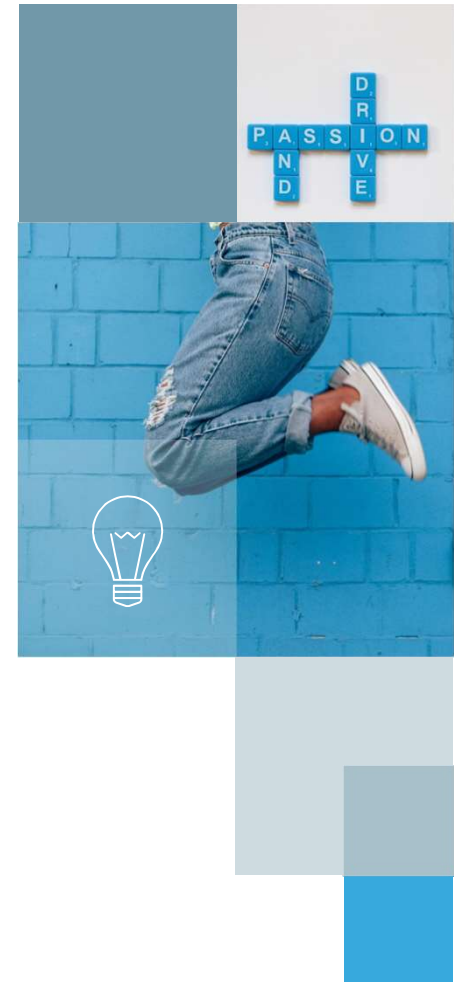
“An estimate isn’t a final answer, it reflects the information that was on hand at the time of communicating it; It should always be permissible to update an estimate in light of new information, either upwards or downwards.”



Better Workload Management

Workload Sizing:

- **T-Shirt Sizing:** A relative estimation technique used in agile projects. With this approach, the development team evaluates whether they think a **task** quantifies as either “**Small**”, “**Medium**” or “**Large**” based on a relative amount of time/work needed per estimated size.

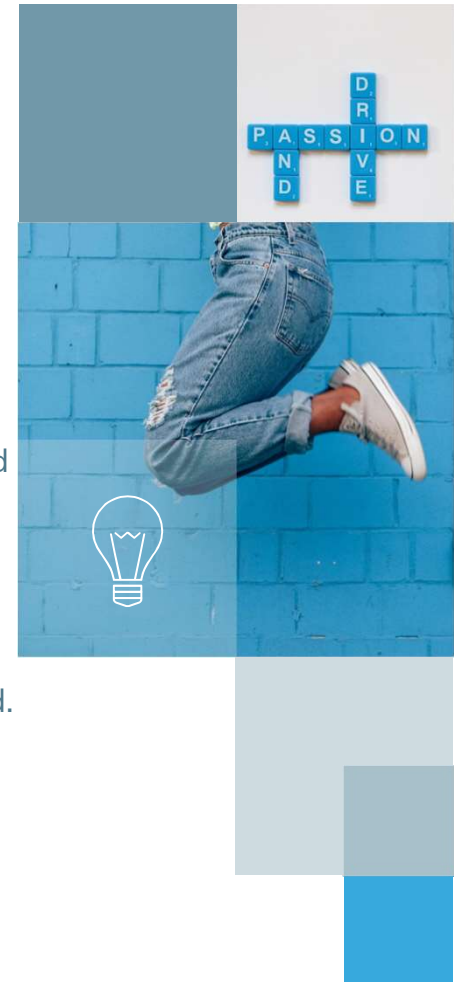


Better Workload Management

Workload Sizing:

FSHQ Estimation Approach: Relative task estimates (T-Shirt Sizing) and corresponding Delivery Timeframe:

- **Extra Small (XS)** – Very minimal work required. Need to combine with other similar sized tasks in order to get it up to at least a **Small(S)** sized and timed deliverable.
- **Small (S)** - 1 hour or less work required. Deliverable within 1-2 days from date received.
- **Medium (M)** - 2 hours or less work required. Deliverable within 1-2 days from date received.
- **Large (L)** - 4 hours or less work required. Deliverable within 2-3 days from date received.
- **Extra Large (XL)** - more than 4 hours work required. Need to do pair programming in order to get it down to either a **Large(L)** or **Medium(M)** sized and timed deliverable.



Better Workload Management

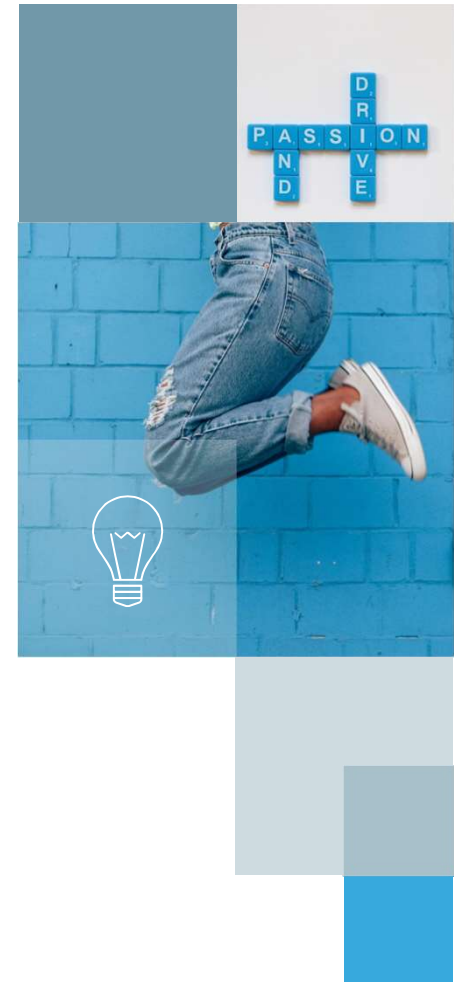
Workload Sizing:

FSHQ Workload Approach: 7.5 Hours Productivity | 1.5 Hours Break/Lunch

- 6 Hours Daily Required Workload Target (based on workload estimates)
- 1.5 Hours Daily Allocated Slack Time (For urgent tasks, scope creep, upskilling, self-QA)
- 1.5 Hours Daily Allocated Lunch/Break Time

WHY THIS APPROACH?

- Having a manageable workload target as well as allotted 10% buffer/slack time serves to address workload concerns, as well as avoid unnecessary overtime.
- Having an allocated slack time helps manage time spent for upskilling or self-improvement, or even assist other squads who are at full capacity.



Better Workload Management

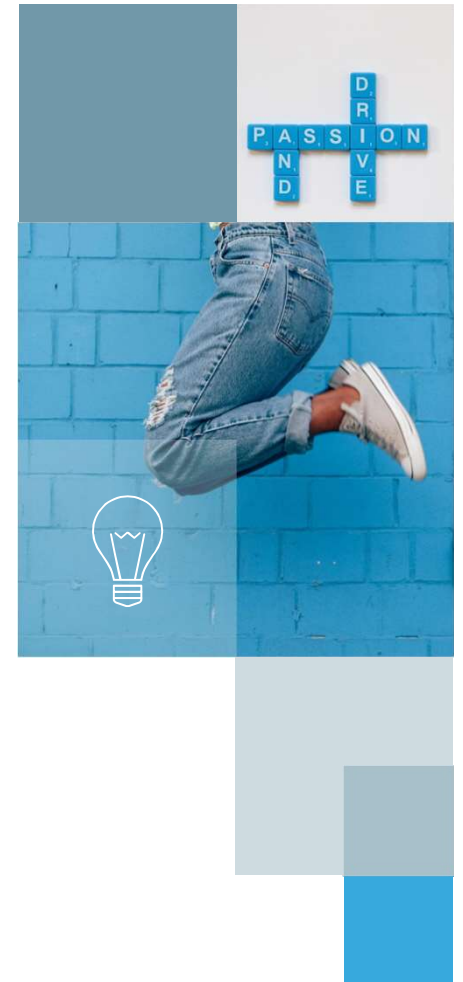
Workload Sizing:

FSHQ Work Schedule Approach: “Just In Time” Schedule Flexibility

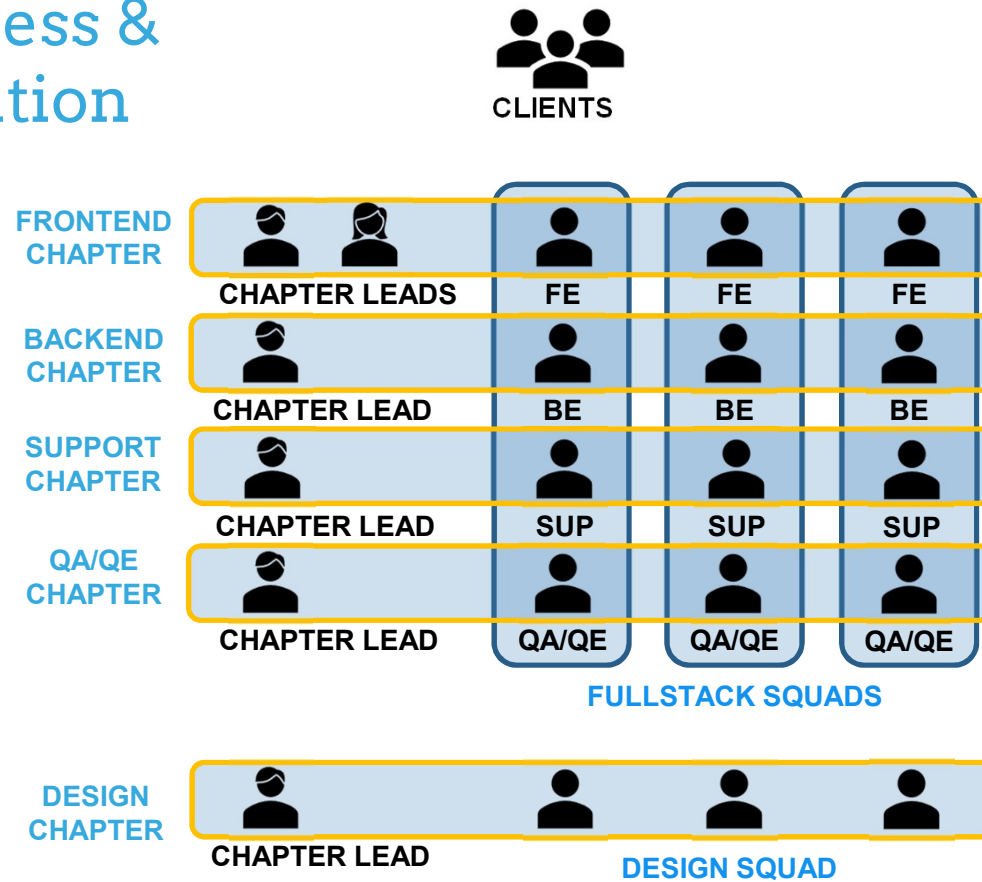
- Work schedule depends on the time zone of the client being handled currently. So for example, if someone is handling an AU based client at the time, she/he should report to work **“just in time”** in order to respond and accommodate the client accordingly.

WHY THIS APPROACH?

- Late response time and dev unavailability affects the business and paints FSHQ in a bad light to our clients. As such, there has to be the right balance between going to work at a certain time and being available to support the client’s needs **“just in time”**.



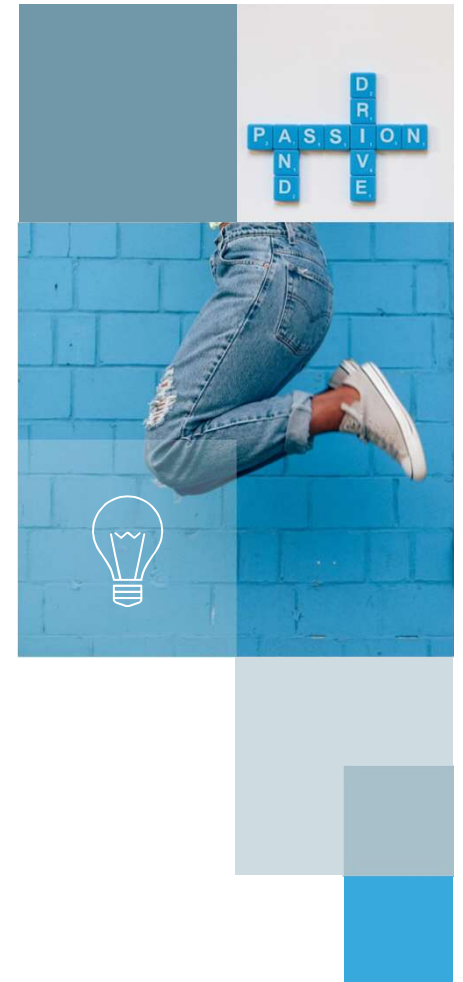
Skill Progress & Compensation



Skill Progress & Compensation

What are Chapters?

- Chapters are a group of people formed based on competency areas, which are often spread out across squads. Chapters often get together and exchange ideas, get help on challenges and discuss new technologies specific to their specialty. **This is a great way to promote innovation and 'cross pollination' of ideas across teams or squads.**
- A **Chapter lead** acts as the line manager for chapter members. They are responsible for **developing people** and **setting salaries**, but they remain part of a squad and still do day-to-day work.



Skill Progress & Compensation

Progress Timeline:

FULLSTACK DEVS

Implementation Timeline

Phase 1	Primary Skill	Frontend	Backend	Support	Q4 2019
Phase 2	Secondary Skill	Support	Support	Front or Backend	Q1 2020
Phase 3	Final Skill	Backend	Frontend	Front or Backend	Q3 2020

QA/QE

Phase 1	Pure QA	Q4 2019	Evolution to QE
Phase 2	Understanding of Frontend and Support Code Logic	Q1 2020	
Phase 3	Understanding of Backend and Support Code Logic	Q3 2020	

*Compensation via Salary Ladder Implementation



Organized Comms & Cadence

What are Daily Standups?

- The **Daily Standup** (or **Daily Scrum**) is a **15-minute** time-boxed event for the squad to synchronize activities and create a plan for the day. This optimizes team collaboration and performance by inspecting the work since the **last** Daily Standup. The activity is held at the same time and place each day to reduce complexity.



Organized Comms & Cadence

How do we proceed?

The Daily Standup is structured around some variant of the following three questions:

1. **What have you completed since the last meeting?**
2. **What do you plan to complete by the next meeting?**
3. **What is getting in your way?**

The intent of these questions is to emphasize completions of tasks, rather than effort spent. Also any topic that leads to a discussion should be cut short, instead added to a “parking lot” to be discussed in greater depth after the meeting between the people affected by the issue.



Organized Comms & Cadence

Expected Benefits:

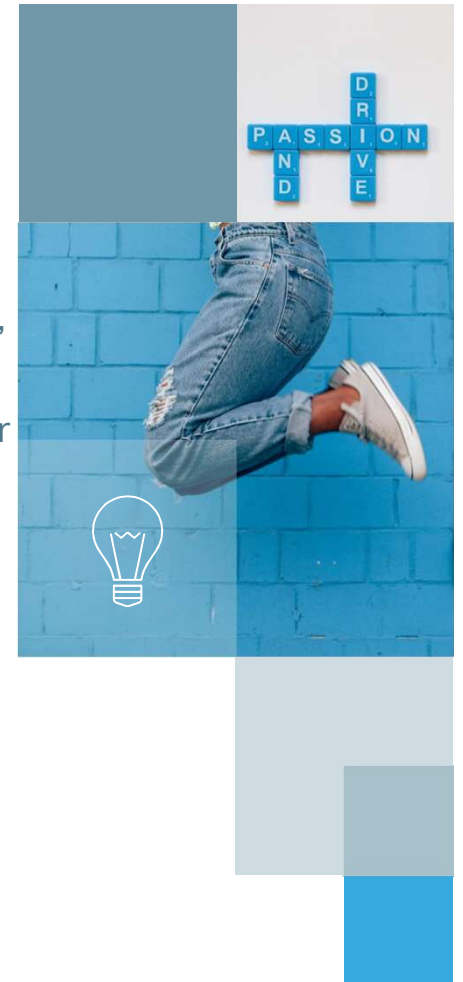
- The daily standup prevents a common failure mode of teams, where in the absence of an explicit occasion to share recent information, some critical knowledge may sometimes “fall through the cracks”.
- Regular peer-to-peer sharing of information in a short, focused and energetic meeting also contributes to team cohesion
- Stand-up meetings are reliably shorter, more pleasant and more effective than sit-down meetings



Organized Comms & Cadence

Common Pitfalls:

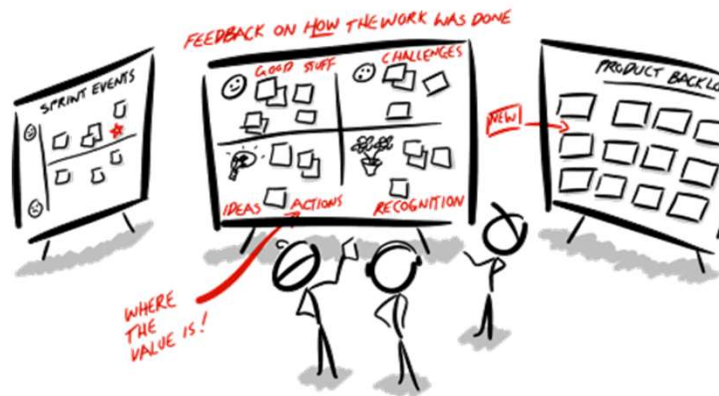
- The most common mistake is to turn the daily standup into a “status report” with each member reporting progress to the same person (e.g. squad lead, management) – exchanges in the daily meeting should be on a peer-to-peer basis.
- The team finds little value in the daily standup, to the point where people will “forget” to have it unless a squad lead or manager takes the initiative; this often reveals a lukewarm commitment to Agile.
- The “no problem” meeting, where no team member ever raises obstacles even though the team is manifestly not delivering peak performance; this is sometimes an indication that the corporate culture makes people uncomfortable with discussing difficulties in a group setting.



Organized Comms & Cadence

What's a Retrospective?

- It is an opportunity for the squad to inspect itself and create a plan for improvements to be enacted during the next “Sprint” or Development Cycle. It is important to ensure that the event takes place and that attendants understand its purpose. This is the opportunity for the squad to improve and all members should be in attendance.



Organized Comms & Cadence

How do we proceed?

During the Retrospective, the squad discusses:

- What went well?
- What could be improved?
- What will we commit to improve in the next development cycle?

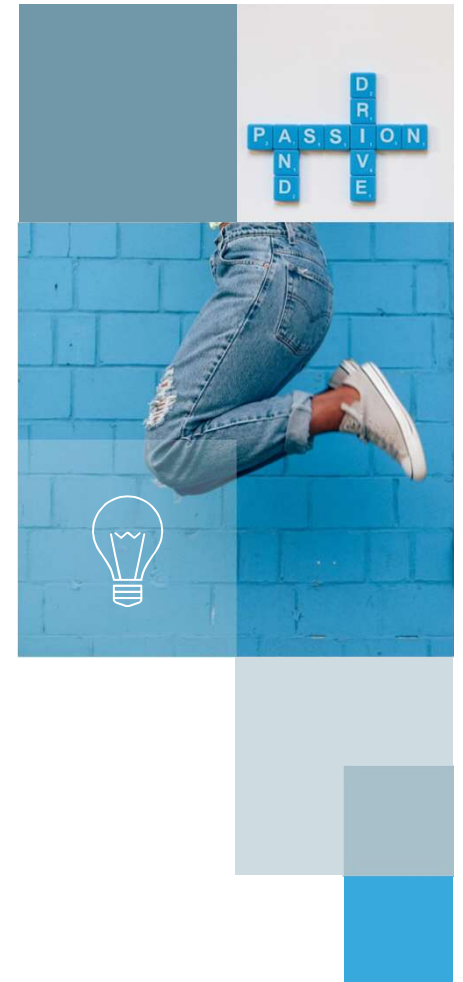
During each Retrospective, the squad plans ways to increase product quality by improving work processes or adapting the definition of “Done” if appropriate and not in conflict with product or organizational standards. By the end of the Retrospective, the squad should have identified improvements that it will implement in the next development cycle.



Organized Comms & Cadence

Expected Benefits:

- Retrospectives leverage the benefits of iterative development: they offer explicit opportunities to improve the team's performance over the duration of the project or the ways of working
- Promotes ownership and responsibility by the squad with respect to all aspects of the process; participants can understand the rationale behind all process decisions



Organized Comms & Cadence

Common Pitfalls:

- A retrospective is intended to reveal facts or feelings which have measurable effects on the team's performance, and to construct ideas for improvement based on these observations. It will not be useful if it devolves into a verbal joust, or a whining session.
- On the other hand, an effective retrospective requires that each participant feel comfortable speaking up. The facilitator is responsible for creating the conditions of mutual trust; this may require taking into accounts such factors as hierarchical relationships, the presence of a manager for instance may inhibit discussion of performance issues.
- Identical issues coming up at each retrospective, without measurable improvement over time, may signal that the retrospective has become an empty ritual.



Organized Comms & Cadence

Common Pitfalls:

- Being an all-hands meeting, a retrospective comes at a significant cost in person-hours. Poor execution, either from the usual causes of bad meetings (lack of preparation, tardiness, inattention) or from causes specific to this format (lack of trust and safety, taboo topics), will result in the practice being discredited.
- An effective retrospective will normally result in decisions, leading to action items; it's a mistake to have too few (there is always room for improvement) or too many (it would be impractical to address “all” issues in the next iteration). One or two improvement ideas per iteration retrospective may well be enough.

