

Workshop Report on *Mob Programming* Team 6 - SumUp

Nico Hambauer





The Concept and Expected Outcomes of our Mob Programming Workshop

The topic was chosen for three major reasons:

- The team thrives on joint coding, yet our (pair / team) coding sessions were rather unstructured compared to the regular team meeting.
- We occasionally created a divergent codebase and needed to resolve merge-conflicts.
- During sprint retros we frequently asked to increase knowledge-sharing and joint problem solving. We want to achieve well-balanced collaboration during coding sessions.

Outcomes we hoped to achieve:

- Even more fun & engagement while coding together.
- Encourage preemptive solutions and decisions to avoid merge-conflicts.
- Fair balance in coding sessions. Use a rotating roles framework for fair decision making.
- The team has developed a tailored Mob Programming approach.



Interactions, Learning Objectives and Measures

Interactions we had during the workshop:

1. During the Icebreaker, the team paints an image with “dots” in Miro.
2. Using sticky notes, we collaboratively identified how we coded in the past sprints:
 1. What was good about it?
 2. What could have been done better?
3. Third, we voted and prioritized on proposed improvements.
4. Forth, as a Workshop facilitator I introduced the team to Mob Programming essential values and let them rank these using sticky notes. During this slot we were running out of time, therefore we adjusted this on-the-fly.
5. Fifth, we collectively formulated and voted on how to design our own Mob Programming implementation.



Interactions, Learning Objectives and Measures

Learning objectives and measures:

- Structured Mob Programming Sessions:
 - Each Retro I ask the team how often did they code together using Mob Programming and how did they like it.
- Shared responsibilities for generating code.
 - The responsibilities are tracked on the feature board.
 - The Team, including me as the SM can measure / assess this balance during the sprint reviews and IP meetings.
 - Ideal case: Two or more people complement each other in presenting and taking responsibility for a large ticket / backlog item.



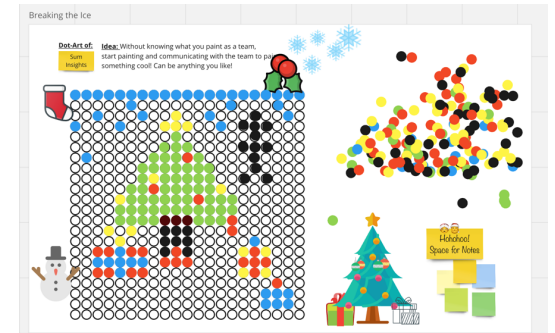
Short Tour Through The Workshop!

[Miro Link](#)

What Worked, What Didn't Work

The structure shows highlights in bold:

1. About the Workshop (3min)
2. Warmup (7min) → **The team had fun and loved it**
3. Introduction to Workshop Topic, Structure and Purpose (10min)
4. Identifying today's Goals (20min)
→ **Nice outcomes, but longer than anticipated. Draws in Voting which is a pity, because opinions might get lost.**
5. The Framework and Values of Mob Programming (20min)
→ **This session was shortened by me on-the-fly. We took a short cut and skipped a divergent & convergent part. The team agreed this makes sense and I asked for confirmation.**
6. Implementing Mob Programming at SumInsights (15min)
→ **Went very well and resulted in tangible outcome**
7. Closing & Next Steps (10min)
→ **Set the outcome into action within the next sprints**





What I Will Do Differently Next Time

Structure and Concept:

- Looking at the scope of time I will reduce this to 80% of the planned time slot. I learned that workshops would take longer than one anticipates. This may be due to the amount of input one receives from the team.
- I would like to slightly reduce the sticky note sessions and try to introduce one break-out sessions to get people (inter-)active again.

Content:

- I tried to fit in too many perspectives in terms of the goals I wanted to achieve.
- We first had a lengthy revision of past coding sessions and identified what went well and what could be improved. This was a nice starting point to know what worked and what we want, but it was too long. I will reduce such sessions in the future.
- Implementing a custom version of Mob Programming was enough already.