

Mob Programming Cheat Sheet

Created by Coaches and Developers @IndustrialLogic

Version 0.1

MODERN AGILE



What is Mob Programming?

"All the brilliant people working on the same thing, at the same time, in the same space, on the same computer." - Woody Zuill

"Mob Programming is continuous integration for ideas." - Joshua Kerievsky

There are many ways to successfully mob. In general, there is one computer, a keyboard and mouse, one or more monitors, a whiteboard, one Driver and one or more Navigators sharing a work environment like the one below:



Image by Mark Pearl

Ideal Mob Size The whole team, although teams of 3-6 people is ideal. Beyond 6 people, a Mob may have difficulties keeping everyone engaged. You can counter this with quicker rotation times.

Driver's Role - The Driver operates the computer to input/implement ideas made by the Navigator(s).

Navigator(s) Role - Navigators direct the Driver. Usually it works to allow everyone in the Mob to interact directly with the Driver. If that causes too much chaos, have one Navigator give directions to the Driver and the Navigator serves as the voice of the Mob. Newbies to mobbing can ask for help on how to navigate.

Driver Dos & Don'ts

- Drivers don't navigate. If the Driver is the only one who knows what to do, they can relinquish their role as Driver to the next person in the rotation.
- Each Driver can share how they prefer to be directed, including asking questions about intent, location and details. Ultimately, navigators must communicate in a way that allows the Driver to understand and take action.
- If no one is navigating, the Driver must stop typing.

Navigator(s) Dos & Don'ts

- Navigator ideas can only be programmed by going through the Driver's hands.
- Navigators must pay attention to the Driver's skill level. If the Driver needs word-by-word instructions, the Mob must explain in detail what the Driver needs to do. If the Driver is more advanced, Navigators give higher-level instructions, like "commit it" or "move that method to the parent class". Over-navigating

entails too much direction, under-navigating too little.

- Don't sit and watch the Driver work. *Everyone learns and contributes in a productive Mob.*

Mob Responsibilities

- Treat everyone with kindness, consideration, and respect
- Plan, discuss, research, and work out ideas on the whiteboard
- If a Driver begins to navigate and drive simultaneously, someone else in the Mob can call them out on it.
- Keep the Mob going as people join or leave
- Stakeholders, Managers, Subject Matter Experts, etc. are welcome to join a Mob, but are not required to drive or navigate.

Leaving/Joining the Mob - It is fine for someone to leave or join the Mob for whatever reason. If they are a Driver, they relinquish that role to the next person in line. The person leaving should tell the team when or if they'll return so that the team can adjust the rotation schedule.

Switching Drivers - The Driver switch over shouldn't take more than a few seconds. The teams should determine the best way to accomplish that goal. Switching Drivers is easiest when the Mob shares a single station and settings. If individuals in the Mob have their

own preferred tools and settings, consider switching machines using a hardware HDMI switch, or use a system like WebEx or Zoom.

Physical Space - Here are some suggestions for the physical space and equipment setup:

- At least one large screen (big enough so that people can easily read code at a distance) is necessary. Some teams use two or more screens. Projectors can work, but are less desirable.
- Consider using wireless keyboards and mice to make it easier to switch Drivers.
- Mobbing can be noisy. Ideally, the team is in a room with a door that closes or is somewhat apart from other's work areas. Conference rooms are usually suboptimal. They are designed for people to sit around a table and talk, not to program. You need to face the screen (without turning your head) as you work, and it's best if participants can sit side by side to facilitate discussions.
- You need chairs or couches that are comfortable for longer than an hour.
- You need a large whiteboard and plenty of sticky notes.

Mob Timing and Breaks

- Use an automated timer (e.g., Dillon Kearns' Mobster App) to initiate role changes and breaks.
- Switch roles every 7 minutes on average (beginners should switch every 2-4 minutes).
- The whole Mob should take regularly scheduled breaks. The pomodoro method, a proven method of taking regular breaks to increase efficiency, suggests taking breaks every 48 minutes.

Bias for Action - When discussing how to solve a problem, get out of the abstract as soon as possible.

- Do not argue for more than 10 minutes.
- If there are 2 ideas, try both, then decide which the Mob likes better.
- Keep the Mob moving with this quote from Brian Marick: "An example would be handy right about now".

The Value of Mobbing - When done well, mobbing helps a team:

- Deliver solutions faster by increasing focus, building skills and sharing knowledge.
- Produce better quality code because the entire team reviews the code as it is being written.
- Feel the pain of tedious tasks. This is good, as it biases you toward fool-proofing and automation.
- Cross train it's members and remove knowledge silos by learning together.
- Deliver results faster by reducing the team's "work in progress" and by eliminating delays from handoffs with the whole team present.

Mobbing Pitfalls - A poorly functioning Mob will produce value slowly. The following are some signs of poor mobbing and what to do if you observe these behaviors:

- Excessive discussion or arguing - Run some experiments, then decide which the Mob likes better.
- Zoning out - Take a break from the Mob
- Ignoring Mob roles/timers - Team members should hold each other accountable
- Producing poor designs or not valuing good design - A Mob that lacks people with good design skills won't magically produce good

designs. To improve the design, the Mob should get expert help.

- Conflict between team members - In order to be a well-functioning Mob, everyone must be treated with kindness, consideration, and respect.

Don't Stop the Mob - The Mob can temporarily delegate a member as a researcher, to find solutions to something they cannot easily figure out. Meanwhile, the Mob can work elsewhere in the code. It is important that people feel comfortable enough to ask questions, but if the Mob is moving slowly due to a lot of questions, it is better to set aside some time outside of the Mob for someone to have questions answered. Fast throughput is an important goal.

Invite Experts - If a Mob gets stuck, they may invite an expert to join them to help resolve a problem. Be sure it is an invitation and not a demand (they are not required to drive).

Resources

- Mob Programming – A Whole Team Approach, by Woody Zuill and Kevin Meadows
- Code with the Wisdom of the Crowd: Get Better Together with Mob Programming, by Mark Pearl
- The Mob Programming Guidebook, by Maaret Pyhäjärvi
- Mobster - free tool that helps teams manage rotations and breaks (<http://mobster.cc>)



This work is licensed under a **Creative Commons Attribution-ShareAlike 4.0 International License**.