

Building Data Vault modelling capability through the Mob



A great way to improve modelling decision making and shorten the time it takes to deliver add-ons to the enterprise data vault model is a concept borrowed from extreme programming called “mob programming”.

The customer that I work with read an article about how effective mob programming has been for coders and product owners that he decided to introduce the idea of mob programming to develop an enterprise data vault model.

First of all, what is mob programming?

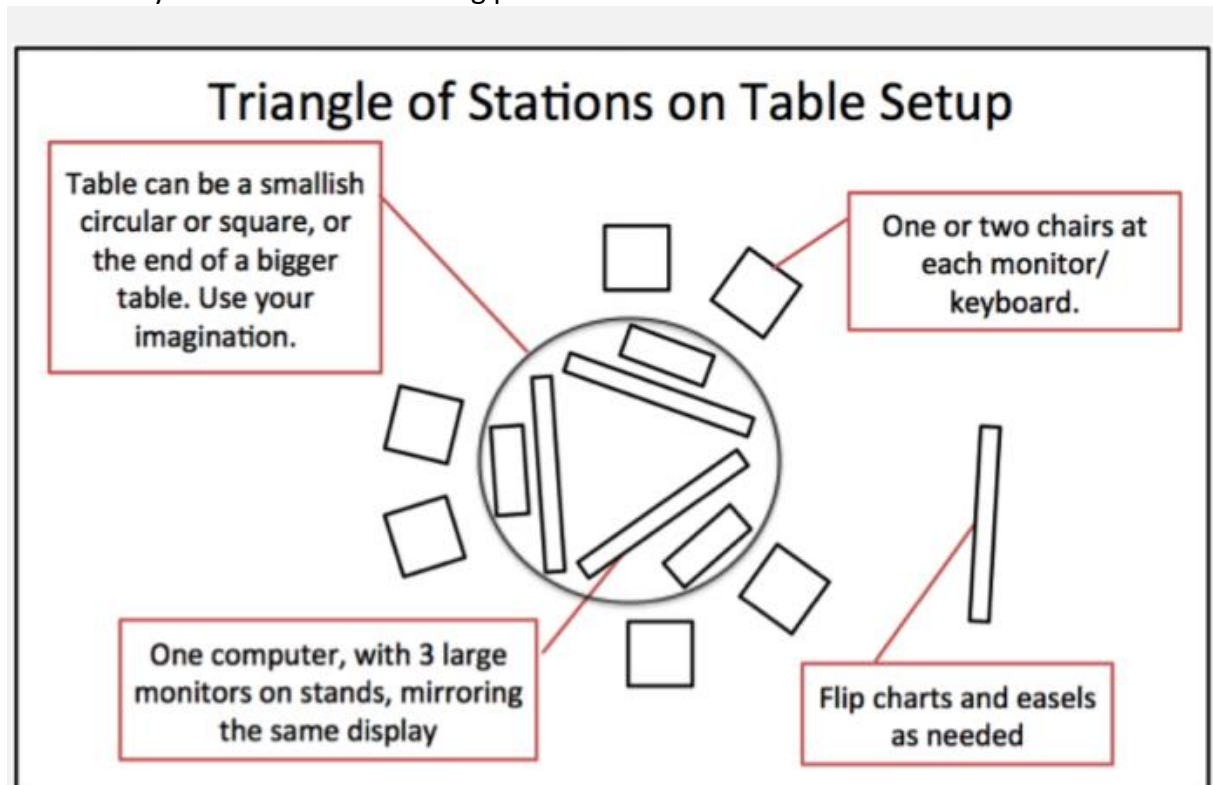
“Mob programming is a software development approach where (all the brilliant minds working on) the whole team works on the same thing, at the same time, in the same space, and at the same computer” — Woody Zuill, founder of mob programming.

With mob programming the collaboration is extended to everyone on the team, while still using a single computer for writing the code and inputting it into the code base. Mob modelling is based on this approach and adapted to model tables into data vault. The roles identified below are scheduled to work together for a determined period of time (typically 1 to 2 hours) and all stakeholders involved gain the benefit of learning together and making decisions together.

Guidelines

Modelling “Explore” phase

1. Agree on a scheduled time where all participants are available
2. Sit all members in a semi-circle in front of a big screen
3. Choose a driver; the person controlling the keyboard
4. Change the driver from time to time (optional), usually in 15 minute intervals
5. Only one conversation taking place at a time



Participants

- Subject Matter Expert (SME) — modelling business logic, relationships and attributes correctly depends on domain knowledge from the SME
- Technical business analyst — understanding how the business rules are applicable to the source and how they will need to be represented in data vault. Identification of technical debt, reporting requirements, business glossary updates etc.
- Data modeler (driver) — ensuring that the meeting meets its objectives in the allotted time block and that discussions do not become exhaustive (cannot be solved without more information). Keeping track of decisions made, action items and tasks assigned.
- Guardians — action takes, action to be followed-through after the mob session has concluded; these can be information gathering tasks or tasks that have a tangible outcome (even sending email)

Challenges

- Keeping focus; falling into everlasting discussions, trying to find the perfect solution, or zone out when we are many. The more chefs the worse the soup... Good enough for now, safe enough to try. Try a solution instead of trying to discuss your way to the

“best” solution. Time-boxing discussions. Park discussions if they are not relevant right now. Learning from each other’s areas of expertise. Stepping out of one’s comfort zone and feeling committed to what’s beyond ones expertise. Visualize. Visualization of concepts, ideas and solutions will be important in order to explain to everyone in the mob. In order for everyone to follow where we are and where we are heading. Whiteboards are remarkable.

- When shared responsibility becomes nobody’s responsibility; guardians are assigned to follow up on action points
- Ego and the need for control; dare to let go over design details and gain control over the whole picture. Facilitate rather than controlling design in detail. Instead of saying “this won’t work”, offer a solution

Benefits

- 100% understanding of the problem / task within the team — everyone focuses on the problem and can make decisions that take us towards the goal. Precise specification of requirements are no longer needed. The knowledge is in the mob — make sure the knowledge is captured
- Engagement throughout the whole process and an understanding of why we choose a particular course or solution
- Easier to focus on one “right” thing at a time. Easier to make better-informed decisions
- It helps us keep direction. Fewer irrelevant side tracks. We get better at prioritizing unnecessary fluff, or those that do not have top priority
- Less explanatory and fewer misunderstandings when the team is involved all the way through the process. Avoid email chains if all the needed roles are involved
- Greater opportunity to influence the final result. When you are in the mob you can create trust by creating interest and understanding of others areas of expertise. This makes it easier to influence and ask questions.
- No one in the team becomes “indispensable” when we learn from each other. The mob can go on even when everyone is not in place. At least for a while
- Fewer surprises along the way when all competencies have taken part in decisions we take along the way
- It is easier to avoid unrealistic specifications from designers to developers or requirements from the organization on features that users do not prove to want. Or developers who build solutions that are difficult to manage or further develop.
- We do not have to wait. Of course we can still be blocked by decisions that need to be taken by the organization, other requirements or other teams
- The model developed will be closer to the final model because the roles are present in the decision making.
- Remote resources are dialed in and benefit from contributing and learning from the experience
- Ensures that work, knowledge creation and decision making are joined together thus leading to much better alignment and more effective action.

When we started employing this technique the session took 2 hours to model 4 tables but as we grew together we found that we could rapidly model more tables in a shorter time. The mob proactively explained the business scenarios and problems through. Now the roles and function of the mob has become second nature and we could easily get through 7 or 8 tables in 30 to 40 minutes! And we only started this approach about a month ago so I expect the momentum will only improve!

The views expressed in this article are that of my own, you should test implementation performance before committing to this implementation. The author provides no guarantees in this regard.