

A Mob Mentality Five Years On

Oli Wennell
@owennell



Hi everyone, thanks for coming



Oli Wennell
@owennell



I'm Oli Wennell, a tech lead at comparethemarket.com
We're a price comparison website
We're the ones with the adverts on the telly
Meerkat adverts, that is

I'm here because 5 years ago, someone in my team suggested we try something radical.

Apparently over on the west coast of the USA, there was a team that weren't just pair programming.

They were all working together on the same thing at the same time on the same computer

The idea was mob programming, also known as mobbing

I thought: nice idea, not going to catch on

And yet...

five years later, the idea's not so radical anymore.

It spread around the world

It's changed how I think about software development.

I'm a massive fan

So that's me, now a bit about you.

Could I ask, how many of you have heard of mob programming before?

How many have tried it?

How many use it regularly?

Ok, thanks

Three Examples

@owennell



I'm going to tell the story of three groups of people at comparethemarket

They all tried mobbing
I'm going to talk about what went well
what went...not so well
and what they learned

I'm hoping that these will encourage you to
give mobbing a go,
or if you already have, find ways to experiment and perhaps make your mobs even
better.

If you have any questions, I'd be happy to answer them at the end
Or come up and talk to me afterwards, I'm here for the whole conference

Mob Number 1

The New Team

@owennell



new team created to gradually rewrite a key system



@owennell

 comparethemarket

i am one of the developers
(i'm the one that's smiling)

We all know bits about domain, tech stack



@owennell

 comparethemarket

Luckily we also had an expert in the domain/the existing system



@owennell

 comparethemarket

One day, the expert went on a long holiday



@owennell

 comparethemarket

And we very quickly got stuck



@owennell

 comparethemarket

We are doing pair programming
Two people to one job
One person types, or “drives” and the other person navigates, or says what to do.
Usually the pair swap after a while.

The trouble is:
It's a new project, we need to make group decisions.
And the pairs don't have enough context to do their work, it's spread around the team.
So we interrupt each other all the time.
Don't seem to get much done.



@owennell

 comparethemarket

So someone said
“There’s this idea, where we all work on the same thing at the same time”

Mob Programming

@owennell



If pair programming is when two people work together on a task at the same time.
Then mob programming takes it to the next level
By adding more people

One person still drives, but there are more people there to support that driver



@owennell

 comparethemarket

And we loved it

Naming Things

@owennell



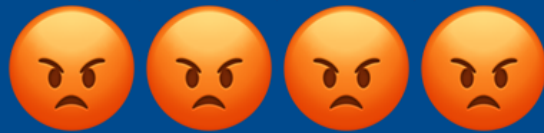
Now, mob programming is a bad name
And that's because your first reaction when hearing about it may have been...



@owennell

 comparethemarket

Gangster movies



@owennell

 comparethemarket

Or masses of angry people with pitchforks



@owennell

 comparethemarket

Whereas actually, mobbing is the opposite of these things.
It is designed to be an inclusive way of helping people work together in harmony.
There may or may not be actual unicorns.

Unicorn programming

*“There are only two hard things in
Computer Science: cache
invalidation and naming things.”*

Phil Karlton

@owennell

 comparethemarket

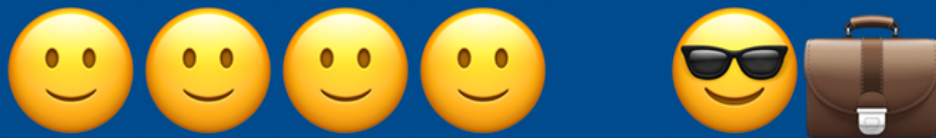
There's a famous quote about this, which is...

There are only two hard things in Computer Science: cache invalidation and naming things

And I guess that applies to techniques as well.

The danger is that people hear mob and get put off
But I think we're stuck with the name

When you hear mob - think "the general population"
We want everyone to get involved



@owennell

 comparethemarket

Anyway, back to the team.
And the expert is back from holiday, hooray!



@owennell

 comparethemarket

But instead of going back to our old ways,
we continue mobbing

Why Carry On?

Better 

Faster 

Stronger 

@owennell

 comparethemarket

Why would we carry on, now that we have the expert back?

It feels great

I for one feel like "I'm in the zone"

Doing my best work, together with the team

When we are mobbing we were:

Better

Faster

and stronger

Better

@owennell

 comparethemarket

Code quality - all eyes on every line, more chance of spotting better ways

Everyone's context of domain, tech, techniques

We get the best out of everyone, which makes the software better

Faster

@owennell

 comparethemarket

Mobbing is a way of shortening feedback loops

Decision making in real time instead of meetings

Code reviews in real time instead of a bottleneck

Stronger

@owennell

 comparethemarket

Less impact when people off sick or on holiday, everyone has context all the time
Feels good to know it's not "all on you" to be able to fix things if things go wrong

Team is new but we get to know each other quicker by working together so much

But...

@owennell

 comparethemarket

But we started to find there were some tricky aspects about mobbing

Intensity 🧠💥

@owennell

 comparethemarket

Mobbing is intense

Conversation with multiple people instead of just your pair
Going on for too long, we'd get exhausted

So we started to take regular breaks
And keep sessions short, say 30 minutes.

Safety 🥲

@owennell

 comparethemarket

Mobbing can be intimidating

I'm not going to, but imagine if I asked for a volunteer to come up and code in front of everyone. I think a lot of people would want to hide.

We don't want people to feel stressed or nervous
We want people to be at ease so they do their best work

We found it important to have psychological safety
To be kind with the driver
To be patient with the driver
Pairing well is a skill. This is even more important in a mob

Mob Number 1



Amplification

@owennell

 comparethemarket

So we learned how to mob
and we love it!
Better code
Happy team

Mobbing takes all the good things about pair programming and amplifies them
But, if you're not careful, it can amplify the bad things too

Mob Number 2

A New Role

@owennell



Now onto the second mob

This time they were an established team

It's a bit later on, and I was in a new role. A tech lead.

Woody Zuill, who was part of the team that discovered mobbing, came in and gave some training and there's a buzz in the air.

And the team are keen to give it a go

So I'm happy about that

And, it turns out, as a tech lead there were things about mobbing that made my life easier...

Context 🙄

@owennell

 comparethemarket

There's much less context switching.

With pairing, I need to keep an eye on multiple pairs, to see if they need help or are going down a wrong path.

With pairing, we can have multiple stories in flight that need to be coordinated and deployed and so on.

But with mobbing, there's one thing going on at a time.

In Lean there's the concept of limiting your work-in-progress, and how that can help things.

Mobbing takes that to the extreme: a work-in-progress limit of one.

Staying Technical

@owennell

 comparethemarket

As a tech lead, I'm often popping out to meetings
If I keep leaving a pair, it can be disruptive.
So to avoid this I'll often find myself picking up small jobs on my own, rather than
being a bad pairing partner.
Not great for me or the team

But with a mob, it's much easier to dip in and out.
Because when I leave there's much less of an impact.
And so mobbing helps me stay technical.

Coaching

@owennell

 comparethemarket™

Part of my job is to help the developers on the team improve.
With pairing, I don't get much of a chance to see how they work
But with mobbing, I can see the entire team at once.

But...

@owennell

 comparethemarket

But we came across some problems I hadn't seen before

Too Crowded 🙄

@owennell

 comparethemarket

The team needed some specialists, and it started to grow.
Our first reaction was to get them to join the mob.
But it started to feel crowded

Hard for everyone to contribute
People were starting to disengage
And it was hard to physically fit everyone around the screen (it was a big screen!)

Unsuitable 🤔

@owennell

 comparethemarket

This was an established team, existing systems that needed small updates
Mobbing not as valuable because it was well known, established, not interesting
The team tried mobbing on these, but it was quite dull. Again it was hard to stay engaged.

Inflexible 🙈

@owennell

 comparethemarket

Sometimes we'd come across things they we were new to.
Our instinct was to stay in the mob, to try and do research as a mob
But watching someone read the documentation is pretty boring.

New Phrases

@owennell

 comparethemarket

Luckily we found ways to solve these problems.

These are best summed up by a couple of phrases to our team's vocabulary.

“Demob”

(di•mob, verb)

@owennell



The first was “demob”
Which means to split up a mob

I think we’ve got a good idea of how this will work end-to-end now. Let’s demob, one pair finish the frontend, one finish the backend.

If we need to use an unfamiliar bit of tech – lets demob, read the docs, then come back later and share what we found

Someone from another team has a question about the library that we wrote – I’ll demob and show them, you folks carry on mobbing

“Mobabble”

(mobb•a•ble, adj)

@owennell

 comparethemarket

The other term was “mobbable”
Whether a story is worth mobbing on.

Bit of a tongue twister!

This story coming up in the next sprint will require a big refactor, it’s definitely mobbable.

I thought this story would be easy, so started it on my own. But there’s lots of edge cases I hadn’t considered. I think it’s mobabble now.

Mob Number 2

🙄😴🙊 - Right Team For Each Job 👍

@owennell

 comparethemarket

So mobbing has started to spread.

It works in multiple teams

It helps teams and those on the edge of teams, like myself as a tech lead. Not just those directly in the mob

But a flaw was that we got fixated on mobbing as being the only way to work.

We had a hammer, and everything we saw was a nail.

We got better when we started treating mobbing as a tool,
and continually thinking: what size team is right for this job, right now

Mob Number 3

The Dojo

@owennell



So the next mob isn't a team in the traditional sense
It's a coding dojo

Coding Dojo



@owennell

 comparethemarket

A coding dojo is where people get together to practice programming.

They do it to learn, to teach, and to have fun.

Earlier this year I setup a dojo in my office

It runs every Friday for 90 minutes

People from around the company come along

We work on katas, which are small programming challenges, together.

Experienced

@owennell



A lot of dojos will have people working on their own, or pairing
But by now, mobbing has spread around the company
Most people who come along to the dojo will be used to mobbing regularly, or at least have tried it.

So I thought, let's try mobbing in the dojo.
And it worked really well
The first people who mobbed did it first in a coding dojo. They wanted to accelerate their learning.

But...

@owennell

 comparethemarket

But a coding dojo is different to a normal team environment

Strangers 🛸

@owennell

 comparethemarket

First off, although there are some regulars, we get different people at most sessions. So we have to be able to start working together quickly, because we only have 90 minutes

Theatre

@owennell

 comparethemarket

Also, imagine there's only one person in the mob who understands how to solve a problem.

If that one person is doing all the talking, or maybe also doing the typing as well, it's hard for the others to stay engaged.

Woody Zuill, who was part of the first team that did mob programming, calls this "Programming Theatre"

Now programming theatre is fine for, say a demo, where an expert shares their knowledge.

But to get the best out of everyone, we'd rather everyone was involved.

Consensus 🖐️

@owennell

 comparethemarket

It's good for a mob to debate things.
That's part of what makes mobbing useful

But it's especially important in a dojo, to not go on for too long., otherwise we'll
spend the entire dojo talking.
We need to be able to quickly come to a consensus.

Continuous Improvement

@owennell



So as before, we changed how we mobbed to try and address these issues.

Rules

@owennell

 comparethemarket

Strong style is a technique where you make the roles of the driver, who types, and the navigator, who says what to do, strict.

The driver is just typing

In the case where one person is the expert, when they're driving, it encourages others to speak.

Tweak Rules !

@owennell



As well as following rules more strictly, we also tried tweaking other rules.

Whereas in a normal team environment, we might swap the driver every 30 minutes or so,

In a dojo environment, we found shorter rotations, say 10 minutes were better to mix things up

We'd have micro retros after each rotation.

We also experimented with rules that make things fun but also each new skills, like today no using the mouse, we've got to figure out all of those keyboard shortcuts today no if statements are allowed

Facilitation

@owennell

 comparethemarket

Like any other kind of meeting, a facilitator can get the best out of people

They can suggest things when they get stuck

They can help when people can't agree

They can remind people to swap drivers, or take a break

They can encourage people who are a bit quiet to give their opinions

Facilitator doesn't have to be a single person

It can be a few

Or everyone

What's important is that facilitation is done, and the mob doesn't stagnate.

Mob Number 3



@owennell

 comparethemarket

The dojo is a bit of an extreme situation: potentially strangers building something together in 90 minutes

But what we learned can be applied to normal teams, which is that we should experiment

Every mob is different, so it makes sense to experiment and find the process that make your mob work the best.

And this process needs to continually evolve as the team and the situation change. Good facilitation helps teams do this.

*“We are uncovering better ways of
developing software...”*

- The Agile Manifesto

@owennell

 comparethemarket

The first bit of the agile manifesto says “We are uncovering better ways of developing software”

And what’s best for me may not be what’s best for you.

(Nearly) The End

@owennell



So coming to the end, now

Mobbing has gone from an experiment, to being commonplace at comparethemarket

Some mob by default

Some mob on tricky things or as an exception

Some use programming theatre

It's caught on

When I talk to people they mention the same themes that I found in my mobs...

Amplification

Right team for each job

Experiment

@owennell

 comparethemarket

Mobbing is an amplifier

It amplifies good things, but beware of amplification the bad things too

Mobbing is a tool you can use at the right time and in the right place

Each mob, mobs differently. And that's a good thing. Find what works for you and keep experimenting.

The Future

@owennell

 comparethemarket

I've talked about three mobs over time
But what's going to happen next?

Community

#mobprogramming

@owennell

 comparethemarket

But best of all, there's a whole bunch of people around the world trying out mobbing
They're experimenting and sharing their experiences
And we can all learn from each other

Your Turn?

Pick a Topic 
Facilitate 

@owennell

 comparethemarket

So maybe you've not mobbed before but you'd like to have a go?

First off you'll need to pick a topic. Maybe you could...

Do a kata, to have fun and learn together

Or maybe there's some legacy code that people are scared to touch on their own

Or maybe there's a new technology the team needs to get up to speed with

And then help the team by being a facilitator

That could involve finding a nice place with a big screen for them to use

Encouraging them to take regular breaks and have a retrospective

Links

#MobProgramming on Twitter

The Mob Programming book

<https://leanpub.com/mobprogramming>

How one team mobs remotely

remotemobprogramming.org

Mob Mentality Show

<https://www.youtube.com/channel/UCgt1IVMrdwIzKbaerxxp2iQ>

(I wasn't involved, the name is a coincidence! 😊)

@owennell

 comparethemarket

And here are some links that you may find useful.

I'll be tweeting these slides out on the Agile Cambridge hashtag on twitter shortly

Any questions?

Thank you