EXPERIMENTING WITH MOB-PROGRAMMING

AMR NOAMAN
Amr Noaman Abdel-Hamid

Email: amr@agileacademy.co
Blog: amr.agileegypt.org
Twitter: @AmrNoaman

Agile practitioner, coach, trainer, consultant, writer and lecturer

Co-founder of
Agile Academy
Egypt Lean and Agile Network
Agile Egypt

Co-initiator of Egypt’s GoAgile program at 2011, to boost lean & agile software development in Egypt

Member of the Conference Program Team –
Development Practices and Craftsmanship –
Agile 2017 & Agile 2018
Let’s do Mob-Drawing!
Agenda

• Brief introduction to mob-programming
• Preliminary feedback from early experiments
• Quantitative results from later experiments
• Final notes and key take-aways
Mob programming is a software development approach where the whole team works on the same thing, at the same time, in the same space, and at the same computer.

- wikipedia
Monday, August 7 • 3:45pm - 5:00pm

Onboarding with the Mob (Sheldon Fuchs, Ravdeep Sekhon)

Click here to add to My Sched.

http://sched.co/ATWf

Tweet  Share
Mob programming is a software development approach where the whole team works on the same thing, at the same time, in the same space, and at the same computer.

- wikipedia
How did it go?
It was fun! and joyful
Some tasks would have never been carried-out without mob-programming them
We used to do the same task again and again so many times. This way, we all do it one time!
Collaboration with the product owner is much more effective; we all ask him the same question, get the same answer, no confusion, no misunderstandings.
It's a better approach to reach the best solution for a problem.
You have an idea, you don’t know if it may work

Let’s do an experiment!
Rules of the Game

• The same physical setup – with me as a facilitator
• We'll break every now and then, probably once every 1:30-2 hours
• It's ok that you leave the room for any side task, not for so long though
• Work is time-boxed from 10 am till 4 pm. No other "significant" work should take place outside these working hours
• Any emerging unplanned task will be put on the backlog, period!
Initial Backlog

7 tasks

The backlog after two days of work

26 tasks
## Readings and Results

<table>
<thead>
<tr>
<th>Team</th>
<th>Task</th>
<th>Effort (hours)</th>
<th>Duration (days)</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Mob</td>
<td>Solo</td>
<td>Mob</td>
</tr>
<tr>
<td>Digipro</td>
<td>Refactoring task</td>
<td>56</td>
<td>54</td>
<td>2</td>
</tr>
<tr>
<td>Digipro</td>
<td>Bug fixing of one bug</td>
<td>2</td>
<td>10</td>
<td>0.3</td>
</tr>
<tr>
<td>Digipro</td>
<td>Architectural investigation and prototype</td>
<td>84</td>
<td>200</td>
<td>1</td>
</tr>
<tr>
<td>Ebrufai</td>
<td>Bug fixing of one bug</td>
<td>0.9</td>
<td>0.6</td>
<td>0.2</td>
</tr>
<tr>
<td>Ebrufai</td>
<td>Developing Feature 1</td>
<td>20</td>
<td>18</td>
<td>1</td>
</tr>
<tr>
<td>Ebrufai</td>
<td>Developing Feature 2</td>
<td>50</td>
<td>56</td>
<td>3</td>
</tr>
</tbody>
</table>
Comparing Duration

- Refactoring task
- Architectural investigation and prototype
- Bug fixing of one bug (*10)
- Bug fixing of one bug (*10)
- Developing Feature 2

* Mob
* Solo
Learning and Knowledge sharing
Does it always work?

• Investigation/research tasks which requires group thinking
• Large refactoring effort of code with high technical debt
• Development tasks which requires different sub-specialties from more than one person
• When doing a new type of task/business area at least for the first couple of times
A final note!

One important trait of high-performing teams is utilizing their diversity in creating the best solutions.
Where else mobbing can be useful?
A step by step guide to clean coding techniques and professional development habits

https://leanpub.com/RefactoringToCleanCode
Keep In Touch!

amr@agileacademy.co
amr.noaman@gmail.com

www.agileacademy.co
Blog: amr.agileegypt.org
@amrnoaman
Thank you