


What I have learned from this training

Things to consider apart from code smells or mind
blowing tests



preparation

do we really need a mob for this?

- is our (everyone's) IDE ready and running
- code checked out
- does the maven build run
- are all existing tests green
- is our test-suite running
- do we really need to refactor this
 - e.g. is it still used

=> to ensure that we are ready to start

Low Hanging Fruits

things which shouldn't do damage

- beautify the code
- remove commented out code
- rename to meaningful names(in our example variables sn, prc, ...)
- bracketize (is this a word? :) code-blocks

=> to gain understanding

At the beginning

- identify which code we have to refactor
- are there external dependencies we can't / mustn't use
 - e.g. exploding database if we call this method in test
- identify code-smells which might hinder us (maybe comment the code)
- do we have tests already / what about the code coverage

=> to do the right things / to not do the wrong things

while refactoring

- keep code clean
- keep tests clean
- use the IDE
- do the things we learned in this training :)

=> to do the things

Mobbing

we want to have a shared mindset, so:

- as navigator
 - explain what we want to do and why
 - if the driver is straggling, try to understand why and help out
 - ask the mob for help if you struggle
- as driver
 - explain what you're doing (especially if it's something
 - don't bother to ask if you do not know what's going on
- as mob
 - bring in your ideas
 - give hints for solutions/shortcuts/...

=> talk to each other

- take regularly breaks (mobbing is hard)
 - recap on return (what did we do, what was the goal, what to do next)

=> while doing the things