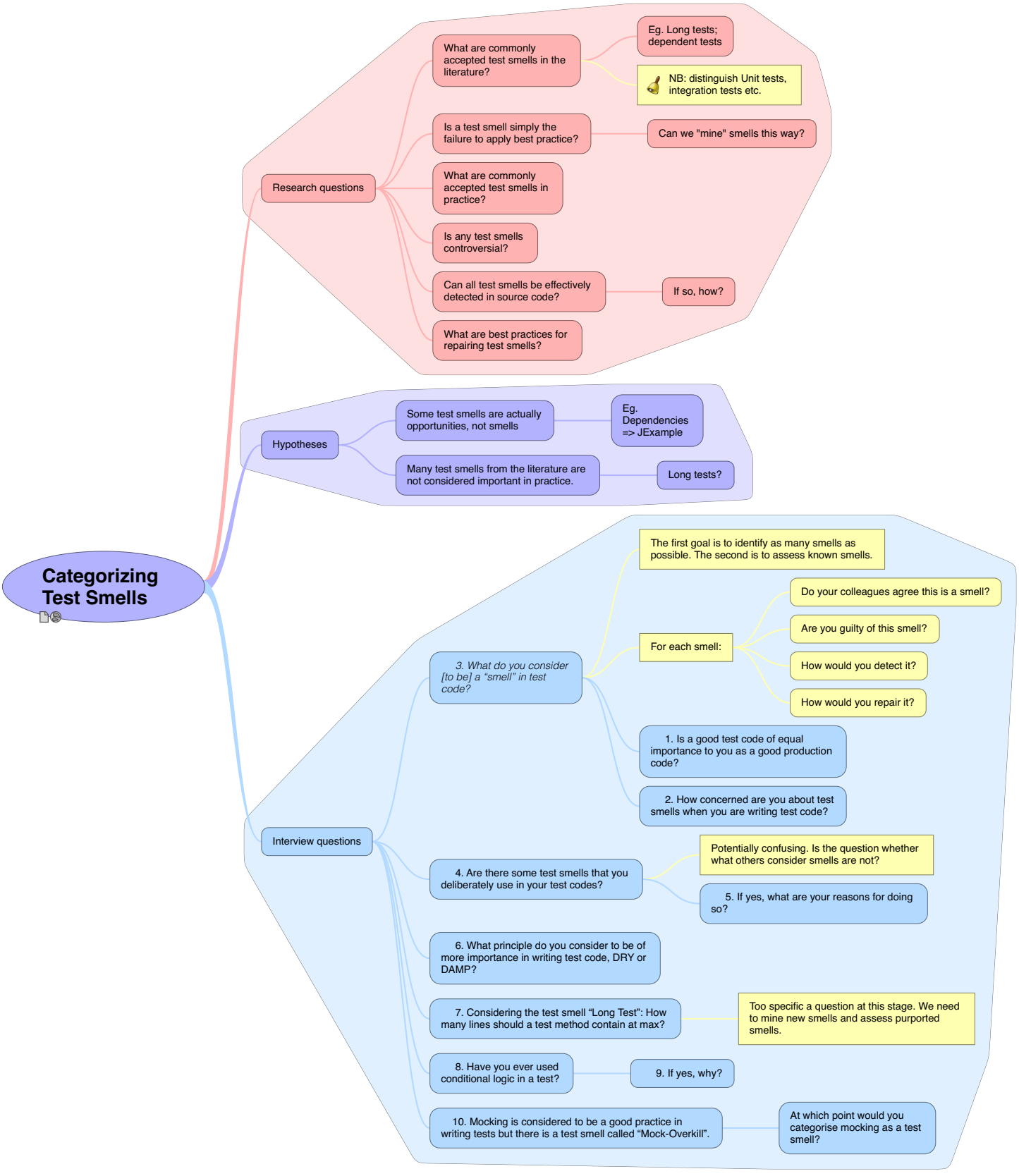
Categorizing Test Smells
Research questions
What are commonly accepted test smells in the literature?
Eg. Long tests; dependent tests
NB: distinguish Unit tests, integration tests etc.
Is a test smell simply the failure to apply best practice?
Can we "mine" smells this way?
What are commonly accepted test smells in practice?
Is any test smells controversial?
Can all test smells be effectively detected in source code?
If so, how?
What are best practices for repairing test smells?
Hypotheses
Some test smells are actually opportunities, not smells
Eg. Dependencies => JExample
Many test smells from the literature are not considered important in practice.
Long tests?
Interview questions
3. What do you consider [to be] a "smell" in test code?
The first goal is to identify as many smells as possible. The second is to assess known smells.
For each smell:
Do your colleagues agree this is a smell?
Are you guilty of this smell?
How would you detect it?
How would you repair it?
1. Is a good test code of equal importance to you as a good production code?
2. How concerned are you about test smells when you are writing test code?
4. Are there some test smells that you deliberately use in your test codes?
Potentially confusing. Is the question whether what others consider smells are not?
5. If yes, what are your reasons for doing so?
6. What principle do you consider to be of more importance in writing test code, DRY or DAMP?
7. Considering the test smell "Long Test": How many lines should a test method contain at max?
Too specific a question at this stage. We need to mine new smells and assess purported smells.
8. Have you ever used conditional logic in a test?
9. If yes, why?
10. Mocking is considered to be a good practice in writing tests but there is a test smell called "Mock-Overkill".
At which point would you categorise mocking as a test smell?

2014-09-22 Brainstorm w ML to discuss MK draft questions

http://scg.unibe.ch/wiki/projects/Categorizing-Test-Smells