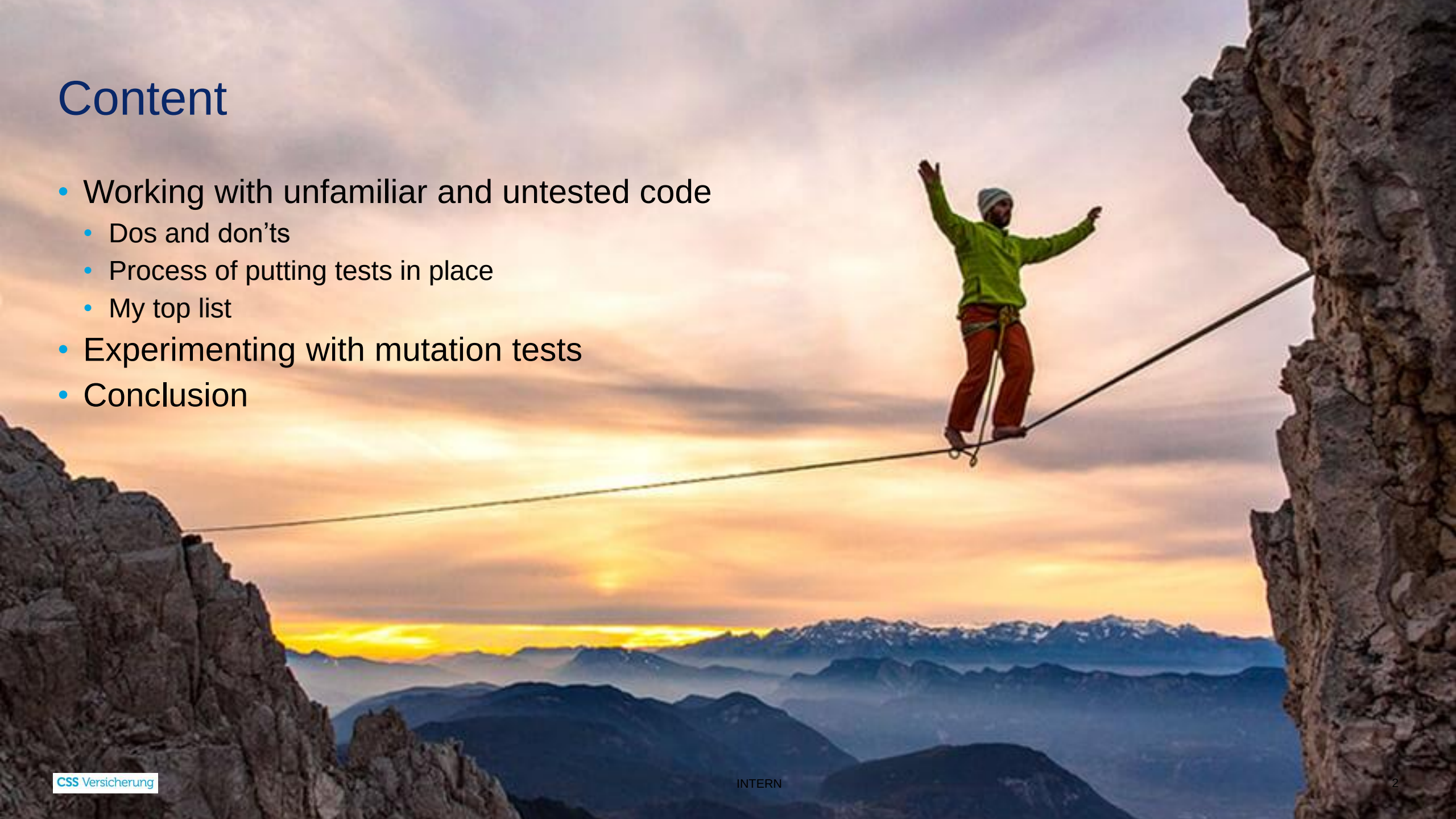
Balancing from Alcor code renovation course to work

30. Septembre 2021

Urs Birrer

Content



- Working with unfamiliar and untested code
  - Dos and don’ts
  - Process of putting tests in place
  - My top list
- Experimenting with mutation tests
- Conclusion

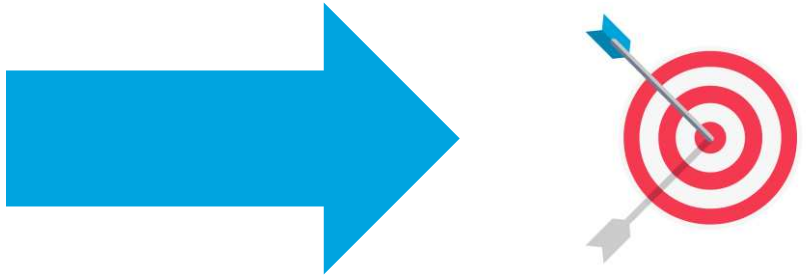
Dos and don’ts

- You should…
  - use the IDE for refactorings as often as possible
  - remember that legacy code «pays the bills»
  - know that every code without tests, is (kind of) legacy code

- You shouldn’t…
  - change the public signature
  - don’t fix a bug while refactoring

Process of putting tests in place

- Where is data read?
- Where is data transformed?
- Where is data sended?



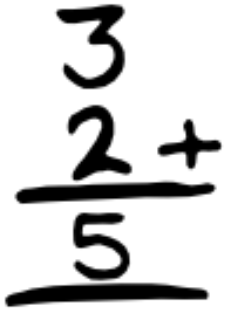
There is no magic, but sometimes it feels that way!

My top list

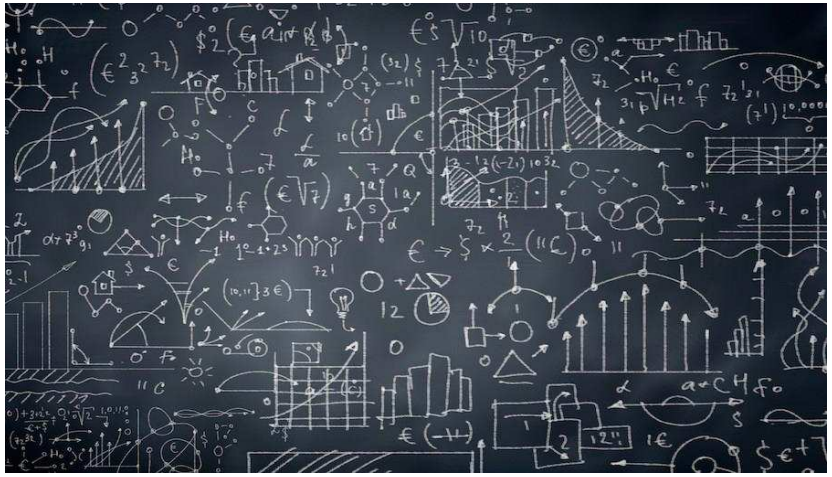
1. break out method object or primitivize parameter
   - smaller chunks are easier to test
2. extract and override factory method
   - hardcoded dependency pattern is often used
3. peel and slice functional parameter
   - «elegant» way to solve, but need’s more practice

Experimenting with mutation tests

- Goal
  - PiTest, how easy to enable?
  - Are the results help to find gaps?



vs



- PiTest on simple «only mapping» service (deckungssuche)
- Result:
  - found mutation problems on producer class

- PiTest on aggregat class with complex time range calculation and state (G365)
- Result:
  - found many mutation problems (>300 for 4 classes)

- I don’t recommend to use PiTest to find gaps in tests
- In my opinion it’s too difficult to find conclusions
- There is more effect, if you and your team think together about edge cases
- What are your experiences and recommendations?

Conclusion / Learning

- We are often focused on using mockito and blind something else
  - I’ll try to use the new technics, must be aware of patterns
  - before writing massive mocks, maybe separation of concerns makes your life easier
- Functional parameters are a nice solution.
  - I definitively need more experience
- We should more often lock behaviour before refactoring
- Don’t fear any beast

Thank you!

Literature & References

- https://spider-slacklines.com/images/slackline/highline.jpg
- https://monzo.com/static/images/blog/2018-07-10-making-quarterly-goals-public/q3-goals-blog.png
- https://s3.amazonaws.com/viking_education/web_development/prep_engineering/complex_problem_small.jpg
- https://lh3.googleusercontent.com/proxy/nVbr4JhWy8-QOlZf7kl6XrdDTF6nh5fAyuUtr-iOmM1WVkW1ZfEHBsIfLbEC9nd90dUzht3k_NsZfqUS3HsiHiOmK6elFexXJCk
- https://previews.123rf.com/images/hobbitfoot/hobbitfoot1907/hobbitfoot190702712/128498050-realistic-golden-trophy-with-winner-ribbon-vector-illustration.jpg