RESORBING THE TECHNICAL DEBT OF LEGACY TEST CASES WITH AI TECHNIQUES

Presented by Arnaud Bouzy
MOTIVATIONS

Why refactoring a manual test repository?
Tests Obsolescence

Test obsolescence means:

• No updates when the application evolves
• Lack of consistency due to multiple test designers
• Redundancy
• Incompleteness
• Too many or too few details
• Too long scenarios
When does test obsolescence become a burden?

- Technical migration of the application
  - New platform, new technology, same business needs
- Stronger need for automation
  - Increasing time to market, agile practice
- Knowledge transfer to another testing team
  - Whether internal or external
AI & GRAPHICAL REPRESENTATION

Algorithms and workflows
3 services to be delivered

- Step refactoring
- Syntactic analysis
- Clustering
- Based on textual similarity
- Workflow generation
- For a group of similar tests
Step Refactoring

- Close steps are unified
  - Mispelling, Case sensitivity, etc
  - Steps can be parameterized
  - Automatic detection for unification

→ Maintenance is made easier
→ Consistent writing allows faster review and manual execution
→ Less keywords to be defined for automation
Clustering

- Grouping tests by similarity helps maintenance and redundancy detection.
Workflow generation

• Immediate display of a group of scenarios in a single workflow
  • Check step sequences consistency
  • Check scenarios accuracy
Clustering algorithms

![Cluster Dendrogram]

AP-Connection/Disconnection → Connect Errors

- TC_234 - Connection Error - CarRadio off, no supported ...
- TC_1493 - Connection Error - no supported ...
- TC_228 - Connection Error - no files
- TC_229 - Connection Error - unsupported d...
- TC_232 - Connection Error - unable to ident...
- TC_235 - Connection Error - CarRadio off, unsuppor...
A single GUI
RETURN ON EXPERIENCE

2 case studies
Case Study #1

Initial State:
- **302 test scenarios**
- **6212 steps (1444 different steps detected)**

Key figures:
- **Steps unification**
  - The number of different steps has decreased by 257 (21%). (time spent 1 man-day)
- **Steps parametrization**
  - The number of different steps has decreased by 158 (15%). (time spent 1/2 man-day)

Globally:
- **Time reduced to 1.5 man-days (compared to 4 man-days expected)**
Case Study #2

Initial State:
- 942 test scenarios
- 5733 steps (2295 different steps detected)

Key figures:
- 5 hours to refactor and reduce the number of steps (207 steps refactored)
CONCLUSION
Conclusion

• Innovative technology:
  • Clustering and text similarity (syntactic then semantic)
  • Automatic workflow generation helps the testers to manage the refactoring

• Return on experience on multiple test repositories:
  • Assisted Refactoring helps the testers to merge similar steps
  • Parametrization of steps allows to raise the level of abstraction

• To be continued...