Testing, the ultimate Sisiphus rock

By Bernard Stepien
University of Ottawa
Heterogeneity of testing methods

• Manual testing
• General purpose languages based testing
• Formal methods (TTCN-3)
• Model based testing
Manual testing

• Probably the most commonly used
• Based on subjective test plans
• No strong typing
• No guaranty of completeness
General purpose languages based testing

• No abstraction
• Test events are mixed with data retrieval operations
• Test verification is performed at the atomic level
Formal methods (TTCN-3)

• Separation of concerns between
  – Abstract test behavior
  – Concrete level for encoding and decoding messages

• Concept of template which is another separation between test behavior and conditions governing behavior

• Extensive re-usability of templates
Model based testing

• So far all methods involve manual coding of test suites

• A model can be verified before automatically generating a test suite
Software Development Visualization

Cláudia Maria Lima Werner
Federal University of Rio de Janeiro - COPPE/UFRJ
Rio de Janeiro, Brazil
werner@cos.ufrj.br
Data is difficult to manage
How to deal with this?
Awareness

cognitive reactions to a condition/event
(being aware of it)

allows software development stakeholders to be percipient of what goes on in the development scenario
Comprehension

assimilation of knowledge
(understanding a fact)
Visualization support

represent software through metaphors, from a particular point of view, helping stakeholders to focus on the kind of task being performed
General Comprehension/Awareness Challenges

Use software tools to seamlessly collect rich data sets on software comprehension activities

Build specialized, personalized visualizations according to the comprehension needs

Identification and development of suitable mechanisms and adequate abstractions

Strengthen and increase the group of researchers interested in software visualization, awareness, and related areas
Software Development Visualization

Cláudia Maria Lima Werner
Federal University of Rio de Janeiro - COPPE/UFRJ
Rio de Janeiro, Brazil
werner@cos.ufrj.br
Big Data and Machine Learning to Democratize Software Development?
Software/Solution Development

Software Engineering:

- “systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software” (IEEE)
- requirement engineering
- design
- testing
Software/Solution Development

Software Engineering:

- “systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software” (IEEE)
- requirement engineering
- design
- testing

Machine Learning:

- data-driven algorithm creation
  ⇒ function approximation (solution creation)
- can be done almost automatically: AutoML
- data, frameworks and resource access proliferation
Motivation: ML

```python
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
import pandas as pd

lr = RandomForestRegressor(n_estimators=100)
df = pd.read_csv('https://...')
X_train, X_test, y_train, y_test = train_test_split(df['X'], df['y'], test_size=0.33, random_state=42)

lr.fit(X_train, y_train)
pred = lr.predict(X_test)
print(f'Score:{lr.score(y_test, pred)}')
```
Motivation: Web Development

```python
#!/usr/bin/env python
from flask import Flask

app = Flask(__name__)

@app.route('/

def index():
    return 'It works!'

app.run(port=8081)
```
Summary

Democratization:

- ML is dramatically lowering the “Barriers to Entry”
- factors: data, resources, frameworks
- everyone can do it now (not in terms of Software Engineer)

but...
Summary

Democratization:

- ML is dramatically lowering the “Barriers to Entry”
- factors: data, resources, frameworks
- everyone can do it now (not in terms of Software Engineer)

but...ML is high-interest credit on technical debt:

- hard to understand & interpret
- hard to extend & hard to incorporate knowledge
- limited testing capabilities: Unit Tests, Integration Tests,...
Summary

Democratization:

- ML is dramatically lowering the “Barriers to Entry”
- factors: data, resources, frameworks
- everyone can do it now (not in terms of Software Engineer)

but...ML is high-interest credit on technical debt:

- hard to understand & interpret
- hard to extend & hard to incorporate knowledge
- limited testing capabilities: Unit Tests, Integration Tests,...

Democratization...really?

- do you have data?
- do you have GPU cluster?