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 completness

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)

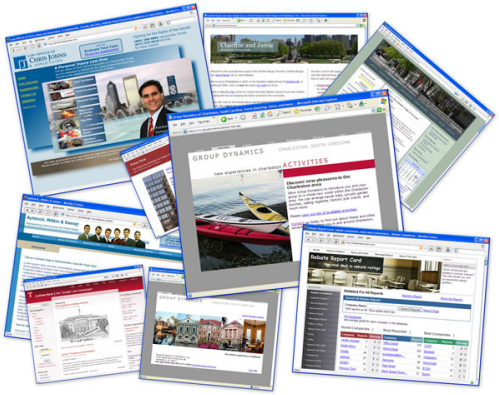
- Separation of concerns between
  - Abstract test behavior
  - Concrete level for encoding and decoding messages
- Concept of template which is another separation between test behavior and conditions governing behavior
- Extensive re-usability of templates

Model based testing

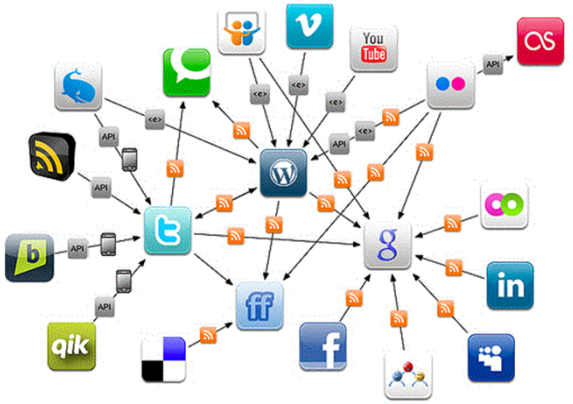
- So far all methods involve manual coding of test suites
- A model can be verified before automatically generating a test suite

Software Development Visualization

Cláudia Maria Lima Werner
Federal University of Rio de Janeiro - COPPE/UFRJ
Rio de Janeiro, Brazil
werner@cos.ufrj.br

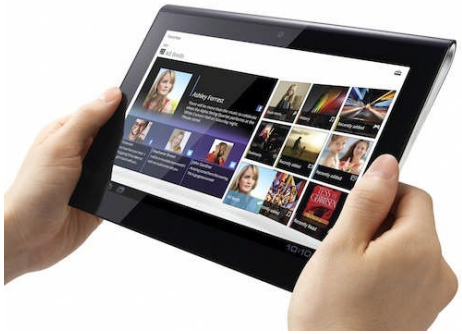



Google
Google Maps
Google Base
Google
Google Patent Search
Google Scholar
Google Code Search
Google
Google Noticias Brasil
Google Finance
Google Trends
Google
Google Video
Google
Google Catalogs
Google
Google Blog Search
Google Groups
Google
Google Answers
Google
Google
Google Image Search
Google Book Search


You Tube
API
qik


SAMSUNG


Cloud
NEWS

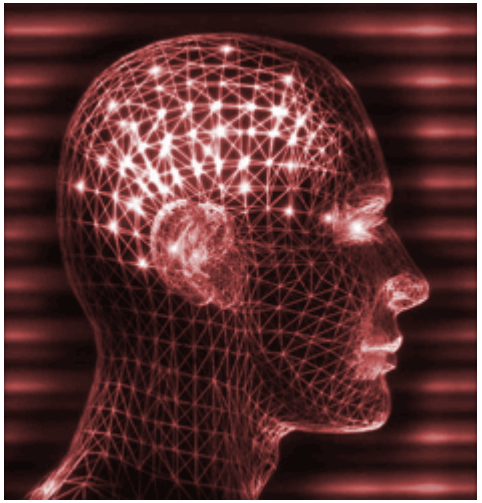


Data is difficult to manage

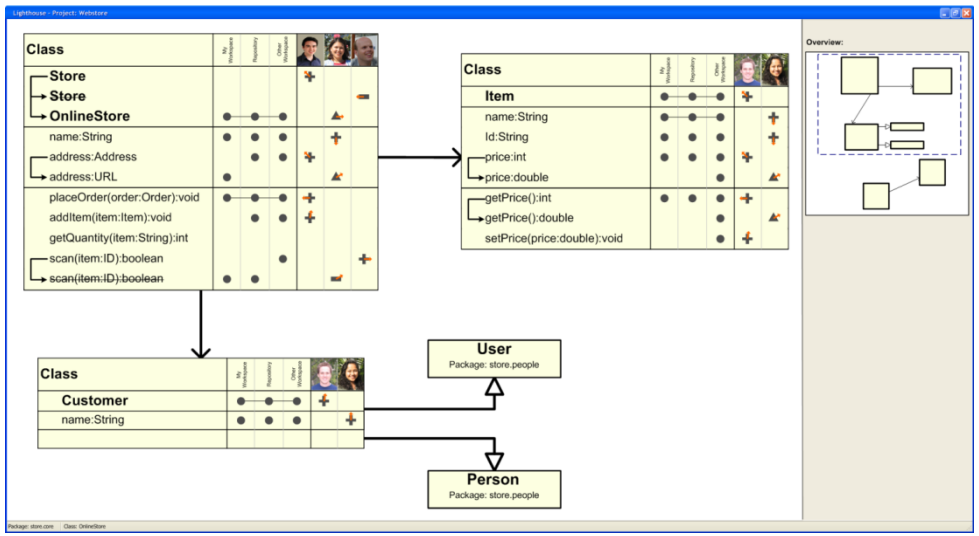

Books/Booklets
Q&A
Manuals
Issue Trackers
Documentation
E-mail Lists
Android Programming Tutorials
Tutorials
Software Development Scenario
Code Repositories & VCSs
Scientific Publications
Discussion Forums
Development Environments


How to
deal with
this?

Awareness



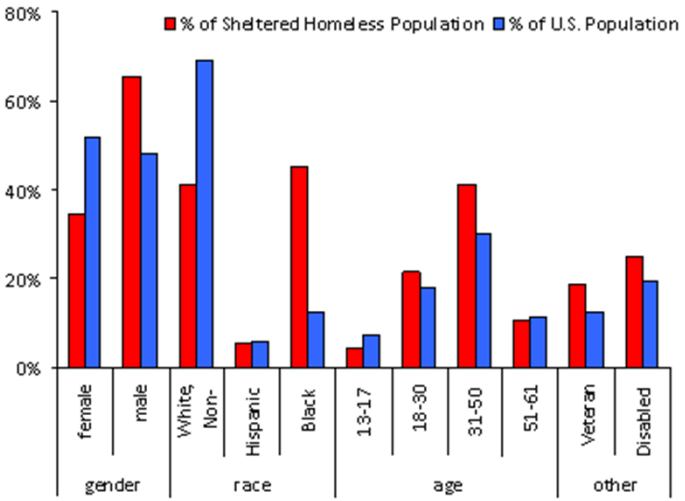
cognitive reactions to a condition/event
(being aware of it)



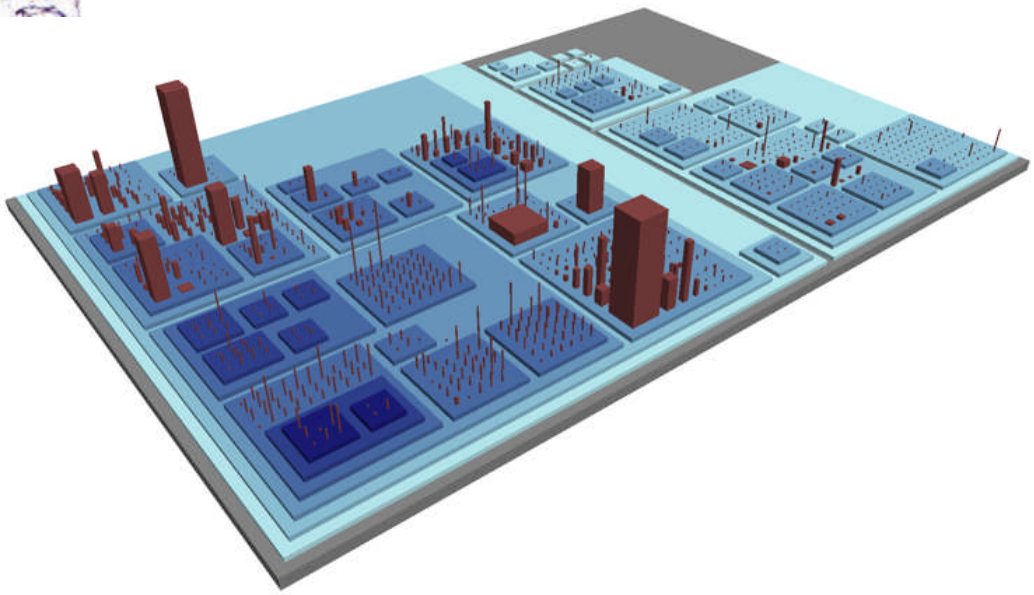
allows software development stakeholders to **be percipient of what goes on** in the development scenario

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

JÜLICH
FORSCHUNGSZENTRUM

Big Data and Machine Learning to Democratize Software Development?

February 25, 2020 | Jędrzej Rybicki

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

$\Rightarrow$ function approximation (solution creation)

- can be done almost automatically: AutoML
- data, frameworks and resource access proliferation

JÜLICH FORSCHUNGSZENTRUM

Motivation: ML

```
1  from sklearn.model_selection import train_test_split
2  from sklearn.ensemble import RandomForestRegressor
3  import pandas as pd
4
5  lr =  RandomForestRegressor(n_estimators=100)
6  df = pd.read_csv('https://...')
7  X_train, X_test, y_train, y_test = train_test_split(
8          df['X'], df['y'], test_size=0.33, random_state=42)
9
10 lr.fit(X_train, y_train)
11
12 pred = lr.predict(X_test)
13 print(f'Score {lr.score(y_test, pred)}')
```

Motivation: Web Development

```
1  #!/usr/bin/env python
2  from flask import Flask
3
4  app = Flask(__name__)
5
6  @app.route('/')
7  def index():
8      return 'It works!'
9
10 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?