

SoCraTes 2022

11th International Conference for Software Craft and
Testing

<https://www.socrates-conference.de/>

8 people from Nordic this year



Source: <https://twitter.com/coderbyheart/status/1563065168422391812>

Unconference

- Main days: Open Space
- <https://www.socrates-conference.de/format>
- No predefined agenda
- ~200 participants
- >100 sessions in 2 days

Full Agenda

<https://scrts.de/socrates2022/>



Source: <https://twitter.com/coderbyheart/status/1563075745697583104>

More pictures

- A view of the crowd (not complete):
<https://twitter.com/coderbyheart/status/1563061865256669189>
- Agenda of day 1:
<https://twitter.com/coderbyheart/status/1563100104474918913>
- Agenda of day 2:
<https://twitter.com/coderbyheart/status/1563443216351133697>

Part of my Agenda

- Legacy code and Approval Testing (with Emily Bache)
- Elixir / Phoenix Intro
- Coupling & Cohesion I: Object Calisthenics
- Enabling my blind grandma to read
- Complain about Pull Requests
- How to bypass your logical brain
- How to design software
- Designing better APIs: One glass at a time
- Security fun: OWASP Juice Shop

Further links:

- Elixir / Phoenix Intro:
<https://github.com/cgrothaus/socrates2022-introduction-to-elixir-phoenix>
- OWASP Juice Shop: <https://owasp.org/www-project-juice-shop/>

Legacy code and Approval Testing

- Emily Bache
- Example: A supermarket receipt, with
 - Items
 - Prices
 - Quantities
 - Discounts
 - Total


```
1      @Test
2      public void tenPercentDiscount() {
3          SupermarketCatalog catalog = new FakeCatalog();
4          Product toothbrush = new Product("toothbrush", ProductU
5          catalog.addProduct(toothbrush, 0.99);
6          Product apples = new Product("apples", ProductUnit.KILO
7          catalog.addProduct(apples, 1.99);
8          Teller teller = new Teller(catalog);
9          teller.addSpecialOffer(
10             SpecialOfferType.TEN_PERCENT_DISCOUNT, toothbrush,
11             ShoppingCart cart = new ShoppingCart();
12             cart.addItemQuantity(apples, 2.5);
13
14             // ACT
15             Receipt receipt = teller.checksOutArticlesFrom(cart);
16
17             // ASSERT
18             assertEquals(4.975, receipt.getTotalPrice(), 0.01);
19             assertEquals(Collections.emptyList(), receipt.getDiscou
20             assertEquals(1, receipt.getItems().size());
21             ReceiptItem receiptItem = receipt.getItems().get(0);
22             assertEquals(apples, receiptItem.getProduct());
23             assertEquals(1.99, receiptItem.getPrice());
24             assertEquals(2.5*1.99, receiptItem.getTotalPrice());
25             assertEquals(2.5, receiptItem.getQuantity());
26     }
```

```

1      @Test
2      public void tenPercentDiscount() {
3          SupermarketCatalog catalog = new FakeCatalog();
4          Product toothbrush = new Product("toothbrush", ProductU
5          catalog.addProduct(toothbrush, 0.99);
6          Product apples = new Product("apples", ProductUnit.KILO
7          catalog.addProduct(apples, 1.99);
8          Teller teller = new Teller(catalog);
9          teller.addSpecialOffer(
10             SpecialOfferType.TEN_PERCENT_DISCOUNT, toothbrush,
11             ShoppingCart cart = new ShoppingCart();
12             cart.addItemQuantity(apples, 2.5);
13             // ACT
14             Receipt receipt = teller.checksOutArticlesFrom(cart);
15
16             // ASSERT
17             String printedReceipt = new ReceiptPrinter().printRecei
18             Approvals.verify(printedReceipt);
19     }

```

- On first run: Test fails
- Creates file `supermarketTest.tenPercentDiscount.received.txt`

```

apples          4.97
  1.99 * 2.500
Total:          4.97

```

- Check the text manually once for correctness
- Then rename file to
`SupermarketTest.tenPercentDiscount.approved.txt`
- On future runs:
 - Succeeds if output the same
 - Fails if deviating
 - On failure the text files are easily diffable
- 🤔 Snapshot tests in Jest

- Contrary to typical unit tests
 - Not only a single aspect but whole scenario
 - Typical unit may be a bit bigger

- Idea

- Use a few approval tests for the most important happy paths of related chunks
- Use more unit tests for special cases

- Similar presentation by Emily:
<https://www.youtube.com/watch?v=0ZVKcFsEp-4>
- Tools: <https://github.com/approvals>
- Emily's example:
<https://github.com/emilybache/Hands-On-Approval-Testing-For-Developers-Materials>
- On the wording 'Golden Master', 'Snapshot', 'Characterization' and 'Approval':
<https://proagile.se/blog/say-approval-testing-instead-of-golden-master>

asdf

<https://asdf-vm.com>

Multi tool version manager

```
File .tool-versions: ruby 2.5.3  
nodejs 10.15.0
```

Plugins for *many* languages and tools, e.g.

- Node
- Ruby
- Elixir
- Java
- Rust
- CMake
- Maven
- Just
- Ninja

<https://github.com/asdf-vm/asdf-plugins>

Recommendation

Visit an Unconference, at least once!

e.g. SoCraTes or it's partner conference

- SoCraTes Canaries
- SoCraTes UK
- CodeFreeze (January, Northern Finland)

<https://www.socrates-conference.de/home#conferences>