Legacy Code Retreat – Softwerkskammer Stuttgart / JUGS

Agenda

- Kurzvorstellung der Referenten
- Was ist Legacy Code
- Mögliche Wege zur Wiederbelebung
- Workshop Trivia Legacy Code
- Diskussion come together

Agenda

- Kurzvorstellung der Referenten

- Was ist Legacy Code

- Mögliche Wege zur Wiederbelebung

- Workshop Trivia Legacy Code

- Diskussion come together

Oliver Böhm

- SW-Arch. beim Optica Abrechnungszentrum
- Board-Mitglied **Java User Group Stuttgart**
- Buch-Autor

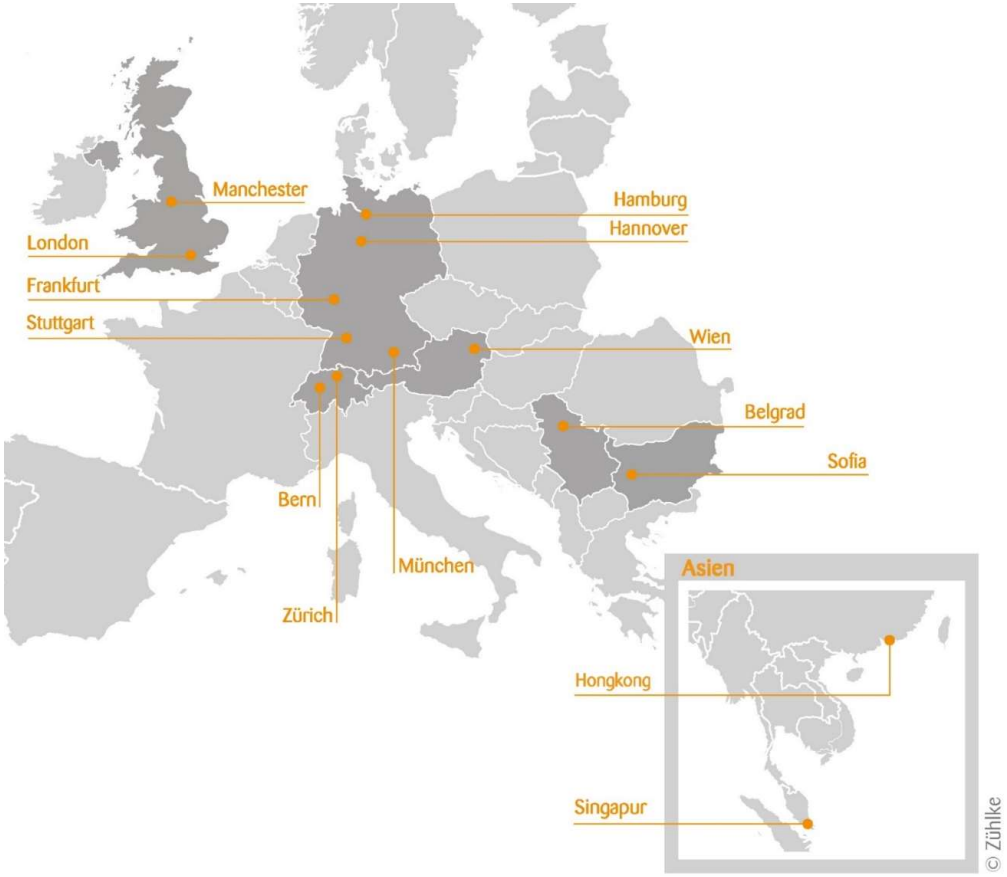
SIG AspectJ

Daniel Georges

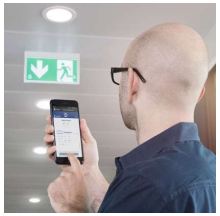
- Head of Competence Unit bei der Zühlke Engineering GmbH in Stuttgart
- Ansprechpartner Softwerkskammer Stuttgart
- 20 Jahre Erfahrung u.a. als
  - Softwareentwickler
  - Softwarearchitekt
  - Projektmanager / Product Owner
  - Bereichsleiter
- Fokus auf Beratung
  - Agile Coach
  - Softwarecraftsmanship

Über Zühlke

Facts & Figures



- Gegründet 1968
- Im Besitz von Partnern
- Teams in Deutschland, Großbritannien, Österreich, Serbien, Sofia, Hongkonk, Singapur und der Schweiz
- Mehr als 8'000 Projekte realisiert
- 1000 Mitarbeiterinnen & Mitarbeiter

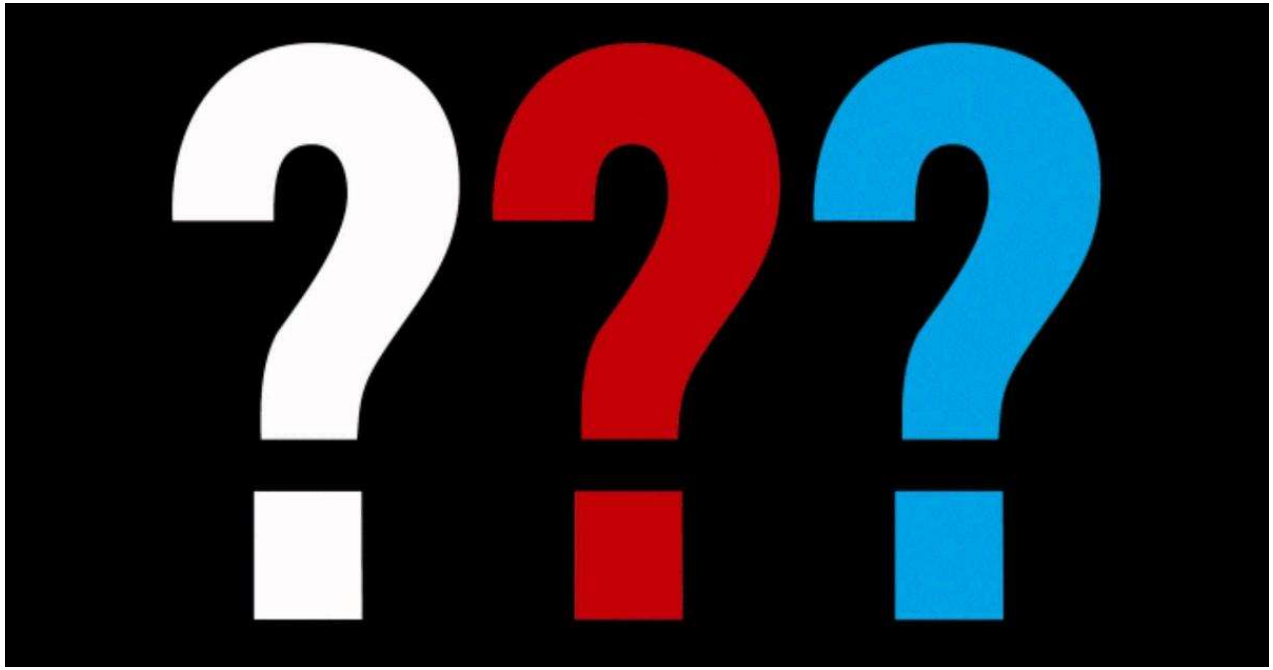


zühlke
empowering ideas

Agenda

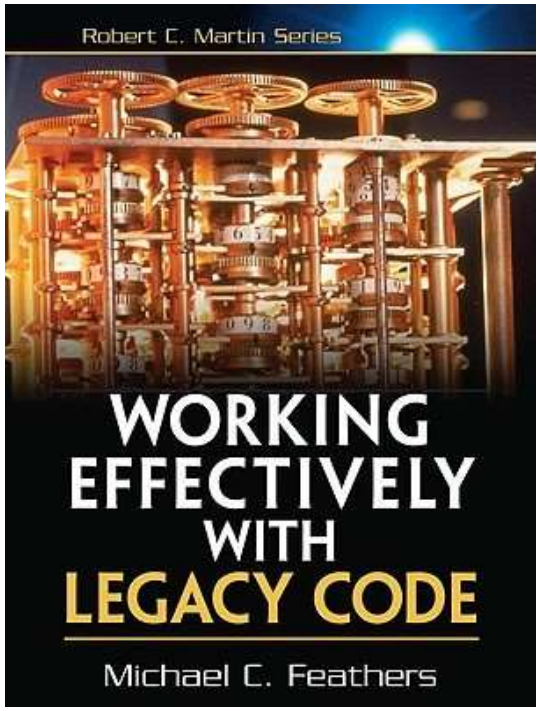
- Kurzvorstellung der Referenten
- Was ist Legacy Code
- Mögliche Wege zur Wiederbelebung
- Workshop Trivia Legacy Code
- Diskussion come together

Definition von Legacy Code?



Quelle:http://bento.de

Definition von Legacy Code?



Definition von Legacy Code?

«Code without tests is bad code. It doesn't matter how well written it is; it doesn't matter how pretty or object-oriented or well-encapsulated it is. With tests, we can change the behavior of our code quickly and verifiably. Without them, we really don't know if our code is getting better or worse.» *Michael Feathers*

Warum sind Tests notwendig?

1. *To improve some piece of code, we must be able to refactor it*
2. *To be able to refactor code, we must have tests that prove our refactoring didn't break anything*
3. *To have reasonable tests, the code has to be testable…*

Warum überhaupt was ändern?

1. *Adding a feature (behavior)*
2. *Fixing a bug (behavior)*
3. *Improving the design*
4. *Optimizing resource usage (internal behavior)*

2 Möglichkeiten

1. *Edit and Pray*
2. *Cover and Modify*

Agenda

- Kurzvorstellung der Referenten
- Was ist Legacy Code
- Mögliche Wege zur Wiederbelebung
- Workshop Trivia Legacy Code
- Diskussion come together

Einleitung der Wiederbelebung

1. *Cover code with tests -> dilemma*
2. *Ensure behavior with Golden Master Technique*
3. *Refactoring using small steps (boy scout)*
4. *Use Software Craftsmanship Techniques*

Agenda

- Kurzvorstellung der Referenten
- Was ist Legacy Code
- Mögliche Wege zur Wiederbelebung
- Workshop Trivia Legacy Code
- Diskussion come together

Workshop - https://github.com/jbrains/trivia

1. *Pairprogramming*
2. *Write unit tests*
3. *Use Golden Master Technique if possible*
4. *5 x changes after 15 mins – 5 mins discussions*

Agenda

- Kurzvorstellung der Referenten
- Was ist Legacy Code
- Mögliche Wege zur Wiederbelebung
- Workshop Trivia Legacy Code
- Diskussion come together

Resumée - Diskussion

1. *What did I learn today?*
2. *What will I use tomorrow?*
3. *Peer group discussion*

Come together

JUGS

Softwerkskammer
Software Craftsmanship Communities