

Test-Driven Development

Learn the "Test First" Approach to Coding



SoftUni Team
Technical Trainers



SoftUni



Software University

<https://softuni.bg>

sli.do

#java-advanced

1. **Code** and **Test**
2. **Test-Driven** Development
3. **Reasons to Use** Test-Driven Development

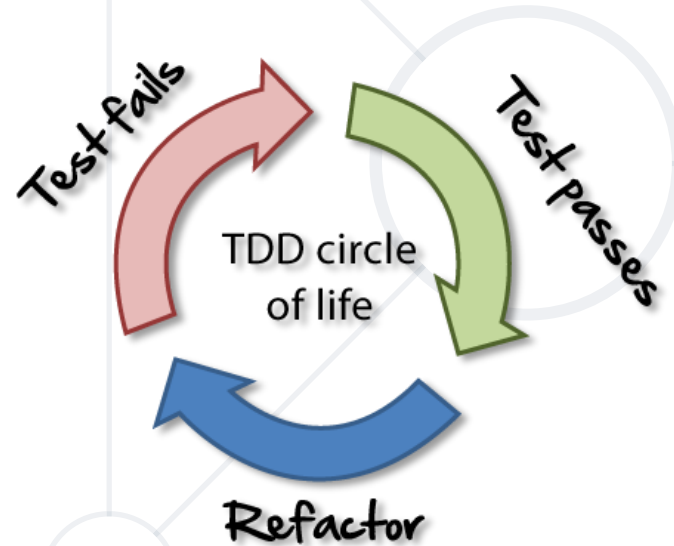




Test-Driven Development

Unit Testing Approaches

- "Code First" (code and test) approach
 - Classical approach
- "Test First" approach
 - Test-driven development (TDD)



The Code and Test Approach

Write code

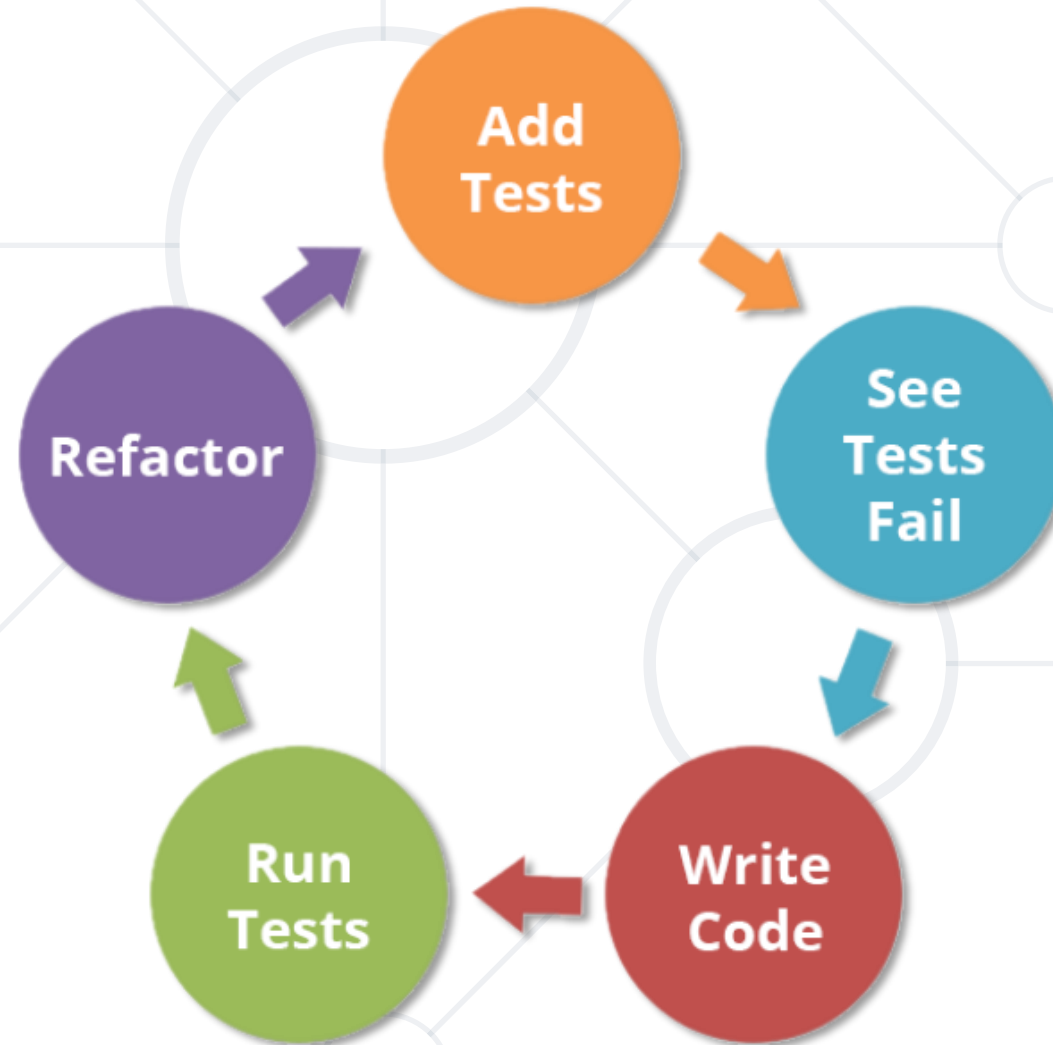
Write unit test

Run and succeed

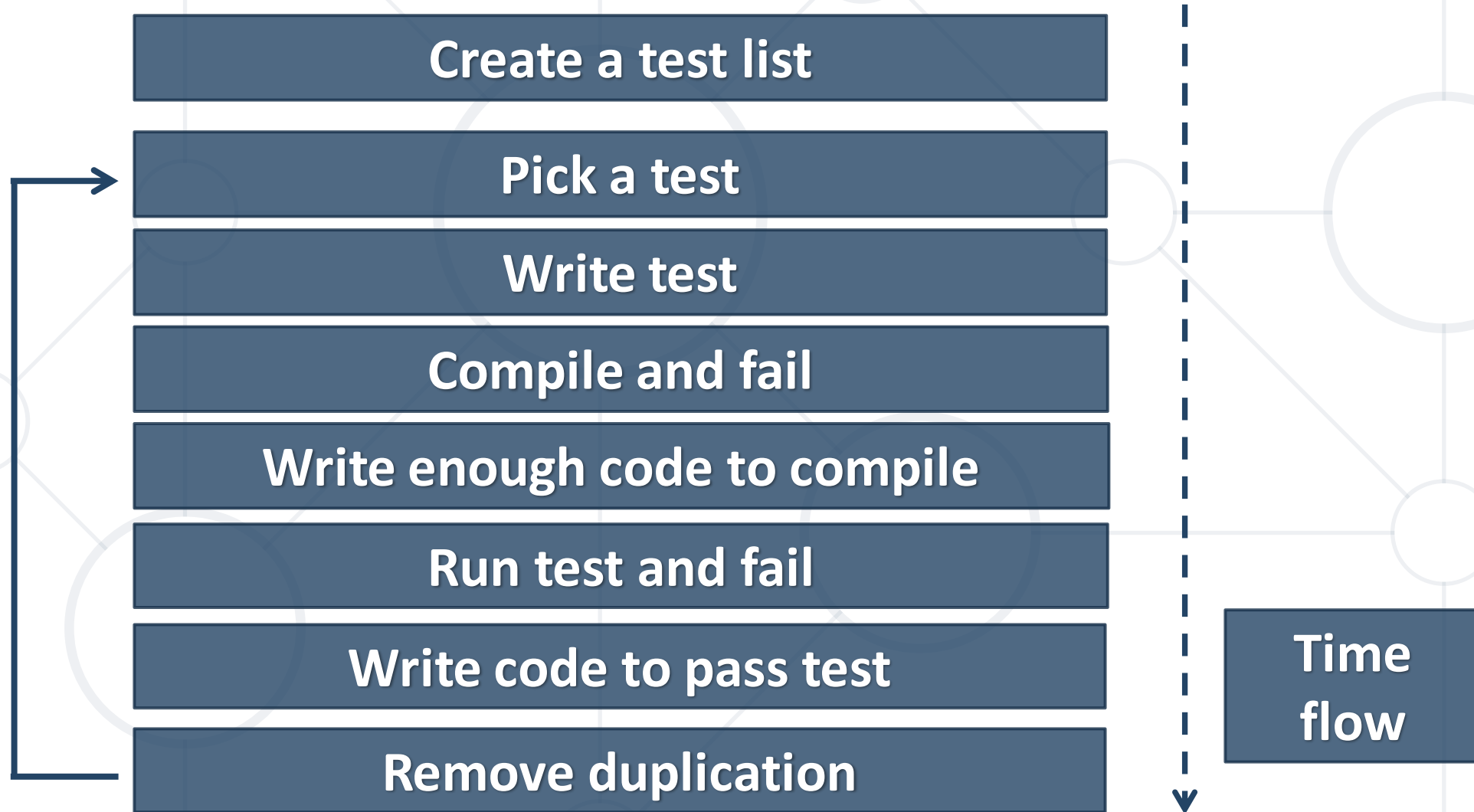


Time flow

The Test-Driven Development Approach



Test-Driven Development (TDD)



Why TDD?

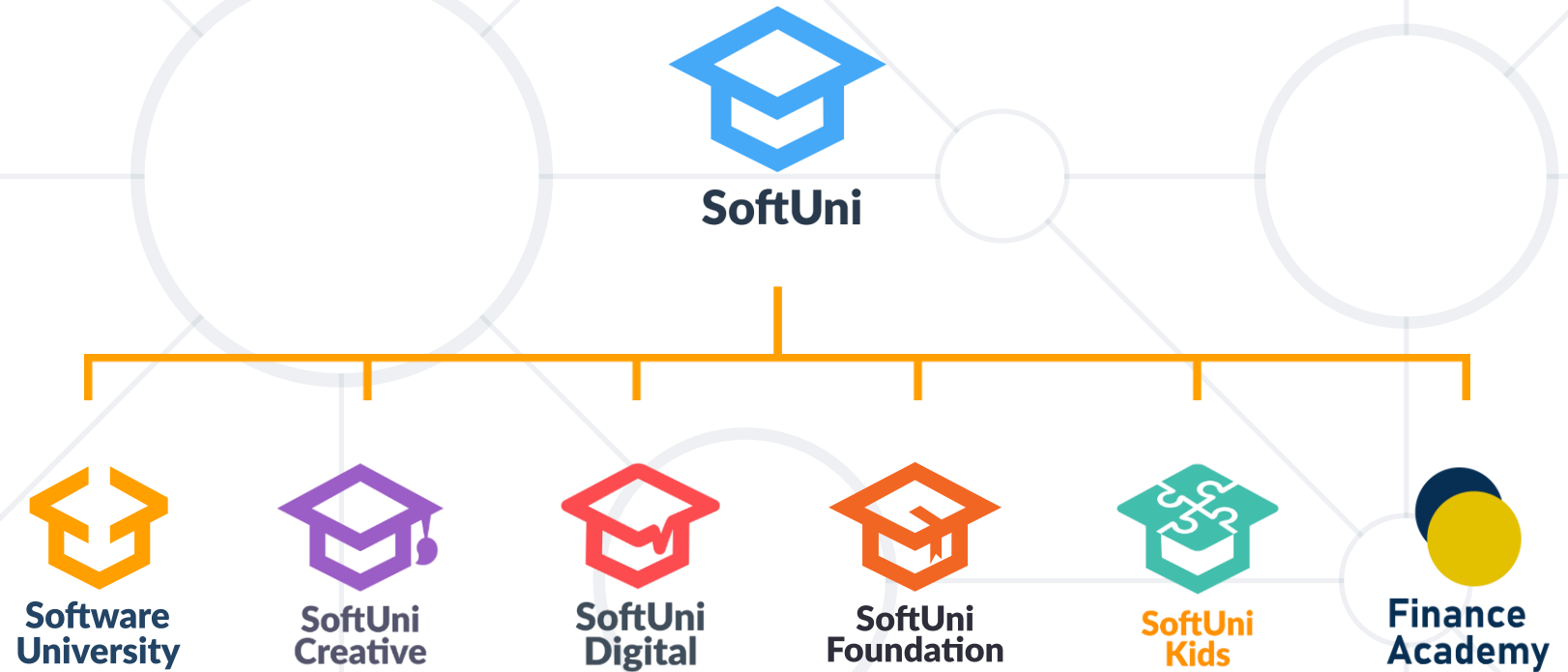
- TDD helps find design issues early
 - Avoids reworking
- Writing code to satisfy a test is a focused activity
 - Less chance of error
- Tests will be more comprehensive than if they are written after the code



- **Code and Test**
 - **Write code**, then test it
- **Test-Driven Development**
 - **Write tests** first
- Reasons to use **TDD**
 - **Prevent** some application design flaws
 - **Manage complexity** more easily



Questions?



SoftUni Diamond Partners



- Software University – High-Quality Education, Profession and Job for Software Developers

- softuni.bg

- Software University Foundation

- softuni.foundation

- Software University @ Facebook

- facebook.com/SoftwareUniversity



- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – <https://about.softuni.bg>
- © Software University – <https://softuni.bg>

