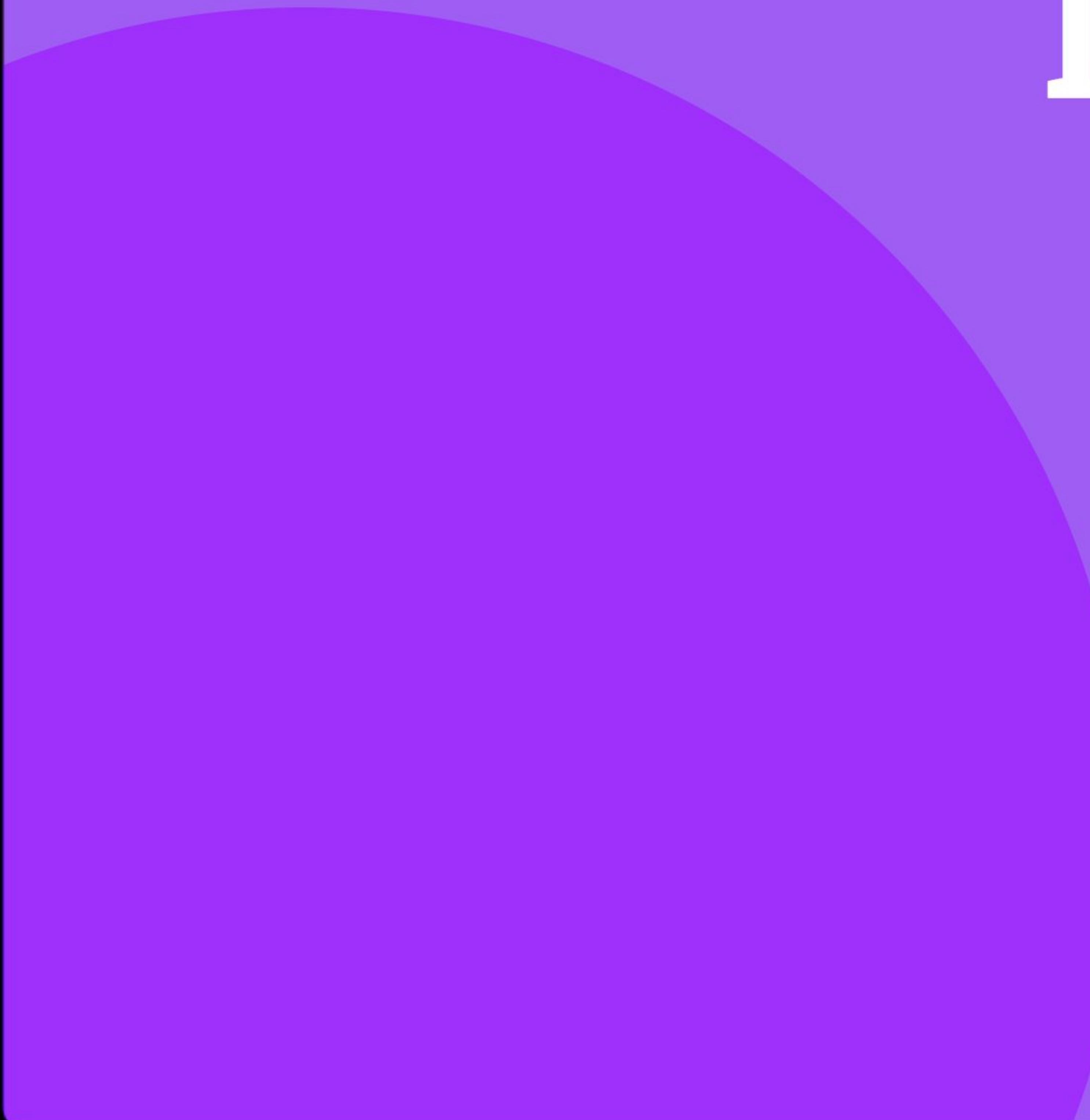


The background features a minimalist design with a white rectangular frame on the right side. The left side is filled with a light gray gradient. Overlaid on this are several large, semi-transparent circles in shades of purple and blue. Scattered throughout the composition are small, solid yellow dots of varying sizes.

TDD



What is TDD?



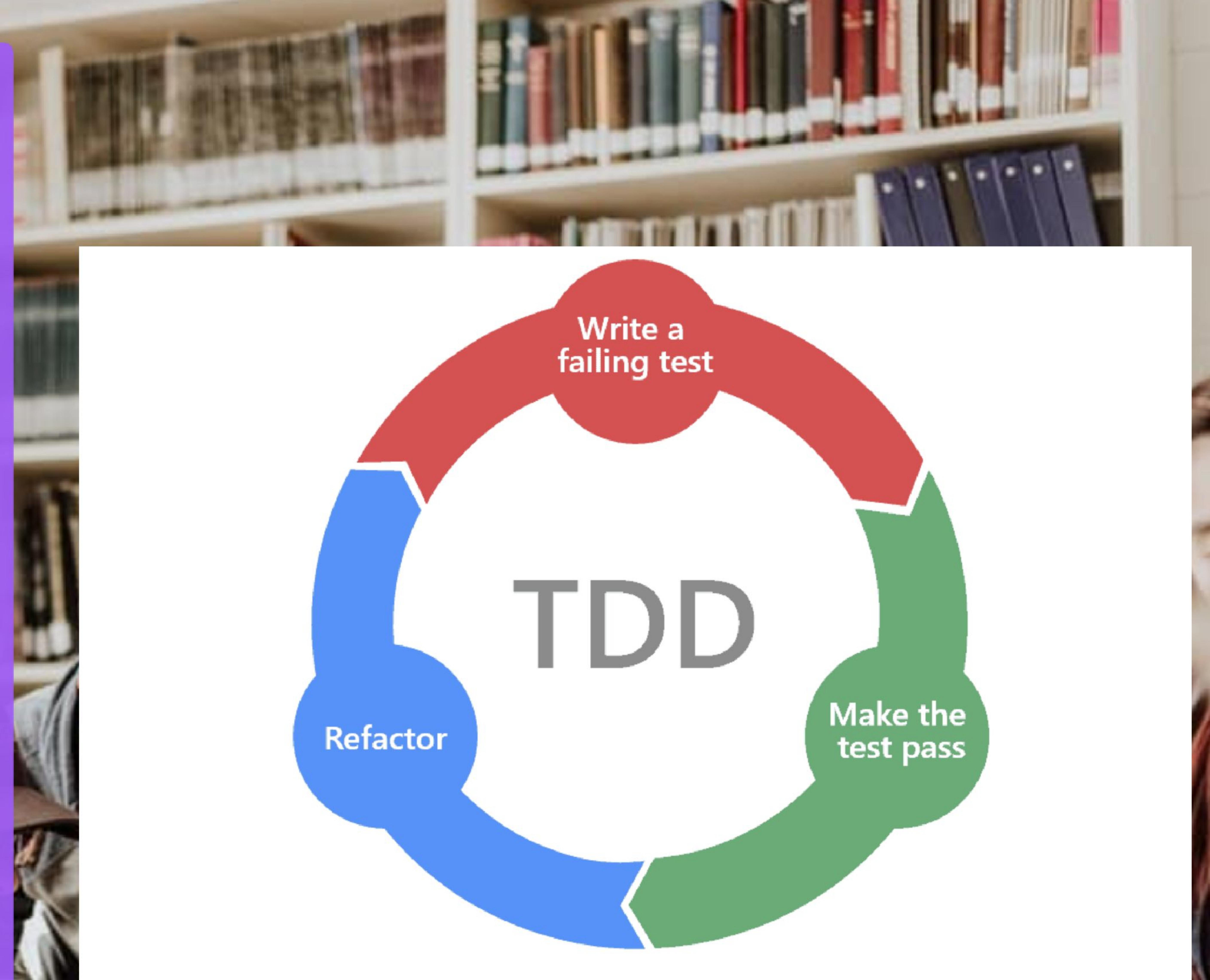
It is not just unit
testing....





TDD is...

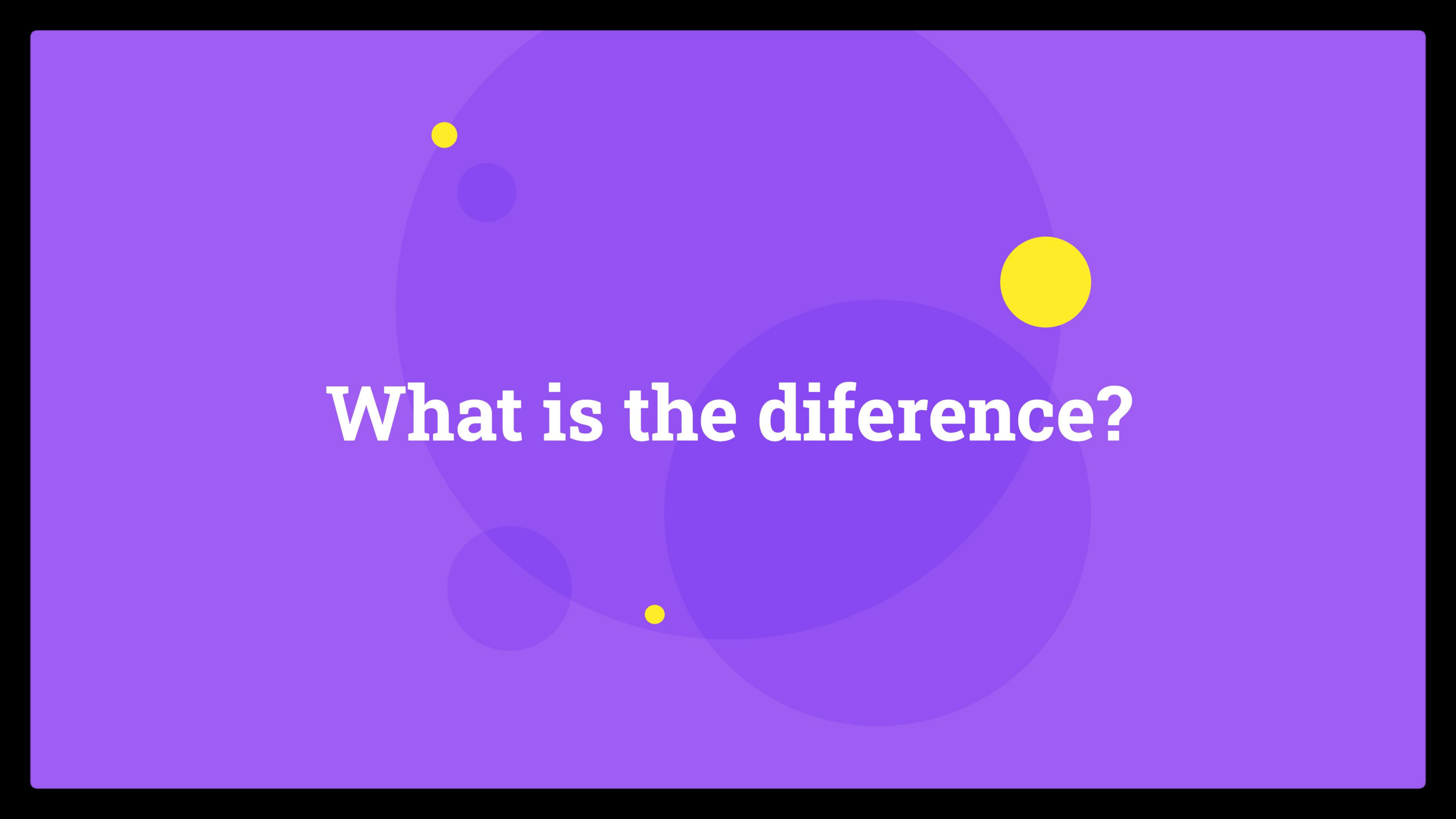
Test-driven development (TDD) is a development technique where you must first write a test that fails before you write new functional code.





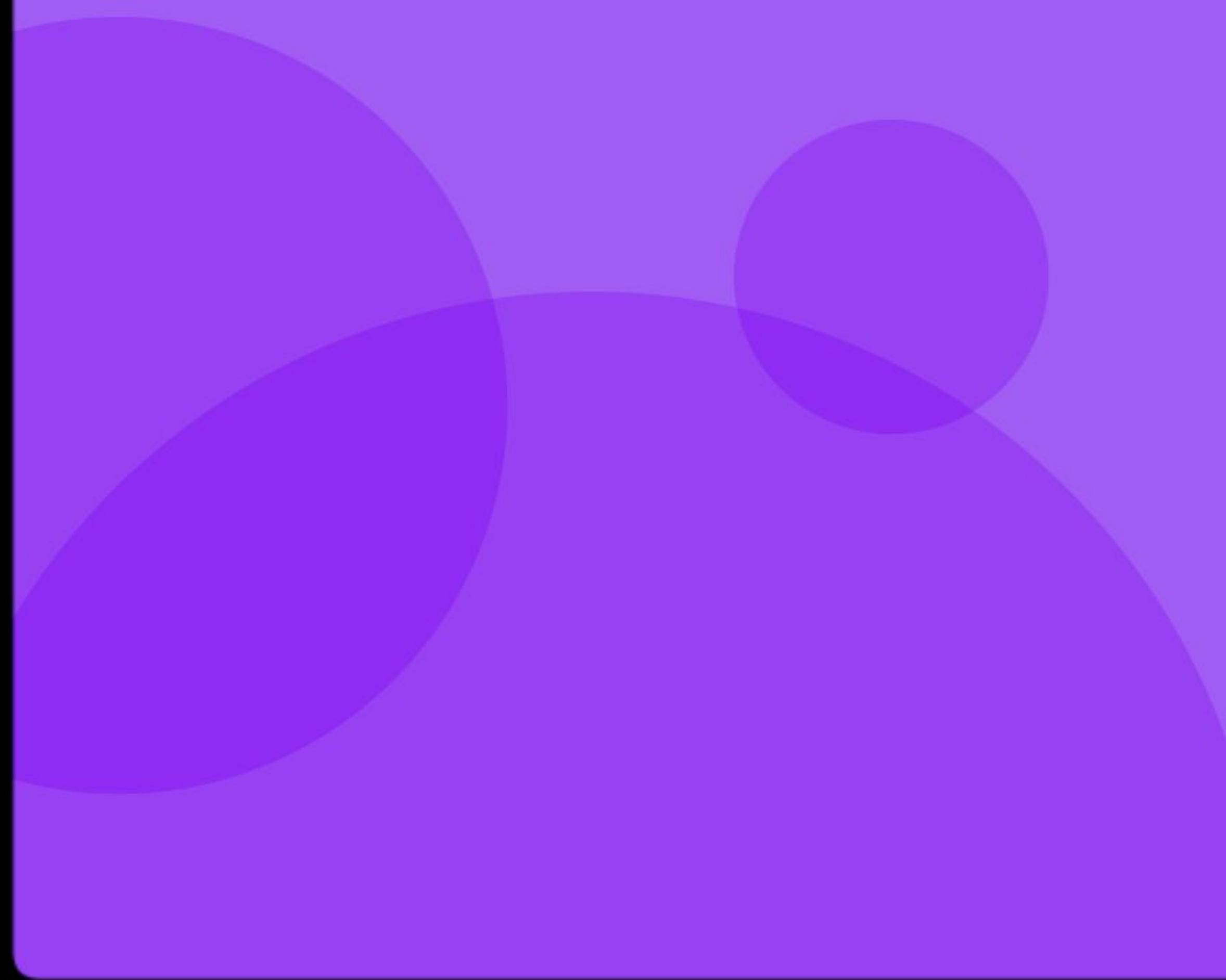
Key points:

- Doing Unit testing is not the same thing as doing TDD
- TDD is taking unit testing a step further
- It is a complete inversion of the unit testing process

The background features a minimalist design with a white rectangular frame on a black background. Inside, there are three overlapping circles: a large light purple circle at the top, a medium-sized light blue circle at the bottom left, and a small light green circle at the bottom right. Each circle contains a single yellow dot of the same color as the circle.

What is the difference?

Unit Testing



Unit Testing

Step 1: Write Logic

Unit Testing

Step 1: Write Logic

Step 1: write logic

```
double calculateLoan (int amount,  
int months, float interestRate)  
{  
    // ...assume functionality  
    // ...  
}
```

Unit Testing

Step 1: Write Logic



Step 2: Write tests

Step 1: write logic

```
double calculateLoan (int amount,  
int months, float interestRate)  
{  
    // ...assume functionality  
    // ...  
}
```

Unit Testing

Step 1: Write Logic



Step 2: Write tests

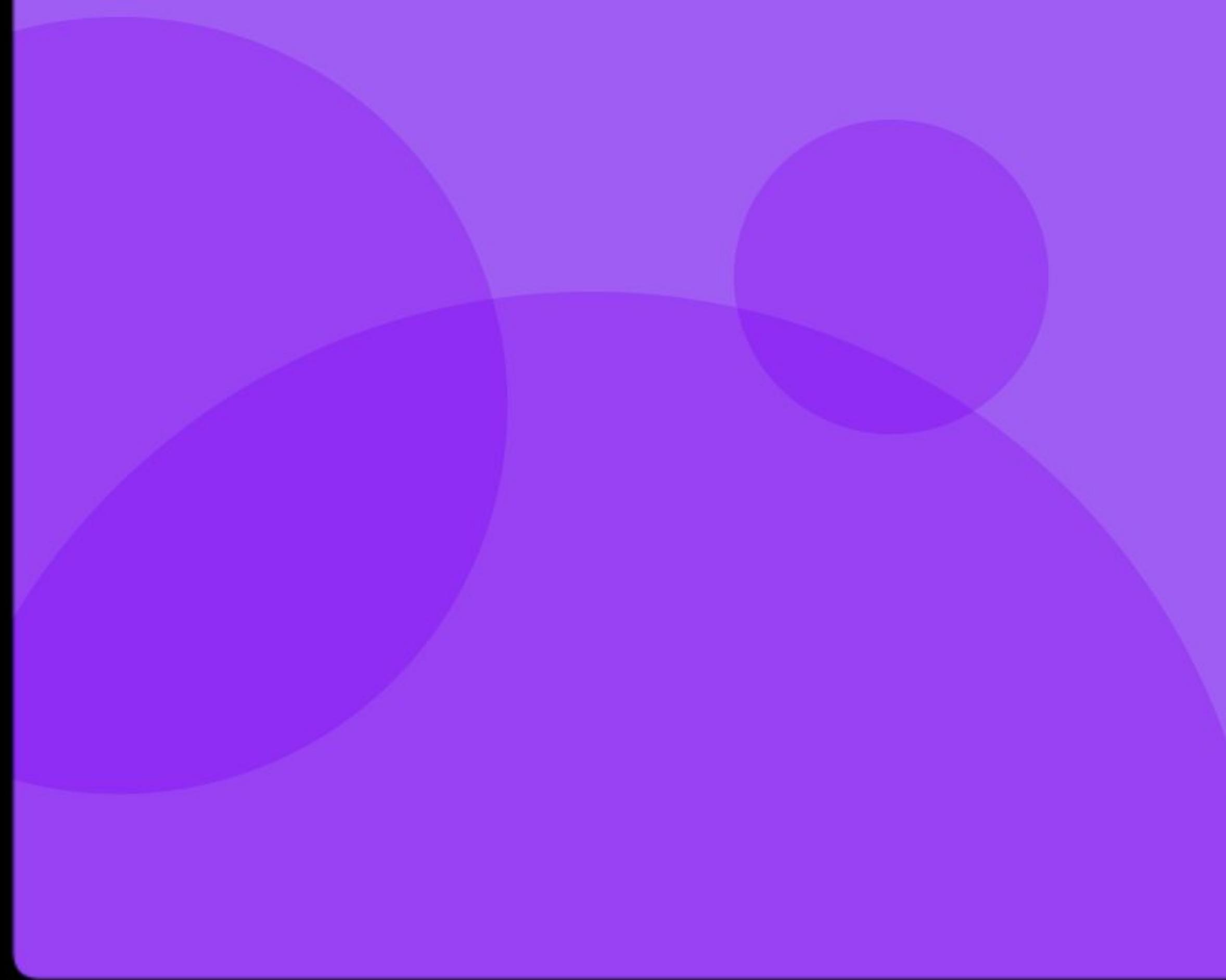
Step 1: write logic

```
double calculateLoan (int amount,  
int months, float interestRate)  
{  
    // ...assume functionality  
    // ...  
}
```

Step 2: write test(s)

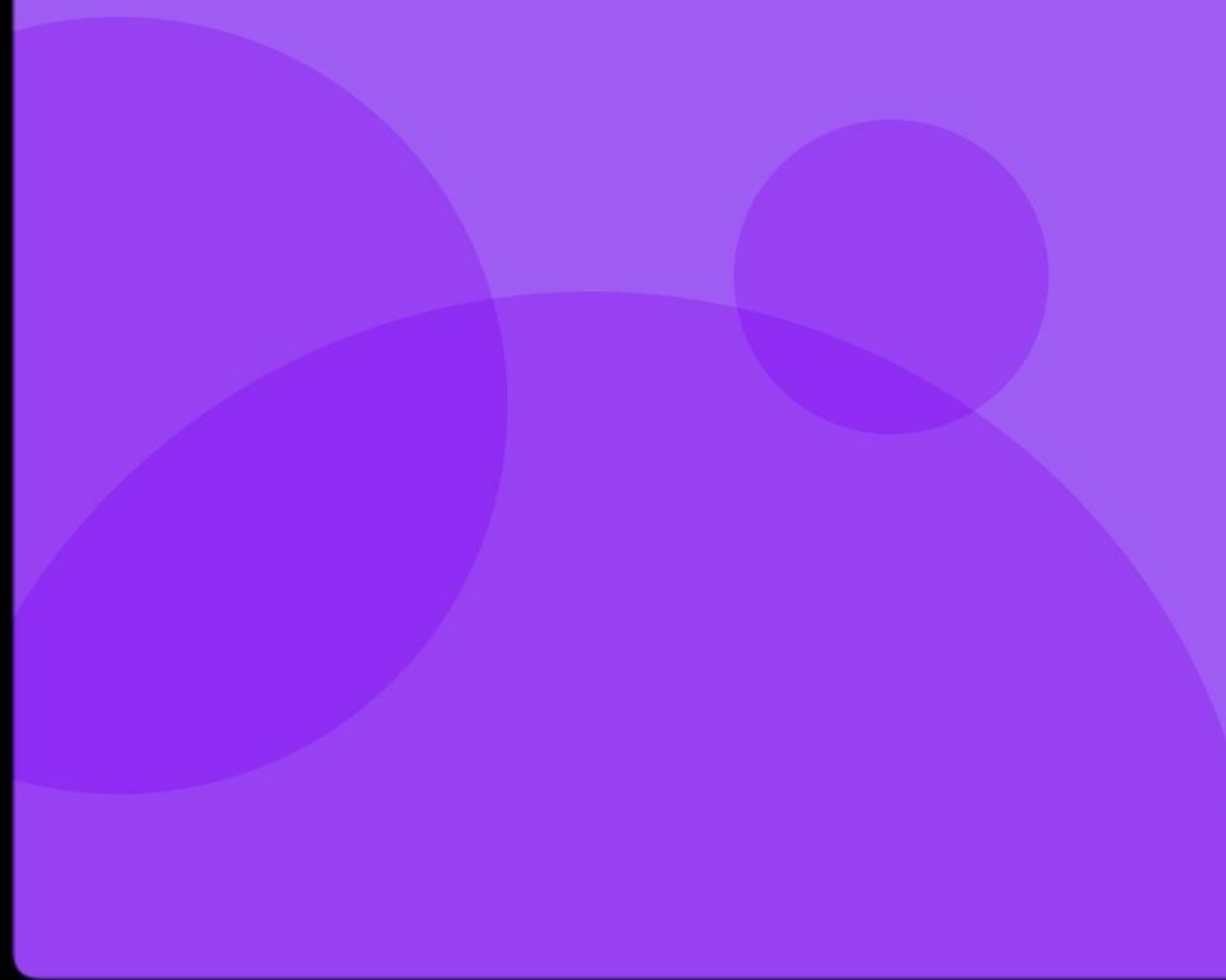
```
calculateLoan(20000,60,5.0);  
// is result 382.02?  
  
calculateLoan(6000,12,10.0);  
// is result 527.50?
```

Test Driven Development



Test Driven Development

Step 1: Write Tests



Test Driven Development

Step 1: Write Tests

Step 1: write test(s) first

```
calculateLoan(20000,60,5.0);  
// is result 382.02?
```

```
calculateLoan(6000,12,10.0);  
// is result 527.50?
```

Test Driven Development

Step 1: Write Tests



Step 2: Write Logic

Step 1: write test(s) first

```
calculateLoan(20000,60,5.0);  
// is result 382.02?
```

```
calculateLoan(6000,12,10.0);  
// is result 527.50?
```

Test Driven Development

Step 1: Write Tests



Step 2: Write Logic

Step 1: write test(s) first

```
calculateLoan(20000,60,5.0);  
// is result 382.02?  
  
calculateLoan(6000,12,10.0);  
// is result 527.50?
```

Step 2: write logic

```
double calculateLoan (int amount,  
int months, float interestRate)  
{  
    // ...functionality  
    // ...  
}
```

“

**What TDD asks of us as developers is
to write the tests first before we write
the application logic**

”

Simon Allardice

Advantages

1/

Focus

TDD helps you focus on development with a simple objective, PASS THE TEST!

2/

Code improvement

Writing tests first require you to really consider what you want from the code.

3/

Simplification

Forces radical simplification of the code – you will only write code in response to the requirements of the tests.

Disadvantages

1/

Time consuming

It necessitates a lot of time and effort up front, which can make development feel slow to begin with.

2/

Major refactoring

Focusing on the simplest design now and not thinking ahead can mean major refactoring requirements.

3/

Design changes

If the design is changing rapidly, you'll need to keep changing your tests. You could end up wasting a lot of time writing tests for features that are quickly dropped.

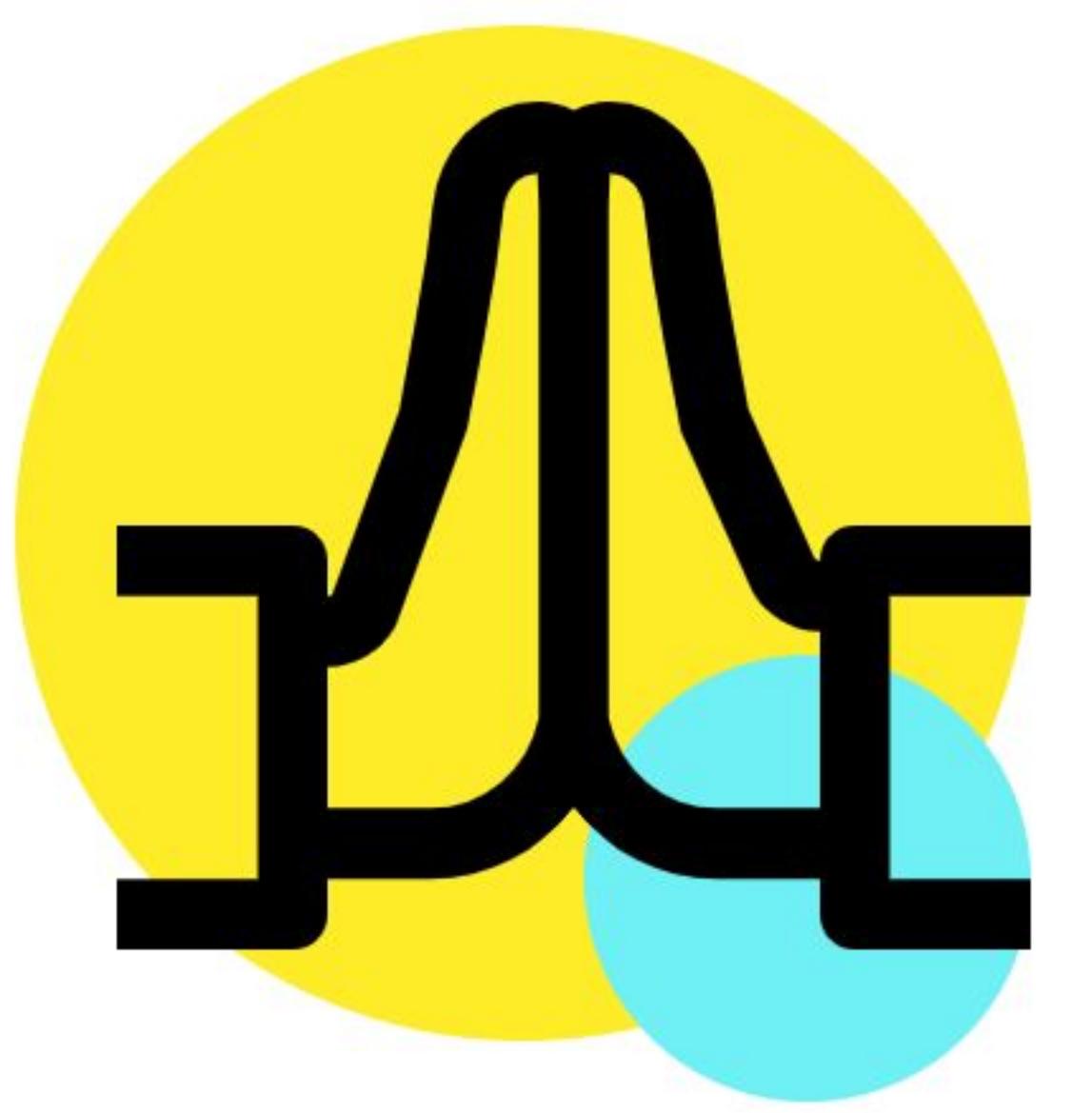
Conclusion:

Before choosing to use TDD in an existing project, you have to consider the unique context of your project in detail. Ask yourself these questions and then decide:

- **How much will TDD increase the cost of implementing your functionality?**
- **What kind and how many automated tests do you plan to create?**
- **Will TDD be used correctly? Will TDD simplify your production code or make it more complex**

In this post we've discussed how TDD is becoming more and more widespread for the following reasons:

- **reducing the number of bugs in production**
- **improving code quality**
- **provides you with the automated tests for regression testing**



Thank you!