

Test-driven Development: The Big Picture

What Is Test-driven Development?



Jason Olson
Staff Software Engineer

@jolson88 www.jolson88.com

Overview/ Summary



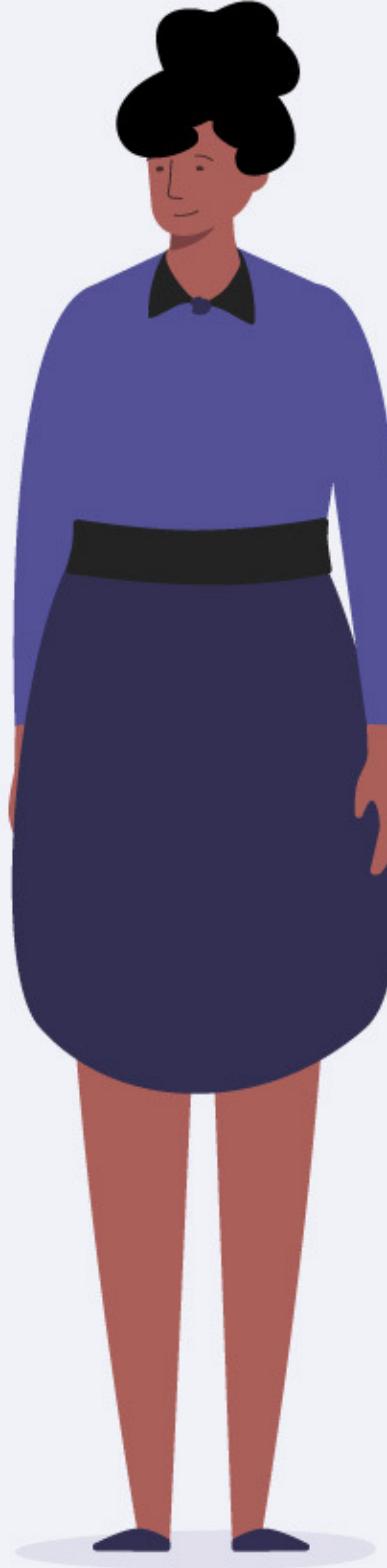
What is test-driven development?

Why does it exist?

How does it work?

Tips, tricks, and gotchas





**No TDD experience
needed!**

**Some experience
with code**

Don't worry!



Thoughts? Comments? Questions?

[Table of contents](#)

[Description](#)

[Transcript](#)

[Exercise files](#)

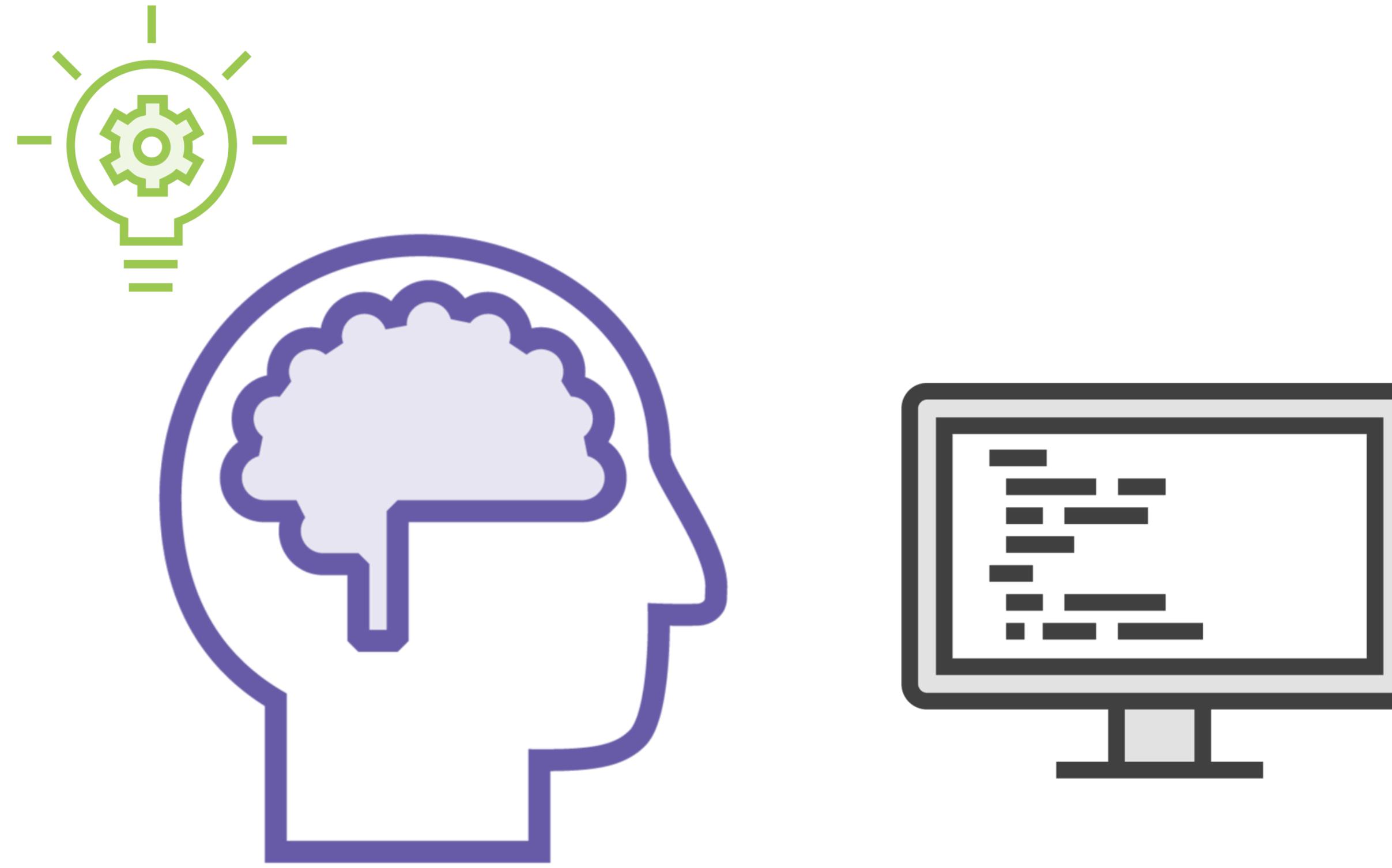
Discussion

[Learning Check](#)

@jolson88

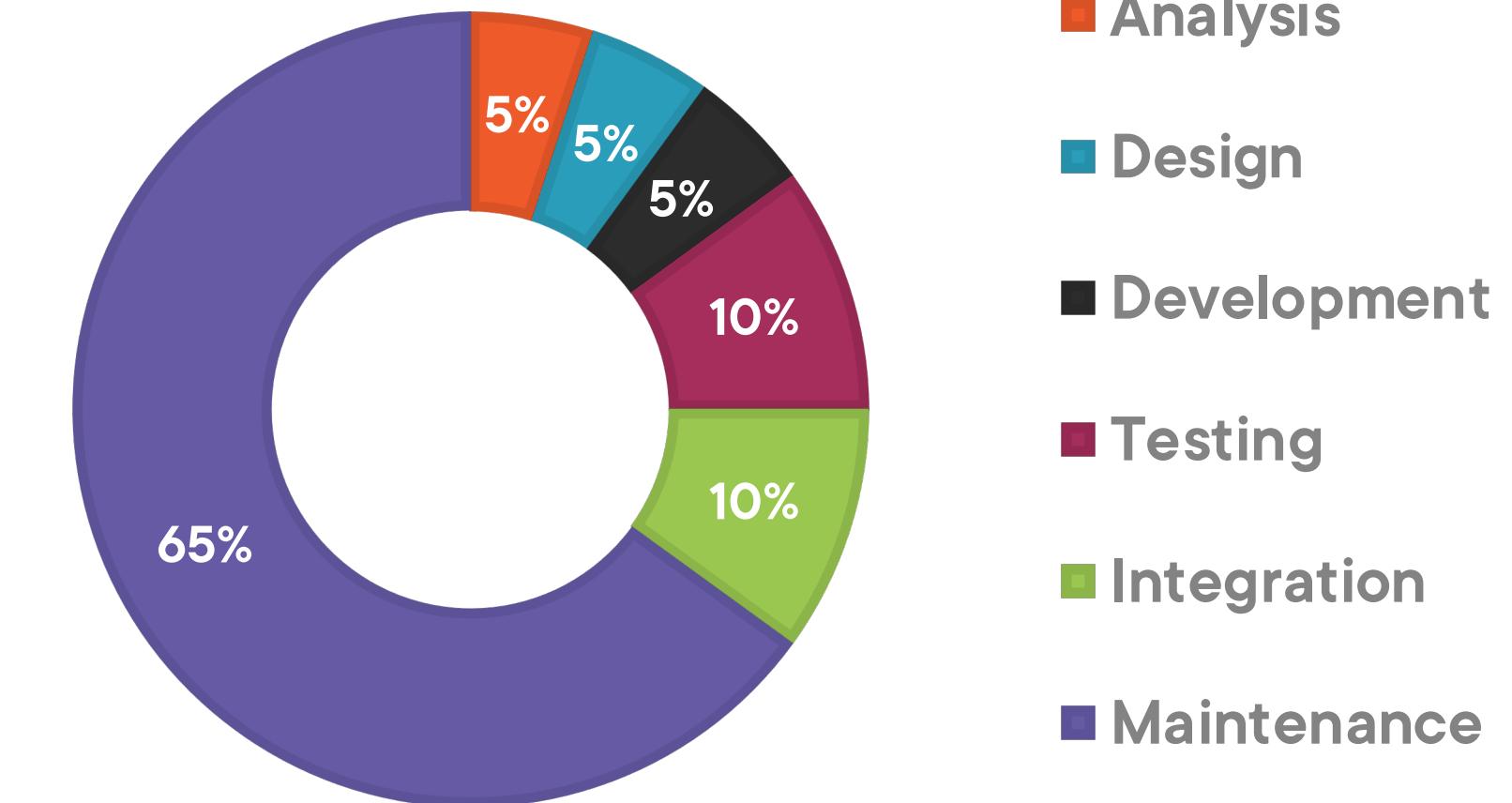
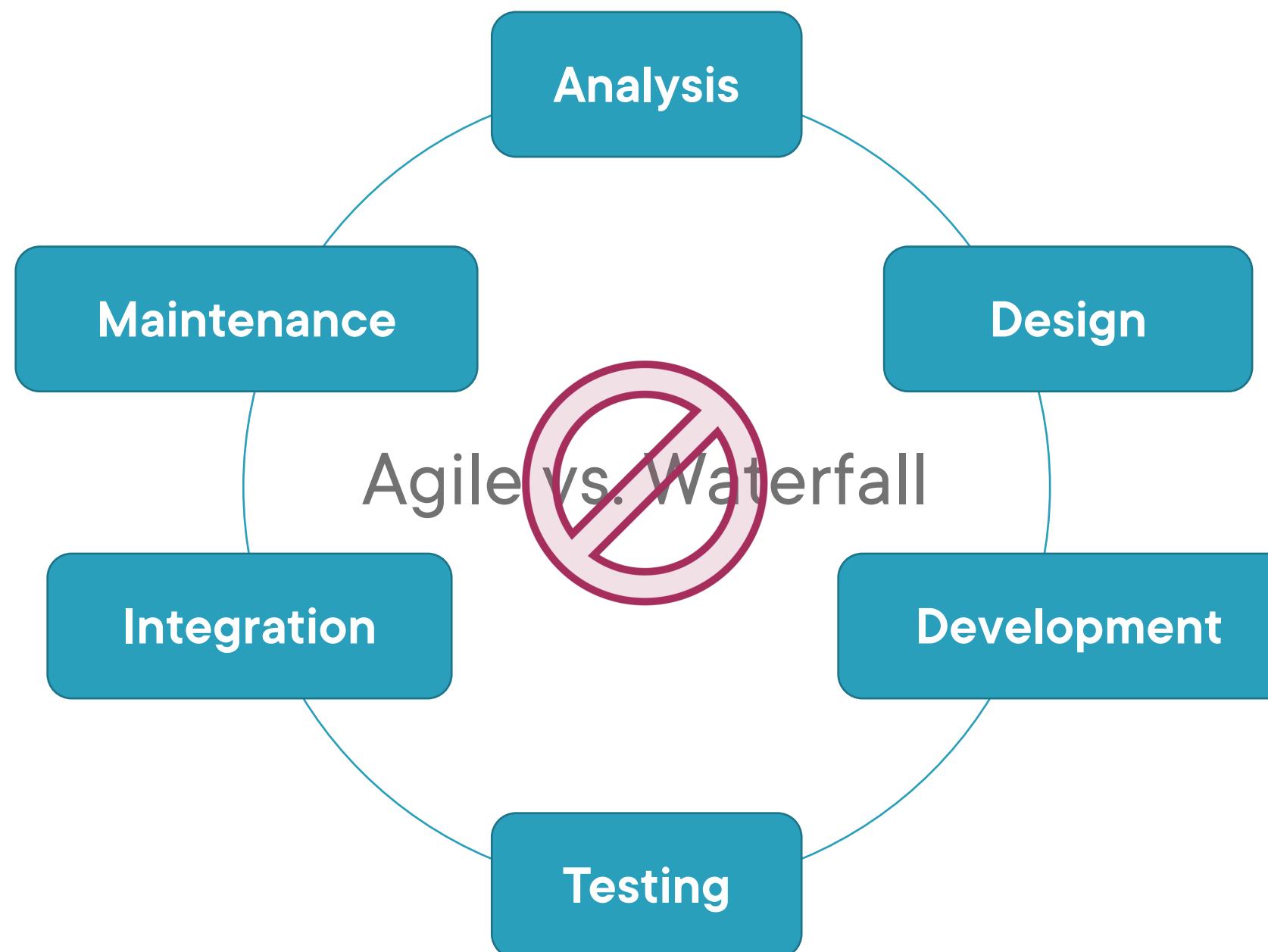


Why Test-driven Development?



Software Development Challenges

Costs of Developing Software



**Maintenance accounts for 65%
of all software development costs!**

Maintenance Challenges



Code Entropy

- *Brittle over time*
- *Increased rigidity*



Isolated Ownership

- *Coder-owned silos*
- *Lack of team empowerment*



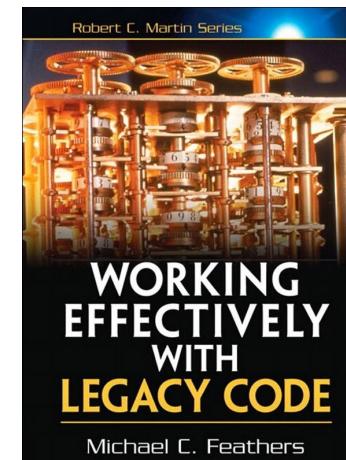
Infrequent Validation

- *Regression prone*
- *High risk of code changes*



Legacy Code

- 1) “source code inherited from someone else”
- 2) “source code inherited from an older version of the software”
- 3) “code without tests”



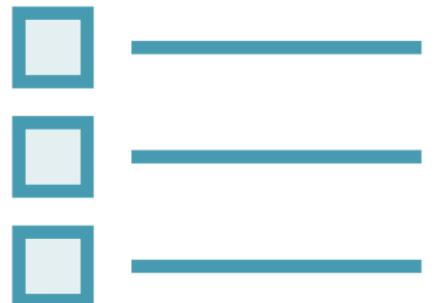
What Is Test-driven Development (TDD)?

Test

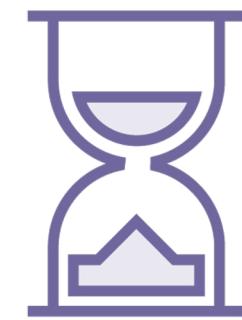
“A procedure intended to establish the quality, performance, or reliability of something, especially before it is taken into widespread use.”



**Satisfies
requirements**



**Responds correctly
to all input**



**Acceptable
performance**



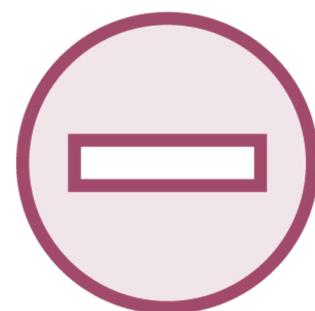
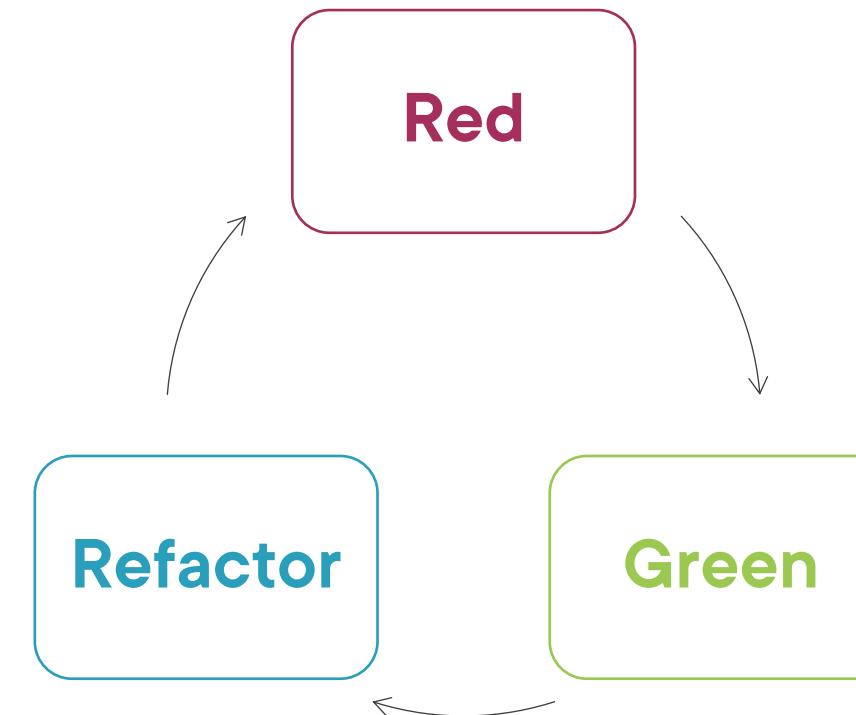
Test-driven Development

“A **software development process** that relies on the repetition of a **very short development cycle**: requirements are turned into very specific test cases, then the software is improved to pass the new tests, only.” - Wikipedia

“**Red – Green - Refactor**”



Red – Green - Refactor



Red

Write test that fails



Green

Make test pass



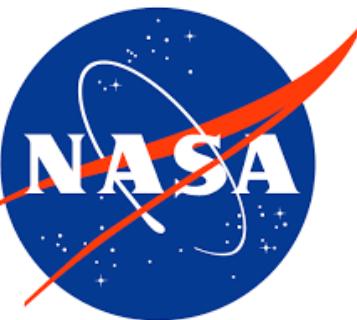
Refactor

Refactor/cleanup code



History of Test-driven Development

Test-driven Development History



"Simple Smalltalk Testing"

SUnit
Kent Beck

1960s

1994

eXtreme Programming
Kent Beck

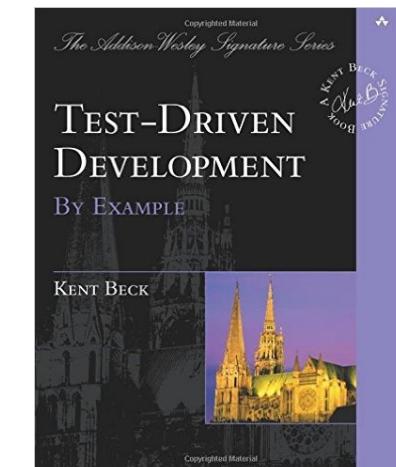
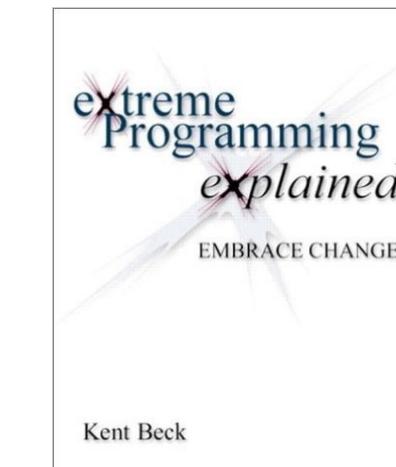
1999

TDD: By Example
Kent Beck

2002

JUnit

*Kent Beck
Erich Gamma*



Test-driven development is
well-established and has been
used for many years now



Why Practice Test-driven Development?

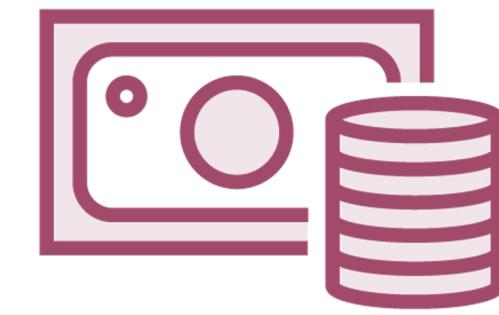
Business Benefits



Requirements
Verification



Regression
Catching



Lower
Maintenance Costs

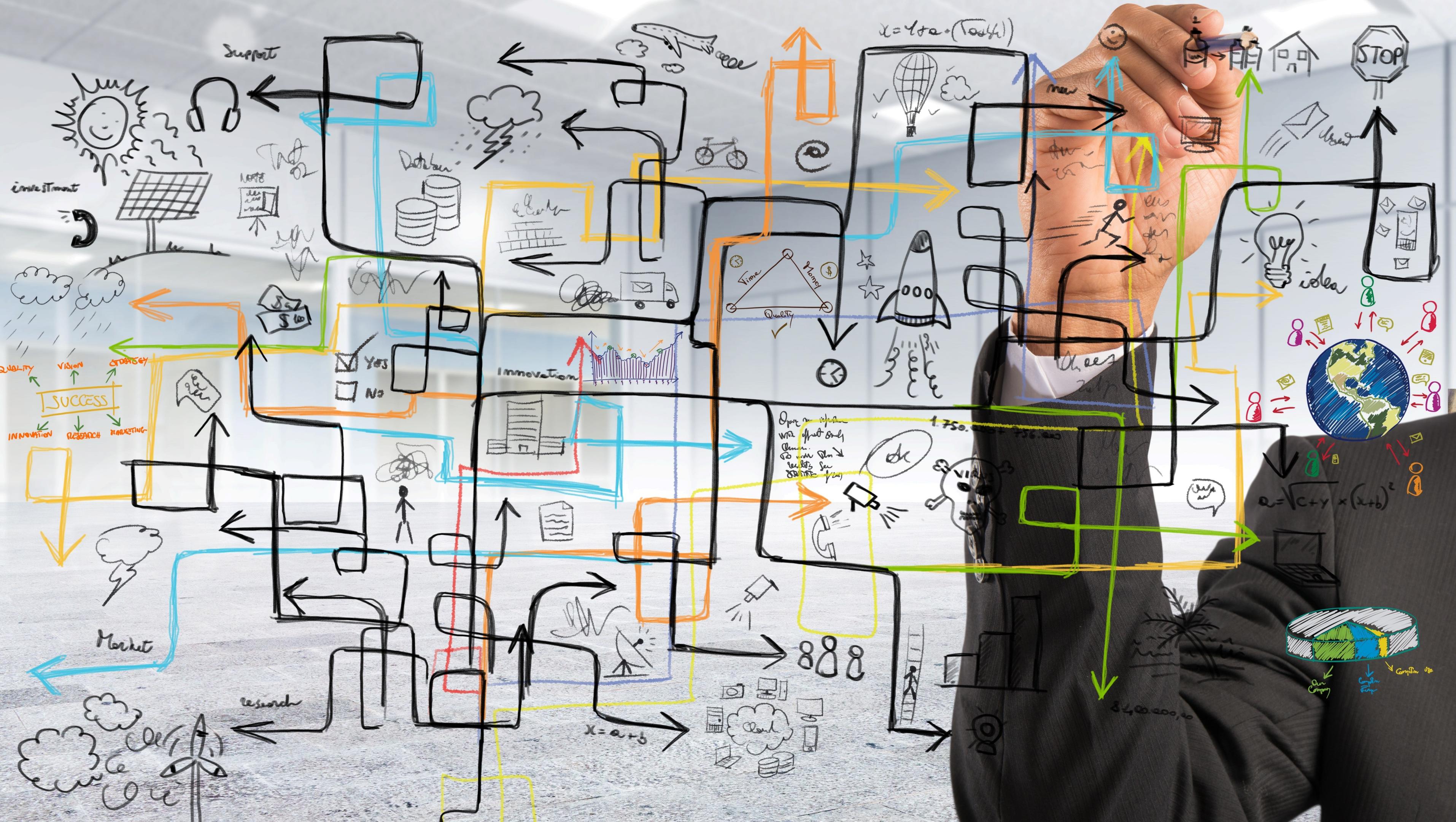


There are developer benefits
too!



Design-first





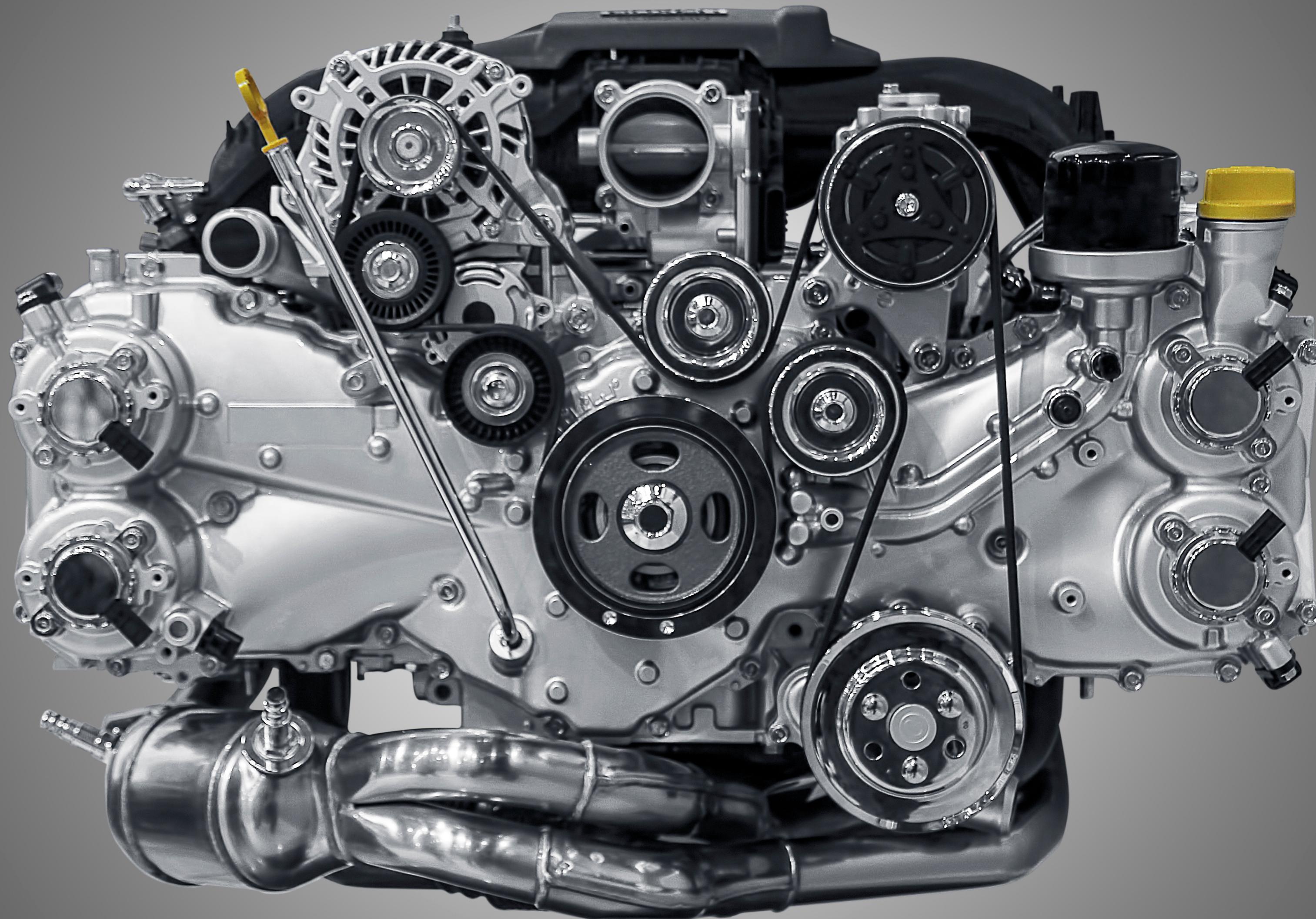




Keep focused on the
customer's needs!









Focus on delivering value!



Summary



Software development challenges

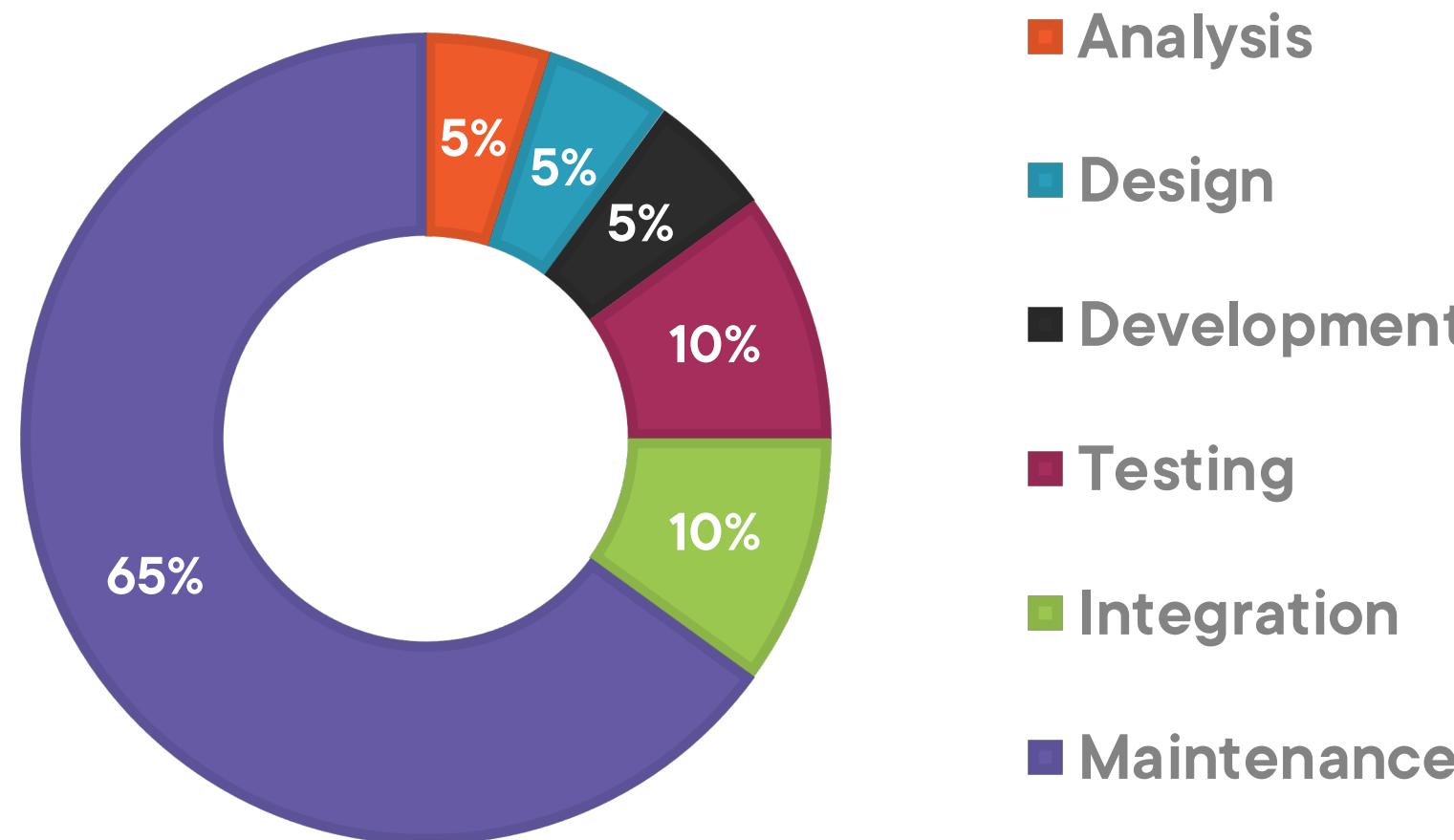
What is test-driven development?

History of test-driven development

Why practice test-driven development?



Costs of Developing Software



**Maintenance accounts for 65%
of all software development costs!**

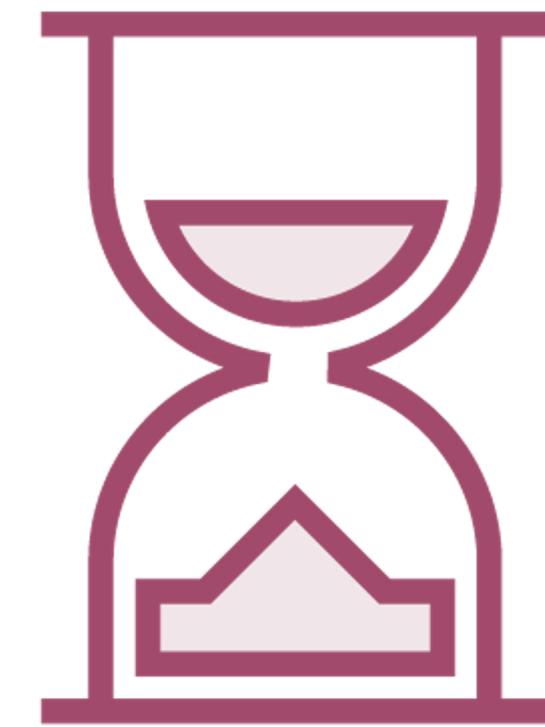
Maintenance Challenges



Code Entropy



Isolated Ownership



Infrequent Validation



Test-Driven Development

“A **software development process** that relies on the repetition of a **very short development cycle**: requirements are turned into very specific test cases, then the software is improved to pass the new tests, only.” - Wikipedia

“**Red – Green - Refactor**”



Business Benefits

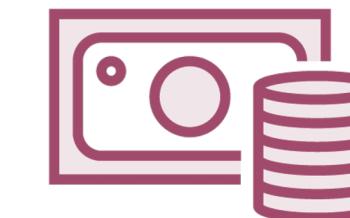
Advantages of a good testing practice...



Verifying requirements



Catching regressions



Lowering maintenance costs





Keep focused on the
customer's needs!

