If I am a developer, why bother with testing?

Rosa Maria Batista Castillo

Prácticas modernas para crear software con calidad y sabor

#SGVirtual

About Me

- Bachelor of Computer Sciencie
- +15 Software Developer
- Technical Lead
- Agile enthusiast
- Cats lover
- Crossfiter

Agenda

01. GLOBAL DISASTERS

02. THE ORIGIN OF EVERYTHING

03. GET TO THE POINT

04. WHY IS TESTING IMPORTANT TO DEVELOPERS

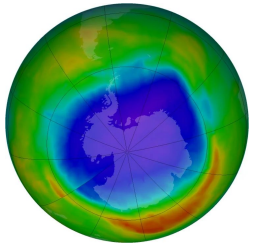
05. TAKEAWAYS

06. SUGGESTED READING

01.

Global Disasters

Software bug examples in history

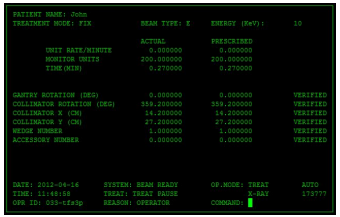


1978

Undetected hole in the ozone layer

1985

Therac 25 Deadly Radiation Therapy



1998

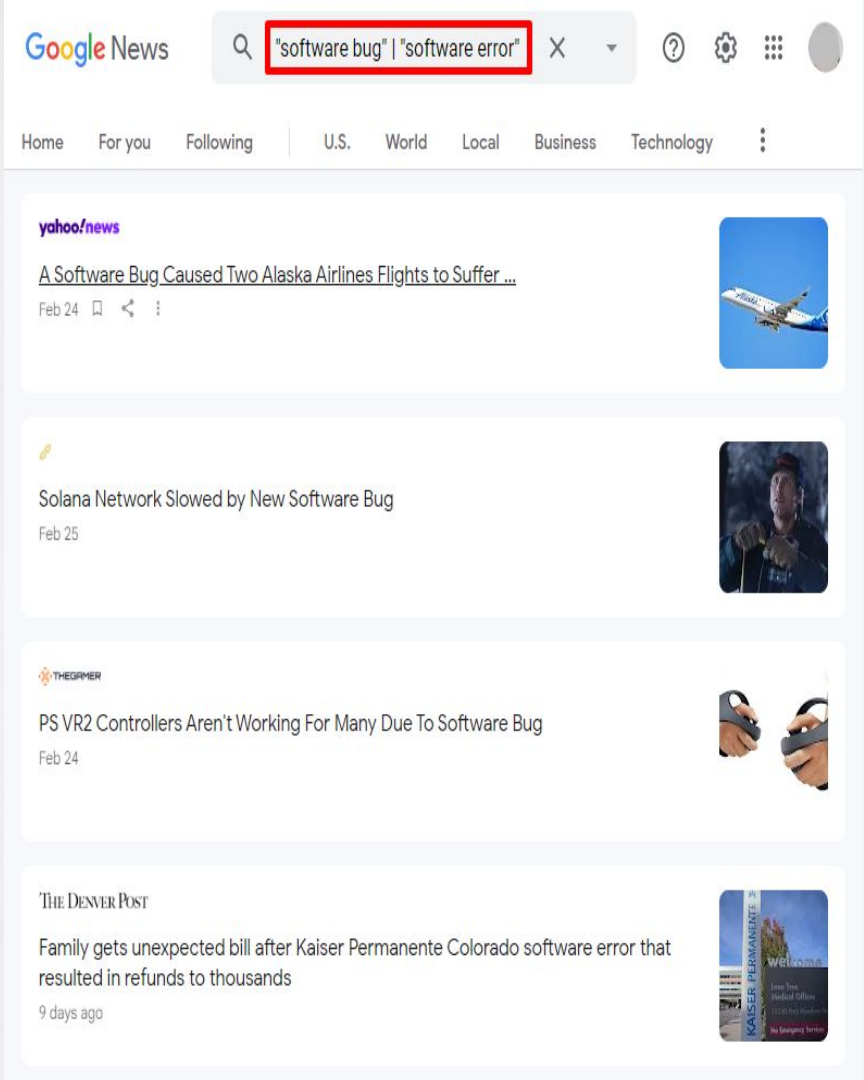
NASA's Mars Climate Orbiter

2013

ObamaCare failed launch

A little experiment

"software bug" | "software error"



Causes of a software error

MISTAKE

MISUNDERSTANDING

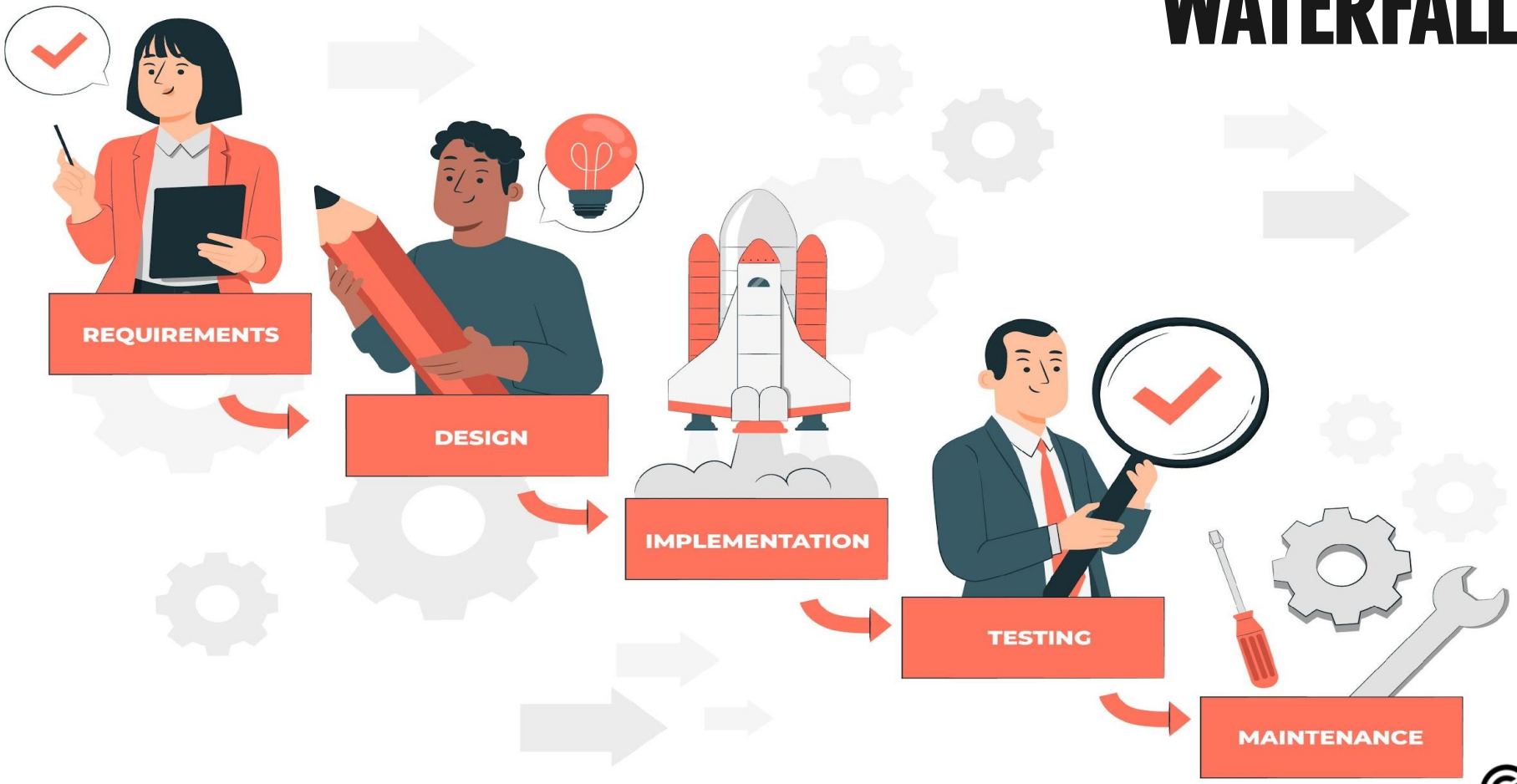
MISASSUMPTION

The effect (the occurrence of the fault) may be random but the cause (the error in the code), is always there: it’s not random.

02.

The origin of everything

WATERFALL


REQUIREMENTS
DESIGN
IMPLEMENTATION
TESTING
MAINTENANCE

AGILE

What is Testing?

Definition

It is the process of evaluating a system or software application to ensure that it meets its specified requirements and works as intended.

Purpose

Identify defects, errors, and bugs that could affect the performance, reliability, security, or user experience of the system or software.

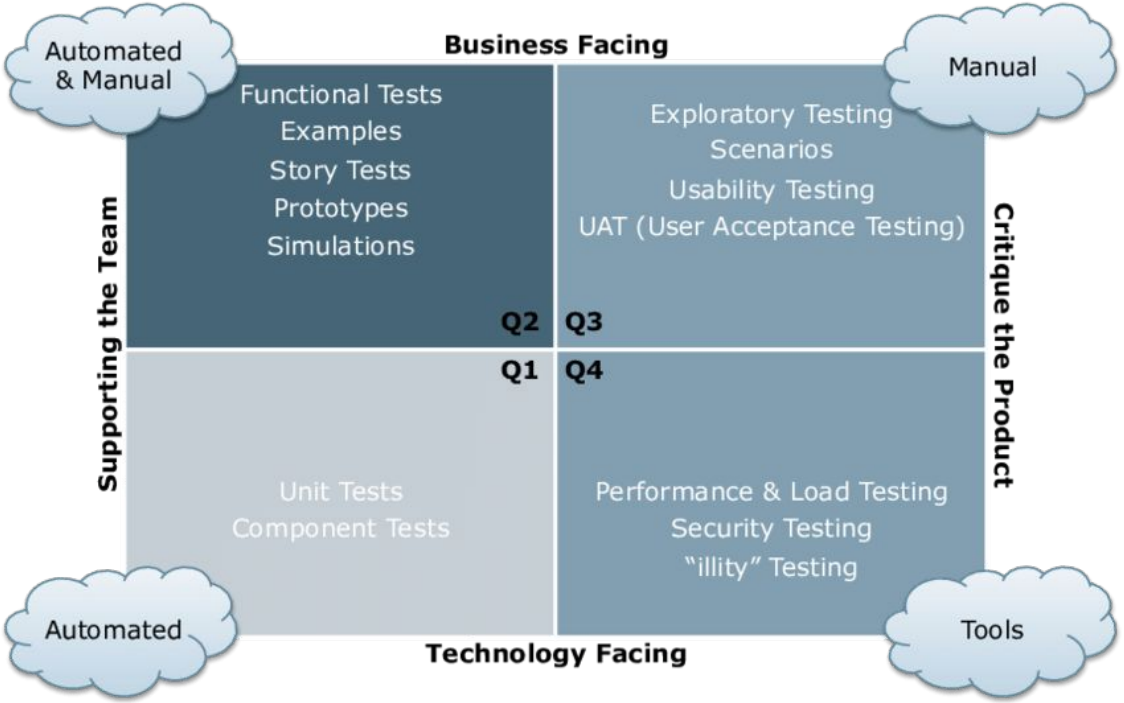
Goal

Improve the quality of the software or system being developed, by catching defects early in the development process, and ensuring that the software meets the needs and expectations of its users and stakeholders.

03.

Get to the point

The Agile Testing Quadrant



Unit Testing - TDD

What is?

It is a type of software testing where individual units or components of a software application are tested in isolation from the rest of the system.

Purpose

To ensure that each unit of code or component of the application performs as intended and meets its specified requirements.

Frameworks

- Junit
- NUnit
- PHPUnit

More details

- Unit testing is typically performed by developers as part of the development process.
- Helps to catch defects early in the development cycle, before they become larger and more difficult to fix, which makes it easier and less costly to fix them.
- Helps to ensure that changes or modifications to the code do not break existing functionality.
- Provides clear and concise documentation of the code and its behavior, making it easier for other developers to understand and maintain the software.
- Provides a safety net that allows developers to make changes with confidence.

Integration Testing

What is?

It is a type of software testing that is performed to verify the interactions between different components or modules of an application.

Purpose

To identify any defects or issues that may arise when the individual components or modules are integrated together as a system.

Types

- **Top-Down integration testing:** involves testing the higher-level modules first and gradually integrating lower-level modules.
- **Bottom-Up integration testing:** involves testing the lower-level modules first and gradually integrating higher-level modules.
- **Incremental integration testing:** involves testing and integrating modules in small increments until the entire system is integrated and tested.

More details

- Integration testing is typically performed by developers as part of the development process.
- Helps to identify defects or issues that may arise due to the interactions between different components or modules of an application.
- Helps to prevent more serious issues from occurring later on, when they may be more difficult and costly to fix.

Acceptance Testing - BDD

What is?

It is a type of software testing that is performed to determine whether a software application meets the acceptance criteria and requirements specified by the customer or end user.

Purpose

To ensure that the software is suitable for use by the customer or end user and meets their needs and expectations.

Types

- **User Acceptance Testing (UAT):** is used to determine whether the product is working for the user correctly.
- **Business Acceptance Testing (BAT):** is used to determine whether the product meets the business goals and purposes or not.
- **Operational Acceptance Testing (OAT)**: is used to determine the operational readiness of the product. It mainly includes testing of recovery, compatibility, maintainability, reliability, etc.

More details

- Any issues or defects that may affect the usability or functionality of the software can be identified and addressed before the software is released.
- Minimize or eliminate the issues arising from production, preventing costly and time-consuming fixes.
- Can improve customer satisfaction and user adoption of the software.

Related Agile Practices

TDD

BDD

Refactoring

Pair/Mob Programming

CI/CD

04.

So, why is Testing important for developers?

FINDING & FIXING DEFECTS

IMPROVING SOFTWARE QUALITY

ENHACING USER EXPERIENCE

SAVING TIME & MONEY

INCREASING CONFIDENCE

FINDING & FIXING DEFECTS

01 Enhance reputation and credibility

02 Build trust

03 Enhance developer skills

04 Security

IMPROVING SOFTWARE QUALITY

01 Promoting modularity

02 Ensuring requirements are met

03 Encouraging better code quality

04 Reducing technical debt

05 Supporting refactoring efforts

06 Supporting continuous integration

07 Supporting continuous improvement

ENHANCING USER EXPERIENCE

01 Identifying usability issues

02 Ensuring functionality

03 Ensuring reliability

04 Supporting continuous integration

05 Enhancing usability

06 Ensuring performance

SAVING TIME & MONEY

01 Early detection of defects

02 Increasing productivity

03 Reducing rework

04 Improving efficiency

05 Avoiding customer impact

06 Facilitating automation

INCREASING CONFIDENCE

01 Ensuring functionality

02 Ensuring reliability

03 Ensuring security

04 Ensuring performance

05 Ensuring compliance

06 Demonstrating quality

05.

Takeaways

Abandon
the old ways

Incorporate testing
is a "must" not
a "nice to have"

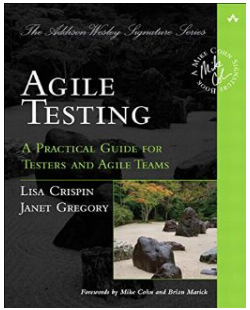
If you haven't started,
start today!

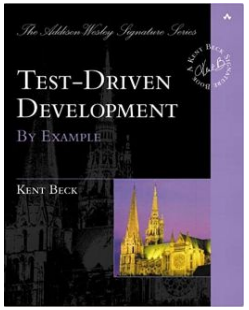
If your organization
is reluctant to adopt it,
start making the change.

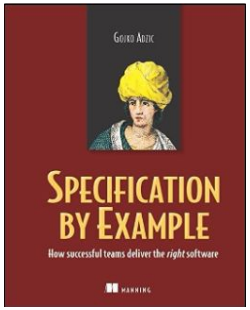
If you already started,
you're on the right path.
It's time to become an
evangelist.

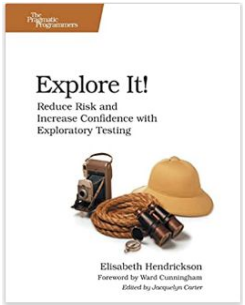
06.

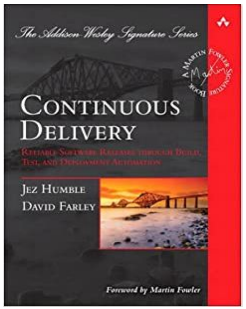
Suggested Reading

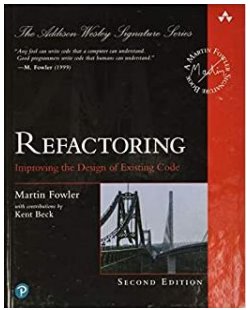

The Addison Wesley Signature Series
AGILE TESTING
A PRACTICAL GUIDE FOR TESTERS AND AGILE TEAMS
LISA CRISPIN
JANET GREGORY
Forewords by Mike Cohn and Brian Marick


The Addison Wesley Signature Series
TEST-DRIVEN DEVELOPMENT
BY EXAMPLE
KENT BECK


GOJKO ADZIC
SPECIFICATION BY EXAMPLE
How successful teams deliver the right software
MANNING


The Pragmatic Programmers
Explore It!
Reduce Risk and Increase Confidence with Exploratory Testing
Elisabeth Hendrickson
Foreword by Ward Cunningham
Edited by Jacquelyn Carter


The Addison Wesley Signature Series
CONTINUOUS DELIVERY
JEZ HUMBLE
DAVID FARLEY
Foreword by Martin Fowler


The Addison Wesley Signature Series
REFACTORING
Improving the Design of Existing Code
Martin Fowler
Kent Beck
SECOND EDITION

Thanks for staying tuned!

Any question?

rosa.batista@improving.com
https://www.linkedin.com/in/rosabatistacastillo/