Software Testing

Exploratory testing.
Regression testing

SOFT ACADEMY

www.softacademy.bg

Overview

- Exploratory testing – session, charters, heuristics
- Regression testing
- Continuous Integration, Continuous Delivery, Continuous Deployment

Exploratory testing

- Exploratory testing is an approach to testing often described as „simultaneous learning, test design and test execution“
- Exploratory testing produces best results when combined with more formal approach (pre-designed tests), but is also very useful when:
  - limited time for testing or no time for test design
  - no clear requirements
  - testing to uncover problems in an existing system (recon sessions)

Exploratory testing

- Exploratory testing can be a never-ending activity
- Adopting **time-boxed sessions** helps keep exploration focused
- Each session has a goal and tester takes notes to keep track of what he/she has found / covered
- **Exploratory charters** – a one-sentence mission statements that define target, resources and information to be sought during each exploratory session

Exploratory testing

- Charters should be focused, but not give detailed instructions on actions and expected result (this is a test case)

Explore creating profile with username containing Cyrillic symbols to discover what error message will be displayed ✗

Explore creating user profile with different set of input characters for username & password, to discover potential problems ✓

Explore the application for invalid input ✗

Exploratory testing

- Exploratory testing can be combined with many testing techniques and heuristics
- Heuristics examples:

  - **Never and always** (*example: banking software should never allow making a transaction without active authentication session*)

  - **Beginning, Middle, End** (*example: putting space in the beginning, middle, end of a string*)

  - **Some, None, All** (*example: none treated as all during search filtering*)

  - **Zero, One, Many** (*example: 0 results found during search*)

  - **Interruptions** (*example: interrupt the program while performing a critical operation*)

Regression Testing

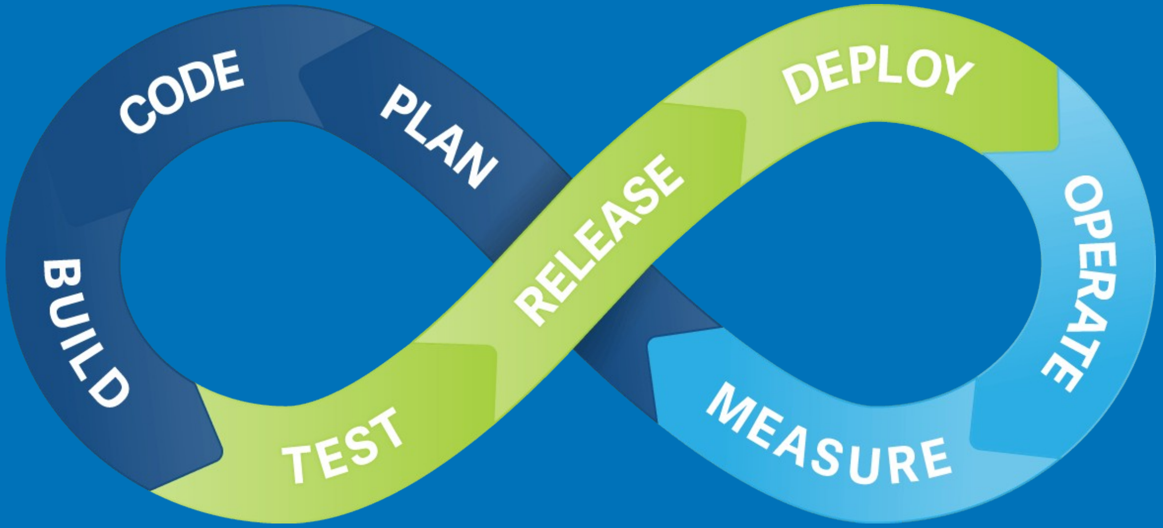
- **Regression testing** - testing of a previously tested program following modification to ensure that defects have not been introduced or uncovered in unchanged areas of the software, as a result of the changes made
- Scope of regression tests can be fixed (all tests executed every time) or based on impact analyses (testing in areas affected by the changes). Approach depends on resources and risks
- As code base grows, so does the time needed to perform regression tests. One of the main solutions to this problem is automation testing

Continuous Integration / Continuous Delivery

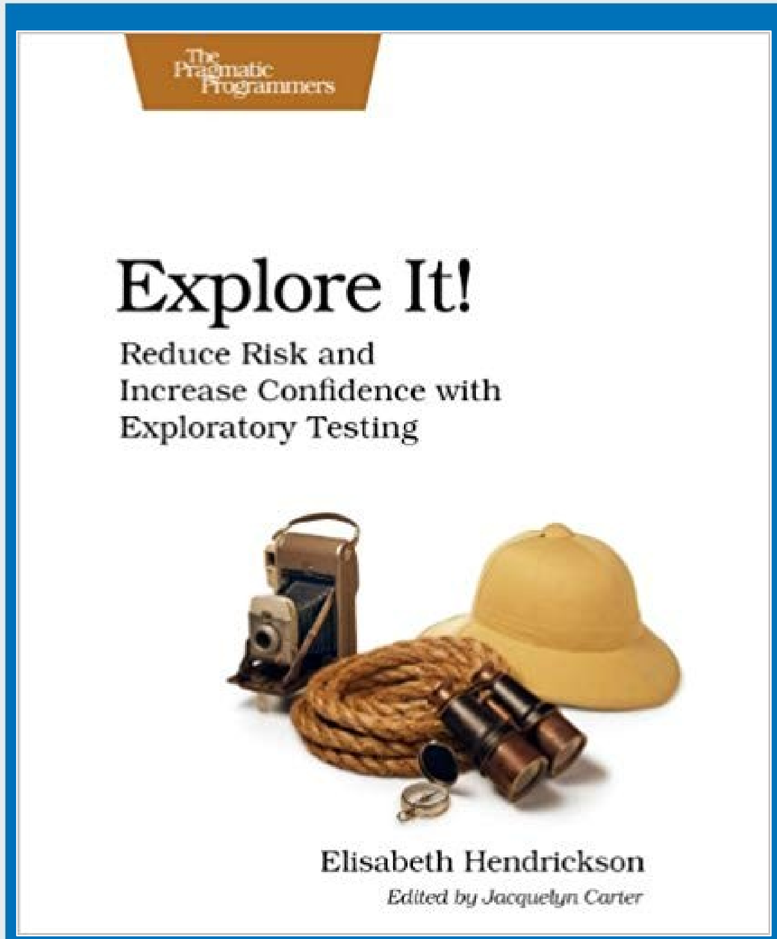
- Automation testing is a key concept in Agile development, which enables teams to deliver a piece of functionality with good quality in a very limited amout of time
- **Continuous Integration (CI)** is a development practice that requires developers to integrate code into a shared repository each day. Each check-in is then verified by an automated build, allowing teams to detect problems early.
- **Continuous Delivery (CD)** is a development approach where each build is automatically tested, allowing teams to release stable software virtually at any time
- **Continuous Deployment** takes these practice further, allowing for fully automated deployment to a staging or production site without need of manual validation

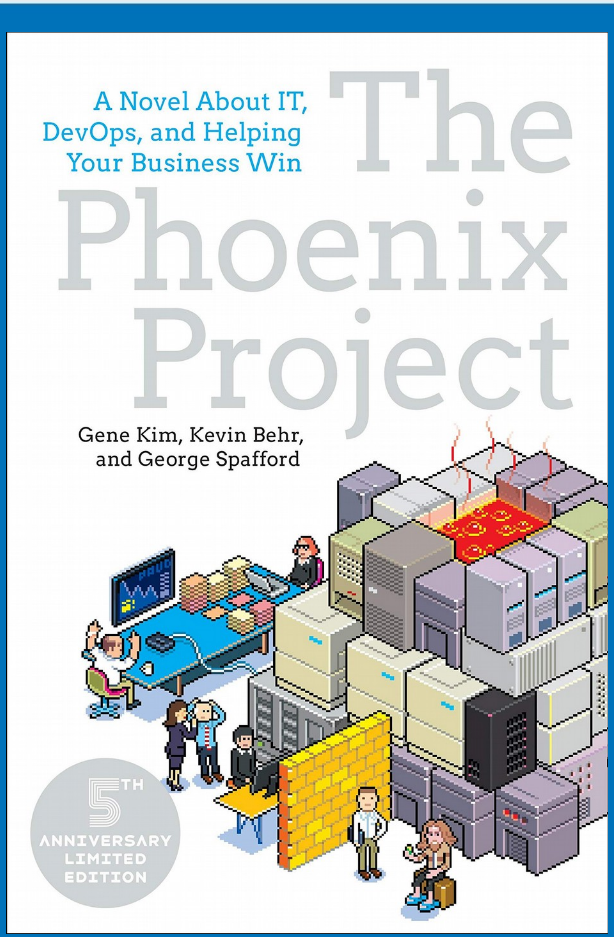
Continuous Integration / Continuous Delivery

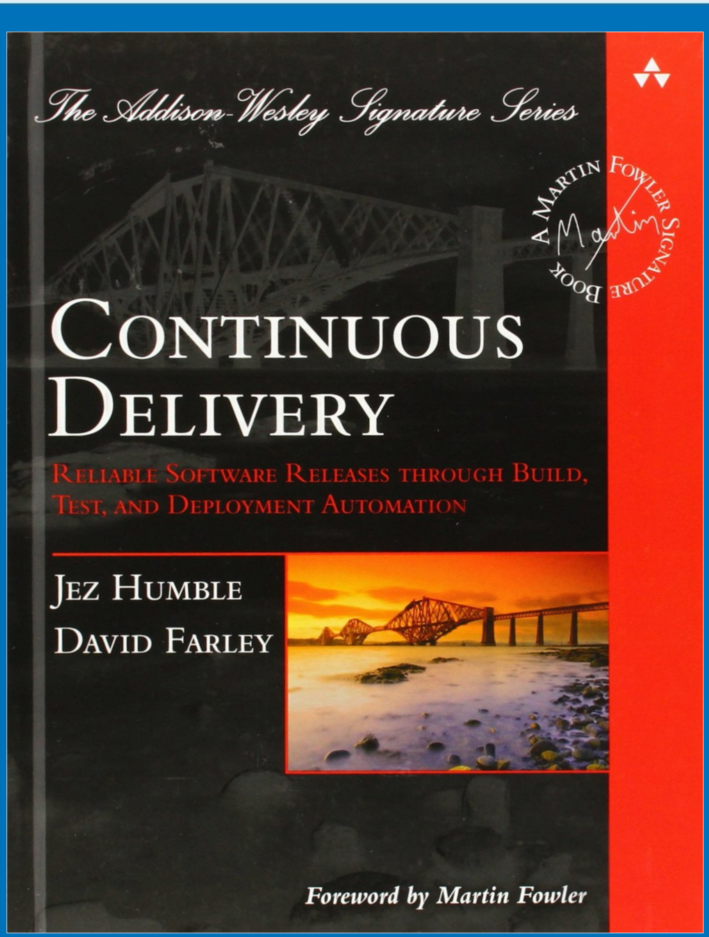
- CI/CD tools are used to orchestrate and manage build – test – deployment pipeline
- Popular CI/CD tools: Jenkins, Bamboo, TeamCity, etc.



Further reading


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


A Novel About IT, DevOps, and Helping Your Business Win
The Phoenix Project
Gene Kim, Kevin Behr, and George Spafford
5TH ANNIVERSARY LIMITED EDITION


The Addison-Wesley Signature Series
A Martin Fowler Signature Book
Continuous Delivery
Reliable Software Releases through Build, Test, and Deployment Automation
Jez Humble
David Farley
Foreword by Martin Fowler

Q
A