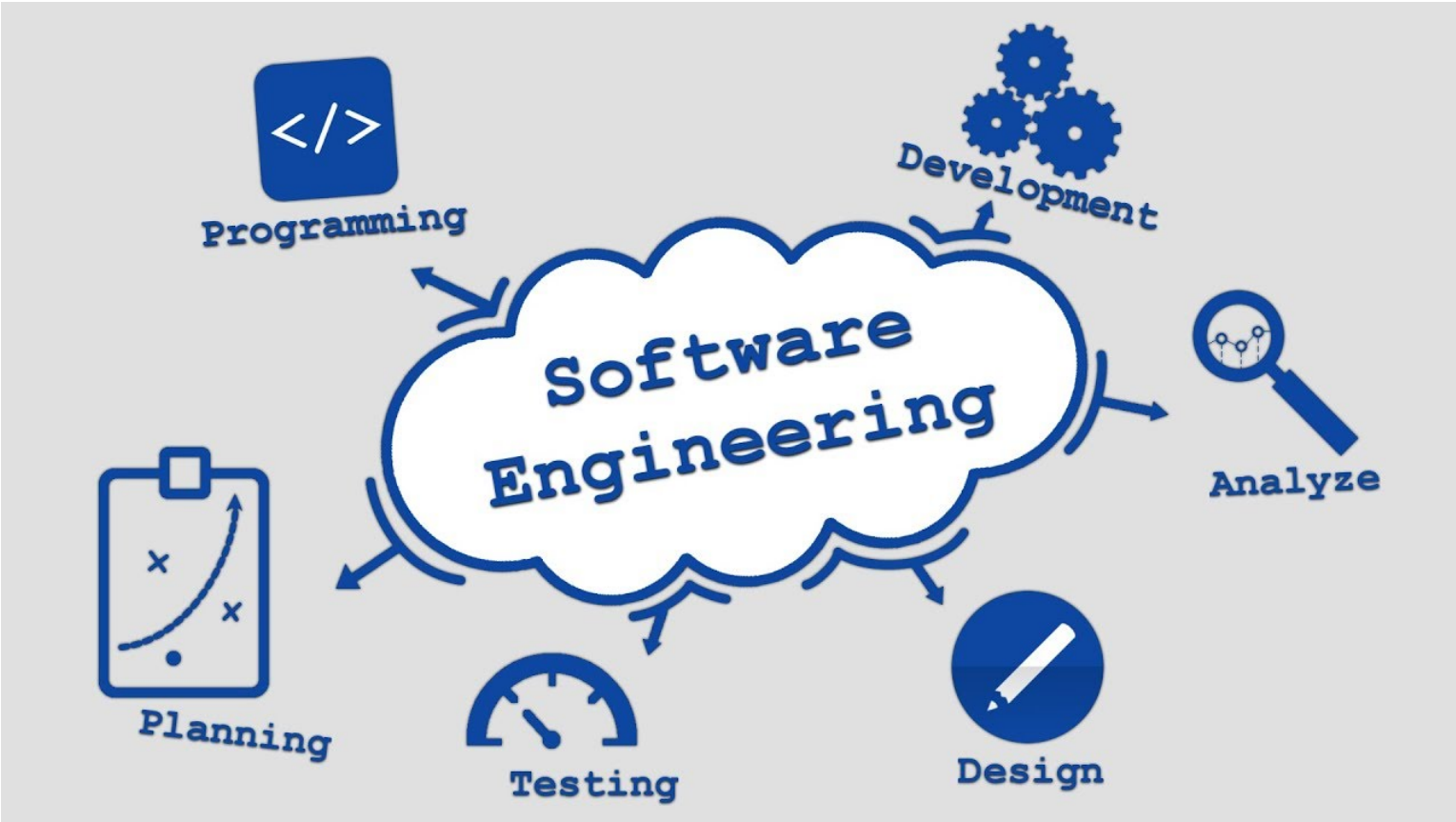
UNIVERSITY OF LEICESTER

INFORMATICS

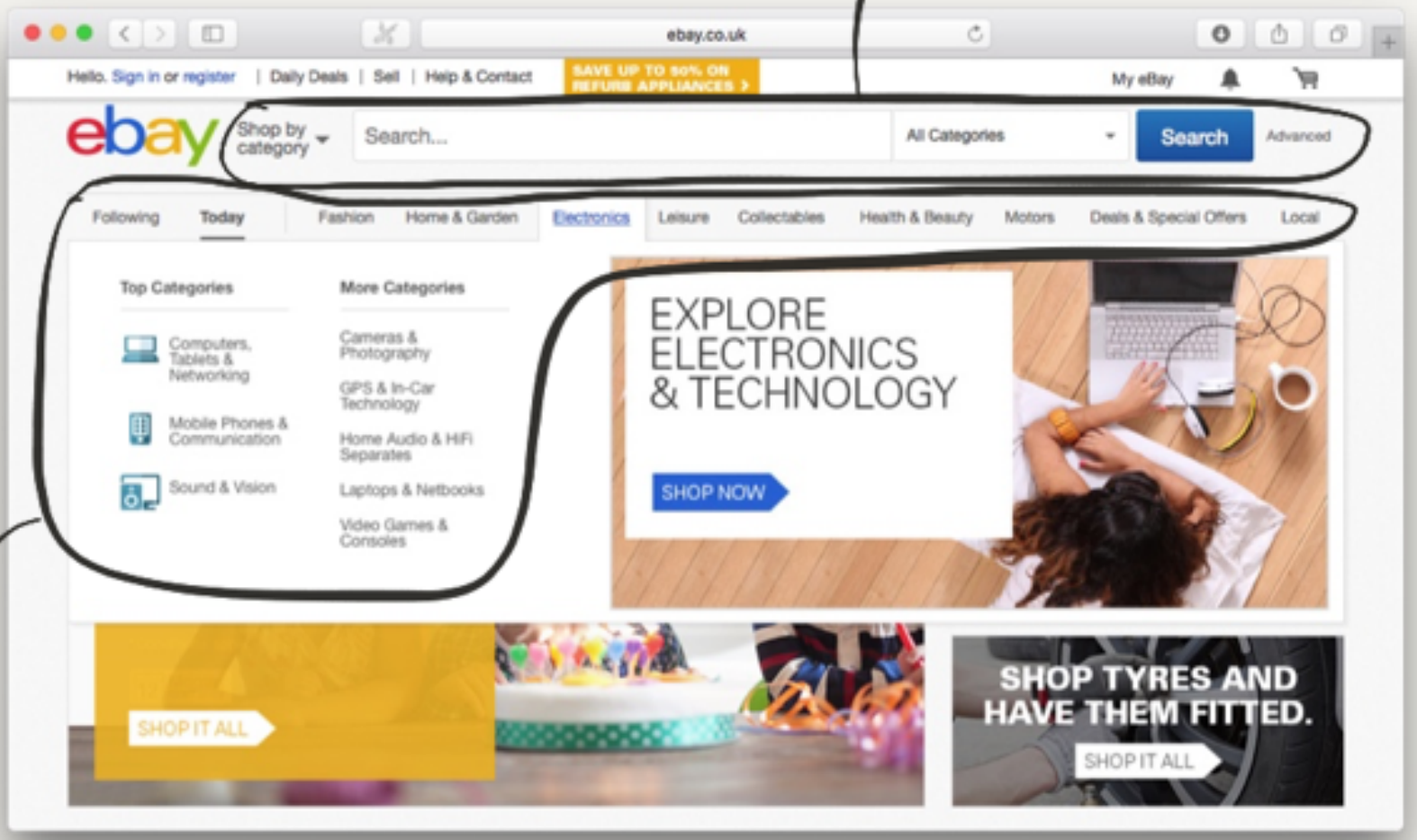
CO2006
Software Engineering and System Development

Artur Boronat



Online shop

search products

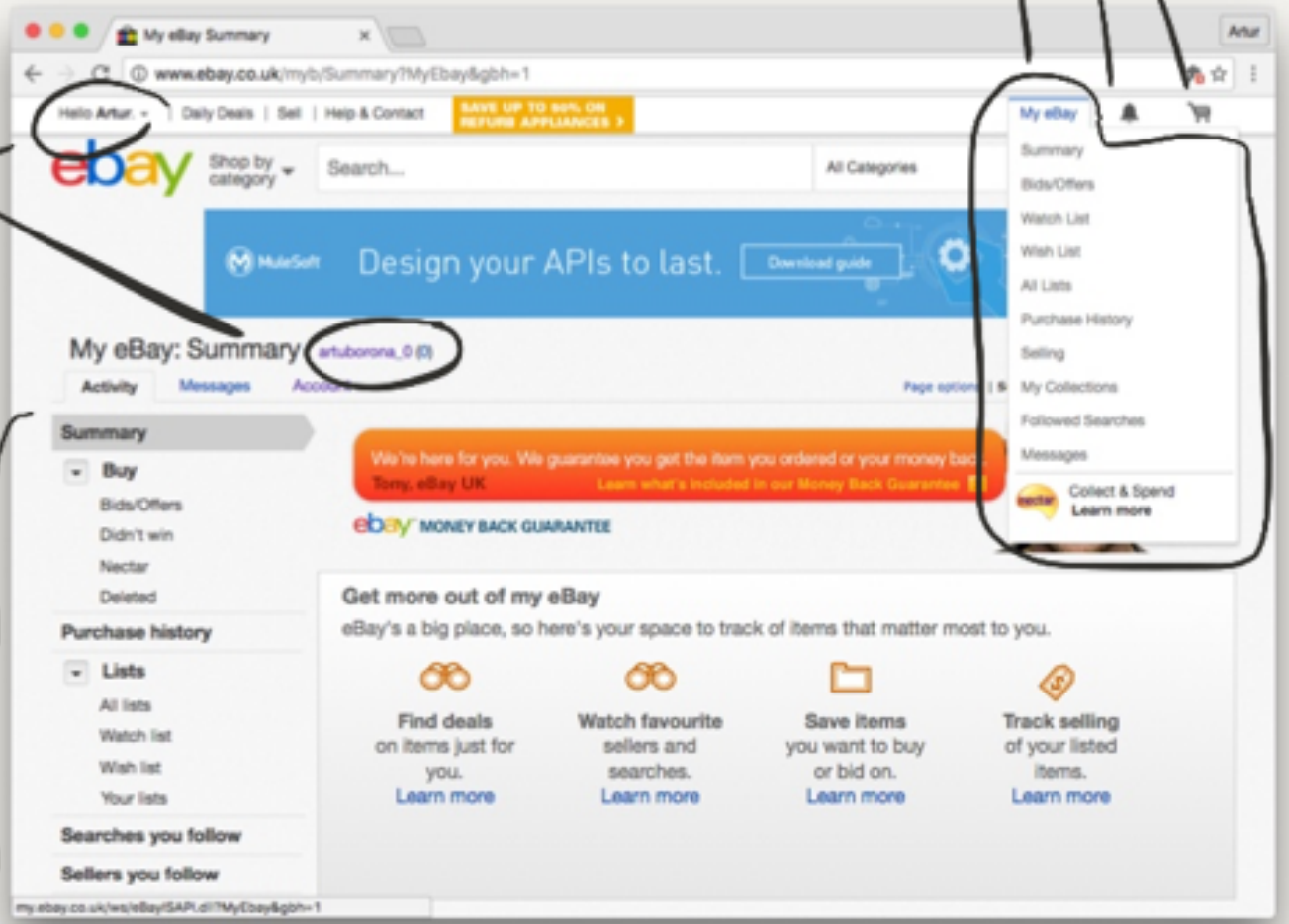


browse the website

Online shop

functional features

EBAY knows me and



remembers what I did

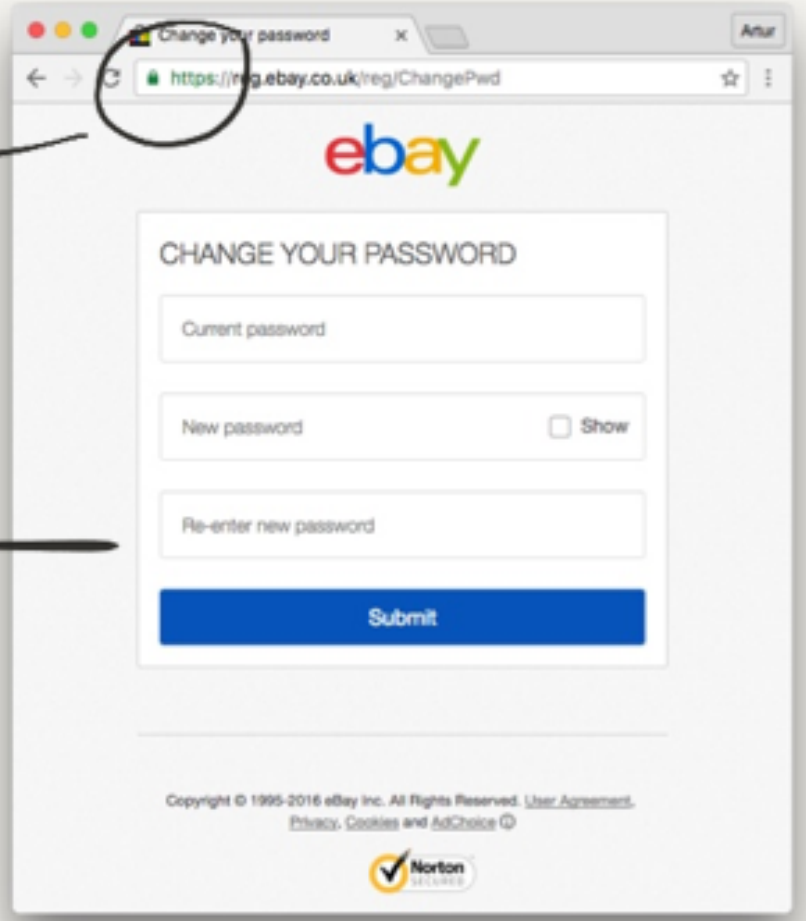
DATA IS PERSISTED

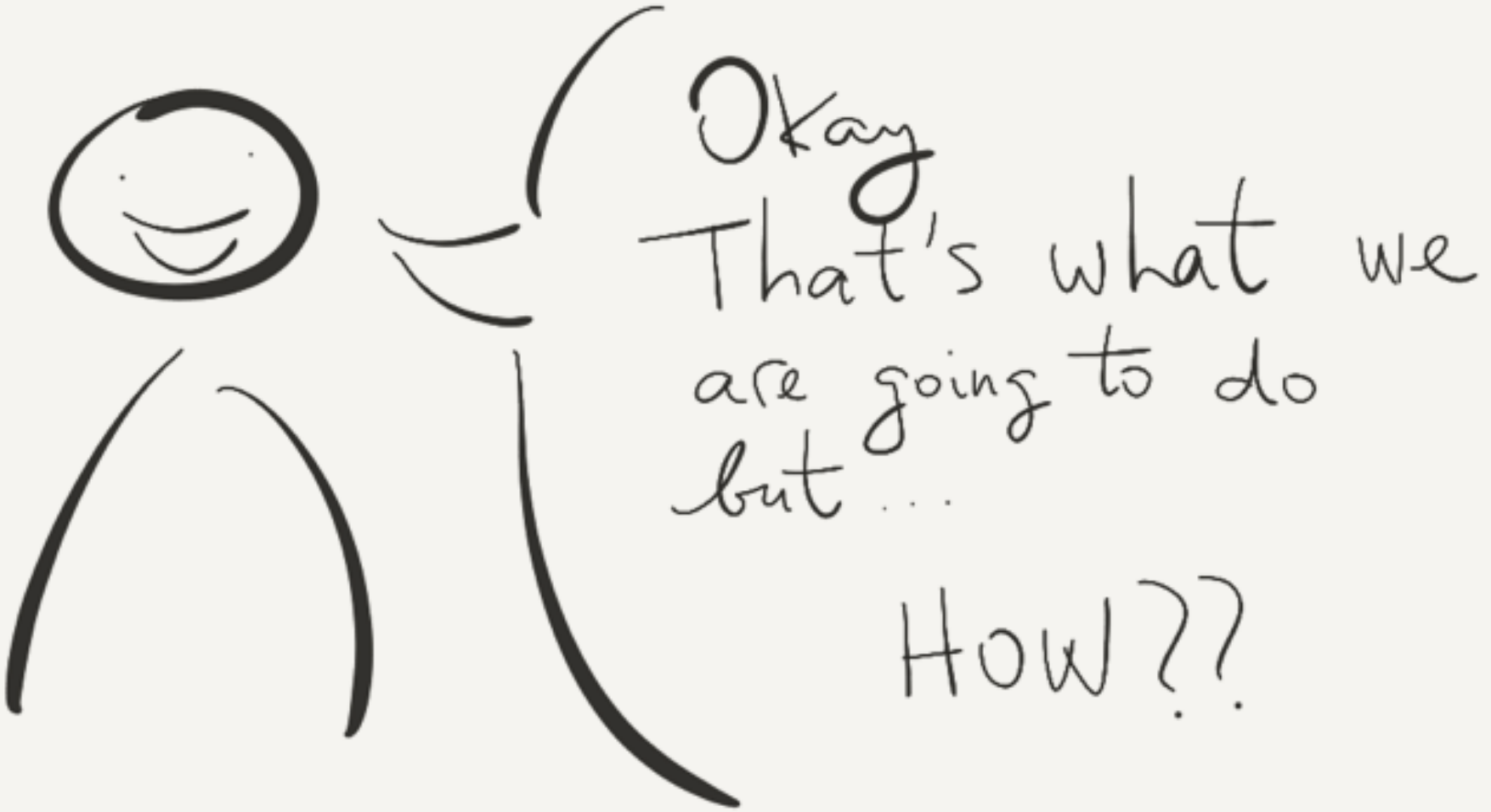
Online shop

secure
Communication

authentication
mechanism

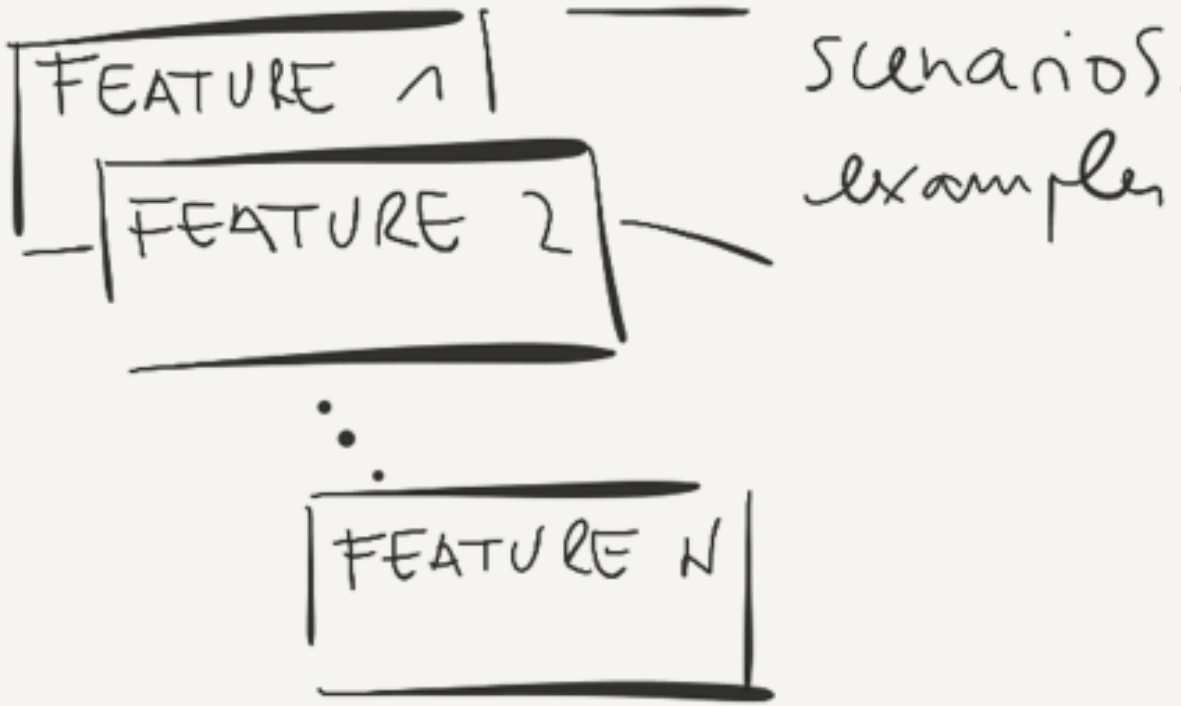
SECURITY




Okay
That's what we
are going to do
but...
HOW??

SOFTWARE METHODOLOGY

(1) REQUIREMENTS SPECIFICATION



SOFTWARE METHODOLOGY

1. REQUIREMENTS SPECIFICATION
2. DESIGN AND DEVELOPMENT

Spring MVC
Spring Data
Spring Security
L used at
EBAY

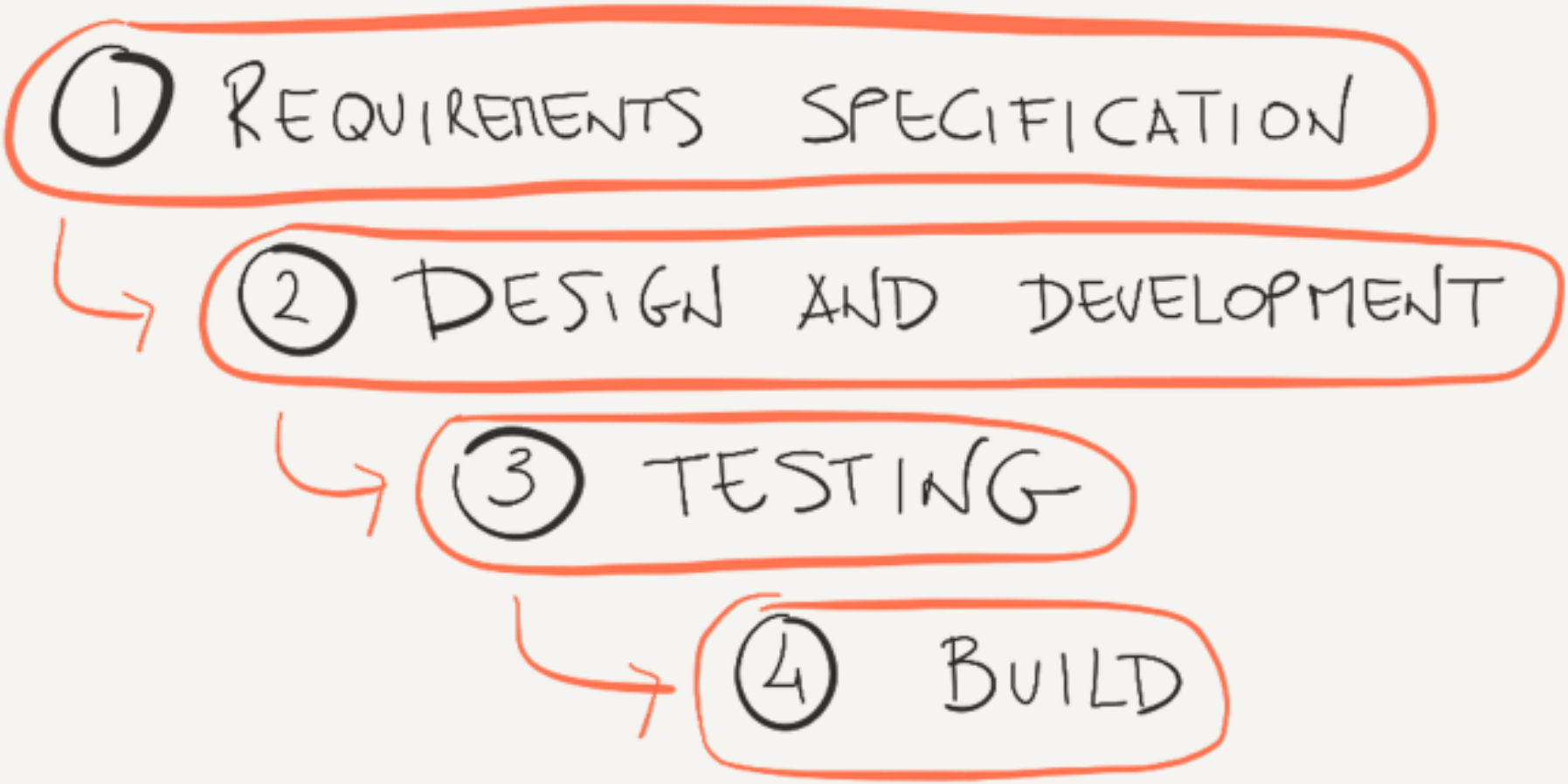
SOFTWARE METHODOLOGY

1. REQUIREMENTS SPECIFICATION
2. DESIGN AND DEVELOPMENT
3. TESTING
   - Test-driven development
   - Behaviour-driven development

SOFTWARE METHODOLOGY

1. REQUIREMENTS SPECIFICATION
2. DESIGN AND DEVELOPMENT
3. TESTING
4. BUILD
   - automates tasks in 1, 2 and 3
   - Gradle

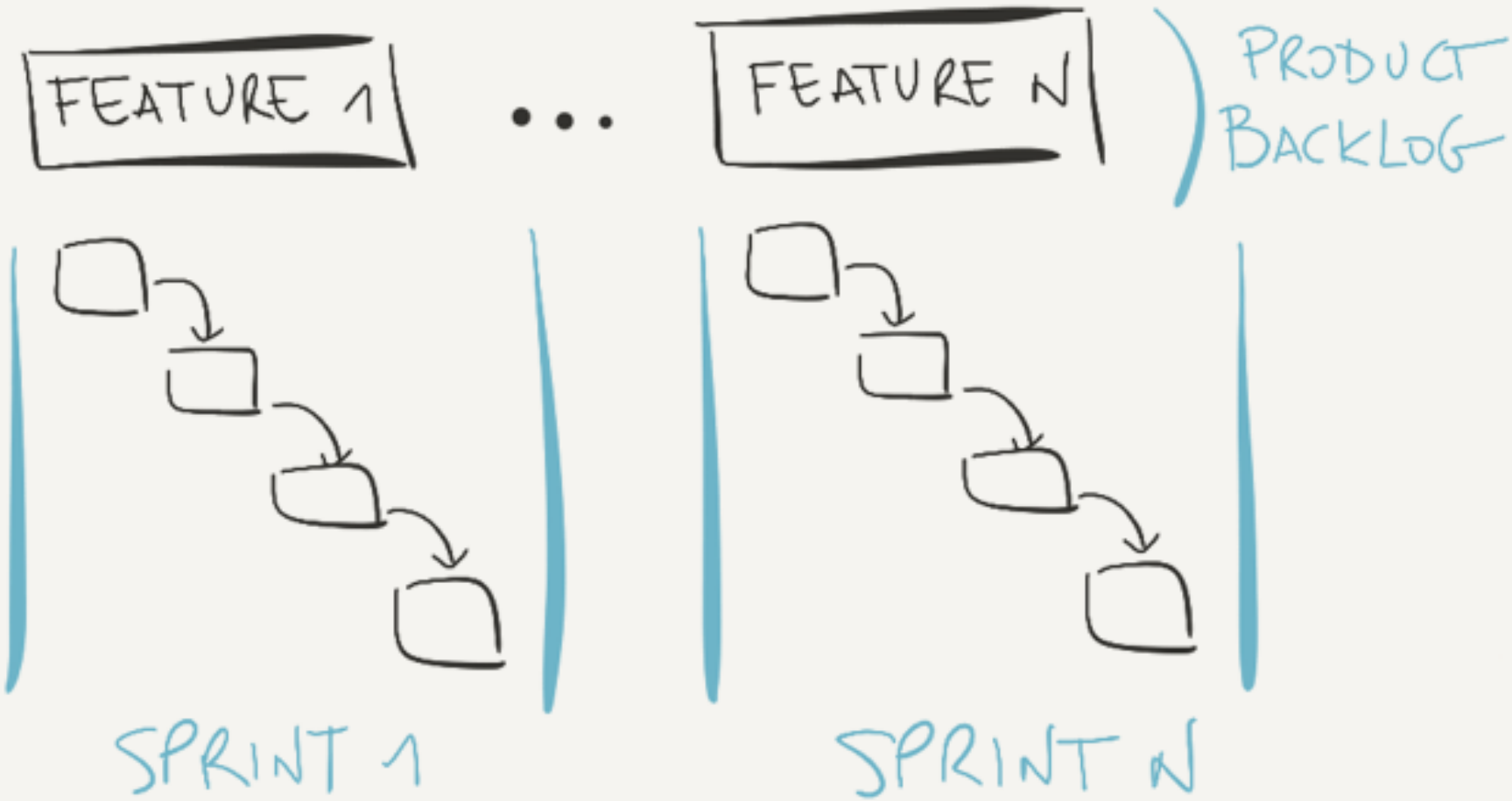
SOFTWARE METHODOLOGY



IS THIS THE WATERFALL MODEL?

SOFTWARE METHODOLOGY

... actually think AGILE



MODULE CONTENTS

SPRINT 1: AGILE + GRADLE

SPRINT 2: SPRING MVC

SPRINT 3: TDD + BDD

MODULE CONTENTS

SPRINT 1: AGILE + GRADLE

SPRINT 2: SPRING MVC

SPRINT 3: TDD + BDD

SPRINT 4: SPRING DATA
+ SECURITY

Challenges: release management

Goal

- Software release: software that is developed and tested, i.e. our goal
- Build: software that is compiled and assembled (a jar file), intermediate goal to achieve a release

Problems in release management

- Does the code compile?
- Does the code pass the tests? (unit tests)
- Does the code meet the business requirements? (functionality)
- Does the code meet the quality criteria? (performance, security, etc.)

Challenge: release management

Solution: continuous delivery

- automatically produce build artifacts (jar files)
- release often and small
- produce a Minimum Viable Product (MVP)
  - to obtain fast feedback from customers
  - reducing risks
  - ensuring continuous progress

How? Agile Methodology

- Iterative, incremental and evolutionary
- People not process (when sensible)
- Focus on quality and on maintaining simplicity
  - continuous delivery: automate as much as possible
    - optimise resources: save time
    - increase quality: to achieve repeatable and consistent processes
  - automated testing
    - quantitative measures
    - consistency
    - release readiness
- Embrace change: very short feedback loop and adaptation cycle

HOW? Tooling

compile > test > build > release

Challenges: dependency hell

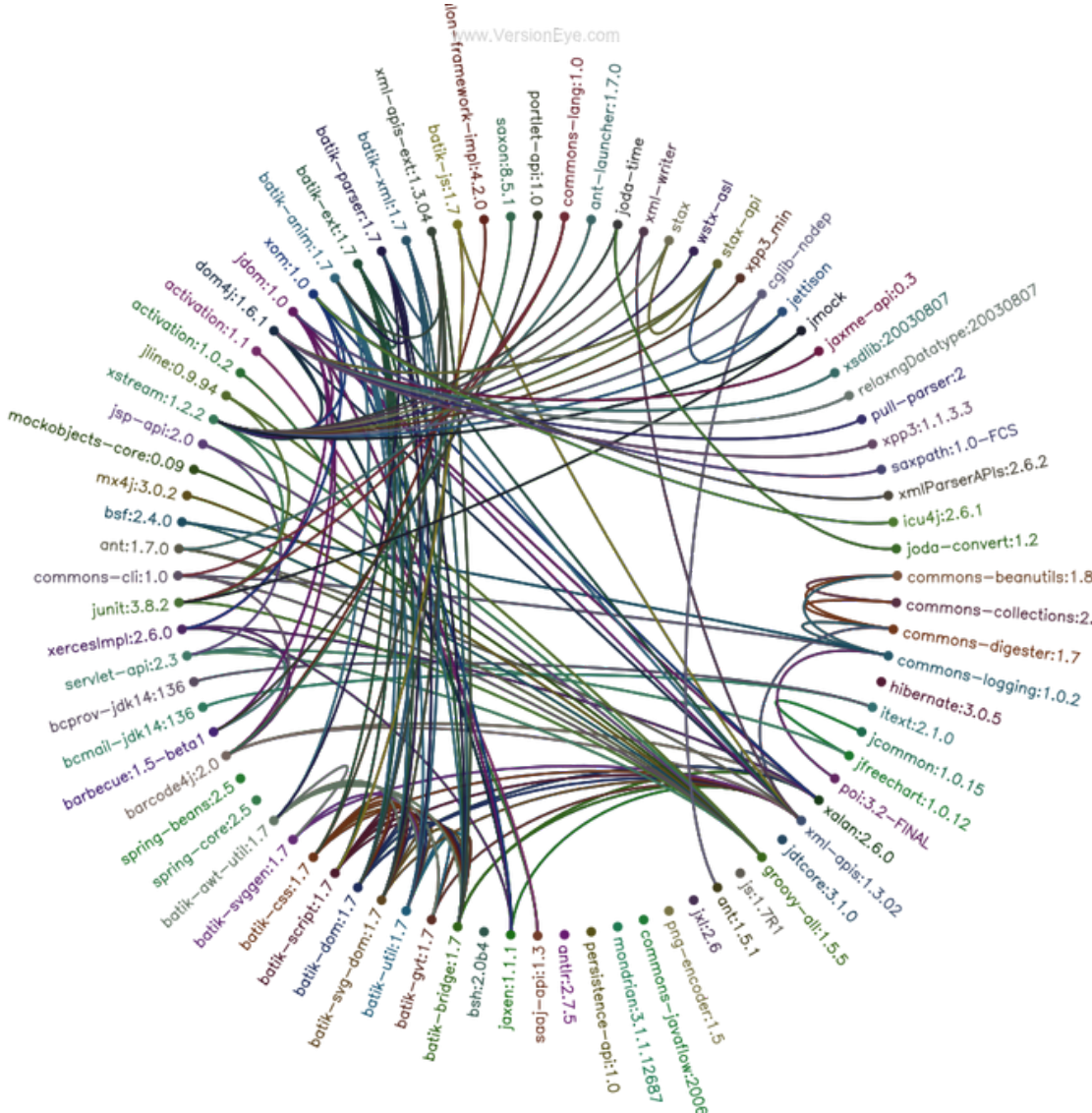
Dependency Hell

- Many dependencies
- Long chains of dependencies
- Conflicting dependencies: when different versions cannot be simultaneously installed
- Circular dependencies

Solution

Dependency management

JasperReports



During this sprint

Syllabus

- Eclipse STS and Java (revision)
- Agile principles (Pluralsight, private study)
- Groovy: scripting in the JVM ecosystem
- Gradle: scripting build automation with Groovy
    - dependency management

Assessment

- Test on 16 October: it may involve small programming exercises

Organisational Matters