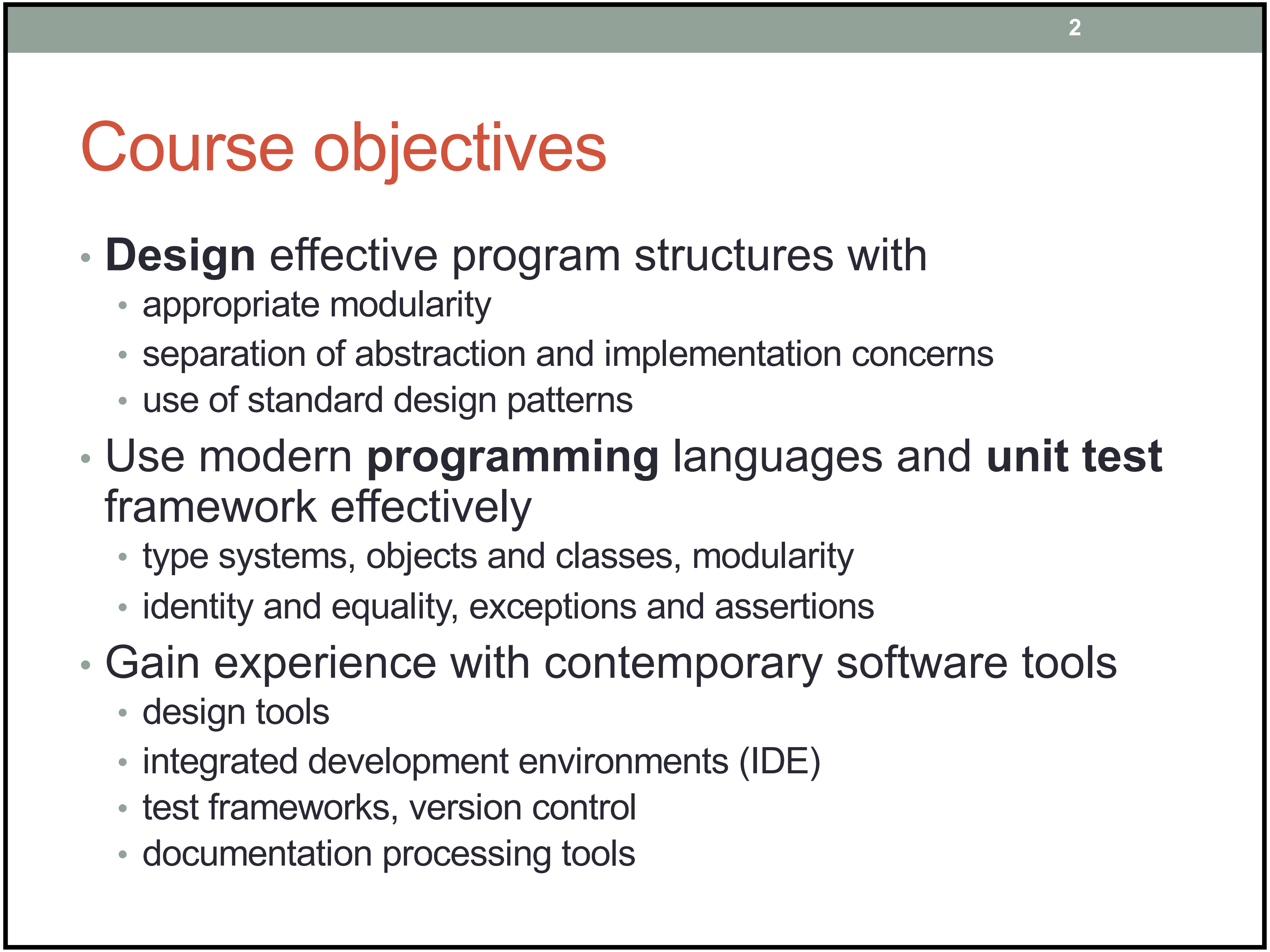
ITSS SOFTWARE DEVELOPMENT

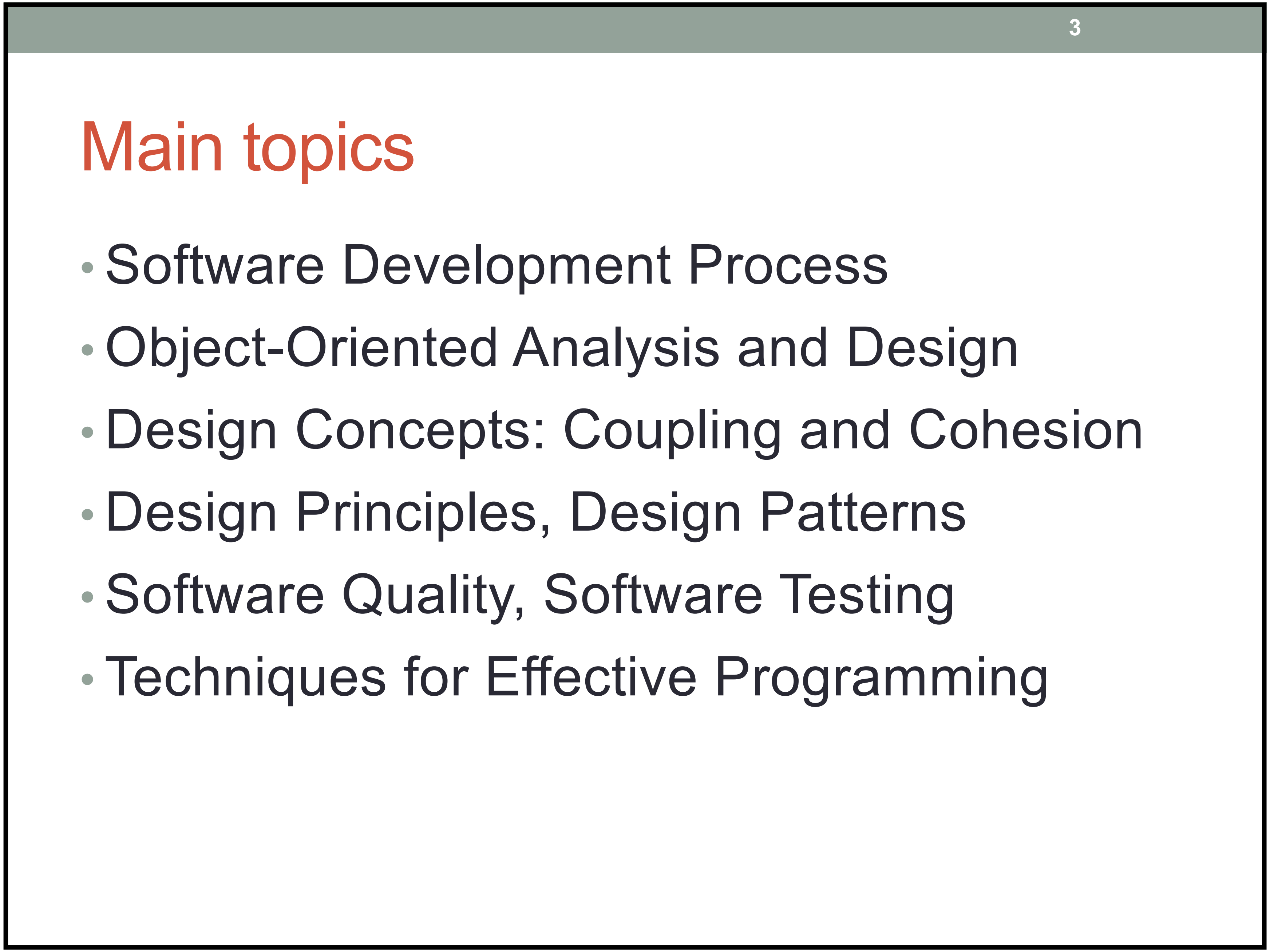
0. INTRODUCTION TO COURSE

Nguyen Thi Thu Trang
trangntt@soict.hust.edu.vn



Course objectives

- **Design** effective program structures with
  - appropriate modularity
  - separation of abstraction and implementation concerns
  - use of standard design patterns
- Use modern **programming** languages and **unit test** framework effectively
  - type systems, objects and classes, modularity
  - identity and equality, exceptions and assertions
- Gain experience with contemporary software tools
  - design tools
  - integrated development environments (IDE)
  - test frameworks, version control
  - documentation processing tools



Main topics

- Software Development Process
- Object-Oriented Analysis and Design
- Design Concepts: Coupling and Cohesion
- Design Principles, Design Patterns
- Software Quality, Software Testing
- Techniques for Effective Programming

Programming language/tools

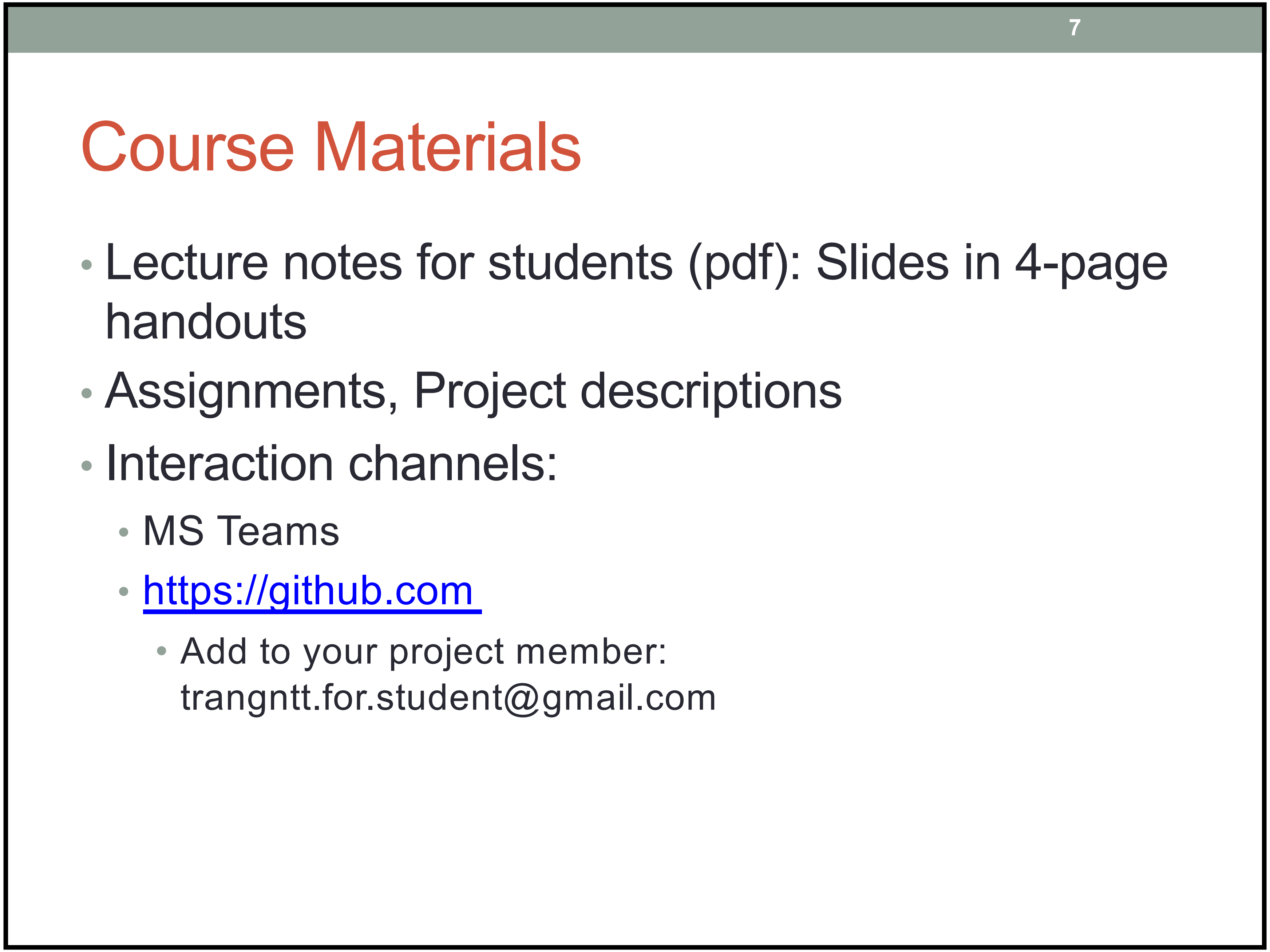
- Software design tool: Astah
  - Free for students
- Programming language: Java
- IDE: Eclipse
- Version control: Github
- Test framework: JUnit
- Architectural model / pattern: 3 tiers / MVC

Assessment

- Mid-term score: 50%
  - Case Study (Hands-on Lab)
    - Work in individual
  - Capstone Project
    - Work in groups, but individual score
  - *Reward or Punish*
    - *Quizzes/Exercises/Attendance at class: ±1*
- Final score: 50%
  - Written exam

Reference books

- D. Budgen. ***Software Design****, 2nd Edition*. Addison-Wesley. 2004.
- Cay Horstmann. ***Object-Oriented Design and Patterns***. John Wiley & Sons, Inc. 2006
- Joshua Bloch. *Effective Java, 2nd ed.* Addison-Wesley, 2008
- Boris Beizer. **Software Testing** Techniques, 2nd Edition. International Thomson Computer Press
- Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. ***Design Patterns:*** *Elements of Reusable Object-Oriented Software.* 1994. Addison-Wesley Professional.
- Robert Martin. ***Agile Software Development:*** *Principles, Patterns, and Practices*. 2003. Pearson Education.
- Robert Martin. ***Clean Code: A Handbook for Agile Software Craftsmanship****.* 2008. Prentice Hall.



Course Materials

- Lecture notes for students (pdf): Slides in 4-page handouts
- Assignments, Project descriptions
- Interaction channels:
  - MS Teams
  - https://github.com
    - Add to your project member: trangntt.for.student@gmail.com

Naming convention

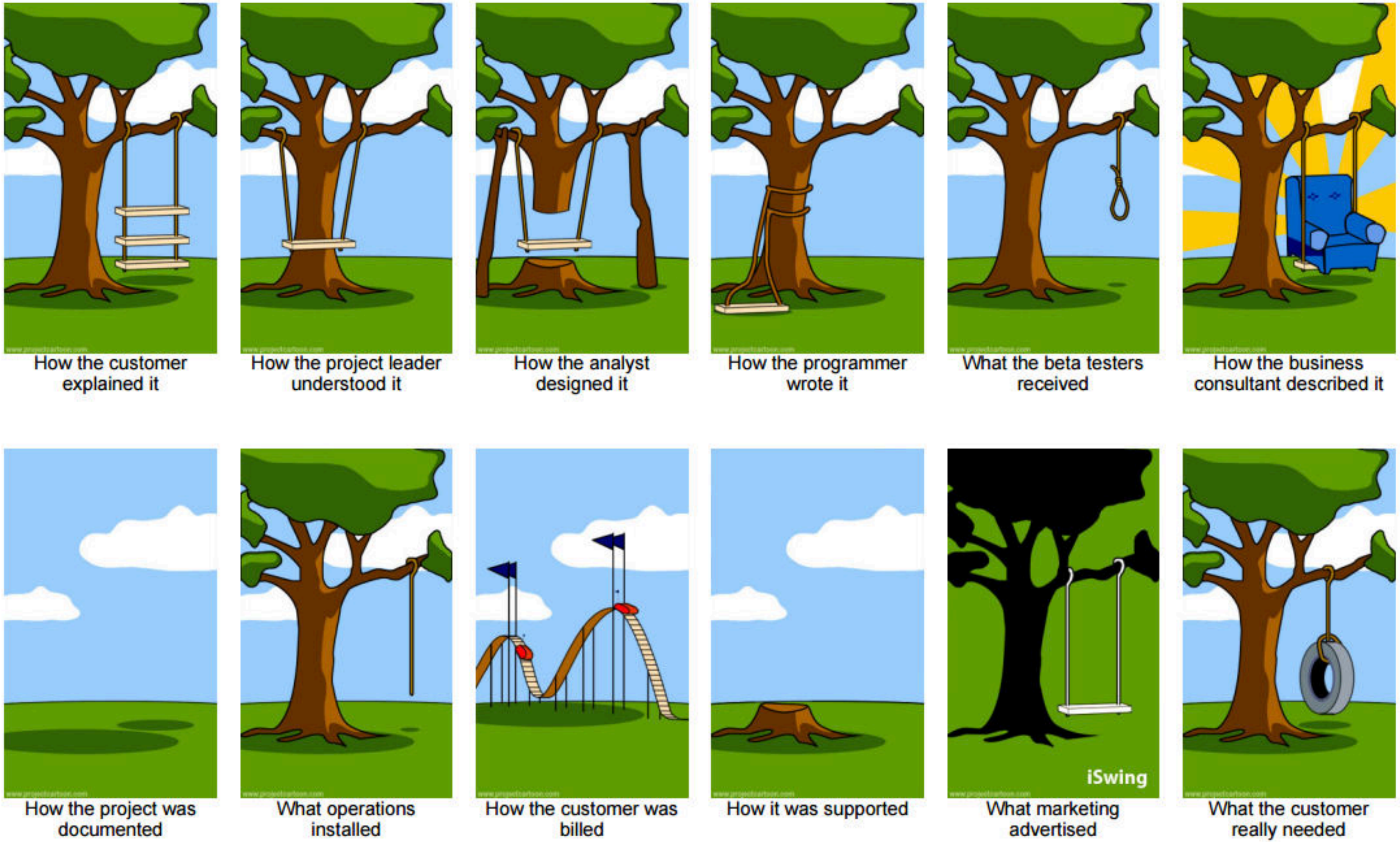
- Naming your private individual repo (Lab)
  - MSTeamName-StudentId.StudentName
  - E.g. ISD.ICT.20221-20191234.NguyenVanHa

    or ISD.VN.20221-20199283.VuHuongLan
- Naming your private group repo (Capstone Project)
  - **TeamName-GroupNo**
  - E.g ISD.ICT.20221-05 or ISD.VN.20221-10
- Monitor?

Introduce yourselves

- Full name
- Experience in Computer Science
  - Operating System
  - Programming Languages
  - (Mini-)Projects
  - …
- Strength / Weakness
- A course you like best / hate
- Desire to study in this course

What is the real software to be built?



How the customer explained it

How the project leader understood it

How the analyst designed it

How the programmer wrote it

What the beta testers received

How the business consultant described it

How the project was documented

What operations installed

How the customer was billed

How it was supported

What marketing advertised

What the customer really needed