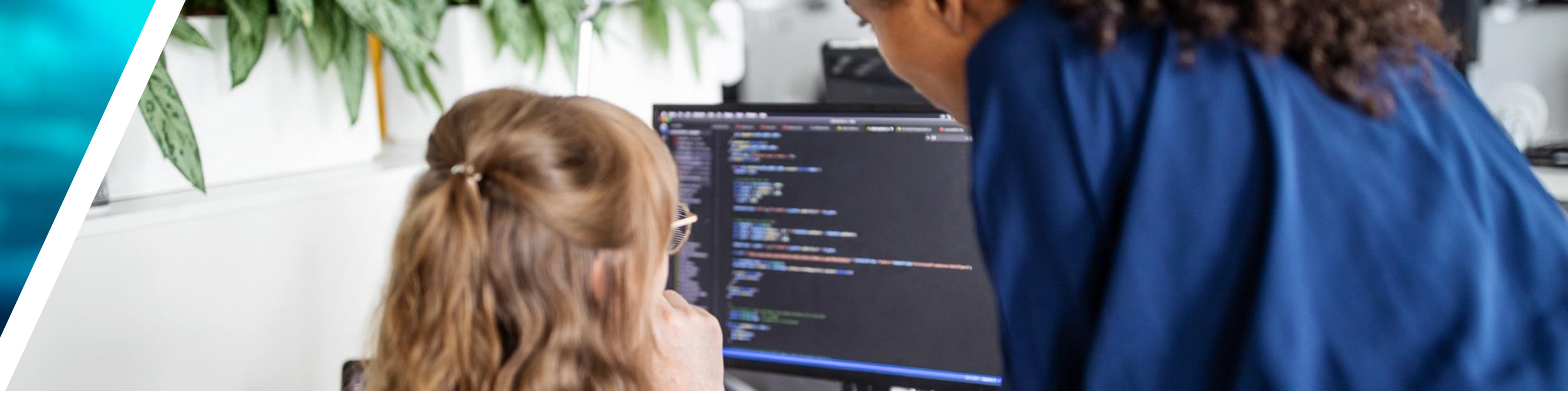


A TOUR OF SOFTWARE DESIGN

CSC207 SOFTWARE DESIGN

Course Overview

Tools (Weeks 1-4)

- Java
- Version Control
- Software Tools

Design (Weeks 5-8)

- Clean Architecture
- SOLID
- Design Patterns

Professional Topics (Weeks 9-12)

- Ethics
- Internships
- GenAI

- This week, you will set up Java, Git, and IntelliJ on your personal computer
- This week, we will talk about
  - The software development lifecycle
  - What Java classes look like
  - Constructors in Java
  - Version control and Git

SOFTWARE DEVELOPMENT TEAM

- Developer (you): build the product
  - Design the architecture
    - How the parts of the program will be organized
    - Where persistent data is stored
    - How data passes between the parts of the program
  - Create the screens
    - Match a high-fidelity prototype that someone else created
    - Add functionality (what happens when a button is clicked?)
  - Test

Stuff to ponder

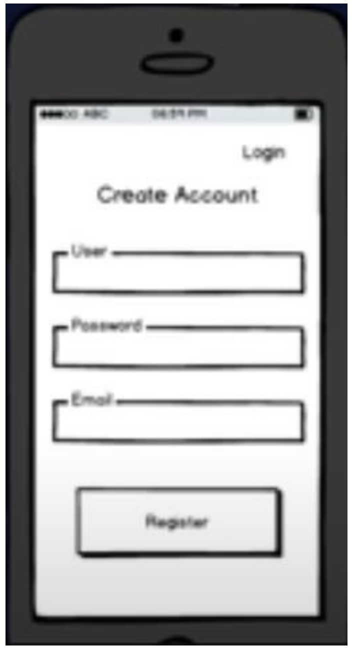
- How do developers know what user interfaces to create?
- How do developers know what data to keep track of?
- How do developers know that they are creating what the client wants?

SOFTWARE DEVELOPMENT TEAM

- **Product manager**: mini-CEO for a project
  - High level focus: understand client needs, turn their idea into reality
  - Stakeholder management: dev company, client, end users, dev team
  - Product success: define Minimum Viable Product (MVP), measure success (user surveys, client interviews), fine tune the product
- **Project manager**: in charge of dev team day-to-day details
  - Identifies *use cases*: what will users need to do with the application?
  - Understand high-level requirements, translate to step-by-step dev plan
  - Liaise between product manager, stakeholders, and dev team

SOFTWARE DEVELOPMENT TEAM (CONTINUED)

- **Designer**: User eXperience (UX) and a pretty User Interface (UI)
  - UX — how the user uses the app, navigating between screens
    - Draw high-level wireframes: focus on usability and user flow
      - No colours or other such details
    - Can the end user accomplish all the use cases?
    - Allows software developers to start planning
  - UI — Draw high-fidelity prototype
    - Based on wireframes, create fully-branded UI
    - Hand to devs to create

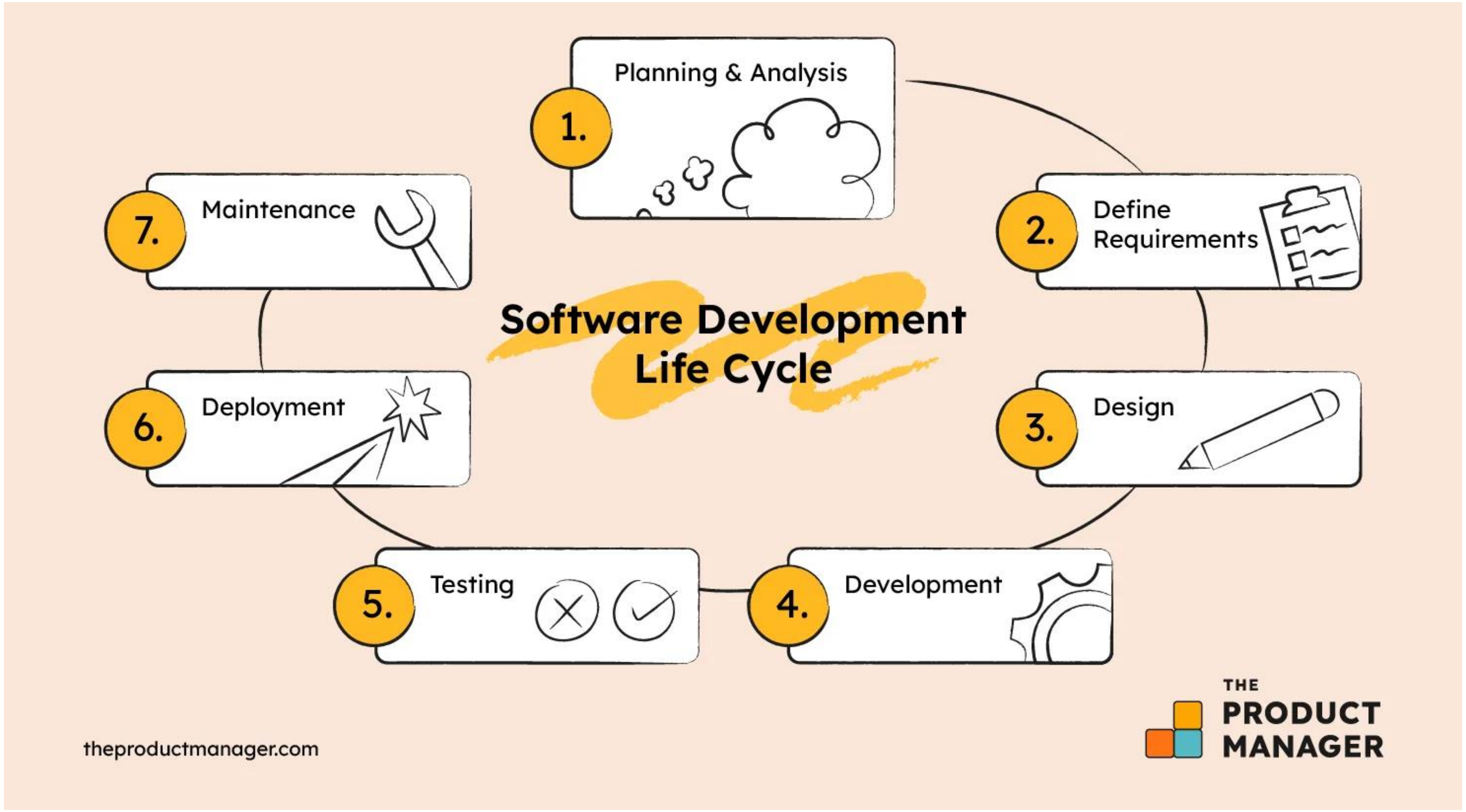


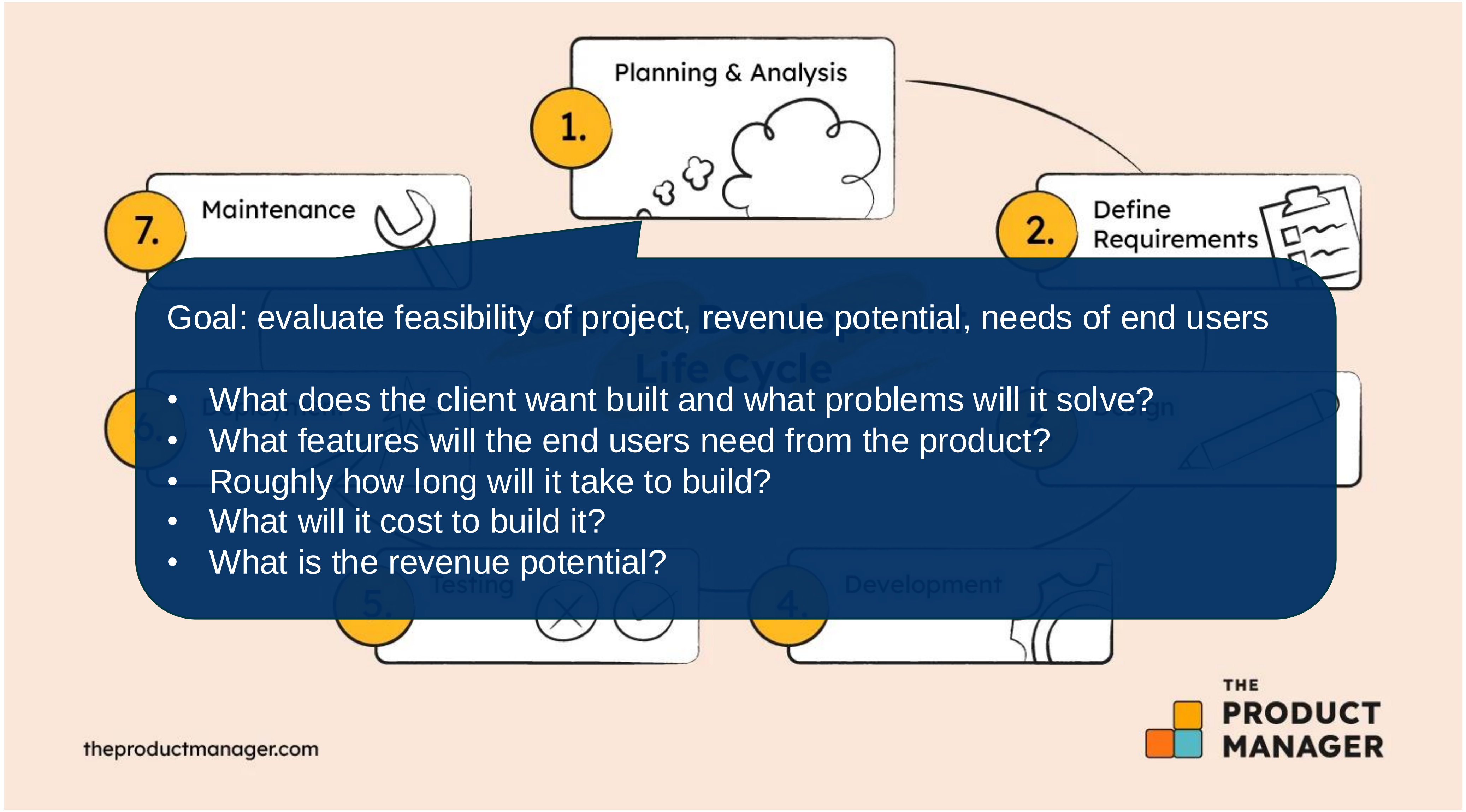
https://youtu.be/CbIMO5EcCD8

Computer Science
UNIVERSITY OF TORONTO

SOFTWARE DEVELOPMENT TEAM (CONTINUED)

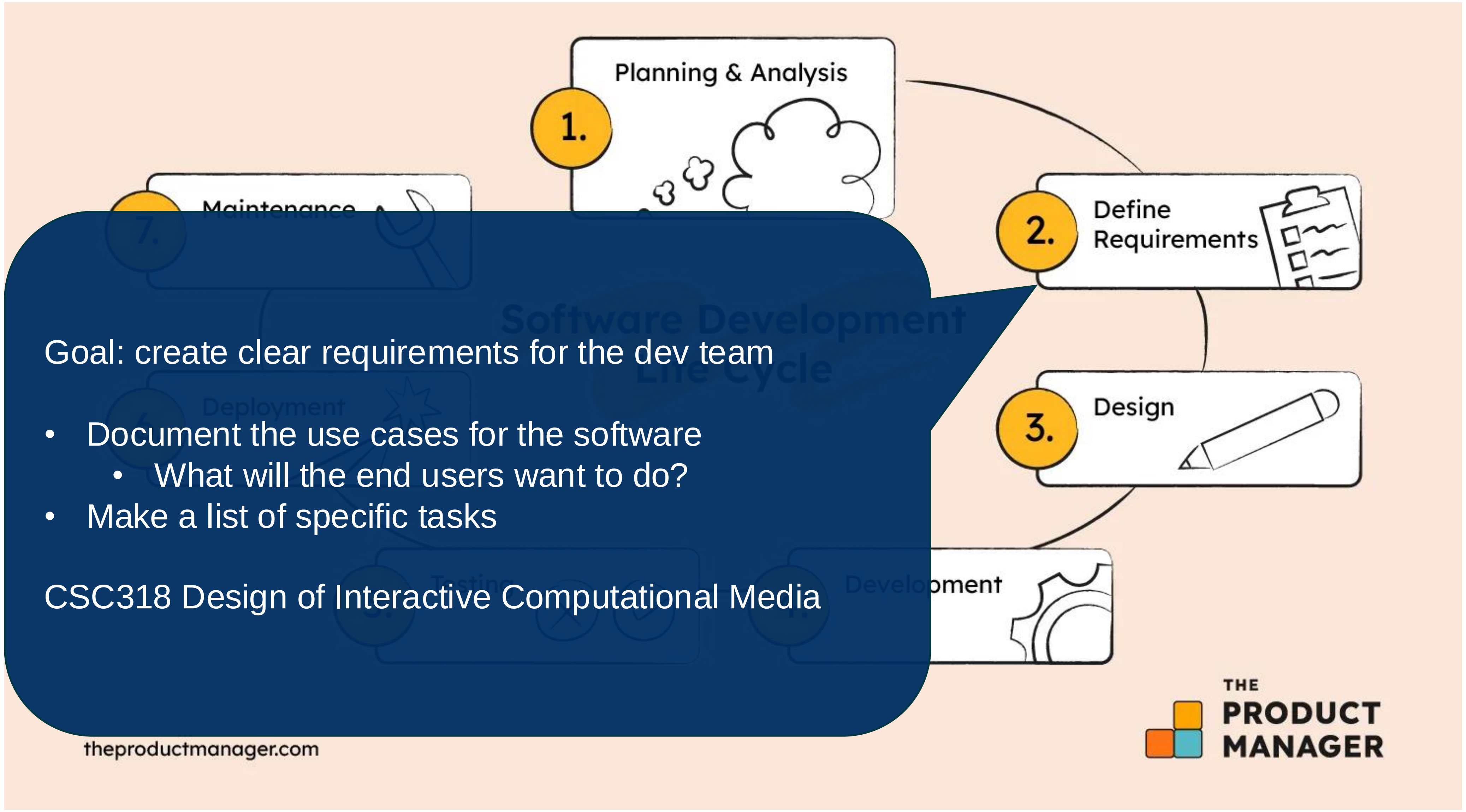
- Quality Assurance: test the product (also you)
  - Review specification and ensure adherence
  - Test software on different browsers, screen sizes, network conditions
  - Try to break the software!


Software Development Life Cycle
1. Planning & Analysis
2. Define Requirements
3. Design
4. Development
5. Testing
6. Deployment
7. Maintenance
theproductmanager.com
THE PRODUCT MANAGER



Goal: evaluate feasibility of project, revenue potential, needs of end users

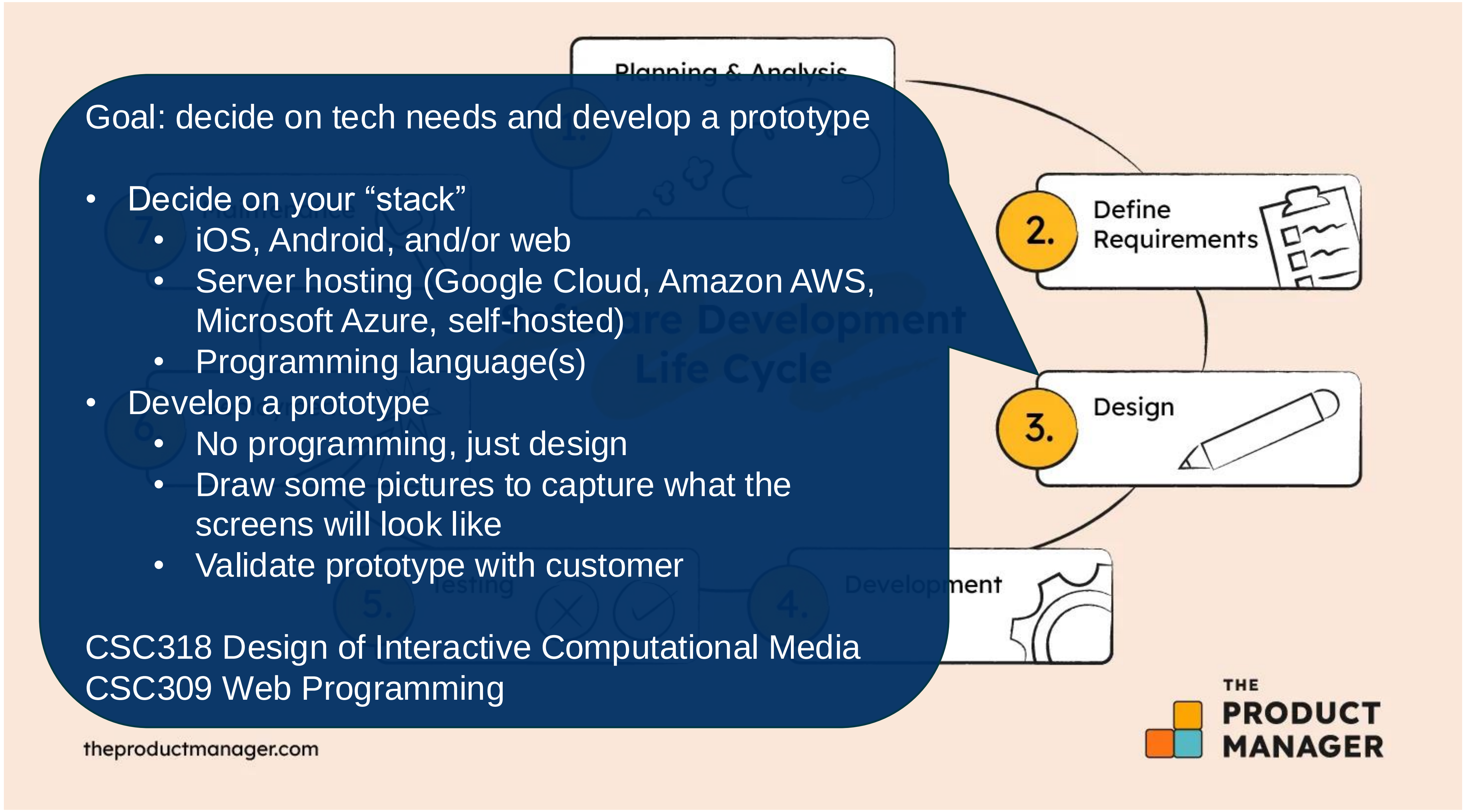
- What does the client want built and what problems will it solve?
- What features will the end users need from the product?
- Roughly how long will it take to build?
- What will it cost to build it?
- What is the revenue potential?



Goal: create clear requirements for the dev team

- Document the use cases for the software
    - What will the end users want to do?
- Make a list of specific tasks

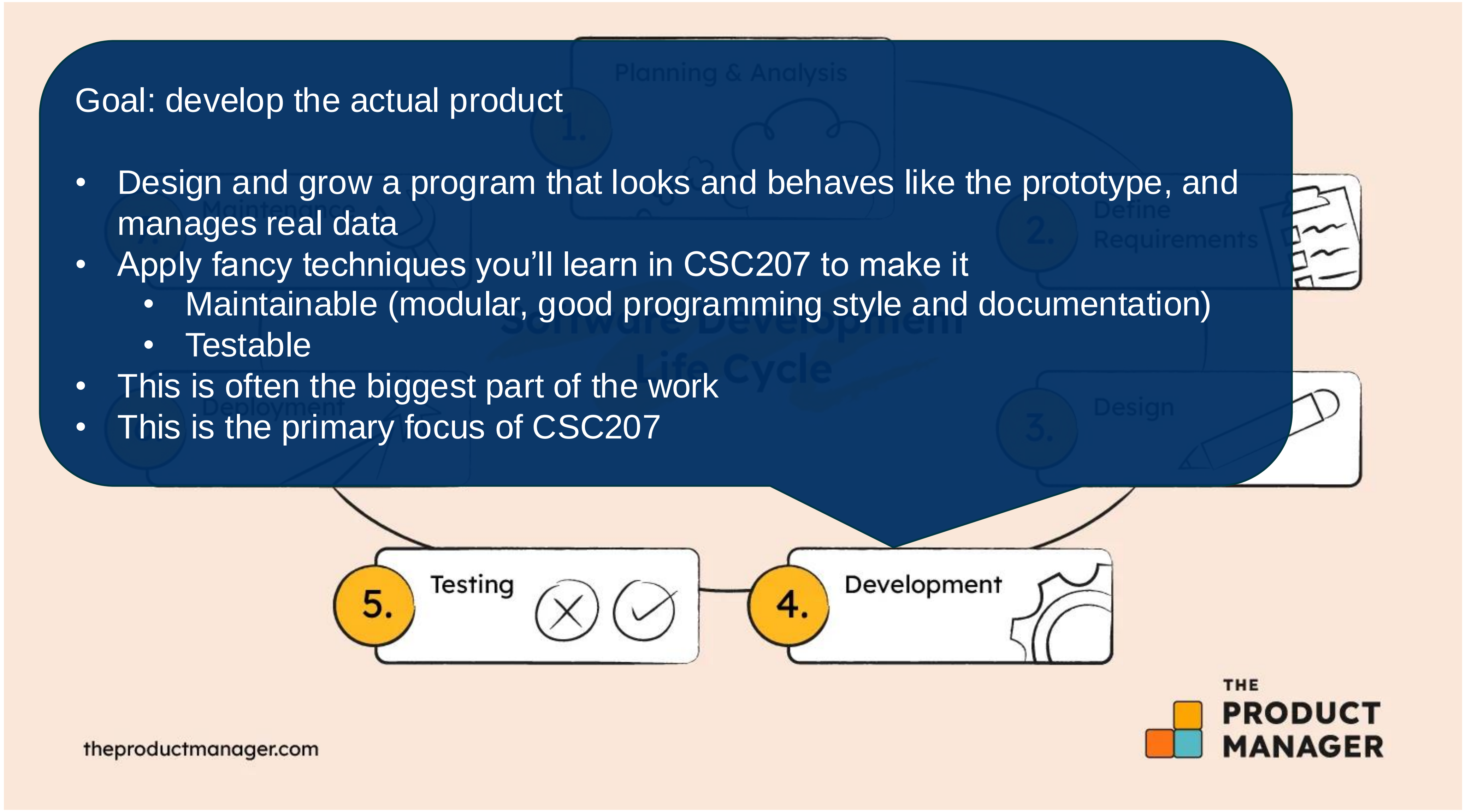
CSC318 Design of Interactive Computational Media



Goal: decide on tech needs and develop a prototype

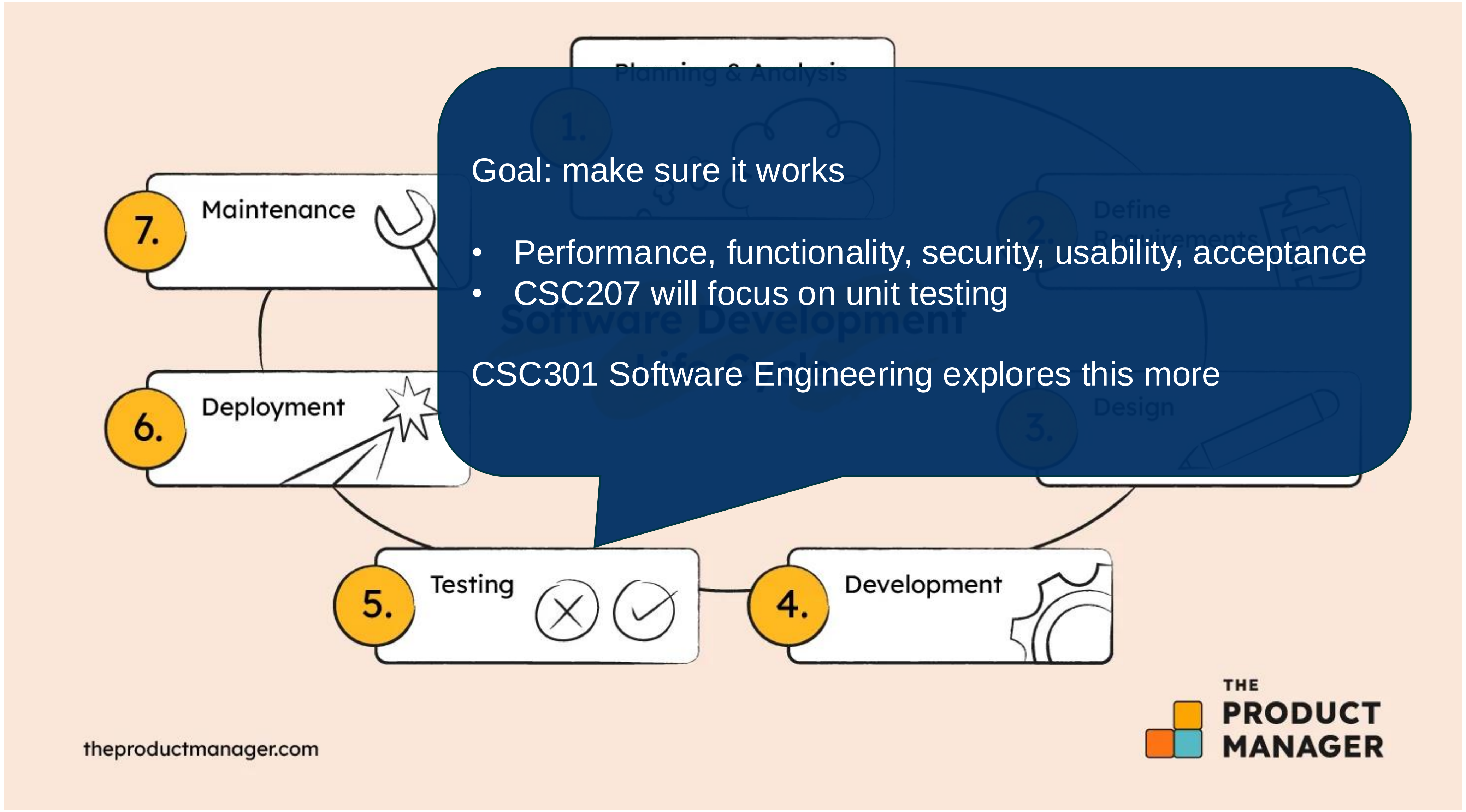
- Decide on your "stack"
  - iOS, Android, and/or web
  - Server hosting (Google Cloud, Amazon AWS, Microsoft Azure, self-hosted)
  - Programming language(s)
- Develop a prototype
  - No programming, just design
  - Draw some pictures to capture what the screens will look like
  - Validate prototype with customer

CSC318 Design of Interactive Computational Media
CSC309 Web Programming



Goal: develop the actual product

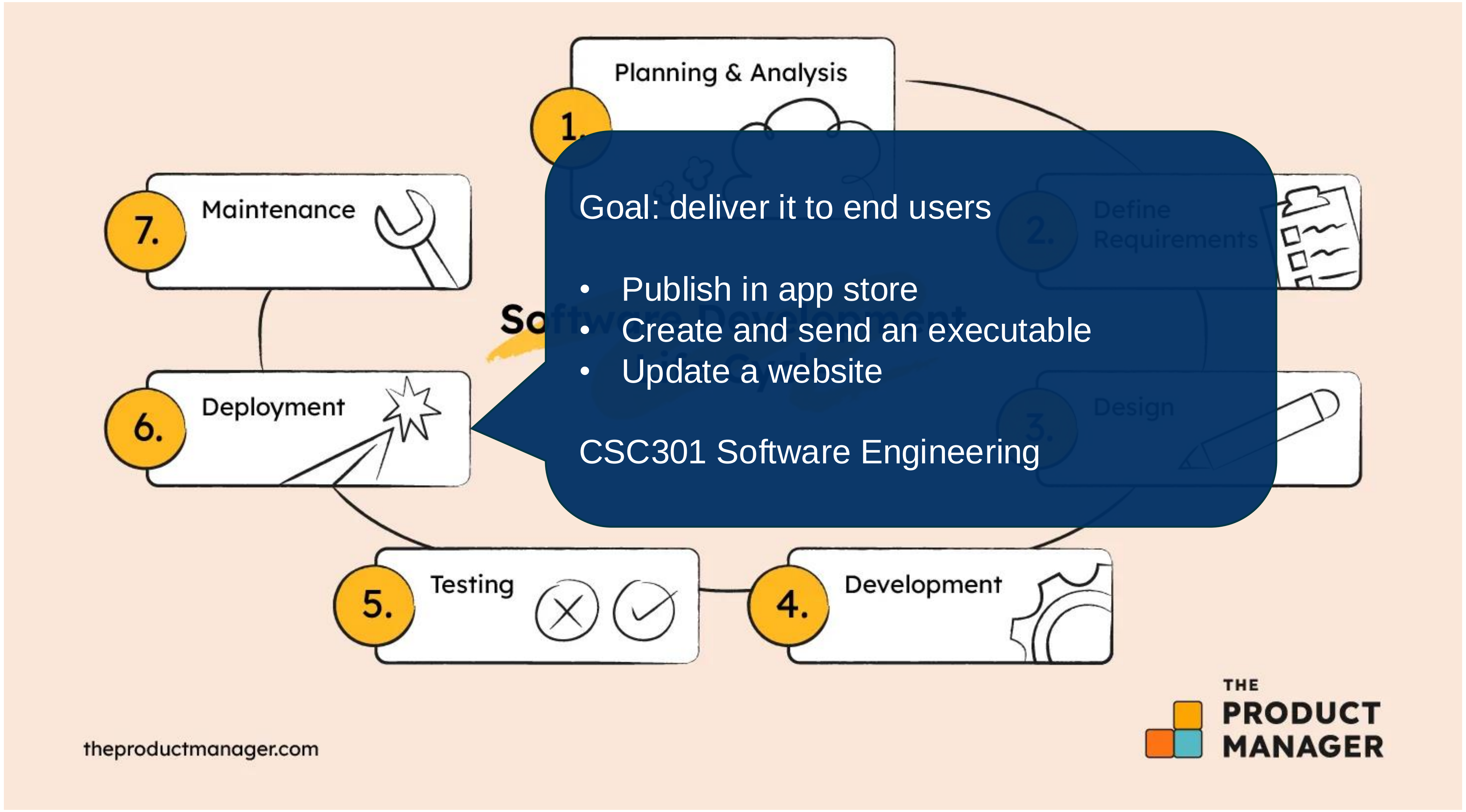
- Design and grow a program that looks and behaves like the prototype, and manages real data
- Apply fancy techniques you'll learn in CSC207 to make it
  - Maintainable (modular, good programming style and documentation)
  - Testable
- This is often the biggest part of the work
- This is the primary focus of CSC207



Goal: make sure it works

- Performance, functionality, security, usability, acceptance
- CSC207 will focus on unit testing

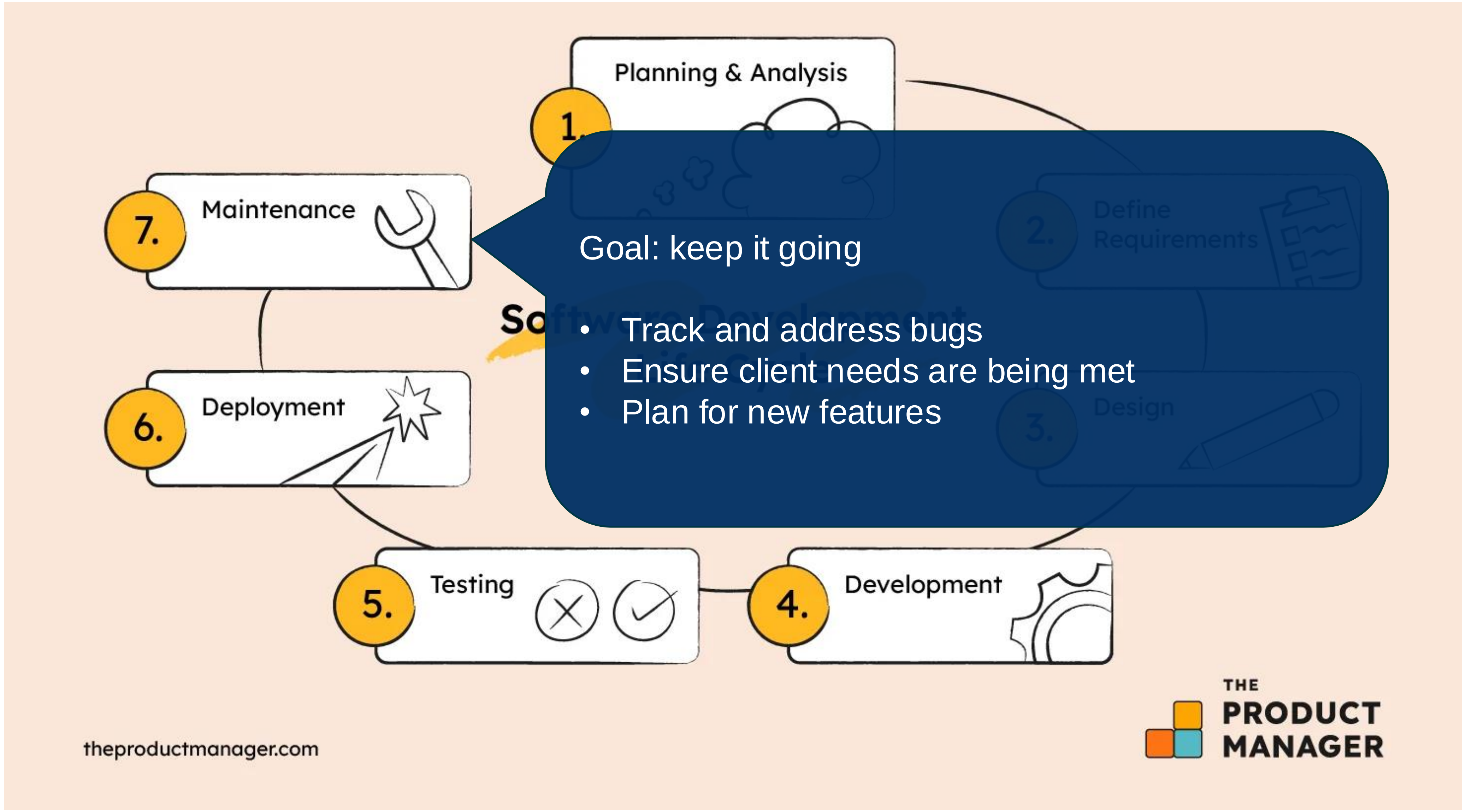
CSC301 Software Engineering explores this more



Goal: deliver it to end users

- Publish in app store
- Create and send an executable
- Update a website

CSC301 Software Engineering



Goal: keep it going

- Track and address bugs
- Ensure client needs are being met
- Plan for new features

Questions to be answered this week...

- What is the structure of a Java class? (declaration, variables,…)
- What are the possible accessibility modifiers and primitive types in Java?
- What is version control? What are some benefits of using it?
- What do these git commands mean?
  - clone, pull, add, commit, push, branch, merge, status, checkout

CSC108/148/110/111 STUFF YOU KNOW

- value and type; expressions
- naming a value using an assignment statement (assigning a value to a variable)
- control flow: sequence of statements, if, while, for, function call, return statement, call stack, recursion
- ADTs and data structures: string, list, dictionary, linked list, stack, queue, tree
- classes and the objects they describe; composition; inheritance (OOP)
- some variables and methods are private (Python: use a leading `_underscore`)
- computational complexity (big-Oh)
- unit testing, debugging
- function and class design recipes — processes by which to write code