WASHINGTON STATE UNIVERSITY
TRI-CITIES

SOFTWARE ENGINEERING

Introduction

Luis de la Torre

Spring 2023

IF YOU'RE GOING TO LEARN ANYTHING IN THIS CLASS

Writing a program is easy

Writing software takes practice

KEY CONCEPTS

- Software engineering …
  - ✓ is a discipline
  - ✓ helps controls costs (time + money)
  - ✓ is beyond writing programs
  - ✓ provides framework for managing your projects
  - ✓ will help you deliver higher quality tools

OVERVIEW

- What is Software Engineering?
- Why?
- Industry Diversity
- Intellectual Property
- Ethics
- Who – Project Teams – Players in Software

WHAT IS SOFTWARE ENGINEERING

- Discipline
  - Produce software products
  - Support lifespan of software product
- Journey from specification through support
- Not just a process – but all the rest – project management, tooling
  - (you'll learn git – you'll be introduced to some industry accepted tools)

WHY?

- Software… is abstract
- Software… is ubiquitous
- Software… systems are complex

- Without SE – we are just writing programs

- We want to deliver products **on time**, to our customers' **needs**, and of **high quality**

INDUSTRY DIVERSITY

- Different applications require different approaches
  - Medicine
  - Flight Control
  - Mobile Gaming
  - Embedded Control
  - Standalone Apps
  - Web Apps
  - Data Collections
  - Scientific Simulations

- Key Buzzwords these days
  - Web App
  - IoT
  - Cloud
  - Crypto and Blockchain
  - AI/ML
  - Security
  - Edge Computing
  - Tools / Frameworks
    - React
    - node
    - javascript
    - WPF/UWP
    - JSON
    - Hadoop

WHAT IS THE BEST LANGUAGE?

- C++
- Python
- C#
- Java
- Javascript
- Ruby / Ruby on Rails
- HTML
- Objective-C
- PHP
- C
- LISP
- SQL
- SWIFT
- GO

CHALLENGES

- ✓ The point is, our field is growing – and each problem – each industry – requires us to select the right tools, and different approaches to tackle each unique problem

- ✓ However, there are patterns to solving problems from both a project management side and a technical implementation side

FRAMEWORKS

- ✓In all approaches we try to identify patterns
- ✓We try to maximize re-use (e.g. frameworks)
  *Web development is a prime example

- ✓The use SE practices is core when developing frameworks that may be consumed by other software suites

ETHICS

You are going to be faced with ethical challenges in your career

EXAMPLES OF ETHICS

Value Issue

Privacy Handling, storing, sharing user data only under the circumstances and for the purposes that the user sets

Sustainability

Energy consumption of the software artifact, caring about energy throughout the SE process and in the documentation

Transparency

Transparent decision-making procedures of intelligent systems, publicly available ethics policies by software development organizations

Diversity

Gender, race, and age distribution of professionals in a development team

ETHICS

- Confidential – keep stuff private
- Competence – don't oversell yourself
  - "Yea I know machine learning"
- Intellectual Property Rights - Who owns what?
- Misuse – yes, you can write a virus but is it a good thing? Should you really use your work computer to play games and watch movies?

WHO?

- Clients
  - Request the software be built
- Project Managers –
  - Help organize activities, schedule, understand and analyze costs
- Development Team
  - Architects – often put together the large view of the system
  - Developers – implement the code based on a design or set of requirements, can also be part of design and requirements elicitation
- End Users –
  - People – persons who use the software
  - Systems – other software or hardware that may use the system
  - **NOTE: NOT ALL 'END USERS' ARE PEOPLE**

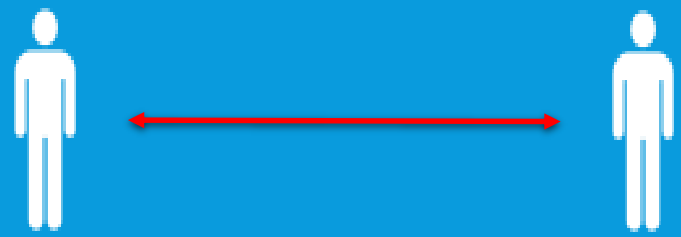
WHO?

- Other Missed People
  - Marketing – people who help market the software to clients
  - Sales – people who work directly with

WHO?

Instructor / Client

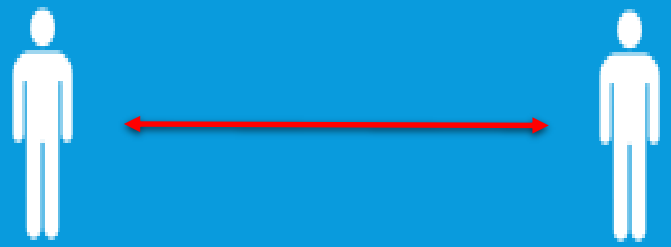
Student



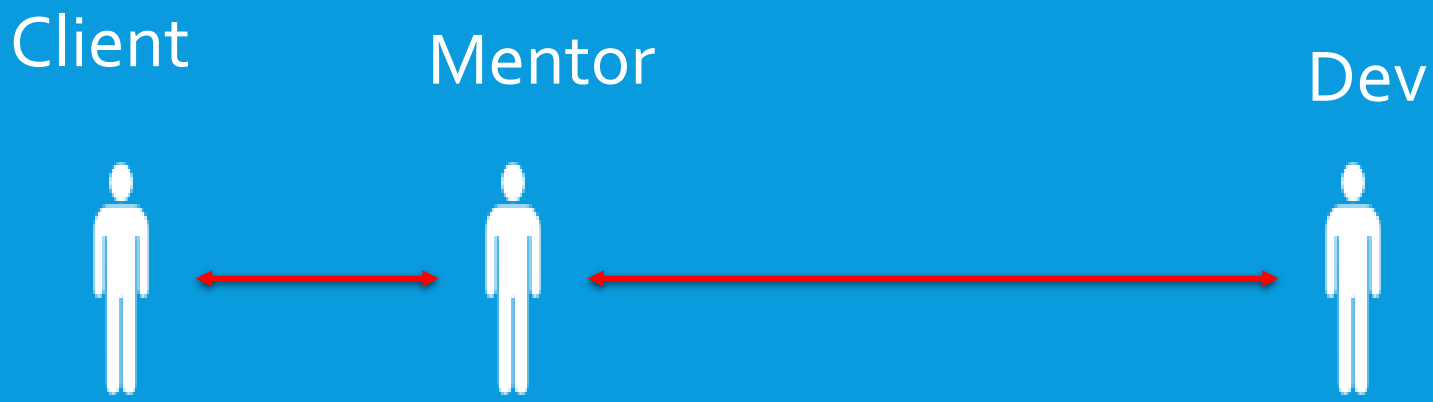
WHO?

Manager / Mentor

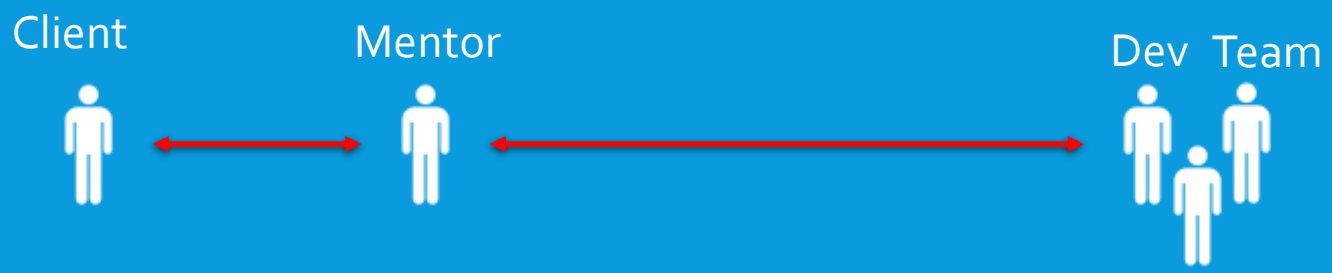
Intern / Entry Dev



WHO?



WHO?



WHO?

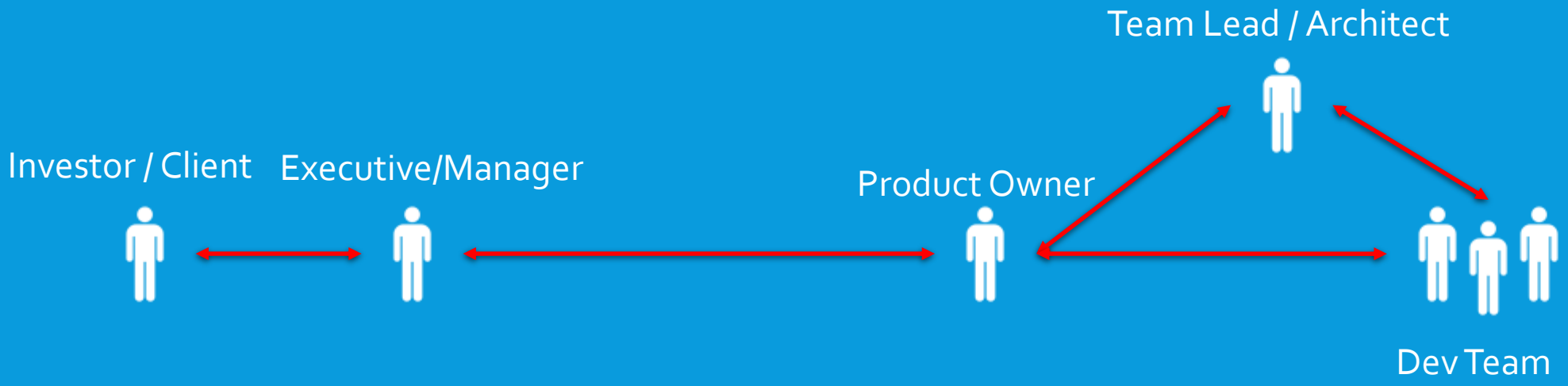
Dev Team

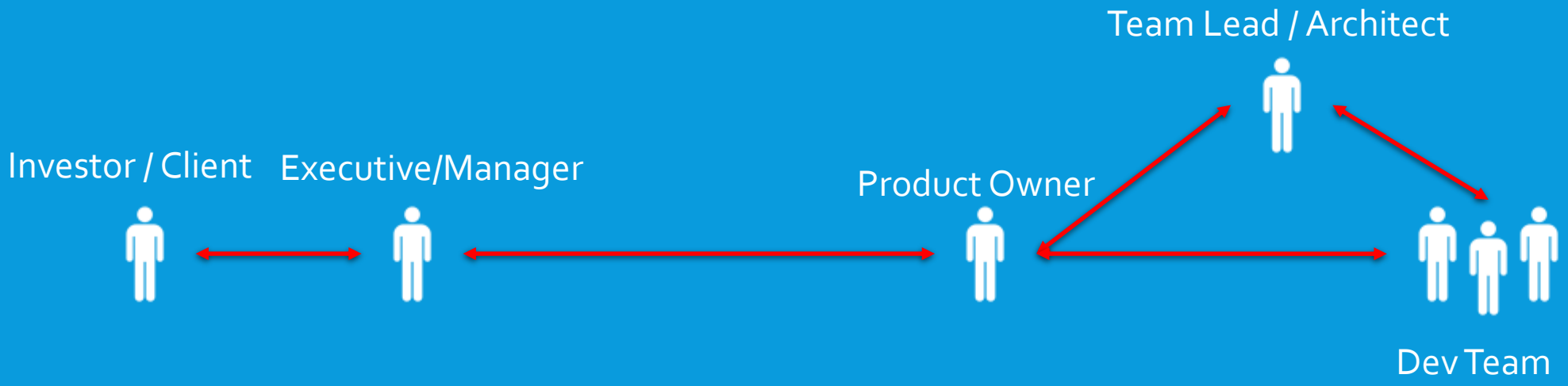
Program Manager

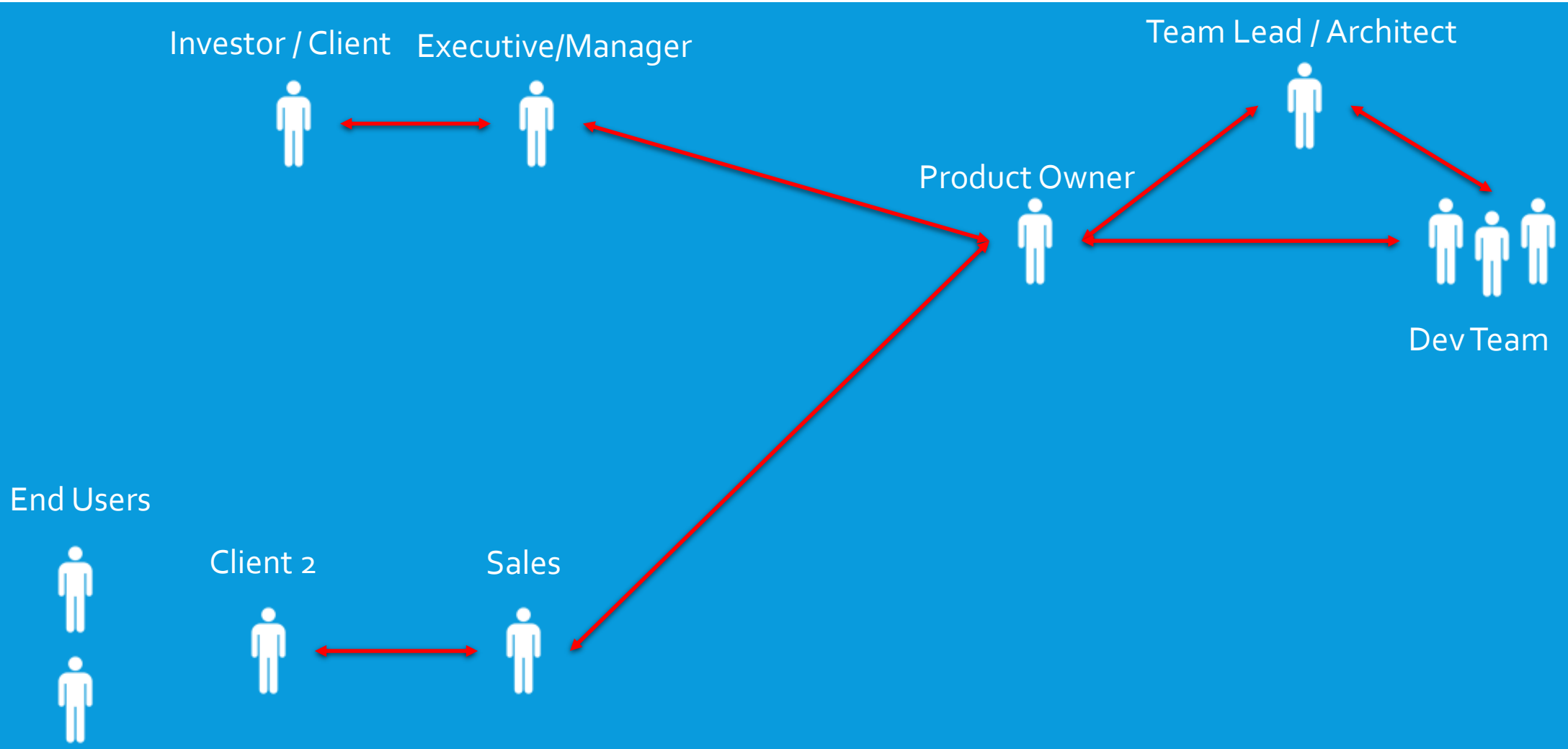
Software Dev Engineer

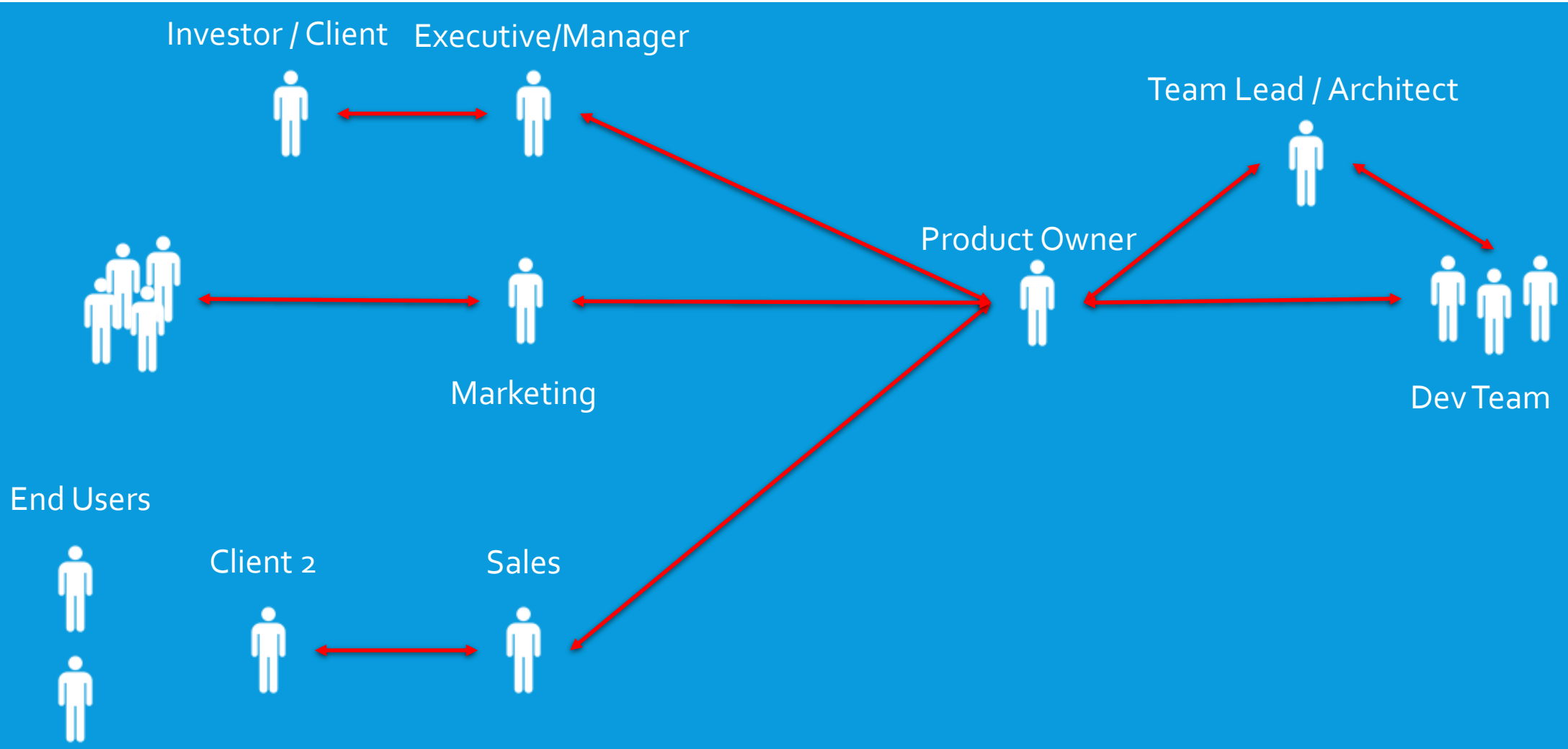
Software Dev Engineer in Test (SDET – Tester)

Tester


Team Lead / Architect
Investor / Client
Executive/Manager
Product Owner
Dev Team


Team Lead / Architect
Investor / Client
Executive/Manager
Product Owner
Dev Team


Investor / Client
Executive/Manager
Team Lead / Architect
Product Owner
Dev Team
End Users
Client 2
Sales


Investor / Client
Executive/Manager
Team Lead / Architect
Product Owner
Marketing
Dev Team
End Users
Client 2
Sales

GAME OF TELEPHONE?

Everyone is going to hear things differently

MARY HAD A LITTLE LAMB

MARY HAD A LITTLE LAMB

…IT WAS SMALL

MARY HAD A LITTLE LAMB

…FOR DINNER

MARY HAD A LITTLE LAMB

…BUT THAT WAS SOME TIME AGO

FROM HERE…

- Processes
  - How to organize your projects, manage tasks, ensure quality
- Requirements
  - How to determine what to build
- Design
  - How to architect what you need to build
- Implementation
  - You will build something, converting your design into a product
- Testing
  - Verifying that you built what was required
- Support
  - Making sure your client gets what they need

QUESTIONS?