

# Product Management

---

CSC301

# Software Engineering Knowledge Matrix

## Known Knowns

Python syntax  
Git basics

## Known Unknowns

Javascript  
Typescript

## Unknowns Knowns

Understanding “if” and “for”

## Unknowns Unknowns

Software engineer career  
(Software crafts, mastery,  
teamwork, communications,  
software delivery, DevOps)

- “Customers care about **their problems**, not your solutions.”

- Dave McClure – Founder of 500 Startups

- “Nobody wakes up saying I want to build some useless software today.”

- David

# Project Status

## 1. For your project, what do you have now?

A bunch of high-level expectations from potential partners (maybe with some code or designs)

## 2. Can you build software with it?

Yes! But it will probably not be what your partner (& users want).

So it will not be used. So... No. You cannot build software with what you have now.

## 3. What should you do then?

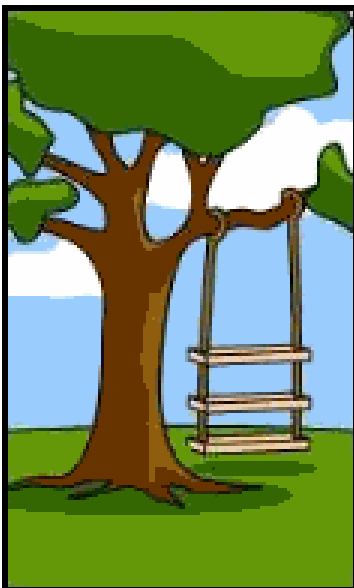
Today we talk about going from vague, high-level project proposals to building software that *is used*

Let's start with a simple case study

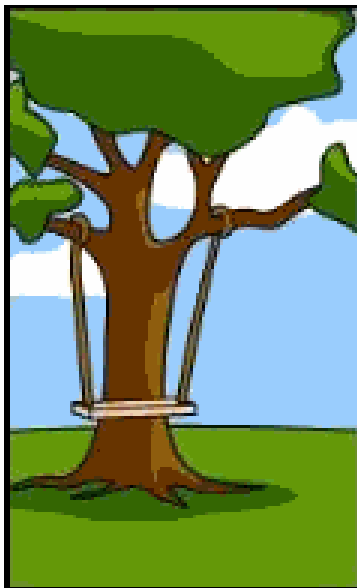
My friend messaged me and asked if I can write a software to automate the process below. How would you do it?

## Process

1. I have a spreadsheet of 200+ clinical studies. I need the start date for all of them.
2. Here's a quick video summary of the problem
3. This is a written down version:
  1. Go to <https://clinicaltrials.gov/ct2/home>
    1. Search for one of the codes, like NCT02711956.
    2. Then click on the link that shows up.
    3. Find start date for the study
  2. Add it to the spreadsheet.



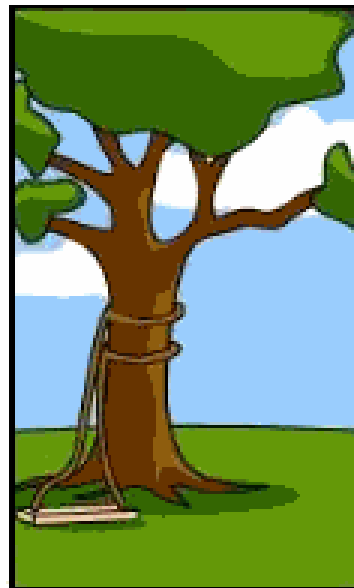
How the customer explained it



How the Project Leader understood it



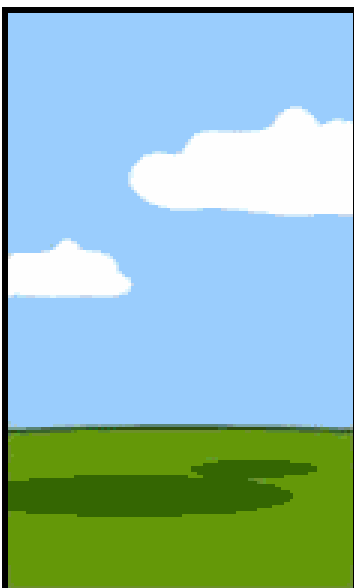
How the Analyst designed it



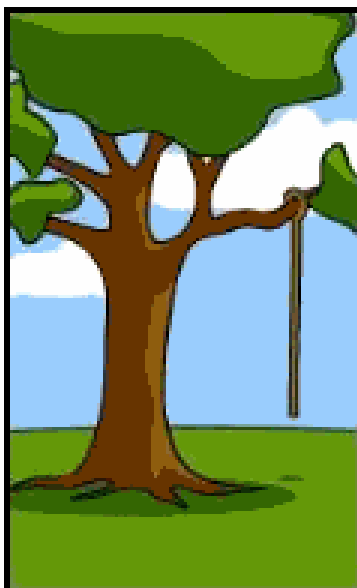
How the Programmer wrote it



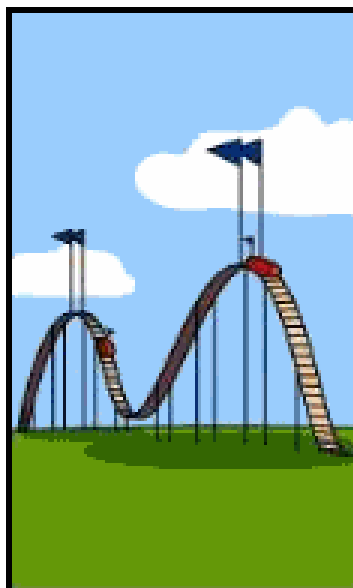
How the Business Consultant described it



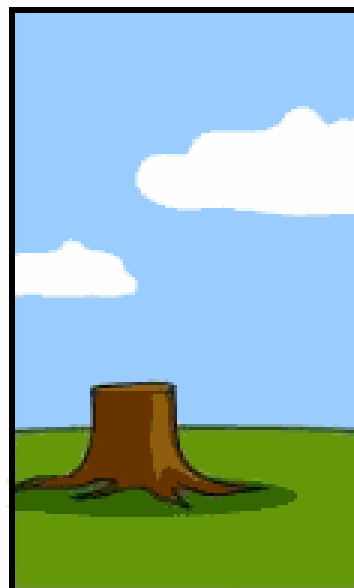
How the project was documented



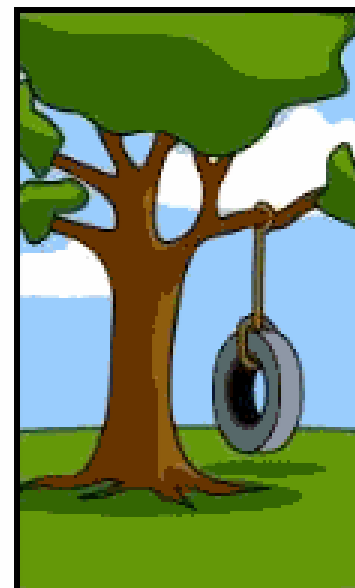
What operations installed



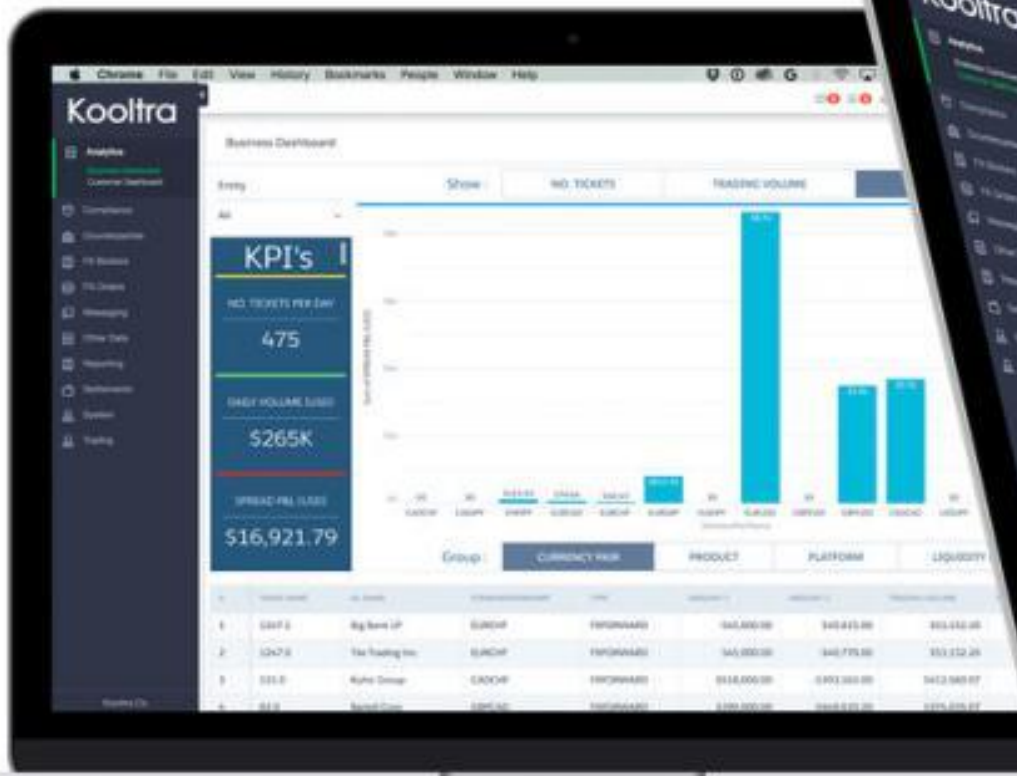
How the customer was billed



How it was supported

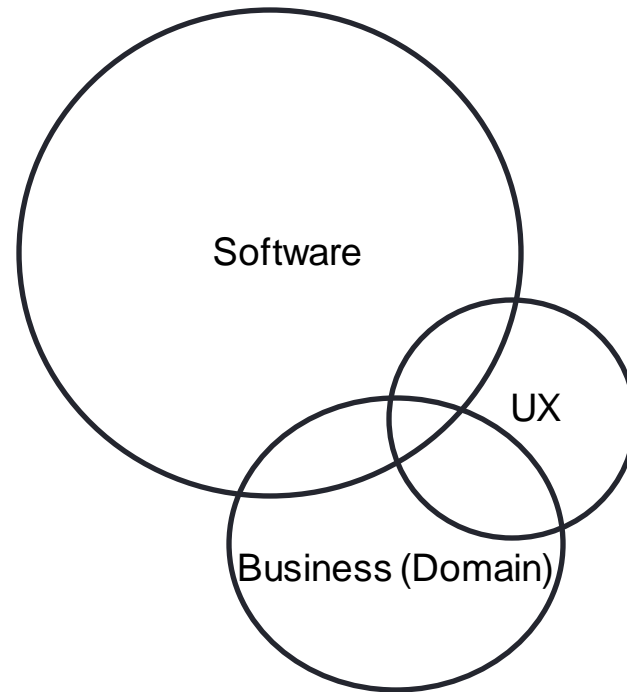


What the customer really needed

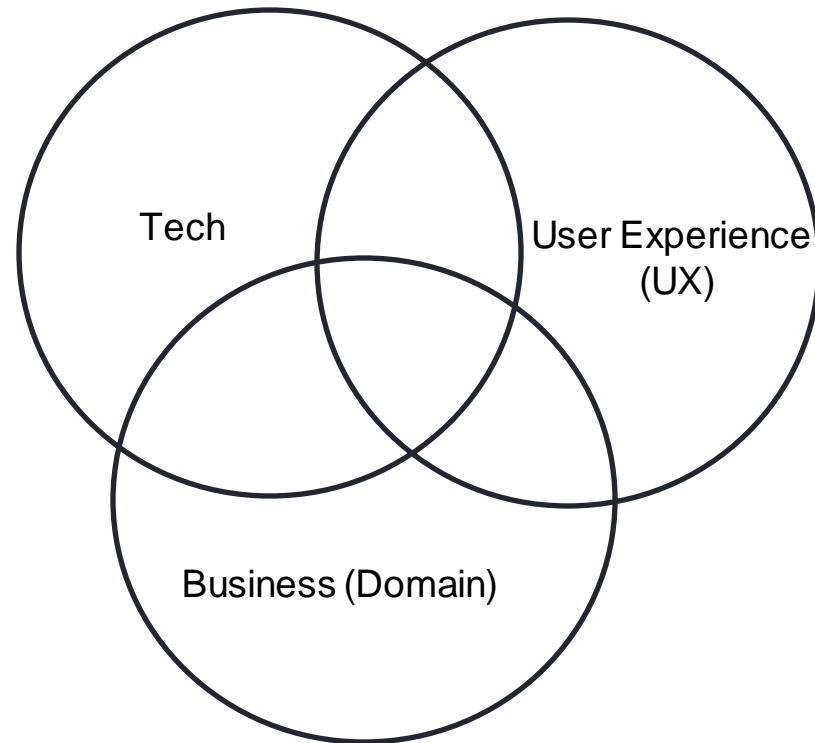




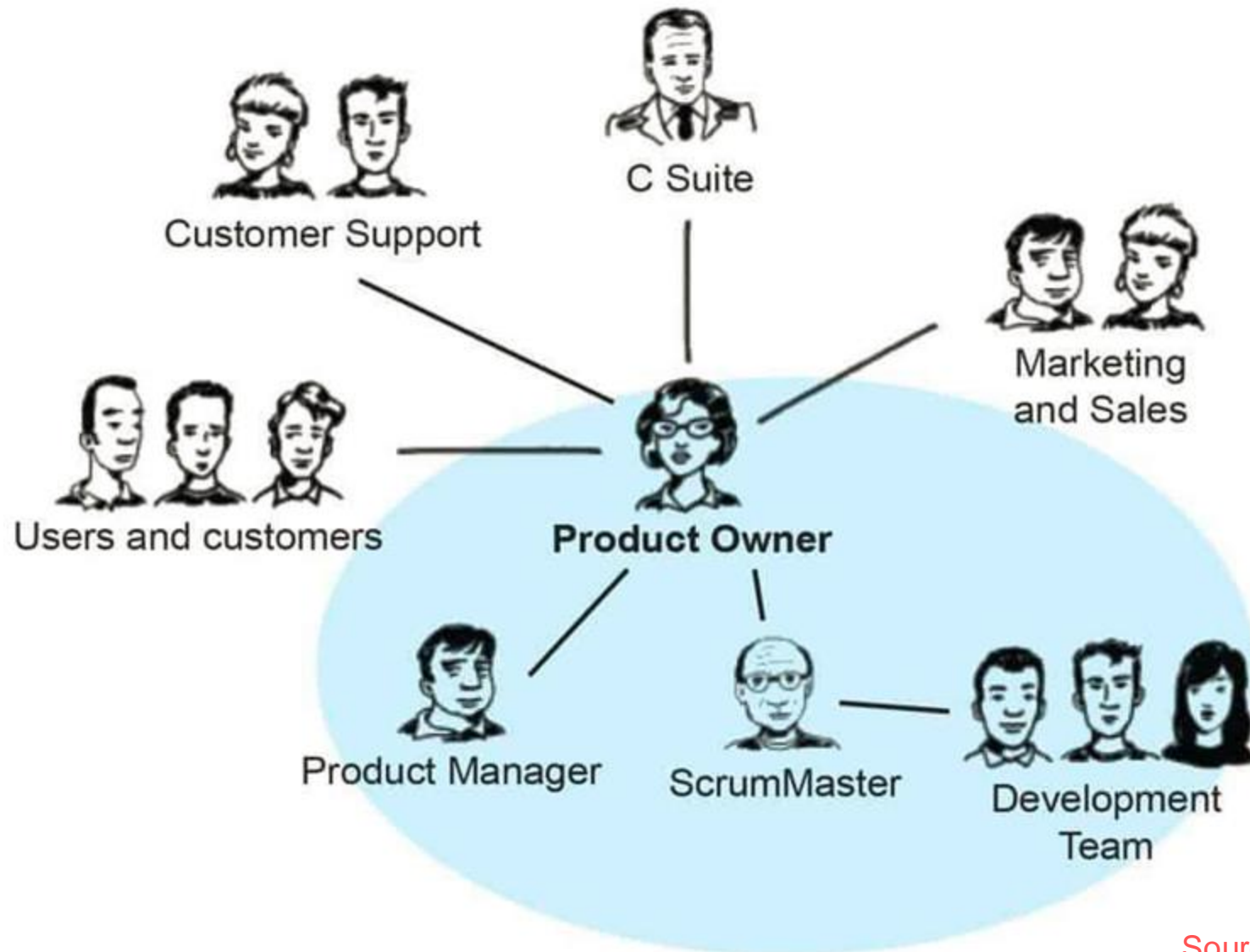
# Your World Today



# What You Need For A Successful Software Project



# Software Engineer Vs Product Manager



[Source](#)



**David Jorjani** 4:35 PM

Hey folks, I'm off tomorrow but will be around if anyone needs me or has questions



[Redacted] 4:40 PM

but...what if I run out of stuff to do?



## Step 1 - **Defining the Product Goals**

- Review your high-level **English description**
  - Ignore technical details
  - Clarify so there is only one interpretation
- **Who** are your **users**?
- **What** is the **objective**?  
**Why** would they use the product?
  - Maybe you can simplify the solution

The **focus** is the **problem and users**,  
not the product.

Does your doctor prescribe medicine  
before diagnosing your problem?

## Defining the Product: **Objectives**

We only talk about the **problem** we want to solve.

We *don't* say what the product is:

*“This should be an Android app that stores my profile details and project interests.”*

What do you do when your partner says:

- “I want an Android app...”
- “I want to build in React...”
- “Use AWS for...”
- “Of course users will...”





Let's look at some of the projects

Activity: **your project's goal**

Let's think about objectives of **your project**

- **Do you know what problem your partner wants to solve?**
- **Has your partner prescribed a solution?**
  - Who prescribes medicine? Doctor or patient?

***Remember:*** We only talk about the **problem** we want to solve. We *don't* say what the product is.

# Customers vs Users

- Users use the product
- Customers pay for it
- Customers (partners) have ***an idea of*** what users want
  - Based on assumptions
  - Following their bias (that they are not aware of)
  - More of a problem in early-stage startups/social ventures

Challenge your partners' assumptions

## Step 2 - **Understanding Users & What They Need**

- *Personas*: Representative **character** (name, picture, background story, etc.) to simplify communication and **empathize** with user groups
  - Examples from our **partner matching application**)
    - David the teacher
    - Mohit the Student
    - Elizabeth the engaged partner

Understanding the job is the first important thing you need to do

# User Stories

- Requirements from the **user's perspective**, using everyday language
  - Follow a basic pattern:  
***As <role>, I want <action/desire>, so that <benefit>***
- For example:
  - As a **partner**, I want to be able to **post proposals for my project**, so that I can **get my project done**.

## User Stories Activity: **your project**

**With your team**, write down a user story for your first persona.

And then, another user story of your choosing.

**As <role>, I want <action/desire>, so that <benefit>**





**Elly** Hoy a la(s) 14:02

**@aquí** I just tried to pull all matching surveys from all Coachees to look at job titles. But the CSV no longer includes job title; it used to. How can I pull the job title of all Coachees in the platform now?

**Barbara** hace 2 horas

Hello **@Elly**. We replaced the Job title along with other fields with the link to the coachee's profile for "Personally Identifiable Information" reasons. The ticket is [here](#).

**David** hace 2 horas

Elly, how do you use the job titles? and how often do you do that?

It's for our CX data. We pulled it once last year for a big look at 2021 data and I want to pull it now to see how many SVPs, EDs, and MDs we currently coach. It's important to assess this as we roll out Executive Coaching.

So it's not a frequent pull by any means but it's an important one.

**David** hace 2 horas

If you show me an example from last year, I can get that for you for this year as we transition to Tableau

**David** hace 2 horas

Or clarify what exactly you'd need in that report

For now, I'll use the 2021 data for see how many of those folks we coached last year. I only need their job title for this.

**David** hace 2 horas

Thanks. Can it be next week or do you need it today? I can do that and it's a good exercise for me :)

**Elly** hace 2 horas

Next week is fine! Thank you!

**David** hace 2 horas

You got it! Thank you

Take a few minutes to develop your project's personas and user stories (at least two)

What are some of the questions you came up with during the exercise?

How do you know when a user story is done?

# Acceptance Criteria

- Acceptance Criteria

- Conditions that a product ***must*** satisfy to be accepted by the user in the user story

- Writing acceptance criteria

- Focus on the problem, not the solution
- Think about how a user will test the user story
  - Implementation-independent and high-level
  - Be specific!
- Remember GIVEN\_WHEN\_THEN? You can use here

# Acceptance Criteria Examples

- *User story:* As a **partner**, I want to be able to **post my project**, so that I can **find teams to work on it**.
- Example acceptance criteria:
  - **New project and details are added to the list**
  - project details are correct and accurate
    - Staff
    - Problem
    - Tech stack
    - ...

## Acceptance Criteria Activity: **your project**

**With your team**, write down the *acceptance criteria* for your first user story.

And then, write it for another user story.

What questions did you come up with?

Ask your partner



## Step 3 – Design Your Software

- Prototype
  - Show what it looks like
  - Show how it works
  - See if your users (and customers) get it and can work with it

# Prototype

- Build absolute minimal
  - Doesn't have to be coded
  - Can “look” like what you intend to build
  - No functionality is needed
- Make assumptions and test them very fast
- Get on the same page with all stakeholders

## Step 3 - Product Design

### ● **Artifacts** (non-code product items):

- **Mockups & Wireframes** - There are many cool tools
  - Figma
  - Invision
  - Moqups
- **CRC cards** (Class Responsibility Collaborator Cards)
- **Class Diagrams** - More detailed than CRC

Let's look at Figma as an example

# Lean Product Management

- Build your Minimum Viable Product (MVP)
- Lean Startup
  - Go from “Can we build this?” to “What’s the minimum we should build to solve the problem?”
  - No time to build everything
  - Optimistic timelines
  - Focus on early/fast rollout
- Build to learn

# Project Tips

- Meet at least bi-weekly with your partner #recipe for success
  - One representative is enough. Definitely not the whole team!
- Use shared/special purpose email instead of personal to setup accounts
  - E.g., [project-wolverine@gmail.com](mailto:project-wolverine@gmail.com) vs [david@gmail.com](mailto:david@gmail.com)
- Use sub-teams and divide responsibility
  - Frontend, backend, database, API, etc.

Your project document and answer the following questions for your project:

1. What's the goal of your project?
2. Who are the users?
3. What do they want to achieve? What are the user stories?

If you don't know the answer to these questions, clarify with your partner

# Mobile Development Options

## MOBILE DEVELOPMENT TIERS





# Sharing Your Mobile Projects

- [Expo](#) or [React Native](#)
- [TestFlight](#) for iOS testing and deployment