Problem Domain and Requirements Elicitation

Presented by Nursultan Askarbekuly

As software developers,
we are engineers *because*
we make useful machines.

How do we make something useful?

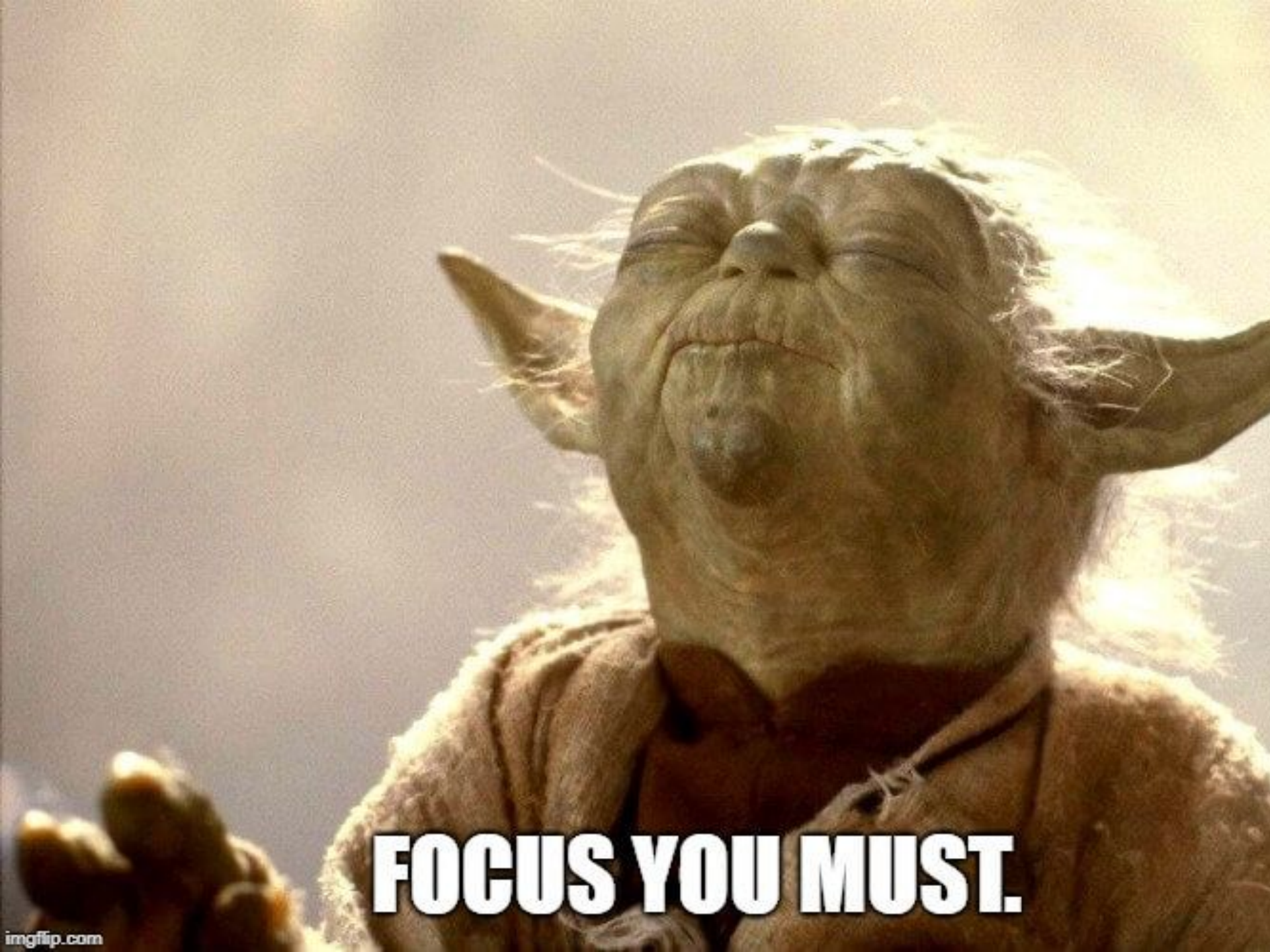
How do we deliver value?

The purpose of a *machine*, which defines its practical value, is located in the *world*.

https://pace.pk/solutions-problems-css-current-affairs/

In a world full of problems; be the solution.

-Iris Watts

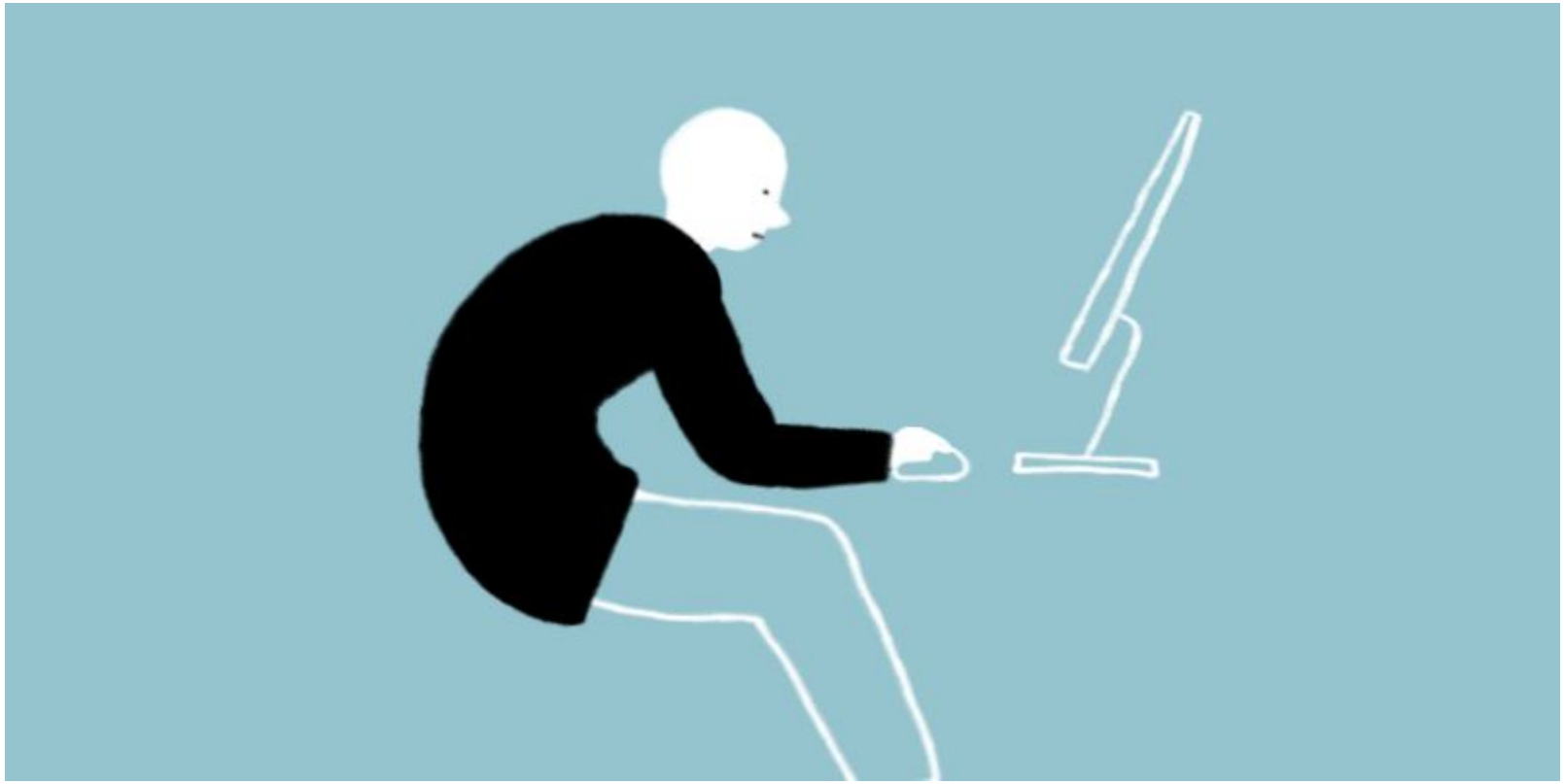

FOCUS YOU MUST.
imgflip.com

domain

an area of knowledge or activity

stakeholders

people who are involved or affected



Strong Posture:
Pomodoro and AI-Powered Reminders

https://strong-posture.web.app/

Sitting is the new smoking

domain

spine health, self-help, personal well-being, pomodoro, trackers, AI

stakeholders

office clerks, students, anyone with desk work, neurologists.

Important questions:

What is the domain of your project?

Who are the key stakeholders?

Understanding the World (a.k.a. Requirements Elicitation)

Presented by Nursultan Askarbekuly

We are concerned both with the *world*, in which the *machine* serves a useful purpose, and with the *machine* itself.

The requirement (problem) is in the world; the machine is the solution we construct.

Discourse about the problem must therefore be discourse about the world and about the requirement that our customer has in the world.

The engineer must understand the properties of the world and manipulate and exploit those properties to achieve the purposes of the system.

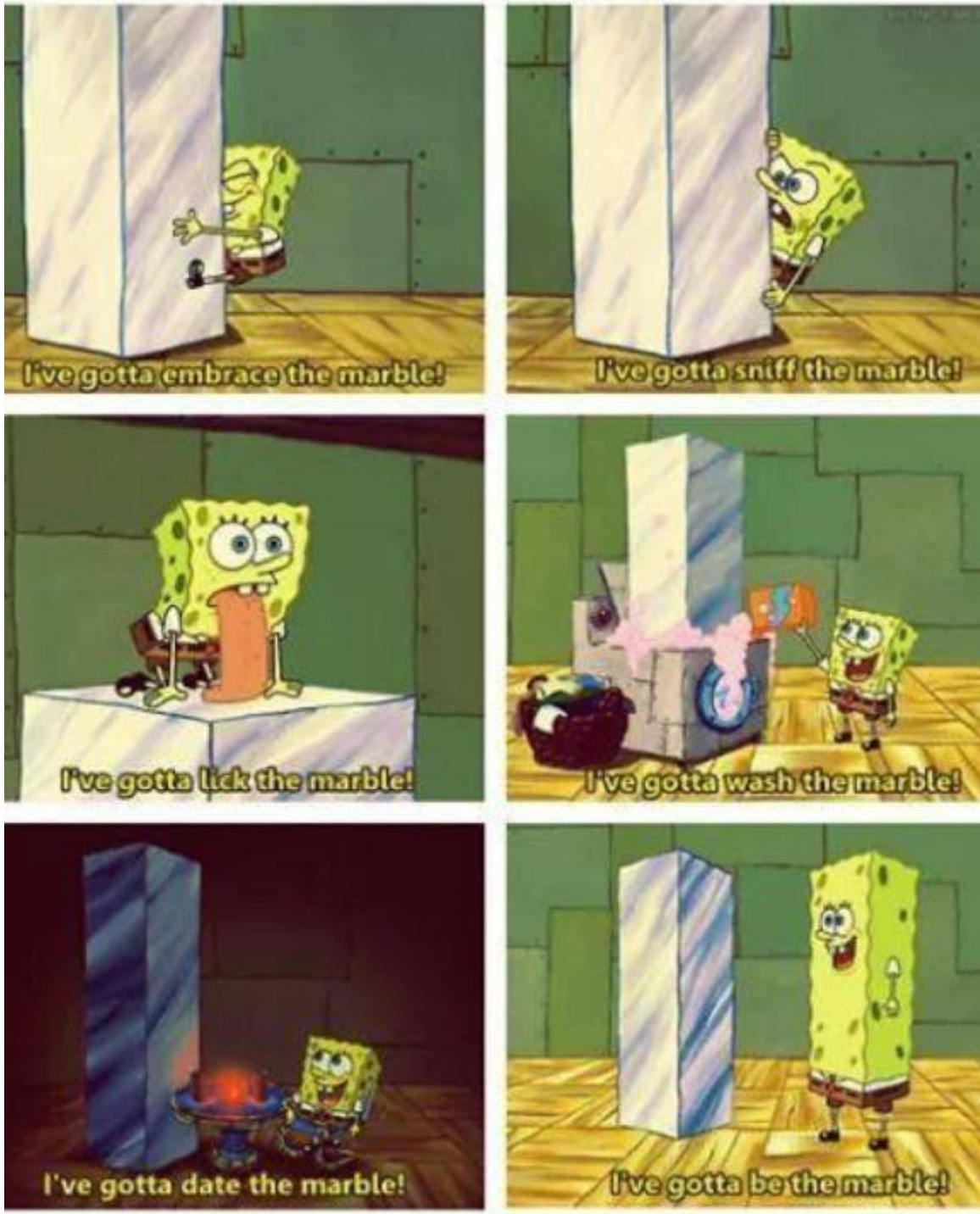
How to explore the domain?

Which stakeholders to talk to?

How to approach them?

Explore the domain:

- talk to people
- read/watch about it
- observe it in real life
- become a part of it
- get the feel for it



The real-world problems are often complex, and demand careful structuring and decomposition;

We want to structure the problem to base the solution on it;

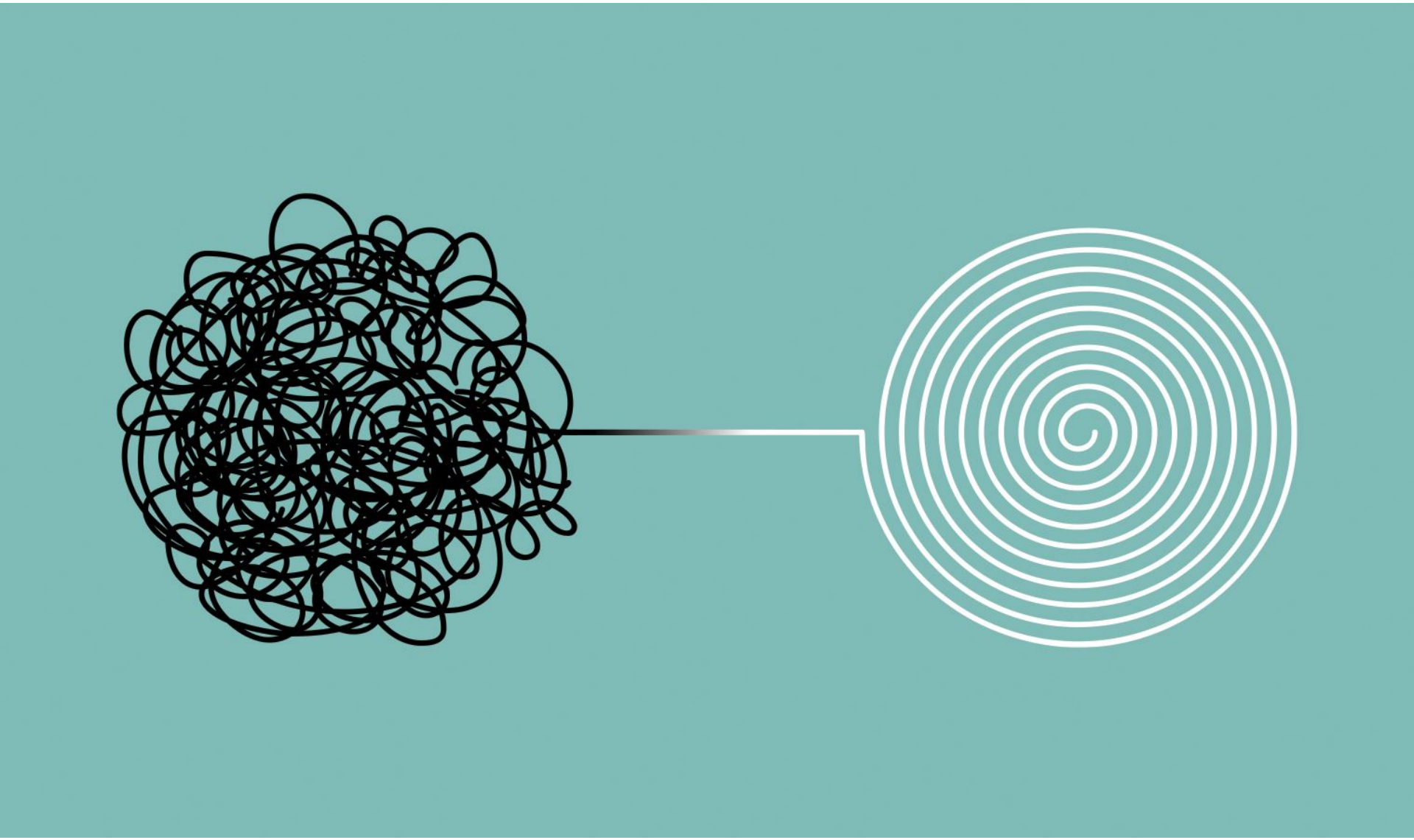


Image source: https://www.davidantrobus.com/2020/problem-or-solution/

Software is a description of a machine. We build the machine by describing it and presenting our description to a general-purpose computer that then takes on the attributes and behaviour of the machine we have described.

Effective problem decomposition means decomposing into problems that are recognised and known to be soluble.

General case: Finding the right stakeholders

- What do I want to learn about?
- Who can give me that information?

- Who has the problem?
- What categories of potential customers are out there?

- Who knows about the domain?
- Should I talk to experts?
- Who is the decision-maker?

- Somebody accessible?
- Somebody who'll provide useful data?
- Bad data is worse than no data.

This course: Customer interview

- Introduce yourself and explain who you are
- Agree on a meeting (duration from 30 to 60 minutes)
- Prepare key questions in advance
- Be on time, come a bit in advance
- The whole team must attend
- Ask the person to describe the general idea
- Ask the questions, and go on tangents if needed
- Take notes or ask to record the interview/zoom call
- Setup a discussion channel, where you can share progress and get feedback

Requirements are concerned solely with phenomena in the world. Our customers want us to engineer effects in the world, not in the machine.

Programs, by contrast, are concerned solely with the machine phenomena. Their purpose is to describe those properties and behaviours of the machine that will, ultimately, satisfy the customers.

The gap between the two is bridged by **specifications**.

Requirements

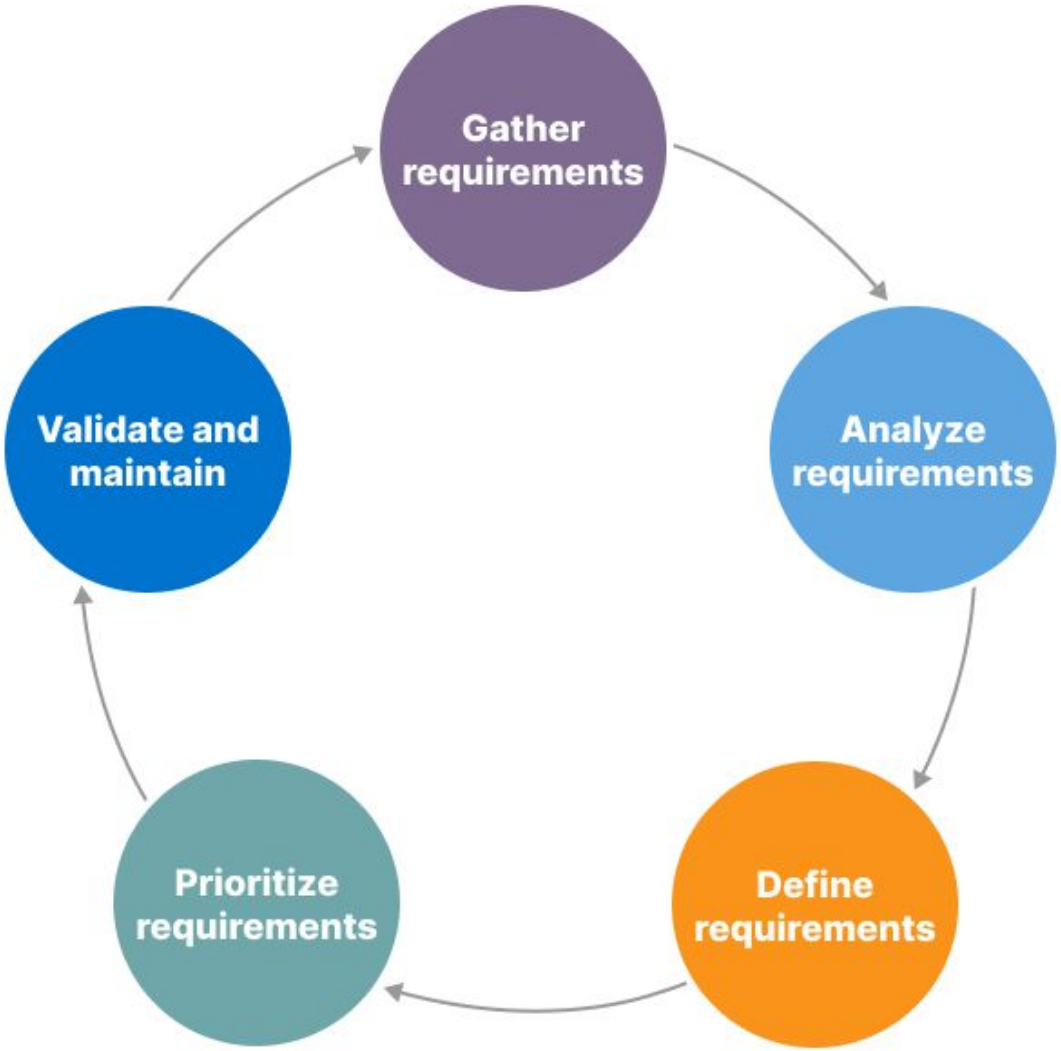
Specification

Design

What is valuable to the customer?

How to build a solution to deliver that value?

Requirements Engineering



Your goal:

Learn the customer needs and domain enough to make informed design decisions.

Meet you customers asap.

Agree on a regular meeting.